

OpenVPN

Şifrelənmiş qapalı şəbəkə

Camal Şahverdiyev

Camal Şahverdiyev

OpenVPN - Şifrələnmiş qapalı şəbəkə

Müəllif: Camal Şahverdiyev

Oxucuya müraciət:

Bu sahə üzrə Azərbaycan dilində kitab ilk dəfə nəşr olunduğundan istifadə edilən termin və sözlər məlumatın daha anlaşıla bilən olması üçün tətbiq edilmişdir. Kitabın daxilində səhv aşkar etsəniz, xahiş edirik, sərt şəkildə tənqid etməyəsiniz. Yalnız söz və ya sintaksis səhvini gördüyünüz halda, bookcorrector@gmail.com mail ünvanına yazmağınız xahiş olunur. Bununla növbəti kitabların daha mükəmməl edilməsinə yardımçı olarsınız.

Bütün müəllif hüquqları qorunur. Kitabın daxilində əks olunan məlumatların yayımlanması, çapı, surətinin çıxarılması və ya digər bir şəkildə istifadə olunması yalnız müəllifdən razılıq alındıqdan sonra mümkündür. Məlumat qeyd olunan məqamlar nəzərə alınmadan istifadə edilərsə, müvafiq qanunvericilik üzrə tədbirlər tətbiq olunacaq.

ISBN: 978-9952-8302-4-8

Kitabdan istifadə qaydaları

Aşağıdakı açıqlamalar kitabın müəllifində oxucuya yardımçı olacaq:

Əsas başlıq - **Bold və böyük hərflər**

Əsas başlığa 1-ci dərəcəli alt başlıq - **Arxa fon qara, şrift ağ**

Əsas başlığa 2-ci dərəcəli alt başlıq - Ümumi mətnlə eyni **font bold** olmaqla

Əmrlər bold qeyd olunub. Əgər hansısa faylın içərisində olan sintaksisdən danışılırsa, öncədən faylın adı və tərkibinə əlavə ediləcək sətirlər bildirilir.

Qeydlər altdan xətt və bold edilmişdir - Qeyd:

- İstənilən UNIX/Linux əməliyyat sistemində faylların içində şərh üçün istifadə edilir.

Simvoldan sonrakı sözlər oxunmur.

/* şərh */ - DNS BIND-da və PHP proqramlaşdırma dilində yazılmış kodlarda göstərilən simvolların daxilində olan istənilən yazı şərhdir.

// - DNS BIND-da və PHP proqramlaşdırma dilində yazılmış kodlarda göstərilən simvollardan

sonra olan ixtiyari yazı şərhdir.

;- DNS BIND-da sətirin sonu deməkdir.

Oxucu tərəfindən kitabın başa düşülməsi üçün tələb edilən biliklər:

1. UNIX/Linux əməliyyat sistemlərində dərin biliklərə sahib olmalı
2. CCNA şəbəkə səviyyəsinə sahib olmalıdır
3. Windows MCITP səviyyəsinə sahib olmalıdır

8 Point-to-Point şəbəkələri

- 9 Qısa işə salma imkanı
- 12 OpenVPN gizli açarları
- 13 Çoxlu gizli açarlar
- 15 Plaintext tunnel
- 16 Routing
- 19 CLI-dan quraşdırma faylları və IP ilə quraşdırma
- 21 Site-to-Site quraşdırması
- 24 3 tərəfli routing

29 Client-server yalnız IP şəbəkələrində

- 30 Public və private açarların quraşdırılması
- 35 Kiçik quraşdırma
- 37 Server tərəfdən route edilmə
- 42 client-config-dir faylların istifadəsi
- 45 Routing: Hər iki tərəfin subnetlərinin route edilməsi
- 50 Default gateway-in yönləndirilməsi
- 53 ifconfig-pool block-un istifadə edilməsi
- 57 status faylının istifadəsi
- 61 Management interface
- 63 Proxy-arp

66 Client-server Ethernet tipli şəbəkələr

- 67 Bridge olmayan şəbəkələrdə adi quraşdırma
- 71 Client-to-client trafikinin aktivləşdirilməsi
- 74 FreeBSD-də Bridge edilməsi
- 79 Windows Bridge edilməsi
- 82 IP olmayan və broadcast olan axının yoxlanılması
- 84 Kənar DHCP serverin istifadə edilməsi
- 88 Status faylının istifadə edilməsi
- 90 Management interfeys

94 PKI, Sertifikatlar və OpenSSL

- 95 Sertifikatın generasiya edilməsi
- 96 xCA: (1-ci hissə) PKI idarəedilməsi üçün GUI
- 99 xCA: (2-ci hissə) PKI idarəedilməsi üçün GUI
- 104 OpenSSL imkanları: x509, pkcs12, çıxışın yoxlanılması
- 106 Sertifikatların revoke (Vaxtını sıfırlamaq) edilməsi
- 108 CRL-lərin istifadə edilməsi
- 110 vaxtı-bitmiş/revoke edilmiş sertifikatların yoxlanılması
- 113 Aralıq CA-lar
- 117 Çoxlu CA-lar: --capath istifadə edərək stacking

121 FreeBSD OS-da OpenVPN üçün bilməli olduqlarımız və OpenVPN-də təcrübə misalları

- 122 ECMP ya da eyni mənsəbə bir neçə marşrut
- 124 ping: sendto: No buffer space available
- 126 FreeBSD OpenSC və PCSC-LITE yüklənməsi
- 127 FreeBSD OS üzərində bir neçə OpenVPN daemon-un eyni vaxtda işə salınması
- 135 OpenVPN şifrələnmiş kanalla AD qeydiyyatı
- 141 Ubuntu 14.04 OpenVPN-in Active Directory ilə inteqrasiyası
- 145 Ubuntu14.04-də OpenVPN üçün FreeRADIUS-la Active-Directory inteqrasiyası
- 153 Ubuntu 14.04 x64 OpenVPN və çoxlu LDAP qrupları

157 Scripting və Pluginlər

- 158 Client tərəfdə up/down scriptin istifadə edilməsi
- 162 Windows login greeter
- 164 client-connect/client-disconnect scriptlərin istifadə edilməsi
- 168 learn-address scriptin istifadə edilməsi
- 172 tls-verify scriptin istifadə edilməsi
- 175 auth-user-pass-verify scriptin istifadə edilməsi
- 178 Script ardıcılığı
- 180 Script təhlükəsizliyi və jurnallama
- 183 down-root pluginin istifadə edilməsi
- 186 PAM authentication pluginin istifadə edilməsi

190 OpenVPN quraşdırmalarının problemlərinin araşdırılması

- 191 Cipher uyğunsuzluğu
- 193 TUN və TAP alətlərinin uyğunsuzluğu
- 194 Kompresiya uyğunsuzluğu
- 196 Açar uyğunsuzluğu
- 197 MTU ve tun-mtu problemlərinin araşdırılma qaydaları
- 199 Şəbəkə qoşulmasının problemlərinin araşdırılması
- 201 Client-config-dir problemlərinin araşdırılması
- 203 OpenVPN jurnal fayllarının oxunulması qaydaları

208 OpenVPN: Routing troubleshooting

- 209 Çatışmayan qayıdış kodu
- 211 iroute istifadə ediləndə çatışmayan qayıdış route-ları
- 215 OpenVPN son nöqtələrindən başqa bütün clientləri funksional etmək
- 217 Source routing
- 221 Windows üzərində routing və yetki
- 223 client-to-client traffic routing problemlərinin həllinin araşdırılması
- 226 'MULTI: bad source' xəbərdarlıqlarının başa düşülməsi

229 Default gateway yönləndirməsində çıxan səhv

234 Performance tuning

235 ping istifadə edərək davamiyyətin optimallaşdırılması
237 iperf istifadə edərək davamiyyətin optimallaşdırılması
239 OpenSSL cipher-in sürəti
240 Kompresiya sınaqları
243 Axının boğulması
245 UDP bazalı qoşulmaların təkmilləşdirilməsi
248 TCP bazalı qoşulmaların təkmilləşdirilməsi
252 tcpdump istifadə edərək davamiyyətin analiz edilməsi

254 OS inteqrasiyası

255 Linux: NetworkManager-in istifadə edilməsi
260 Linux: pull-resolv-conf istifadə edilməsi
262 Mac OS: Tunnelblick istifadə edilməsi
266 Windows7: yetkilərin artırılması
268 Windows: CryptoAPI yığımının istifadəsi
271 Windows: DNS cache-in yenilənməsi
273 Windows: OpenVPN-in servis kimi işə düşməsi
277 Windows: PUBLIC ya da Private şəbəkə kartları
279 Windows: routing metodları

282 Genişlənmiş quraşdırma

283 Quraşdırma fayllarının quraşdırma fayllarına include(əlavə) edilməsi
284 Multiple remote və remote-random
287 ifconfig-pool-persist detalları
290 SOCKS proxy istifadə edərək qoşulma
293 HTTP proxy istifadə edərək qoşulma
298 Authentifikasiyası olan HTTP proxy ilə qoşulma
302 dyndns-in istifadə edilməsi
305 IP daha az olan quruluşlar(ifconfig-noexec)

309 OpenVPN 2.2-nin yeni imkanları

310 Sətir arası sertifikatlar
312 Qoşulma blokları
314 HTTPS server ilə portun yayımlanması
318 Routing bacarıqları: redirect-private, allow-pull-fqdn
321 PUBLIC IP ünvanların mənimsədilməsi
323 OCSP dəstəklənməsi
331 OpenVPN2.2-də yenilik: x509_user_name parametri

BÖLÜM 1

Point-to-Point şəbəkələri

Bu başlıqda biz aşağıdakıları açıqlayacağıq:

- Qısa işə salma imkanı
- OpenVPN gizli açarları
- Çoxlu gizli açarlar
- Plaintext tunnel
- Routing
- CLI-dan quraşdırma faylları
- IP ilə quraşdırma
- Site-to-Site quraşdırması
- 3 tərəfli routing

Giriş

Bu başlıqda biz OpenVPN-in quraşdırılmasına ətraflı baxacağıq. Point-to-point qoşulma o deməkdir ki, ancaq bir client eyni vaxtda qoşula bilər. Point-to-point qoşulma üsulunu kiçik olan sayt və istifadəçilər olan halda istifadə etmək düzgündür. Onu quraşdırmaq asandır ona görə ki, heç bir sertifikat (PKI) yaratmağa ehtiyac qalmır. Həmçinin routing quraşdırması daha asandır ona görə ki, client tərəfdə heç bir routeri quraşdırmağa gerek qalmır.

Qısa işə salma imkanı

Test üçün həm TUN həm də TAP alətlərindən istifadə edəcəyik. TUN alət həmişə IP trafik istifadə edilən VPN tunnelində olur. **TAP** alət isə tam **Ethernet** frame-də olan və bütün Protokolları OpenVPN ilə dəstəkləyir (AppleTalk və IPX misal üçün).

Deyək ki, iki ədəd maşınımız var:

Windows 7 - **openvpn-install-2.3.2-I003-x86_64.exe** yüklənilib. Server və Client rejimində işləyən Program. Ancaq burda Client kimi istifadə edəcəyik.

FreeBSD 9.2 - Portlar yenilənib və ordan **OpenVPN 2.3.2 amd64-portbld-freebsd9.2** yüklənilib.

Hər iki maşın şəbəkə ilə bir-birlərini Router və ya Switch vasitəsilə görürlər (Ya Internet yada LAN şəbəkə üzərindən). Misal üçün deyək ki, bizim halda maşınların aşağıdakı IP-ləri var.

Windows7 - **10.50.12.31**

FreeBSD9.2 - **10.50.12.32**

Windows7 isə aşağıdakı sintaksis ilə FreeBSD maşına qoşulacaq. **openvpn.exe** faylı isə **C:\Program Files\OpenVPN\bin** ünvanında yerləşir.

```
openvpn.exe --ifconfig 10.200.0.2 10.200.0.1 --dev tun --remote  
openvpnsrver.example.com
```

Hər ikisinin DNS adı var. Əgər yoxdursa, Windows maşından FreeBSD maşına tunel açmaq üçün **c:\windows\system32\drivers\etc\hosts** faylına həmin maşın üçün ad əlavə etməlisiniz. Çünki **Windows7 OpenVPN** client ilə həmin serverə ad ilə müraciət edəcək.

10.50.12.32 openvpnsrver.example.com

```
cd /usr/ports/security/openvpn # Port ünvanına daxil oluruq.  
make config # Lazımi modulları seçirik.
```

```
openvpn-2.3.2  
+-----+  
: [x] EASYRSA  Install security/easy-rsa RSA helper package  
: [x] PKCS11   Use security/pkcs11-helper  
: [x] PH_PASS  Interactive passwords may be read from a file  
:-----+  
:          SSL protocol support  
: (*) OPENSLL SSL/TLS support via OpenSSL  
: ( ) POLARSSL SSL/TLS support via PolarSSL  
+-----+  
      < OK >      <Cancel>
```

```
make install # Yükləyirik
```

FreeBSD kernel-i tap alətləri dəstəkləməsi üçün **/sys/amd64/conf/GENERIC** faylında "**Pseudo devices**" bölümündə "**device tap**" əlavə eləyib kerneli yenidən compile etmək lazımdır ki, TAP alətləri openvpn istifadə edə bilsin. OpenVPN-in versiyası 2.3.32-dir.

Həmçinin gələcək üçün OpenVPN-in OpenSSL sertifikatlarını daha rahat management etmək üçün ssl-admin portunuda yükləmək lazımdır.

```
root@siteA:~ # cd /usr/ports/security/ssl-admin/ # Port ünvanına daxil oluruq
root@siteA:/usr/ports/security/ssl-admin # make install # Yükləyirik
```

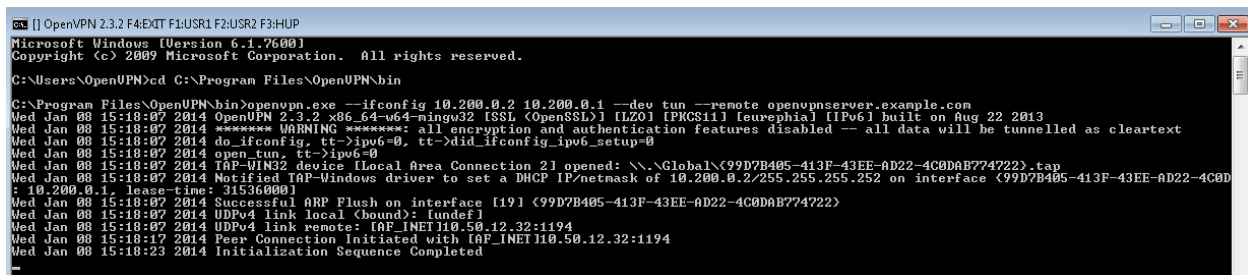
Necə edək:

1. Biz server (Yəni FreeBSD maşında olan OpenVPN serveri) **listening** rejimdə TUN tipli şəbəkə ilə işə salırıq:

```
root@openvpnserver:~ # openvpn --ifconfig 10.200.0.1 10.200.0.2 --dev tun
```

2. Windows7 yeni client tərəfdə isə OpenVPN-i client kimi işə salırıq:
cd C:\Program Files\OpenVPN\bin # Binar fayl yerləşən ünvana daxil oluruq
openvpn.exe --ifconfig 10.200.0.2 10.200.0.1 --dev tun --remote openvpnserver.example.com

Əgər hər şey normaldırsa, **Window7**-də aşağıdakı screenshot çıxmalıdır.



```

C:\Users\OpenVPN>cd C:\Program Files\OpenVPN\bin
C:\Program Files\OpenVPN\bin>openvpn.exe --ifconfig 10.200.0.2 10.200.0.1 --dev tun --remote openvpnserver.example.com
Wed Jan 08 15:18:07 2014 OpenVPN 2.3.2 x86_64-w64-mingw32 [SSL (OpenSSL)] [LZO] [PRCS] [lurephia] [IPo6] built on Aug 22 2013
Wed Jan 08 15:18:07 2014 ***** WARNING *****: all encryption and authentication features disabled -- all data will be tunneled as cleartext
Wed Jan 08 15:18:07 2014 do_ifconfig: tt->ip6=0, tt->did_ifconfig_ip6_setup=0
Wed Jan 08 15:18:07 2014 open_tun: tt->ip6=0
Wed Jan 08 15:18:07 2014 TAP-WIN32 device [Local Area Connection 2] opened: \\.\Global\{99D7B405-413F-43EE-AD22-4C0DAB774722}.tap
Wed Jan 08 15:18:07 2014 Notified TAP-Windows driver to set a DHCP IP/netmask of 10.200.0.2/255.255.255.252 on interface {99D7B405-413F-43EE-AD22-4C0DAB774722} [lease-time: 31536000]
Wed Jan 08 15:18:07 2014 Successful ARP Flush on interface [19] {99D7B405-413F-43EE-AD22-4C0DAB774722}
Wed Jan 08 15:18:07 2014 UDPv4 link local (bound): [undef]
Wed Jan 08 15:18:07 2014 UDPv4 link remote: [AF_INET]10.50.12.32:1194
Wed Jan 08 15:18:17 2014 Peer Connection Initiated with [AF_INET]10.50.12.32:1194
Wed Jan 08 15:18:23 2014 Initialization Sequence Completed

```

Artıq siz **Window7** maşından **FreeBSD** maşına **10.200.0.1**-e ping ata bilərsiniz.

3. Sonra **F4**-u sıxaraq hər iki maşında tuneli dayandırın ki, **TAP** alət istifadə edərək yeni tunel yaradaq.

4. FreeBSD OpenVPN-i listen modda TAP şəbəkə tipində işə salaq:

```
root@openvpnserver:~ # openvpn --ifconfig 10.200.0.1 255.255.255.0 --dev tap
```

5. **Windows7** Client tərəfdə isə **OpenVPN**-i **TAP** üçün prosesini işə salaq:

```
cd C:\Program Files\OpenVPN\bin # Binar faylın ünvanına daxil olaq  
openvpn.exe --ifconfig 10.200.0.2 255.255.255.0 --dev tap --remote openvpnserver.example.com
```



```

C:\Program Files\OpenVPN\bin>openvpn.exe --ifconfig 10.200.0.2 255.255.255.0 --dev tap --remote openvpnserver.example.com
Wed Jan 08 15:36:33 2014 OpenVPN 2.3.2 x86_64-w64-mingw32 [SSL (OpenSSL)] [LZO] [PRCS] [lurephia] [IPo6] built on Aug 22 2013
Wed Jan 08 15:36:33 2014 ***** WARNING *****: all encryption and authentication features disabled -- all data will be tunneled as cleartext
Wed Jan 08 15:36:33 2014 do_ifconfig: tt->ip6=0, tt->did_ifconfig_ip6_setup=0
Wed Jan 08 15:36:33 2014 open_tun: tt->ip6=0
Wed Jan 08 15:36:33 2014 TAP-WIN32 device [Local Area Connection 2] opened: \\.\Global\{99D7B405-413F-43EE-AD22-4C0DAB774722}.tap
Wed Jan 08 15:36:33 2014 Notified TAP-Windows driver to set a DHCP IP/netmask of 10.200.0.2/255.255.255.0 on interface {99D7B405-413F-43EE-AD22-4C0DAB774722} [DHCP-server: 10.200.0.0, lease-time: 31536000]
Wed Jan 08 15:36:33 2014 Successful ARP Flush on interface [19] {99D7B405-413F-43EE-AD22-4C0DAB774722}
Wed Jan 08 15:36:33 2014 UDPv4 link local (bound): [undef]
Wed Jan 08 15:36:33 2014 UDPv4 link remote: [AF_INET]10.50.12.32:1194
Wed Jan 08 15:36:43 2014 Peer Connection Initiated with [AF_INET]10.50.12.32:1194
Wed Jan 08 15:36:49 2014 Initialization Sequence Completed

```

Qoşulma yarandıqdan sonra isə **10.200.0.1** FreeBSD maşına ping ata bilərsiniz.

Bu necə işləyir

Server 1194-cu UDP portuna qulaq asır hansı ki, susmaya görə OpenVPN gələn qoşulmaları onda qəbul edir. Client isə serverə həmin porta qoşulur. Handshake(əl sıxışması) olduqdan sonra isə, server ilk istifadə edilməyən TUN alətinə 10.200.0.1 IP ünvanı mənimsədir və gözləyir ki, uzaqda olan ünvan(Peer-address) 10.200.0.2(Windows7 client) IP ünvanı alacaq.

Müştəri isə əksinə edir: Seans başlanğıcından sonra, ilk **TUN** yada **TAP** aləti özündə **10.200.0.2 IP** ünvanını quraşdırır. O gözləyir ki, uzaq ünvan(Peer address) **10.200.0.1 IP** ünvanında olsun. Məhz bundan sonra **VPN** qoşulması baş verir.

TAP tipli qoşulmada isə server ilk mövcud olan **TAP** aləti **10.200.0.1** və **255.255.255.0** mask ilə quraşdırır. Eynilə də client öz **TAP** alətini **10.200.0.2** və mask **255.255.255.0** ilə quraşdırır.

Qeyd: Gördüyünüz ******* WARNING *****: all encryption and authentication features disabled -- all data will be tunneled as cleartext** xəbərdarlıq isə o deməkdir ki, VPN üzərindən keçən data şifrələnmiş olmayacaq.

Daha da ətraflı

TCP protocol istifadə edək

Öncəki misalda biz UDP protocol istifadə edərək test elədik. Bu misalda isə yeganə fərq odur ki, TCP protocol-undan istifadə edəcəyik. Aşağıdakıları biz server tərəfdə edirik(burda **--remote** istifadə edilmir)

```
root@openvpnserver:~ # openvpn --ifconfig 10.200.0.1 10.200.0.2 --dev tun --proto tcp-server
```

Eynilə client tərəfdə:

```
openvpn.exe --ifconfig 10.200.0.2 10.200.0.1 --dev tun --proto tcp-client --remote openvpnserver.example.com
```

```
C:\Program Files\OpenVPN\bin>openvpn.exe --ifconfig 10.200.0.2 10.200.0.1 --dev tun --proto tcp-client --remote openvpnserver.example.com
Wed Jan 08 16:26:46 2014 OpenVPN 2.3.2 x86_64-w64-mingw32 [SSL [OpenSSL] [LZO] [PKCS11] [Auth] built on Aug 22 2013
Wed Jan 08 16:26:46 2014 ***** WARNING *****: all encryption and authentication features disabled -- all data will be tunneled as cleartext
Wed Jan 08 16:26:46 2014 do_ifconfig, tt->ip0=0, tt->did_ifconfig_ip0_setup=0
Wed Jan 08 16:26:46 2014 open tun, tt->ip0=0
Wed Jan 08 16:26:46 2014 TAP-WIN32 device [Local Area Connection 2] opened: \\.\Global\{99D7B405-413F-43EE-AD22-4C0DAB774722}.tap
Wed Jan 08 16:26:46 2014 Notified TAP-Windows driver to set a DHCP IP/netmask of 10.200.0.2/255.255.255.0 on interface {99D7B405-413F-43EE-AD22-4C0DAB774722} [DHCP-serv
r 10.200.0.1, lease-time: 31536000]
Wed Jan 08 16:26:46 2014 Successful ARP Flush on interface [191 {99D7B405-413F-43EE-AD22-4C0DAB774722}]
Wed Jan 08 16:26:46 2014 Attempting to establish TCP connection with [AF_INET]10.50.12.32:1194
Wed Jan 08 16:26:46 2014 TCP connection established with [AF_INET]10.50.12.32:1194
Wed Jan 08 16:26:46 2014 TCPv4_CLIENT link local: [undef]
Wed Jan 08 16:26:46 2014 TCPv4_CLIENT link remote: [AF_INET]10.50.12.32:1194
Wed Jan 08 16:26:56 2014 Peer Connection Initiated with [AF_INET]10.50.12.32:1194
Wed Jan 08 16:27:03 2014 Initialization Sequence Completed
```

non-IP trafikin tunel üzərindən yönləndirilməsi

Bu artıq mümkündür ki, IP olmayan trafiki tunel üzərindən ötürə biləsiniz. Misal üçün əgər hər iki tərəfdə AppleTalk düzgün quraşdırılıbsa, biz müraciəti remote host-a **aecho** əmri ilə yollaya bilərik.

```
aecho openvpnserver
```

```
22 bytes from 65280.1: aep_seq=0. time=26. ms
```

```
22 bytes from 65280.1: aep_seq=1. time=26. ms
```

```
22 bytes from 65280.1: aep_seq=2. time=27. ms
```

```
tcpdump -nnel -i tap0 əmri ilə görə bilərsiniz ki, bu trafik AppleTalk-dır.
```


OpenVPN gizli açarlar

Burda OpenVPN gizli açarları istifadə edəcəyik ki, tuneli təhlükəsiz edək. Bu öncəki misala oxşayır amma, artıq server və client arasında pre-shared key istifadə ediləcək hansı ki, gələn və gedən datanı şifrələyəcək.

Hazırlaşaq

OpenVPN-i iki computer-də yükləyək. Əmin olaq ki, hər iki computer ya LAN yada internet üzərindən bir-birlərini görürlər. Serverimiz FreeBSD 9.2 maşınında və clientimiz isə Windows7 maşınındadır. Hər iki maşında OpenVPN 2.3.3 yüklənmişdir.

Necə edək ...

1. İlk olaraq server(listener) tərəfdə secret açar generasiya edək:
root@openvpnserver:~/keys # **openvpn --genkey --secret secret.key**
2. Bu açarı təhlükəsiz yolla client maşına ötürün.
3. Sonra isə FreeBSD maşında OpenVPN Server-i listen moda keçirək:
root@openvpnserver:~/keys # **openvpn --ifconfig 10.200.0.1 10.200.0.2 --dev tun --secret secret.key**
4. Sonra, biz client tərəfdə OpenVPN prosesini işə salırıq(Sadəcə **secret.key** faylını **C:\Program Files\OpenVPN\bin** ünvanına nüsxələməyi unutmayın):
cd C:\Program Files\OpenVPN\bin # Binar faylların ünvanına daxil oluruq
openvpn.exe --ifconfig 10.200.0.2 10.200.0.1 --dev tun --secret secret.key --remote openvpnserver.example.com

Qoşulma uğurlu olarsa aşağıdakı şəkil çap ediləcək və ping ilə serveri görəcəksiniz.

```
C:\Program Files\OpenVPN\bin>openvpn.exe --ifconfig 10.200.0.2 10.200.0.1 --dev tun --secret secret.key --remote openvpnserver.example.com
Thu Jan 09 08:09:11 2014 OpenVPN 2.3.2 x86_64-w64-mingw32 [SSL (OpenSSL)] [LZO] [PKCS11] [cryptapi] [IPv6] built on Aug 22 2013
Thu Jan 09 08:09:11 2014 do_ifconfig, tt->ip6=0, tt->did_ifconfig_ip6_setup=0
Thu Jan 09 08:09:11 2014 open_tun, tt->ip6=0
Thu Jan 09 08:09:11 2014 TAP-WIN32 driver: Local Area Connection 21 opened: \\.\Global\{99D7B405-413F-43EE-AD22-4C0DAB774722}.tap
Thu Jan 09 08:09:11 2014 Notified TAP-Windows driver to set a DHCP IP/netmask of 10.200.0.2/255.255.255.252 on interface {99D7B405-413F-43EE-AD22-4C0DAB774722} [DHCP-server: 10.200.0.1, lease-time: 31536000]
Thu Jan 09 08:09:11 2014 Successful ARP Flush on interface [19] {99D7B405-413F-43EE-AD22-4C0DAB774722}
Thu Jan 09 08:09:11 2014 UDPv4 link local (bound): [undef]
Thu Jan 09 08:09:11 2014 UDPv4 link remote: [AF_INET110.50.12.32:1194]
Thu Jan 09 08:09:21 2014 Peer Connection Initiated with [AF_INET110.50.12.32:1194]
Thu Jan 09 08:09:27 2014 Initialization Sequence Completed
```

Bu necə işləyir

Bu misal eyni olaraq birinci göstərdiyimiz kimi işləyir: Server gələn qoşulmalar üçün UDP 1194-cu portunda qulaq asır. Client serverin həmin portuna qoşulur. Uğurlu handshake olduğundan sonra, server ilk boş olan TUN alətini 10.200.0.1 IP ünvanı ilə quraşdırır və gözləyir ki, client(Peer-address) 10.200.0.2 IP-si alacaq. Client-də əksinə olur.

Daha da ətraflı

Susmaya görə OpenVPN, point-to-point qoşulmalar üçün 2 simmetrik açar istifadə edir.

- Cipher key - paketlər mübadiləsində onların tərkibini şifrələyir.

- HMAC key - paketləri imzalayır. Paketlər HMAC-la imzalanmadan çatarsa, onlar gözlənilmədən dayandırılır. Bu DoS-dan müdafiənin ilk yoludur.

Eyni açar dəsti hər iki sonda istifadə edilir və hər iki açar **--secret** parametri sayəsində fayldan göstərilmişdir.

OpenVPN secret açarın formatı aşağıdakı kimidir:

```
root@openvpnserver:~/keys # cat secret.key
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
6e17a04e49ee58d075390542ba4f6a67
a3c5b293329b8b9779218537f232c2f3
7b3882892e01188c49cba4926fb35ad2
cae100e6e2bc4fd4e0dfaa67d9768c9b
debdd399d9ce8a6e05de099f606a92d0
f75b1de731754e66391adcecbd147cdf
7c01376065730a71d3ff92fe47e5c9b6
dd844981cdfc6849c717e69882599211
b43610ae4bf332dfef8d9322a6d0cfef
57928abcf707cafe2d0d8604589b6657
892cd375a197829ad0a58bf54d335e1c
1b18f51ee2066d9f98ab99fbaf47e2aa
541d65da8f3b3b6b327cec8828445e7b
3f03aed178efcd26d13b75392efd9cd5
7f113859478c1aa743b55eb1f1827ecb
d1040f09467d2031f8791c32fc5164a9
-----END OpenVPN Static key V1-----
```

Təsadüfi baytlardan, OpenVPN və HMAC cipher əldə edilib.

Qeyd: Bu açarlar hər bir sessiya üçün eynidir.

Həmçinin baxaq

Növbəti dəfə də, müxtəlif secret açarlar olacaq və biz secret açarlarla bağlı detallı danışacağıq.

Çoxlu gizli açarlar

Öncəki misalda olduğu kimi, point-to-point qoşulmalarında OpenVPN 2 simmetrik açar istifadə edir. Point-to-point qoşulmalarında birgə amma asimetrik açarlar istifadə etmək mümkündür. OpenVPN bu halda 4 açar istifadə edəcək.

- **Client terefde cipher key**
- **Client terefde HMAC key**
- **Server terefde cipher key**
- **Server terefde HMAC key**

Eyni key materiaları hər iki point-to-point qoşulmalarında yayımlanıb ancaq, bu açarlar yaradılmışdır ki, hər iki tərəfdə müxtəlif açarlarla data şifrelənsin və imzalsın.

İşə başlayaq

Bu halda biz öncə generasiya elədiyimiz **secret.key**-dən istifadə edəcəyik. Əmin olaq ki, hər iki maşın ya LAN yada internet üzərindən bir-birlərini görürlər. Serverimiz FreeBSD 9.2 maşınında və clientimiz isə Windows7 maşınındadır. Hər iki maşında OpenVPN 2.3.3 yüklənmişdir.

Necə edək:

1. Biz OpenVPN serveri **--secret** və daha çox log(jurnal) olan rejimdə işə salırıq(listen rejimdə):

```
root@openvpnserver:~/keys # openvpn --ifconfig 10.200.0.1 10.200.0.2 --dev tun --secret secret.key 0 --verb 7
```

2. Həmçinin client tərəfdə OpenVPN prosesini işə salırıq:

```
cd C:\Program Files\OpenVPN\bin # Binar fayılın ünvanına daxil oluruq  
openvpn.exe --ifconfig 10.200.0.2 10.200.0.1 --dev tun --secret secret.key 1 --remote openvpnserver --verb 7
```

Qoşulma debug mesajları ilə uğurla olacaq və sonra ping ilə test edə bilərsiniz.

3. Əgər biz server tərəfinə baxsaq(**crypt** sözüne görə axtarış edin), orda danışıqlada istifadə olunan açarlar var. Çıxış aşağıdakı formada görünəcək:

```
Thu Jan 9 11:56:36 2014 us=235650 Static Encrypt: Using 160 bit  
message hash 'SHA1' for HMAC authentication  
Thu Jan 9 11:56:36 2014 us=235674 Static Encrypt: HMAC KEY: debdd399  
d9ce8a6e 05de099f 606a92d0 f75b1de7  
Thu Jan 9 11:56:36 2014 us=235688 Static Encrypt: HMAC size=20  
block_size=20  
Thu Jan 9 11:56:36 2014 us=235775 Static Decrypt: Cipher 'BF-CBC'  
initialized with 128 bit key  
Thu Jan 9 11:56:36 2014 us=235797 Static Decrypt: CIPHER KEY: b43610ae  
4bf332df eb8d9322 a6d0cfef  
Thu Jan 9 11:56:36 2014 us=235811 Static Decrypt: CIPHER block_size=8  
iv_size=8  
Thu Jan 9 11:56:36 2014 us=235897 Static Decrypt: Using 160 bit  
message hash 'SHA1' for HMAC authentication  
Thu Jan 9 11:56:36 2014 us=235917 Static Decrypt: HMAC KEY: 541d65da  
8f3b3b6b 327cec88 28445e7b 3f03aed1
```

Client tərəfdə isə biz eyni açarları tapa bilərik amma, 'Encrypt' və 'Decrypt' açarlar rezerv edilmiş olacaq:

```
Thu Jan 09 13:21:15 2014 us=163940 Static Encrypt: Cipher 'BF-CBC'  
initialized with 128 bit key  
Thu Jan 09 13:21:15 2014 us=163940 Static Encrypt: Using 160 bit  
message hash 'SHA1' for HMAC authentication  
Thu Jan 09 13:21:15 2014 us=163940 Static Decrypt: Cipher 'BF-CBC'  
initialized with 128 bit key
```

Thu Jan 09 13:21:15 2014 us=163940 Static Decrypt: Using 160 bit message hash 'SHA1' for HMAC authentication

Əgər siz diqqət yetirsəniz görəcəksiniz ki, açarların hər biri həm server həm də client tərəfdə nüsxələnir.

Bu necə işləyir

OpenVPN bütün açarları static.key açıqdan alır o şərtlə ki, faylda kifayət qədər fərqlilik var (Bu dörd açarın etibarlı generasiya edilməsi üçündür). Generasiya edilmiş bütün açarların fərqli istifadəsi üçün aşağıdakı əmr kifayət edir:

```
openvpn --genkey --secret secret.key
```

OpenVPN static açar faylı **2048** bit həcmə malikdir. Cipher key-in hər biri 128 bitdir, HMAC key-in hər biri 160 bitdir və ümumilikdə 776 bit edir. Bu OpenVPN-ə asanlıqla imkan yaradır ki, hər bir **static.key** fayldan 4 ədəd təsadüfi açar generasiya edə bilsin hətta, cipher seçilsədə belə bu açarın genişlənmiş inisializasiyasını tələb edir.

Daha da ətraflı

Eyni secret key faylları aşağıdakı parametrlə həm server həm də client tərəfdə istifadə edilərsə, işləyəcək: **tls-auth ta.key**

Həmçinin baxaq

- 2-ci başlıqda PUBLIC və Private açarların istifadəsi açıqlanır hansı ki, tls-auth key generasiya ediləcək və ətraflı danışılacaq.

Plaintext tunnel

İlk misalımızda biz tunel yaratmışdıq hansı ki, içi ilə gedən data trafiki şifrələnmiş deyildi. Tamamilə açıq şəkildə data ötürən tunelin yaradılması üçün HMAC autentikasiyanı söndürmək lazımdır. Bu pis qoşulmanın debug edilməsi vaxtında istifadə edilə bilər və tunel üzərindən gedən bütün trafik aşan şəkildə monitoring edilə bilər. Bu başlıqda biz onu necə etməyimizə baxacağıq. Tunelin bu tipi həmçinin, tab gətirmə imkanlarının yoxlanılmasında da və ən az resurs istifadə edən tunel kimidə istifadə edilə bilər.

İşə başlayaq

Əmin olaq ki, hər iki maşın ya LAN ya da internet üzərindən bir-birlərini görürlər. Serverimiz FreeBSD 9.2 maşınında və clientimiz də həmçinin FreeBSD 9.2 maşınındadır. Hər iki maşında OpenVPN 2.3.3 yüklənmişdir. Heç bir şifrələmə istifadə etmədiyimizə görə **secret.key**-ə də ehtiyac olmayacaq.

Necə edək

1. Serverdə OpenVPN Prosesini işə salaq (Listen rejimə keçirək):

```
root@openvpnserver:~/keys # openvpn --ifconfig 10.200.0.1 10.200.0.2 --dev tun --auth none
```

- Client tərəfdə isə öncə `/etc/hosts` faylına `10.50.12.32 openvpnserver.example.com` sətirini əlavə edin və sonra aşağıdakı sətiri əlavə edin ki, OpenVPN işə düşsün:

```
root@openvpn-client:/usr/ports/security/openvpn # openvpn --ifconfig 10.200.0.2 10.200.0.1 --dev tun --auth none --remote openvpnserver.example.com
```
- Qoşulma aşağıdakı warning ilə yerinə yetiriləcək:

```
Thu Jan 9 18:38:23 2014 ***** WARNING *****: all encryption and authentication features disabled -- all data will be tunneled as cleartext
```

Bu necə işləyir

Bu tip qoşulmada heç bir şifrələmə olmur. Bütün tuneldən keçən traffic OpenVPN paketinə encapsulyasiya ediləcək və `as-is` kimi ötürüləcək.

Biraz ətraflı

Trafikə açıq baxmaq üçün isə biz `tcpdump` istifadə edə bilərik:

- Yazdığımız kimi qoşulma hazırdır.
- Client maşının şəbəkə kartında `tcpdump`-i işə salın (fiziki şəbəkə kartında logic yox).

```
root@openvpn-client:~ # tcpdump -w -e -i em0 -s 0 host openvpnserver.example.com | strings
```
- İndi `nc` (yada `netcat`)-yə oxşar utilit ilə isə tunelin içindən müəyyən bir tekst ötürün. Öncə server tərəfdə `nc`-ni işə salın.

```
root@openvpnserver:~ # nc -l 31000
```
- Client tərəfdə isə `nc`-ni client rejimdə işə salın və `hello` ilə `goodbye` sözlərini daxil edin.

```
root@openvpn-client:~ # nc 10.200.0.1 31000
hello
goodbye
```
- Server-də yerinə `tcpdump` işə saldığınız maşında aşağıdakı nəticəni əldə edəcəksiniz.

```
root@openvpn-client:~ # tcpdump -l -w - -i em0 -s 0 host openvpnserver.example.com | strings
tcpdump: listening on em0, link-type EN10MB (Ethernet), capture size 65535 bytes
hello
goodbye
```

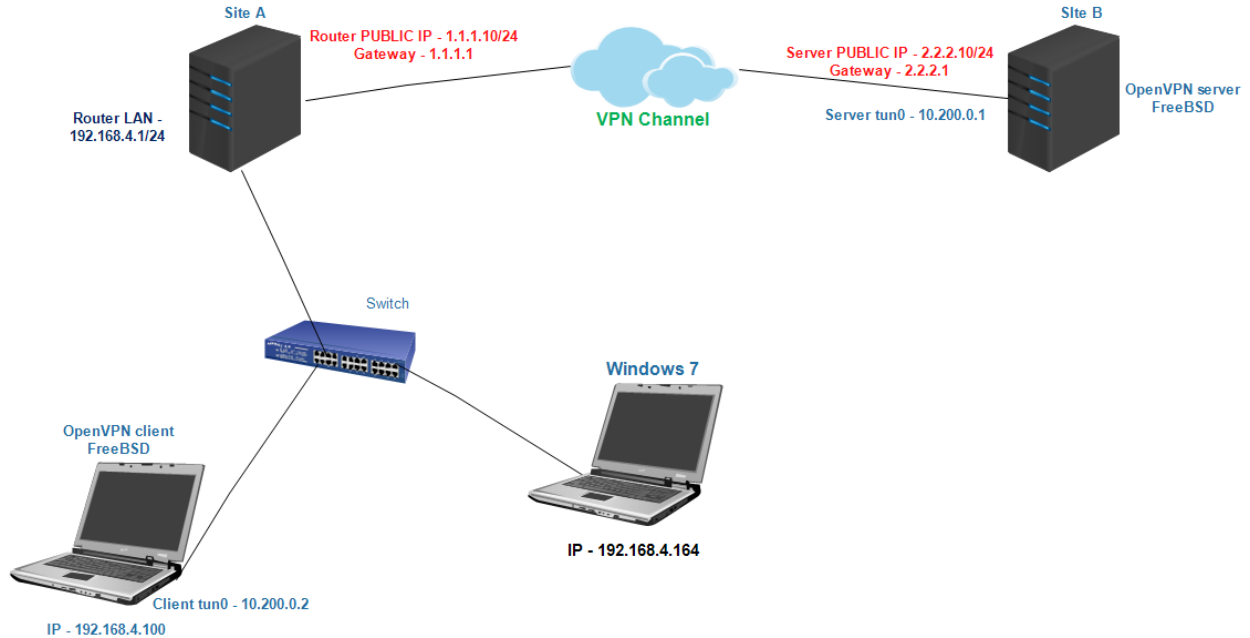
Routing

Point-to-Point qoşulma o halda yaxşıdır ki, əgər siz iki subnet-i arasında static olaraq şifrələnmiş tunel yaratmaq istəyirsiniz. Əgər son compyüter

sayı azdırsa(4-dən çox olmayaraq), o halda bu Client/Server qoşulmasından çox asandır(2-ci başlıqda yalnız IP şəbəkələri üçün bunu ətraflı danışacağıq).

İşə başlayaq

Bu başlıqda biz aşağıdakı şəbəkə quruluşunu istifadə edəcəyik:



Serverimiz FreeBSD 9.2 maşınında və clientimiz də həmçinin FreeBSD9.2 maşınındadır. Hər iki maşında OpenVPN 2.3.3 yüklənmişdir. Biz OpenVPN-in haqqında danışdığımız Secret açarını **secret.key** istifadə edəcəyik(Nəzərdə tutulur ki, **openvpnserver.example.com** adı ya DNS-də ya da ki, hosts faylında təyin edilib).

Bunu necə edək

1. İlk olaraq qoşulmanı quraq amma, əmin olun ki, OpenVPN özü tərəfindən daemonlaşdırılıb(Daemon-u bu əmrə dayandırıb test-i yenidən edə bilərsiniz `kill -9 `ps -ax | grep openvpn | grep -v grep | awk '{ print $1 }'``):

```
root@openvpnserver:~ # openvpn --ifconfig 10.200.0.1 10.200.0.2 --dev tun --secret /root/keys/secret.key --daemon --log /tmp/openvpnserver.log
```

2. Sonra işə client tərəfdə OpenVPN prosesini işə salın(Serverdə olan secret.key faylını /root/keys qovluq yaradıb ora nüsxələyin):
- ```
root@openvpn-client:~ # openvpn --ifconfig 10.200.0.2 10.200.0.1 --dev tun --secret /root/keys/secret.key --remote openvpnserver.example.com --daemon --log /tmp/openvpnclient.log
```

Qoşulma uğurlu oldu:

```
root@openvpnserver:~/keys # tail -1 /tmp/openvpnserver.log
Thu Jan 9 23:34:53 2014 Initialization Sequence Completed
```

İndi isə routing əlavə edək:

1. Server tərəfdə static route əlavə edirik:

```
root@openvpnserver:~ # route add -net 192.168.4.0/24 10.200.0.2
```

2. Client tərəfdə isə biz iki iş görməliyik:

- Əmin olun ki, sizdə IP trafikinin yönləndirilməsi aktivdir. FreeBSD maşında biz bunu reboot eləmədən aşağıdakı əmrlə edəcəyik:

```
root@openvpn-client:~ # sysctl -w net.inet.ip.forwarding=1
```

- Əmin olun ki, Windows client maşınında LAN trafikinin routing-i elə OpenVPN serverə qayıdır:

```
C:> route add 10.200.0.0 mask 255.255.255.0 192.168.4.100
```

Burda 192.168.4.100 IP OpenVPN clientin LAN ipsidir.

3. Artıq biz serverdən clientin LAN-nı ping edə bilərik. İlk olaraq biz OpenVPN client-in LAN IP-sini ping edək.

```
root@openvpnserver:~ # ping -c2 192.168.4.100
```

```
PING 192.168.4.5 (192.168.4.5): 56 data bytes
```

```
64 bytes from 192.168.4.5: icmp_seq=0 ttl=63 time=0.334 ms
```

```
64 bytes from 192.168.4.5: icmp_seq=1 ttl=63 time=1.051 ms
```

4. Və OpenVPN client LAN-da olan növbəti client IP-sini ping edək:

```
root@openvpnserver:~ # ping -c 2 192.168.4.164
```

```
PING 192.168.4.164 (192.168.4.164): 56 data bytes
```

```
64 bytes from 192.168.4.164: icmp_seq=0 ttl=127 time=8.144 ms
```

```
64 bytes from 192.168.4.164: icmp_seq=1 ttl=127 time=0.478 ms
```

### **Bu necə işləyir**

Bizim şəbəkə quruluşumuzda, çatmaq istədiyimiz LAN, OpenVPN client-in arxasındadır və ona görə də biz serverə route yazmalıyıq.

```
root@openvpnserver:~ # route add -net 192.168.4.0/24 10.200.0.2
```

Client tərəfdə isə biz iki iş görməliyik:

- Əmin olaq ki, routing rejimi aktivdir. Əgər siz route rejiminin reboot-dan sonra işləməsini istəyirsinizsə, onda **/etc/sysctl.conf** faylına aşağıdakı sətiri əlavə edin:

```
net.inet.ip.forwarding=1
```

- Biz həmçinin əmin olmalıyıq ki, client LAN-ından geri OpenVPN serverə routing mövcuddur. Biz bunu LAN gateway-e route yazmaqla yada client LAN-da olan hər bir maşına static route yazmaqla edə bilərik. Bizim halda Windows client-ə route əlavə elədik hansı ki, OpenVPN FreeBSD client ilə eyni şəbəkədədir.

```
C:\Users\OpenVPN>route add 10.200.0.0 mask 255.255.255.0 192.168.4.100
```

**192.168.4.100** IP ünvanı isə FreeBSD OpenVPN client-in IP ünvanıdır.

## Dahada ətraflı

### Routing problemləri

Internet-də əlavə edilən routing ilə bağlı çoxlu problemlər ilə rastlaşacaqsınız, ancaq onların əksərini OpenVPN özü həll etmişdir. Həll edilməyənlər isə OS-a məxsus qalmış buglardır. Ancaq 8-ci başlıqda bu tip və ümumiyyətlə problemləri aradan qaldırılması haqqında daha detallı yazılmışdır.

### İşimizi avtomatlaşdıraraq

Həmçinin mümkündür ki, tunel özü qalxan kimi elə route əlavə eləsin. Biz bunu `--route` əmri ilə parametri ilə edə bilərik:

```
root@openvpnserver:~ # openvpn --ifconfig 10.200.0.1 10.200.0.2 --dev tun --secret secret.key --daemon --log /var/log/openvpnserver-1.5.log --route 192.168.4.0 255.255.255.0
```

Unutmayın ki, client LAN-da geri serverə route hələ də əllə əlavə edilməlidir.

### Həmçinin baxaq

- Bu başlıqda olan son başlıq, 3-yollu routing hansı ki, əksər hallarda 3 remote qoşulmalarda açıqlanır.
- 8-ci başlıq, Routing problemlərində OpenVPN-in troubleshoot edilməsi.

## CLI-dan quraşdırma faylları və IP ilə quraşdırma

Bu kitabın əksər açıqlamaları quraşdırma faylları istifadə edilmədən edilir. Ancaq real heyatda CLI-dan uzun əmrlərin daxil edilməsi əvəzinə quraşdırma fayllardan istifadə etmək daha da məntiqləyicidir. Bilməyiniz önəmlidir ki, OpenVPN-də cli-dan daxil edilən quraşdırmalar və ya quraşdırma fayllarında olan quraşdırmalar tam identikdir. Fərq yalnız ondan ibarətdir ki, CLI-dan daxil edilən əmrlər iki ədəd tire `--` ilə olur (quraşdırma fayllarında tire `--` olmur). Məhz buna görə də quraşdırma faylını istifadə etmək dahada asandır.

### İşə başlayaq

OpenVPN 2 və daha yuxarı versiyanı iki kompyüterdə yükləyin. Əmin olun ki, həmin kompyüterlər eyni şəbəkədə yada routing vasitəsilə bir-birlərini görürlər. Server maşında FreeBSD9.2 OpenVPN 2.3 və client maşında da Windows7 yüklənmişdir. Bu başlıqda da öncə generasiya elədiyimiz secret.key istifadə edəcəyik.



## Bunu necə etməliyik..

1. Öncəki misalımıza uyğun olan quraşdırmalara əsasən Windows7 client maşında quraşdırma faylını yaradaq və aşağıdakı tərkibi əlavə edək:

```
dev tun
port 1194
ifconfig 10.200.0.2 10.200.0.1
secret secret.key
remote openvpnserver.example.com
verb 3
```

Faylı **C:\Program Files\OpenVPN\bin\example1-6-client.conf** adı ilə yadda saxlayıb çıxın.

2. FreeBSD OpenVPN Serveri Listen rejimdə işə salaq, ancaq qeyri standart port ilə:

```
root@openvpnserver:~ # openvpn --ifconfig 10.200.0.1 10.200.0.2 --dev tun --secret /root/keys/secret.key --port 11000
```

3. Sadəcə öncə şəkildə isitifadə edilən topologiyanı istifadə edirsinizsə, 'c:\windows\system32\drivers\etc\hosts' faylına 'OpenVPN\_Server\_IP openvpnserver.example.com' setiri əlavə etməyi unutmayın.

Windows7 Client tərəfdə OpenVPN-i işə salanda əlavə parametr ilə quraşdırma faylının ünvanını göstərək.:

```
cd C:\Program Files\OpenVPN\bin # Binar faylın ünvanına daxil oluruq
openvpn.exe --config example1-6-client.conf --port 11000
```

Qoşulma uğurlu olduqda aşağıdakı şəkil çap ediləcək.

```
C:\Program Files\OpenVPN\bin>openvpn.exe --config example1-6-client.conf --port 11000
Sat Jan 11 18:15:34 2014 OpenVPN 2.3.2 x86_64-w64-mingw32 [SSL (OpenSSL)] [LZO] [PKCS11] [eurephia] [IPv6] built on Aug 22 2013
Sat Jan 11 18:15:34 2014 Static Encrypt: Cipher 'BF-CBC' initialized with 128 bit key
Sat Jan 11 18:15:34 2014 Static Encrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Sat Jan 11 18:15:34 2014 Static Decrypt: Cipher 'BF-CBC' initialized with 128 bit key
Sat Jan 11 18:15:34 2014 Static Decrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Sat Jan 11 18:15:34 2014 Socket Buffers: R=[8192->8192] S=[8192->8192]
Sat Jan 11 18:15:34 2014 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0
Sat Jan 11 18:15:34 2014 open_tun, tt->ipv6=0
Sat Jan 11 18:15:34 2014 TAP-WING32 device [Local Area Connection 2] opened: \\.\Global\{99D7B405-413F-43EE-AD22-4C0DAB774722}.tap
Sat Jan 11 18:15:34 2014 TAP-Windows Driver Version 9.9
Sat Jan 11 18:15:34 2014 Notified TAP-Windows driver to set a DHCP IP/netmask of 10.200.0.2/255.255.255.252 on interface {99D7B405-413F-43EE-AD22-4C0DAB774722} [DHCP-serve: 10.200.0.1, lease-time: 21536000]
Sat Jan 11 18:15:34 2014 Successful ARP Flush on interface [19] {99D7B405-413F-43EE-AD22-4C0DAB774722}
Sat Jan 11 18:15:34 2014 UDPv4 link local (bound): [undef]
Sat Jan 11 18:15:34 2014 UDPv4 link remote: [AF_INET]10.198.0.10:11000
Sat Jan 11 18:15:44 2014 Peer Connection Initiated with [AF_INET]10.198.0.10:11000
Sat Jan 11 18:15:51 2014 TEST ROUTES: 0/0 succeeded len=0 ret=1 a=0 u/d=up
Sat Jan 11 18:15:51 2014 Initialization Sequence Completed
```

## Bu necə işləyir

CLI və quraşdırma faylı sətiri soldan sağa oxuyur və mənimsədir. Bu o deməkdir ki, quraşdırma faylından öncə təyin edilən əksər opsiyalar bu fayl sayəsində etiraz edilə bilər. Uyğun olaraq, aşağıdakı direktivdən sonra təyin edilən opsiyalar bu fayla yazmanın qarşısını alacaq.

```
--config example1-6-client.conf
```

Uyğun olaraq növbəti opsiya quraşdırma faylında olan '**port 1194**' sətirinin oxunmasına etiraz edəcək:

```
--port 11000
```

Yalnız bəzi opsiyalar bir neçə dəfə təyin edilə bilər ancaq, bu halda ilk olanı işləyəcək. Bu halda həmçinin **--config** direktivinin önündə opsiyanı da təyin eləmək olar.

### Birazda ətraflı

Başqa bir misal çəmək ki, CLI-dan əlavə edilən parametrlərin önəmliliyini açıqlayaq:

```
C:\>"\Program Files\OpenVPN\bin\openvpn.exe" --verb 0 --config "\Program Files\OpenVPN\bin\example1-6-client.conf" --port 11000
```

Bu halda verbose rejim 3-də işləyəcək ona görə ki, sonda olan **example1-6-client.conf**-fayldan işə düşmüşdür.

Aşağıdakı misalda isə verbose rejim 3-də işləməyəcək ona görə ki, sonda CLI-dan oxunmuşdur. Yeni ki, sonda olan işə düşəcək.

```
C:\>"\Program Files\OpenVPN\bin\openvpn.exe" --config "\Program Files\OpenVPN\bin\example1-6-client.conf" --port 11000 --verb 0
```

### OpenVPN 2.1 spesifikasiyaları

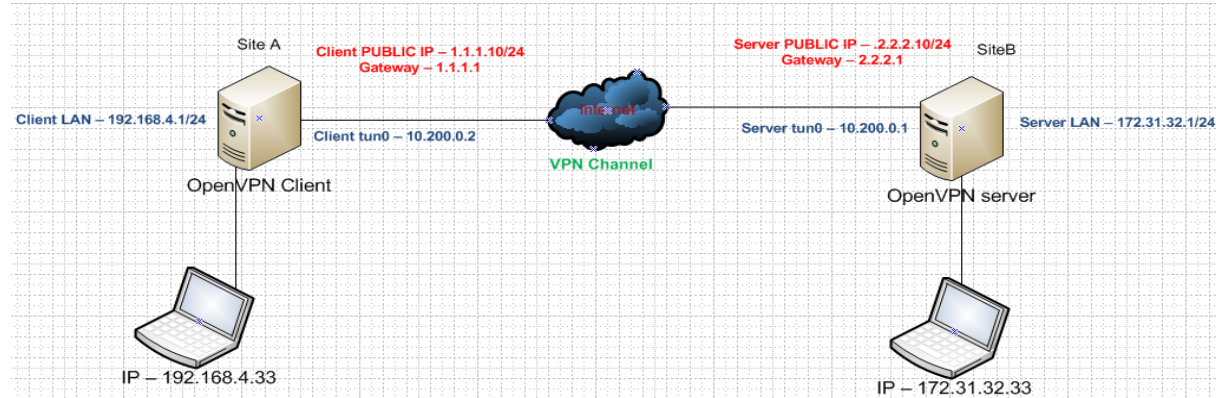
OpenVPN2.1-in bəzi spesifikasiyaları öz prinsiplərindən fərqlənir misal üçün, <connection> blokları və daxili sertifikatlar. 12-ci başlıqda biz bu barədə daha da ətraflı danışacağıq.

### Site-to-Site quraşdırması

Bu başlıqda biz site-to-site quraşdıracağıq və OpenVPN-in tərkibində olan əksər daxili təhlükəsizlik funksiyalarından istifadə edəcəyik. Bu "one-stop-shop" misalı olaraq point-to-point qoşulmanı göstərir.

### İşə başlayaq

Biz aşağıdakı şəkilə uyğun olan şəbəkəni qururuq



OpenVPN-i iki maşında yükləyin. Bizim testimizdə iki ədəd FreeBSD9.2 maşın istifadə ediləcək. Onlardan 1-i client və digəri isə server olacaq. Aralarında isə öncə istifadə elədiyimiz **secret.key** istifadə ediləcək.

Əmin olun ki, hər iki maşın routing rejimində işləyir (IP forwarding yeni **/etc/rc.conf** faylında **gateway\_enable="YES"** mövcuddur). Və hər iki maşının **/etc/hosts** faylında aşağıdakı sətirlər mövcud olmalıdır.

```
1.1.1.10 openvpnclient.example.com
2.2.2.10 openvpnserver.example.com
```

## Necə edək

1. Server quraşdırma faylını yaradın(**example-7-server.conf** adında saxlayıb yadda saxlayın):

```
dev tun
proto udp
local openvpnserver.example.com
lport 1194
remote openvpnclient.example.com
rport 1194
secret /root/keys/secret.key 0
ifconfig 10.200.0.1 10.200.0.2
route 192.168.4.0 255.255.255.0
user nobody
group nobody
persist-tun
persist-key
keepalive 10 60
ping-timer-rem
verb 3
daemon
log-append /tmp/openvpn.log
```

2. Client tərəfdə isə quraşdırma faylına aşağıdakıları əlavə edin(**example1-7-client.conf** adı ilə yadda saxlayın):

```
dev tun
proto udp
local openvpnclient.example.com
lport 1194
remote openvpnserver.example.com
rport 1194
secret /root/keys/secret.key 1
ifconfig 10.200.0.2 10.200.0.1
route 172.31.32.0 255.255.255.0
user nobody
group nobody
persist-tun
persist-key
keepalive 10 60
ping-timer-rem
verb 3
daemon
log-append /tmp/openvpn.log
```

3. Hər iki tərəfdə tuneli işə salın:

```
root@openvpnserver:~/keys # openvpn --config example-7-server.conf
```

Və:

```
root@openvpn-client:~/keys # openvpn --config example1-7-client.conf
```

Artıq bizim site-to-site tunelimiz hazırdır.

4. Hər iki maşının log fayllarını yoxlayın və dəqiqləşdirin ki, qoşulma uğurla başa çatmışdır. VPN qalxan kimi, hər iki maşının LAN tərəfində olan son nöqtələr bir birlərini VPN üzərindən görə bilərlər.

5. Misal üçün biz serverdən, client-in LAN tərəfində olan maşını ping edə bilərik.

```
root@openvpnserver:~/keys # ping -c2 192.168.4.164
PING 192.168.4.164 (192.168.4.164): 56 data bytes
64 bytes from 192.168.4.164: icmp_seq=0 ttl=127 time=1.182 ms
64 bytes from 192.168.4.164: icmp_seq=1 ttl=127 time=2.573 ms

--- 192.168.4.164 ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.182/1.877/2.573/0.696 ms
root@openvpnserver:~/keys #
```

### Bu necə işləyir

Client və serverin quraşdırmaları çox oxşardılar:

- Server yalnız 1 şəbəkə kartı və 1 UDP port-da qulaq asır
- Server qoşulmanı yalnız 1 IP ünvan və 1 portdan qəbul edir
- Client aşağıdakı uyğun parametrlərə malikdir.

Burda quraşdırmanın çox opsiyaları var:

```
user nobody
group nobody
persist-tun
persist-key
keepalive 10 60
ping-timer-rem
```

Bunlar qoşulmanı daha etibarlı və təhlükəsiz eləməyə şərait yaradır. Dəqiq desək:

- OpenVPN prosesi qoşulma uğurlu olduqdan sonra **nobody** istifadəçi və qrup adından işə düşür. Ona görə ki, əgər kiməsə OpenVPN prosesini ələ keçirsə o hələki **nobody** istifadəçi olacaq **root** yox. Yalnız bezi Linux distroları ola bilər ki, **nogroup** istifadə edilir.
- **persist-tun** və **persist-key** opsiyaları, əsas şəbəkənin qırılması olduğu halda VPN-in avtomatik işə düşməsinə əmin olmaq üçün istifadə edilirlər. Bu opsiyalar ancaq istifadəçi və qrup nobody istifadə edilərkən lazımdır.
- **keepalive** və **ping-timer-rem** opsiyaları istifadə edilir ki, OpenVPN vaxtaşırı ping mesajları yollayaraq hər iki tunel sonluğunu yoxlayır ki, hər iki tərəfin işlək vəziyyətdə olmasını görsün.

### Daha ətraflı

Bu point-to-point qoşulma həmçinin firewall tərəfindən filter edilib yerinə yetirilə bilər. Hər iki sonluq arasındakı datanı açmaq və deşifrə eləmək çox çətinidir. OpenVPN client/server rejimdə işə düşdükdə (2-ci başlıqda Multi-client TUN-style şəbəkələrdən danışacağıq) trafik ilk TLS handshake hesabına OpenVPN trafiki kimi təyin edilir.

### Həmçinin baxaq

- 8-ci başlıqda OpenVPN problemlərinin həll edilməsi: Routing problemləri hansı ki, əksər routing problemlərini açıqlayır.

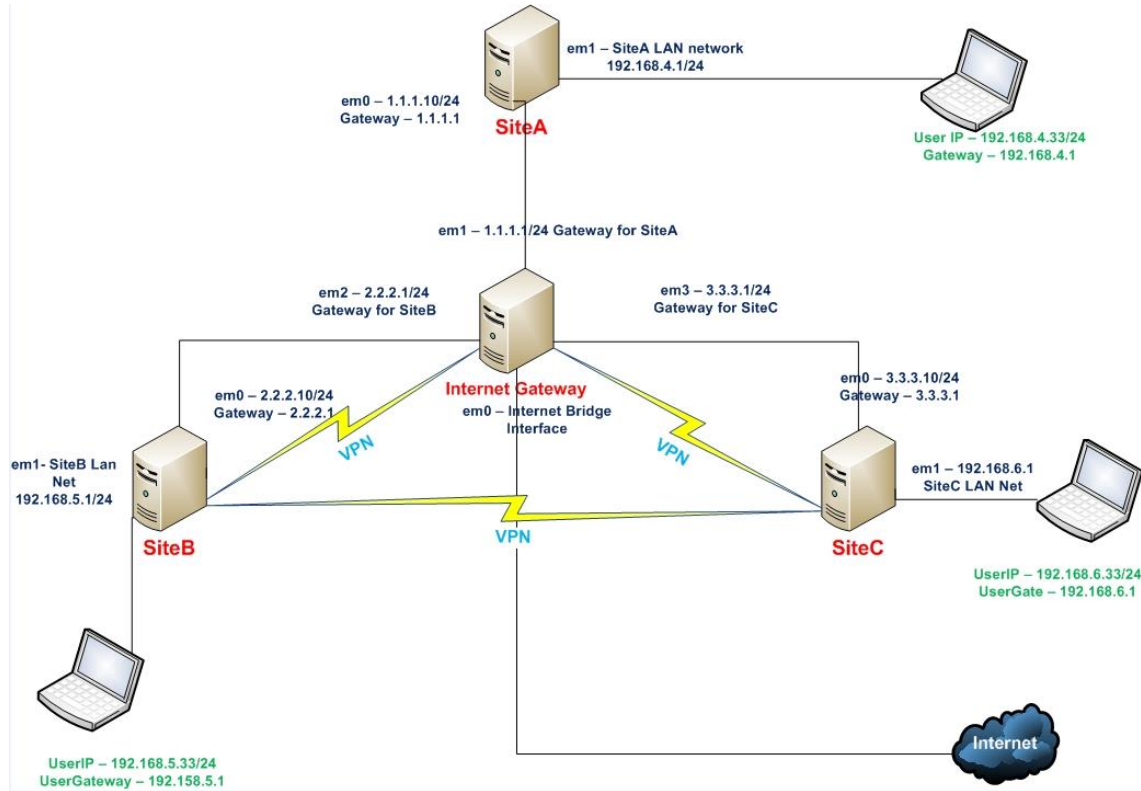
### 3 tərəfli routing

Kiçik(4-dən kiçik olanlar) sayılı qoşulmalar üçün point-to-point ən yaxşıdır. Bu dəfə biz 3 OpenVPN tuneli 3 ayrı tərəf arasında edəcəyik və son nöqtələridə daxil olmaqla. 3 ayrı tunel yaratmaqla biz dayanıqlı routing edəcəyik ona görə ki, son nöqtələrdən biri çöksə trafik o birisi üzərindən keçəcək.

Yeni bir maşın hər bir halda 3-cüyə ikinci üzərindən keçə biləcək.

### İşə başlayaq

Biz aşağıdakı şəbəkə quruluşunu yerinə yetirəcəyik:



OpenVPN-i 3 maşında yükləyin. Hər 3 maşın bizim halda FreeBSD9.2 x64-dür. Göstərilən şəbəkəyə uyğun olaraq şəbəkəmiz var. Əmin olun ki, hər 3 maşında `/etc/rc.conf` faylında `gateway_enable="YES"` və ya `/etc/sysctl.conf` faylında `net.inet.ip.forwarding=1` mövcuddur. Hər 3 maşında `/root/keys` qovluq yaradın.

**Qeyd:** Unutmayın ki, test müddətində PUBLIC IP ünvanlarının istifadəsi mütləqdir əks halda OpenVPN routing-i düzgün başa düşmür.

### Necə edək:

1. İlk olaraq **SiteA** serverdə `/root/keys` ünvanına daxil olaraq static açarları generasiya edəcəyik:  

```
root@siteA:~/keys # openvpn --genkey --secret AtoB.key
root@siteA:~/keys # openvpn --genkey --secret AtoC.key
```

```
root@siteA:~/keys # openvpn --genkey --secret BtoC.key
```

Bu açarların üçünüdə **scp** ilə digər maşınlarla köçürün(**scp /root/\*.key 3.3.3.10:/root/keys/**).

Hər 3 maşının **/etc/hosts** faylında aşağıdakı sətirlər mövcuddur ki, adı IP-yə çecirmək mümkün olsun.

```
127.0.0.1 localhost
1.1.1.10 siteA
2.2.2.10 siteB
3.3.3.10 siteC
```

Hər 3 maşında kernel aşağıdakı opsiyalarla compile edilməlidir ki, MultiPath routing dəstəklənsin.

```
options RADIX_MPATH # Mutipath Routing işləməsi üçün
options ROUTETABLES=15 # Müxtəlif proqram təminatları üçün
route table-a izin verək
```

2. Server maşınlar(Listen edən) üçün **example1-8-serverBtoA.conf** adında quraşdırma faylı yaradın və aşağıdakı sətirləri həmin fayla əlavə edin. Bu və aşağıda göstəriləcək quraşdırma faylları hər üç maşının **/root/keys** qovluğuna nüsxələyin ki, işiniz asanlaşsın çünki, hər üç maşında lazımi quraşdırma faylı həm **listener** həm də **client** kimi istifadə ediləcək(hansı quraşdırma faylının hansı serverdə işə düşməsinə siz CLI-da göstərilən maşının Hostname-inə görə təyin edə bilərsiniz):

```
dev tun
proto udp
port 1194

secret /root/keys/AtoB.key 0
ifconfig 10.200.0.1 10.200.0.2

route 192.168.4.0 255.255.255.0 vpn_gateway 5
route 192.168.6.0 255.255.255.0 vpn_gateway 10
route-delay

keepalive 10 60
verb 3
```

Sonra **example1-8-serverCtoA.conf** adlı fayl yaradın və içine aşağıdakıları əlavə edin:

```
dev tun
proto udp
port 1195

secret /root/keys/AtoC.key 0
ifconfig 10.200.0.5 10.200.0.6

route 192.168.4.0 255.255.255.0 vpn_gateway 5
route 192.168.5.0 255.255.255.0 vpn_gateway 10
route-delay

keepalive 10 60
verb 3
```

Sonda isə **example1-8-serverBtoC.conf** adında quraşdırma faylına aşağıdakı sətirləri əlavə edin:

```
dev tun
proto udp
port 1196

secret /root/keys/BtoC.key 0
ifconfig 10.200.0.9 10.200.0.10

route 192.168.4.0 255.255.255.0 vpn_gateway 10
route 192.168.6.0 255.255.255.0 vpn_gateway 5
route-delay

keepalive 10 60
verb 3
```

İndi isə **Client(Connector)** quraşdırma fayllarını yaradaq. İlk olaraq **example1-8-clientAtoB.conf** faylını aşağıdakı tərkib ilə:

```
dev tun
proto udp
remote siteB
port 1194

secret /root/keys/AtoB.key 1
ifconfig 10.200.0.2 10.200.0.1

route 192.168.5.0 255.255.255.0 vpn_gateway 5
route 192.168.6.0 255.255.255.0 vpn_gateway 10
route-delay

keepalive 10 60
verb 3
```

Həmçinin **example1-8-clientAtoC.conf** faylını aşağıdakı tərkib ilə yaradırıq:

```
dev tun
proto udp
remote siteC
port 1195

secret /root/keys/AtoC.key 1
ifconfig 10.200.0.6 10.200.0.5

route 192.168.5.0 255.255.255.0 vpn_gateway 10
route 192.168.6.0 255.255.255.0 vpn_gateway 5
route-delay

verb 3
```

Və sonda **example1-8-clientCtoB.conf** faylını aşağıdakı tərkib ilə yaradırıq:

```
dev tun
proto udp
remote siteB
port 1196

secret /root/keys/BtoC.key 1
ifconfig 10.200.0.10 10.200.0.9

route 192.168.4.0 255.255.255.0 vpn_gateway 10
route 192.168.5.0 255.255.255.0 vpn_gateway 5
route-delay

keepalive 10 60
verb 3
```

Yaratdığımız quraşdırma faylları hər Gateway serverdən hər 3 serverə nüsxələyək.

```
root@vpngate:~/keys # scp example1-8-* 1.1.1.10:/root/keys
root@vpngate:~/keys # scp example1-8-* 2.2.2.10:/root/keys/
root@vpngate:~/keys # scp example1-8-* 3.3.3.10:/root/keys/
```

İlk olaraq Listener tunelləri işə salacayıq.

```
root@siteB:~/keys # openvpn --config example1-8-serverBtoA.conf
root@siteB:~/keys # openvpn --config example1-8-serverBtoC.conf
root@siteC:~/keys # openvpn --config example1-8-serverCtoA.conf
```

Sonra işə Connector tunelləri işə salacayıq

```
root@siteA:~/keys # openvpn --config example1-8-clientAtoB.conf
root@siteA:~/keys # openvpn --config example1-8-clientAtoC.conf
root@siteC:~/keys # openvpn --config example1-8-clientCtoB.conf
```

Və bununlada belə bizim 3 tərəfli site-to-site şəbəkəmiz uğurla başa çatmış oldu.

### **Bu necə işləyir**

Prinsipcə iki tunel kifayət edərdi ki, 3 remote obyektı qoşmaq mümkün olsun ancaq, o halda heç bir dayanıqlıq olmayacaq.

3-cü tunellə və quraşdırma opsiyaları ilə:

```
route 192.168.5.0 255.255.255.0 vpn_gateway 5
route 192.168.6.0 255.255.255.0 vpn_gateway 10
route-delay
keepalive 10 60
```

Həmişə hər iki şəbəkə üçün 2 route olacaq.



Misal üçün, siteA-dan siteB-yə 2 ədəd route var (LAN 192.168.5.0/24 üçün). Aşağıdakı əmr ilə biz bu cədvəli görə bilərik.

```
root@siteA:~ # netstat -rn | grep 192.168.5.0/24
192.168.5.0/24 10.200.0.1 UGS 0 12 tun0 =>
192.168.5.0/24 10.200.0.5 UGS 0 0 tun1
```

Route:

- Birbaşa tunel ilə siteB-yə; bu route-un kiçik metriki olacaq.
- Birbaşa olmayan tunelin içi ilə; öncə siteC və sonra da siteB; Bu route-un böyük metriki var və ilk route çökməyəndək bu route işə düşməyəcək.

Bu quruluşun üstünlüyü ondan ibarətdir ki, əgər bir tunel çöxsə, onda 60 saniyədən sonra qoşulmalar və uyğun olan route-lar drop edilib restart ediləcək. Backup route avtomatik işə düşəcək və yenidən hər 3 məşın bir-birini görəcək.

Birbaşa tunel geri qaytarıldıqda isə, birbaşa route-da həmçinin geri qaytarılacaq və şəbəkə trafikini avtomatik olaraq remote site-lara daha yaxşı yolu seçərək keçəcəklər.

## Daha da geniş

### Genişlik

Bu başlıqda biz 3 məşını bir-birilə əlaqələndirəcəyik. Bu nəticə 6 müxtəlif quraşdırma fayllarına gətirib çıxarır hansı ki, buda point-to-point qoşulmasını limitləyir. Ümumiyyətlə,  $N$  mövcud sayda olan və tam rezervləməni nəzərə almaqla olan qoşulmalar üçün sizin  $N*(N-1)$  sayında quraşdırma fayllarınız olacaq. Bu 4 site-dədək idarə edilə bilər ancaq bundan sonrakı başlıqlarda server/client qoşulmaları dahada asan yolla açıqlanacaq.

### Routing protokolları

Şəbəkə dayanıqlığı üçün yaxşı metod Routing protokollardan RIPv2 və ya OSPF-in istifadəsidir. Bu protokolları istifadə etməklə siz düşən route-ları daha tez təyin edib və ünvanını elə dəyişdirə bilərsiniz ki, daha az düşmə vaxtı meydana gələr.

### Həmçinin baxaq

- 8-ci başlıq, OpenVPN-in troubleshoot edilməsi: Routing problemləri hansı ki, əksər routing problemləri açıqlanır.

## BÖLÜM 2

### Client-server yalnız IP şəbəkələrində

Bu başlıqda biz aşağıdakıları açıqlayacağıq:

- Public və private açarların quraşdırılması
- Kiçik quraşdırılma
- Server tərəfdən route edilmə
- **client-config-dir** faylların istifadəsi
- **Routing**: Hər iki tərəfin subnetlərinin route edilməsi
- Default gateway-in yönləndirilməsi
- **ifconfig-pool** block-un istifadə edilməsi
- status faylın istifadəsi
- Management interface
- Proxy-arp

#### **Giriş**

Bu başlıqda olan misallar OpenVPN-in istifadə edilən əksər modellərini açıqlayacaq: IP routing trafikə uyğun olan çoxlu clientlər və bir server.

Biz həmçinin əsas routing quraşdırmalarına baxacağıq, əsas olaraq hər iki client və server tərəfdə management interfeysə.

Bu başlığın son misalında göstərəcəyik ki, kütləvi praktikada şəbəkə bridge-ləri necə istifadə edilir.

Əksər quraşdırmalarda TUN alətlərindən istifadə edildiyinə görə bu başlığımızda istifadə edilən misallar digər başlıqlarda təkrar istifadə edilə bilər. Misal üçün server-tərəf routing-də əksər hallarda **basic-udp-server.conf**, **basicudp-client.conf**, **basic-tcp-server.conf** və **basic-tcp-client.conf** faylları və Windows client quraşdırmalarında isə **basic-udp-client.ovpn**, **basic-tcp-client.ovpn** quraşdırma faylları istifadə ediləcək.

## Public və Private açarların quraşdırılması

Client/Server VPN yaratmadan öncə biz PUBLIC açar(PKI) infrastrukturunu yaratmalıyıq. PKI özünə Certificate Authority, Private açarları və certificates(Public açarları) həm client və həm də server üçün daxil edir. Həmçinin biz Diffie-Hellman parametrli açar generasiya eləməliyə ki, gizliliyi ideal yönlədirə bilək.

PKI yaratmaq üçün biz OpenVPN tərəfindən yaradılmış **easy-rsa** scriptlərindən istifadə eləməliyə.

### İşə başlayaq

PKI tam inandığımız bir kompyuterdə olmalıdır. O həmçinin elə OpenVPN serverin özündə də ola bilər ancaq, təhlükəsizlik tələblərinə görə o tamam ayrı bir server üzərində olmalıdır. Əsas tələblərindən biri odur ki, **CA(Certificate Authority)** açarı tamam başqa yerdə saxlayaq. Misal üçün external storage hansı ki, yalnız tələb ediləndə istifadə edilsin. Digər əsas tələb odur ki, CA private açarı tamam şəbəkədən ayrılmış bir kompyuterdə saxlamaq lazımdır.

Bu resepti FreeBSD9.2 x64 maşında istifadə etmişəm. Ancaq Linux və Windows maşında da eyni əmrlərlə istifadə edə bilərsiniz. Ancaq **easy-rsa** scriptlərin işlənməsi üçün BASH shell tələb edilir ona görə də maşınıınıza öncədən bash-ı yükləməyi(**pkg install -y bash**) unutmayın.(easy-rsa portlarda **/usr/ports/security/easy-rsa** ünvanında yerləşir)

### Necə edək

1. PKI üçün qovluqları yaradın və **easy-rsa** scriptlərini həmin qovluğa nüsxələyin:

```
root@siteA:~ # mkdir -m 700 -p /usr/local/etc/openvpn/itvpn/keys
root@siteA:~ # cd /usr/local/etc/openvpn/itvpn
root@siteA:~ # cp -R /usr/local/share/easy-rsa/* .
```

2. Bu əmrlərin root istifadəçi adından işə salınmasına gerek yoxdur.

3. Sonra biz **vars** faylını yaradaq. Faylı yaradın və aşağıdakıları içinə əlavə edin.

```
export EASY_RSA=/usr/local/etc/openvpn/itvpn
export OPENSSL="openssl"
export KEY_CONFIG=`$EASY_RSA/whichopensslcnf $EASY_RSA`
export KEY_DIR="$EASY_RSA/keys"
export PKCS11_MODULE_PATH="dummy"
export PKCS11_PIN="dummy"
export KEY_SIZE=2048
export CA_EXPIRE=3285
export KEY_EXPIRE=1000
export KEY_COUNTRY="AZ"
export KEY_PROVINCE=
export KEY_CITY=
export KEY_ORG="Itvpn"
```

```
export KEY_EMAIL="openvpn-ca@itvpn.example.com"
```

**Qeyd:** **PKCS11\_MODULE\_PATH** və **PKCS11\_PIN** verilənləri o halda tələb edilir ki, siz SmartCard istifadə etmirsiniz. Susmaya görə olan **KEY\_SIZE** 2048 bitdir və bu uzunluq növbəti 2-3 il üçün təhlükəsizdir. Həmçinin geniş uzunluqlu **4096**-bitlik açar mümkündür ancaq şifrələnmə böyük olduğuna görə effektivlik aşağı düşəcək. Biz 4096 bitlik CA private açar yaradacağıq ona görə ki, burada effektivlik heç nəyə gerek deyil. Həmçinin dəyişənlər var ki, sizin təşkilata (**KEY\_ORG**, **KEY\_EMAIL**) xasdır. Bu quraşdırmaların açılmasını birazdan daha detallı şəkildə danışacağıq.

4. 4096 bitlik modul istifadə edərək **vars** faylı yerinə yetirək, CA private açar və sertifikat generasiya edək. CA sertifikat üçün çətin şifrə seçin. Bundan sonra hər dəfə script işə düşdükdən sonra həmin şifrəni daxil edin:

```
root@siteA:~ # cd /usr/local/etc/openvpn/itvpn/
root@siteA: /usr/local/etc/openvpn/itvpn # bash # BASH-a keçirik.
[root@siteA /usr/local/etc/openvpn/itvpn/# source ./vars
[root@siteA /usr/local/etc/openvpn/itvpn/# ./clean-all
[root@siteA /usr/local/etc/openvpn/itvpn/# KEY_SIZE=4096 ./build-ca --pass
```

```
[root@siteA /usr/local/etc/openvpn/cookbook]# KEY_SIZE=4096 ./build-ca --pass
Generating a 4096 bit RSA private key
.....++
.....++
writing new private key to 'ca.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [NL]:
State or Province Name (full name) []:
Locality Name (eg, city) []:
Organization Name (eg, company) [Cookbook]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [Cookbook CA]:
Name []:
Email Address [openvpn-ca@atl.az]:
```

5. Sonra server sertifikatını generasiya edəcəyik. Script daxil edilməsini istəyəndə şifrə daxil edib enter-i sıxın. Script **ca.key** şifrəsini istəyəndə isə CA sertifikatı üçün şifrəni daxil edin. Sonda isə script soruşacaq **[y,n]** siz **y** edin.

```
[root@siteA /usr/local/etc/openvpn/itvpn/# ./build-key-server openvpnserver
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'openvpnserver.key'

You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
```

For some fields there will be a default value,  
If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [AZ]:  
State or Province Name (full name) []:  
Locality Name (eg, city) []:  
Organization Name (eg, company) [Itvpn]:  
Organizational Unit Name (eg, section) []:  
Common Name (eg, your name or your server's hostname) [openvpnserver]:  
Name []:  
Email Address [openvpn-ca@domain.lan]:

Please enter the following 'extra' attributes  
to be sent with your certificate request

A challenge password []:  
An optional company name []:  
Using configuration from /usr/local/etc/openvpn/itvpn/openssl-0.9.8.cnf  
Enter pass phrase for /usr/local/etc/openvpn/itvpn/keys/ca.key:  
Check that the request matches the signature  
Signature ok  
The Subject's Distinguished Name is as follows  
countryName :PRINTABLE:'AZ'  
organizationName :PRINTABLE:'Itvpn'  
commonName :PRINTABLE:'openvpnserver'  
emailAddress :IA5STRING:'openvpn-ca@domain.lan'  
Certificate is to be certified until Oct 9 19:15:14 2016 GMT (1000  
days)  
Sign the certificate? [y/n]:**y**

1 out of 1 certificate requests certified, commit? [y/n]**y**  
Write out database with 1 new entries  
Data Base Updated

6. Client üçün ilk sertifikat **build-key** ilə yaradılır. Bu client sertifikatının yaradılması üçün çox sürətli metodikadır ancaq, bu halda clientin private key faylına şifrə təyin etmək olmur.  
[root@siteA /usr/local/etc/openvpn/itvpn]# **./build-key openvpnclient1**

```

Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'openvpnclient1.key'

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [NL]:
State or Province Name (full name) []:
Locality Name (eg, city) []:
Organization Name (eg, company) [Cookbook]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [openvpnclient1]:
Name []:
Email Address [openvpn-ca@atl.az]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /usr/local/etc/openvpn/cookbook/openssl-0.9.8.cnf
Enter pass phrase for /usr/local/etc/openvpn/cookbook/keys/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName :PRINTABLE:'NL'
organizationName :PRINTABLE:'Cookbook'
commonName :PRINTABLE:'openvpnclient1'
emailAddress :IASSTRING:'openvpn-ca@atl.az'
Certificate is to be certified until Oct 12 04:07:55 2016 GMT (1000 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated

```

7. İkinci client sertifikatı şifrə ilə generasiya edilmişdir. Çətin şifrə seçin(Ancaq CA sertifikat-da seçdiyiniz şifrədən fərqli olmalıdır!). Aydınlıq üçün çıxış qısa göstərilmişdir:

```

[root@siteA /usr/local/etc/openvpn/itvpn]# ./build-key-pass openvpnclient2
Using configuration from /usr/local/etc/openvpn/itvpn/openssl-0.9.8.cnf
Enter pass phrase for /usr/local/etc/openvpn/itvpn/keys/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName :PRINTABLE:'AZ'
organizationName :PRINTABLE:'Itvpn'
commonName :PRINTABLE:'openvpnclient2'
emailAddress :IA5STRING:'openvpn-ca@domain.lan'
Certificate is to be certified until Oct 10 05:08:03 2016 GMT (1000
days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated

```

8. Ardınca server üçün Diffie-Hellman parametrlı fayl qurun:

```
[root@siteA /usr/local/etc/openvpn/itvpn]# ./build-dh
```

9. Sonda isə **tls-auth** key faylı:

```
[root@siteA /usr/local/etc/openvpn/itvpn]# openvpn --genkey --secret keys/ta.key
```

Bütün bu gördüyümüz işlərdən sonra **/usr/local/etc/openvpn/itvpn/keys** qovluğunda aşağıdakı fayllar yaranacaq:

**ca.crt** - Əsas CA sertifikat, bu fayl həm client və həm də serverə lazımdır  
**dh2048.pem** - Diffie Hellman açarı, bu fayl yalnız serverə lazımdır

**Qeyd:** Əgər bu açar yaranmazsa, sadəcə **/usr/local/etc/openvpn/keys** ünvanında **./build-dh** əmrini daxil etməyiniz yetər ki, **dh2048.pem** açarı yaransın.

**openvpnserver.crt** - Serverin sertifikatı, yalnız server üçündür  
**openvpnserver.key** - Serverin açarı, yalnız server üçündür (gizli fayl)  
**openvpnclient1.crt** - Clientin sertifikatı, yalnız client üçündür  
**openvpnclient1.key** - Clientin açarı, yalnız client üçündür (gizli fayl)  
**ta.key** - TLS-açar, həm client və həm də serverə lazımdır

Uyğun olaraq serverdə **ca.crt**, **dh2048.pem**, **openvpnserver.crt**, **openvpnserver.key**, **ta.key** faylları və ilk client-də isə **ca.crt**, **dh2048.pem**, **openvpnclient1.crt**, **openvpnclient1.key**, **ta.key** faylları olmalıdır.

### Bu necə işləyir

**easy-rsa** scriptləri openssl CA əmrləri ilə işləyir və çox rahatdır. **openssl ca** əmrləri əsasən PKI-in **X509** sertifikatları ilə istifadəsində tələb edilir. **build-dh** scripti isə **openssl dh** əmri üçündür.

### Daha da ətraflı

**easy-rsa scriptlərin Windows-da istifadə edilməsi.**

**easy-rsa** açarların istifadə edilməsi üçün Windows-da cmd-dən daxil olub scriptləri işə salmaq lazımdır. Misal üçün:

```
[Win]C:> vars
[Win]C:> clean-all
[Win]C:> build-ca
```

### Müxtəlif dəyişənlər haqqında bəzi qeydlər

Aşağıdakı dəyişənlər **vars** faylında istifadə edilmişdir:

- **KEY\_SIZE=2048:** Bu bütün private açarlar üçün imzalanan uzunluqdur. Uzun açar daha da çətin şifrələmə deməkdir. Ancaq bu şifrələmə müddətini artırır.
- **CA\_EXPIRE=3650:** CA sertifikatın gündəmdə olması müddətini təyin edir və bu 10 il deməkdir. Orta səviyyəli təhlükəsizlik üçün bu müddət kifayət edir amma, yüksək səviyyəli təhlükəsizlik üçün isə bu azdır.
- **KEY\_EXPIRE=1000:** Bu client və server üçün olan sertifikatın gündəmdə olması müddətini təyin edir və demək olar ki, 3 il deməkdir.
- **KEY\_COUNTRY="AZ", KEY\_PROVINCE=, KEY\_CITY=, KEY\_ORG="Itvpn", KEY\_EMAIL=openvpn-ca@itvpn.example.com:** Bu dəyişənlərin hamısı certificate **Distinguished Name (DN)**-də istifadə edilir. Onların hamısı tələb edilmir ancaq, həm OpenVPN həm də OpenSSL-də **KEY\_COUNTRY**-nin olması önəmlidir ki, sertifikatın harda generasiya edilməsini təyin etmək olsun.

## Həmçinin baxın

Başlıq 4-də PKI, Certificates və OpenSSL-ə ətraflı baxın ki, easy-rsa scriptləri və openssl əmləri dərinədən başa düşəsiniz.

## Kiçik quraşdırma

Bu başlıqda sertifikatları istifadə edərək həm client həm də server qoşulmalarının necə edilməsini açıqlayacağıq.

## Tələbatlar

İki maşında OpenVPN yükləyin. Əmin olun ki, bu maşınlar şəbəkə üzərindən birlərini görürlər. Öncə haqqında danışdığımız client və serverin sertifikatlarını artıq quraşdıraraq. Bizim misalda hər iki maşın FreeBSD 9.2 x64 üzərində işləyir.

## Necə edək

1. Server quraşdırma faylını yaradaq:

```
proto udp
port 1194
dev tun
server 192.168.200.0 255.255.255.0

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnserver.crt
key /usr/local/etc/openvpn/openvpnserver.key
dh /usr/local/etc/openvpn/dh2048.pem
```

Bunu **example2-2-server.conf** faylında yadda saxlayın.

2. Açarları **keys** sertifikat qovluğundan OpenVPN-in **/usr/local/etc/openvpn/** qovluğuna nüsxələyin:

```
[root@siteA /usr/local/etc/openvpn]# cd /usr/local/etc/openvpn
[root@siteA /usr/local/etc/openvpn]# cp itvpn/keys/ca.crt .
[root@siteA /usr/local/etc/openvpn]# cp itvpn/keys/openvpnserver.crt
openvpnserver.crt
[root@siteA /usr/local/etc/openvpn]# cp itvpn/keys/openvpnserver.key
openvpnserver.key
[root@siteA /usr/local/etc/openvpn]# cp itvpn/keys/dh2048.pem
dh2048.pem
```

3. Qeyd edin ki, öncəki əmlərin istifadəsində **'root'** istifadəçi olmağa ehtiyac yoxdur.

4. Serveri işə salaq:

```
[root@siteA /usr/local/etc/openvpn]# openvpn --config example2-2-
server.conf
```

5. **siteB** hostname-li server əslində clientdir. Serverdən clientə aid olan sertifikat, key faylını və CA-nin sertifikatını client maşına copy edirik. Biz öncə **siteB** maşında openvpn public qovluq yaradırıq(yeni



`/usr/local/etc/openvpn)` və ora daxil olub client quraşdırma faylını yaradaq və konteninə aşağıdakı sətirləri əlavə edək:

```
client
proto udp
remote openvpnserver.example.com
port 1194
dev tun
nobind
```

```
ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnclient1.crt
key /usr/local/etc/openvpn/openvpnclient1.key
```

Faylı `example2-2-client.conf` adı ilə yadda saxlayın. Həmçinin `/etc/hosts` faylına server resolve etmək üçün `'1.1.1.10 openvpnserver.example.com'` sətirini əlavə etməyi unutmayın.

6. Öncə dediyim kimi, `ca.crt`, `openvpnclient1.crt` və `openvpnclient1.key` faylını `siteA`-dan `siteB`-yə `scp` ilə köçürək.

```
root@siteA:/usr/local/etc/openvpn/itvpn/keys # scp ca.crt
2.2.2.10:/usr/local/etc/openvpn/
root@siteA:/usr/local/etc/openvpn/itvpn/keys # scp openvpnclient1.crt
2.2.2.10:/usr/local/etc/openvpn/
root@siteA:/usr/local/etc/openvpn/itvpn/keys # scp openvpnclient1.key
2.2.2.10:/usr/local/etc/openvpn/
```

7. Və sonda client-i işə salaq:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config example2-2-
client.conf
Thu Jan 16 08:20:13 2014 /sbin/ifconfig tun0 192.168.200.10
192.168.200.9 mtu 1500 netmask 255.255.255.255 up
add net 192.168.200.1: gateway 192.168.200.9 fib 0
Thu Jan 16 08:20:13 2014 Initialization Sequence Completed
```

Qoşulma başa çatdıqdan sonra serverin UP olmasını ping ilə yoxlayaq:

```
root@siteB:~ # ping -c2 192.168.200.1
PING 192.168.200.1 (192.168.200.1): 56 data bytes
64 bytes from 192.168.200.1: icmp_seq=0 ttl=64 time=0.937 ms
64 bytes from 192.168.200.1: icmp_seq=1 ttl=64 time=1.613 ms
```

### **Bu necə işləyir**

Server işə düşən kimi o, mövcud olan ilk TUN alətini 192.168.200.1 IP ünvanı və yalançı 192.168.200.2 IP ünvanı ilə quraşdırır. Ardınca server 1194-çü port-da UDP ilə qulaq asmağa başlayır.

Client serverə həmin port ilə qoşulur. Uyğun olan TLS razılaşma yerinə yetirildikdən sonra, hər iki tərəfdə sertifikatlar istifadə edilir və client-ə 192.168.200.6 IP ünvanı mənimsədilir (yada kiçik şəbəkə desək daha yaxşı olar **192.168.200.4-192.168.200.7**). Həmçinin client verilən bu informasiyanı istifadə edərək ilk boş olan TUN alətindən istifadə edəcək və bundan sonra da VPN işə düşəcək.

## Daha da ətraflı

### 'net30' ünvanlandırma

Qoşulma başa çatdıqdan sonra siz **TUN** interfeysinə aşağıdakı əmr ilə baxa bilərsiniz:

```
root@siteB:~ # ifconfig tun0 | grep inet
```

Aşağıdakı sətirə baxın:

```
inet 192.168.200.10 --> 192.168.200.9 netmask 0xffffffff
```

192.168.200.5 IP ünvanı burda sadəcə yeri istifadə etmək üçündür ki, prinsipimiz işləsin və ona ping çatmayacaq. OpenVPN 2.1-dən başlayaraq artıq müştərilərə "**linear**" ünvanlar təyin etmək olur hansı ki, sizə şərait yaradır ki, eyni IP aralığında çoxlu müştəri istifadə edə bilərsiniz. Bunu növbəti misalda açıqlayacağıq.

"/30" aralığında olan IP-nin ilki müştərinin özü və ikinci IP ünvan isə yalançı son nöqtənin IP ünvanıdır. Hər bir "/30" subnet-i 4 IP ünvan ilə işə düşür və VPN clientin ünvanı isə həmin ünvan və üstəgəl 2 ilə işə düşür:

- **192.168.200.[0-3]**, VPN IP-si **192.168.200.1**-dir. Bu block adi halda həmçinin elə OpenVPN-in özündə olur.
- **192.168.200.[4-7]**, client IP-si **192.168.200.6**-dır. Bu block adi halda ilk qoşulmaq istəyən müştəri üçün nəzərdə tutulur.
- **192.168.200.[8-11]**, **[12-15]**, **[16-19]** və daha çox ardıcıl yazı bilərsiniz.

## Server tərəfdən route edilmə

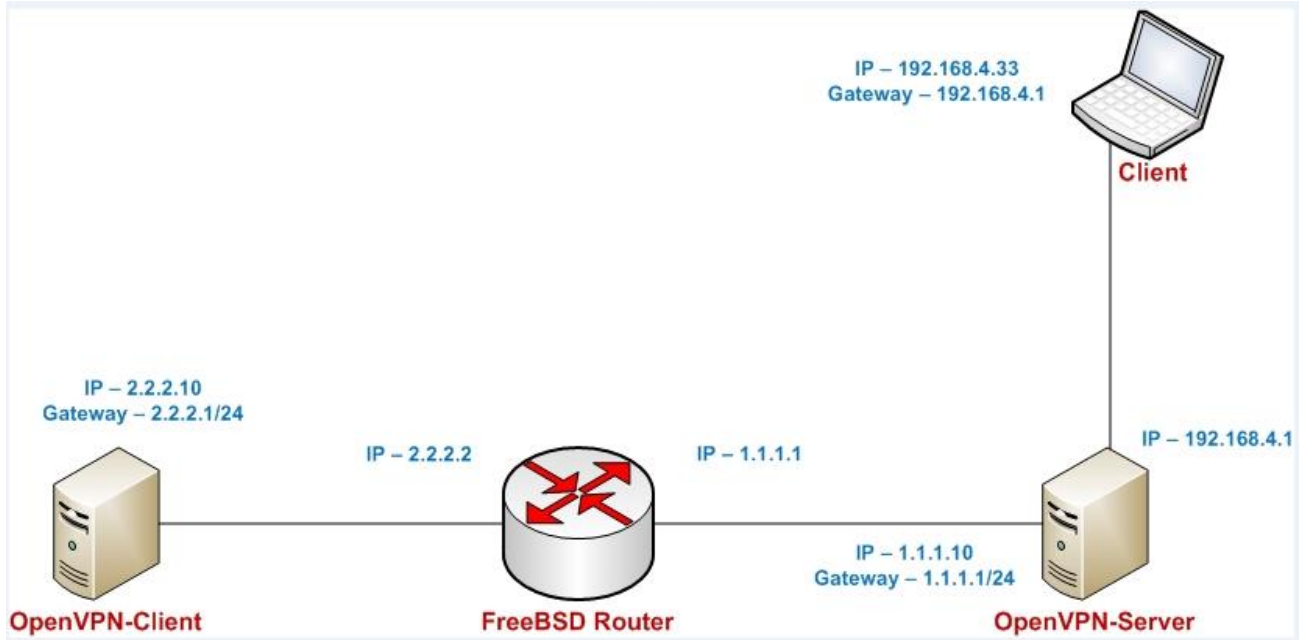
Bu misalda biz server tərəfdə olan routing-i həm client həm də server rejimində göstərəcəyik. Bu misalımızda isə OpenVPN client maşını OpenVPN server arxasında olan bütün maşınlarla qoşulmadan sonra çatmalıdır.

Öncəki misal ilə müqayisə elədikdə, bu misalda əlavə quraşdırmalar var hansı ki, əksər hallarda real istifadədə olur. OpenVPN-in öz funksionallığına baxsaq artıq linear ünvanları istifadə etmək imkanımız var(**topology subnet**).

Bu misalda istifadə edilən quraşdırma faylları həmçinin digər başlıqlarda istifadə edilmək üçün çox yaxşı quruluşa malikdir. Faylların adları **basic-udp-server.conf**, **basic-udp-client.conf** olacaq.

## İşə başlayaq

Biz aşağıdakı şəbəkə quruluşuna görə işlərimizi görəcəyik:



Bu misal PKI faylları istifadə edir hansı ki, bu başlığın ilk misalında yaradılmışdır. OpenVPN-i iki maşında yükləyin. Bizim misalda server və client maşın **FreeBSD 9.2 x64**-də işləyir.

### Bunu necə edək

1. Server üçün quraşdırma faylını yaradaq:

```
proto udp
port 1194
dev tun
server 192.168.200.0 255.255.255.0

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnsrver.crt
key /usr/local/etc/openvpn/openvpnsrver.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0

persist-key
persist-tun
keepalive 10 60

push "route 192.168.4.0 255.255.0.0"
topology subnet

user nobody
group nobody

daemon
log-append /var/log/openvpn.log
```

Sonra faylı **basic-udp-server.conf** adı ilə yadda saxlayın.

**Qeyd:** Bəzi Linux distributivlərində ola bilər ki, **nogroup** adlı qrup olsun.

**Qeyd:** Əgər siz OpenVPN-in reboot-dan sonra işləməsini istəsəniz onda onu startup-a əlavə etməlisiniz. Bunun üçün aşağıdakı sətirləri **/etc/rc.conf** faylına əlavə etsəniz yetər.

```
openvpn_enable="YES"
openvpn_if="tun"
openvpn_configfile="/usr/local/etc/openvpn/basic-udp-server.conf"
openvpn_dir="/usr/local/etc/openvpn"
```

2. Sonra **tls-auth** gizli açar faylını **/usr/local/etc/openvpn/itvpn/keys** ünvanından quraşdırmada olan ünvana nüsxələyin:

```
[root@siteA ~]# cp /usr/local/etc/openvpn/itvpn/keys/ta.key
/usr/local/etc/openvpn/
```

3. Və serveri işə salaq:

```
[root@siteA /usr/local/etc/openvpn]# openvpn --config basic-udp-
server.conf
```

4. Əmin olun ki, serverdə trafikın yönləndirilməsi aktivdir:

```
[root@siteA /usr/local/etc/openvpn]# sysctl -w net.inet.ip.forwarding=1
```

Yada **/etc/rc.conf** faylında **gateway\_enable="YES"** əlavə edin ki, rebootdan sonra işləsin.

5. Sonra client-in quraşdırma faylını yaradaq:

```
client
proto udp
remote openvpnserver.example.com
port 1194
dev tun
nobind

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnclient1.crt
key /usr/local/etc/openvpn/openvpnclient1.key
tls-auth /usr/local/etc/openvpn/ta.key 1
```

```
ns-cert-type server
```

Faylı **basic-udp-client.conf** adında yadda saxlayın.

Həmçinin **/etc/rc.conf** faylında **'1.1.1.10 openvpnserver.example.com'** sətirini yazmağı unutmayın.

6. **tls-auth** üçün client tərəfdə istifadə ediləcək **ta.key** faylını serverdən client-e **scp** ilə köçürün:

```
root@siteB:~ # scp 1.1.1.10:/usr/local/etc/openvpn/itvpn/keys/ta.key
/usr/local/etc/openvpn/
```

7. Client-i işə salaq:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config basic-udp-
```

### **client.conf**

```
Thu Jan 16 14:15:57 2014 OpenVPN 2.3.2 amd64-portbld-freebsd9.2 [SSL
(OpenSSL)] [LZO] [eurephia] [MH] [IPv6] built on Jan 9 2014
Thu Jan 16 14:15:57 2014 Control Channel Authentication: using
'/usr/local/etc/openvpn/ta.key' as a OpenVPN static key file
Thu Jan 16 14:15:57 2014 UDPv4 link local: [undef]
Thu Jan 16 14:15:57 2014 UDPv4 link remote: [AF_INET]1.1.1.10:1194
Thu Jan 16 14:15:57 2014 [openvpnserver] Peer Connection Initiated with
[AF_INET]1.1.1.10:1194
Thu Jan 16 14:15:59 2014 TUN/TAP device /dev/tun0 opened
Thu Jan 16 14:15:59 2014 do_ifconfig, tt->ipv6=0, tt-
>did_ifconfig_ipv6_setup=0
Thu Jan 16 14:15:59 2014 /sbin/ifconfig tun0 192.168.200.2
192.168.200.2 mtu 1500 netmask 255.255.255.0 up
add net 192.168.200.0: gateway 192.168.200.2 fib 0
add net 192.168.4.0: gateway 192.168.200.1 fib 0
Thu Jan 16 14:15:59 2014 Initialization Sequence Completed
```

8. İndi işe son nöqtədə olan Windows7 maşına ping ata bilərsiniz.

```
root@siteB:~ # ping -c2 192.168.4.33
PING 192.168.4.33 (192.168.4.33): 56 data bytes
64 bytes from 192.168.4.33: icmp_seq=0 ttl=127 time=1.192 ms
64 bytes from 192.168.4.33: icmp_seq=1 ttl=127 time=2.467 ms
```

### **Bu necə işləyir**

Server ilk boş olan **TUN** alətini işe salır və **192.168.200.1** IP ünvanını mənimsədir. **'topology subnet'** direktivi ilə yalançı uzaq ünvanda **192.168.200.1** IP ünvanı daşıyacaq. Bundan sonra server UDP port 1194-cü portda qulaq asmağa başlayır. Təhlükəsizlik üçün işe OpenVPN prosesini nobody istifadəçi və qrupu adından işe salırıq. Əgər Hacker OpenVPN-in prosesini hack eləsə o root ala bilməyəcək. Əgər **'user'** və **'group'** direktivləri istifadə edilirsə, həmçinin lazımdır ki, aşağıdakı sətirləridə quraşdırmanıza əlavə edəsiniz, əks halda servis düzgün **restart** edilməyəcək:

```
persist-key
persist-tun
```

Digər təhlükəsizlik işi işe aşağıdakı sətiri server tərəfdə istifadə eləməkdir (həmçinin **ta.key 1** client tərəfdə olmalıdır):

```
tls-auth /etc/openvpn/itvpn/ta.key 0
```

Bu serveri **Distributed Denial of Service (DDoS)** hücumlarının qarşısını almağa kömək edir. Və tez olaraq biz bu paketləri **HMAC** fərqli olan kimi ignore edirik.

Aşağıdakı sətir işe həm server və həm də client üçün yaşama müddəti təyin edir:

```
keepalive 10 60
```

Client-dən serverə və serverdən clientə hər bir 10 saniyədən sonra paket yollanılır ki, VPN tunelin işləyib işləməməsini yoxlayaq. Əgər cavab gəlməzsə 60 saniyədən sonra OpenVPN avtomatik olaraq restart ediləcək. Server tərəfdə işe gözləmə müddəti ikiyə vurulur. Yəni ki, server VPN prosesini 120 saniyədən sonra restart edəcək.

Sonda isə aşağıdakı direktivlər çox önəmlidir ona görə ki, OpenVPN servisini daemon rejimdə işə salır və jurnalları seçilmiş fayla yazır.

```
daemon
log-append /var/log/openvpn.log
```

Jurnalları çıxışda online baxmaq üçün siz **tail -f /var/log/openvpn.log** əmrindən istifadə edə bilərsiniz. Client serverə qoşulur və TLS razılaşma yerinə yetirildikdən sonra isə client və server sertifikatları istifadə edilir və client-ə **192.168.200.2** IP ünvanı mənimsədir. Client ilk boş olan TUN şəbəkə kartını quraşdırır və server tərəfdən arxasında olan subnet üçün routing məlumatını alır.

### Daha da ətraflı

Bu misalda istifadə etdiyimiz quraşdırma fayllarını biz birazdan daha da çox istifadə edəcəyik.

### Linear ünvanlama

Qoşulma uğurlu başa çatdıqdan sonra isə siz tun0 şəbəkə kartına aşağıdakı əmr ilə baxa bilərsiniz:

```
root@siteB:~ # ifconfig tun0 | grep inet
inet 192.168.200.2 --> 192.168.200.2 netmask 0xffffffff0
```

Bu **topology** direktivi sayəsində çağırılmışdır hansı ki, OpenVPN2.1 versiyasından başlayaraq yenidir. Bu OpenVPN-ə deyir ki, hər bir client-ə 1 ədəd IP ünvan mənimsət. OpenVPN2.0-da isə hər client üçün mənimsənilən IP ünvanın sayı 4 ədəd idi.

### TCP protocolun istifadəsi

Öncəki misalda biz UDP protocol-undan istifadə elədik. Ancaq quraşdırma faylları qısa formada TCP protokol-una aşağıdakı sətiri dəyişməklə convert edilə bilər:

```
proto udp
```

Bu aşağıdakı sətirə dəyişdirilməlidir:

```
proto tcp
```

Bu həm client həm də server quraşdırma fayllarında dəyişdirilməlidir. Bu faylları gələcək üçün saxlayın. Sadəcə adlarını **basic-tcp-server.conf** və **basic-tcp-client.conf** etməyiniz yetər.

### Server sertifikatları və ns-cert-type server

Client tərəfdə **ns-cert-type** server direktivi əksər hallarda server sertifikatı ilə kombinasiyada istifadə edilir hansı ki, aşağıdakı əmrlə qurulur:

```
build-key-server
```

Bu əsasən MITM hücumların qarşısını almaq üçün istifadə edilir. Idea odur ki, Client spesifik server sertifikatına malik olmazsa, serverə qoşula

bilməyəcək. Bunu eləməklə, hücumçunun özünü server rolunda oynaması şansını əlindən alır. OpenVPN 2.1-dən başlayaraq dəstək edir.

#### **remote-cert-tls server**

Bu həmçinin tam açarlar təyin edilmiş sertifikatları da dəstəkləyir və həmçinin RFC3280 TLS-də yazılmış açar istifadəsinin genişlənməsində dəstəkləyir.

### **Masquerading**

Bu başlıqda biz server tərəfdə olan LAN trafikini VPN üzərindən routing etmişik. Linux IPTABLES əmri ilə masquerading edilmişdir:

```
iptables -t nat -I POSTROUTING -o eth0 -s 192.168.200.0/24 -j MASQUERADE
```

Bu əmr Linux kernel-ə deyir ki, 192.168.200.0/24 (Bu OpenVPN subnetidir) subnetdən gələn bütün trafiki yönləndir və o Ethernet kartı tərk edir. Və hər bir bu paketin öz source ünvanı var amma o elə dəyişdirilir ki, guya bu OpenVPN client yox serverin özündən gəlir. Iptables bu dəyişdirilmiş paketləri izləyir və əgər qayıdan paket reversi düzdürsə, paketlər yenidən client-ə qaytarılır. Bu routing-in işləməsinin asan yoludur ancaq çatışmazlıq var. Çoxlu istifadəçi istifadə ediləndə, o SiteB tərəfdən gələn trafik ilə OpenVPN serverin özünün trafikini ayırd edə bilmir. O təyin edə bilmir ki, tunnel ilə gələn client1-dir yada client.

### **'client-config-dir' faylların istifadəsi**

Bir serverin çoxlu istifadəçiləri emal edə biləcəyi quraşdırmalarımızda elə hallar ola bilər ki, hər istifadəçinin özünə aid olan quraşdırmaları təyin edək hansı ki, oda **'global'** opsiyaları silib öz quraşdırmalarını onun yerinə yazacaq. Məhz bunun üçün də **client-config-dir** opsiyası əladır. Bu inzibatçıya hər client-ə spesifik IP ünvan, xüsusi opsiyaların təyin edilməsinə imkan yaradır (misal üçün hər client üçün fərqli DNS server, trafik sınırlanması ya da ümumiyyətlə həmin client-i deaktiv edilməsi)

### **İşə başlayaq**

Bu misal öncəkinin davamıdır. OpenVPN-i iki məşində yükləyin. Hər iki məşin **FreeBSD 9.2 x64**-də yüklənmişdir. Eynilə server üçün **basic-udp-server.conf** və client üçün uyğun olaraq öncəki misalda istifadə elədiyimiz **basic-udp-client.conf** fayllarını istifadə edin.

### **Bunu necə edək**

1. Server quraşdırma faylında dəyişik edək və aşağıdakı sətiri içinə əlavə edək:

```
client-config-dir /usr/local/etc/openvpn/clients
```

Sonra faylı **example2-4-server.conf** adı ilə yadda saxlayın.

2. Ardınca clientlərin quraşdırma faylları üçün qovluq və həmçinin client sertifikatının adı ilə eyni olan faylı serverdə yaradırıq. Client-in quraşdırma faylında client üçün IP ünvan və mask-i təyin edək:

```
root@siteA:/usr/local/etc/openvpn/clients # mkdir -m 755
/usr/local/etc/openvpn/clients
root@siteA:/ # cd /usr/local/etc/openvpn/clients/
root@siteA:/usr/local/etc/openvpn/clients # echo "ifconfig-push
192.168.200.6 255.255.255.240" > openvpnclient1
```
3. Adı siz client-in sertifikatından aşağıdakı əmr ilə əldə edə bilərsiniz.

```
root@siteB:/usr/local/etc/openvpn # openssl x509 -subject -noout -in
openvpnclient1.crt
subject= /C=AZ/O=Itvpn/CN=openvpnclient1/emailAddress=openvpn-
ca@domain.lan
```
4. Serveri işə salaq:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example2-4-
server.conf
```
5. Öncəki misalda istifadə elədiyimiz kimi client-in quraşdırma faylını işə salın:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config basic-udp-
client.conf
Sat Jan 18 17:34:47 2014 OpenVPN 2.3.2 amd64-portbld-freebsd9.2 [SSL
(OpenSSL)] [LZO] [eurephia] [MH] [IPv6] built on Jan 9 2014
Sat Jan 18 17:34:47 2014 Control Channel Authentication: using
'/usr/local/etc/openvpn/ta.key' as a OpenVPN static key file
Sat Jan 18 17:34:47 2014 UDPv4 link local: [undef]
Sat Jan 18 17:34:47 2014 UDPv4 link remote: [AF_INET]1.1.1.10:1194
Sat Jan 18 17:34:47 2014 [openvpnserver] Peer Connection Initiated with
[AF_INET]1.1.1.10:1194
Sat Jan 18 17:34:49 2014 TUN/TAP device /dev/tun0 opened
Sat Jan 18 17:34:49 2014 do_ifconfig, tt->ipv6=0, tt-
>did_ifconfig_ipv6_setup=0
Sat Jan 18 17:34:49 2014 /sbin/ifconfig tun0 192.168.200.6
192.168.200.6 mtu 1500 netmask 255.255.255.240 up
add net 192.168.200.0: gateway 192.168.200.6 fib 0
add net 192.168.4.0: gateway 192.168.200.1 fib 0
Sat Jan 18 17:34:49 2014 Initialization Sequence Completed
```

### **Bu necə işləyir**

Client serverə sertifikatla qoşulanda və client sertifikatın **Common Name** bölməsində olan ad openvpnclient1 olduqda, OpenVPN server yoxlamağa başlayır ki, həmin client-ə aid olan əlavə quraşdırma faylı hardadır. Eynilə quraşdırma faylını client-config-dir qovluğunda tapdıqdan sonra isə ona aid olan əlavə opsiyaları həmin fayldan götürüb mənimsədir. Quraşdırma faylında biz client-ə spesifik IP ünvan və secdiyimiz mask mənimsədirik. Yeni client-in IP ünvanı 192.168.200.6 və mask-i 255.255.255.240 olacaq.



## Daha da ətraflı

### Susmaya görə olan quraşdırma faylı

Əgər aşağıdakı şərtlər yerinə yetirilirsə, onda susmaya görə olan fayl oxunur və yerinə yetirilir o halda ki:

- **client-config-dir** direktivi təyin edilib
- Orda client-in sertifikatına uyğun olan quraşdırma fayl tapılmamışdır.
- DEFAULT file həmin qovluqda yoxdur.

Nəzərə alın ki, client-in adının böyük və kiçik hərflə yazılmasının fərqi var(**reqistra hissiyatlıdır**).

### Troubleshoot etmə

Quraşdırma problemlərinin CCD ilə araşdırılması və həll edilməsi, OpenVPN-in mail listində ən üstün təşkil edən hissəsidir. Əsas üzə çıxan quraşdırma səhvləri aşağıdakılardır:

- Həmişə **client-config-dir**-də qovluğun ünvanını tam yazın.
- Əmin olun ki, təyin etdiyiniz qovluq və CCD faylları OpenVPN prosesi tərəfindən oxunula bilir(əksər hallarda **nobody** və **openvpn** olur)
- Əmin olun ki, CCD qovluqda istifadə elədiyiniz ad düzgün Common Name-də olan istifadəçi adıdır və ad genişlənmə ilə yazıla bilməz.

### OpenVPN 2.0 'net30' uyğunluğu

OpenVPN2.0 topology subnet direktivini dəstəkləmir. O ancaq net30 rejimini dəstəkləyir hardaki, hər bir client /30 maskası alır. Həmin maskada 4 IP ünvan olur. CCD faylda net30 üçün sintaksis aşağıdakı kimi fərqlənir:

```
ifconfig-push 192.168.200.34 192.168.200.33
```

İlk olanı həmin /30 şəbəkəsində ilk mümkün olan client-in IP ünvanıdır hansı ki, **192.168.200[32-35]**. İkinci isə yalançı remote(uzaq) ünvanıdır hansı ki, heç vaxt istifadə edilməyəcək.

Bu həmçinin imkan yaradır ki, OpenVPN2.0-da quraşdırılmış clientləri elə **topology subnet** ilə quraşdırılmış serverlərə qoşula bilsinlər. CCD faylı yaradaraq aşağıdakı sətirləri əlavə etsək, OpenVPN2.0 clientləri hələ də qoşula bilər:

```
push "route-gateway 192.168.200.33"
ifconfig-push 192.168.200.34 192.168.200.33
```

**Qeyd:** Nəzərə alın ki, məcbur olaraq göstərilir ki, **192.168.200.1** IP ünvanı gateway kimi istifadə edilməsin və OpenVPN2.0-da olduğu kimi, **'topology net30'**-un istifadə edilməsinə heç bir gərəkdir.

### 'client-config-dir'-də istifadə edilən fayla izin verilmiş opsiyalar

Aşağıdakı quraşdırma opsiyalarının istifadəsinə CCD faylda izin verilmişdir:

- **push** - DNS, Wins serverlərin və route-un əlavə edilməsi üçün istifadə edilir.
- **push-reset** - global push opsiyasını silir və bunu əlavə edir

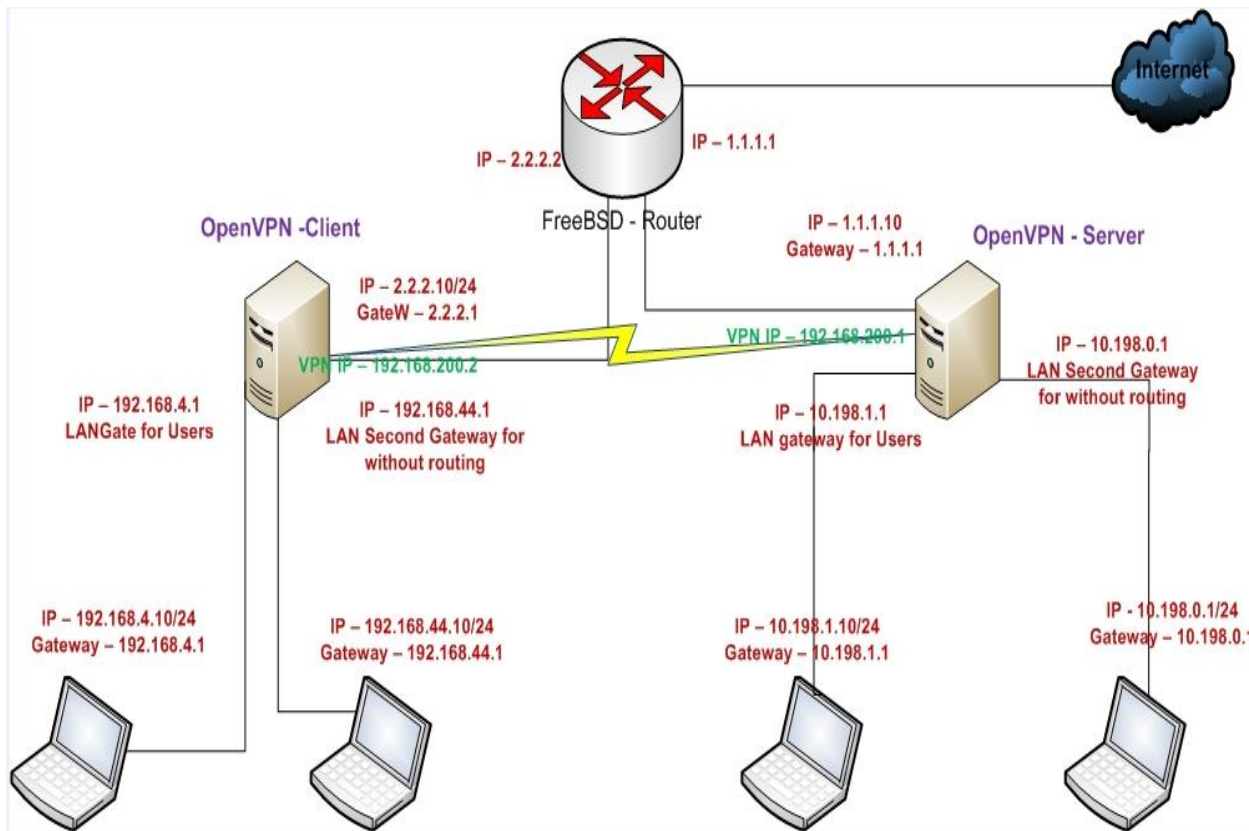
- **iroute** - client-in subnetlərini serverə route eləmək üçün istifadə edilir
- **ifconfig-push** - Bu misalda göstərdiyimiz kimi client-ə spesifik IP ünvan mənimsədir.
- **disable** - müəyyən vaxt üçün client-i ümumiyyətlə dayandırırıq.
- **config** - əlavə quraşdırma faylını artırmaq üçün istifadə edilir

### Hər iki tərəfin subnetlərinin route edilməsi

Bu resept göstərəcək ki, necə client/server rejimində həm client və həm də server tərəfi routing edəcəyik. Bu misalda OpenVPN client maşını OpenVPN server arxasında olan bütün maşınlarla və OpenVPN server həmçinin OpenVPN client-in arxasında olan maşınlarla çata biləcək.

### İşə başlayaq

Biz aşağıdakı şəbəkə quruluşunu istifadə edəcəyik:



Bu misal PKI faylları istifadə edir hansı ki, bu başlığın əvvəlində yaratmışdıq. Misalımızda yenə də FreeBSD9.2 x64 həm client və həm də server tərəfdə istifadə edilir. Server quraşdırma faylı elə **basic-udp-server.conf** və client quraşdırma faylı isə **basic-udp-client.conf** olacaq hansı ki, **Server-side routing**-də istifadə eləmişdik.

### Bunu necə edək

1. **basic-udp-server.conf** faylını başqa fayla nüsxələyin və içinə aşağıdakı sətirləri əlavə edin:

```
client-config-dir /usr/local/etc/openvpn/clients
route 192.168.4.0 255.255.255.0 192.168.200.1 #SiteB serverə
 qoşulanda bu route
 serverdə yazılacaq
```

Sonra isə faylı **example2-5-server.conf** adı ilə yadda saxlayın. Server Tərəfdə olan strukturu açıqlayaq ki, işinizi tam başa düşəsiniz. SiteA bizim Serverdir. SiteB isə Client-imiz. SiteA-da **/etc/rc.conf** faylı aşağıdakı kimidir.

```
ifconfig_em0="inet 1.1.1.10 netmask 255.255.255.0"
ifconfig_em1="inet 10.198.1.1 netmask 255.255.255.0"
ifconfig_em2="inet 10.198.0.1 netmask 255.255.255.0"
defaultrouter="1.1.1.1"
gateway_enable="YES" # Routing rejimi aktivdir
hostname="siteA"
```

Ümumiyyətlə serverin quraşdırma faylı yeni **example2-5-server.conf** faylı aşağıdakı kimi olacaq.

```
proto udp
port 1194
dev tun
server 192.168.200.0 255.255.255.0
```

```
client-config-dir /usr/local/etc/openvpn/clients
route 192.168.4.0 255.255.255.0 192.168.200.2 # SiteA-da VPN
 qalxanda
 192.168.4.0/24
 şəbəkəsini görmək
 üçün,
 192.168.200.2 IP-
 si üzərindən
 keçməsi üçün
 routing-i özündə
 əlavə edəcək.
 192.168.200.1
 bizim SiteA-nın
 öz IP ünvanı
 olacaq
```

```
ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnserver.crt
key /usr/local/etc/openvpn/openvpnserver.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0
```

```
persist-key
persist-tun
keepalive 10 60
```

```

push "route 10.198.1.0 255.255.255.0" # SiteB qoşulanda bu routu
 SiteA üstünə yazacaq

topology subnet

user nobody
group nobody

daemon
log-append /var/log/openvpn.log

```

2. Sonra isə client-in quraşdırma faylları üçün CCD(Client Config Directory) qovluq yaradın:

```

root@siteA:/usr/local/etc/openvpn # mkdir -m 755
/usr/local/etc/openvpn/clients

```

3. Client sertifikatında **Common Name**-də olduğu kimi, eyni adlı faylı **"/usr/local/etc/openvpn/clients"** qovluğunda yaradın. Yəni

```

openvpnclient1 faylını yaradın və içinə aşağıdakı sətiri əlavə edin:
iroute 192.168.4.0 255.255.255.0 # Eynilə openvpnclient1
 qoşulanda bu routing-i
 serverə verir

```

Siz bu adı client-in sertifikatından aşağıdakı əmr ilə əldə edə bilərsiniz:

```

root@siteA:/usr/local/etc/openvpn/itvpn/keys # openssl x509 -subject -
noout -in openvpnclient1.crt
subject= /C=AZ/O=Itvpn/CN=openvpnclient1/emailAddress=openvpn-
ca@domain.lan

```

Client tərəfdə yeni SiteB-də **/etc/rc.conf** faylı aşağıdakı quruluşa malik olacaq.

```

ifconfig_em0="inet 1.1.1.10 netmask 255.255.255.0"
ifconfig_em1="inet 10.198.1.1 netmask 255.255.255.0"
ifconfig_em2="inet 10.198.0.1 netmask 255.255.255.0"
defaultrouter="1.1.1.1"
gateway_enable="YES" # Routing rejimi ishleyir
hostname="siteA"

```

Eynilə SiteB-nin quraşdırma faylı aşağıdakı kimi olacaq.

```

client
proto udp
remote openvpnsrvr.example.com
port 1194
dev tun
nobind

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnclient1.crt
key /usr/local/etc/openvpn/openvpnclient1.key
tls-auth /usr/local/etc/openvpn/ta.key 1

ns-cert-type server

```

Quraşdırma faylını **basic-udp-client.conf** adında yadda saxlayın.

4. Serveri işə salaq.

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example2-5-server.conf
```

5. Client-i işə salın:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config basic-udp-client.conf
Sun Jan 19 13:50:50 2014 OpenVPN 2.3.2 amd64-portbld-freebsd9.2 [SSL (OpenSSL)]
[LZO] [eurephia] [MH] [IPv6] built on Jan 9 2014
Sun Jan 19 13:50:50 2014 Control Channel Authentication: using
'/usr/local/etc/openvpn/ta.key' as a OpenVPN static key file
Sun Jan 19 13:50:50 2014 UDPv4 link local: [undef]
Sun Jan 19 13:50:50 2014 UDPv4 link remote: [AF_INET]1.1.1.10:1194
Sun Jan 19 13:50:50 2014 [openvpnsrver] Peer Connection Initiated with
[AF_INET]1.1.1.10:1194
Sun Jan 19 13:50:53 2014 TUN/TAP device /dev/tun0 opened
Sun Jan 19 13:50:53 2014 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0
Sun Jan 19 13:50:53 2014 /sbin/ifconfig tun0 192.168.200.2 192.168.200.2 mtu
1500 netmask 255.255.255.0 up
add net 192.168.200.0: gateway 192.168.200.2 fib 0
add net 10.198.1.0: gateway 192.168.200.1 fib 0
Sun Jan 19 13:50:53 2014 Initialization Sequence Completed
```

6. Qoşulma bitdikdən sonra işə şəkildəki topologiyamıza uyğun olaraq, həm SiteA-dan SiteB-yə 2 ədəd ping paketləri yollayaq.

```
root@siteA:/usr/local/etc/openvpn # ping -c2 192.168.4.10
PING 192.168.4.10 (192.168.4.10): 56 data bytes
64 bytes from 192.168.4.10: icmp_seq=0 ttl=127 time=1.154 ms
64 bytes from 192.168.4.10: icmp_seq=1 ttl=127 time=2.428 ms
```

7. Eynilə SiteB-dən SiteA-ya 2 ədəd ping paketləri yollayaq.

```
root@siteB:~ # ping -c2 10.198.1.10
PING 10.198.1.10 (10.198.1.10): 56 data bytes
64 bytes from 10.198.1.10: icmp_seq=0 ttl=127 time=1.525 ms
64 bytes from 10.198.1.10: icmp_seq=1 ttl=127 time=1.618 ms
```

8. Ancaq şəkildə gördüyümüz kimi SiteA-dan 192.168.44.1-ə ping atsaq və SiteB-dən 10.198.1.1-ə ping atsaq getməyəcək.

```
root@siteA:/usr/local/etc/openvpn # ping -c2 192.168.44.1
PING 192.168.44.1 (192.168.44.1): 56 data bytes
36 bytes from 1.1.1.1: Destination Host Unreachable
Vr HL TOS Len ID Flg off TTL Pro cks Src Dst
4 5 00 5400 1c03 0 0000 3f 01 70f2 1.1.1.10 192.168.44.1
```

```
root@siteB:~ # ping 10.198.0.1
PING 10.198.0.1 (10.198.0.1): 56 data bytes
36 bytes from 2.2.2.1: Destination Host Unreachable
Vr HL TOS Len ID Flg off TTL Pro cks Src Dst
4 5 00 5400 0c3e 0 0000 3f 01 6099 2.2.2.10 10.198.0.1
```

9. Bunu etmək üçün işə siz serverin quraşdırma faylında eynilə uyğun şəbəkələr üçün routing əlavə etməlisiniz. Yeni aşağıdakı kimi.

```
proto udp
```

```
port 1194
dev tun
server 192.168.200.0 255.255.255.0
ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnserver.crt
key /usr/local/etc/openvpn/openvpnserver.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0
persist-key
persist-tun
keepalive 10 60

push "route 10.198.1.0 255.255.255.0"
push "route 10.198.0.0 255.255.255.0"
topology subnet

user nobody
group nobody

daemon
log-append /var/log/openvpn.log
client-config-dir /usr/local/etc/openvpn/clients
route 192.168.4.0 255.255.255.0 192.168.200.2
route 192.168.44.0 255.255.255.0 192.168.200.2
```

Həmçinin client spesifik faylda da əlavə etməlisiniz aşağıdakı kimi:

```
iroute 192.168.4.0 255.255.255.0
iroute 192.168.44.0 255.255.255.0
```

### **Bu necə işləyir**

Client öz sertifikatı ilə server qoşulduqda və onun sertifikatınının Common Name-ində **openvpnclient1** olduqda, OpenVPN server client-in quraşdırma faylını və həmçinin client-config-dir-də özünə aid olan faylı (CCD faylı) oxuyur. Aşağıdakı sətirlə OpenVPN server deyir ki, 192.168.4.0/24 subnet-i openvpnclient1 istifadəçisinin üzərindən görmək olar:

```
iroute 192.168.4.0 255.255.255.0
```

Bu direktiv sistem-in kernel səviyyəsində olan route cədvəli ilə heç bir əlaqəyə girmir və yalnız OpenVPN-in daxili prosesi üzərində routing yazır.

Aşağıdakı server direktivi isə OpenVPN tərəfindən istifadə edilir ki, serverin özünün OS səviyyəsində 192.168.4.0/24 şəbəkəsinə müraciətləri getdikdə onu 192.168.200.2 IP ünvanlı interfeysesə yönləndirsin. Bu IP ünvan (Yeni 192.168.200.1) elə VPN serverin özüdür:

```
route 192.168.4.0 255.255.255.0 192.168.200.2
```

Gördüyünüz kimi artıq tam olaraq hər iki tərəf üçün site-to-site routing yerinə yetirildi.

## Daha da ətraflı

### Masquerading

Biz həmçinin masqalamanı hər iki sonda istifadə edə bilərik ki, çoxlu istifadəçidən istifadə edək. Ancaq bu halda istifadəçilərin trafikinin idarə edilməsi çox çətin olacaq.

### Client-to-client subnet routing

Əgər hansısa başqa bir istifadəçi olsa ki, `openvpnclient1`-in arxasındakı routing-i görsün onda, serverin quraşdırma faylına aşağıdakı sətiri əlavə etmək lazım olacaq.

```
push "route 192.168.4.0 255.255.255.0"
```

Bu sətir bütün müştərilərə başa salır ki, **192.168.4.0/24** şəbəkəsinə VPN tunel üzərindən çatmaq mümkündür yalnız, `openvpnclient1`-in özünü çıxmaq şərtilə. `openvpnclient1`-in özü isə `iroute` verilənləri ilə üst-üstə düşdüynə görə çıxarılır.

### Həmçinin baxaq

- 1-ci başlıqda Site-to-Site VPN açıqlanır hansı ki, Point-to-Point quruluşunda iki uzaq network-u VPN tunel üzərindən necə daşımaq lazımdır.

### Default gateway-in yönləndirilməsi

VPN əsas istifadəsinin mənası odur ki, bütün trafiki təhlükəsiz tunel üzərinə yönləndirmək imkanı var. Bu bizə şərait yaradır ki, hətta ən zəif qorunan və virusla dolu olan şəbəkənin üzərindən öz şəbəkəmizə heç bir narahatçılıq olmadan daxil ola bilək. Elə məhz bu misalımızda biz bunu edəcəyik. Bu misal demək olar ki, server-side routing misalına çox oxşardır ancaq, bütün trafikin VPN tunel üzərindən ötürülməsində bəzi çətinliklər var.

### İşə hazırlaşaq

Bu misalda istifadə elədiyimiz şəbəkə elə **Server-Side** routing ilə eynidir. Bu misalda da həmçinin başlığın əvvəlində yaratdığımız PKI açarlardan istifadə edəcəyik. Bu misalda da əvvəllər olduğu kimi, client və server üçün FreeBSD 9.2 x64 maşını və OpenVPN 2.3-dən istifadə edəcəyik. Quraşdırma faylını Server-side routing-də istifadə elədiyimizi elə burda da istifadə edəcəyik. Server üçün `basic-udp-server.conf` və client üçün `basic-udp-client.conf` istifadə edəcəyik.

### Bunu necə edək

1. `basic-udp-server.conf` faylını `example2-6-server.conf` adlı fayla nüsxələyin və faylın sonuna aşağıdakı sətiri əlavə edin:  

```
push "redirect-gateway def1"
```
2. Serveri işə salın:  

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example2-6-server.conf
```

3. Serverin başqa terminalında routing rejimdə işləməsi üçün aşağıdakı əmri daxil edin:  
root@siteA:/usr/local/etc/openvpn # **sysctl -w net.inet.ip.forwarding=1**
4. Client-i işə salaq:  
root@siteB:/usr/local/etc/openvpn # **openvpn --config basic-udp-client.conf**  
Sun Jan 19 16:24:46 2014 OpenVPN 2.3.2 amd64-portbld-freebsd9.2 [SSL (OpenSSL)] [LZO] [eurephia] [MH] [IPv6] built on Jan 9 2014  
Sun Jan 19 16:24:46 2014 Control Channel Authentication: using '/usr/local/etc/openvpn/ta.key' as a OpenVPN static key file  
Sun Jan 19 16:24:46 2014 UDPv4 link local: [undef]  
Sun Jan 19 16:24:46 2014 UDPv4 link remote: [AF\_INET]1.1.1.10:1194  
Sun Jan 19 16:24:46 2014 [openvpnserver] Peer Connection Initiated with [AF\_INET]1.1.1.10:1194  
Sun Jan 19 16:24:48 2014 TUN/TAP device /dev/tun0 opened  
Sun Jan 19 16:24:48 2014 do\_ifconfig, tt->ipv6=0, tt->did\_ifconfig\_ipv6\_setup=0  
Sun Jan 19 16:24:48 2014 /sbin/ifconfig tun0 192.168.200.2 192.168.200.2 mtu 1500 netmask 255.255.255.0 up  
add net 192.168.200.0: gateway 192.168.200.2 fib 0  
add net 1.1.1.10: gateway 2.2.2.1 fib 0  
**add net 0.0.0.0: gateway 192.168.200.1 fib 0**  
add net 128.0.0.0: gateway 192.168.200.1 fib 0  
add net 10.198.0.0: gateway 192.168.200.1 fib 0  
Sun Jan 19 16:24:48 2014 Initialization Sequence Completed
5. VPN qoşulması uğurlu olduğdan sonra əmin olun ki, bütün trafik tunel üzərindən keçir:  
root@siteB:~ # **traceroute 8.8.8.8**  
traceroute to 8.8.8.8 (8.8.8.8), 64 hops max, 52 byte packets  
1 **192.168.200.1** (192.168.200.1) 0.980 ms 2.375 ms 1.857 ms  
  
Traceroute əmrinin nəticəsində ilk ünvan OpenVPN serverin IP ünvanıdır və uyğun olaraq bütün trafik tunel üzərindən keçir.

### **Bu necə işləyir**

Client OpenVPN serverə qoşulduqda, spesifik route server tərəfindən OpenVPN client-ə təyin edilir:

```
push "redirect-gateway def1"
```

Quraşdırmada **option def1** deyir ki, OpenVPN client aşağıdakı göstərilən 3 ədəd routing-i öz əməliyyat sistemində əlavə etməlidir:

```
add net 10.198.0.0: gateway 192.168.200.1 fib 0
add net 0.0.0.0: gateway 192.168.200.1 fib 0
add net 128.0.0.0: gateway 192.168.200.1 fib 0
```

İlk route client tərəfindən serverin **10.198.0** şəbəkəsini görmək üçündür. Həmçinin tunel üzərindən keçir. Sonrakı iki route isə susmaya görə olan maşın gateway-ə deyir ki, artıq sən üstünlük daşımırsan və bütün trafik vpn tərəfindən yazılan gateway IP ilə tunelin üzərindən keçəcək.



## Daha da ətraflı

### Redirect-gateway parametrləri

Reallığa qalsa OpenVPN yalnız aşağıdakı direktivi dəstəkləyir:

```
push "redirect-gateway"
```

Bu serverin original default routunu silib OpenVPN serverə üzərinə gedən default route-u yazmaq üçün istifadə edilir. Bu daha düzgün üsul sayılır ancaq, bəzi hallarda isə OpenVPN mövcud olan default route-u tapa bilmir. Bu adətən clientlərin UMTS (Universal Mobile Telecommunications System) ilə qoşulduqlarında olur. Bu həmçinin lookup routelərin edilməsində də istifadə edilir hansı ki, bütün trafik tunel üzərindən ötürülür və clientlərdə daxil olmaqla.

OpenVPN ilə redirect-gateway direktivində çoxlu flaglar var və onları açıqlayaq:

- **local:** Bu clientdən serverə birbaşa route əlavə etmir. Bu o zaman istifadə edilir ki, client və server eyni LAN şəbəkə üzərindədirlər. Misal üçün wireless şəbəkəsi.
- **bypass-dhcp:** Bu local DHCP serverə birbaşa route əlavə edir. Bu avtomatik olaraq Windows clientlər üçün nəzərdə tutulub. Digər OS-lar üçün isə plugin və ya script tələb edilir.
- **bypass-dns:** Bu local DNS serverə birbaşa route əlavə edir. Buda həmçinin windows clientlər üçün nəzərdə tutulur və digər OS-lar üçün plugin və ya script tələb edilir.

## Split tunneling

Bəzi hallar ola bilər ki, **redirect-gateway** parametri çox məhdudiyətli ola bilər. Ola bilər ki, siz istəyəsiniz ki, müəyyən routingləri local şəbəkəyə və digər qalan trafik üçün isə VPN tunel-in üstünə yazasınız.

- **net\_gateway:** Bu spesifik gateway-dir hansı ki, LAN gateway ünvanıdır və OpenVPN tərəfindən işə düşəndə təyin edilir. Misal üçün LAN 192.168.4.0/24 şəbəkəsinə birbaşa route əlavə etmək üçün siz aşağıdakı sətiri client quraşdırma faylına əlavə etməlisiniz:  

```
route 192.168.4.0 255.255.255.0 net_gateway
```
- **vpn\_gateway:** Bu spesifik gateway-dir hansı ki, VPN gateway ünvanını göstərir. Əgər siz seçdiyiniz subnet üçün trafikin VPN tunel üzərindən keçməsi üçün route yazmaq istəyirsinizsə aşağıdakı əmrədən istifadə edə bilərsiniz (nəzərə alın ki, bu bütün local marşrutları təkzib edir):
- **route 10.198.0.0 255.255.0.0 vpn\_gateway:** Bu tip opsiya əsasən TAP tipli alətlərdə istifadə edilir hansı ki, VPN gateway öncədən belli olmur.

## Həmçinin baxaq

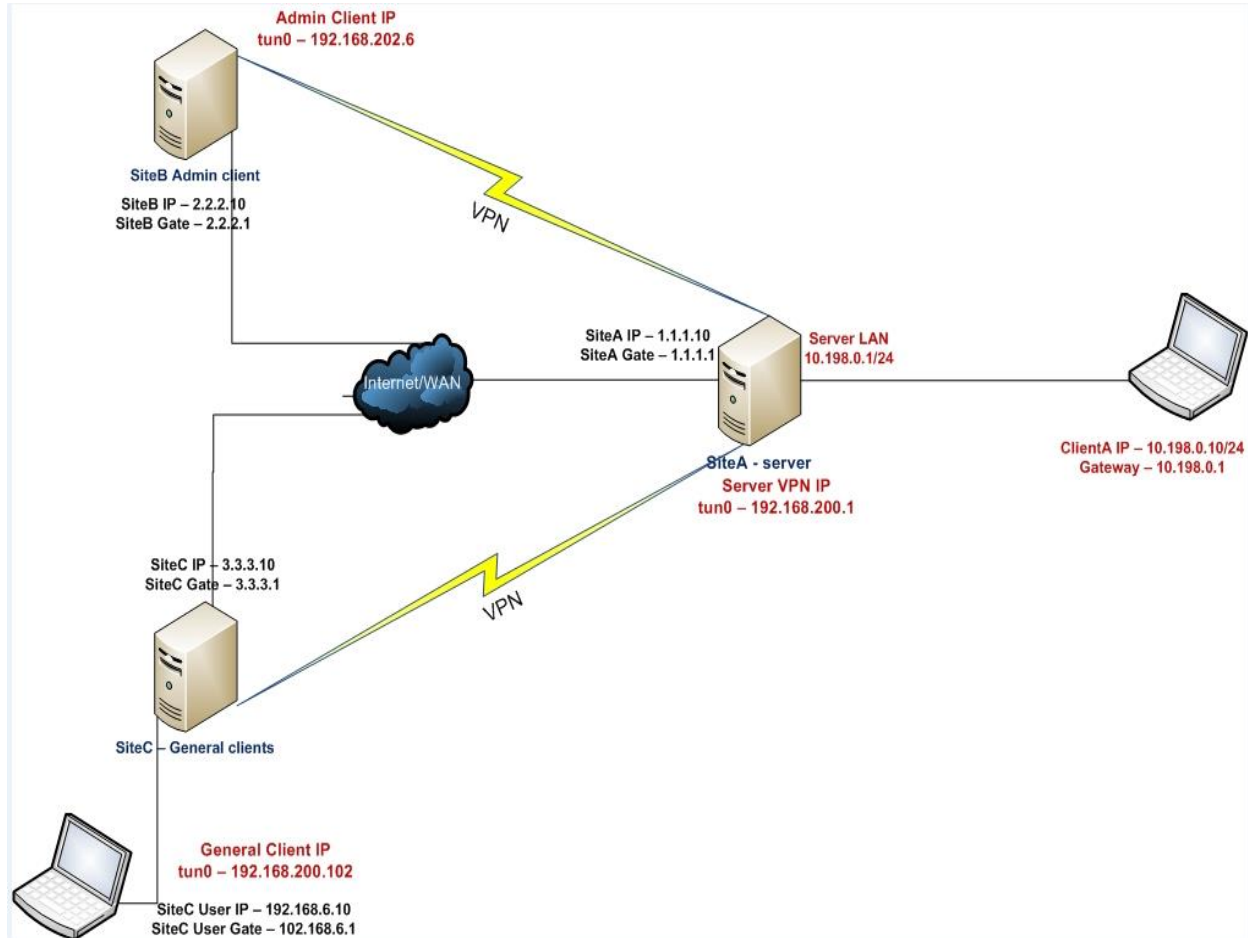
Server-sidə routing-ə baxın hansı ki, başlanğıc səviyyədə server tərəf routing-i açıqlayır.

## 'ifconfig-pool' block-un istifadə edilməsi

Bu misalımızda biz **ifconfig-pool** block-undan istifadə edəcəyik ki, adi VPN istifadəçilərini administrative VPN istifadəçilərindən ayıraq. Bu bizim işimizi asan edir ki, Firewall tərəfindən inzibatçı istifadəçilər üçün spesifik qaydalar yazı bilək.

### Başlanğıc tələblər

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



Misalımızda başlığımızın önündə yaratdığımız PKI açarlardan yenidən istifadə edəcəyik. Bu misalımızda da **Serverimiz FreeBSD9.2 x64**-də olacaq. **VPN adi client** isə **Windows7**-də olacaq və **192.168.200.0** VPN şəbəkəsində olacaq. **Admin VPN client** isə **FreeBSD9.2 x64**-də olacaq və **192.168.202.0** şəbəkəsində olacaq. **Linux/UNIX** maşınlarda olan clientlər üçün isə **basic-udp-client.conf** faylından istifadə edəcəyik hansı ki, Server-side routing-də istifadə eləmişdik..

### İşimizə başlayaq

1. **example2-7-server.conf** adlı server quraşdırma faylınyaradaq və içinə aşağıdakı tərkibi əlavə edək:

```
proto udp
port 1194
```

```
dev tun

mode server
tls-server

ifconfig 192.168.200.1 192.168.200.2
ifconfig-pool 192.168.200.100 192.168.200.120
route 192.168.200.0 255.255.248.0
push "route 192.168.200.1"
push "route 192.168.200.0 255.255.248.0"

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnsrvr.crt
key /usr/local/etc/openvpn/openvpnsrvr.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0

persist-key
persist-tun
keepalive 10 60

user nobody
group nobody

daemon
log-append /var/log/openvpn.log

client-config-dir /usr/local/etc/openvpn/clients
```

**Qeyd:** Nəzərə alın ki, burda **topology subnet** istifadə edilməmişdir.

2. Serveri işə salaq:  
root@siteA:/usr/local/etc/openvpn # **openvpn --config example2-7-server.conf**
3. Administrativ VPN client spesifik IP ünvanla təyin ediləcək:  
root@siteA:/usr/local/etc/openvpn # **mkdir -m 755 /usr/local/etc/openvpn/clients**  
root@siteA:/usr/local/etc/openvpn # **cd /usr/local/etc/openvpn/clients**  
root@siteA:/usr/local/etc/openvpn/clients # **cp openvpnclient1 backupopenvpnclient1**  
root@siteA:/usr/local/etc/openvpn/clients # **echo "ifconfig-push 192.168.202.6 192.168.202.5" > openvpnclient1**
4. Unutmayın **clients** qovluğu hər kəs tərəfindən oxunula bilən olmalıdır ona görə ki, OpenVPN server prosesi **nobody** adından işə düşür.
5. Sonra isə biz FreeBSD OpenVPN client-i öncəki misallarımızda istifadə elədiyimiz **basic-udp-client.conf** faylı ilə işə salaq:  
root@siteB:/usr/local/etc/openvpn # **openvpn --config basic-udp-client.conf**  
Mon Jan 20 17:00:56 2014 [openvpnsrvr] Peer Connection Initiated with [AF\_INET]1.1.1.10:1194

```

Mon Jan 20 17:00:58 2014 TUN/TAP device /dev/tun0 opened
Mon Jan 20 17:00:58 2014 do_ifconfig, tt->ipv6=0, tt-
>did_ifconfig_ipv6_setup=0
Mon Jan 20 17:00:58 2014 /sbin/ifconfig tun0 192.168.202.6
192.168.202.5 mtu 1500 netmask 255.255.255.255 up
add net 192.168.200.1: gateway 192.168.202.5 fib 0
add net 192.168.200.0: gateway 192.168.202.5 fib 0
Mon Jan 20 17:00:58 2014 Initialization Sequence Completed

```

Gördüyünüz kimi Admin client üçün ayrılmış IP ünvanı işarələnmişdir.

6. İndi işə Windows client üçün quraşdırma faylını aşağıdakı tərkib ilə yaradın:

```

client
proto udp
remote openvpnserver.example.com
port 1194
dev tun
nobind
auth-nocache

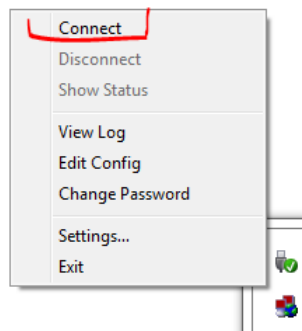
ca "c:/Program files/openvpn/config/ca.crt"
cert "c:/Program files/openvpn/config/openvpnclient2.crt"
key "c:/Program files/openvpn/config/openvpnclient2.key"
tls-auth "c:/program files/openvpn/config/ta.key" 1

```

**ns-cert-type server**

Sonra faylı **basic-udp-client.ovpn** adında yadda saxlayın. Nəzərə alın ki, tərs slash '\\' əvəzinə düz slash '/'-dən istifadə edilib.

7. Ardınca **ca.crt**, **openvpnclient2.crt**, **openvpnclient2.key** və **tls-auth** üçün istifadə edilən **ta.key** faylını Windows maşına köçürün. Bunu WinSCP ilə və ya Putty PSCP ilə götürə bilərsiniz.
8. Windows maşınının **c:\windows\system32\drivers\etc\hosts** faylına **'1.1.1.10 openvpnserver.example.com'** sətirinə əlavə edin və OpenVPN GUI istifadə edərək Windows client-i işə salın:



Unutmayın client-in private key-i şifrə tələb edəcək çünki, siz onu generasiya elədikdə, özünüz şifrəni təyin etmişdiniz. Həmçinin şifrənin cache-də qalmaması üçün biz clientin quraşdırma faylına **"auth-nocache"**

əlavə etmişik. Hər iki client qoşulduqdan sonra onların hər birini və serveri ping ilə yoxlaya bilərik (nəzərə alın ki, burda firewall yoxdur).

9. Öncə Admin client-də yoxlayaq (Yəni SiteB-də):

```
root@siteB:/usr/local/etc/openvpn # ping -c2 192.168.200.1
PING 192.168.200.1 (192.168.200.1): 56 data bytes
64 bytes from 192.168.200.1: icmp_seq=0 ttl=64 time=30.117 ms
64 bytes from 192.168.200.1: icmp_seq=1 ttl=64 time=1.631 ms

root@siteB:/usr/local/etc/openvpn # ping -c2 192.168.200.102
PING 192.168.200.102 (192.168.200.102): 56 data bytes
64 bytes from 192.168.200.102: icmp_seq=0 ttl=127 time=2.293 ms
64 bytes from 192.168.200.102: icmp_seq=1 ttl=127 time=2.114 ms
```

10. İndi isə adi client-də yoxlayaq.

```
C:\Users\ClientC>ping -n 2 192.168.200.1
Pinging 192.168.200.1 with 32 bytes of data:
Reply from 192.168.200.1: bytes=32 time=1ms TTL=64
Reply from 192.168.200.1: bytes=32 time=2ms TTL=64

C:\Users\ClientC>ping -n 2 192.168.202.6
Pinging 192.168.202.6 with 32 bytes of data:
Reply from 192.168.202.6: bytes=32 time=2ms TTL=63
Reply from 192.168.202.6: bytes=32 time=2ms TTL=63
```

### Bu necə işləyir

Server quraşdırma faylı adi qaydada olduğu kimi aşağıdakı direktivi istifadə edərək client-lər üçün IP aralığı təyin edir:

```
server 192.168.200.0 255.255.255.0
```

Bu direktiv daxili olaraq aşağıdakı hissələrə bölünmüşdür:

```
mode server
tls-server
```

```
ifconfig 192.168.200.1 192.168.200.2
ifconfig-pool 192.168.200.4 192.168.200.251
route 192.168.200.0 255.255.255.0
push "route 192.168.200.1"
```

Server direktivini təyin etməsəkdə, ancaq öz ifconfig-pool-muzu təyin etməklə nəticəni dəyişə bilərik. Sonra isə biz spesifik CCD fayl istifadə edirik ki, admin client üçün spesifik IP ünvan təyin edək hansı ki, ifconfig-pool aralığından kənar subnet-də yerləşir. Ancaq uyğun olan **push** və **push route** bölmələrini istifadə etməklə biz təminat veririk ki, bütün clientlər digərlerini **ping** ilə görə bilərlər.

### Daha da ətraflı

#### Windows maşında quraşdırma faylları

Windows maşında işləyən **OpenVPN GUI** proqramı həmişə quraşdırmaları aşağıdakı qovluqdan oxuyur:

**C:\Program Files\OpenVPN\config**

Eynilə sertifikatlar üçün ünvanı da eyni seçmişik ona görə ki, quraşdırma faylımızda oranı göstərmişik.

### **Topology subnet**

Nəzərə alın ki, bu misalımızda biz aşağıdakı direktivdən istifadə etmədik:  
**topology subnet**

**subnet** topology OpenVPN2.1-de yaranıb və **ifconfig-pool** ilə birgə istifadə edə bilməz.

### **Client-to-client yetkisi**

Bu imkanın sayəsində VPN clientlər bir-birinə qoşula bilərlər ancaq, biz aşağıdakı direktivi server-side quraşdırmasında istifadə eləməmişik:

**client-to-client**

Bunu server quraşdırma faylında **push route** və **route** bölmələri ilə eləmək mümkündür. **client-to-client** direktivinin istifadə edilməməsinin üstünlüyü ondan ibarətdir ki, istənilməyən trafik **IPTABLES** və ya **IPFW** firewall sayəsində filter edə bilərsiniz.

Əgər inzibatçı clientlərin adi clientlərə qoşulmasına (və əksinə) ehtiyac yoxdursa onda netmask aşağıdakı kimi quraşdırıla bilər:

```
route 192.168.200.0 255.255.255.0
push "route 192.168.200.0 255.255.255.0"
```

İndi şəbəkələr tam olaraq ayrılmışdır.

### **TCP protocol-un istifadə edilməsi**

Bu misalda biz UDP protocol seçdik. Client quraşdırma faylı TCP protocol ilə işləməsi üçün sadəcə aşağıdakı sətiri dəyişməklə işlədə bilərsiniz:

```
proto udp
```

Dəyişirik aşağıdakı sətirə:

```
proto tcp
```

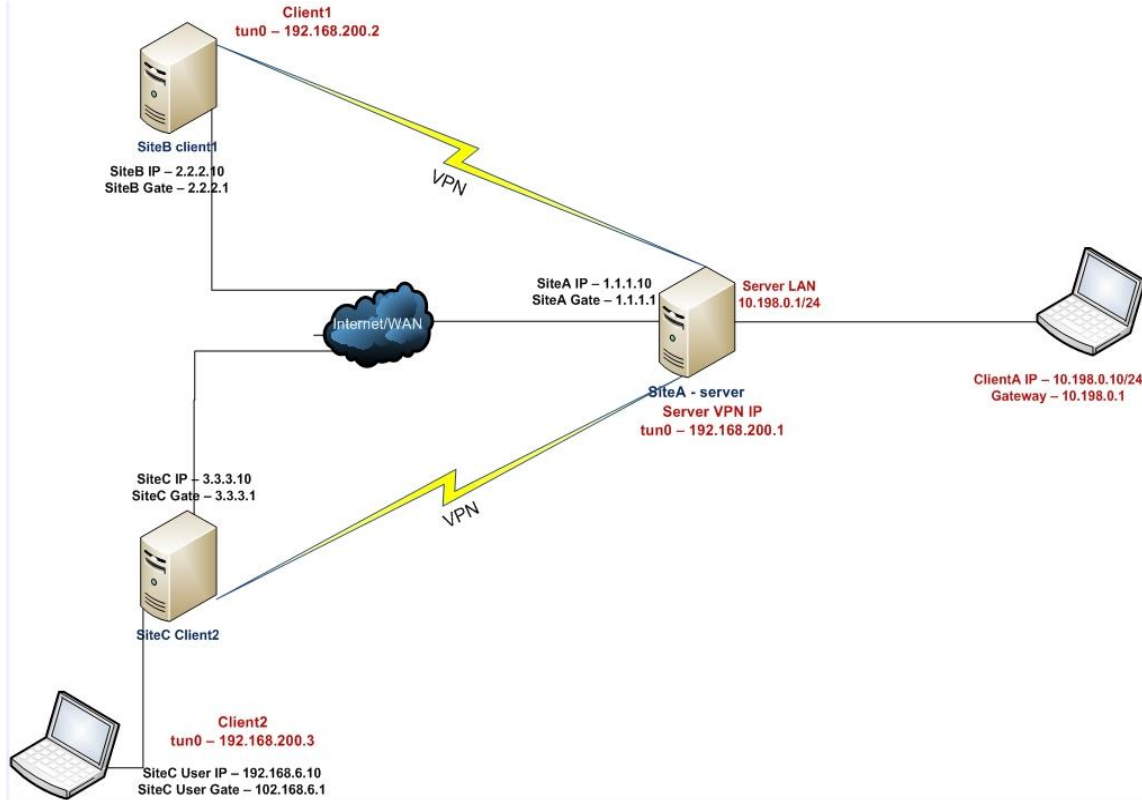
Və faylı Windows maşında **basic-tcp-client.ovpn** adı ilə yadda saxlayın.

### **Status faylının istifadəsi**

OpenVPN öz serverinə qoşulmuş clientlərin monitoring üçün bir neçə üsul təklif edir. Ən çox istifadə edilən üsul **status** faylıdır. Bu misalda biz OpenVPN-in status faylının necə oxunulması və istifadəsini göstərəcəyik:

### **İşə başlayaq**

Aşağıdakı şəbəkə quruluşunu istifadə edəcəyik:



Bu misalda da həmçinin başlığımızın əvvəlində generasiya etdiyimiz PKI açarlardan istifadə edəcəyik. Misalımızda Server və 1-ci client maşın FreeBSD 9.2 x64-də OpenVPN 2.3 ilə işləyir. Client2 maşın isə Windows7-dir. Server maşın üçün Server-side routing misalımızda olan quraşdırma faylı **basic-udp-server.conf**, FreeBSD client maşın üçün **basic-udp-client.conf** və həmçinin Windows7 maşın üçün isə öncəki misalımızda olan **basic-udp-client.ovpn** faylından istifadə edin.

### Necə edək

1. **basic-udp-server.conf** faylına sadəcə `'status /var/log/openvpn.status'` sətiri əlavə edin və **example2-8-server.conf** adı ilə yadda saxlayın (Fayl aşağıdakı kimi olacaq):

```
proto udp
port 1194
dev tun

server 192.168.200.0 255.255.255.0

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnserver.crt
key /usr/local/etc/openvpn/openvpnserver.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0

persist-key
persist-tun
keepalive 10 60
```

```
push "route 10.198.0.0 255.255.0.0"
topology subnet
```

```
user nobody
group nobody
```

```
daemon
log-append /var/log/openvpn.log
```

```
status /var/log/openvpn.status
```

2. Serveri işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example2-8-server.con
```

3. FreeBSD client işə salın:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config basic-udp-
client.conf
```

4. Qoşulma uğurla başa çatdıqdan sonra işə **openvpn.status** faylına baxaq:

```
root@siteA:/usr/local/etc/openvpn # cat /var/log/openvpn.status
OpenVPN CLIENT LIST
Updated,Tue Jan 21 09:45:34 2014
Common Name,Real Address,Bytes Received,Bytes Sent,Connected Since
openvpnclient1,2.2.2.10:14374,17331,19012,Tue Jan 21 09:19:20 2014
ROUTING TABLE
Virtual Address,Common Name,Real Address,Last Ref
192.168.200.2,openvpnclient1,2.2.2.10:14374,Tue Jan 21 09:19:44 2014
GLOBAL STATS
Max bcst/mcast queue length,0
END
```

5. **ca.crt**, **openvpnclient2.crt**, **openvpnclient2.key** ve **tls-auth** üçün secret key faylı **ta.key**-idə həmçinin **Windows7** maşına ya **WinSCP** ya da **Putty pscp** ilə transfer edin:

6. Windows client-i CLI-dan işə salın:

```
C:\>cd \Program files\openvpn\config
C:\Program Files\OpenVPN\config>..\bin\openvpn --config basic-udp-
client.ovpn
```

Yadda saxlayın ki, client key fayla generasiya edilən vaxt şifrə təyin edilmişdi və siz həmin şifrəni daxil etməlisiniz.

7. Yenidən serverin status faylına baxaq:

```
root@siteA:/usr/local/etc/openvpn # cat /var/log/openvpn.status
```

```
root@siteA:/usr/local/etc/openvpn # cat /var/log/openvpn.status
OpenVPN CLIENT LIST
Updated,Tue Jan 21 09:57:39 2014
Common Name,Real Address,Bytes Received,Bytes Sent,Connected Since
openvpnclient1,2.2.2.10:14374,21147,22828,Tue Jan 21 09:19:20 2014
openvpnclient2,3.3.3.10:57872,22889,12293,Tue Jan 21 09:52:09 2014
ROUTING TABLE
Virtual Address,Common Name,Real Address,Last Ref
192.168.200.2,openvpnclient1,2.2.2.10:14374,Tue Jan 21 09:19:44 2014
192.168.200.3,openvpnclient2,3.3.3.10:57872,Tue Jan 21 09:52:09 2014
GLOBAL STATS
Max bcst/mcast queue length,0
END
```



### **Bu necə işləyir**

Hər dəfə client OpenVPN serverə qoşulduqda, status faylı qoşulma informasiyaları ilə yenilənir. **OpenVPN CLIENT LIST** və **ROUTING TABLE** əsas istifadə edilən cədvəllərdəndir və aşağıda açıqlanır:

- Hansı clientlər qoşuludur
- Hansı IP ünvanlardan clientlər qoşulurlar
- Hər clientin göndərdiyi və qəbul etdiyi byte-ların rəqəmi
- Hər bir clientin qoşulu olduğu vaxt

Bundan başqa routing cədvəli hansı şəbəkənin hansı istifadəçiyə getdiyini də göstərir. Gördüyünüz kimi, qurduğumuz şəbəkədə **1.1.1.10 server**, **2.2.2.10 client1** və **3.3.3.10** isə **NAT serverdir** hansı ki, **client2**-ni NAT edir.

### **Daha da ətraflı**

#### **Status parametrləri**

Status direktivinin iki direktivi var:

- Status faylının adını təyin etmək
- Status faylının yenilənmə vaxtı aralığı. Susmaya görə 60 saniyədən bir yenilənir.

#### **Clientlərin disconnect edilməsi**

Clientlər disconnect olduqda status faylı həmin anda yenilənmir. OpenVPN ilk olaraq clientin serverdə olan **keepalive** parametrlərinə baxaraq yenidən qoşulmağa çalışır. Bu misalda server quraşdırma faylı aşağıdakı sətiri istifadə edir:

```
keepalive 10 60
```

Bu serverə deyir ki, clientə hər 10 saniyədən bir ping elə. Əgər o **60\*2** saniyədən sonra cavab almırsa, qoşulma qırılır və yenidən qoşulmağa çalışır. OpenVPN server həmişə ikinci mənanı 2-yə vurur. Həmçinin server clientə deyir ki, hər 10 saniyədən bir ping yolla və əgər cavab yoxdursa qoşulmanı 60 saniyədən sonra qır və yenidən qoşulmağa çalış.

#### **Explicit-exit-notify**

OpenVPN-in ən az tanınan direktivi isə aşağıdakıdır:

```
explicit-exit-notify [N]
```

Bu client tərəfdə təyin edilir ki, o qoşulmadan çıxanda açıq formada **OCC\_EXIT** mesajını serverə yollayır (əgər mümkündürsə). Bu qoşulması qırılmış istifadəçilərin silinməsini sürətləndirir. N mütləq olmayan parametrində isə mesajın necə dəfə göndərilməsi sayı təyin edilir. Susmaya görə ancaq **OCC\_EXIT** mesajı göndərilir hansı ki, problemlərə gətirir ona görə ki, UDP protokolu təminat vermir ki, o paket çatacaq.

## Management Interface

Bu misedə OpenVPN serverin idarəetmə interfeysindən server tərəfdə necə idarə edilməsi açıqlanır.

### Hazırlaşaq

Bu başlığımızın əvvəlində generasiya elədiyimiz PKI faylları yenidən istifadə edəcəyik. Bu misalımızda biz server tərəfdə **FreeBSD92 x64 və OpenVPN2.3** istifadə edəcəyik. **Windows7** Client isə **OpenVPN2.3**-dən istifadə edirik. Həmçinin server üçün quraşdırma faylı Server-side routing-də istifadə elədiyimiz **basic-udp-server.conf** olacaq. **Windows7** client üçün isə elə **'ifconfig-pool'** block-da istifadə elədiyimiz eyni **basic-udp-client.ovpn** quraşdırma faylından istifadə edəcəyik.

### İşə başlayaq

1. Susmaya görə olan server quraşdırma faylından istifadə edərək serveri işə salaq:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-server.conf
```

**basic-udp-server.conf** faylının tərkibi aşağıdakı kimi olacaq.

```
proto udp
port 1194
dev tun
server 192.168.200.0 255.255.255.0

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnserver.crt
key /usr/local/etc/openvpn/openvpnserver.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0

persist-key
persist-tun
keepalive 10 60

push "route 10.198.0.0 255.255.0.0"
topology subnet

user nobody
group nobody

daemon
log-append /var/log/openvpn.log
```

2. Windows7 client üçün elə **basic-udpclient.ovpn** quraşdırma faylını nüsxələyin **example2-9.ovpn** quraşdırma faylına və aşağıdakı sətiri sonuna əlavə edin:

```
management tunnel 23000 stdin
```

3. **ca.crt**, **openvpnclient2.crt**, **openvpnclient2.key** və **tls-auth** üçün **ta.key** faylını serverimizdən **Windows7** maşına təhlükəsiz kanal ilə (WinSCP yada Putty PSCP) küçürün:
4. Windows7 client maşını CLI ilə işə salın:  
C:\>**cd \Program files\openvpn\config**  
C:\Program Files\OpenVPN\config>..\bin\openvpn --config example2-9.ovpn  
  
OpenVPN client artıq management interfeys üçün şifrə istəyəcək (Yaxşı şifrə daxil edin). Bundan sonra private açarın şifresini daxil edin.
5. VPN qoşulması uğurlu olduqdan sonra isə biz öz serverimizdən OpenVPN client-in management interfeysinə **'telnet'** programı ilə qoşula bilərik (**Windows7** maşında VPN aldığı IP ünvanı CLI-dan **ipconfig /all | more** əmri ilə baxa bilərsiniz, bizim halda 192.168.200.2-dir ona görə ki, ilk client Windows7 maşınıdır):  
root@siteA:/usr/local/etc/openvpn # **telnet 192.168.200.2 23000**  
Trying 192.168.200.2...  
Connected to 192.168.200.2.  
Escape character is '^]'.  
ENTER PASSWORD:**freebsd**  
SUCCESS: password is correct  
>INFO:OpenVPN Management Interface Version 1 -- type 'help' for more info  
**status**  
OpenVPN STATISTICS  
Updated,Tue Jan 21 17:30:17 2014  
TUN/TAP read bytes,11532  
TUN/TAP write bytes,337  
TCP/UDP read bytes,11338  
TCP/UDP write bytes,23370  
Auth read bytes,385  
TAP-WIN32 driver status,"State=AT?c Err=[(null)/0] #O=2 Tx=[73,0]  
Rx=[6,0] IrpQ=[1,1,16] PktQ=[0,2,64] InjQ=[0,1,16]"  
END  
  
**signal SIGTERM**
6. **Ctrl+] -> quit** yada birbaşa **quit** əmrini daxil etməyiniz yetər ki, telnet programından çıxış edə bilərsiniz.

### **Bu necə işləyir**

OpenVPN client maşını serverə qoşulduqdan sonra avtomatik olaraq aşağıdakı direktivi istifadə edərək management interfeys işə düşür:

```
management tunnel 23000 stdin
```

Onun aşağıdakı parametrləri olur:

- tunnel - management interfeysi VPN tunelin özü ilə əlaqələndirir. Bu test məqsədilə və bəzi irəliləmiş clientlər üçün nəzərdə tutulur. Server tərəfdə management interfeysin üçün ən yaxşı seçim 127.0.0.1 IP ünvanıdır.

- 23000-cü port management interfyesin qulaq asacağı portdur.
- Son parametr isə şifrə faylı vəya OpenVPN işə düşdükdən sonra daxil ediləcək şifredir. Diqqət yetirin ki, bu şifrənin clientə aid olan private key faylının şifrəsi ilə heç bir əlaqəsi yoxdur.

Bundan sonra management interfeys işə dushur. Artıq siz server-dən **telnet** istifadə edərək clientə qoşula və müraciət göndərə bilərsiniz. Client həmçinin aşağıdakı əmri daxil edə bilər:

**signal SIGTERM**

Bu tamamilə client-in sessiyasını dayandırır və VPN-dən çıxarır. Bunu ona görə göstəririk ki, anlayasınız ki, management interfeys və onun şifrəsinin qorunması nə qədər önəmlidir.

### Daha da ətraflı

#### Server-side management interface

Management interfeys həmçinin OpenVPN serverin özündə də işə salına bilər. Bu halda isə bir əmrlə biz qoşulmuş clientlərin siyahısını əldə edə, disconnect edə və müəyyən inzibatçı işləri görə bilərik.

Düşünülür ki, management interfeys OpenVPN-in həm client və həm də server versiyalarında gələcəkdə daha vacib sayılacaq.

### Həmçinin baxın

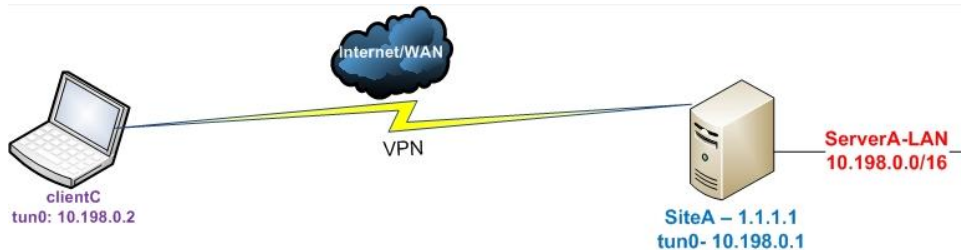
3-cü başlıqda misal yerinə yetirəcəksiniz hansı ki, Management interfeysi daha detallı şəkildə açıqlayır.

### Proxy-arp

Bu misalda biz sistem kernelinin **proxy-arp** imkanından istifadə edəcəyik ki, qoşulmuş VPN clientləri serverin Local şəbəkəsinin bir hissəsi kimi göstərək.

### İşə hazırlaşaq

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



Həmçinin başlığımızın əvvəlində yaratdığımız PKI açarlar burdada istifadə ediləcək. Serverimiz FreeBSD9.2 x64 OpenVPN2.3-də və clientimiz Windows7 OpenVPN2.3-də işləyəcək. Server üçün quraşdırma faylı **basic-udp-server.conf** istifadə ediləcək hansı ki, Server-side routing misalında yaratmışdıq. **Windows7** client quraşdırması üçün isə **ifconfig-pool block**-da işlətdiyimiz **basic-udp-client.ovpn**-i istifadə edəcəyik.

## Necə edək

1. **basic-udp-server.conf** faylını **example2-10-server.conf** adlı fayla nüsxələyin və içinə aşağıdakı tərkibi əlavə edin:

```
proto udp
port 1194
dev tun
server 10.198.0.0 255.255.0.0

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnsrvr.crt
key /usr/local/etc/openvpn/openvpnsrvr.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0

persist-key
persist-tun
keepalive 10 60

push "route 10.198.0.0 255.255.0.0"
topology subnet
user root
group wheel

daemon
log-append /var/log/openvpn.log

script-security 2
client-connect /usr/local/etc/openvpn/proxyarp-connect.sh
client-disconnect /usr/local/etc/openvpn/proxyarp-disconnect.sh
```

2. Server maşının göstərdiyiniz ünvanında **proxyarp-connect.sh** adlı script yaradın və tərkibinə aşağıdakı sətirləri əlavə edin:

```
#!/usr/local/bin/bash
/usr/sbin/arp -i em0 -Ds $ifconfig_pool_remote_ip em0 pub
```

Və eynilə göstərdiyiniz ünvanda **proxyarp-disconnect.sh** adlı script yaradıb tərkibinə aşağıdakı sətirləri əlavə edin:

```
#!/usr/local/bin/bash
/usr/sbin/arp -i em0 -d $ifconfig_pool_remote_ip
```

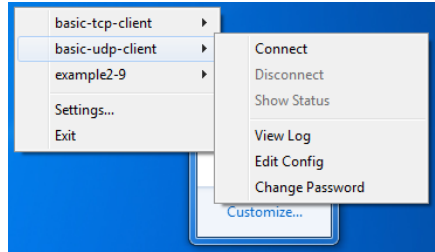
3. Əmin olun ki, scriptlər yerinə yetiriləndir:

```
root@siteA:/ # cd /usr/local/etc/openvpn/
root@siteA:/usr/local/etc/openvpn # chmod 755 proxyarp-*
```

4. Serveri **proxy-arp** rejimdə işləməsini aktivləşdiririk və openvpn-i işə salırıq:

```
root@siteA:/usr/local/etc/openvpn # sysctl
net.link.ether.inet.proxyall=1
root@siteA:/usr/local/etc/openvpn # openvpn --config example2-10-
server.conf
```

5. Windows7-də OpenVPN client GUI-ni açın və aşağıdakı kimi işə salın(Öncədən deyim ki, yazarın bu imkanı FreeBSD-də işləmədi):



Client uğurla qoşulduqdan sonra, OpenVPN serverində arp cədvəlində yeni yazı olacaq aşağıdakı kimi:

```
(10.198.0.2) at 00:0c:29:f2:8e:00 on em2 permanent published [ethernet]
```

Server tərəf LAN-da olan maşından artıq biz VPN client-ə ping yollaya bilərik:

```
[clientC]C:> ping 10.198.0.2
```

Qeyd edin ki, siteA lan tərəfdə heç bir spesifik routingə ehtiyac yoxdur. VPN client həqiqətəndə server LAN-i kimi olacaq.

### **Bu necə işləyir**

proxy-arp əksər UNIX və Linux OS-ların kernelləri tərəfindən dəstəklənən imkandır. Bu əksər hallarda Server tərəfin LAN-ına birbaşa yetki almaq üçün istifadə edilir və ADSL providerlərdə tez-tez istifadə edilir. OpenVPN client qoşulanda IP ünvanı SiteA aralığında olan LAN şəbəkədən alacaq. Eyni vaxt-da da server tərəf özünə aid olan MAC ünvanı client-ə təyin edilən IP üçün sərt yazı əlavə edəcək. Bu o deməkdir ki, SiteA tərəfdə olan müştəri 10.198.0.2 IP ünvanlı maşının harda olmasını öyrənmək istəyəndə ona cavab serverin özünə aid olan ARP cədvəlindən öz MAC ünvanı ilə veriləcək.

### **Daha ətraflı**

#### **User 'nobody'**

Qeyd edin ki, öncəki misalımızda biz nobody istifadəçi və qrup istifadə eləmədik ona görə ki, OS-umuz mac ünvanı sistemə əlavə eləmək üçün onun root yetkisi olmalıdır(Yada öncədən sistemə sudo yükləyib openvpn-ə arp əmrinə yetki verə bilərsiniz)

### **TAP-style şəbəkələri**

proxy-arp imkanları həmçinin elə TAP stilli şəbəkələrdə də istifadə edilə bilər. External DHCP server quruluşunda da həmçinin eyni nəticə əldə eləmək olur hansı ki, Ethernet-i bridge rejimdə işə salırıq.

### **Broadcast traffic həmişə işləməyə bilər**

proxy-arp istifadə edilən şəbəkə üzərindən broadcast ötürülməsi gizli baş verir. Əksər hallarda proxy-arp işləyir. Ancaq elə hallar olur ki, bütün clientləri bir broadcast domain-də olması tələb edilir. Bu halda isə Ethernet bridge ən yaxşı üsuldur.

Həmçinin 3-cü bölümdə broadcast və IP olmayan trafiklərdə yoxlanış edin.

## BÖLÜM 3

### Client-server Ethernet tipli şəbəkələr

Bu başlıqda biz aşağıdakıları açıqlayacağıq:

- Bridge olmayan şəbəkələrdə adi quraşdırma
- Client-to-client trafikinin aktivləşdirilməsi
- FreeBSD-də Bridge edilməsi
- Windows Bridge edilməsi
- IP olmayan və broadcast olan trafiklərin yoxlanılması
- Kənar DHCP serverin istifadə edilməsi
- Status faylının istifadə edilməsi
- Management interfeys

#### **Giriş**

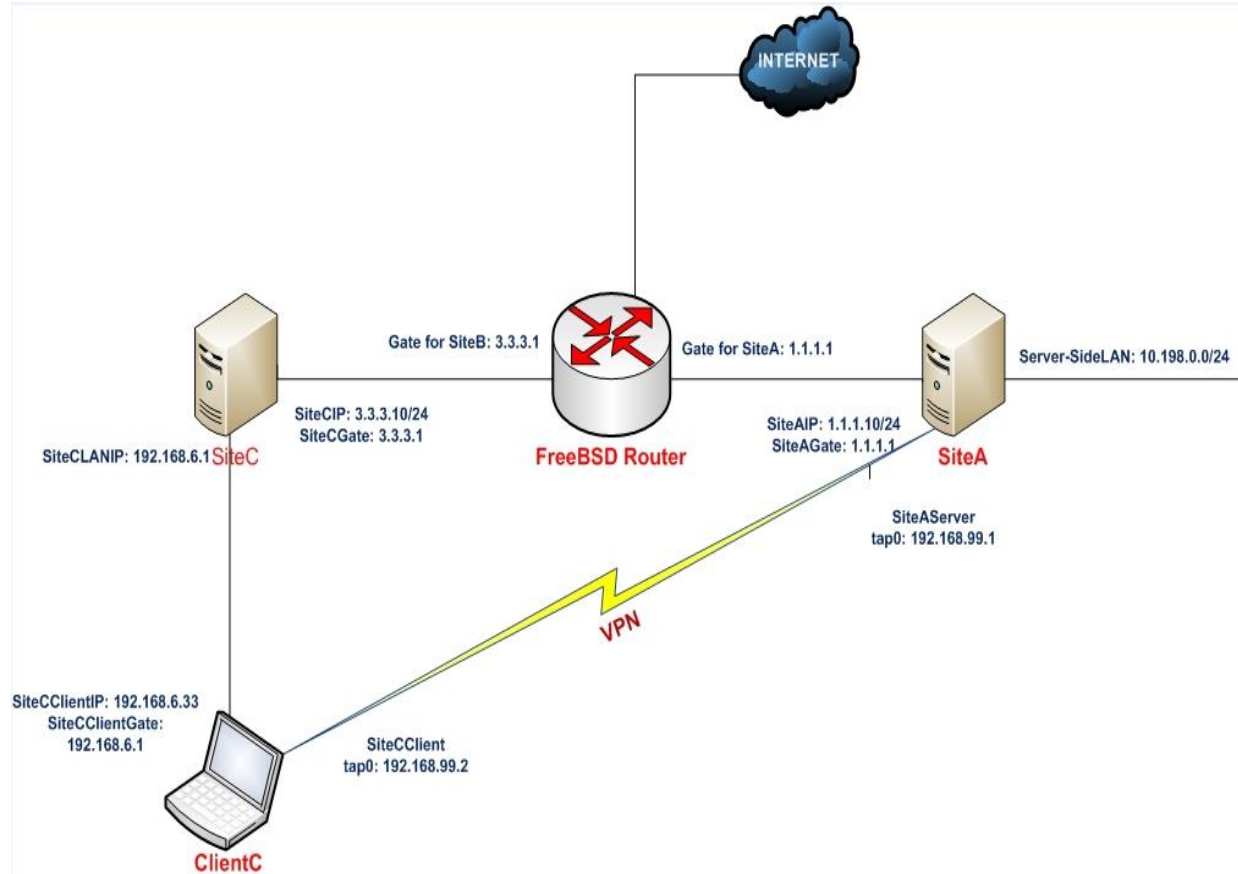
Bu başlıqda biz tək server və Ethernet trafikinin yönləndirilməsi imkanı olan çoxlu uzaq müştərilərlə işləyəcəyik. Biz çoxlu əsas quraşdırmalara baxacağıq. Həmçinin bridgləmə istifadə edəcəyik ki, kənar DHCP serveri istifadə edə bilək və həmçinin OpenVPN status faylının istifadə edilməsinə baxacağıq. Qeyd edin ki, bridge rejimi yalnız imkansız qalan halda son imkan kimi istifadə edilməlidir. Həmçinin qeyd edin ki, bridge rejiminin istifadə edilməsinin də öz çatışmazlıqları var hansı ki, davamiyyət və təhlükəsizlikdir.

## Bridge olmayan şəbəkələrdə adi quraşdırma

Bu başlıqda biz TAP tipli alətlərin istifadəsile client və server tərəfdə sertifikatların istifadə olunması ilə qoşulmaları göstərəcəyik. Burda həmçinin OpenVPN server arxasında olan maşınlara çatmaq üçün OpenVPN clientlərdə masquerading istifadə eləmək imkanı yaradılacaq. Bunun istifadəsinin üstünlüyü ondan ibarətdir ki, Server LAN tərəfdə heç bir spesifik routingə ehtiyac yoxdur. OpenVPN serverlər üçün masquerading rejimi yalnız Linux və UNIX maşınlarında mövcuddur. Bu elə öncə keçdiyimiz Server-side routing misalına oxşayır:

### İşə başlayaq:

Biz aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



2-ci başlıqda olan generasiya elədiyimiz Client-server sertifikatlarını yenidən istifadə edəcəyik (Yalnız IP şəbəkələrdə). Bu misalımızda Server maşın **FreeBSD9.2 x64 OpenVPN2.3** və client maşın isə **Windows7 OpenVPN2.3** maşında işləyəcək.

1. Server quraşdırma faylını yaradaq:

```
tls-server
proto udp
port 1194
dev tap
```

```
server 192.168.99.0 255.255.255.0
```



```
ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnsrver.crt
key /usr/local/etc/openvpn/openvpnsrver.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0
```

```
persist-key
persist-tun
keepalive 10 60
```

```
push "route 10.198.0.0 255.255.0.0"
user nobody
group nobody
daemon
log-append /var/log/openvpn.log
```

Faylı **example3-1-server.conf** adı ilə yadda saxlayın. Nəzərə alın ki, bəzi Linux distrolarda **nogroup** adlı qrup olur.

2. Serveri işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example3-1-server.conf
```

3. Sonra serverdə IP forwarding və firewall ilə masquerading rule-u yazın:

```
root@siteA:/usr/local/etc/openvpn # sysctl -w net.inet.ip.forwarding=1
```

Kernelinizi aşağıdakı sətirləri əlavə etdikdən sonra kompilyasiya edin ki, PF firewall işlədə biləsiniz.

```
device pf
device pflog
device pfsync
```

Ardınca da **/etc/rc.conf** faylına aşağıdakı sətirləri əlavə edin ki, PF startup-da işə düşsün.

```
pf_enable="YES"
pf_rules="/etc/pf.conf"
pflog_enable="YES"
pflog_logfile="/var/log/pflog"
```

PF üçün **/etc/pf.conf** faylında aşağıdakı sətirləri əlavə edək və işə salaq.

```
ext_if="em0"
ext_ip="1.1.1.10"
vpn_if="tap0"
table <clientler> { 192.168.99.0/24 }

rdr on $vpn_if from 192.168.99.0/24 to any -> $ext_if

pass in quick all
pass out quick all
```

4. Sonra işə client quraşdırma faylını yaradaq:

```

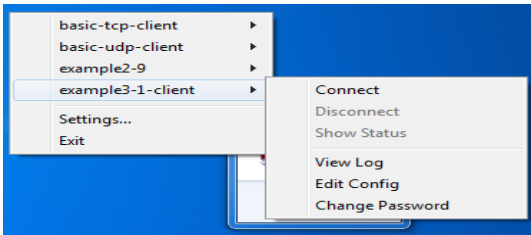
client
proto udp
remote openvpnserver.example.com
port 1194
dev tap
nobind

ca /etc/openvpn/itvpn/ca.crt
cert /etc/openvpn/itvpn/client1.crt
key /etc/openvpn/itvpn/client1.key
tls-auth /etc/openvpn/itvpn/ta.key 1
ns-cert-type server

```

Faylı **example3-1-client.conf** adından Windows7 maşınının

5. Sonra Client-i işə salaq:



```

Wed Jan 29 22:40:47 2014 OpenVPN 2.3.2 x86_64-w64-mingw32 [SSL (OpenSSL)] [LZO] [PKCS11] [eurephia] [IPv6] built on Aug 22 2013
Wed Jan 29 22:40:50 2014 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
Wed Jan 29 22:40:50 2014 Control Channel Authentication: using 'c:/program files/openvpn/config/ta.key' as a OpenVPN static key file
Wed Jan 29 22:40:50 2014 UDPv4 link local: [undef]
Wed Jan 29 22:40:50 2014 UDPv4 link remote: [AF_INET]1.1.1.10:1194
Wed Jan 29 22:40:50 2014 [openvpnserver] Peer Connection Initiated with [AF_INET]1.1.1.10:1194
Wed Jan 29 22:40:52 2014 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0
Wed Jan 29 22:40:52 2014 open_tun, tt->ipv6=0
Wed Jan 29 22:40:52 2014 TAP-WIN32 device [Local Area Connection 2] opened: \\.\Global\{B835D57D-F453-48B3-A987-077A7A6E65DC}.tap
Wed Jan 29 22:40:52 2014 Notified TAP-Windows driver to set a DHCP IP/netmask of 192.168.99.2/255.255.255.0 on interface {B835D57D-F453-48B3-A987-077A7A6E65DC} [DHCP-s
Wed Jan 29 22:40:52 2014 Successful ARP Flush on interface [17] {B835D57D-F453-48B3-A987-077A7A6E65DC}
Wed Jan 29 22:40:57 2014 Initialization Sequence Completed

```

6. Qoşulma bitdikdən sonra isə biz ping ilə hər şeyin işlənməsini yoxlaya bilərik:

```

C:\Users\ClientC>ping -n 2 192.168.99.1

Pinging 192.168.99.1 with 32 bytes of data:
Reply from 192.168.99.1: bytes=32 time=2ms TTL=64
Reply from 192.168.99.1: bytes=32 time=2ms TTL=64

Ping statistics for 192.168.99.1:
 Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
 Approximate round trip times in milli-seconds:
 Minimum = 2ms, Maximum = 2ms, Average = 2ms

C:\Users\ClientC>ping -n 2 10.198.0.1

Pinging 10.198.0.1 with 32 bytes of data:
Reply from 10.198.0.1: bytes=32 time=1ms TTL=64
Reply from 10.198.0.1: bytes=32 time=2ms TTL=64

Ping statistics for 10.198.0.1:
 Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
 Approximate round trip times in milli-seconds:
 Minimum = 1ms, Maximum = 2ms, Average = 1ms

```

### Bu necə işləyir

Server işə düşən kimi o ilk IP ünvanını TAP adlı virtual alətinə mənimsədir. Bundan sonra isə, server UDP 1194-cü portda qulaq asmağa başlayır və gələn qoşulmaları gözləyir.

Client serverə bu port ilə qoşulur. Həm client və həm də server sertifikatlarını istifadə edərək TLS əl sıxışması yerinə yetirildikdən sonra,

client 192.168.99.2 IP ünvanını özünə mənimsədir. Client ilk TAP alətinin öz quraşdırmasını oxuyur və VPN uğurla qoşulduqdan sonra uyğun olan IP ünvana mənimsədir.

OpenVPN quraşdırmasının hissəsi olaraq, bu açıqlamada PF firewall istifadə edildi və şərait yaradıldı ki, Serverin LAN tərəfinə heç bir route yazılmadan bütün şəbəkənin görülmə imkanı yaradıldı. Aşağıdakı əmr UNIX kernelə deyir ki, **tun0** şəbəkə kartından və **192.168.99.0/24** subnetindən gələn bütün şəbəkə yönləndirilir **em0** şəbəkə kartının üstünə.

```
rdr on $vpn_if from 192.168.99.0/24 to any -> $ext_if
```

Bu paketlərin hər birinin mənbə ünvanı var hansı ki, əslində ora elə yazılır ki, güya OpenVPN client-dən yox OpenVPN serverin özündən gəlir. PF bu paketləri izləyir ki, geri qayıdanda da düzgün ünvana qayıtsın. Ancaq burda bir çatışmamazlıq var ki, clientin sayı çox olduqda Server tərəfin LAN-ından gələn trafik serverin özündən gələn trafikin client1-indən və ya VPN tunneldə olan clientN-dən gələnlə ayırmaq olmur.

### **TUN və TAP arasında olan fərqlər**

Bu misal ilə öncəki çəkdiyimiz Server-side routing arasındakı misal çox azdır. Çox kiçik fərqlər var ancaq, siz onları diqqətə almasanız problemləriniz çıxa bilər. Gəlin onları açıqlayaq:

- TAP adapter istifadə edərkən tam Ethernet frame encapsulyasiya edilir. Buna gözlədiyimizdən çox resurs gədir.
- TAP tipli şəbəkəyə qoşulmuş bütün maşınlar tək broadcast domain-də olurlar. Bunun açıqlanması növbəti misalımızda göstəriləcək.
- Əgər bridging etmək tələb edilirsə, TAP stilli tunel tələb edilir

### **TCP protocol-un istifadə edilməsi**

Bu misalda biz UDP protocol seçdik. TCP seçilməsi üçün sadəcə client və server tərəfdə aşağıdakı sətiri:

```
proto udp
```

Dəyişməlisiniz:

```
proto tcp
```

UDP protocol adi halda performansını artırır ancaq, bəzi router və firewall-ların UDP trafikin yönləndirilməsində çox ciddi problemləri olur. Bu hallara görə TCP protocol-u daha çox istifadə edilir.

### **IP yönləndirilmənin startup-a əlavə edilməsi**

FreeBSD OS üzərindən aşağıdakı sətirlə System Control-dan:

```
echo "net.inet.ip.forwarding=1" >> /etc/sysctl.conf
```

Ya da startup script ilə **/etc/rc.conf** faylına aşağıdakı sətiri əlavə etməklə edə bilərsiniz:

```
gateway_enable="YES"
```

### **Həmçinin baxın**

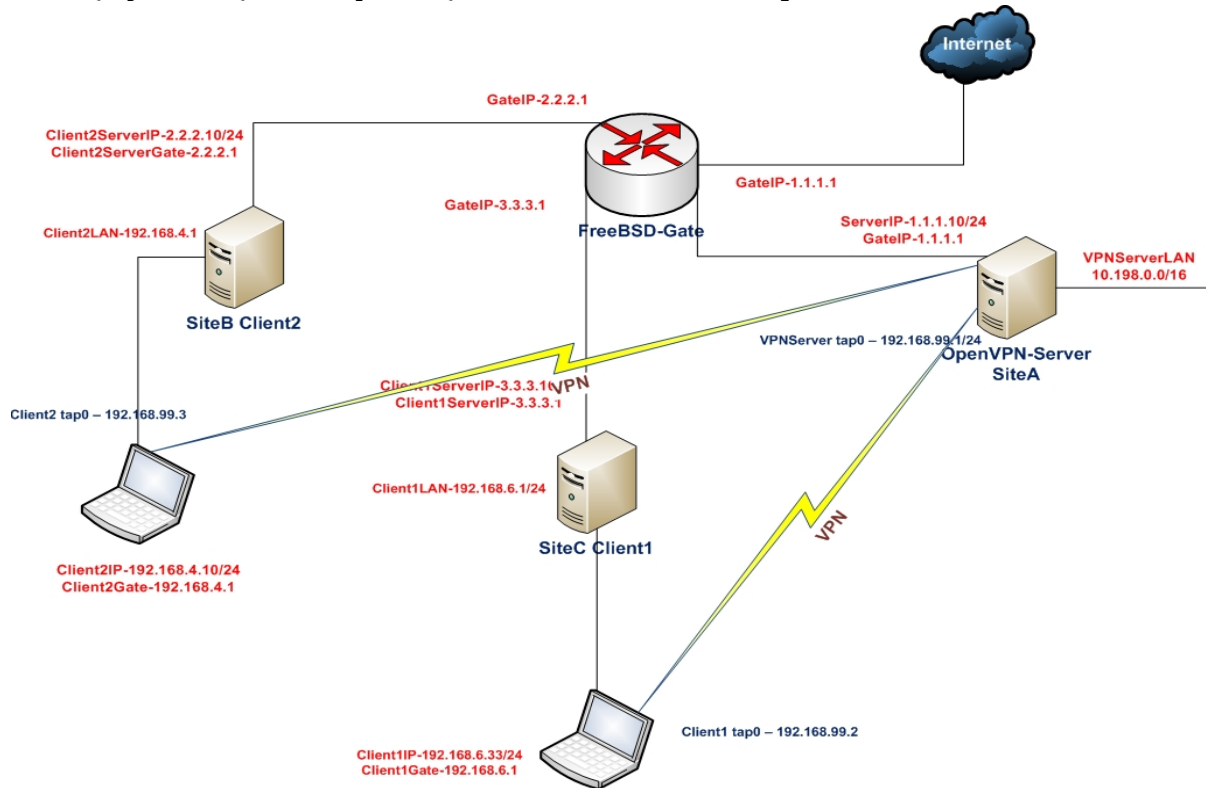
- 2-ci başlıqda Server-side routing-ə hansı ki, əsas TUN stilli qoşulma açıqlanır.

## Client-to-client trafikinin aktivləşdirilməsi

Bu misal öncəkinin davamıdır. Burda TAP alətlərinin sertifikatların istifadəsilə server və client rejimində istifadəsi açıqlanacaq. Client-to-client direktivinin istifadəsi sayəsində imkan yaranır ki, müxtəlif VPN clientlər bir-birlərinə qoşula bilsinlər. TAP tipli qoşulmalarda müəyyən bir çatışmamazlıqlar var.

### İşə hazırlaşaq

Biz aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



2-ci başlıqda yaratdığımız client və server sertifikatlarını istifadə edəcəyik. Bu misalımızda Server FreeBSD9.2 x64 OpenVPN2.3 və 2 Client maşını olacaq. Client-in hər biri Windows7 OpenVPN2.3 olacaq. Server üçün yenidən **example3-1-server.conf** faylını istifadə edəcəyik. Client kimi istifadə etdiyimiz maşınlarda **ClientC** nəzərdə tutulur ki, **client1**-dir və **ClientB** isə nəzərdə tutulur ki, **Client2**-dir.

### Necə edək

1. **example3-1-server.conf** faylına aşağıdakı sətiri əlavə edərək server quraşdırma faylını yaradın:  
**client-to-client**

Əlavə etdikdən sonra faylı **example3-2-server.conf** adında yadda saxlayın.

2. Serveri işə salın:  

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example3-2-server.conf
```
3. PF ilə IP yönləndirməni edin:  

```
root@siteA:/usr/local/etc/openvpn # sysctl -w net.inet.ip.forwarding=1
```

**/etc/pf.conf** faylına aşağıdakı sətiri əlavə edib işə salın(**vpn\_if** dəyişəni **tap0** alətidir.).  
**rdr on \$vpn\_if from 192.168.99.0/24 to any -> \$ext\_if**

4. Sonra işə ilk client quraşdırmasını yaradaq:

```
client
proto udp
remote openvpnserver.example.com
port 1194

dev tap
nobind
auth-nocache

ca "c:/Program files/openvpn/config/ca.crt"
cert "c:/Program files/openvpn/config/openvpnclient1.crt"
key "c:/Program files/openvpn/config/openvpnclient1.key"
tls-auth "c:/Program files/openvpn/config/ta.key" 1

ns-cert-type server

verb 5
```

Faylı **example3-2-client1.ovpn** adı ilə yadda saxlayın.

5. Uyğun olaraq ikinci client üçün də quraşdırma faylını yaradın (Uyğun olaraq 2-ci client-də **c:\windows\system32\drivers\etc\hosts** faylına **1.1.1.10 openvpnserver.example.com** sətirini əlavə etməyi unutmayın):

```
client
proto udp
remote openvpnserver.example.com
port 1194

dev tap
nobind
auth-nocache

ca "c:/program files/openvpn/config/ca.crt"
cert "c:/program files/openvpn/config/openvpnclient2.crt"
key "c:/program files/openvpn/config/openvpnclient2.key"
tls-auth "c:/program files/openvpn/config/ta.key" 1

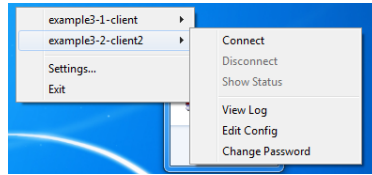
ns-cert-type server

verb 5
```

Həmçinin **example3-2-client2.ovpn** adı ilə yadda saxlayın.

6. Client1-i windows CLI-dan və 2-ci client-i OpenVPN GUI-dən işə salın.  
 C:\Users\ClientC>**cd \Program files\openvpn\config**  
 C:\Program Files\OpenVPN\config>..\bin\openvpn --config example3-2-client1.ovpn

Client2-də işə OpenVPN GUI ilə qoşulaq.



7. Qoşulma uğurla olduqdan sonra işə biz hər şeyin işləməsini ping ilə yoxlaya bilərik(Client1-dən həm server və həm də clien2-ni ping ilə yoxlayaq):

```
C:\Users\ClientC>ping -n 2 192.168.99.1
Pinging 192.168.99.1 with 32 bytes of data:
Reply from 192.168.99.1: bytes=32 time=3ms TTL=64
Reply from 192.168.99.1: bytes=32 time=3ms TTL=64
```

```
C:\Users\ClientC>ping -n 2 192.168.99.3
Pinging 192.168.99.3 with 32 bytes of data:
Reply from 192.168.99.3: bytes=32 time=8ms TTL=128
Reply from 192.168.99.3: bytes=32 time=3ms TTL=128
```

8. Sonda da server Lan tərəfin ping edilməsini yoxlayaq.

```
C:\Users\ClientC>ping -n 2 10.198.0.10
Pinging 10.198.0.10 with 32 bytes of data:
Reply from 10.198.0.10: bytes=32 time=3ms TTL=64
Reply from 10.198.0.10: bytes=32 time=1ms TTL=64
```

### **Bu necə işləyir**

Hər iki client adi qaydada serverə qoşulur. Aşağıdakı direktiv sayəsində bütün client-lər bir-birlərini görürlər.

**client-to-client**

Clientlər arasında qoşulma OpenVPN server üzərindən keçəcək hansı ki, ICMP paketlərin içində görmək olar. ICMP(ping) echo və reply axınını aşağıdakı kimi açıqlamaq olar:

1. OpenVPN client paketlərini şifrələyir və serverin üzərinə təhlükəsiz yolla yönləndirir.
2. Server paketləri deşifrə edir və təyin edir ki, gələn paket digər müştəri üçün nəzərdə tutulur. Beləliklə paket kernelin rutinginə ötürülmür və yenidən şifrələnərək ikinci clientə ötürülür.
3. 2-ci client paketi əldə edir, deşifrə edir və yenidən təhlükəsiz kanal ilə serverə qaytarır.

4. Server yenidən paketi deşifrə edir və təyin edir ki, paket ilk clientə çatdırılmalıdır. Həmçinin burda da paket kernel routingə yönləndirilmədən şifrlənir və yenidən original clientə qaytarılır.

### **Daha da ətraflı**

#### **Broadcast trafik geniş yayım effecti gətirə bilər.**

Bütün maşınlar bir broadcast domain ilə TAP stilli şəbəkə ilə qoşulurlar. client-to-client aktiv olanda bu o deməkdir ki, clientlərdən gələn bütün broadcast domain trafiki yönləndirilir digər clientlərə. client2-də işləyən wireshark çoxlu paketlər göstərir hansı ki, client1-dən gəlir hansı ki, hamısı OpenVPN server üzərindən gəlir. Bu çoxlu client sayı olanda problemlərə gətirib çıxara bilər.

#### **Trafikin filter edilməsi**

OpenVPN-nin hal-hazırkı versiyasında client-to-client qoşulmasında VPN clientlər arasında trafiki filter eləmək mümkün deyil. OpenVPN-nin gələcək versiyalarında bunu eləmək mümkün olacaq. Həmçinin mümkündür ki, client-dən clientdə qoşulmanı client-to-client directivi olmadan edə bilərsiniz ancaq bunu Firewall ruleları sayəsində edə bilərsiniz. Üstünlüyü ondan ibarətdir ki, siz özünüz lazım olan client trafikini filter edə biləcəksiniz. Çatışmamazlığı odur ki, bu daha az effektivdir.

#### **TUN stilli şəbəkələr**

client-to-client direktivi həmçinin TUN stilli şəbəkələrdə istifadə edilə bilər. Bu həmçinin öncəki misalımıza uyğun işləyir ancaq, clientlər tək broadcast domaində olurlar.

#### **FreeBSD-də Bridge edilməsi**

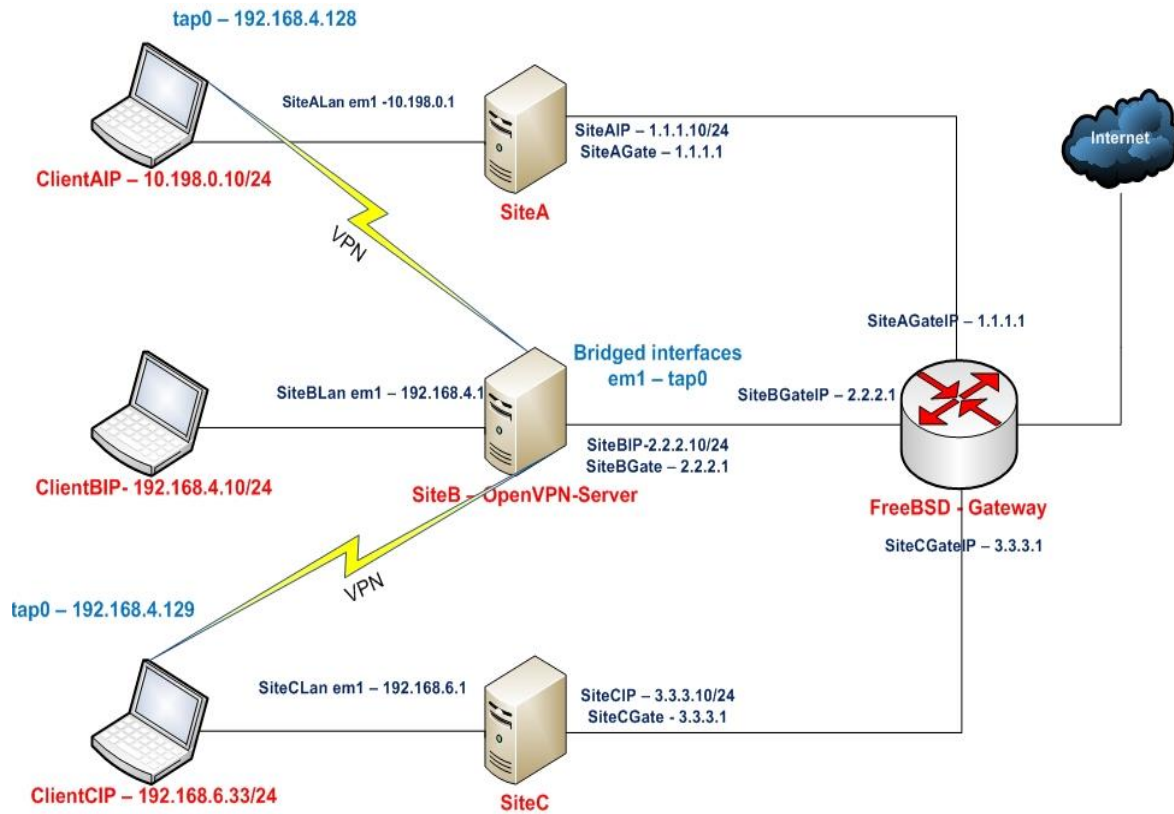
Bu başlıqda OpenVPN serverin necə bridge edilməsini açıqlayacağıq. Biz local şəbəkə və VPN şəbəkəsini bridge(körpü) edəcəyik. Yeni ki, bir şəbəkədən gələn bütün trafik digər şəbəkə üzərinə və geriye yönləndiriləcək.

Bu imkan adətən istifadə edilir ki, Windows bazalı olan şəbəkələrdə təhlükəsiz yolla uzaq istifadəçilərə qoşulmağa imkan yaradır ancaq, bunun düzgün qurulması çox çətinidir. Demək olar ki, əksər hallarda TUN tipli alətləri WINS server ilə birlikdə clientlər üçün və server tərəfdə TUN tipli alət istifadə edilməsi kifayət edir. VPN bridge edilmənin öz üstünlükləri var və növbəti başlıqda biz onları açıqlayacağıq.

Həmçinin bridge edilmənin istifadəsinin öz çatışmamazlıqları var hansı ki, serverin gücünü alır: 100 megabitlik adi Ethernet adapterin tab gətirmə qabiliyyəti bridge rejimdə demək olar ki, yarıya bərabər olur.

#### **İşə başlayaq**

Biz aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



İkinci başlıqda əldə etdiyimiz sertifikatları istifadə edəcəyik. Client-server yalnız IP şəbəkələrində işləyir. Bu başlıqda VPN server FreeBSD9.2 x64 OpenVPN2.3 versiyası ilə, clientA isə Windows7 OpenVPN2.3 və clientC Windows7 OpenVPN2.3 ilə işləyəcək. Windows7 clientlər üçün elə həmin **example3-2-client1.ovpn** və **example3-2-client2.ovpn** quraşdırma fayllarından istifadə edəcəyik.

### Necə edək

1. Server quraşdırma faylını yaradaq:

```

proto udp
port 1194
dev tap0 ## '0' olması çox önəmlidir

server-bridge 192.168.4.65 255.255.255.0 192.168.4.128 192.168.4.200
push "route 192.168.4.0 255.255.255.0"

script-security 2
client-connect "/usr/local/etc/openvpn/up-bridge.sh"
client-disconnect "/usr/local/etc/openvpn/down-bridge.sh"

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnserver.crt
key /usr/local/etc/openvpn/openvpnserver.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0

persist-key

```



```
persist-tun
keepalive 10 60
```

```
user root
group wheel
```

```
daemon
log-append /var/log/openvpn.log
```

Faylı **example3-3-server.conf** adında yadda saxlayın.

2. Bridge olan interfeysin avtomatik qalxması və avtomatik bağlanması üçün aşağıdakı scripdlərimizi yaradırıq.

**/usr/local/etc/openvpn/up-bridge.sh** - Tərkibinə aşağıdakı sətirləri əlavə edirik.

```
#!/bin/sh
/sbin/ifconfig bridge0 addm ${dev}
/sbin/ifconfig ${dev} up
```

```
exit 0
```

**/usr/local/etc/openvpn/down-bridge.sh** - Həmçinin tərkibinə aşağıdakı sətirləri əlavə edirik.

```
#!/bin/sh
/sbin/ifconfig bridge0 deletem ${dev}
```

```
exit 0
```

Hər iki scriptə yerinə yetirilmə yetkisi veririk.

```
chmod +x /usr/local/etc/openvpn/up-bridge.sh
/usr/local/etc/openvpn/down-bridge.sh
```

Sonra serverin kernelini aşağıdakı alətlərlə kompilyasiya edib yükləyin.

```
device tap
device if_bridge
```

3. Eynilə serverimizin startup quraşdırma faylına **/etc/rc.conf**-a aşağıdakı sətirləri əlavə edək ki, həm bridge kartımız və həm də VPN-imiz avtomatik işə düşsün.

```
ifconfig_em0="inet 2.2.2.10 netmask 255.255.255.0"
defaultrouter="2.2.2.1"
ifconfig_em1="inet 192.168.4.1 netmask 255.255.255.0"
hostname="siteB"
gateway_enable="YES"
firewall_enable="YES"
firewall_type="OPEN"
natd_enable="YES"
natd_interface="em0"
```

```
cloned_interfaces="bridge0"
autobridge_interfaces="bridge0"
```

```

autobridge_bridge0="em1"
ifconfig_bridge0="inet 192.168.4.65 netmask 255.255.255.0 up"

openvpn_enable="YES"
openvpn_if="tap bridge"
openvpn_configfile="/usr/local/etc/openvpn/example3-3-
server.conf"
openvpn_dir="/usr/local/etc/openvpn"

```

4. Sonra isə OpenVPN serverimizi işə salırıq.  
 root@siteB:/usr/local/etc/openvpn # /usr/local/etc/rc.d/openvpn start

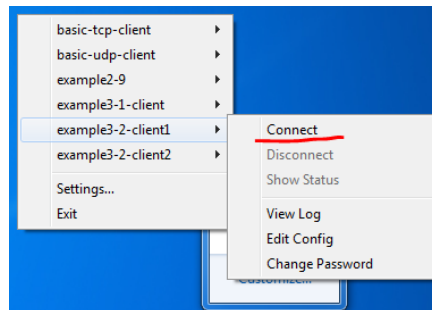
5. Hər iki clientimizdə **c:\windows\system32\drivers\etc\hosts** faylına lazımi verilənləri əlavə edək və ardınca işə salaş.

```

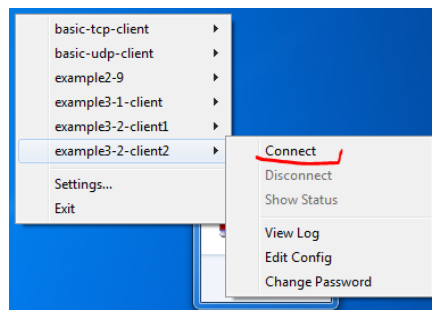
127.0.0.1 localhost
#1.1.1.10 openvpnserver.example.com
2.2.2.10 openvpnserver.example.com
#3.3.3.10 openvpnserver.example.com

```

**ClientA**



**ClientC**



6. Clientlərimizin VPN-dən aldığı IP ünvanlara baxaq.

```

C:\Users\ClientA>ipconfig /all|more
Ethernet adapter Local Area Connection 2:
 IPv4 Address. : 192.168.4.128 (Preferred)
 Subnet Mask : 255.255.255.0
 Lease Obtained. : Friday, January 31, 2014 9:32:00 PM
 Lease Expires : Saturday, January 31, 2015 9:31:59 PM
 Default Gateway :
 DHCP Server : 192.168.4.0

```

```

C:\Users\ClientC>ipconfig /all|more
Ethernet adapter Local Area Connection 2:
 IPv4 Address. : 192.168.4.129 (Preferred)
 Subnet Mask : 255.255.255.0

```

```
Lease Obtained. : Friday, January 31, 2014 9:33:02 PM
Lease Expires : Saturday, January 31, 2015 9:33:02 PM
Default Gateway :
DHCP Server : 192.168.4.0
```

7. Sonra Client-lərin birindən Serverin LAN-ında olan bir IP ünvanına ping yollayırıq.

```
C:\Users\ClientC>ping -n 2 192.168.4.10
Pinging 192.168.4.10 with 32 bytes of data:
Reply from 192.168.4.10: bytes=32 time=5ms TTL=128
Reply from 192.168.4.10: bytes=32 time=4ms TTL=128
```

### **Bu necə işləyir**

**up-bridge.sh** scripti iki şəbəkə kartı arasında yeni, LAN tərəfdə olan şəbəkə kartı və virtual yaranan tap aləti arasında körpü yaradır. Bridge yaratmağın əsas özəlliyi odur ki, bütün şəbəkə axını bir şəbəkə kartından digərinə nüsxələnir və geriyyə qayıdır. Bu bize client-lərin aldığı IP ünvan aralığı ilə server tərəfdə olan LAN subnetlə eyni olan halda köməklik göstərir.

Bridge interfeysin çatışmamazlığı ondan ibarətdir ki, OpenVPN serverdə dayanıqlıq aşağı düşür və resurslar həddən artıq çox istifadə edilir. Əgər client-lər tərəfdən çoxlu broadcast axın gəlsə körpü tam dola bilər.

### **Daha da ətraflı**

#### **Fixed addresses və default gateway**

Bu başlıqda OpenVPN serverə Serverin LAN tərəfindən seçilmiş və öncədən təyin edilmiş IP ünvan verilmişdi hansı ki, bridge interfeyslərdə əksər hallarda belə edilir. Şəbəkə Bridge-nə dinamik IP ünvan təyin edilməsinin çətinliyi ondan ibarətdir ki, təyin edilmiş dinamik aralıq yox digərində IP ünvan seçilə bilər. Bu həmçinin bizə şərait yaradır ki, serverin quraşdırma faylında öncədən bridge interfeys üçün təyin etdiyimiz IP ünvanı yazmağa bilək.

Bridge istifadə elədikdə həmçinin önəmlidir baxasınız ki, bridge interfeys qalxdıqdan sonra default route yazılmış olsun (**192.168.4.65 link#8 UHS 0 0 100**).

#### **Adın resolve edilməsi**

Bridge interfeysin istifadə edilməsində ən çətin yerlərdən biri isə adın resolve edilməsidir. OpenVPN ancaq Layer2 yada IP bazalı routing imkanı yaradır. Düzdü adın resolve edilməsi üçün şəbəkəndə olan DNS server (DC yada Wins server) həmçinin çox çətinliklə bridge edilə bilər.

#### **Həmçinin baxılmalı**

Bu başlığın növbəti misalında Windows maşının necə bridge edilməsi açıqlanır.

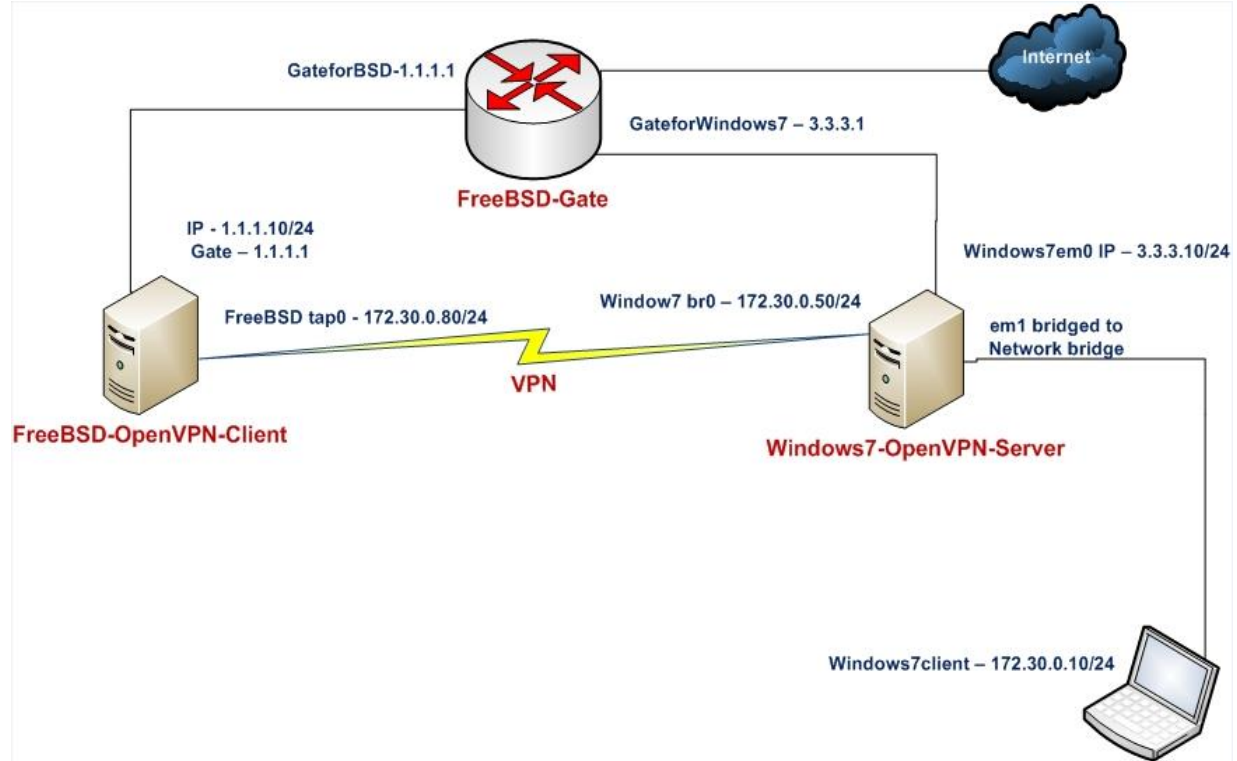
## Windows bridge edilməsi

Bu başlıqda isə Windows maşında bridge edilmiş OpenVPN server haqqında danışacağıq. Windows maşında bridge etmək UNIX və Linux-a baxdıqda çox fərqlidir ancaq concept eynidir.

Bu misal öncəkinə oxşayır ancaq, bridge edilmənin fərqli metodikası istifadə edilir.

## İşə başlayaq

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik.



Eynilə 2-ci başlıqda (**Client-server IP şəbəkələri**) yaratdığımız client/server sertifikatlarını burdada istifadə edəcəyik.

Bu misalda serverimiz Windows7 OpenVPN2.3 maşınında və client isə FreeBSD9.2 x64 OpenVPN2.3-də olacaq. FreeBSD maşın üçün isə **example3-1-client.conf** quraşdırma faylından istifadə edəcəyik.

## İşə başlayaq

1. Server quraşdırma faylını yaradaq:

```
proto udp
port 1194
dev tap
dev-node tap-bridge

server-bridge 172.30.0.50 255.255.255.0 172.30.0.80 172.30.0.250
```

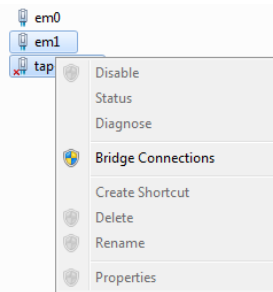
```
ca "c:/Program files/openvpn/config/ca.crt"
cert "c:/Program files/openvpn/config/openvpnserver.crt"
key "c:/Program files/openvpn/config/openvpnserver.key"
dh "c:/Program files/openvpn/config/dh2048.pem"
tls-auth "c:/Program files/openvpn/config/ta.key" 0

push "route 172.30.0.0 255.255.255.0"

persist-key
persist-tun
keepalive 10 60
```

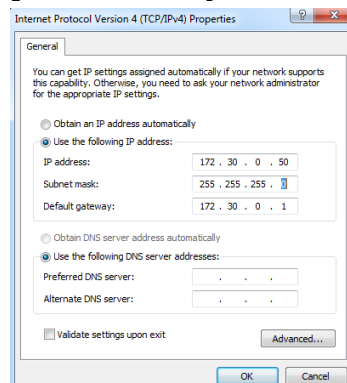
Faylı **example3-4-server.ovpn** adı ilə yadda saxlayın.

2. Sonra isə şəbəkə bridge-ni yaradaq:
  - o İstənilən **TAP-Windows Adapter V9** kartı sistemdə **Local Area Connection2** kimi qeydə alınır. Control Panel-də olan **Network Connections**-a gedin və onun adını dəyişib **tap-bridge** edin.
  - o Sonra isə **tap-bridge** kartını və LAN kartınızı (Yəni **em1**) **Ctrl** düyməsini sıxaraq seçib və seçilən iki kartın üstündə sağ düyməni sıxın. Açılan pəncərədə isə **Bridge Connections** düyməsini sıxın ki, körpü yaransın.



Bu **Network Bridge** adlı körpü kart yaradacaq.

3. Artıq yaranan Bridge kartı aşağıdakı IP ünvanla quraşdıraraq.



4. CLI-dan dəqiq baxaq ki, bridge kartımız düzgün quraşdırılıb:

```
C:\Users\ClientD>netsh interface ip show address "Network Bridge"
Configuration for interface "Network Bridge"
DHCP enabled: No
IP Address: 172.30.0.50
Subnet Prefix: 172.30.0.0/24 (mask 255.255.255.0)
```

```
Default Gateway: 172.30.0.1
Gateway Metric: 256
InterfaceMetric: 10
```

5. OpenVPN serveri işə salın:

```
C:\Program Files\OpenVPN\config>cd \Program files\openvpn\config
C:\Program Files\OpenVPN\config>..\bin\openvpn --config example3-4-
server.ovpn
```

6. Sonra clienti işə salın(FreeBSD client maşında /etc/hosts faylına **3.3.3.10 openvpnsrver.example.com** sətirini əlavə etməyi unutmayın):

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example3-1-
client.conf
```

7. Artıq VPN clientin aldığı IP ünvanı və uzaq server tərəfdə olan LAN IP-ləridən birini ping ilə yoxlayaq:

```
root@siteA:/usr/local/etc/openvpn # ifconfig tap0
tap0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
options=80000<LINKSTATE>
ether 00:bd:3d:ea:03:00
inet 172.30.0.80 netmask 0xfffff00 broadcast 172.30.0.255
media: Ethernet autoselect
status: active
Opened by PID 1856
```

```
root@siteA:/usr/local/etc/openvpn # ping -c 2 172.30.0.10
PING 172.30.0.10 (172.30.0.10): 56 data bytes
64 bytes from 172.30.0.10: icmp_seq=0 ttl=128 time=3.785 ms
64 bytes from 172.30.0.10: icmp_seq=1 ttl=128 time=4.342 ms
```

### Bu necə işləyir

Əsas yadda saxlanası odur ki, bu misal öncəki ilə eynidir. Sadəcə Windows maşınlarında bridge adapter aşağıdakı sətirlərlə seçilir.

```
dev tap
dev-node tap-bridge
```

UNIX və Linux maşınlarında isə sadəcə 1 sətirlə yeni aşağıdakı kimi təyin edilir:

```
dev tap0
```

Ancaq **Windows** maşınlarında **TAP** adapter üçün ad fərqli olur. Məhz bunun qarşısının alınması üçün **dev-node** istifadə edilir.

### Həmçinin baxaq

Öncəki misala baxın ki, UNIX maşında bridge edilmənin üsulunu öyrəne bilərsiniz.

## IP olmayan və Broadcast olan axının yoxlanılması

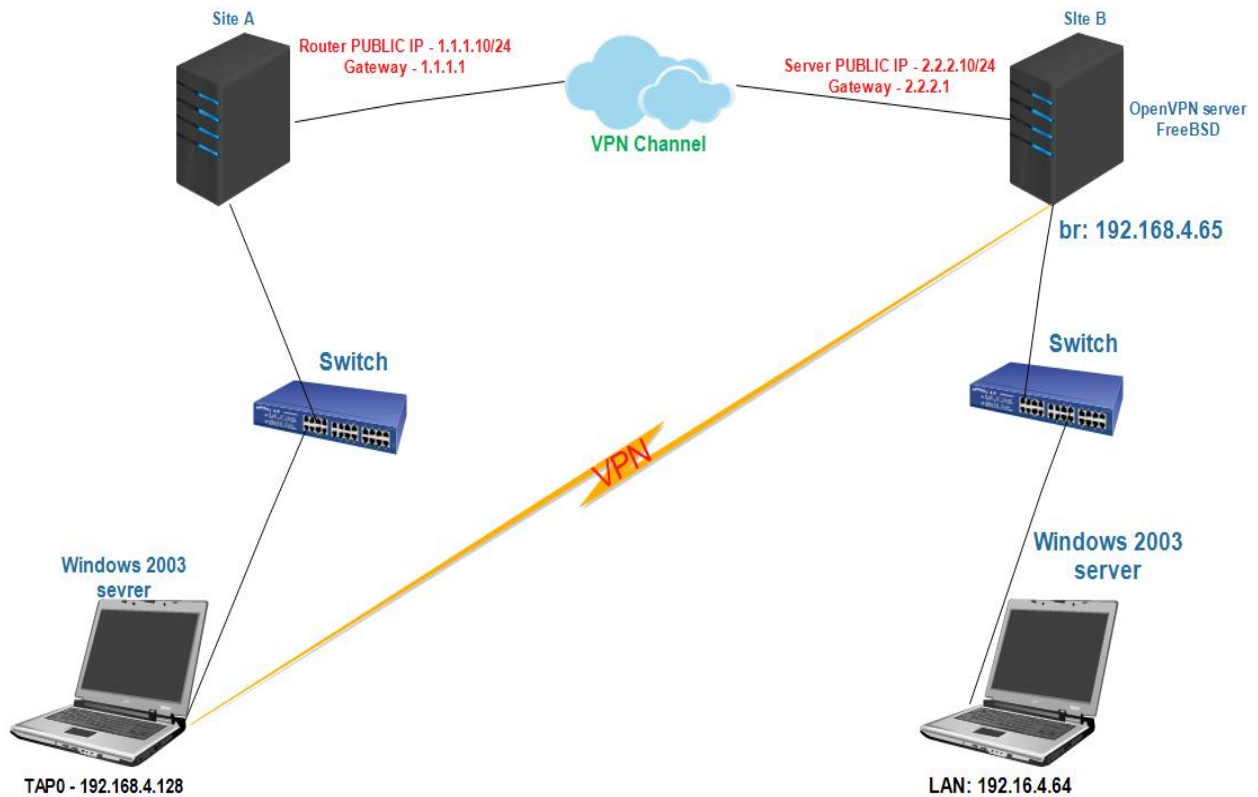
Bridge şəbəkə kartının istifadə edilməsinin əsas səbəbi ondan ibarətdir ki, bütün qoşulmuş clientlər üçün tək broadcast domain yaratmaqdır (Yeni adı LAN şəbəkə və VPN istifadəçilər eyni SUBNET üzərində olacaqlar).

Digər səbəbi isə IP olmayan trafik yönləndirilməsidir (IPX və ya AppleTalk protocoolları).

Bu başlıq harda broadcast domain funksionallığı olarsa və əgər düzgün kanaldan IP olmayan trafik axını ötürülərsə, **tcpdump** və **wireshark** alətlərinin istifadəs ilə onu təyin edəcək.

## İşə hazırlaşaq

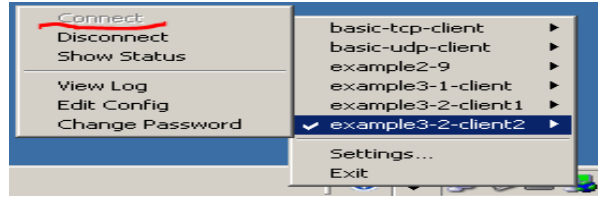
Bu misalımızda biz UNIX-Bridge misalının quruluşunu eyni olaraq istifadə edəcəyik. Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



Bu misalda server maşınımız FreeBSD9.2 x64 OpenVPN2.3 və quraşdırma faylı isə eynilə Bridge-UNIX-də istifadə elədiyimiz kimi **example3-3-server.conf** olacaq. Serverin LAN-ı ilə eyni segmentdə olan client Windows2k3 server var. VPN Client kimi istifadə edilən maşın isə 2003 server OpenVPN2.3-də və quraşdırma faylı isə **client-to-client traffic**-də istifadə etdiyimiz **example3-2-client2.ovpn** olacaq.

Əmin olun ki, hər iki windows maşınlarında AppleTalk və IPX protocoolları yüklənmişdir. Bu protocoolları 2003 server maşınlarının Local Area Network adapterlərinə mənimsədin. Həmçinin hər 2 windows maşına Wireshark snifferi yükləyin.

1. Öncə göstərdiyimiz kimi FreeBSD maşında network bridge yaradın və əmin olun ki, hər şey işləyir.
2. OpenVPN client-i işə salaq.



3. Client uğurla qoşulduqdan sonra öncə ARP mesajları yoxlayaq. `tcpdump -nnel -i bridge0 arp` əmri ilə.

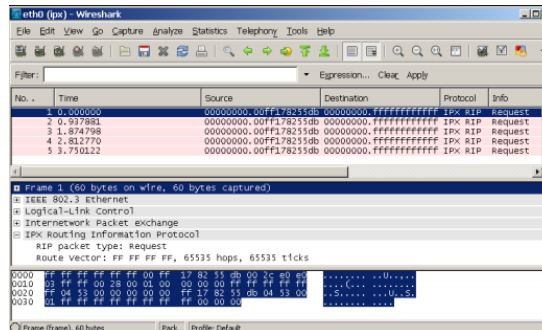
```
root@site:/usr/local/etc/openvpn # tcpdump -nnel -i bridge0 arp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on bridge0, link-type EN10MB (Ethernet), capture size 65535 bytes
23:34:52.285232 00:ff:9c:64:f6:c3 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: Request who-has 192.168.4.128 tell 192.168.4.128, length 28
23:34:52.482448 00:ff:9c:64:f6:c3 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: Request who-has 192.168.4.128 tell 192.168.4.128, length 28
23:34:53.482409 00:ff:9c:64:f6:c3 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: Request who-has 192.168.4.128 tell 192.168.4.128, length 28
23:39:26.939028 00:00:29:03:72:08 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 60: Request who-has 192.168.4.128 tell 192.168.4.64, length 46
23:39:26.741617 00:ff:9c:64:f6:c3 > 00:00:29:03:72:08, ethertype ARP (0x0806), length 42: Reply 192.168.4.128 is-at 00:ff:9c:64:f6:c3, length 28
23:39:26.745751 00:ff:9c:64:f6:c3 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: Request who-has 192.168.4.65 tell 192.168.4.128, length 28
23:39:26.745775 02:ef:90:ec:8f:00 > 00:ff:9c:64:f6:c3, ethertype ARP (0x0806), length 42: Reply 192.168.4.65 is-at 02:ef:90:ec:8f:00, length 28
23:43:32.042609 00:00:29:03:72:08 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 60: Request who-has 192.168.4.1 tell 192.168.4.64, length 46
```

4. Sonra işə serverlə eyni LAN sergmentdə yerləşən windows 2003 serverin broadcast axınını yoxlayaq hansı ki, OpenVPN client olan windows 2003 server tərəfdən gəlir. Bunun üçün LAN Wireshark istifadə edəcəyik. Wireshark susmaya görə seçdiyiniz adapter üzərində olan bütün trafiki tutur.

|     |            |                     |                   |         |                                                               |
|-----|------------|---------------------|-------------------|---------|---------------------------------------------------------------|
| 195 | 90.3544570 | 192.168.4.128       | 192.168.4.64      | SMB     | 93 Tree Disconnect Request                                    |
| 196 | 90.3544980 | 192.168.4.64        | 192.168.4.128     | SMB     | 93 Tree Disconnect Response                                   |
| 197 | 90.3568260 | 192.168.4.128       | 192.168.4.64      | TCP     | 60 nim > netbios-ssn [FIN, ACK] Seq=1072 Ack=931 win=63310 L  |
| 198 | 90.3569020 | 192.168.4.64        | 192.168.4.128     | TCP     | 54 netbios-ssn > nim [FIN, ACK] Seq=931 Ack=1073 win=63169 L  |
| 199 | 90.3569940 | 192.168.4.128       | 192.168.4.255     | BROWSEF | 228 Request Announcement CAMAL-GKBDQ462X                      |
| 200 | 90.3574030 | 00000000.000c290372 | 00000000.ffffffff | BROWSEF | 234 Local Master Announcement CAMAL-ZBCKELNER, Workstation, S |
| 201 | 90.3576780 | 192.168.4.64        | 192.168.4.255     | BROWSEF | 243 Local Master Announcement CAMAL-ZBCKELNER, Workstation, S |
| 202 | 90.3589770 | 192.168.4.128       | 192.168.4.64      | TCP     | 60 nim > netbios-ssn [ACK] Seq=1073 Ack=932 win=63310 Len=0   |

Bu çıxışda biz həmçinin çoxlu NetBios broadcast trafikini görə bilərik hansı ki, OpenVPN client şəbəkə ilk qoşulanda göndərir.

5. İndi işə gəlin VPN-ə qoşulmuş 2003 serverin IPX trafikinə baxaq.



Burda IP olamayan axını siz görə bilərsiniz hansı ki, bridge üzərindən gəlir.

### Bu necə işləyir

Bridge üzərindən keçən axın Wireshark proqramı tərəfindən ələ keçirilmişdir. Düzgün filter etməklə siz yalnız özünüze lazım olan OpenVPN client-in datasını Wireshark ilə sniff edə bilərsiniz. Problemin araşdırılması məqamına çatdıqda bu çox önəmli olur.

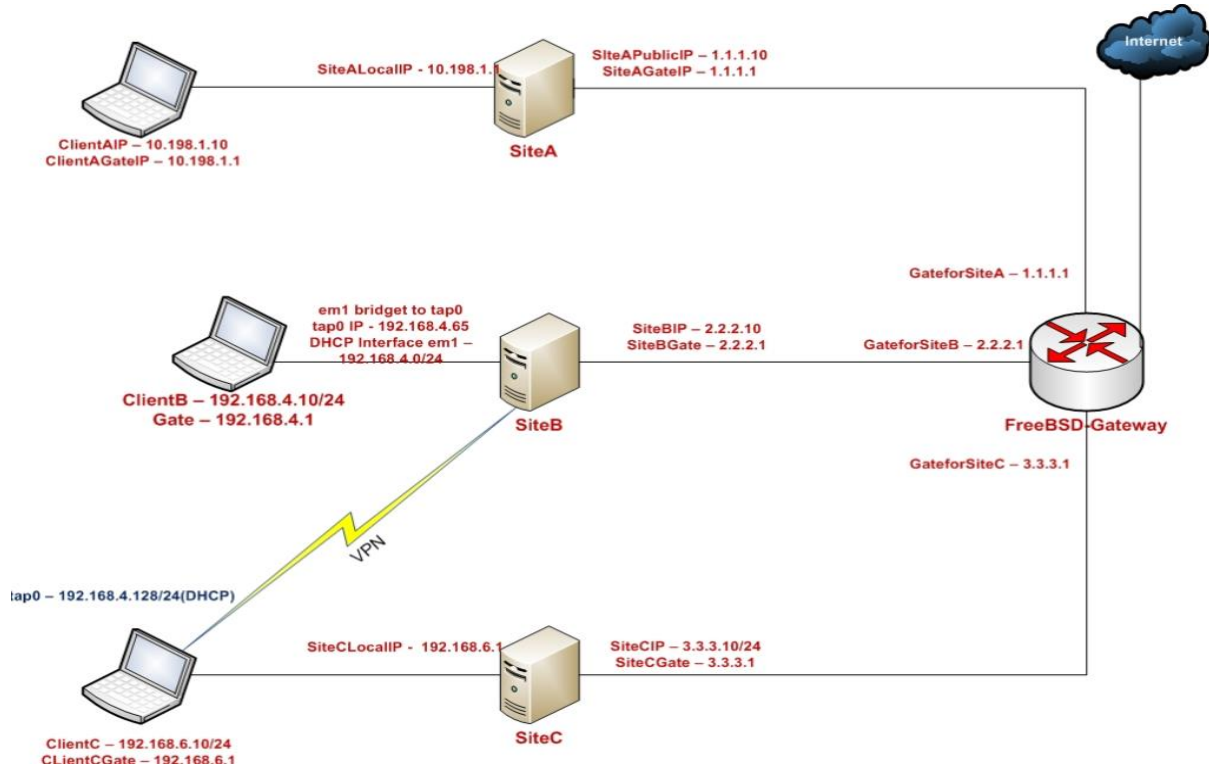


## Kənar DHCP serverin istifadə edilməsi

Bu misalda biz OpenVPN serverimiz üçün client-lərə IP ünvanı öz DHCP-sindən deyil artıq kənar DHCP serverdən verəcəyik.

### İşə başlayaq

Biz aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



Client və server sertifikatlarını eynilə 2-ci başlıqda olanı istifadə edəcəyik. Yalnız client-server IP şəbəkələrində işləyir.

Bu başlıq üçün server maşını FreeBSD9.2 x64 OpenVPN2.3-də işləyir. Client isə Windows7 OpenVPN2.3-də işləyir. Bu client üçün **example3-2-client2.ovpn** quraşdırma faylından istifadə edəcəyik hansı ki, client-to-client misalımızda istifadə etmişdik.

Bunu necə edək

1. Server quraşdırma faylını yaradaq:

```

proto udp
port 1194
dev tap0

server-bridge

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnserver.crt
key /usr/local/etc/openvpn/openvpnserver.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0

```

```

persist-key
persist-tun

keepalive 10 60

user nobody
group nobody

daemon
log-append /var/log/openvpn.log

```

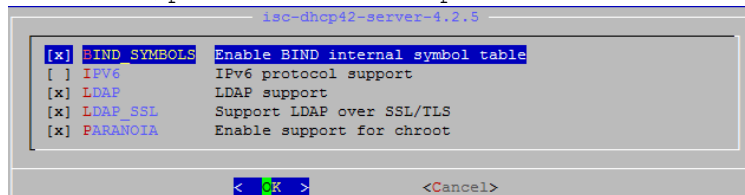
Sonra həmin faylı **example3-6-server.conf** adında yadda saxlayın.

2. Sonra serveri işə salaq:

Öncə DHCPD serveri qaldıraq.

```
root@siteB:/ # cd /usr/ports/net/isc-dhcp42-server/ # Portuna daxil olaq.
```

```
root@siteB:/usr/ports/net/isc-dhcp42-server # make config
```



```
root@siteB:/usr/ports/net/isc-dhcp42-server # make install # Yükləyirik
```

DHCP üçün quraşdırma faylı aşağıdakı kimi olacaq.

```
root@siteB:/usr/ports/net/isc-dhcp42-server # cat
```

```

/usr/local/etc/dhcpd.conf
option domain-name "internal.freebsd";
option domain-name-servers 188.72.128.10;

```

```

default-lease-time 3600;
max-lease-time 86400;
ddns-update-style none;

```

```

subnet 192.168.4.0 netmask 255.255.255.0 {
 range 192.168.4.129 192.168.4.200;
 option routers 192.168.4.1;
}

```

Startup quraşdırmamız aşağıdakı kimi olacaq.

```

cloned_interfaces="bridge0"
autobridge_interfaces="bridge0"
autobridge_bridge0="em1 tap0"
ifconfig_bridge0="up"
ifconfig_tap0="inet 192.168.4.65 netmask 255.255.255.0 up"

openvpn_enable="YES"
openvpn_if="tap bridge"
openvpn_configfile="/usr/local/etc/openvpn/example3-6-server.conf"
openvpn_dir="/usr/local/etc/openvpn"

dhcpd_enable="YES"

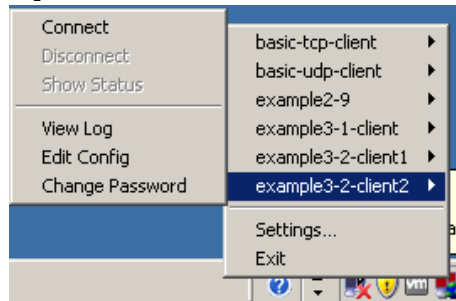
```

```
dhcpd_ifaces="em1"
```

Serverimizi reboot etdikdən sonra VPN servisimizi işə salaq.  
**/usr/local/etc/rc.d/openvpn start**

### 3. Windows Client-i işə salaq:

**Qeyd:** Client-i işə salanda nəzərə alın ki, UNIX/Linux-larda bu avtomatik işləməyəcək. VPN qoşulma olduqdan sonra avtomatik olaraq IP ünvan DHCP ilə ötürülmür(**dhclient tun0** əmri ilə IP alınacaq.).



**Qeyd:** Nəzərə alın ki, ilk qoşulmada IP ünvan gəlməyə bilər. Ona görə də öncədən aşağıdakı əmrlərlə yoxlanış edin. Sonra **disconnect** edib yenidən yoxlayın.

```
ipconfig /release
ipconfig /renew
```

### 4. VPN qoşulma uğurlu olduqdan sonra isə VPN serverinizdə aşağıdakı əmrlə DHCP-dən gələn IP ünvanı görə bilərsiniz.

```
C:\>ipconfig /all|more
```

```
Ethernet adapter TAP:
 Connection-specific DNS Suffix . : internal.freebsd
 Description : TAP-Windows Adapter V
 Physical Address. : 00-FF-9C-64-F6-C3
 DHCP Enabled. : Yes
 Autoconfiguration Enabled : Yes
 IP Address. : 192.168.4.131
 Subnet Mask : 255.255.255.0
 Default Gateway :
 DHCP Server : 192.168.4.1
 DNS Servers : 188.72.128.10
```

Routing cədvəlimizə baxaq.

```
C:\>netstat -rn
```

Active Routes:

| Network        | Destination    | Netmask         | Gateway       | Interface     | Metric |
|----------------|----------------|-----------------|---------------|---------------|--------|
| 0.0.0.0        | 0.0.0.0        | 0.0.0.0         | 10.198.1.1    | 10.198.1.10   | 10     |
| 10.198.0.0     | 10.198.0.0     | 255.255.0.0     | 10.198.1.10   | 10.198.1.10   | 10     |
| 10.198.1.10    | 10.198.1.10    | 255.255.255.255 | 127.0.0.1     | 127.0.0.1     | 10     |
| 10.255.255.255 | 10.255.255.255 | 255.255.255.255 | 10.198.1.10   | 10.198.1.10   | 10     |
| 127.0.0.0      | 127.0.0.0      | 255.0.0.0       | 127.0.0.1     | 127.0.0.1     | 1      |
| 192.168.4.0    | 192.168.4.0    | 255.255.255.0   | 192.168.4.131 | 192.168.4.131 | 30     |
| 192.168.4.131  | 192.168.4.131  | 255.255.255.255 | 127.0.0.1     | 127.0.0.1     | 30     |
| 192.168.4.255  | 192.168.4.255  | 255.255.255.255 | 192.168.4.131 | 192.168.4.131 | 30     |
| 224.0.0.0      | 224.0.0.0      | 240.0.0.0       | 10.198.1.10   | 10.198.1.10   | 10     |
| 224.0.0.0      | 224.0.0.0      | 240.0.0.0       | 192.168.4.131 | 192.168.4.131 | 30     |

```
255.255.255.255 255.255.255.255 10.198.1.10 10.198.1.10 1
255.255.255.255 255.255.255.255 192.168.4.131 192.168.4.131 1
Default Gateway: 10.198.1.1
```

5. Sonda VPN serverimizin LAN-ında olan clientlərin birinə Windows7 client-dən ping atıb yoxlayaq.

```
C:\>ping -n 2 192.168.4.10
Pinging 192.168.4.10 with 32 bytes of data:
Reply from 192.168.4.10: bytes=32 time=1ms TTL=128
Reply from 192.168.4.10: bytes=32 time=4ms TTL=128
```

### **Bu necə işləyir.**

Server direktivi:

```
server-bridge
```

Rəsmi saytında yazıldığı kimi desək, DHCP-proxy istifadə edərək Ethernet bridging-i aktivləşdirir hansı ki, clientlər OpenVPN server tərəfdə olan DHCP serverə deyir ki, xahiş edirəm mənə IP ünvan və DNS serverlər haqqında məlumatları ver. Ancaq siz öncə öz serverinizdə DHCP tərəfə baxan LAN kartınız ilə TAP alətinizi öncədən bridge etməlisiniz. Bu rejim yalnız Windows tipli clientlərdə işləyir harda ki, client tərəfdə olan TAP aləti **dhclient** əmrlərini daxil edir.

Qeyd: Ancaq mən yenədə FreeBSD9.2 x64-də yoxladım işlədi ☺.

### **Daha da ətraflı**

#### **DHCP serverin quraşdırılması**

DHCP serverin düzgün quraşdırılması ondan ibarətdir ki, VPN clientlərdən gələn DHCP müraciətləri birbaşa default gateway-i özünə götürə bilməsin. Çünki, bu DHCP serverin inzibatçılığına əlavə yük verir.

Bu halda, bizə daha rahat olar ki, hər bir clientin quraşdırma faylında unikal MAC ünvan təyin edək. Məsəl üçün:

```
lladdr CA:C6:F8:FB:EB:3B
```

Linux/UNIX maşınlarında TAP interfeys qalxan kimi MAC ünvanlar təsadüfi hesablanır ona görə ki, OpenVPN client hər dəfə dayananda və yenidən işə salınanda fərqli IP ünvan əldə edilə bilsin. Ancaq siz yenədə birdəfəlik sətir olan MAC ünvanı TAP alətinizə mənimsədə bilərsiniz ki, OpenVPN client işə düşəndə avtomatik eyni qalsın.

Həmçinin bunu Windows maşındada eləmək olar ancaq, siz Windows reestrində bu dəyişikliyi etməlisiniz.

#### **DHCP relay**

Başqa bir yolda vardır ki, bridge rejim istifadə edilmədən kənar DHCP serveri istifadə edə bilərsiniz. Əgər OpenVPN server üçün quraşdırma faylını bu başlıqda olanı istifadə edirsinizsə və əgər TAP aləti serveri işə düşməzdən öncə quraşdırılıbsa, bu misalda external DHCP serveri siz Linux **dhrelay** və UNIX **dhcprelay** ilə çağırma bilərsiniz. UNIX-də DHCP relay kimi **dhcprelaya** paketindən istifadə edə bilərsiniz.

```
root@dhcp:/usr/local/share/doc/dhcrelya # dhcrelya -i em1 tap0
```

Əmin olun ki, sizin em1 kartınızın DHCP serverə çıxışı mövcuddur. 2-ci başlıqda istifadə elədiyiniz Proxy-ARP-dan oxuduğumuz kimi əgər **proxy-arp** scriptini istifadə eləmək istəsək, əksər hallarda bridge interfeyslərdən istifadə eləmək lazımdır.

### Status faylının istifadə edilməsi

OpenVPN serverin çoxlu imkanı vardır hansı ki, serverə qoşulmuş clientləri monitoring eləmək üçün imkanları mövcuddur. Əksər istifadə edilən üsulu **status** faylıdır. Bu başlıqda OpenVPN status faylının necə oxunmasını araşdıracağıq. Biz həmçinin tap stilli şəbəkələrin status faylını araşdıracağıq.

### İşə başlayaq

Eynilə 2-ci başlıqda yaratdığımız Client-Server sertifikatlarından yenidən istifadə edəcəyik (Yalnız IP şəbəkələr üçün). Bu başlıq üçün vpn serverimiz FreeBSD9.2 x64-də olacaq. Clientlər biri FreeBSD9.2-də o biri isə Windows7-də olacaq. FreeBSD client üçün **example3-1-client.conf** faylından, **Windows7** client üçün isə **example3-2-client2.ovpn** istifadə edəcəyik.

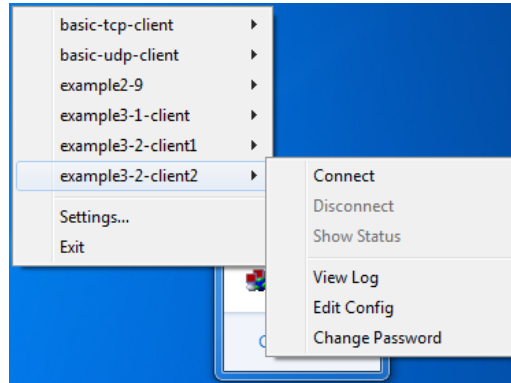
### Bunu necə edək

1. **example3-1-server.conf** faylını **example3-7-server.conf** faylına nüsxələyin və içinə aşağıdakı sətiri əlavə edin:  

```
status /var/log/openvpn.status
```
2. Serveri işə salın:  
root@siteA:/usr/local/etc/openvpn # **openvpn --config example3-7-server.conf**
3. İlk olaraq FreeBSD clienti öncə istifadə elədiyimiz quraşdırma ilə işə salaq və server tərəfin LAN gateway-inə ping atıb yoxlayaq:  
root@siteB:/usr/local/etc/openvpn # **openvpn --config example3-1-client.conf**  
root@siteB:~ # **ping 10.198.0.1**
4. VPN qoşulması uğurlu olduğdan sonra serverinizdə **openvpn.status** faylını (root istifadəçi adından) list edin:  
root@siteA:/usr/local/etc/openvpn # **cat /var/log/openvpn.status**  
OpenVPN CLIENT LIST  
Updated, Fri Feb 7 15:20:13 2014  
Common Name, Real Address, Bytes Received, Bytes Sent, Connected Since  
openvpnclient1, 2.2.2.10:48131, 10134, 11722, Fri Feb 7 15:17:15 2014  
ROUTING TABLE  
Virtual Address, Common Name, Real Address, Last Ref  
00:bd:e9:8f:00:00, openvpnclient1, 2.2.2.10:48131, Fri Feb 7 15:17:52 2014  
GLOBAL STATS  
Max bcst/mcast queue length, 0

END

5. İndi isə **Windows7** client-i işə salaq:



6. Client-dən serverin LAN IP-sinə ping ataq.

```
C:\Users\ClientC>ping 10.198.0.1
```

7. Sonra yeniden serverin status faylına baxaq:

```
root@siteA:/usr/local/etc/openvpn # cat /var/log/openvpn.status
OpenVPN CLIENT LIST
Updated,Fri Feb 7 15:38:22 2014
Common Name,Real Address,Bytes Received,Bytes Sent,Connected Since
openvpnclient1,2.2.2.10:46325,14508,16096,Fri Feb 7 15:20:33 2014
openvpnclient2,3.3.3.10:49157,47760,11588,Fri Feb 7 15:35:03 2014
ROUTING TABLE
Virtual Address,Common Name,Real Address,Last Ref
00:ff:b8:35:d5:7d,openvpnclient2,3.3.3.10:49157,Fri Feb 7 15:35:05
2014
00:bd:4f:dd:00:00,openvpnclient1,2.2.2.10:46325,Fri Feb 7 15:20:35
2014
GLOBAL STATS
Max bcast/mcast queue length,0
END
```

### **Bu necə işləyir...**

Hər dəfə yeni müştəri qoşulan kimi OpenVPN serverdə olan status faylı yenilənir. **OPENVPN CLIENT LIST** və **ROUTING TABLE** və çoxlu maraqlı digər hissələr hansı ki, aşağıda açıqlanır:

- Hansı clientlər qoşuludur
- Hansı IP ünvanlardan clientlər qoşulurlar
- Hər bir clientin ötürdüyü və qəbul etdiyi baytlar rəqəmlərlə
- Hansı müştəri qoşulmuşdur və onun vaxtı

Həmçinin routing cədvəlidə göstərilir ki, hansı şəbəkələr hər bir clientə route edilmişdir. Bu routing cədvəli, clientlər trafik ötürən kimi yaranmağa başlanır. Biz istifadə etdiyimiz ping əmri ilə məhz özümüz routing table-in yaranmasının işini öncədən görürük.

**Daha da ətraflı...**

### **TUN stilli şəbəkələrin fərqi**

TUN və TAP alətlərinin əsas fərqi (2-ci başlıqda status faylının istifadə edilməsinə baxın), ROUTING TABLE-dir. Məsələn öncəki başlıqdan göstərilir:  
192.168.200.2, openvpnclient1, 192.168.4.65:56764, <Date>

Hardaki bu misalda biz görürük:

```
00:bd:4f:dd:00:00,openvpnclient1,2.2.2.10:46325,Fri Feb 7 15:20:35 2014
```

Bu isə -> 00:bd:4f:dd:00:00 təsadüfi MAC ünvanıdır hansı ki, openvpnclient1 maşınında TAP alət tərəfindən generasiya edilib.

### **Client-lərin qoşulmadan ayrılması**

Qeyd edin ki, clientlər qoşulmadan ayrılan kimi, status faylı həmin anda da yenilənmir. OpenVPN ilk olaraq serverdə yazılmış **keepalive** parametrində olan vaxta əsasən client-ə yenidən qoşulmağa çalışır. Bu misalda server quraşdırma faylı aşağıdakı sətirdən istifadə edir:

```
keepalive 10 60
```

Bu serverə deyir ki, hər **10** saniyədən bir ping eləsin. OpenVPN ikinci argumenti ikiye vurur. Əgər o **2\*60** saniyədən sonra cavab almırsa, qoşulmaya restart gedir. Server həmçinin client-ə deyir ki, serveri **10** saniyədən bir **ping** elə və əgər **60** saniyədən sonra ping yenədə getməzsə, qoşulmanı qır və yenidən yarat.

Əgər client öz qoşulmasını **explicit-exit-notify** direktivi sayəsində kəsirsə, yada TCP bazalı quruluşdan istifadə edilirsə, server ping cavabını client-dən gözləmir.

### **Həmçinin baxaq**

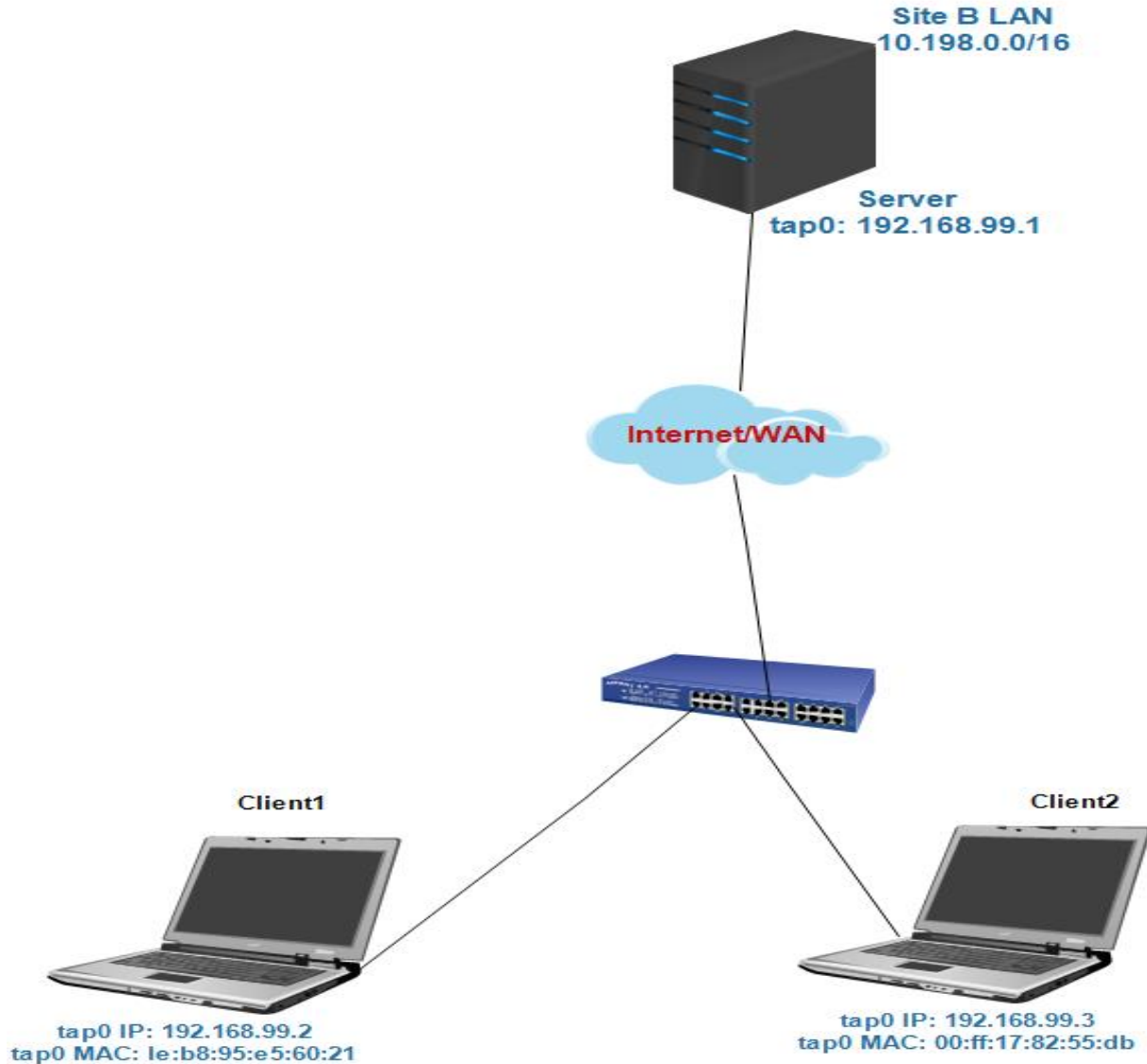
2-ci başlıqda status faylının istifadəsi var hansı ki, IP şəbəkələrində status faylının daha əhəmiyyətli istifadəsini açıqlayır.

### **Management Interfeys**

Bu başlıqda göstərilir ki, OpenVPN-i management interfeys sayəsində necə manage etmək olar.

### **İşə başlayaq**

Biz aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



2-ci başlıqda yaratdığımız sertifikatları eynilə burda istifadə edəcəyik. Bu misalda server üçün FreeBSD9.2 x64 OpenVPN2.3 istifadə edəcəyik. Server üçün isə **example3-1-server.conf** quraşdırma faylını istifadə edəcəyik. İlk client FreeBSD9.2 x64 OpenVPN2.3, ikinci client isə Windows7 OpenVPN2.3-də olacaq. FreeBSD client üçün **example3-1-client.conf** faylından, Windows7 client üçün isə **example3-2-client2.ovpn** client faylından istifadə edəcəyik.

#### Necə edək...

1. Server quraşdırma faylı üçün **example3-1-server.conf** faylını **example3-8-server.conf** faylına nüsxələyin və **example3-8-server.conf** faylına aşağıdakı sətiri əlavə edin:
 

```
management tunnel 23000 stdin
```

2. Serveri işə salaq:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example3-8-server.conf
Enter Management Password:
```

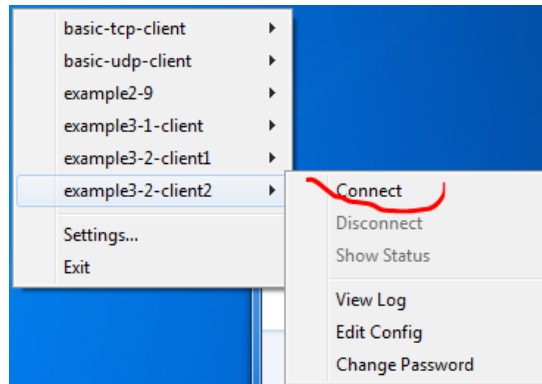


OpenVPN server sizdən Management interfeys üçün şifrə istəyəcək onu daxil edin.

3. Öncəki misalımızdan olan quraşdırma faylını istifadə edərək client1-i işə salın:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config example3-1-client.conf
```

Həmçinin 2-ci olan Windows7 client-i işə salaq:



4. VPN qoşulmalar uğurlu olduqdan sonra isə biz serverin özündən öz management interfeysinə telnet proqramı ilə daxil ola bilərik:

```
root@siteA:/usr/local/etc/openvpn # telnet 192.168.99.1 23000
Trying 192.168.99.1...
Connected to 192.168.99.1.
Escape character is '^]'.
ENTER PASSWORD:freebbsd
SUCCESS: password is correct
>INFO:OpenVPN Management Interface Version 1 -- type 'help' for more info
status
OpenVPN CLIENT LIST
Updated,Fri Feb 7 19:38:18 2014
Common Name,Real Address,Bytes Received,Bytes Sent,Connected Since
openvpnclient1,2.2.2.10:43684,10480,12068,Fri Feb 7 19:33:22 2014
openvpnclient2,3.3.3.10:53211,50303,11270,Fri Feb 7 19:36:00 2014
ROUTING TABLE
Virtual Address,Common Name,Real Address,Last Ref
00:bd:d9:02:18:00,openvpnclient1,2.2.2.10:43684,Fri Feb 7 19:33:25 2014
00:ff:b8:35:d5:7d,openvpnclient2,3.3.3.10:53211,Fri Feb 7 19:36:03 2014
GLOBAL STATS
Max bcast/mcast queue length,0
END
```

5. Həmçinin imkanımız var ki, client-i bu interfeys ilə qoşulmadan ayıraq.

```
kill openvpnclient2
SUCCESS: common name 'openvpnclient2' found, 1 client(s) killed
status
OpenVPN CLIENT LIST
Updated,Fri Feb 7 19:41:36 2014
Common Name,Real Address,Bytes Received,Bytes Sent,Connected Since
openvpnclient1,2.2.2.10:43684,11487,13075,Fri Feb 7 19:33:22 2014
ROUTING TABLE
Virtual Address,Common Name,Real Address,Last Ref
00:bd:d9:02:18:00,openvpnclient1,2.2.2.10:43684,Fri Feb 7 19:33:25 2014
GLOBAL STATS
```

```
Max bcast/mcast queue length,0
END
```

6. **Ctrl+] ya "exit"** əmrindən istifadə edin ki, telnet programından çıxasınız.

### **Bu necə işləyir.**

OpenVPN server spesifik öz spesifik management interfeysini işə salmaq üçün aşağıdakı direktivdən istifadə edir:

```
management 127.0.0.1 23000 stdin
```

Və aşağıdakı parametrlərlə:

- tunnel əmri VPN serverin özünün gateway kimi aldığı IP ünvanında qulaq asmağa başlayır.
- Management interfeysin qulaq asması üçün lazım olan port.
- Sonuncu parametr isə ya şifrə faylının ünvanı ya da server işə düşəcək anda daxil edəcəyiniz şifredir. Nəzərə alın ki, bu şifrənin bağlı açarın şifrəsi ilə ya da openvpn-də istifadə edilən istifadəçilər üçün yaradılmış digər şifrelərlə heç bir əlaqəsi yoxdur.

Management interfeys işə düşən kimi serverin operatoru telnet programı ilə interfeysə daxil ola və lazım olan müraciətləri yollaya bilər. Aşağıdakı əmri istifadə edərək operator client-i qoşulmadan ayıra bilər:

```
kill <clientcommonname>
```

Qeyd edin ki, əgər OpenVPN client avtomatik qoşulmaya quraşdırılıbsa, o bir neçə dəqiqədən sonra yenidən qoşulacaq.

Əgər siz 2-ci başlıqda istifadə edilən management interfeysin status çıxışı ilə indikini müqayisə eləsəniz görə bilərsiniz ki, əsas fərqi VPN IP ünvanların əvəzinə MAC ünvanlar list edilmişdir. OpenVPN istənilən hallarda client-in IP ünvanını bilmək məcburiyyətində deyil hansı ki, onlar kənar DHCP serverdən IP ünvanlar ala bilərlər.

### **Daha da ətraflı**

#### **Client tərəfin management interfeysi**

Management interfeysi həmçinin client tərəfdə işə salına bilər. 2-ci başlıqda Client-Server IP şəbəkələrində management interfeysə baxın.

Planlaşdırılır ki, gələcəkdə həm client və həm də server tərəfdə management interfeysi daha bol imkanlarla inkişaf etdiriləcək.

### **Həmçinin baxın**

- 2-ci başlıqda Management interfeys hansı ki, client tərəfdə olan management interfeysi daha da açıqlayır.
- 2-ci başlıqda olan status faylı hansı ki, TUN tipli alətlər üçün status faylının detallarını açıqlayır.

## BÖLÜM 4

### PKI, Sertifikatlar və OpenSSL

Bu başlıqda aşağıdakılar açıqlanacaq:

- Sertifikatın generasiya edilməsi
- xCA: (1-ci hissə) PKI idarəedilməsi üçün GUI
- xCA: (2-ci hissə) PKI idarəedilməsi üçün GUI
- OpenSSL imkanları: x509, pkcs12, çıxışın yoxlanılması
- Sertifikatların revoke (Vaxtını sıfırlamaq) edilməsi
- CRL-lərin istifadə edilməsi
- vaxtı-bitmiş/revoke edilmiş sertifikatların yoxlanılması
- Aralıq CA-lar
- Çoxlu CA-lar: **--capath istifadə edərək stacking**

#### **Giriş**

Bu başlıqda Public Key Infrastructure (PKI)-ə, certificates və openssl əməlləri haqqında qısa gəzinti edəcəyik. Bu başlıqda istifadə edilən misallarda biz göstərəcəyik ki, OpenVPN üçün istifadə edilən sertifikatlar necə generasiya edilir, necə idarə edilir, necə baxılır və OpenVPN ilə OpenSSL arasında olan əlaqələrin necə olduğu açıqlanacaq.

## Sertifikatın generasiya edilməsi

Bu misalda biz openssl istifadə edərək sertifikatın necə generasiya edilib imzalanmasını göstərəcəyik. Bu əslində **easy-rsa** scriptlərindən biraz fərqlidir ama daha çox öyrədir.

### İşə başlayaq

2-ci başlıqdakı hissəni təkrar istifadə edərək easy-rsa scriptləri üçün mühit yaradaq və **vars** faylını işə salaq. Bu misalı biz həmişəki kimi FreeBSD9.2 x64 maşınında istifadə edirik.

### Necə edək...

OpenSSL-dən istifadə edərək Sertifikata imzalama və müraciət açma işlərindən öncə biz müəyyən mühit dəyişənlərini təyin etməliyik. Bu dəyişənlər **vars** faylında susmaya görə təyin edilməyib.

1. Çatışmayan mühit dəyişənlərini əlavə edək.

```
[root@siteA /]# cd /usr/local/etc/openvpn/itvpn
[root@siteA /usr/local/etc/openvpn/itvpn]# bash
[root@siteA /usr/local/etc/openvpn/itvpn]# source ./vars
[root@siteA /usr/local/etc/openvpn/itvpn]# export KEY_CN=dummy
[root@siteA /usr/local/etc/openvpn/itvpn]# export KEY_OU=dummy
[root@siteA /usr/local/etc/openvpn/itvpn]# export KEY_NAME=dummy
[root@siteA /usr/local/etc/openvpn/itvpn]# export
OPENSSL_CONF=/usr/local/etc/openvpn/itvpn/openssl-1.0.0.cnf
```

Qeyd edin ki, **openssl.cnf** faylı həm OS-un özündə **/etc/ssl** qovluğunda həm də, **easy-rsa** scriptlərinin içində susmaya görə gəlir. Biz **openssl-1.0.0.cnf** faylından ona görə istifadə edəcəyik ki, onu öncədən istifadə eləmişik. Və bizə lazım olan CA sertifikatları da həmçinin artıq **key** qovluğunda yerləşir.

2. Sonra isə biz sertifikat müraciətini **şifresiz** generasiya edək. Biz bunu **openssl req** əmrinə **-nodes** əlavə edərək edəcəyik.

```
[root@siteA /usr/local/etc/openvpn/itvpn]# cd keys/
[root@siteA /usr/local/etc/openvpn/itvpn/keys]# openssl req -nodes -
newkey rsa:1024 -new -out client.req -subj "/C=AZ/O=Itvpn/CN=MyClient"
Generating a 1024 bit RSA private key
.....++++++
....++++++
writing new private key to 'privkey.pem'

```

3. Sonda isə biz sertifikat müraciətini Certificate Authority gizli açarı ilə imzalayacağıq:

```
[root@siteA /usr/local/etc/openvpn/itvpn/keys]# openssl ca -in
client.req -out client.crt
Using configuration from /usr/local/etc/openvpn/itvpn/openssl-1.0.0.cnf
Enter pass phrase for /usr/local/etc/openvpn/itvpn/keys/ca.key:
Şifrəni Daxil-Edirik
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName :PRINTABLE:'AZ'
organizationName :PRINTABLE:'Itvpn'
commonName :PRINTABLE:'MyClient'
```

```
Certificate is to be certified until Feb 7 09:25:20 2024 GMT (3650 days)
Sign the certificate? [y/n]:y
```

```
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

### **Bu necə işləyir...**

İlk addımda həmişəki kimi gizli açar generasiya elədik. Bu misalda biz gizli açarda şifrə təyin etmədik və bu çox təhlükəlidir. Sertifikat müraciəti ona görə gizli açarla imzalanmışdır ki, sübut edə bilək ki, sertifikatın müraciəti və gizli açarı birlikdə olmalıdır. **openssl req** əmri həm sertifikatın müraciətini və həm də gizli açarı eyni anda generasiya edir.

İkinci addımda isə biz sertifikat müraciətini öz CA(Certificate Authority)-nin private(gizli) açarını istifadə edərək imzaladıq. Bu nəticədə X.509 sertifikat faylı verdi hansı ki, OpenVPN-də istifadə edilə bilər.

Həmçinin X.509(PUBLIC) sertifikatın bir nüsxəsidə **/usr/local/etc/openssl/itvpn/keys** ünvanında saxlanıldı. Bu nüsxə çox önəmlidir ona görə ki, çünki əgər biz bu sertifikatı gələcəkdə revoke eləmək istəsək elə burda olsun ki, rahat tapılsın.

### **Daha da ətraflı...**

Həmçinin mümkündür ki, **private.key**(gizli açar)-i şifrə ilə qoruya biləsiniz(OpenSSL terminlərində "**pass phrase**"). Bunun üçün sadəcə **-nodes** parametrini öncə istifadə elədiyimiz əmrdən silməyiniz yetər:  
[root@siteA /usr/local/etc/openssl/itvpn/keys]# **openssl req -newkey rsa:1024 -new -out client1.req -subj "/C=AZ/O=Itvpn/CN=MyClient1"**

OpenSSL bu hissədə şifrə tələb edir.

**Enter PEM pass phrase:**

**Verifying - Enter PEM pass phrase:**

### **Həmçinin baxaq**

2-ci başlıqda Public və Private açarların qurulması hansı ki, easy-rsa scriptlərini istifadə edərək PKI yaradıldı.

### **xCA: PKI-in idarə edilməsi üçün GUI(1-ci hissə)**

Bu misalda biz xCA-nin istifadəsini göstərəcəyik hansı ki, PKI-in public açarlarının idarə edilməsi üçün istifadə edilən qrafik alətdir. xCA opensource-dur Linux, Unix, Windows, MacOS üçün mövcuddur və <http://xca.sourceforge.net/> linkindən əldə edilə bilər. Bu misalımızda biz Windows xCA versiyasından istifadə edəcəyik. Bu misal iki hissəlikdir və 1-ci hissəsini açıqlayır. Hal-hazırda xCA bazasını yaradacağıq və CA sertifikatı

ilə private açarı import edəcəyik. Növbəti misalda biz xCA GUI istifadə edərək yeni sertifikatı yaradacağıq.

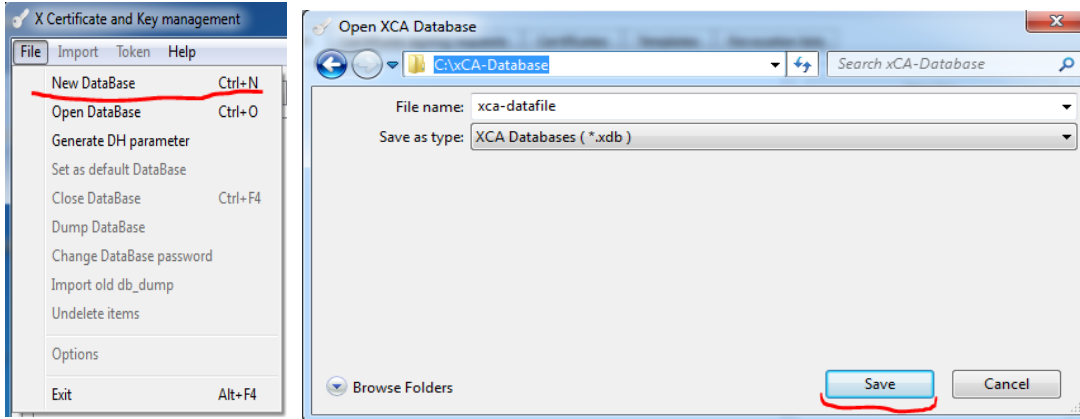
### İşə hazırlaşaq

<http://sourceforge.net/projects/xca/> ünvanından **setup\_xca-0.9.3.exe** faylını endirək. Bu misalda biz Windows7 üzərində sınaq etdik.

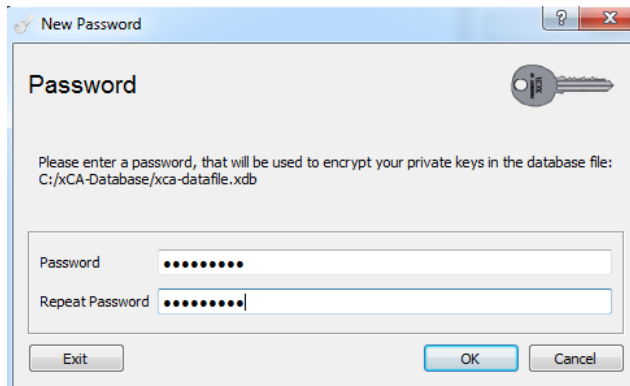
2-ci başlıqda **easy-rsa** ilə generasiya etdiyimiz **ca.key** və **ca.crt** fayllarının bir nüsxəsini Windows7 maşınına nüsxələyin.

### Necə edək...

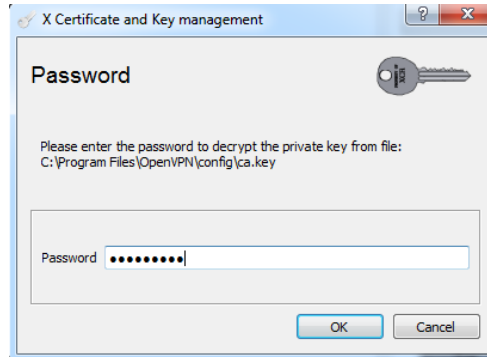
1. **xCA**-ni yüklədikdən sonra işə salın və **File -> New DataBase** düyməsini sıxaraq yeni database yaradın. Yeni yaradılacaq baza faylı üçün ünvan seçin və **OK** düyməsini sıxın.



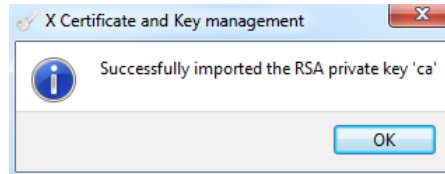
2. Sonra işə database faylı üçün sizə uyğun olan lakin bütün simvollar olan şifrəni iki dəfə daxil edin:



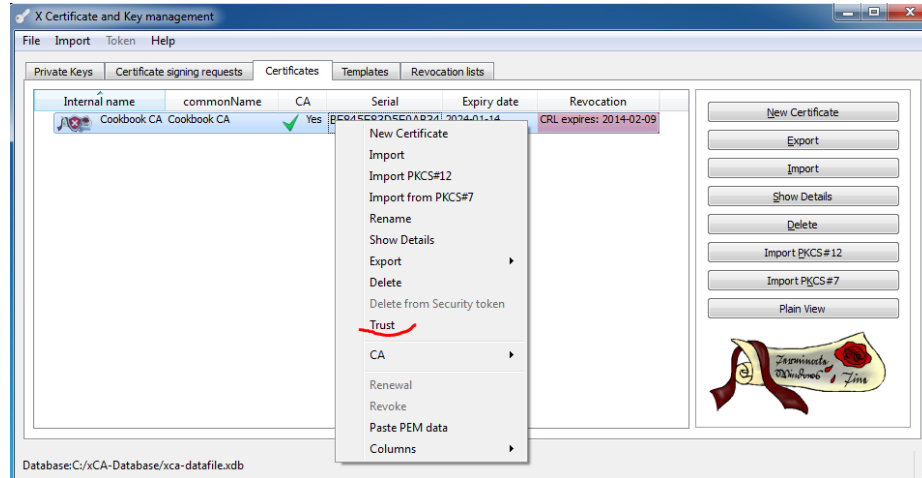
3. Sonra **Private key** bölümündə, **Import** düyməsinə sıxın ki, **ca.key** faylını import edəsiniz. Sizdən CA-nin generasiya edilməsində Private key faylı üçün istifadə edilən şifrə soruşulacaq:



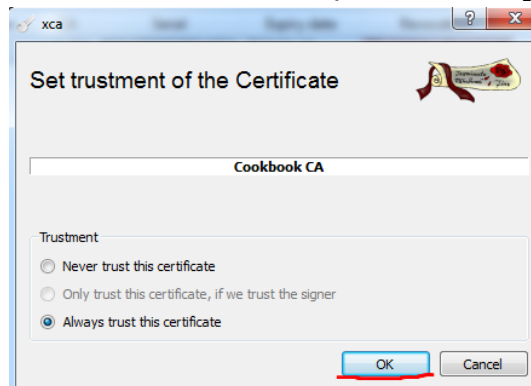
Nəticədə aşağıdakı şəkildə kimi uğurlu nəticə çap edilməlidir.



4. Sonra isə **Certificates** bölümündə keçin və **Import** düyməsini yenidən sıxın. Ardınca isə **ca.crt** faylını **Import** edin. Sonda **ca.crt** faylı uğurla yükləndikdən sonra üstündə sağ düyməni sıxın və **Trust** düyməsini sıxaraq inamlı edin.



5. **Always trust this certificate** seçin və **OK** düyməsini sıxın:



CA sertifikatını **trust** eləməklə siz artıq özünüzə şərait yaratdınız ki, yeni sertifikatlar generasiya edə və imzalayasınız.

### **Bu necə işləyir...**

xCA bütün public və private açarları öz bazasında saxlayır. Bazanı çətin şifrə ilə təyin etməlisiniz ona görə ki, onunla OpenVPN-də istifadə etdiyimiz bütün sertifikatlar imzalana və revoke edilə bilər.

### **Daha da ətraflı...**

Bu misalda biz PKI imkanı olaraq xCA seçdik. Həmçinin çoxlu PKI imkanları mövcuddur. Pullu və pulsuz:

- tinyCA: <http://tinyca.sm-zone.net/>
- OpenCA: <http://www.openca.org>

### **xCA: PKI idarəedilməsi üçün GUI (2-ci hissə)**

Bu başlıq ikinci hissədir hansı ki, xCA-nin istifadəsini açıqlayır. Bu başlıqda biz xCA GUI istifadə edərək yeni sertifikat yaradacağıq.

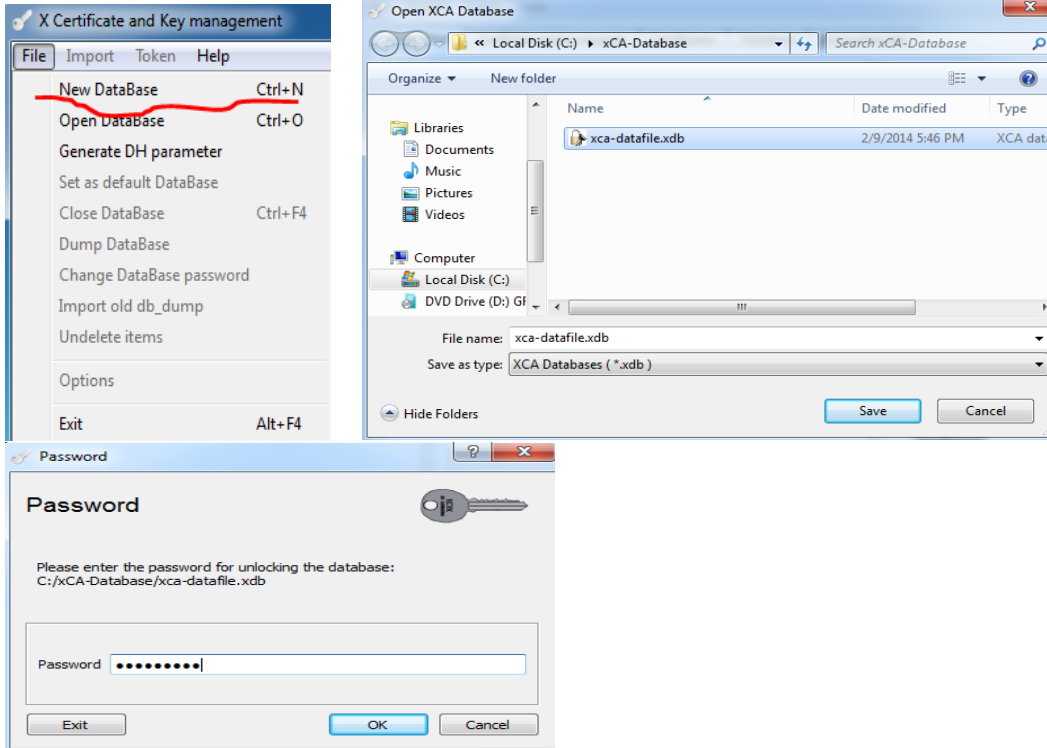
### **İşə başlayaq**

İlk olaraq öncəki misalı oxuyun və sonra aşağıdakı instruksiya ilə davam edin.

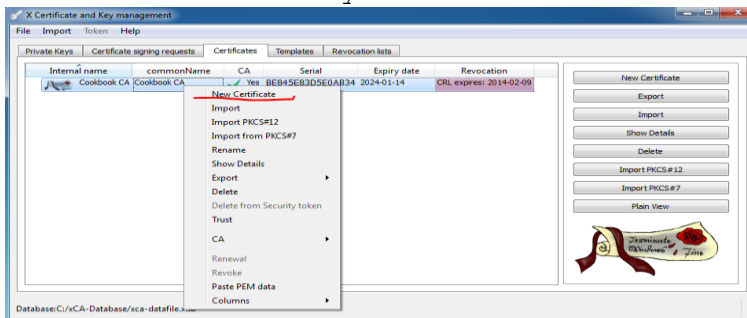
### **Necə edək...**

1. **xCA**-ni işə salın və **File -> Open Database** düyməsini sıxaraq yaratdığımız bazanı açın.

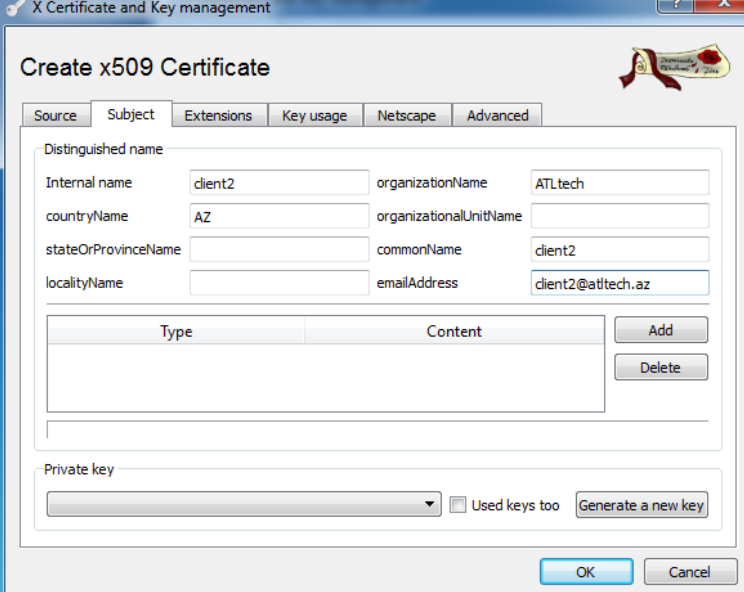




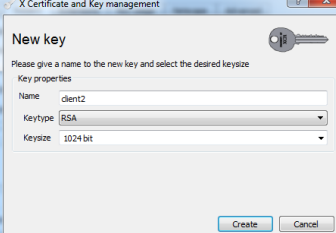
**Certificates** bölümünə keçin və öz **CA** sertifikatımızın üstündə sağ düyməni sıxın və **New Certificate** düyməsinə sıxın.



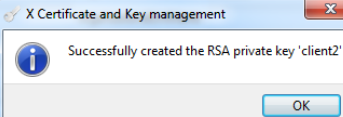
Yeni açılan interactive pəncərədə **Source** bölümünə keçin və **Internal name**, **Country Code**, **Organization**, **Common name** və **Email address** bölümünü yeni client üçün şəkilə uyğun olaraq doldurun:



Sonra **OK** düyməsinə sıxmayın; **Generate new key** düyməsinə sıxın



2. **Key size**-da **1024 bit** seçin (əgər istəmənsiz daha da böyük seçə bilərsiniz) və **Create** düyməsinə sıxın. Nəticə aşağıdakı kimi olmalıdır:

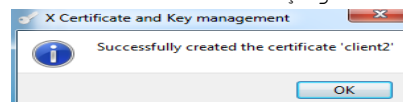


3. Ardınca da **Extensions** bölümünü aşağıdakı şəkildəki kimi doldurun:

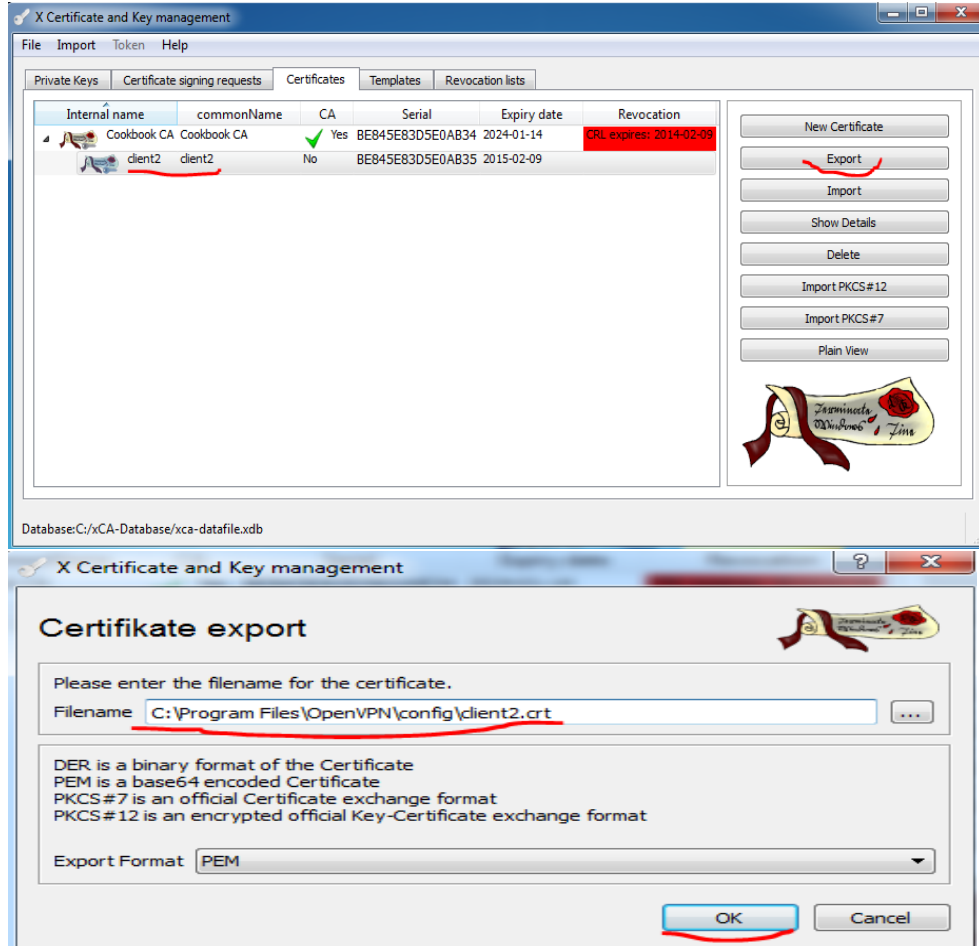
- **Type**-da **End Entity** seçin.
- **Subject Key Identifier**-ə işarə təyin edin.
- **Authority Key Identifier**-ə işarə təyin edin.

4. Sonda isə **Key Usage** bölümünə keçib şəkildəki kimi edin:

- **Key Usage** sütununda **Digital Signature** seçin.
- **Extended Key Usage** sütununda isə OpenVPN client sertifikatı üçün **TLS Web Client Authentication** seçin. Əgər OpenVPN server sertifikatı üçün seçmək istəsəniz onda **TLS Web Server Authentication** seçmək lazım olardı. Eyni sertifikat üçün heç vaxt ikisini də eyni anda seçməyin. Sonda **OK** düyməsini sıxın ki, sertifikat generasiya edilsin. Son addımdan sonra aşağıdakı şəkil çap edilməlidir.

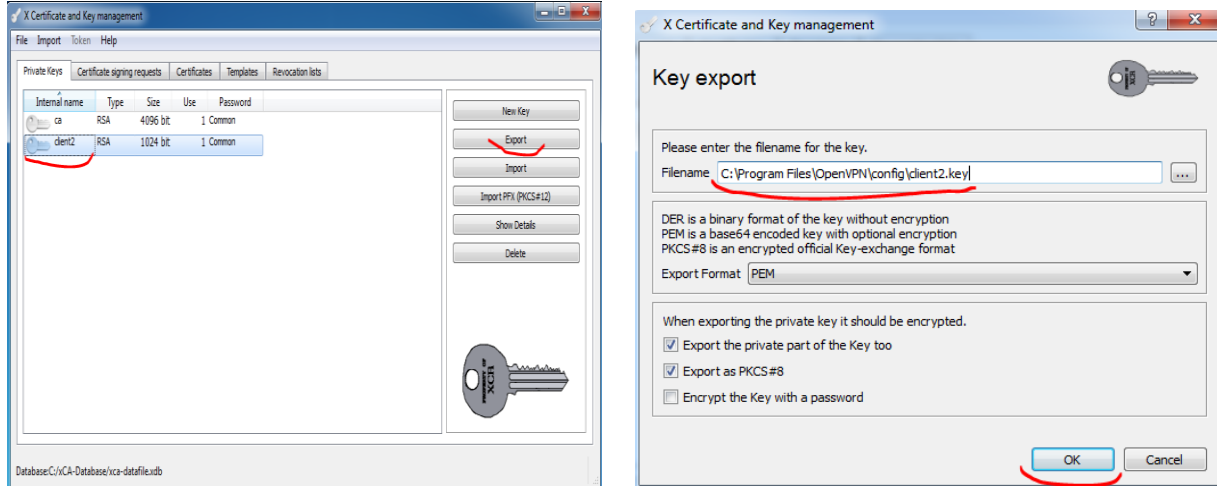


- Öncəki addımdan sonra isə biz yaratdığımız sertifikatı OpenVPN-də istifadə eləyə bilməmiş üçün export edəcəyik. Bunun üçün **Certificates** bölümündə **client2**-ni seçirik və **Export** düyməsinə sıxırıq:



Adını **client2.crt** seçin və **OK** düyməsinə sıxın.

Sonra **Private Keys** bölümünə gedin və eyni işi **client2** private key üçün edin. Adını isə **client2.key** seçin. Ardıcılığını şəkillərdə göstəririk:



### Bu necə işləyir...

Öz CA sertifikatımızı və **New certificate** seçərək, xCA yeni sertifikat generasiya elədi və bizim CA ilə imzaladı. Sertifikat imzalanmadan öncə bütün tələb edilən **x.509** sütunları dolu olmalıdır hansı ki, **Key usage** və **Extended Key usage**-də mütləqdir. Bu misalda həmçinin göstərir ki, hətta Public Key Infrastructre (PKI)-da GUI istifadə edilməsində elədə rahat deyil.

### Daha da ətraflı...

xCA GUI-nin çoxlu digər imkanları da vardır. Bunlar **sertifikatların generasiyası**, **Certification Revocation Lists (CRLs)** və digər **PKI-a** adı olan **subyektlər**. Ancaq bizim kitabın mövzusu deyil.

### OpenSSL imkanları: x509, pkcs12, çıxışın yoxlanılması

OpenSSL-in əmrleri ilk baxışdan çox çətin görünə bilər ancaq, OpenSSL alətlərin siyahısında çoxlu xeyirli olanları var hansı ki, x.509 sertifikatlarının idarə edilməsi üçün və bağlı açarlara baxmaq üçün istifadə edilir. Bu misalımızda biz bu imkanlardan bəzilərinin istifadə qaydasını göstərəcəyik.

### Hazırlaşaq

2-ci başlıqda örgəndiyimiz kimi **vars** faylından istifadə edərək **easy-rsa** sertifikat mühitini yaradın. Bu misal FreeBSD9.2 x64 maşında yerinə yetirilmişdir ancaq, Windows, MacOS və Linux maşındada yerinə yetirilə bilər.

### Necə edək...

1. Nəzərimizdə tutduğumuz sertifikatın subyektinə və bitmə müddətinə baxmaq üçün aşağıdakı əmrdən istifadə edirik (Windows7-də generasiya edib imzaladığımız sertifikatı yeni client2-ni FreeBSD maşına upload edin):

```
root@siteB:/ # cd /usr/local/etc/openvpn/
```

```
root@siteB:/usr/local/etc/openvpn # openssl x509 -subject -enddate -
noout -in client2.crt
subject= /C=AZ/O=ATLtech/CN=client2/emailAddress=client2@atltech.az
notAfter=Feb 9 13:53:00 2015 GMT
```

2. Sertifikat və private key-i PKCS12 formatına export edin:

```
root@siteB:/usr/local/etc/openvpn # openssl pkcs12 -export -in
client2.crt -inkey client2.key -out client2.p12
Enter Export Password:
Verifying - Enter Export Password:
```

```
root@siteB:/usr/local/etc/openvpn # chmod 600 client2.p12
```

Nəzərə alın ki, **chmod 600** o deməkdir ki, PKCS12 fayli yalnız istifadəçi tərəfindən oxunula bilər.

3. Verilmiş sertifikatın doğru olmasını yoxlayaq:

```
root@siteB:/usr/local/etc/openvpn # openssl verify -purpose sslclient -
CAfile ca.crt client2.crt
client2.crt: OK
```

4. Nəzərə alın ki, əgər biz səhv təyinat seçsək, bizə səhv qayıdacaq (**sslclient** əvəzinə **sslserver** olmalı idi):

```
root@siteB:/usr/local/etc/openvpn # openssl verify -purpose sslclient -
CAfile ca.crt openvpnsrver.crt
openvpnsrver.crt: /C=AZ/O=Itvpn/CN=openvpnsrver/emailAddress=openvpn-
ca@domain.lan
error 26 at 0 depth lookup:unsupported certificate purpose
OK
```

5. Sertifikatın şifrəsinin dəyişdirilməsi:

```
root@siteB:/usr/local/etc/openvpn # openssl rsa -in client2.key -aes256
-out cclient2.key
writing RSA key
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

### **Bu necə işləyir...**

OpenSSL-in kifayət qədər utilitləri vardır ki, x.509 sertifikatlarını həmçinin onlara uyğun olan private keyləri generasiya və idarə etsin. Bu başlıqda olan əmrlər onların müəyyən bir hissəsidir. UNIX və Linux OS-larda siz **openssl -h** əmrini daxil etməklə siz **x509**, **pkcs12** və **req** üçün man səhifələrini əldə edə bilərsiniz. Man səhifələr həmçinin aşağıdakı linkdəndə əldə edilə bilər:

<http://www.openssl.org/docs/apps/openssl.html>

## Sertifikatların Revoke (Vaxtını sıfırlamaq) edilməsi

Demək olar ki, sertifikatların Revoke edilməsi tez-tez hallarda lazım olmur. Ancaq elə bir hallar ola bilər ki, certificate verilmiş şəxsin yetkisini almaq lazım olur. Bu misalda biz easy-rsa scriptlərin istifadəs ilə sertifikatların revoke edilməsini örgənəcəyik və OpenVPN-nin Certificate Revocation List(CRL)-lə necə quraşdırılmasına baxacağıq.

### İşə başlayaq

2-ci başlıqda istifadə etdiyimiz kimi, client və server sertifikatlarını quraq. Bu misalda FreeBSD9.2 x64 istifadə edilir ancaq, siz Windows və ya Linux OS-dan istifadə edə bilərsiniz.

### Necə edək...

- İlk olaraq sertifikatı generasiya edək:

```
[root@siteA ~]# cd /usr/local/etc/openssl/itvpn/
[root@siteA /usr/local/etc/openssl/itvpn]# bash
[root@siteA /usr/local/etc/openssl/itvpn]# ./clean-all
[root@siteA /usr/local/etc/openssl/itvpn]# source ./vars
[root@siteA /usr/local/etc/openssl/itvpn]# ./build-key-pass
opensslclient4
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'opensslclient4.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:

You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AZ]:AZ
State or Province Name (full name) []:BAKU
Locality Name (eg, city) []:YeniYasamal
Organization Name (eg, company) [Itvpn]:ATL
Organizational Unit Name (eg, section) []:IT
Common Name (eg, your name or your server's hostname) [opensslclient4]:
Name []:
Email Address [openssl-ca@domain.lan]:opensslclient4@domain.lan

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /usr/local/etc/openssl/itvpn/openssl-0.9.8.cnf
Enter pass phrase for /usr/local/etc/openssl/itvpn/keys/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName :PRINTABLE:'AZ'
```

```
stateOrProvinceName :PRINTABLE:'BAKU'
localityName :PRINTABLE:'YeniYasamal'
organizationName :PRINTABLE:'ATL'
organizationalUnitName :PRINTABLE:'IT'
commonName :PRINTABLE:'openvpnclient4'
emailAddress :IA5STRING:'openvpnclient4@domain.lan'
Certificate is to be certified until Nov 6 05:10:45 2016 GMT (1000
days)
Sign the certificate? [y/n]:y
```

```
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

2. Sonra biz dərhal yaratdığımız sertifikatı silək:
 

```
[root@siteA /usr/local/etc/openvpn/itvpn]# ./revoke-full openvpnclient4
Using configuration from /usr/local/etc/openvpn/itvpn/openssl-0.9.8.cnf
Enter pass phrase for /usr/local/etc/openvpn/itvpn/keys/ca.key:
Revoking Certificate 05.
Data Base Updated
Using configuration from /usr/local/etc/openvpn/itvpn/openssl-0.9.8.cnf
Enter pass phrase for /usr/local/etc/openvpn/itvpn/keys/ca.key:
openvpnclient4.crt:
/C=AZ/ST=BAKU/L=YeniYasamal/O=ATL/OU=IT/CN=openvpnclient4/emailAddress=
openvpnclient4@domain.lan
error 23 at 0 depth lookup:certificate revoked
```
3. Öncəki əmrdən sonra CRL listi yenilənəcək. Siz CRL-ə aşağıdakı əmri istifadə edərək baxa bilərsiniz:
 

```
[root@siteA /]# openssl crl -text -noout -in
/usr/local/etc/openvpn/itvpn/keys/crl.pem
Certificate Revocation List (CRL):
 Version 1 (0x0)
 Signature Algorithm: md5WithRSAEncryption
 Issuer: /C=AZ/O=Itvpn/CN=Itvpn CA/emailAddress=openvpn-
ca@domain.lan
 Last Update: Feb 10 05:26:45 2014 GMT
 Next Update: Mar 12 05:26:45 2014 GMT
Revoked Certificates:
 Serial Number: 04
 Revocation Date: Feb 10 05:26:42 2014 GMT
 Serial Number: 05
 Revocation Date: Feb 10 05:22:30 2014 GMT
Signature Algorithm: md5WithRSAEncryption
09:51:bb:50:fa:76:5a:0c:c6:bc:b5:52:62:ae:68:b0:ed:3d:
80:d2:41:4f:58:59:69:8a:7a:b4:b3:13:70:a0:d8:66:b4:df:
f5:f0:f3:4b:70:f9:43:03:92:05:95:85:b7:99:1a:13:24:39:
27:32:bf:74:cb:d4:62:ea:70:2a:af:0c:08:09:59:b3:d7:f4:
70:6c:28:02:d7:0d:c8:38:6d:cd:14:56:6d:4d:79:04:18:2e:
f7:52:80:80:0a:c6:75:cd:3e:06:bd:34:96:5d:cc:1f:b1:79:
89:df:e1:2c:c8:5b:7e:d0:55:26:45:ac:b8:6b:d5:dc:90:a6:
9f:8e:33:2b:a4:ba:36:3e:ae:f8:f2:70:a6:55:26:da:b4:b2:
1a:2e:4c:98:8e:33:84:06:fa:df:a8:31:ac:09:53:c0:42:bf:
ea:c4:e1:f6:f3:9a:15:be:ec:2c:dc:b5:fc:ba:fd:10:d2:b7:
4d:85:24:d7:3f:84:b5:28:ab:8b:07:6c:b7:8a:dc:0d:11:a5:
```



```
7c:18:04:4a:29:d8:2b:2e:42:fa:8f:87:de:ca:76:1c:a1:3f:
70:73:5c:79:9d:b1:98:06:f4:3e:b0:8d:9b:5a:75:c4:f9:93:
fe:d2:47:37:47:e2:09:e4:03:fa:af:a7:4d:4f:d9:a9:6d:fb:
c7:c8:1e:2c:cf:e8:2d:c2:41:80:8a:2d:7e:9f:20:00:b5:fe:
f5:d5:4f:8a:c8:81:3f:e4:3b:dc:19:61:39:7a:0d:1f:9a:7c:
fd:4f:ec:e2:65:6e:19:2e:65:c3:a9:47:4e:99:f3:c1:67:88:
98:71:ce:db:ec:b5:dd:7a:d9:15:60:7c:95:da:3e:a3:50:de:
0e:c2:46:db:56:42:f6:ac:22:29:17:8d:71:cf:ab:45:23:87:
5a:89:8a:99:69:73:03:25:74:4c:b5:c0:5e:e3:6e:6a:84:26:
de:a4:5f:4d:64:c3:50:ad:7a:a2:2f:4c:a8:47:18:59:90:1e:
4d:ca:a2:1a:38:4e:ec:34:27:24:dd:61:da:fc:d2:9c:5d:7f:
7b:4a:a5:9c:8f:f0:e2:27:12:c6:a6:dd:ca:eb:db:6b:08:17:
03:b9:09:da:9d:a1:d9:c1:43:11:7a:7b:67:39:57:0b:9d:50:
ca:2a:f5:26:28:ed:32:6a:a8:05:4d:74:bb:40:f1:ec:9e:1b:
07:64:66:d6:0d:4c:e3:e4:ec:e5:85:91:ad:b6:b6:3b:09:24:
a1:13:5d:50:57:5d:a9:21:a1:a3:64:33:e2:27:30:54:1f:bf:
b2:3a:e5:65:1d:fc:ae:8e:b8:6d:59:75:9a:65:cf:ba:c0:d3:
eb:94:0b:5b:53:88:9f:ff
```

### **Necə işləyir...**

CRL-in daxilində sertifikatın serial rəqəmləri olur hansı ki, revoke edilmişdir. Hər bir serial number CA tərəfindən yalnız bir dəfə verilə bilər. CRL isə CA private key tərəfindən təminat vermək üçün imzalanmışdır.

### **Daha da ətraflı...**

Sual "sertifikatı revoke eləmək üçün nə tələb edilir?" həmişə verilir. Aşağıdakı bölmədə bu haqda daha ətraflı danışılır.

### **Sertifikatı revoke eləmək üçün nə tələb edilir**

Sertifikatı qayda ilə revoke eləmək üçün onun Subject("DN")-i tələb edilir hansı ki, sertifikatın serial rəqəmini açıqlayır. Əgər sertifikat itibəsə, onda onu sadəcə revoke eləmək mümkün olmayacaq. Bu onu göstərir ki, PKI quruluşuna necə diqqətli inzibatçılıq eləmək lazımdır. Hansı ki, həmçinin istənilən istifadəçi üçün yaradılan sertifikatın rezerv nüsxəsi kənar bir yerdə mütləq saxlanılmalıdır.

### **Həmçinin baxın**

- Növbəti başlıqda CRL-lərin istifadə edilməsi
- Bu bölümün son misalında, **Multiple CA's: stacking, -capath** istifadə edilməsi

### **CRL-lərin istifadə edilməsi**

Bu başlıqda biz OpenVPN-nin Certificate Revocation List (CRL) ilə necə quraşdırılmasını göstərəcəyik. Burda öncəki başlıqda istifadə elədiyimiz CRL-

dən istifadə edəcəyik. Bu misal elə 2-ci başlıqda olan Routing Masquerading misalının davamıdır.

## İşə başlayaq

2-ci başlıqda olan IP şəbəkələr üçün Client-Server sertifikatlarını qurun. Öncəki başlığı istifadə edərək CRL-i generasiya edin. Bu misal üçün server maşınımız FreeBSD9.2 x64 və OpenVPN2.3 istifadə edilir. Client maşını isə yenədə FreeBSD9.2 x64 və OpenVPN2.3-də işləyir. Server quraşdırması **basic-udp-server.conf** faylından istifadə edilir hansı ki, 2-ci başlıqda Server-side routing-də istifadə eləmişdik.

## Necə edək...

1. Generasiya elədiyimiz CRL faylını PUBLIC qovluğumuza nüsxələyək.  
[root@siteA /]# **cd /usr/local/etc/openvpn/itvpn**  
[root@siteA /usr/local/etc/openvpn/itvpn]# **cp keys/crl.pem ../**
  2. Server quraşdırma faylı üçün **basic-udp-server.conf**-u **example4-6-server.conf**-a copy edib içinə aşağıdakı sətiri əlavə edin:  
[root@siteA /usr/local/etc/openvpn]# **cp basic-udp-server.conf example4-6-server.conf**  
[root@siteA /usr/local/etc/openvpn]# **echo "crl-verify /usr/local/etc/openvpn/crl.pem" >> example4-6-server.conf**
  3. Sonra serveri işə salın:  
[root@siteA /usr/local/etc/openvpn]# **openvpn --config example4-6-server.conf**
  4. Sonra client quraşdırma faylını yaradaq(**openvpnclient4** üçün bütün yaradılmış faylları öncədən server maşınından bu client-ə **scp** ilə **/usr/local/etc/openvpn** ünvanına nüsxələyin):  
**client**  
**proto udp**  
**remote openvpnsrver.example.com**  
**port 1194**  
**dev tun**  
**nobind**  
**auth-nocache**  
  
**ca /usr/local/etc/openvpn/ca.crt**  
**cert /usr/local/etc/openvpn/openvpnclient4.crt**  
**key /usr/local/etc/openvpn/openvpnclient4.key**  
**tls-auth /usr/local/etc/openvpn/ta.key 1**  
  
**ns-cert-type server**
- Faylı **example4-6-client.conf** adı ilə yadda saxlayın:
5. Sonda clienti işə salın:  
root@siteB:/usr/local/etc/openvpn # **openvpn --config example4-6-client.conf**
- Client qoşula bilməyəcək ancaq server jurnallarında aşağıdakı sətirlər çap ediləcək.

```
[root@siteA /usr/local/etc/openvpn]# tail -f /var/log/openvpn.log
Mon Feb 10 12:36:04 2014 /sbin/ifconfig tun0 192.168.200.1 192.168.200.1 mtu 1500 netmask 255.255.255.0 up
add net 192.168.200.0: gateway 192.168.200.1 fib 0
Mon Feb 10 12:36:04 2014 GID set to nobody
Mon Feb 10 12:36:04 2014 UID set to nobody
Mon Feb 10 12:36:04 2014 UDPv4 link local (bound): [undef]
Mon Feb 10 12:36:04 2014 UDPv4 link remote: [undef]
Mon Feb 10 12:36:04 2014 Initialization Sequence Completed
Mon Feb 10 12:36:10 2014 2.2.2.10:35275 TLS ERROR: BIO read tls_read_plaintext error: error:140890B2:SSL routines:SSL3_GET_CLIENT_CERTIFICATE:no certificate returned
Mon Feb 10 12:36:10 2014 2.2.2.10:35275 TLS Error: TLS object -> incoming plaintext read error
Mon Feb 10 12:36:10 2014 2.2.2.10:35275 TLS Error: TLS handshake failed
```

Burdaki crypt mesajında göstərilir clientə izin verilməyib ki qoşulsun ona görə ki, sertifikat düzgün deyil.

### Bu necə işləyir...

Hər dəfə client OpenVPN serverə qoşulduqda, Certificate Revocation List (CRL) yoxlanış edir ki, görək client öz siyahısında var ya yox. Əgər bu istifadəçinin sertifikatı CRL-də oldusa, onda OpenVPN sadəcə onun girişinə etiraz edir və qoşulma olmur.

### Daha da ətraflı...

CRL-in generasiya edilməsi bir işdir və onun gündəmdə qalması isə digər işdir. Bu çox önəmlidir ki, CRL-i gündəmdə saxlaya bilərsiniz. Bu səbəbdən sizin CRON yazmanız yaxşıdır ki, bu iş gecələr görülsün. OpenVPN-də CRL-in yenilənməsi ilə bağlı bir BUG var hansı ki, client qoşulduqda OpenVPN server çalışır ki, CRL faylına yetki alsın. Əgər fayl yoxdursa və ya fayla çatmaq mümkün deyilsə, onda OpenVPN serverin prosesi səhv qaytararaq kəsilir. Düzgün cavab isə ümumiyyətlə müştərilərə müvəqqəti cavab verməmək idi.

### Həmçinin baxaq

Bu bölümün son misalı, Multiple CAs: **stacking**, **-capath** istifadə edilməsi hansı ki, CA və CRL-in istifadə edilməsini daha da ətraflı başa salır.

### vaxtı bitmiş/revoke edilmiş sertifikatların yoxlanılması

Bu misalın məqsədi ondan ibarətdir ki, OpenSSL CA-nin bəzi daxili imkanlarını açıqlasın. Biz göstərəcəyik ki, sertifikatın status necə dəyişir. "Valid", "Revoked" yada "Expired".

### İşə hazırlaşaq

2-ci başlıqda istifadə elədiyimiz client və server sertifikatlarını istifadə edək. Bu misalı biz həmişəki kimi ☺, FreeBSD9.2 x64 maşında test edirik ancaq, siz Windows/Linux və MacOS-də də eyni işi görə bilərsiniz.

### Necə edək...

1. openssl əmrlərini istifadə etməzdən öncə, təyin ediləcək müəyyən mühit dəyişənləri vardır. Bu dəyişənlər **vars** faylında susmaya görə təyin edilməmişdir.

```
root@siteA:/ # bash
```

```
[root@siteA /]# cd /usr/local/etc/openvpn/itvpn/
[root@siteA /usr/local/etc/openvpn/itvpn]# source ./vars
[root@siteA /usr/local/etc/openvpn/itvpn]# export KEY_CN=dummy
[root@siteA /usr/local/etc/openvpn/itvpn]# export KEY_OU=dummy
[root@siteA /usr/local/etc/openvpn/itvpn]# export KEY_NAME=dummy
[root@siteA /usr/local/etc/openvpn/itvpn]# export
OPENSSL_CONF=/usr/local/etc/openvpn/itvpn/openssl-1.0.0.cnf
```

2. Artıq biz serial number istifadə edərək sertifikatın status-una müraciət yollayıb baxa bilərik:

```
[root@siteA /usr/local/etc/openvpn/itvpn]# cd keys/
[root@siteA /usr/local/etc/openvpn/itvpn/keys]# openssl x509 -serial -
noout -in openvpnserver.crt
serial=01
```

```
[root@siteA /usr/local/etc/openvpn/itvpn/keys]# openssl ca -status 01
Using configuration from /usr/local/etc/openvpn/itvpn/openssl-1.0.0.cnf
01=Valid (V)
```

Bu göstərir ki, bizim OpenVPN serverimizin sertifikatı hələ (**valid**) aktivdir.

3. Öncəki misalımızda **revoke** elədiyimiz sertifikat aşağıdakı kimi görəcəyəmiz.

```
[root@siteA /usr/local/etc/openvpn/itvpn/keys]# openssl x509 -serial -
noout -in openvpnclient4.crt
serial=05
```

```
[root@siteA /usr/local/etc/openvpn/itvpn/keys]# openssl ca -status 05
Using configuration from /usr/local/etc/openvpn/itvpn/openssl-1.0.0.cnf
05=Revoked (R)
```

4. Əgər siz **/usr/local/etc/openvpn/itvpn/keys** ünvanında **index.txt** faylına baxsanız aşağıdakı sətirləri görə bilərsiniz:

```
[root@siteA /usr/local/etc/openvpn/itvpn/keys]# cat index.txt
V 161012040603Z 01 unknown
/C=AZ/O=Itvpn/CN=openvpnserver/emailAddress=openvpn-ca@domain.lan
V 161012040755Z 02 unknown
/C=AZ/O=Itvpn/CN=openvpnclient1/emailAddress=openvpn-ca@domain.lan
V 161012041139Z 03 unknown
/C=AZ/O=Itvpn/CN=openvpnclient2/emailAddress=openvpn-ca@domain.lan
R 161106052039Z 140210052642Z 04 unknown
/C=AZ/ST=BAKU/L=YeniYasamal/O=ATL/OU=IT/CN=openvpnclient3/emailAddress=
openvpnclient3@domain.lan
R 161106052137Z 140210052230Z 05 unknown
/C=AZ/ST=BAKU/L=YeniYasamal/O=ATL/OU=IT/CN=openvpnclient4/emailAddress=
openvpnclient4@domain.lan
```

5. Sonra biz adi mətn redaktoru ilə bu text faylında seçdiyimiz (Məsəl üçün **04** sətirdə olan **openvpnclient3**) Revoke edilmiş bir sertifikatın verilənlərində dəyişiklik edib **R**-i adi **E** ilə dəyişdiririk və **3-cü** sütunda olan rəqəmləri **140210052642Z** silib yerini boş saxlayırıq. Bu

sütün sertifikatın revoke edilmiş vaxt möhürüdür. Həmin sətir aşağıdakı kimi olacaq:

```
E 161106052039Z 04 unknown
/C=AZ/ST=BAKU/L=YeniYasamal/O=ATL/OU=IT/CN=openvpnclient3/emailAddress=
openvpnclient3@domain.lan
```

6. Ardınca da biz statusu yenidən yoxlaya bilərik:

```
[root@siteA /usr/local/etc/openvpn/itvpn/keys]# openssl ca -status 04
Using configuration from /usr/local/etc/openvpn/itvpn/openssl-1.0.0.cnf
04=Expired (E)
```

Əgər biz CRL-i yenidən generasiya eləsək görəəcəyik ki, sertifikat "un-revoked" olmuş görəəcəyik:

```
[root@siteA /usr/local/etc/openvpn/itvpn/keys]# openssl ca -config
../openssl-1.0.0.cnf -gencrl -out crl.pem
Using configuration from ../openssl-1.0.0.cnf
Enter pass phrase for /usr/local/etc/openvpn/itvpn/keys/ca.key:
CA_ŞİFRƏSİ_YAZIRIQ
```

İndi CRL faylımızı analiz eləsək nəticə dediyimiz kimi olacaq:

```
[root@siteA /usr/local/etc/openvpn/itvpn/keys]# openssl crl -text -
noout -in crl.pem | head -15
Certificate Revocation List (CRL):
 Version 1 (0x0)
 Signature Algorithm: md5WithRSAEncryption
 Issuer: /C=AZ/O=Itvpn/CN=Itvpn CA/emailAddress=openvpn-ca@domain.lan
 Last Update: Feb 12 16:34:47 2014 GMT
 Next Update: Mar 14 16:34:47 2014 GMT
```

Revoked Certificates:

```
 Serial Number: 05
 Revocation Date: Feb 10 05:22:30 2014 GMT
 Signature Algorithm: md5WithRSAEncryption
 2a:19:43:18:e6:92:d7:f3:de:8d:cf:00:de:f4:20:d5:74:22:
 9c:b0:d9:9a:d0:a6:5b:80:6f:5e:8d:3e:51:ac:2d:0d:0f:ae:
 19:09:f3:9e:31:37:37:f9:65:09:77:a5:ab:6c:ff:c4:43:58:
 37:4c:50:23:d0:c0:02:ca:68:b6:73:3a:1d:ea:4a:e7:53:68:
 49:49:9d:dd:44:d9:b6:4c:0d:0d:56:3e:2f:f9:b4:0b:13:14:
```

### **Bu necə işləyir...**

OpenSSL **ca** əmri **index.txt** faylını oxuyaraq yeni CRL generasiya edir. **R** statusu ilə olan hər bir sətir CRL-lə əlavə edilmiş olacaq hansı ki, CRL criptoqrafik olaraq CA private açarı istifadə edərək imzalanmışdır.

Revoke edilmiş sertifikatın statusunu **E**-dən **V**-yə dəyişdirməklə biz həmin sertifikatı yenidən **unrevoke** elan etmiş olacağıq.

### **Daha da ətraflı...**

Bu başlıqda biz sertifikatı **Revoked**-dən **Expired**-ə dəyişdik. Bu öncə yaratdığımız client-ə hələdə serverə girişin qarşısını almadı ona görə ki, CRL-də olan həmin client-in statusu hələdə validdir. **index.txt** faylında client-in statusunun **Valid**-dən **Expired**-ə keçirməyin əsas səbəbi ondan ibarətdir ki, biz eyni adlı yeni sertifikatı generasiya edə bilərik.

## Aralıq CA-lar

Bu başlıq bizə aralıq CA-nin necə yaradılması və onun OpenVPN-də istifadə edilməsi üçün quraşdırma qaydasını öyrədəcək. **easy-rsa** scriptlərinin tərkibində həmçinin aralıq CA serverlərinin qurulması üçün funksionallıq mövcuddur. Aralıq CA (Yada alt CA)-nin istifadə edilməsinin üstünlüyü ondan ibarətdir ki, üst-səviyyə olan CA (Həmçinin root CA kimi tanınır) server dahada gizli (qapalı) saxlanıla bilər. Aralıq CA server isə məsul təyin edilmiş şəxslərə Server və Client sertifikatlarının generasiya edilməsi üçün verilə bilər.

## İşə hazırlaşaq...

2-ci başlıqda istifadə elədiyiniz client və server sertifikatlarından istifadə edin. Bu misalımızda server üçün FreeBSD9.2 x64 və OpenVPN2.3, client məşında da həmçinin FreeBSD9.2 x64 OpenVPN2.3 istifadə ediləcək.

## Necə edək...

- İlk olaraq, biz aralıq CA sertifikatını generasiya edəcəyik (Yeni yaratdığımız **IntermediateCA** sertifikatı **root CA** ilə imzalayırıq):

```
root@siteA:~ # cd /usr/local/etc/openvpn/itvpn/
root@siteA:/usr/local/etc/openvpn/itvpn # bash
[root@siteA /usr/local/etc/openvpn/itvpn]# source ./vars
[root@siteA /usr/local/etc/openvpn/itvpn]# ./build-inter IntermediateCA
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'IntermediateCA.key'

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AZ]:AZ
State or Province Name (full name) []:Baku
Locality Name (eg, city) []:YeniYasamal
Organization Name (eg, company) [Itvpn]:OPSO
Organizational Unit Name (eg, section) []:IT
Common Name (eg, your name or your server's hostname) [IntermediateCA]:
Name []:
Email Address [openvpn-ca@domain.lan]:jamal.shahverdiyev@example.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /usr/local/etc/openvpn/itvpn/openssl-0.9.8.cnf
```

```

Enter pass phrase for /usr/local/etc/openvpn/itvpn/keys/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName :PRINTABLE:'AZ'
stateOrProvinceName :PRINTABLE:'Baku'
localityName :PRINTABLE:'YeniYasamal'
organizationName :PRINTABLE:'OPSO'
organizationalUnitName:PRINTABLE:'IT'
commonName :PRINTABLE:'IntermediateCA'
emailAddress :IA5STRING:'jamal.shahverdiyev@example.com'
Certificate is to be certified until Nov 8 20:49:43 2016 GMT (1000 days)
Sign the certificate? [y/n]:y

```

```

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated

```

2. Əmin olmaq üçün yoxlayaq görək bu sertifikat həqiqətən CA server kimi istifadə edilə bilər ya yox:
 

```

[root@siteA /usr/local/etc/openvpn/itvpn]# openssl x509 -text -noout -
in keys/IntermediateCA.crt | grep -C 1 CA
Signature Algorithm: sha1WithRSAEncryption
Issuer: C=AZ, O=Itvpn, CN=Itvpn CA/emailAddress=openvpn-ca@domain.lan
Validity
--
 Not After : Nov 8 20:49:43 2016 GMT
 Subject: C=AZ, ST=Baku, L=YeniYasamal, O=OPSO, OU=IT,
CN=IntermediateCA/emailAddress=jamal.shahverdiyev@example.com
 Subject Public Key Info:
--

keyid:B4:1F:42:8A:B4:C3:9A:B5:3A:CB:C8:D3:91:D0:FD:B6:5F:DC:E6:A4
 DirName:/C=AZ/O=Itvpn/CN=Itvpn CA/emailAddress=openvpn-ca@domain.lan
 serial:BE:84:5E:83:D5:E0:AB:34
--

 X509v3 Basic Constraints:
 CA:TRUE
 Signature Algorithm: sha1WithRSAEncryption

```
3. Artıq biz öz aralıq CA-miz üçün və açarları üçün qovluqları yaradaq(Hal-hazırda yerləşdiyimiz ünvan `/usr/local/etc/openvpn/itvpn-`dur):
 

```

root@siteA:~ # cd /usr/local/etc/openvpn/itvpn/
root@siteA:/usr/local/etc/openvpn/itvpn # bash
[root@siteA /usr/local/etc/openvpn/itvpn]# mkdir -m 700 -p IntermediateCA/keys
[root@siteA /usr/local/etc/openvpn/itvpn]# cp [a-z]* IntermediateCA
cp: keys is a directory (not copied).
[root@siteA /usr/local/etc/openvpn/itvpn]# cd IntermediateCA

```
4. IntermediateCA qovluğunda olan `vars` faylında dəyişiklik edib `EASY_RSA` dəyişənini aşağıdakı kimi edirik:
 

```

export EASY_RSA=/usr/local/etc/openvpn/itvpn/IntermediateCA

```

5. Sonra **vars** faylını yerinə yetirib yeni **keys** qovluğunu işə salırıq (Həmçinin əsas CA sertifikatın sertifikatlar üçün generasiya elədiyi **keys** qovluğunda olan **IntermediateCA** serverin **key** və **crt** fayllarını **IntermediateCA**-ya aid olan **keys** qovluğuna **ca** adı ilə nüsxələyirik):
- ```
[root@siteA /usr/local/etc/openvpn/itvpn/IntermediateCA]# source ./vars
[root@siteA /usr/local/etc/openvpn/itvpn/IntermediateCA]# ./clean-all

[root@siteA /usr/local/etc/openvpn/itvpn/IntermediateCA]# cp
../keys/IntermediateCA.crt keys/ca.crt
[root@siteA /usr/local/etc/openvpn/itvpn/IntermediateCA]# cp
../keys/IntermediateCA.key keys/ca.key
```
6. Artıq biz ilk client-i **ImmediaryCA** ilə yarada bilərik:
- ```
[root@siteA /usr/local/etc/openvpn/itvpn/IntermediateCA]# ./build-key
IntermediateClient
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'IntermediateClient.key'

You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AZ]:AZ
State or Province Name (full name) []:BAKU
Locality Name (eg, city) []:YeniYasamal
Organization Name (eg, company) [Itvpn]:OpSO
Organizational Unit Name (eg, section) []:IT
Common Name (eg, your name or your server's hostname)
[IntermediateClient]:
Name []:
Email Address [openvpn-ca@domain.lan]:intermediateca@example.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from
/usr/local/etc/openvpn/itvpn/IntermediateCA/openssl-0.9.8.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName :PRINTABLE:'AZ'
stateOrProvinceName :PRINTABLE:'BAKU'
localityName :PRINTABLE:'YeniYasamal'
organizationName :PRINTABLE:'OPSO'
organizationalUnitName:PRINTABLE:'IT'
commonName :PRINTABLE:'IntermediateClient'
```



```
emailAddress :IA5STRING:'intermediateca@example.com'
Certificate is to be certified until Nov 9 04:44:41 2016 GMT (1000
days)
Sign the certificate? [y/n]:y
```

```
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

7. Yoxlayaq ki, yeni client sertifikatı **IntermediateCA** tərəfindən yaradılmışdır ya yox:  
[root@siteA /usr/local/etc/openssl/itvpn/IntermediateCA]# **openssl x509 -subject -issuer -noout -in keys/IntermediateClient.crt**  
subject=  
/C=AZ/ST=BAKU/L=YeniYasamal/O=OPSO/OU=IT/CN=IntermediateClient/emailAddress=intermediateca@example.com  
issuer=  
/C=AZ/ST=Baku/L=YeniYasamal/O=OPSO/OU=IT/CN=IntermediateCA/emailAddress=jamal.sahverdiyev@example.com
8. Və sonda biz əmin olmalıyıq ki, sertifikat həqiqətən düzgündür ya yox. Bunu eləmək üçün isə biz root **CA(PUBLIC)** sertifikatı ilə **IntermediateCA** sertifikatını "**stack**"(birləşdirmə) edib bir fayla salmalıyıq.  
[root@siteA /usr/local/etc/openssl/itvpn/IntermediateCA]# **cd /usr/local/etc/openssl/itvpn/**  
[root@siteA /usr/local/etc/openssl/itvpn]# **cat keys/ca.crt IntermediateCA/keys/ca.crt > ca+subca.pem**  
  
[root@siteA /usr/local/etc/openssl/itvpn]# **openssl verify -CAfile ca+subca.pem IntermediateCA/keys/IntermediateClient.crt**  
IntermediateCA/keys/IntermediateClient.crt: OK

### **Bu necə işləyir...**

IntermediateCA sertifikatın Certificate Authority kimi olmaq yetkisi vardır. Bu o deməkdir ki, o özü yeni sertifikatları imzalaya bilər. Bunun üçün də IntermediateCA-nin qovluq strukturu olmalıdır hansı ki, root CA-nin qovluq strukturuna çox bənzəyir. İlk olaraq biz yeni direktoriya strukturunu yaradıırıq və sonra ora lazım olan faylları nüsxələyirik. Bundan sonra biz client sertifikatını yaradıırıq və onun valid olmasını yoxlayırıq. Qayda ilə bu yoxlanışı eləmək üçün, hal-hazırkı sertifikat zənciri root-level CA səviyyəsindən intermediateCA-ya və client sertifikatınadək olmalıdır. Məhz buna görə də root CA public sertifikatı və intermediate CA public sertifikatı bir fayla "**stack**"(birləşmə) edilmişdir. Artıq bu tək fayl vasitəsilə seçdiyimiz sertifikatın zəncir validasiyası yoxlanılır.

### **Daha da ətraflı...**

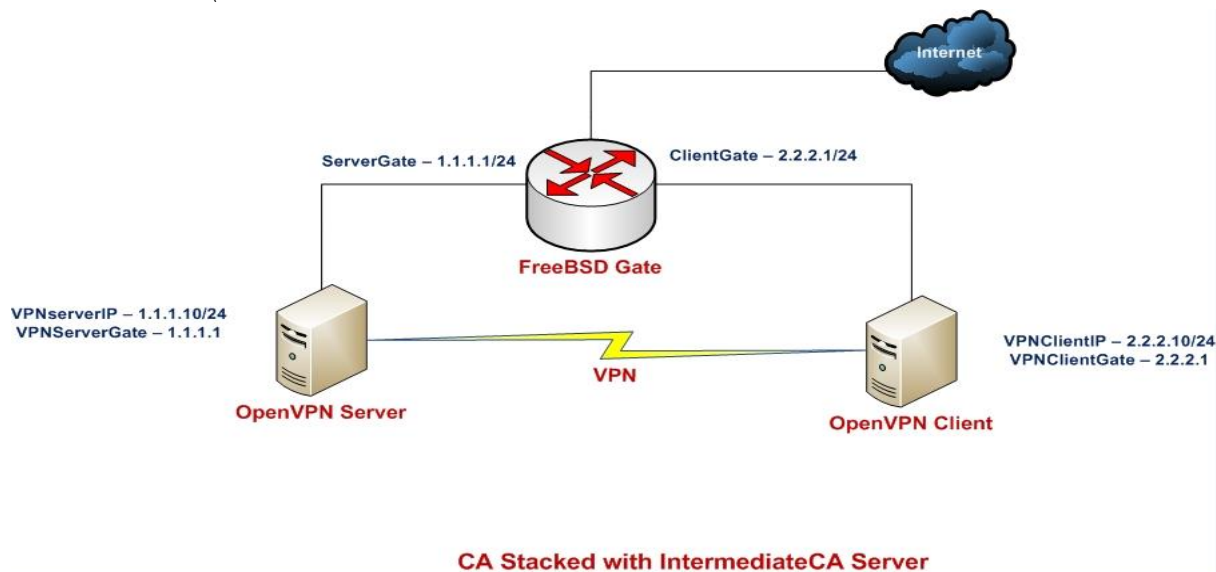
IntermediateCA tərəfindən yaradılmış sertifikatlarda həmçinin eyni CA tərəfindən **revoke** edilmə ehtiyacına malik ola bilər. Bu o deməkdir ki, çoxlu CA-lar üçündə çoxlu CRL-lər istifadə edilməlidir. Xoşbəxtlikdən CRL-lərdə həmçinin öncə CA-da göstərdiyimiz kimi **cat** əmri ilə bir fayla stack edilə bilər.

## Çoxlu CA-lar: --capath istifadə edərək stacking

Bu misalın əsas məğzi OpenVPN-in elə qurulmasıdır ki, harda client sertifikatları var o "client-only" CA tərəfindən imzalanıb və server sertifikatı isə fərqli "server-only" CA sertifikatı tərəfindən imzalanır. Bu operativ təhlükəsizliyin əlavə səviyyəsini açıqlayır hansı ki, bir şəxsə yalnız client sertifikatlarının yaradılmasına izin verilir və digər bir şəxsə yalnız server sertifikatının generasiya edilməsinə izin verilir. Bu o deməkdir ki, client və server sertifikatları heç zaman MITM hücumu məruz qala bilməz.

### İşə hazırlaşaq

2-ci başlıqda istifadə elədiyimiz sertifikatları istifadə edək. Öncəki misalda olan **IntermediateCA** sertifikatını və hansısa bir client sertifikatını istifadə edək. Server maşınımız FreeBSD9.2 x64 və OpenVPN2.3-dədir. Client maşında eynilə FreeBSD9.2 x64 və OpenVPN2.3-dədir.



### Necə edək...

1. Server üçün quraşdırma faylı yaradaq.

```
tls-server
proto udp
port 1194
dev tun
```

```
server 192.168.200.0 255.255.255.0
```

```
ca /usr/local/etc/openvpn/itvpn/ca+subca.pem
cert /usr/local/etc/openvpn/openvpnsrver.crt
key /usr/local/etc/openvpn/openvpnsrver.key
```

```
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0
```

```
persist-key
persist-tun
keepalive 10 60
```

```
user nobody
group nobody
```

```
daemon
log-append /var/log/openvpn.log
```

Faylı **example4-9-server.conf** adında yadda saxlayaq.

2. OpenVPN serveri işə salırıq:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example4-9-server.conf
```

3. Sonra işə client quraşdırma faylını yaradaq(Öncədən IntermediateClient sertifikatlarını server maşından client-ə **scp** ilə nüsxələyək.):

```
root@siteA:/usr/local/etc/openvpn/itvpn/IntermediateCA/keys # scp IntermediateClient* 2.2.2.10:/usr/local/etc/openvpn/
IntermediateClient.crt 100% 5148 5.0KB/s 00:00
IntermediateClient.csr 100% 1070 1.0KB/s 00:00
IntermediateClient.key 100% 1675 1.6KB/s 00:00
```

Ardıncada client quraşdırma faylını yaradaq:

```
client
proto udp
remote openvpnserver.example.com
port 1194
```

```
dev tun
nobind
```

```
ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/IntermediateClient.crt
key /usr/local/etc/openvpn/IntermediateClient.key
tls-auth /usr/local/etc/openvpn/ta.key 1
```

```
ns-cert-type server
```

Faylı **example4-9-client.conf** adında yadda saxlayın. Nəzərə alın ki, bu **ca+subca.pem** faylını client quraşdırmasında təyin etməmişik.

4. Client-i işə salaq.

```
root@siteB:/usr/local/etc/openvpn # openvpn --config example4-9-client.conf
```

5. Serverin jurnal fayllarında baxsanız görəcəksiniz ki, client artıq qoşulma üçün **IntermediateCA** tərəfindən imzalanmış sertifikat ilə qoşulmuşdur:

```
Thu Feb 13 21:37:56 2014 2.2.2.10:53552 [IntermediateClient] Peer Connection Initiated with [AF_INET]2.2.2.10:53552
```

### **Bu necə işləyir...**

Client serverə qoşulduqda, client-in(public) sertifikatı yoxlanış üçün göndərildi serverə. Serverin isə yoxlanışı qayda ilə edə bilməsi üçün tam sertifikat zəncirinə ehtiyacı var hansı ki, bizdə **rootCA** sertifikatı ilə **IntermediateCA** (yada **subCA**) sertifikatını artıq bir yerdə **stack** etmişik. Məhz bu imkan verir ki, client serverə qoşula bilsin.

Əksinə, client qoşulduqda, server(public) sertifikatı həmçinin client-ə göndərilir. Elə serverin sertifikatı original **rootCA** tərəfindən imzalandığına görə, bizim burda tam sertifikat stack-ını göstərməyə ehtiyacımız qalmadı.

**Qeyd:** Əgər biz OpenVPN server quraşdırma faylında **ca+subca.pem** faylının ünvanını göstərməyi unutmuş olsaq, aşağıdakı səhvi görmüş olmalıyıq:  
Thu Feb 13 21:53:39 2014 2.2.2.10:58697 VERIFY ERROR: depth=0,  
error=unable to get local issuer certificate: C=AZ, ST=BAKU,  
L=YeniYasamal, O=OPSO, OU=IT, CN=IntermediateClient,  
emailAddress=intermediateca@example.com

### **Dahada ətraflı...**

CA sertifikatların stack edilməsinin bir hissəsi isə həmçinin elə CRL-lərin stack edilməsi ya da başqa bir mexanizmdir hansı ki, çoxlu CA sertifikatları və onların CRL-lərinin dəstəklənməsi üçün istifadə edilir.

### **CRL-lərin Stack edilməsi**

**/usr/local/etc/openvpn/itvpn/IntermediateCA/openssl-1.0.0.cnf** faylının tələb etdiyi dəyişənləri öncədən təyin edirik, çünki buna görə giləylənir (Bu dəyişənlər **optional** (istəyə bağlıdır) -dır. Yeni elədə önəmli deyil. Ancaq təyin etməsəniz işləməyəcək). Əgər quraşdırma faylında CRL-lər istifadə edilirsə onda, həm **rootCA**-dan və **IntermediateCA**-da olan CRL-lər stack edilməlidir:

```
[root@siteA /usr/local/etc/openvpn/itvpn/IntermediateCA/keys]# bash
[root@siteA /usr/local/etc/openvpn/itvpn/IntermediateCA/keys]# source ../vars
[root@siteA /usr/local/etc/openvpn/itvpn/IntermediateCA]# export KEY_OU=ATL
[root@siteA /usr/local/etc/openvpn/itvpn/IntermediateCA]# export
KEY_CN=atltech.az
[root@siteA /usr/local/etc/openvpn/itvpn/IntermediateCA]# export
KEY_NAME=ATLtech
[root@siteA /usr/local/etc/openvpn/itvpn/IntermediateCA/keys]# openssl ca -
config ../openssl-1.0.0.cnf -gencrl -keyfile ca.key -cert ca.crt -out crl.pem

[root@siteA /]# cd /usr/local/etc/openvpn/itvpn/
[root@siteA /usr/local/etc/openvpn/itvpn]# cat keys/crl.pem
IntermediateCA/keys/crl.pem > crl-stack.pem
```

Sonra isə aşağıdakı sətiri OpenVPN server quraşdırma faylında istifadə edə bilərik:

```
crl-verify /usr/local/etc/openvpn/crl-stack.pem
```

### **--capath direktivinin istifadə edilməsi**

OpenVPN server quraşdırmasında çoxlu CA-lar və CRL-lərin təyin edilməsinin başqa yolu isə aşağıdakı direktivdəki kimidir:

```
capath /etc/openvpn/itvpn/ca-dir
```

Bu direktoriya spesifik ad quruluşuna uyğun olaraq bütün CA və CRL sertifikatlarını özündə saxlamalıdır:

- Bütün CA sertifikatlarının adı öz CA sertifikat **'HASH'**-ləri kimi və sonu **.0** ilə bitməlidir.
- Bütün CRL-lərin adı öz CA sertifikatlarının **'HASH'**-ləri kimi və sonu **.r0** ilə bitməlidir.

Bizim **rootCA** və **IntermediateCA** üçün bu aşağıdakı kimi olacaq:

```
root@siteA:/ # cd /usr/local/etc/openvpn/
root@siteA:/usr/local/etc/openvpn # mkdir ca-dir
root@siteA:/usr/local/etc/openvpn # openssl x509 -hash -noout -in ca.crt
c912cd1e
```

Bu **c912cd1e** rootCA sertifikatının hexadecimal rəqəmdir:

```
root@siteA:/usr/local/etc/openvpn # cp ca.crt ca-dir/c912cd1e.0
root@siteA:/usr/local/etc/openvpn # cp crl.pem ca-dir/c912cd1e.r0
```

Uyğun olaraq eyni işi **IntermediateCA** üçün görürük:

```
root@siteA:/usr/local/etc/openvpn # openssl x509 -hash -noout -in
itvpn/IntermediateCA/keys/ca.crt
bf44dcb2
```

```
root@siteA:/usr/local/etc/openvpn # cp itvpn/IntermediateCA/keys/ca.crt ca-
dir/bf44dcb2.0
root@siteA:/usr/local/etc/openvpn # cp itvpn/IntermediateCA/keys/crl.pem ca-
dir/bf44dcb2.r0
```

Çoxlu fərqli CA sertifikatları və uyğun olan CRL-lərin istifadə edilməsində bu üsul **stack** edilmiş faylların idarəedilməsi üçün çox yaxşıdır.

## BÖLÜM 5

### FreeBSD OS-da OpenVPN üçün bilməli olduqlarımız və OpenVPN-də təcrübə misalları

- ECMP ya da eyni mənsəbə bir neçə marşrut
- ping: sendto: No buffer space available
- FreeBSD OpenSC və PCSC-LITE yuklenmesi
- FreeBSD OS üzərində bir neçə OpenVPN daemon-un eyni vaxtda işə salınması
- OpenVPN şifrələnmiş kanalla AD qeydiyyatı
- Ubuntu 14.04 OpenVPN-in Active Directory ilə inteqrasiyası
- Ubuntu14.04-də OpenVPN üçün FreeRADIUS-la Active-Directory inteqrasiyası
- Ubuntu 14.04 x64 OpenVPN və çoxlu LDAP qrupları

#### **Giriş**

Başlıqda biz OpenVPN-in FreeBSD əməliyyat sistemi üzərində qurulması zamanı qarşımıza çıxan problemlərin həll edilməsi haqqında danışacağıq. Eyni zamanda da gündəlik həyatda OpenVPN-dən tələb edilən vacib quraşdırmaları açıqlayacağıq. Başlıq nisbətən sizə çətin gələcək və məsləhətdir ki, başlığı sadəcə sonadək oxuyub sonradan yenə bu bir də qayıdasınız.

## ECMP yada eyni mənsəbə bir neçə marşrut

**Qeyd:** Mənsəb - Görmək istədiyimiz subnet.

**ECMP (Equal-cost multi-path routing)** - dəyərindən asılı olaraq yolu seçmək. Susmaya görə FreeBSD OS-da görmək istədiyimiz bir mənsəb üçün yalnız bir route yazmaq olar (Linux-dan fərqli olaraq bu funksionallıq çoxdan mövcuddur). Ancaq **Kip Macy** bu imkanı yaratmışdır və təkə indi eyni mənsəb üçün bir neçə route yazmaq yox həmçinin çəki təyin etmək olar yeni qoşulma səviyyəsində balanslaşma.

Bu funksionallıq FreeBSD 8.0-dan başlayaraq işləyir.

Bu imkanı istifadə etmək üçün isə kernel-i aşağıdakı opsiya ilə yenidən yığmaq lazımdır.

```
options RADIX_MPATH
```

Ancaq FreeBSD 9.1 Release-də BUG var idi hansı ki, bəzi marşrutları silmək mümkün deyildi <http://www.freebsd.org/cgi/query-pr.cgi?pr=kern/173477>. Ona görə ehtiyatlı olun.

Bundan sonra isə artıq istənilən marşrutu istifadə etmək olar. Aşağıda bir neçə misal göstərmək olar:

- **bir neçə marşrut əlavə edək**

```
route add -net 192.103.54.0/24 10.10.10.44
add net 192.103.54.0: gateway 10.10.10.44
route add -net 192.103.54.0/24 10.10.10.66
add net 192.103.54.0: gateway 10.10.10.66
```
- **seçilmiş marşrutun silinməsi**

```
route del 192.103.54.0/24 10.10.10.66
del net 192.103.54.0: gateway 10.10.10.66
```
- **seçilmiş marşrut haqqında olan informasiyaya baxaq:**

```
route show 192.103.54.0/24 10.10.10.1
 route to: 192.103.54.0
destination: 192.103.54.0
 mask: 255.255.255.0
gateway: 10.10.10.1
interface: em0
 flags: <UP,GATEWAY,DONE,STATIC>
recvpipe sendpipe ssthresh rtt,msec mtu weight expire
 0 0 0 0 1500 1 0
```
- Marşrut üçün **'weight'** (çəkini) dəyişdiririk. (təyin etdiyiniz marşrut üçün **weight** rəqəmi nə qədər kiçik olsa, o qədər də böyük prioritetli sayılacaq)

```
#route change -weight 10 192.103.54.0/24 10.10.10.1
#route show 192.103.54.0/24 10.10.10.1
```

```

 route to: 192.103.54.0
destination: 192.103.54.0
 mask: 255.255.255.0
 gateway: 10.10.10.1
interface: em0
 flags: <UP,GATEWAY,DONE,STATIC>
recvpipe sendpipe ssthresh rtt,msec mtu weight expire
0 0 0 0 1500 10 0

```

- **balanslaşmanı əlavə edirik**(10.10.10.1 və 10.10.10.88 gatewaylərini qoşulmaların balanslaşması üçün təyin edirik. Bununla belə **weight** parametri ilə balanslaşmanı idarə edə bilərik hansının ki, qiymətləri axının bölgüsünə tərs mütənasib olacaq.

```

route change -sticky 192.103.54.0/24 10.10.10.1
change net 192.103.54.0: gateway 10.10.10.1

route change -sticky 192.103.54.0/24 10.10.10.88
change net 192.103.54.0: gateway 10.10.10.88

route show 192.103.54.0/24 10.10.10.1
route to: 192.103.54.0
destination: 192.103.54.0
 mask: 255.255.255.0
 gateway: 10.10.10.1
interface: em0
 flags: <UP,GATEWAY,DONE,STATIC,STICKY>
recvpipe sendpipe ssthresh rtt,msec mtu weight expire
0 0 0 0 1500 10 0

route show 192.103.54.0/24 10.10.10.88
route to: 192.103.54.0
destination: 192.103.54.0
 mask: 255.255.255.0
 gateway: 10.10.10.88
interface: em0
 flags: <UP,GATEWAY,DONE,STATIC,STICKY>
recvpipe sendpipe ssthresh rtt,msec mtu weight expire
0 0 0 0 1500 1 0

```

Göstərilən misalda trafik **1:10** uyğunluğunda bölünür. Balanslaşmanı **nostick** parametri ilə yığışdırmaq olar.

## **FIB**

FIB haqqında bir neçə söz deyək - çoxlu marşrut cədvəli. Bu bizə imkan yaradır ki, müxtəlif proqram təminatları üçün öz route cədvəllərini istifadə edə bilsinlər. Misal üçün email üçün bir route cədvəl və ya VPN üçün tamam ayrı route cədvəl. Bu imkanın işə düşməsi üçün isə siz kernel-i aşağıdakı opsiya ilə kompilyasiya etməlisiniz.

```
options ROUTETABLES=N
```



**N** - marşrut cədvəlinin sayıdır. Maksimal təyinatı **15**-dir(**N=15**). Beləliklə maksimal istifadə sayı **16** alırıq(sıfırıncı adətən ilk table-dır).

Müxtəlif cədvəllərlə işləmək üçün isə adətən **setfib** əmri istifadə edilir.

Bu funksional 7.0-ci versiyadan başlayaraq işə düşmüşdür.

### **ping: sendto: No buffer space available**

Əgər siz öz serverinizdə belə bir mesajla qarşılaşsanız demək ki, serverə düşən şəbəkə yükünü qiymətləndirməyin vaxtı gəlib çatıb. Buferlərin hal-hazırkı vəziyyətinə aşağıdakı əmr ilə baxa bilərsiniz.

```
root@openvpnserver:~ # netstat -m
811/734/1545 mbufs in use (current/cache/total)
730/304/1034/262144 mbuf clusters in use (current/cache/total/max)
730/294 mbuf+clusters out of packet secondary zone in use (current/cache)
0/9/9/16896 4k (page size) jumbo clusters in use (current/cache/total/max)
0/0/0/8448 9k jumbo clusters in use (current/cache/total/max)
0/0/0/4224 16k jumbo clusters in use (current/cache/total/max)
1662K/827K/2490K bytes allocated to network (current/cache/total)
0/0/0 requests for mbufs denied (mbufs/clusters/mbuf+clusters)
0/0/0 requests for mbufs delayed (mbufs/clusters/mbuf+clusters)
0/0/0 requests for jumbo clusters delayed (4k/9k/16k)
0/0/0 requests for jumbo clusters denied (4k/9k/16k)
0/0/0 sbufs in use (current/peak/max)
0 requests for sbufs denied
0 requests for sbufs delayed
0 requests for I/O initiated by sendfile
0 calls to protocol drain routines
```

Ya da:

```
root@openvpnserver:~ # vmstat -z | grep mbuf
mbuf_packet: 256, 0, 734, 290,83147492, 0, 0
mbuf: 256, 0, 79, 442,13238553, 0, 0
mbuf_cluster: 2048, 262144, 1024, 10, 1024, 0, 0
mbuf_jumbo_page: 4096, 16896, 0, 9, 7, 0, 0
mbuf_jumbo_9k: 9216, 8448, 0, 0, 0, 0, 0
mbuf_jumbo_16k: 16384, 4224, 0, 0, 0, 0, 0
mbuf_ext_refcnt: 4, 0, 0, 0, 0, 0, 0
```

Həmçinin interfeyslərdə olan kolliziya sayına baxa bilərsiniz:

```
root@openvpnserver:~ # netstat -id
```

| Name  | Mtu   | Network    | Address           | Ipkts | Ierrs | Idrop | Opkts | Oerrs | Coll | Drop |
|-------|-------|------------|-------------------|-------|-------|-------|-------|-------|------|------|
| usb0  | 0     | <Link#1>   |                   | 0     | 0     | 0     | 0     | 0     | 0    | 0    |
| em0   | 1500  | <Link#2>   | 00:0c:29:f2:8e:ec | 328   | 0     | 0     | 329   | 0     | 0    | 0    |
| em0   | 1500  | 10.198.0.0 | 10.198.0.10       | 325   | -     | -     | 326   | -     | -    | -    |
| usb0  | 0     | <Link#3>   |                   | 0     | 0     | 0     | 0     | 0     | 0    | 0    |
| plip0 | 1500  | <Link#4>   |                   | 0     | 0     | 0     | 0     | 0     | 0    | 0    |
| lo0   | 16384 | <Link#5>   |                   | 8     | 0     | 0     | 8     | 0     | 0    | 0    |

```
lo0 16384 your-net localhost 5 - - 8 - - -
tun0 1500 <Link#6> 1 0 0 92935003 0 0 8711667
tun0 1500 10.200.0.1/32 10.200.0.1 4 - - 92935002 - - -
```

Qısa formada desək bu nədir? Bu əməliyyatı yerinə yetirmək üçün sistem buferlərinin çatışmazlığıdır. Qaydaya əsasən bu problem adətən pis şəbəkə kartlarında yaranır və həmişə yükəndən asılı olaraq yaranmır. Misal üçün demək olar ki, adi 8 paketlik **ping-i cron-a** yerləşdirib 20 Megabitlik trafikedə 5 dəqiqədən bir yoxlanış edilsə də belə, problem çıxır.

Bundan birdəfəlik canımızı qurtarmaq üçün nə etmək lazımdır? Ən yaxşı yol keyfiyyətli şəbəkə kartı qoymaqdır (Məsələn: **Intel igb**). İkinci variant isə nəticəni **50%** verir. Yəni ki, biz bunun üçün sistemin özünün və kernelin parametrlərində tuning edəcəyik. Aşağıda açıqlayaq:

**-nmbclusters** və buferlərin sayını artırırıq:

```
echo 'kern.ipc.nmbclusters=524288' >> /boot/loader.conf
echo 'kern.ipc.maxsockbuf=1048576' >> /boot/loader.conf
echo 'hw.em.rxd=4096' >> /boot/loader.conf
echo 'hw.em.txd=4096' >> /boot/loader.conf
```

**em** olan yazı şəbəkə kartının adıdır. Əgər sizdə başqa brenddirsə, onda adını dəyişib onu yazın.

### Xəbərdarlıq

64-bitlik sistemlərdə böyük yaddaş olarsa bunu **kern.ipc.nmbclusters=1000000** sayınadək artırmaq olar.

Buferin həcmi artırırıq (aşağıdakı sətirləri '/etc/sysctl.conf' faylına əlavə edirik və sysctl servisi restart edirik):

```
net.inet.tcp.sendbuf_max=16777216
net.inet.tcp.recvbuf_max=16777216
net.inet.tcp.sendbuf_inc=16384
net.inet.tcp.recvbuf_inc=524288
net.inet.tcp.recvbuf_auto=1
net.inet.tcp.sendbuf_auto=1
net.inet.tcp.maxtcptw=102400
```

**-buferlərin mənasını və istifadəçilərin sayını artırırıq.**

Kerneli aşağıdakı parametrlərlə kompilyasiya etmək lazımdır.

```
maxusers 512
options NBUF=4096
```

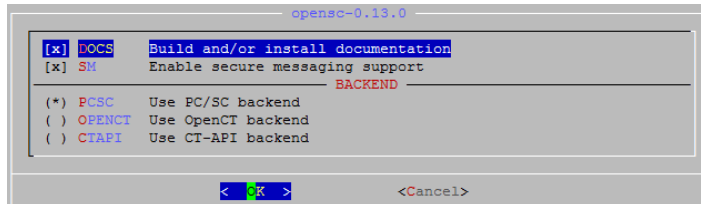
Əgər siz NetGrgraph istifadə edirsinizsə, onda onun mənalərini artırmaq olar:

```
net.graph.maxdgram=524288
net.graph.recvspace=524288
```

Görülən işlərdən sonra heç bir dəyişiklik olmadısa, onda şəbəkə kartını dəyişin.

## FreeBSD OpenSC və PCSC-LITE yüklənməsi

```
root@siteA:~ # cd `whereis opensc | awk '{ print $2 }'` # Portuna daxil oluruq
root@siteA:/usr/ports/security/opensc # make config # Lazımi modulları seçirik
```



```
root@siteA:/usr/ports/security/opensc # make install # Yükləyirik
```

```
root@siteA:/ # cd `whereis pcsc-lite | awk '{ print $2 }'` # Portuna daxil oluruq
root@siteA:/usr/ports/devel/pcsc-lite # make install # Yükləyirik
```

`/etc/devd.conf` quraşdırma faylına aşağıdakı sətirləri əlavə etdikdən sonra, `/etc/rc.d/devd restart` əmri ilə devd servisi restart edirik.

```
attach 100 {
 device-name "ugen[0-9]+";
 action "/usr/local/sbin/pcscd -H";
};
```

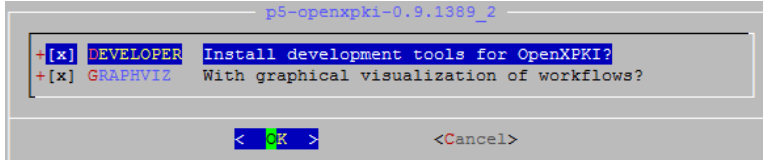
```
detach 100 {
 device-name "ugen[0-9]+";
 action "/usr/local/sbin/pcscd -H";
};
```

```
root@siteA:/ # echo 'pcscd_enable="YES"' >> /etc/rc.conf # Daemon-u
startup-a əlavə edirik
```

```
root@siteA:~ # cp /usr/local/etc/opensc.conf-sample
/usr/local/etc/opensc.conf # Quraşdırma faylını nüsxələyək
```

```
root@siteA:~ # /usr/local/etc/rc.d/pcscd start # Daemon-u işə salırıq
p5-openxpki
p5-openxpki-client
p5-openxpki-client-html-mason
p5-openxpki-client-scep
p5-openxpki-deployment
p5-openxpki-il8n
```

```
root@siteA:/usr/ports # cd `whereis p5-openxpki | awk '{ print $2 }'`
root@siteA:/usr/ports/security/p5-openxpki # make config
```



```
root@siteA:/usr/ports/security/p5-openxpk1 # make install
```

## FreeBSD OS üzərində bir neçə OpenVPN daemon-un eyni vaxtda işə salınması

Öncə məntiqi açıqlayaq.

Deyək ki, sizə lazımdır ki, fərqli istifadəçilər fərqli routing-lərlə fərqli ünvanlara daxil ola bilsinlər. Həmçinin onların eyni Active Directory strukturunda fərqli qruplarda olmalarına tələb olacaq. Ya da lazımdır ki, ayrı-ayrı qrup istifadəçilər fərqli CA serverlər tərəfindən fərqli expire date üçün sertifikatlar əldə etsinlər. Məhz bu hallarda sizin fərqli portlarda və fərqli protokollarla işləyən OpenVPN-in çoxlu instanslarına ehtiyacınız olacaq.

Başlayaq:

Öncədən demək istəyirdim ki, openvpn portlarda **/usr/ports/security/openvpn** ünvanında yerləşir. Yüklənmə bitdikdən sonra openvpn daemon-un control edilməsi üçün **/usr/local/etc/rc.d/openvpn** scripti yaranacaq.

Məhz bu scriptin sayəsində bir neçə daemon işə salmaq olur. Deyək ki, 2 ədəd fərqli quraşdırmada olan openvpn daemon işə salacağıq.

Bunun üçün bir daemon adı **openvpn** və digəri isə **openvpn1** olacaq. Ona görə də ikinci daemon-u birincidən nüsxələyirik:

```
root@atlweb:~ # cp /usr/local/etc/rc.d/openvpn /usr/local/etc/rc.d/openvpn1
```

Sonra isə startup quraşdırma faylınızda hər iki daemon üçün lazımı quraşdırmaları edirik ki, reboot-dan sonra işə düşsün(aşağıdakı sətirləri **/etc/rc.conf** faylına əlavə edirik):

```
OpenVPN Multiple instances
```

```
openvpn_enable="YES"
```

```
openvpn_if="tun"
```

```
openvpn_configfile="/usr/local/etc/openvpn/openvpn.conf"
```

```
openvpn_dir="/usr/local/etc/openvpn"
```

```
openvpn1_enable="YES"
```

```
openvpn1_if="tun"
```

```
openvpn1_configfile="/usr/local/etc/openvpn1/openvpn.conf"
```

```
openvpn1_dir="/usr/local/etc/openvpn1"
```

Gördüyümüz kimi, **openvpn** daemon üçün quraşdırma qovluğu **/usr/local/etc/openvpn** və **openvpn1** daemon üçün isə **/usr/local/etc/openvpn1** olacaq.

Bundan sonra görecəyimiz işlər sırf daemon-ların özlərinə aid olan quraşdırma ardıcılığıdır.

Hər iki daemon üçün quraşdırma qovluğu və quraşdırma faylları yaradaq.

```
root@atlweb:~ # mkdir -m 700 -p /usr/local/etc/openvpn/keys
root@atlweb:~ # mkdir -m 700 -p /usr/local/etc/openvpn1/keys
```

Fayllar:

```
root@atlweb:~ # cat /usr/local/etc/openvpn/openvpn.conf
proto tcp
port 1194
dev tun
server 192.168.200.0 255.255.255.0
```

```
ca /usr/local/etc/openvpn/keys/keys/ca.crt
cert /usr/local/etc/openvpn/keys/keys/openvpnsrver.crt
key /usr/local/etc/openvpn/keys/keys/openvpnsrver.key
dh /usr/local/etc/openvpn/keys/keys/dh2048.pem
tls-auth /usr/local/etc/openvpn/keys/keys/ta.key 0
```

```
persist-key
persist-tun
keepalive 10 60
```

```
push "route 192.168.4.0 255.255.0.0"
topology subnet
```

```
user root
group wheel
```

```
daemon
log-append /var/log/openvpn.log
```

İkinci daemon quraşdırması:

```
root@atlweb:/ # cat /usr/local/etc/openvpn1/openvpn.conf
proto tcp
port 1195
dev tun
server 192.168.202.0 255.255.255.0
```

```
ca /usr/local/etc/openvpn1/keys/keys/ca.crt
cert /usr/local/etc/openvpn1/keys/keys/openvpnserver1.crt
key /usr/local/etc/openvpn1/keys/keys/openvpnserver1.key
dh /usr/local/etc/openvpn1/keys/keys/dh2048.pem
tls-auth /usr/local/etc/openvpn1/keys/keys/ta1.key 0
```

```
persist-key
persist-tun
keepalive 10 60
```

```
push "route 192.168.6.0 255.255.0.0"
topology subnet
```

```
user root
group wheel
```

```
daemon
log-append /var/log/openvpn1.log
```

```
root@atlweb:~ # touch /var/log/openvpn.log ; touch /var/log/openvpn1.log
```

İndi isə CA və sertifikatların yaradılması üçün lazımi hazır scriptləri həm **openvpn/keys** həm də **openvpn1/keys** ünvanına nüsxələyək:

```
root@atlweb:/ # cp -R /usr/local/share/easy-rsa/*
/usr/local/etc/openvpn/keys/
root@atlweb:/ # cp -R /usr/local/share/easy-rsa/*
/usr/local/etc/openvpn1/keys/
```

Hər iki daemona aid olan **keys** qovluğunda olan **vars** faylı aşağıdakı kimi olacaq (İstəyinizə uyğun olaraq dəyişə bilərsiniz).

```
export EASY_RSA=/usr/local/etc/openvpn1/keys
export OPENSSL="openssl"
export KEY_CONFIG=`$EASY_RSA/whichopensslcnf $EASY_RSA`
export KEY_DIR="$EASY_RSA/keys"
export PKCS11_MODULE_PATH="dummy"
export PKCS11_PIN="dummy"
export KEY_SIZE=2048
export CA_EXPIRE=3285
export KEY_EXPIRE=1000
export KEY_COUNTRY="AZ"
export KEY_PROVINCE=
export KEY_CITY=
export KEY_ORG="OpSO"
export KEY_EMAIL="jamal.shahverdiyev@DOMAIN.LAN"
```

Aşağıdakı addımlar hər bir proses üçün nəzərdə tutulmuşdur və mən yalnız **openvpn1** üçün bu işi görəcəm:

```
[root@atlweb /]# cd /usr/local/etc/openvpn1/keys/
root@atlweb:/usr/local/etc/openvpn1/keys # bash
[root@atlweb /usr/local/etc/openvpn1/keys]# source ./vars
[root@atlweb /usr/local/etc/openvpn1/keys]# ./clean-all
```

### CA server yaradırıq

```
[root@atlweb /usr/local/etc/openvpn1/keys]# KEY_SIZE=4096 ./build-ca --pass
Generating a 4096 bit RSA private key
.....++
.....++
writing new private key to 'ca.key'
Enter PEM pass phrase:
```

Verifying - Enter PEM pass phrase:

-----

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [AZ]:

State or Province Name (full name) []:BAKU

Locality Name (eg, city) []:YeniYasamal

Organization Name (eg, company) [OpSO]:

Organizational Unit Name (eg, section) []:IT

Common Name (eg, your name or your server's hostname) [OpSO CA]:openvpnCA1

Name []:

Email Address [jamal.shahverdiyev@DOMAIN.LAN]:

**OpenVPNServer1 üçün sertifikat yaradarıq:**

```
[root@atlweb /usr/local/etc/openvpn/keys]# ./build-key-server openvpnserver1
```

Generating a 2048 bit RSA private key

.....+++

.....+++

writing new private key to 'openvpnserver1.key'

-----

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [AZ]:

State or Province Name (full name) []:BAKU

Locality Name (eg, city) []:YeniYasamal

Organization Name (eg, company) [OpSO]:

Organizational Unit Name (eg, section) []:IT

Common Name (eg, your name or your server's hostname) [openvpnserver1]:

Name []:

Email Address [jamal.shahverdiyev@DOMAIN.LAN]:

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:

An optional company name []:

Using configuration from /usr/local/etc/openvpn/keys/openssl-0.9.8.cnf

Enter pass phrase for /usr/local/etc/openvpn/keys/keys/ca.key:CA\_PASS

Check that the request matches the signature

Signature ok

The Subject's Distinguished Name is as follows

countryName :PRINTABLE:'AZ'

stateOrProvinceName :PRINTABLE:'BAKU'

```
localityName :PRINTABLE:'YeniYasamal'
organizationName :PRINTABLE:'OpSO'
organizationalUnitName:PRINTABLE:'IT'
commonName :PRINTABLE:'openvpnserver1'
emailAddress :IA5STRING:'jamal.shahverdiyev@DOMAIN.LAN'
Certificate is to be certified until Mar 4 17:33:31 2017 GMT (1000 days)
Sign the certificate? [y/n]:y
```

```
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

```
Sonra Diffie Hellman açarını yaradaq(generasiya bir az zaman alacaq):
[root@atlweb /usr/local/etc/openvpn/keys]# ./build-dh
```

```
Sonra TA açarını yaradaq:
[root@atlweb /usr/local/etc/openvpn/keys]# openvpn --genkey --secret
keys/ta.key
```

```
Sertifikatları VPN istifadəçisi yaradaq:
[root@atlweb /usr/local/etc/openvpn1/keys]# ./build-key-pass openvpnclient1-1
Generating a 2048 bit RSA private key
```

```
.....+++
```

```
.....+++
```

```
writing new private key to 'openvpnclient1-1.key'
```

```
Enter PEM pass phrase:Client-Cert-Pass
```

```
Verifying - Enter PEM pass phrase: Client-Cert-Pass-Repeat
```

```

```

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
```

```
What you are about to enter is what is called a Distinguished Name or a DN.
```

```
There are quite a few fields but you can leave some blank
```

```
For some fields there will be a default value,
```

```
If you enter '.', the field will be left blank.
```

```

```

```
Country Name (2 letter code) [AZ]:
```

```
State or Province Name (full name) []:BAKU
```

```
Locality Name (eg, city) []:YeniYasamal
```

```
Organization Name (eg, company) [OpSO]:
```

```
Organizational Unit Name (eg, section) []:IT
```

```
Common Name (eg, your name or your server's hostname) [openvpnclient1-1]:
```

```
Name []:
```

```
Email Address [jamal.shahverdiyev@DOMAIN.LAN]:
```

```
Please enter the following 'extra' attributes
```

```
to be sent with your certificate request
```

```
A challenge password []:
```

```
An optional company name []:
```

```
Using configuration from /usr/local/etc/openvpn/keys/openssl-0.9.8.cnf
```

```
Enter pass phrase for /usr/local/etc/openvpn/keys/keys/ca.key:CA_PASS
```

```
Check that the request matches the signature
```

```
Signature ok
```

```
The Subject's Distinguished Name is as follows
```



```
countryName :PRINTABLE:'AZ'
stateOrProvinceName :PRINTABLE:'BAKU'
localityName :PRINTABLE:'YeniYasamal'
organizationName :PRINTABLE:'OpSO'
organizationalUnitName:PRINTABLE:'IT'
commonName :PRINTABLE:'openvpnclient1-1'
emailAddress :IA5STRING:'jamal.shahverdiyev@DOMAIN.LAN'
Certificate is to be certified until Mar 4 17:37:25 2017 GMT (1000 days)
Sign the certificate? [y/n]:y
```

```
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Sonra **openvpnclient1** üçün quraşdırma faylını hansısa windows maşınının **x64** və ya **x86**-dan fərqli olaraq, **C:\Program Files\OpenVPN\config** ünvanında uyğun adla yaradın və içinə aşağıdakı sətirləri elavə edin(Mənim halımda adı **openvpnclient1-1.ovpn** idi):

```
client
proto tcp
remote vpn.atl.az
port 1195
dev tun
nobind
```

```
key-direction 1
```

```
<ca>
```

```
-----BEGIN CERTIFICATE-----
MIIGoTCCBIImgAwIBAgIJAI4E9wZTW8tVMA0GCSqGSIb3DQEBBQUAMIGRMQswCQYD
VQQGEwJBWJjENMAsGA1UECBMEQkFLVTEUMBIGA1UEBxMLWVWVuaVlhczFtYWwxFDAS
BgNVBAoTC0FUTEluZm9UZWN0MQswCQYDVQQLLEwJJDVEMMAoGA1UEAxMDQ0ExMSww
KgYJKoZIhvcNAQkBFh1qYW1hbC5zaGFodmVyzG15ZXZAYXRsdGVjaC5hejAeFw0x
NDA2MDgxODE5MzdaFw0yMzA2MDYxODE5MzdaMIGRMQswCQYDVQQGEwJBWJjENMAsG
A1UECBMEQkFLVTEUMBIGA1UEBxMLWVWVuaVlhczFtYWwxFDASBgNVBAoTC0FUTElu
Zm9UZWN0MQswCQYDVQQLLEwJJDVEMMAoGA1UEAxMDQ0ExMSwwKgYJKoZIhvcNAQkBF
h1qYW1hbC5zaGFodmVyzG15ZXZAYXRsdGVjaC5hejCCAiIwDQYJKoZIhvcNAQEB
BQADggIPADCCAgocGgIBAPkq3E3j6Ep57rdZrvuTHHEeCXtaai431GqlqL/lSERr
ygfS/krR3HEgWkUrE5TFSnfr90YZ3z3Wrl90ls87iNBxi/Au8Xt40e49SYK7xc1I
1W4hyn3Zw7tHAcFY8/7swfQmQgmLrq7Qxxx8S+MJe5GJNDgZT49bkiF70hj+yI7M
7W2Uh4ywcujna5VKe0k97PzHaAVLCJripnrlkoMH18jkTtP57pTtgu2kv9xs2YzF
1sIJm9VgbEun+mzm0iDrQ8mpwpABGAqka+WaGcDhHiRtFI/aNsGfaWU2jmemAEkT
qsUAvus895N0EHZNe6FKfhElmZy1A6Gg5A/LHoRItvLsGUKhiCefWDzVSbfhmel
Tqk2Buet2frkILQSVIG/pt4JPRln9BbtStSeDFVR6em1WI84VwEgAlxfA8s740Dc
TaLWT45iSd67H3lmafAwjw6mWzHvxATU5JwetVnZhibtXBiEQk5eBKiNNhkswKhVH
STo6mNPNyLb4DQHNv69JsA6ypwLl5cG+YZcXkz26ZlTGhYW7/xqAuGSVrovILK3X
qdRwnm8UDdHCTzpLUPRrPEjhC+iu7IynxhJieeAnavdQRr6BtAUV8zvi7vV2fRW
OkTSJv4Wa9qdCtXdeTyZsXs/tf0P9X+tCJMullQIKKrKfqpFs/I7zJIbZnAgp0r
AgMBAAGjgfkWgfYwHQYDVR0OBBYEFN5R+WwEFO9En1fS90Mp7hlHWjDGMIHGBgNV
HSMEgb4wgbuAFN5R+WwEFO9En1fS90Mp7hlHWjDGoYGXpIGUMIGRMQswCQYDVQQG
EwJBWJjENMAsGA1UECBMEQkFLVTEUMBIGA1UEBxMLWVWVuaVlhczFtYWwxFDASBgNV
BAoTC0FUTEluZm9UZWN0MQswCQYDVQQLLEwJJDVEMMAoGA1UEAxMDQ0ExMSwwKgYJ
KoZIhvcNAQkBFh1qYW1hbC5zaGFodmVyzG15ZXZAYXRsdGVjaC5heoIJA14E9wZT
```

```
W8tVMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEFBQADggIBAK7uU/ ifYdKRr4tM
aYS6CZaUoxbgKK2dvpXSCo0zK2yoWEsXUjI1XMcXqJzCKlyzBpUUo8G3eoLX3z5B
8redkd11TI05RHva/ iwnYmhY+qr14FgVu/ kFEWmZSNDLmNTJ3OrspY07Ui4rRhuM
z9/26Lu+g3ZX4kpGmjEI1xwWynGAdBUBkvY7iTiVEydcNkWKm/ sSUxJlW6ZTMASi
fIQSIqOS083uLAYR01NvFHV9IUWA6mpL77eja19BN0tJUFWryTlywZs1vLnY+Ijj
Zu1L8P7dg5866iZyBd9eJGh/ UILBoUPmF5DnDoHX9h3sEGI/ jTNRf77SWru19B3e
BG/ ksvOV/ as4VntsUIFzaURrhulzi4VJFYAF51z3X3smwMjxNFQn2bXqsY94+9cu
xHB1F1Wm2GOW3mGHXaa6GNjuyLVvV39rNTmFAgeIJsDdOXW2TEBxA3q5upQ7tjQQ
W5HLoIksrhmoIGzvbvtu2kdCoyll3Lm8dKilix75ieplq85HsXzk/ eeAxSPZSmGro
tMRH8bPm6wGtlyAnSnsMKAwq3jgSw+uiXuDJ5JbFjrMgLD+PxRlo1fvGn7HzR9XF
0+EE01RW8C4y/ cQrjWPn5opiERIXgtDM9b32VHjIaGL2H73y3278qWIsiSn2icOn
LaLaF62OT3w+svz5CZaiwR6GUIou
```

-----END CERTIFICATE-----

</ca>

<cert>

-----BEGIN CERTIFICATE-----

```
MIIF9jCCA96gAwIBAgIBAJANBgkqhkiG9w0BAQUFADCBkTELMAkGA1UEBhMCQVox
DTALBgNVBAGTBEBJS1UxFDASBgNVBAcTC1llbmlZYXNhbWFSMRQwEgYDVQKEwtB
VExJbmZvVGvjaDELMAkGA1UECXMCSVQxDDAKBgNVBAMTA0NBMTESMCoGCSqGSIB3
DQEJARYdamFtYWwuc2hhaHhZcmRpeWV2QGf0bHRlY2guYXowHhcNMTQwNjA4MTgy
MjI2WhcNMTcwMzA0MTgyMjI2WjCBnjELMAkGA1UEBhMCQVoxDTALBgNVBAGTBEBJ
S1UxFDASBgNVBAcTC1llbmlZYXNhbWFSMRQwEgYDVQKEwtBVExJbmZvVGvjaDEL
MAkGA1UECXMCSVQxGTAXBgNVBAMTEG9wZW52cG5jbGllbnQxLDEuLmF6MjI2WhcN
MTQwNjA4MTgyMjI2WjCBnjELMAkGA1UEBhMCQVoxDTALBgNVBAGTBEBJS1UxFDAS
BgNVBAcTC1llbmlZYXNhbWFSMRQwEgYDVQKEwtBVExJbmZvVGvjaDELMAkGA1UEC
XMCSVQxGTAUOEAQ8AMIIBCgKCAQEA3JYvHplixKaq+/QSRLkyEN381Sv8PQ5SPO1T
cPc3u2PhMRJ+D57XdIjWdx3uLlKcfewLUSDNnxxcWl9soG/oxR24G5j11ALu8wsY
LimjfsrVvkzBeCYq4ji+Ut/7jkv4rW4QVf6XJsS4WXYGQWVVSb8fyOWxqj3E93+uR
2qErM4TA/ ZvXntOQM6WsVIRQxhsA7t8/Ns6d0YJ0ETbTKaDPukjk8FIwOoqn13mx
ss/ Gq2QHxWe7hVpGgYP2NLareC6bo1E+9y7Lcz2PXESK666AizGur6ct4fIyxrEh
ock69sItBG95+LCbZn6Kqk8B+no6xjD+zhCR7Y/ WsrcmHn4FfQIDAQABo4IBSDCC
AUQwCQYDVR0TBAlwADAtBglgkghkgBhvhaCAQ0EIBYerWFzeS1SU0EgR2VuZXJhdGVk
IENlcnRpZmljYXRlMB0GA1UdDgQWBbT5mmsS7D1uBZ8OwgC535hN+iMN1zCBxgYD
VR0jBIG+MIG7gBTeUflsBBTvRJ9X0vTpj+4ZR1owxqGB16SB1DCBkTELMAkGA1UE
BhMCQVoxDTALBgNVBAGTBEBJS1UxFDASBgNVBAcTC1llbmlZYXNhbWFSMRQwEgYD
VQKEwtBVExJbmZvVGvjaDELMAkGA1UECXMCSVQxDDAKBgNVBAMTA0NBMTESMCoG
CSqGSIB3DQEJARYdamFtYWwuc2hhaHhZcmRpeWV2QGf0bHRlY2guYXowCCQCOBPCG
U1vLVTATBgNVHSUEDDAKBggrBgEFBQcDAjALBgNVHQ8EBAMCB4AwDQYJKoZIhvcN
AQEFBQADggIBAI2X6noXm2QTY3pG7R8e3ccGykNmihfM0qxrGzynefuVYy24BRwY
beh/ ix88sczJEN8RH+ERHCUN3wSznU1PBajmdpIQIq8jovtY5oW+1XkkhT681mle
IXolPdaNu1UrAv2S/7XIouLBM+ /xNjFJ5GdMcB6jS9oQ3mUbxNRqrAeuOtrDq7Sh
Vjjc0qHerT0D39YrhWhiDvz0X+7+pcV+EJzIpmHcbilrQtFRni/ x0mHZNv3cY2ri
D4f+8CY+s2HS/Op6zoJ4Tytd7PQNmFri1StE5RedtwRKNDsqz52hQ0VMuuiBIXFq
c15ZVG8BtcUn42oLzSnb+Hb15QOoe5W/ RDH39NpbvP/ cNPGzrPIInKMDmZ/ 94Kqz
5Bfj8q1U8ECxRGjAdQlG9YKr8c8x4yxmXtPiJ5vnuIZBrd+XMxGdfhxLLrCxgKo6
0dKKtnVlMgeMPIK7eZaKTQ01nqblLZJ9jUS2LbmNvSdD9J7ft998Jp+ENaAVkpcZ
DjBV3AjfyENbuTJzt/ +857CQ86LbDrUd+3vPcZSUjgIAYeuv6uq1MUHwJN2L2C
9foh+wMyyMIKrmur7fgou2AIY8pN0+35MMYSzuzOtyQzvmMs41qptZzlw4eaWCz
SGe6WKYQeqwEIj6BhY74XM8vln6k0mmTNei66k2Bm4ijCozwWdyIBuEs
```

-----END CERTIFICATE-----

</cert>

<key>

-----BEGIN RSA PRIVATE KEY-----

```
Proc-Type: 4, ENCRYPTED
DEK-Info: DES-EDE3-CBC, 46235181079E450C
```

```
T2CABn4zgQqj0d4fkrVbzAGXm/bKnEJTrm+MjLuk7SY31DFwA66LBT4cm9IgtY37
Yf/icse4o6mCE4Ootp+25oaqdlw9KtnlagsxAVlP38/opcDCZNqlQRBKqSbY9h+o
vqa64kO4GoeMLAMySmJPimcPDqE2La6KAfWTzESZ7gBXaaugW7dNSjdoBuk3gA2j
WXspD6R4KjclU4T5IfYS1Ylq/blxUfo07wW2mrwL9SGYwiYp51rwXj0eKEDTvjP
lzaCG/HIgzM761wqmGMzkHKhhVVqLkGae0MEN6DX2Z5kB89YZJ9UfuzmkBWl/IE
Y+8YY6Uv5NYyVSokl1rtAHUBH6f4RVrt3vgclICitBtZtpBzrbZpyQJUybAM/xlxY
uun2V37C3ntsHL3jqs8AVnSbyyPKoQ8uuHISTWerL1tCaTQh9GrWooLvsm0X1FBD
wrWtVvHJX3Ge32YwzNihpV90IHUV8mFcLQJlLcB8qa5cKMYdnpJFjvSgcVSlcbhm
R9ynQqP0y/mo4ngmRgbsUZTcDJQy3PNC7DB6BEVTxhnRXEX8YJif6oqPh6vdI8rj
81FJg2zyuyzcp1FAL9zv78XvIak9QJ9HBXDZNYwWfWOHh8Wx5j7JEZhY15jSyn
yC+4nnaVmHb3Ld4UtuCIYulQ9c2YaJ4DgNazth5kfzCARhJV7E50C0hnoFKHyy/
4+4aEvKOBWUMfJBdr/5SbGYVrNTU0rsIwnPWOL9OEXD85IRsD3Zn5J2lcy+Sh5F3
2on3xnJcsf7oV6F+nX5KGKkSFwjaP5T0qBj49LtKht9j1v7X88cr7wEkWBQHFId
EGZI/hfCcku5emthiZlE7sTzKyrWUwB5P+g748TMnZK0IJBhsmSI+6T8a7thumVf
u7J/sq7jUDISHap+iDaDrqatlXSzbixANCu67katiMaOf9T4WhB3Gj52Xd40eynn
8f0HaP+yip+3qMmk6DCEf7uBgdrQNCRREqmg/gdmc6G87JWSY/6HXr9I8JVbdSg1
GfCNFNnCcZalklqFu2maYMW0q30cwavKfrFel3IpED2QQVz8vyuu4q5zEesE550li
vuyyGhjaZaWCxhtSvUyO+N6H1+dXiRjxtQYbLeEaDaleEYRNYplQGTTP9I0IgdD6
nHEkavpIQKAgCyu/ExDWX4ID+f3whC5/+lamsS9pGMC1081Bq/92y7vymUgwimbO
rzQerTXVds35v/S65pGQZoLfYzOTquBJHNSz6DVTQpSjEKpx2f6XUa1IVZKETppx
ub7i2hh9wvktEuEWUmsjmvTRfBOfwAnxij1vY/OymFr9i6uo8USQ0vhm4zQ1rhL
S69E78Knhydp2av0p990BNHpWYVcTWUPHqbYXPDJlNs+k8HjAgOKHo92LMGls47A
fjdDheQiR9+w9R/zn/jHvmjP6+LLJkFO10VNtoILL19p1MQPCjCitKVP4+7uDvM+
FSaqeMKrIoBt0bQq1HbjILblZYk658hc42iBc/KM4aTkVrR5I/GsN3nZxyXCwcf
utlnVFXi8rGMRfsuUIr00Vjo60s3bSyS808rVYvxwQQocsp3nvfJTLimANN55/lp
-----END RSA PRIVATE KEY-----
</key>
```

```
<tls-auth>
-----BEGIN OpenVPN Static key V1-----
365a88d1ef263eba18e2c9cacb851fbd
97b011ab9b80a780c1e8f721287b2689
b4025a08e20e1224526f2a88a2c566c6
5131e2cbda49df030e1367d3fc2cd272
0646a995a5b2a492d0c92420eb17adb0
eaad45547daf33da72f7fdcf25e65ce3
b04512a10d33173834f14a3c121a2a99
8aeced3c4fd243c9e19f5b9da4d82159
a7388e5fa4064187eb5dde2032365544
f6032dd8bbeec3553e3c56c61eaf0f93
e1b21449fc18912770ef7361e82f4de5
ed73f9f252fc6f803d319986bf03c43d
eb515fd9790b3739736b734bea11413b
49692fbbf197b7e396e87fc2c60d3ef8
014636573468a2b773cf011b8116e61e
ce0d61f142d5e665b07cf3df7b3f74a7
-----END OpenVPN Static key V1-----
</tls-auth>
```

#### ns-cert-type server

Göstərilən tərkibdə sertifikatlar haqqında bir az açıqlama verək:  
**ca.crt**, **openvpnclient1-1.crt**, **openvpnclient1-1.key** və **ta.key** fayllarının  
tərkibini quraşdırma faylına əlavə edin. Yəni aşağıdakı kimi:

```
<ca>
```

```
-----BEGIN CERTIFICATE-----
ca.crt sertifikatının base64 formatında olan tərkibi bura əlavə edin
-----END CERTIFICATE-----
</ca>

<cert>
-----BEGIN CERTIFICATE-----
openvpnclient1.crt sertifikatının base64 formatında olan tərkibi bura
əlavə edin
-----END CERTIFICATE-----
</cert>

<key>
-----BEGIN PRIVATE KEY-----
openvpnclient1.key keyinin base64 formatında olan tərkibi bura əlavə
edin
-----END PRIVATE KEY-----
</key>

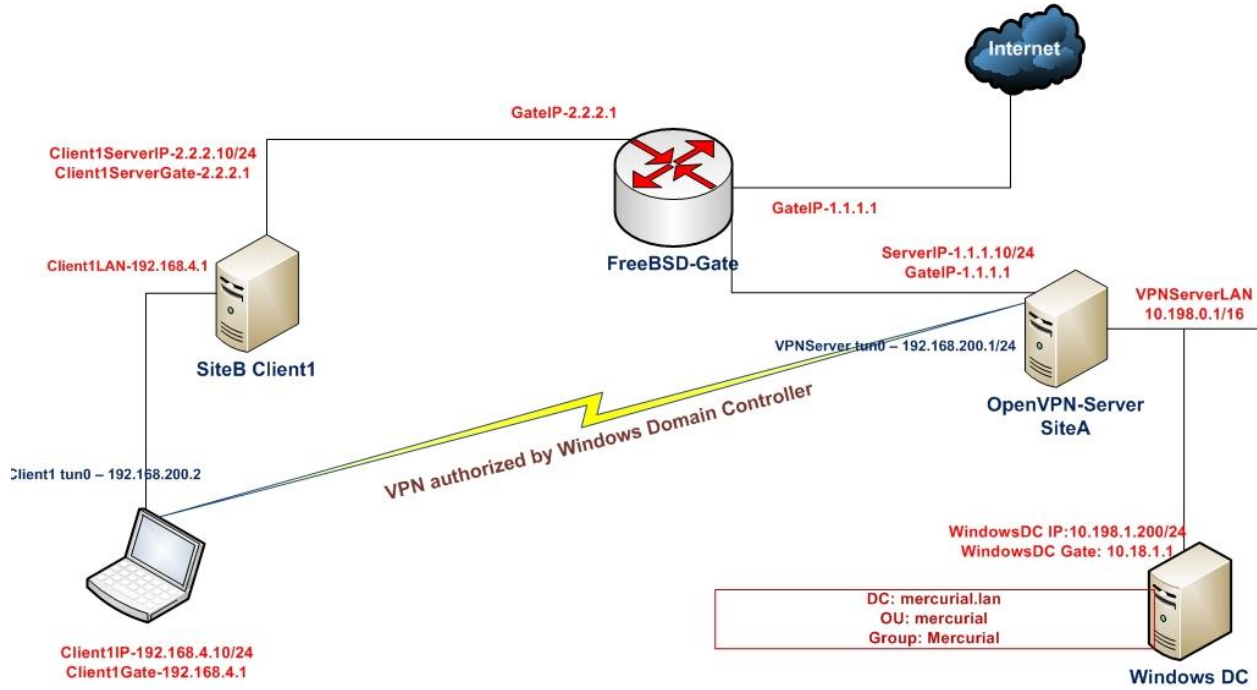
<tls-auth>
-----BEGIN OpenVPN Static key V1-----
ta.key-in tərkibini bura əlavə edin.
-----END OpenVPN Static key V1-----
</tls-auth>
```

Sonda isə qoşuluruq.

### **OpenVPN şifrələnmiş kanalla AD qeydiyyatı**

Bu başlıqda biz OpenVPN-i Windows Domain Controller ilə inteqrasiya edəcəyik. Ancaq hər bir halda client və server arasında olan yol generasiya elədiyimiz CA açarıyla yoxlanacaq və openvpnserver açarı ilə şifrələnəcək.

Aşağıdaki şəbəkə quruluşundan istifadə edəcəyik:

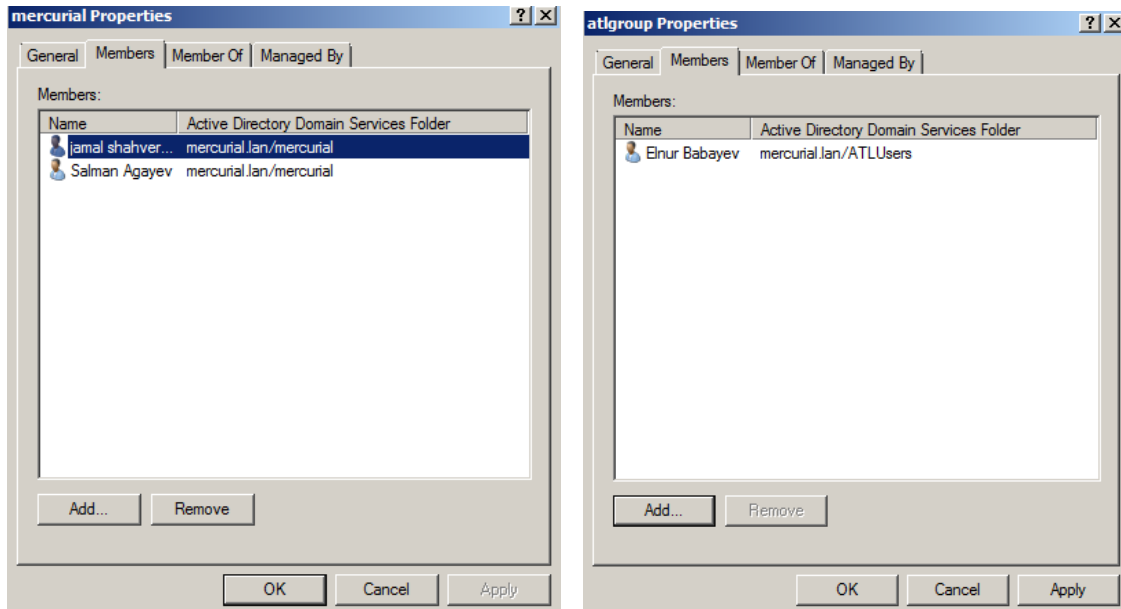


Bu misalımızda 2-ci başlıqda generasiya elədiyimiz CA və server sertifikatlarını həm server həm də client üçün istifadə edəcəyik. Server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Client maşını isə Windows7 x64 OpenVPN2.3-də olacaq. Həmçinin OpenVPN serverimizin daxili şəbəkəsinə qoşulmuş Windows 2008 server Domain Controller olacaq.

Domain Controller aşağıdakı verilənlərdən ibarətdir:

DC: **mercurial.lan**  
 OU: **mercurial**  
 Group: **mercurial**  
 Test user: **jamal**

Domain controller maşında **mercurial** adlı qrupun içində **jamal** adlı istifadəçi mövcuddur ki, sınaqlarımızı edə bilək. Həmçinin **Users** qrupunun içində **elnur** adlı istifadəçi mövcuddur ki, giriş edə bilməyən istifadəçi kimi onunla sınaq edək.

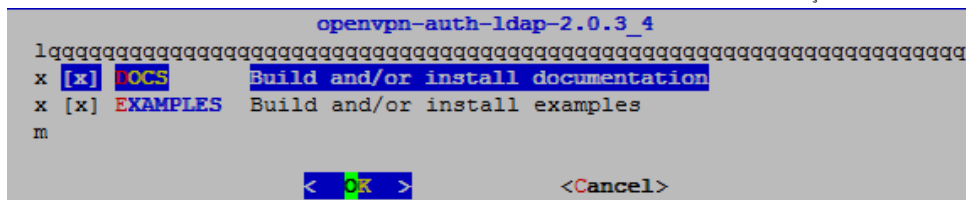


## İşə başlayaq:

1. Öncə server maşınımıza lazımi paketləri yükləyək:

```
root@siteA:/usr/local/etc/openvpn # cd /usr/ports/security/openvpn-auth-ldap/
```

```
root@siteA:/usr/ports/security/openvpn-auth-ldap # make config #
Lazımi modulları
seçirik
```



```
root@siteA:/usr/ports/security/openvpn-auth-ldap # make -DBATCH install
Yükləyirik
```

2. Auth-LDAP paketi serverə yükləndikdən sonra o `/usr/local/lib/openvpn-auth-ldap.so` ünvanına öz pluginini əlavə edir. Biz məhz bu plugini AD-yə qoşulmaq üçün istifadə edəcəyik. `/usr/local/etc/openvpn/ad-auth.conf` adlı server quraşdırma faylını yaradaq və tərkibinə aşağıdakı sətirləri əlavə edək:

```
plugin /usr/local/lib/openvpn-auth-ldap.so
"/usr/local/etc/openvpn/openvpn-auth-ldap.conf"
proto udp
port 1194
dev tun
server 192.168.200.0 255.255.255.0

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnsrvr.crt
```

```
key /usr/local/etc/openvpn/openvpnsrver.key
client-cert-not-required
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0
```

```
persist-key
persist-tun
keepalive 10 60
```

```
push "route 10.198.0.0 255.255.0.0"
topology subnet
```

```
user nobody
group nobody
```

```
daemon
log-append /var/log/openvpn.log
verb 5
```

Domain Controller-ə qoşulmaq üçün `/usr/local/etc/openvpn/openvpn-auth-ldap.conf` quraşdırma faylının tərkibi aşağıdakı kimi olacaq:

```
<LDAP>
 URL ldap://10.198.1.200
 BindDN Administrator@mercurial.lan
 Password B123456789b
 Timeout 15
</LDAP>
<Authorization>
 BaseDN "DC=mercurial,DC=lan"
 SearchFilter "(&(sAMAccountName=%u)(memberOf=CN=mercurial,OU=mercurial,DC=mercurial,DC=lan))"
</Authorization>
```

Həmçinin unutmayın ki, OpenVPN server-də `/usr/local/etc/openvpn/openvpn-auth-ldap.conf` faylın içində olan Domain adının resolve edilməsi üçün `/etc/resolve.conf` faylına aşağıdakı sətir əlavə edilmişdir.

```
nameserver 10.198.1.200
```

- OpenVPN Serveri işə salaq:
 

```
root@siteA:/usr/local/etc/openvpn # openvpn --config ad-auth.conf
```
- İndi isə Windows7 maşında client quraşdırma faylını yaradaq. `C:\Program Files\OpenVPN\config` ünvanında `ad-udp-client.ovpn` adlı fayl yaradaq və tərkibinə aşağıdakı məzmunu əlavə edək:
 

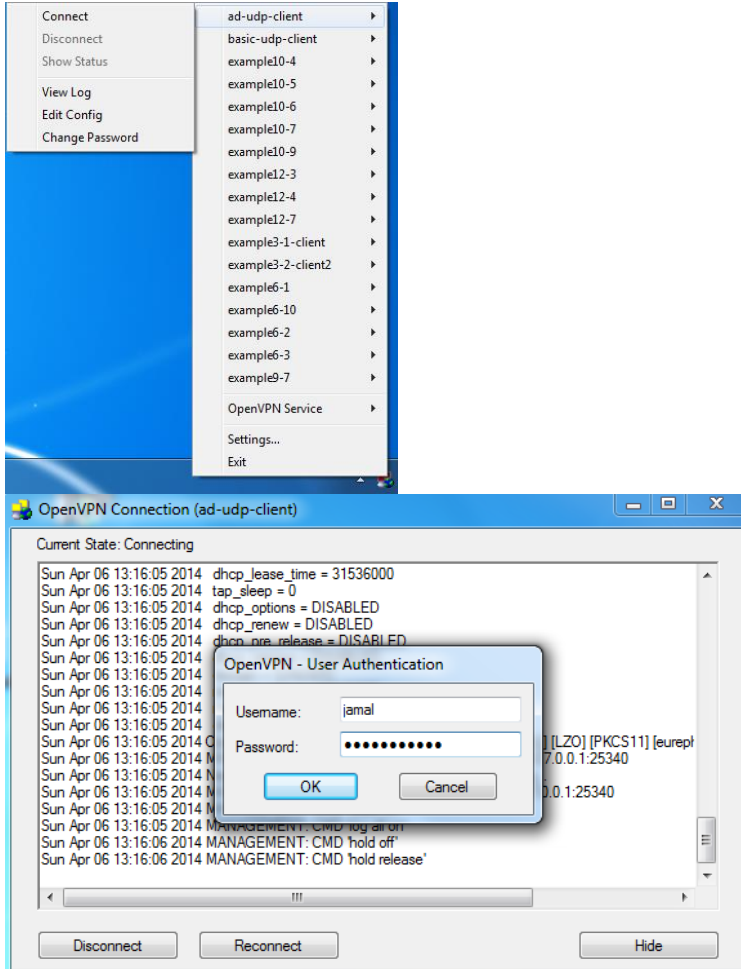
```
client
auth-user-pass
proto udp
remote openvpnsrver.example.com
port 1194
dev tun
```

**nohbind**

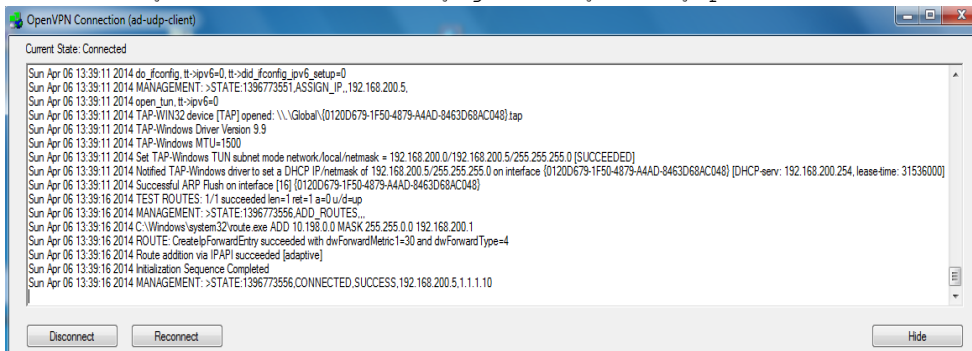
```
ca "c:/program files/openvpn/config/ca.crt"
tls-auth "c:/program files/openvpn/config/ta.key" 1
```

```
ns-cert-type server
verb 5
```

5. Windows7 client maşını işə salaq:



Client maşının statusunda aşağıdakı şəkil çap edilməlidir:





6. Server maşında `/var/log/openvpn.log` jurnal faylına baxıb aşağıdakı sətirləri görməliyik:

```
Sun Apr 6 13:17:43 2014 us=626543 2.2.2.10:53829 PLUGIN_CALL: POST
/usr/local/lib/openvpn-auth-ldap.so/PLUGIN_AUTH_USER_PASS_VERIFY
status=0
```

```
Sun Apr 6 13:17:43 2014 us=626715 2.2.2.10:53829 TLS:
Username/Password authentication succeeded for username 'jamal'
```

```
Sun Apr 6 13:17:43 2014 us=627135 2.2.2.10:53829 Data Channel Encrypt:
Cipher 'BF-CBC' initialized with 128 bit key
```

```
Sun Apr 6 13:17:43 2014 us=627163 2.2.2.10:53829 Data Channel Encrypt:
Using 160 bit message hash 'SHA1' for HMAC authentication
```

```
Sun Apr 6 13:17:43 2014 us=627235 2.2.2.10:53829 Data Channel Decrypt:
Cipher 'BF-CBC' initialized with 128 bit key
```

```
Sun Apr 6 13:17:43 2014 us=627282 2.2.2.10:53829 Data Channel Decrypt:
Using 160 bit message hash 'SHA1' for HMAC authentication
```

**Qeyd:** Nəzərə alın ki, OpenVPN server ilk dəfə işə düşəndə, yolun şifrələnməsindən sonra client ilk dəfə qoşulanda, qoşulmaya bilər. Ancaq bundan sonra bütün qoşulmalarda problemsiz işləyəcək.

Əgər OpenVPN serveri startup-a əlavə etmək istəsəniz, sadəcə aşağıdakı sətirləri `/etc/rc.conf` faylına əlavə etməyiniz yetər:

```
openvpn_enable="YES"
openvpn_if="tun"
openvpn_configfile="/usr/local/etc/openvpn/ad-auth.conf "
openvpn_dir="/usr/local/etc/openvpn"
```

## **Ubuntu 14.04 OpenVPN-in Active Directory ilə inteqrasiyası**

Məqsədimiz Ubuntu 14.04 server üzərində OpenVPN yükləyib Active Directory ilə əlaqələndirməkdir çünki, VPN istifadəçi bazasını AD-dən almaq istəyirik. İstifadə edilən OS-lar:

**Windows 2012 Server R2** - DC  
**Windows 8.1 x64** - Client maşını  
**Ubuntu 14.04 x64** - OpenVPN

Qabaqcadan istifadə etdiyimiz Domain Controller-in verilənlərini açıqlayaq  
Domain Controller: **DOMAIN.LAN**  
DC RO User: **ADMINISTRATOR**  
DC RO PASS: **DOMAIN\_ADMIN\_PASS**  
DC VPN Group: **OpenVPNFAUsers** - Tam yetkisi olan VPN istifadəçiləri (Bütün şəbəkəyə routing olacaq)

Windows 8.1 client quraşdırma faylı aşağıdakı kimi olacaq (faylımızın adı **DOMAIN-ad-auth.ovpn**). Faylın genişlənməsi mütləq **.ovpn** olmalıdır:

```
client
auth-user-pass
auth-nocache
reneg-sec 86400
proto tcp
remote ovpn.dc.DOMAIN.az
port 1194
dev tun
nobind
```

```
key-direction 1
```

```
ns-cert-type server
```

```
OpenVPN serverdə yaradılan ca.crt
```

```
<ca>
-----BEGIN CERTIFICATE-----
MIIGxDCCBKygAwIBAgIJALsV/eQc/V5+MA0GCSqGSIb3DQEBBQUAMIGcMQswCQYD
VQQGEwJBWjENMAsGA1UECBMEQkFLVTEUMBIGA1UEBxMLWVWVuaVlhc2FtYWwxFDAS
BgNVBAoTC0FUTEluZm9UZWN0MQswCQYDVQQLLEwJJVDEXMBUGA1UEAxMOQVRMSW5m
b1RlY2ggQ0ExLDAqBgkqhkiG9w0BCQEWHWphbWFsLnNoYWh2ZXJkaXl1dkBhdGx0
ZWN0LmF6MjE0MDYwODE3NDUzOFoXDTIzMDYwNjE3NDUzOFowZGZzZGZzZGZzZGZz
BAYTAKFaMQ0wCwYDVQQIEwRCQUtVMRQwEgYDVQQHEwtZGZzZGZzZGZzZGZzZGZzZGZz
A1UEChMLQVRMSW5mb1RlY2ggQ0ExLDAqBgkqhkiG9w0BCQwAIAIBAgIBAgIBAgIBAg
VGVjaCBDQTEsMCoGCSqGSIb3DQEJARYdamFtYWwuc2hhaHZlcmRpeWV2QGf0bHRl
Y2guYXowggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQC0kYn6jZf/R1eA
Xs1YH/g36sIQJcxJBmcbXh/atZTy7W8rlXsCw05+RU7OaXrFQUEbed0lnjYiKfri
CutMpT5c7iY6fgfMMoPaIqk8q17qydk8HvqQoac3kjo9wMD7XWlDYiLFk1FkQjEW
BIqI2z6vh9/9ka54s6WNRgzT+7+EzqSuwCfc6Dm/0qxp4AvEjapwjaURJ6yEuQYe
Odh5ydTsIcueNnBzkuFZRx505iNcaBQZ2fUVpQvuetCCsHkPt1BGU3TqWIYTUVZl
04wPQoOyXC9YUvWaYWSLTDMDNdVcVGFYfc5C3++nijtfWpO8LLDZgiwC7ScYj+Boo
SZ9dkEpIYdb03KBnn+LCO3STVukpwTr+vyKjPITceuelHXDwvXi7wgtopwQhQ+3j
sDCvB+Wg2Bt5zBPC43WTeLANOGZFQN1f1kyBNXlBm1tM0kl3k75skkj9TXHjrM44
+aVdxlPjkQ86e6/A04wCUoBnf4a00Q8r6PwCfPkqatDn6hCh6ChAYYuqAR5W3eRs
p2D31AWGEHlB1f/+397E66f3ByHvPGQ5n1AQ3wI7q+tLH+qPsoFUKcyfEbctuYvG
D0+9jPhvAAQwc4hBhn+TXRXPKaaaI89iiaJoiF1//R8kqs8t3yxpjEy0hs2nrx
```

```
tboZl91cO2fj8e2HvhbMs9v+j6oVTQIDAQABo4IBBTCCAQEwHQYDVR0OBByEFiCi
KzboRhxacra8qkU+xvRM4df7MIHRBgNVHSMegckwgcaAFiCiKzboRhxacra8qkU+
xvRM4df7oYGipIGfMIGcMQswCQYDVQQGEwJWwJENMAsGA1UECBMEQkFLVTEUMBIG
A1UEBxMLWwVuaVlhc2FtYWwxFDASBgNVBAoTC0FUTEluZm9UZWNOMQswCQYDVQQQL
EwJJVDEEXMBUGA1UEAxMOQVRMSW5mb1RlY2ggQ0ExLDAqBggkqhkiG9w0BCQEWHWph
bWFsLnNoYWh2ZXJkaXllckBhdGx0ZWNoLmF6ggkAuxX95Bz9Xn4wDAYDVR0TBAUw
AwEB/zANBgkqhkiG9w0BAQUFAAOCAgEAT+K70oaUfXDEfSFmBTrppvbcGqoVsaE1
5NjMh206D5KWtErhKbP7id20sdt6Ygq1PQWW3I3thVQ0L686rhz/cr6Vzj41cFI
EqCt4uqZrkoMcvPq82PONvrzKCauxv5kmZJhWQTB3WXMo0A4KnQqW6/HVzSmbQgC
QR6CqNt1Z21a1RIQR1CmqRankKC4yQBKbzDwB1XLHvjITdyhJlHXZxBcdXurMX
U7AsHOTxbHy4nbyB+ZzlnO37wza6FBeunIqJ/I5eKDCn1lyGELjDsEvrSUcbRRg
IenV9/D9LP4y2KghMkiuDn7vhY3IifCjxQg3JWia5BdQ/lU1Accsxi0/nyQtZF7
5NadlwoSOjEe2H6bwxhlnGcItQyic34HghNKUF16eYLlMEzGkP7UNLwQN32b3IiA
q9+HTP6TQoci43AoaA3NFaUjuKC3zHykesNS8QqOH7MVB4L38/piaGD/K8CsiZH+
QhKICaJJ7hx/Cfp3VUIKr9yxtAnC5QNbXr9QVCC+mwi/sH9laThPlm1Xd2tKdoZa
My/K6o5fZnZSpzOeFa9j6bRgF2tpbG3jxiWT00F9xUv5EtXZdfies5BRHa1FYGK4
yvVIA/ZJBSB/6CT8mnMjGJcn85CcRggOrOc7lQNmgFKw/YopPYAKzjgilEKtNm3
pmPKIhXPdvc=
```

-----END CERTIFICATE-----

</ca>

# OpenVPN serverdə yaradılan TA açarı

<tls-auth>

-----BEGIN OpenVPN Static key V1-----

```
7148f7b12478b04aee1445e18bb96509
b7f8d3c62d20ffb59241a13b714e951d
6e14ef9254097803e76b75e051866287
2cb6db296bbb2a7322b4d641d235b6e3
6426f086ecb6d0650ed61285a5e2a78b
f0f7b2352193c12cbff21ccc82054d00
a00a13d304d7d1365e955eeb30aece8f
15ca06b1c2f504de1ce03f9e955d17f6
a70db5635fd3d3fce914dc090a3f3d59
71db3e9955adf3797c50c50bbe0cbc4b
1aa8d3f363de18474eaeb0b7116edaba
00325fa6fd15da57ad10f9e81cf8d7f2
f1c16d95af55071365cefd8513c906af
e830c0c83f01eea30add98f734fd6011
f5c89c1822d516e0a0c3452c869a5940
929a37e3e064f307b17b8f8e8acb73c3
```

-----END OpenVPN Static key V1-----

</tls-auth>

# Journalları detallı görmək istəsək aşağıdakı sətirdən şərhı silirik

#verb 3

Ubuntu maşınımız yükləndikdən sonra onu yeniləyirik və tələb edilən paketləri yükləyirik:

**apt-get update**

# Anbarları yeniləyirik

**apt-get dist-upgrade**

# Kernel və mövcud paketləri yeniləyirik

# OpenVPN, OpenSSL, LDAP və hər hal üçün RADIUS üçün inteqrasiya paketləri yükləyirik

**apt-get install openvpn easy-rsa openvpn-auth-ldap openvpn-auth-radius**

**openvpn-auth-radius-dbg**

```
apt-get install freeradius freeradius-common freeradius-dbg freeradius-utils
freeradius-ldap
```

```
LDAP utilitlerini yükləyirik
apt-get install ldap-utils
```

```
cd /etc/openvpn # OpenVPN quraşdırma qovluğuna daxil olub, aşağıdakı
 kimi quraşdırma faylını yaradıırıq
cat openvpn.conf # Quraşdırma faylımız aşağıdakı kimi olacaq
plugin /usr/lib/openvpn/openvpn-auth-ldap.so "/etc/openvpn/openvpn-auth-
ldap.conf"
proto tcp
port 1194
dev tun
server 192.168.200.0 255.255.255.0
```

```
Açarlarının generasiya edilməsi qaydası haqqında ətrafalı müzakirə ediləcək
ca /etc/openvpn/keys/keys/ca.crt
cert /etc/openvpn/keys/keys/openvpnsrvr.crt
client-cert-not-required
key /etc/openvpn/keys/keys/openvpnsrvr.key
dh /etc/openvpn/keys/keys/dh2048.pem
tls-auth /etc/openvpn/keys/keys/ta.key 0
```

```
reneg-sec 86400
persist-key
persist-tun
keepalive 10 60
```

```
Müştərilərin hər birinə ayrı quraşdırma yazmaq istəsək aşağıdakı
sətirlərdən şərh silirik.
```

```
#client-to-client
#client-config-dir /usr/local/etc/openvpn/ccd
push "route 10.50.2.0 255.255.255.0"
push "route 10.50.3.0 255.255.255.0"
push "route 10.50.12.0 255.255.255.0"
push "route 10.50.14.0 255.255.255.0"
push "route 10.50.17.0 255.255.255.0"
push "route 10.50.19.0 255.255.255.0"
push "route 192.168.10.0 255.255.255.0"
push "dhcp-option DNS 10.50.3.2"
push "dhcp-option DNS 10.50.3.3"
topology subnet
```

```
user root
group root
```

```
log-append /var/log/openvpn.log
```

Active Directory-ə qoşulmaq üçün LDAP quraşdırma faylımız aşağıdakı kimi olacaq:

```

cat /etc/openvpn/openvpn-auth-ldap.conf # LDAP qoşulmamız üçün
 quraşdırma faylımlz aşağıdaki
 kimidir

<LDAP>
URL ldap://DOMAIN.LAN
BindDN "CN=ADMINISTRATOR,CN=Users,DC=DOMAIN,DC=lan"
Password "DOMAIN_ADMIN_PASS"
Timeout 15
TLSEnable no
FollowReferrals no

</LDAP>

<Authorization>
BaseDN "DC=DOMAIN,DC=lan"
SearchFilter "((&(sAMAccountName=%u)))"
RequireGroup true
<Group>
BaseDN "DC=DOMAIN,DC=lan"
SearchFilter "(cn=OpenVpnFAUsers)"
MemberAttribute "member"
</Group>
</Authorization>

```

Ubuntu maşınımızda şəbəkə və routing quraşdırmaları aşağıdakı kimi olacaq:

```

cat /etc/network/interfaces # Şəbəkə quraşdırma faylı
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 85.132.71.131
netmask 255.255.255.240
network 85.132.71.128
broadcast 85.132.71.143
gateway 85.132.71.129
dns-* options are implemented by the resolvconf package, if
installed
dns-nameservers 10.50.3.2 10.50.3.3
dns-search DOMAIN.az

auto eth1
iface eth1 inet static
address 10.50.3.40
netmask 255.255.255.0
network 10.50.3.0
broadcast 10.50.3.255
Tələb edilən route-lar
up route add -net 10.50.2.0/24 gw 10.50.3.1
up route add -net 10.50.12.0/24 gw 10.50.3.1
up route add -net 10.50.14.0/24 gw 10.50.3.1
up route add -net 10.50.17.0/24 gw 10.50.3.1
up route add -net 10.50.19.0/24 gw 10.50.3.1

```

```
up route add -net 192.168.10.0/24 gw 10.50.3.1
```

**Qeyd:** Unutmayın ki, yazdığınız routing eynilə şəbəkəninizdə olan Router-in üzərindən geriye qayıtmalıdır. Yeni sizin virtual VPN şəbəkəninizin hamı tərəfindən görünməsi üçün router-nizdə aşağıdakına uyğun olan bir routing mütləq əlavə etməlisiniz (Yeni virtual 192.168.200.0/24 şəbəkəsini görmək üçün 10.50.3.40 IP-sindən keçmək lazımdır):

```
ip route 192.168.200.0 255.255.255.0 10.50.3.40
```

Həmçinin Ubuntu maşınıınızı Routing rejimə salmalısınız. Bunu aşağıdakı kimi edəcəyik:

```
sysctl -w net.ipv4.ip_forward=1 # CLI-dan işə salırıq
```

reboot-dan sonra işləməsi üçün `/etc/sysctl.conf` faylında aşağıda sətirin qarşısından şərhini silirik:

```
net.ipv4.ip_forward=1
```

## Ubuntu14.04-də OpenVPN üçün FreeRADIUS-la Active-Directory inteqrasiyası

Məqsədımız Ubuntu 14.04 OS üzərində OpenVPN server qurmaq və onu FreeRADIUS ilə inteqrasiya etməkdir. Sonra isə FreeRADIUS serveri Active Directory ilə inteqrasiya edib, seçilmiş MS LDAP qrupdan istifadəçilərə yetki verməkdir:

İstifadə edilən OS-lar:

```
Windows 2012 Server R2 - DC
Windows 8.1 x64 - Client maşını
Ubuntu 14.04 x64 - OpenVPN
```

Öncə istifadə etdiyimiz Domain Controller-in verilənlərini açıqlayaq

Domain Controller: **DOMAIN.LAN**

DC RO User: **ADMINISTRATOR**

DC RO PASS: **DOMAIN\_PASS**

DC VPN Group: **OpenVPNFAUsers** - Tam yetkisi olan VPN istifadəçiləri (Bütün şəbəkəyə routing olacaq)

Windows 8.1 client quraşdırma faylı aşağıdakı kimi olacaq (faylınızın adı **atl-ad-auth.ovpn**). Faylın genişlənməsi mütləq **.ovpn** olmalıdır:

```
client
auth-user-pass
auth-nocache
proto tcp
remote ovpndc.atl.az
port 1194
```

```
dev tun
nobind
```

```
key-direction 1
```

```
ns-cert-type server
```

```
OpenVPN serverdə yaradılan ca.crt
```

```
<ca>
-----BEGIN CERTIFICATE-----
MIIGxDCCBKygAwIBAgIJALsV/eQc/V5+MA0GCSqGSIb3DQEEBQUAMIGcMQswCQYD
VQQGEWJbWjENMAsGA1UECBMEQkFLVTEUMBIGA1UEBxMLWVuaVlhczFtYVwxFDAS
BgNVBAoTC0FUTEluZm9UZWN0MQswCQYDQQLlEwJjVDEEMBU1UEAxMOQVRMSW5m
b1RlY2ggQ0ExLDAqBgkqhkiG9w0BCQEWHPbWFsLnNoYWWh2ZXJkaXl1dkBhdGx0
ZWN0LmF6MB4XDTE0MDYwODEzNDUzOFoXDTIzMDYwNjE3NDUzOFowZGZwCzAxBGNV
BAYTAKFaMQ0wCwYDVQOIEwRCQUtVMRQwEgYDVQQHEwtZW5pWWFzYV1hbDEUMBIG
A1UEChMLQVRMSW5mb1RlY2gxZCZAJBgNVBAsTAklUMRcwFQYDVQQDEw5BVEsJbmv
VGvjaCBDQTEsMCoGCSqGSIb3DQEJARYdamFtYVwuc2hhaHZlcmRpeWV2QGF0bHRl
Y2guYXowggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQC0kYn6jZf/R1eA
Xs1YH/g36sIQJcxJBmcbXh/atZTy7W8rlXsCw05+RU7OaxrFQUEbed0lnjYiKfri
CutMpT5c7iY6fgfMMoPaIqk8q17qydk8HvqQoac3kjo9wMD7XWLDYiLFk1FkJEW
BIqI2z6vh9/9ka54s6WNRgzT+7+EZqSuwCfC6Dm/0qxp4AvEjapwjaURJ6yEuQYe
Odh5ydTsIcueNnBzkuFZRx505iNcaBQZ2fUVpQvueTCCSHkPt1BGU3TqWIYTUVZL
O4wPQoOyXC9YUvWaysYSLTDMNDVcVgFYfc5C3++nijtFwP08LLDZgiwC7ScYj+Boo
SZ9dkEpIYdb03KBnn+LCO3STVukpwTr+vYKjPITceuElHXDwvXi7wgtopwQhQ+3j
sDCvB+Wg2Bt5zBPC43WTELANOGZFQN1f1kyBNXlBm1tM0k13k75sckj9TXHjrm44
+aVdx1PjKQ86e6/A04wCUOBNf4a00Q8r6PWCfPkqatDn6hCh6ChAYYUqAR5W3eRs
p2D31AWGEHlB1f/+397E66f3ByHvPGQ5n1AQ3wI7q+tLH+qPsoFUKcyfEbctuYvG
D0+9jPhvAAQwc4hBhn+TXRXPkaaaI89iiaJoiF1//R8kqs8t3yxpxjEy0hs2nrx
tboZl91c02fj8e2HvhbMs9v+j6oVTQIDAQAB04IBBTCCAQEWhQYDVR0OBBYEFiCi
KzboRhxhacra8qkU+xvRM4df7MIHRBgNVHSMEgckwgcaAFIciKzboRhxhacra8qkU+
xvRM4df7oYGipIGFMIGcMQswCQYDQQLlEwJjVDEEMBU1UECBMEQkFLVTEUMBIG
A1UEBxMLWVuaVlhczFtYVwxFDASBgNVBAoTC0FUTEluZm9UZWN0MQswCQYDQQL
EwJjVDEEMBU1UEAxMOQVRMSW5mb1RlY2ggQ0ExLDAqBgkqhkiG9w0BCQEWHPb
bWFsLnNoYWWh2ZXJkaXl1dkBhdGx0ZWN0LmF6ggAuxX95Bz9Xn4wDAYDVR0TBAlU
AwEB/zANBgkqhkiG9w0BAQUFAAOCAgEAT+K700aUfXDEfSfMTRppvbcGqoVsaE1
5NjMh206D5KWtErhKbP7id20sdt6Ygq1PQQW3I3thVQ0L686rhz/cr6Vzj41cFI
EqCt4uqZrkoMcvPq82PONvrzKCAuxv5kmZJhWQTB3WXM00A4KnQqW6/HVzSmbQgC
QR6CqNtTlZ21a1RIQRr1CmqRankKC4yQBKbzDwB1XLHVjITdyhJlHXZxBcdXurMX
Uh7AsHOTxbHy4nbyB+Zz1nO37wza6FBeunIqJ/I5eKDCn1lyGELjDsEvrSUCbRRg
IenV9/D9LP4y2KghMkiuDn7vhY3IifCjxQg3JWIA5BdQ/lU1Accsxi0/nyQtZF7
5Nadlw0SOjEe2H6bwxhngcItQyiC34HghNKUF16eYLlMEzGkP7UNLwQN32b3Iia
q9+HTP6TQoci43AoA3NFaUjuKC3zHykesNS8QqOH7MVB4L38/piaGD/K8CsiZH+
QhkiCaJJ7hx/Cfp3VUIKr9yxtAnC5QNbXr9QVCC+mwi/sh9laThPlm1Xd2tKdoZa
My/K605fZnSpzOeFa9j6bRgF2tpbG3jxiWT00F9xUv5EtXZdfies5BRHa1FYGK4
yvVIA/ZJBSB/6CT8mnMjGJcn85CcRggOrOc7lQNmGFKw/YopPYAKzjgilEkNm3
pmPKIhXPdvc=
-----END CERTIFICATE-----
</ca>
```

```
OpenVPN serverdə yaradılan TA açarı
```

```
<tls-auth>
```

```
-----BEGIN OpenVPN Static key V1-----
7148f7b12478b04aee1445e18bb96509
b7f8d3c62d20ffb59241a13b714e951d
6e14ef9254097803e76b75e051866287
2cb6db296bbb2a7322b4d641d235b6e3
```

```
6426f086ecb6d0650ed61285a5e2a78b
f0f7b2352193c12cbff21ccc82054d00
a00a13d304d7d1365e955eeb30aece8f
15ca06b1c2f504de1ce03f9e955d17f6
a70db5635fd3d3fce914dc090a3f3d59
71db3e9955adf3797c50c50bbe0cbc4b
1aa8d3f363de18474eaeb0b7116edaba
00325fa6fd15da57ad10f9e81cf8d7f2
f1c16d95af55071365cefd8513c906af
e830c0c83f01eea30add98f734fd6011
f5c89c1822d516e0a0c3452c869a5940
929a37e3e064f307b17b8fbe8acb73c3
-----END OpenVPN Static key V1-----
</tls-auth>
```

```
Journalları detallı görmək istəsək aşağıdakı sətirdən şərhı silirik
#verb 3
```

Ubuntu maşınımız yükləndikdən sonra onu yeniləyirik və lazımı paketləri yükləyirik:

```
apt-get update # Anbarları yeniləyirik
apt-get dist-upgrade # Kernel və mövcud paketləri yeniləyirik

OpenVPN, OpenSSL, LDAP və hər hal üçün RADIUS üçün integrasiya paketləri
yükləyirik
apt-get install openvpn easy-rsa openvpn-auth-radius openvpn-auth-radius-dbg
FreeRADIUS-u da yükləyirik ki, eyni maşında RADIUS quraşdıraq.
apt-get install freeradius freeradius-common freeradius-dbg freeradius-utils
freeradius-ldap

LDAP utilitlərini yükləyirik
apt-get install ldap-utils

cd /etc/openvpn # OpenVPN quraşdırma qovluğuna daxil olub, aşağıdakı
 kimi quraşdırma faylını yaradıriq
cat openvpn.conf # Quraşdırma faylımız aşağıdakı kimi olacaq
plugin /usr/lib/openvpn/radiusplugin.so /etc/openvpn/radiusplugin.cnf
proto tcp
port 1194
dev tun
server 192.168.200.0 255.255.255.0

ca /etc/openvpn/keys/keys/ca.crt
cert /etc/openvpn/keys/keys/openvpnsrvr.crt
client-cert-not-required
key /etc/openvpn/keys/keys/openvpnsrvr.key
dh /etc/openvpn/keys/keys/dh2048.pem
tls-auth /etc/openvpn/keys/keys/ta.key 0

persist-key
persist-tun
```



```
keepalive 10 60
```

```
Client-lərimizin hər birinə ayrı quraşdırma yazmaq istəsək aşağıdakı
sətirlərdən şərh silirik.
```

```
#client-to-client
#client-config-dir /usr/local/etc/openvpn/ccd
push "route 10.50.2.0 255.255.255.0"
push "route 10.50.3.0 255.255.255.0"
push "route 10.50.12.0 255.255.255.0"
push "route 10.50.14.0 255.255.255.0"
push "route 10.50.17.0 255.255.255.0"
push "route 10.50.19.0 255.255.255.0"
push "route 192.168.10.0 255.255.255.0"
push "dhcp-option DNS 10.50.3.2"
push "dhcp-option DNS 10.50.3.3"
topology subnet
```

```
user root
group root
```

```
log-append /var/log/openvpn.log
```

OpenVPN ilə FreeRADIUS-u birləşdirən quraşdırma faylı aşağıdakı kimi olacaq:

```
cat /etc/openvpn/radiusplugin.cnf # RADIUS-a qoshulan quraşdırma faylı
NAS-Identifier=OpenVpn
Service-Type=5
Framed-Protocol=1
NAS-Port-Type=5
NAS-IP-Address=127.0.0.1
OpenVPNConfig=/etc/openvpn/openvpn.conf # OpenVPN quraşdırma faylı
overwriteccfiles=true
server
{
 acctport=1813 # RADIUS accounting portu
 authport=1812 # RADIUS autentifikasiya portu
 name=127.0.0.1 # RADIUS IP
 retry=1
 wait=1
 sharedsecret=freebsd # FreeRADIUS ilə OpenVPN arasında
 # istifadə edilən açar
}
```

### İndi isə keçək FreeRADIUS-un quraşdırmasına

OpenVPN client-i qoşulmaq üçün FreeRADIUS-un clientlər siyahısına əlavə edirik(Quraşdırma faylı aşağıdakı kimi olacaq):

```
cat /etc/freeradius/clients.conf # Clientlər quraşdırma faylı
```

```
client localhost {
 ipaddr = 127.0.0.1 # OpenVPN server
 secret = freebsd # OpenVPN şifre
 require_message_authenticator = no
 shortname = localhost
 nastype = other
}
```

FreeRADIUS-un öz quraşdırma faylı aşağıdakı kimi olacaq:

```
cat /etc/freeradius/radiusd.conf # FreeRADIUS quraşdırma faylı
prefix = /usr
exec_prefix = /usr
sysconfdir = /etc
localstatedir = /var
sbindir = ${exec_prefix}/sbin
logdir = /var/log/freeradius
raddbdir = /etc/freeradius
radacctdir = ${logdir}/radacct
name = freeradius
confdir = ${raddbdir}
run_dir = ${localstatedir}/run/${name}
db_dir = ${raddbdir}
libdir = /usr/lib/freeradius
pidfile = ${run_dir}/${name}.pid
user = freerad
group = freerad
max_request_time = 30
cleanup_delay = 5
max_requests = 1024
listen {
 type = auth
 ipaddr = 127.0.0.1
 port = 1812
}
listen {
 ipaddr = 127.0.0.1
 port = 1813
 type = acct
}
hostname_lookups = no
allow_core_dumps = no
regular_expressions = yes
extended_expressions = yes
log {
 destination = files
 file = ${logdir}/radius.log
 syslog_facility = daemon
 stripped_names = no
 auth = no
 auth_badpass = no
 auth_goodpass = no
}
```

```

checkrad = ${sbindir}/checkrad
security {
 max_attributes = 200
 reject_delay = 1
 status_server = yes
}
proxy_requests = no
$INCLUDE proxy.conf
$INCLUDE clients.conf
thread pool {
 start_servers = 5
 max_servers = 32
 min_spare_servers = 3
 max_spare_servers = 10
 max_requests_per_server = 0
}
modules {
 $INCLUDE ${confdir}/modules/
 $INCLUDE eap.conf
}
instantiate {
 exec
 expr
 expiration
 logintime
}
$INCLUDE policy.conf
$INCLUDE sites-enabled/

```

FreeRADIUS ilə OpenVPN arasında olan qoşulmanın düzgünlüyünü test etmək üçün **/etc/freeradius/users** faylına aşağıdakı sətiri əlavə etmək lazımdır (Test üçün Vasif adlı istifadəçi **freebsd** şifrəsi ilə):

```
"vasif" Cleartext-Password := "freebsd"
```

```

/etc/init.d/openvpn restart # OpenVPN serveri restart edirik
/etc/init.d/freeradius restart # FreeRADIUS serveri restart edirik

```

```
freeradius -fX # FreeRADIUS-u debug etmək üçün bu əmrəndən istifadə edirik
```

Windows 8.1 client-dən OpenVPN serverə qoşulub uğurlu nəticə əldə etməlisiniz. Əgər uğurlu nəticə olmasa debug edilir. Əgər hər şey uğurlu olsa keçirik RADIUS-un MS LDAP-la inteqrasiya edilməsinə.

OpenVPN-in istifadəçilərinin AD-dən yoxlanılması üçün FreeRADIUS serveri MS LDAP ilə inteqrasiya etmək lazımdır. Bunun üçün aşağıdakıları edirik:

```
cat /etc/freeradius/sites-enabled/default # Susmaya görə olan Virtual
 RADIUS-u quraşdırırıq
```

```

authorize {
 files
 ldap # Qeydiyyat LDAP-ın

```

```

 if (LDAP-Group == "OpenVpnFAUsers") { # OpenVpnFAUsers DC qrupundan
 ok olarsa izin verilir.
 }
 else {
 reject # Əks halda bağlayırıq
 }
}
authenticate {
 Auth-Type LDAP {
 ldap # Həmçinin autentifikasiyanı
 ldap qrupundan alacağımız
 }
}
preacct {
 preprocess
 acct_unique
 suffix
 files
}
accounting {
 detail
 unix
 radutmp
 exec
 attr_filter.accounting_response
}
session {
}
post-auth {
 exec
}
pre-proxy {
}
post-proxy {
}

```

LDAP modulunu quraşdırırıq ki, muraciət edən istifadəçilərin təyin edilməsi üçün, Domain Controller-ə qoşulub filter edə bilsin.

```

cat /etc/freeradius/modules/ldap # LDAP modulunun quraşdırması aşağıdakı
 kimi olacaq

```

```

ldap {
 server = "DOMAIN.LAN"
 identity = "CN=ADMINISTRATOR,CN=Users,DC=atl,DC=lan"
 password = "DOMAIN_PASS"
 basedn = "DC=atl,DC=lan"
 filter = "(sAMAccountName=%{%{Stripped-User-Name}:-%{User-Name}})"
 ldap_connections_number = 5
 timeout = 4
 timelimit = 3
 net_timeout = 1
 tls {

```

```

 start_tls = no
 }
 dictionary_mapping = ${confdir}/ldap.attrmap
 edir_account_policy_check = no
 groupname_attribute = "cn"
 groupmembership_filter =
"(| (&(objectClass=GroupOfNames) (member=%{control:Ldap-
UserDn})) (&(objectClass=GroupOfUniqueNames) (uniquemember=%{control:Ldap-
UserDn})))"
 groupmembership_attribute = "memberOf"
 compare_check_items = no
 do_xlat = yes
 access_attr_used_for_allow = yes
 chase_referrals = yes
 rebind = yes
 set_auth_type = yes
 ldap_debug = 0
 keepalive {
 idle = 60
 probes = 3
 interval = 3
 }
}

```

```

freeradius -fX # Debug rejimdə jurnallarda uğurlu nəticə
 olaraq seçdiyim aşağıdakı jurnalların
 oxşarlarını sizdə mütləq görməlisiniz.

```

```

[ldap] performing user authorization for jamal
[ldap] expand: %{Stripped-User-Name} ->
[ldap] ... expanding second conditional
[ldap] expand: %{User-Name} -> jamal
[ldap] expand: (sAMAccountName=%{%{Stripped-User-Name}:-%{User-Name}}) ->
(sAMAccountName=jamal)
[ldap] expand: DC=atl,DC=lan -> DC=atl,DC=lan
 [ldap] ldap_get_conn: Checking Id: 0
 [ldap] ldap_get_conn: Got Id: 0
 [ldap] attempting LDAP reconnection
 [ldap] (re)connect to DOMAIN.LAN:389, authentication 0
 [ldap] bind as CN=ADMINISTRATOR,CN=Users,DC=atl,DC=lan/DOMAIN_PASS to
DOMAIN.LAN:389
 [ldap] waiting for bind result ...
 [ldap] Bind was successful

[ldap] Setting Auth-Type = LDAP
[ldap] user jamal authorized to use remote access
 [ldap] ldap_release_conn: Release Id: 0
++[ldap] returns ok
++? if (LDAP-Group == "OpenVpnFAUsers")
 [ldap] Entering ldap_groupcmp()
 expand: DC=atl,DC=lan -> DC=atl,DC=lan

```

```

 expand: (|(&(objectClass=GroupOfNames) (member=%{control:Ldap-
UserDn})) (&(objectClass=GroupOfUniqueNames) (uniquemember=%{control:Ldap-
UserDn}))) -> (|(&(objectClass=GroupOfNames) (member=CN\3dJamal
Shahverdiyev\2cOU\3dATLTech
Users\2cOU\3dATLTech\2cDC\3dat1\2cDC\3dlan)) (&(objectClass=GroupOfUniqueNames
) (uniquemember=CN\3dJamal Shahverdiyev\2cOU\3dATLTech
Users\2cOU\3dATLTech\2cDC\3dat1\2cDC\3dlan)))
 [ldap] ldap_get_conn: Checking Id: 0
 [ldap] ldap_get_conn: Got Id: 0
 [ldap] performing search in DC=atl,DC=lan, with filter
(&(cn=OpenVpnFAUsers) (|(&(objectClass=GroupOfNames) (member=CN\3dJamal
Shahverdiyev\2cOU\3dATLTech
Users\2cOU\3dATLTech\2cDC\3dat1\2cDC\3dlan)) (&(objectClass=GroupOfUniqueNames
) (uniquemember=CN\3dJamal Shahverdiyev\2cOU\3dATLTech
Users\2cOU\3dATLTech\2cDC\3dat1\2cDC\3dlan))))

 [ldap] performing search in CN=Jamal Shahverdiyev,OU=ATLTech
Users,OU=ATLTech,DC=atl,DC=lan, with filter (objectclass=*)
 [ldap] performing search in CN=OpenVpnFAUsers,OU=ATLTech
Groups,OU=ATLTech,DC=atl,DC=lan, with filter (cn=OpenVpnFAUsers)
rlm_ldap::ldap_groupcmp: User found in group OpenVpnFAUsers
 [ldap] ldap_release_conn: Release Id: 0
? Evaluating (LDAP-Group == "OpenVpnFAUsers") -> TRUE
++? if (LDAP-Group == "OpenVpnFAUsers") -> TRUE
++- entering if (LDAP-Group == "OpenVpnFAUsers") {...}
+++[ok] returns ok

[ldap] user jamal authenticated succesfully
++[ldap] returns ok

```

**Qeyd:** Unutmayın ki, **/etc/freeradius/users** faylında heç bir istifadəçi qeyd edilməyib və fayl tamamilə boshdur.

```
/etc/init.d/freeradius start # Sonda FreeRADIUS-u işə salırıq
```

### **Ubuntu 14.04 x64 OpenVPN və çoxlu LDAP qrupları**

Məqsəd odur ki, bizim OpenVPN serverdən istifadə edib daxili şəbəkəyə yetki almaq istəyən istifadəçi sayı həddən artıq çox ola bilər. Həmçinin hər kəs istəyəcək ki, öz Domain Controller istifadəçi adından və şifrəsindən istifadə edərək OpenVPN serverə daxil olsun. Ancaq nəzərə almalıyıq ki, adi istifadəçinin daxili şəbəkəni görməsi lazım deyil. Bunun üçün Domain Controller-də iki ədəd ayrı-ayrı qrup yaradıb hər bir istifadəçini yetkiyə uyğun qrupa elavə etmək lazımdır. Yəni deyək ki, DC-mizdə iki ədəd qrup var.

**OpenVpnFAUsers** və **OpenVpnMAUsers**. **OpenVpnFAUsers** qrupunun bütün şəbəkəyə yetkisi var ancaq, **OpenVpnMAUsers** qrupunun isə minimal yetkisi var.

Nəzərdə tutulur ki, öncəki başlıqlarımıza uyğun olaraq artıq serverinizi AD ilə integrasiya etmişiniz. Ancaq **/etc/openvpn/** qovluğumuzda iki ədəd ayrı quraşdırma yaradacağıq hansı ki, hər biri ayrı portda qulaq asır, ayrı LDAP quraşdırmasına muraciət edir və yetki təyin edilmiş client-inə əsasən uyğun virtual şəbəkədən İP ünvan paylayır.

Artıq quraşdırmamıza başlaya bilərik:

```
cd /etc/openvpn # Quraşdırma qovluğumuza daxil oluruq

İlk olaraq OpenVpnFAUsers qrupu üçün 1194-cu portda qulaq asan quraşdırmamızı
və ona aid olan LDAP quraşdırmamızı açıqlayaq:
cat /etc/openvpn/openvpn.conf # İlk quraşdırma faylımız
plugin /usr/lib/openvpn/openvpn-auth-ldap.so "/etc/openvpn/openvpn-auth-ldap.conf"
proto tcp
reneg-sec 86400
port 1194
dev tun
server 192.168.200.0 255.255.255.0

ca /etc/openvpn/keys/keys/ca.crt
cert /etc/openvpn/keys/keys/openvpnsrvr.crt
client-cert-not-required
key /etc/openvpn/keys/keys/openvpnsrvr.key
dh /etc/openvpn/keys/keys/dh2048.pem
tls-auth /etc/openvpn/keys/keys/ta.key 0

reneg-sec 86400
persist-key
persist-tun
keepalive 10 60

push "route 10.50.2.0 255.255.255.0"
push "route 10.50.3.0 255.255.255.0"
push "route 10.50.12.0 255.255.255.0"
push "route 10.50.14.0 255.255.255.0"
push "route 10.50.17.0 255.255.255.0"
push "route 10.50.19.0 255.255.255.0"
push "route 192.168.10.0 255.255.255.0"
push "dhcp-option DNS 10.50.3.2"
push "dhcp-option DNS 10.50.3.3"
topology subnet

user root
group root

log-append /var/log/openvpn.log
```

```
cat /etc/openvpn/openvpn-auth-ldap.conf # openvpn.conf faylına aid
 olan ldap quraşdırma faylımız
```

```
<LDAP>
 URL ldap://DOMAIN.LAN
 BindDN "CN=ADMINISTRATOR,CN=Users,DC=DOMAIN,DC=LAN"
 Password "DOMAIN_USER_PASS"
 Timeout 15
 TLSEnable no
 FollowReferrals no
</LDAP>

<Authorization>
 BasedN "DC=DOMAIN,DC=LAN"
 SearchFilter "(&(sAMAccountName=%u))"
 RequireGroup true
 <Group>
 BasedN "DC=DOMAIN,DC=LAN"
 SearchFilter "(cn=OpenVpnFAUsers)"
 MemberAttribute "member"
 </Group>
</Authorization>
```

Həmçinin 1195-ci portda qulaq asan və **OpenVpnMAUsers** qrupunun istifadəçilərini qəbul edən ikinci quraşdırma faylımızı açıqlayaq:

```
cat /etc/openvpn/openvpnma.conf # DC-mizin OpenVpnMAUsers qrupunda olan
 istifadəçilər üçün quraşdırma faylı
plugin /usr/lib/openvpn/openvpn-auth-ldap.so "/etc/openvpn/openvpnma-auth-
ldap.conf"
proto tcp
reneg-sec 86400
port 1195
dev tun
server 192.168.201.0 255.255.255.0

ca /etc/openvpn/keys/keys/ca.crt
cert /etc/openvpn/keys/keys/openvpnsrvr.crt
client-cert-not-required
key /etc/openvpn/keys/keys/openvpnsrvr.key
dh /etc/openvpn/keys/keys/dh2048.pem
tls-auth /etc/openvpn/keys/keys/ta.key 0

reneg-sec 86400
persist-key
persist-tun
keepalive 10 60

push "route 10.50.3.0 255.255.255.0"
topology subnet

user root
```



```
group root
```

```
log-append /var/log/openvpnma.log
```

```
cat /etc/openvpn/openvpnma-auth-ldap.conf # openvpnma.conf faylına aid
 olan ldap quraşdırma faylımız
```

```
<LDAP>
 URL ldap://DOMAIN.LAN
 BindDN "CN=ADMINISTRATOR,CN=Users,DC=DOMAIN,DC=LAN"
 Password "DOMAIN_USER_PASS"
 Timeout 15
 TLSEnable no
 FollowReferrals no
```

```
</LDAP>
```

```
<Authorization>
 BaseDN "DC=DOMAIN,DC=LAN"
 SearchFilter "(&(sAMAccountName=%u))"
 RequireGroup true
 <Group>
 BaseDN "DC=DOMAIN,DC=LAN"
 SearchFilter "(cn=OpenVpnMAUsers)"
 MemberAttribute "member"
```

```
</Group>
```

```
</Authorization>
```

**Qeyd:** Ancaq Minimal istifadəçilər və tam yetkili istifadəçilər üçün şəbəkəmizin geri qayıdan routinginin işləməsi üçün unutmamaq lazımdır ki, daxili şəbəkəyə aid olan router-dən **192.168.200.0/24** və **192.168.201.0/24** şəbəkəsi üçün routing yazmaq lazımdır.

```
/etc/init.d/openvpn restart # OpenVPN daemon-u restart edirik
```

Client ilə qoşulub sinaqları etdikdə unutmayın ki, hər bir client-i siz təyin etdiyiniz şəbəkə yetkisinə görə öz profile quraşdırmasında ayırmaq lazımdır. Yeni ki, tam yetkisi olan client quraşdırmasında uyğun sertifikatlar və **1194**-cu porta qoşulma olmalıdır. Eynilə də **1195**-ci porta aid olan client minimal yetkili istifadəçi və öz sertifikatları ilə quraşdırılmalıdır.

**Qeyd:** Unutmayın ki, siz həmçinin **/etc/openvpn** qovluğunun altında eyni vaxtda həm bir neçə LDAP-la inteqrasiya edilmiş **\*.conf** faylları və sertifikatlarla DOMAIN-ə inteqrasiya edilmiş **\*.conf** faylları istifadə edə bilərsiniz ☺.

## BÖLÜM 6

### Scripting və Pluginlər

Bu başlıqda biz aşağıdakıları açıqlayacağıq:

- Client tərəfdə up/down scriptin istifadə edilməsi
- Windows login greeter
- client-connect/client-disconnect scriptlərin istifadə edilməsi
- learn-address scriptin istifadə edilməsi
- tls-verify scriptin istifadə edilməsi
- auth-user-pass-verify scriptin istifadə edilməsi
- Script ardıcılığı
- Script təhlükəsizliyi və jurnallama
- down-root pluginin istifadə edilməsi
- PAM authentication pluginin istifadə edilməsi

#### **Giriş**

OpenVPN-in əsas güclü bacarıqlarından biridə odur ki, scriptləmə imkanı və öz imkanlarını pluginlər vasitəsilə artırmaqdır. client-side scripting istifadə edilməsi, qoşulma prosesi site-spesifik tələblərə uyğun olaraq dəyişdirilə bilər (Məsələn genişlənmiş routing ya da şəbəkə disklərinin xəritələnməsi). Server-side scripting ilə bu mümkündür ki, müxtəlif müştərilərə seçilmiş IP ünvanlar verilsin ya da autentifikasiya prosesinin genişləndirilməsi ilə əlavə istifadəçi adı və şifrə yoxlanışını artırmaq olar. Pluginlər çox istifadə edilmir olur o halda ki, OpenVPN autentifikasiyasını mövcud olan autentifikasiya frameworkları ilə integrasiya edilir (Məsələn: PAM, LDAP ya da Active Directory).

Bu başlıqda biz fikrimizi scriptləməyə yönləndirəcəyik. Hər iki istiqamətdə həm client və həm də server tərəfdə və əsas istifadə edilən pluginlər.

## Client tərəfdə up/down scriptin istifadə edilməsi

Bu başlıqda biz client tərəfdə adi up və down scriptdən istifadə edib görəceyik ki, OpenVPN necə bu scriptləri çağırır. Mesajların fayla jurnallanması və həmçinin dəyişən mühiti ilə, biz tez görə bilərik ki, OpenVPN hansı informasiyanı UP və Down scriptlərində təqdim edir.

### İşə başlayaq

OpenVPN 2.3-ü ya da daha yuxarı versiyasını 2 maşına yükləyək. Əmin olaq ki, maşınlar şəbəkə ilə bir birini görür. 2-ci başlığın ilk misalında olan client və server sertifikatlarını istifadə edin. Bu misalda server maşını FreeBSD9.2 x64 OpenVPN2.3-də, client isə Windows7 maşını OpenVPN2.3-də olacaq.

### Bunu necə edək...

1. Server quraşdırma faylını aşağıdakı sətirlərlə yaradaq:

```
proto udp
port 1194
dev tun
server 192.168.200.0 255.255.255.0

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnsrver.crt
key /usr/local/etc/openvpn/openvpnsrver.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0

persist-key
persist-tun
keepalive 10 60

topology subnet

user nobody
group nobody # nogroup olur bəzi distributivlərdə

daemon
log-append /var/log/openvpn.log
```

Faylı **example6-1-server.conf** adında yadda saxlayın.

2. Serveri işə salın:  
root@siteA:/usr/local/etc/openvpn # **openvpn --config example6-1-server.conf**

3. Client quraşdırma faylını yaradaq:

```
client
proto udp
remote openvpnsrver.example.com
port 1194

dev tun
nobind

ca "c:/program files/openvpn/config/ca.crt"
cert "c:/program files/openvpn/config/client2.crt"
key "c:/program files/openvpn/config/client2.key"
```

```
tls-auth "c:/program files/openvpn/config/ta.key" 1
```

```
ns-cert-type server
```

```
script-security 2 system
```

```
up "C:\\updownfold\\updown.bat"
```

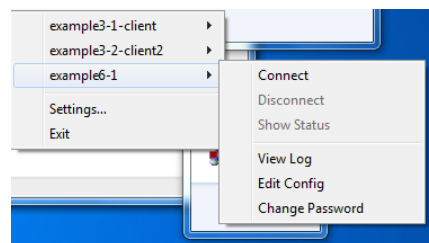
```
down "C:\\updownfold\\updown.bat"
```

Qeyd edin ki, **ca**, **cert**, **key** və **tls-auth** direktivlərinin istifadə edilməsində UNIX slash istifadə edilə bilər ancaq, **up** yada **down** scriptlərində deyil. Bu tərkibi **example6-1.ovpn** adı ilə yadda saxlayın. Öncədən **C:\updownfold** və **c:\temp\** qovluqlarını yaradırıq. Bat faylımızı **C:\updownfold** qovluğunda yerləşdiririk. Mütləq nəzərə alın ki, OpenVPN windows7-də olan qovluq adlarında boşluq olduqda onu anlamır, ona görə də yaratdığınız qovluqda diqqətli olun ki, boşluq olmasın.

4. Windows client məşında lazımı ünvanda **c:\updownfold\updown.bat** adlı batch faylı yaradın:

```
@echo off
echo === BEGIN '%script_type%' script === >> c:\temp\openvpn.log
echo Script name: [%0] >> c:\temp\openvpn.log
echo Command line argument 1: [%1] >> c:\temp\openvpn.log
echo Command line argument 2: [%2] >> c:\temp\openvpn.log
echo Command line argument 3: [%3] >> c:\temp\openvpn.log
echo Command line argument 4: [%4] >> c:\temp\openvpn.log
echo Command line argument 5: [%5] >> c:\temp\openvpn.log
echo Command line argument 6: [%6] >> c:\temp\openvpn.log
echo Command line argument 7: [%7] >> c:\temp\openvpn.log
echo Command line argument 8: [%8] >> c:\temp\openvpn.log
echo Command line argument 9: [%9] >> c:\temp\openvpn.log
set >> c:\temp\openvpn.log
echo === END '%script_type%' script === >> c:\temp\openvpn.log
```

5. Sonda OpenVPN client-i işə salaq:



OpenVPN serverə qoşulma uğurla başa çatdıqdan sonra, jurnal faylı **c:\temp\openvpnlog** aşağıdakı sətirlərə uyğun olan tərkibi daşıyacaq:

```
=== BEGIN 'up' script ===
Script name: [C:\updownfold\updown.bat]
Command line argument 1: ["Local Area Connection 2"]
Command line argument 2: [1500]
Command line argument 3: [1541]
Command line argument 4: [192.168.200.2]
Command line argument 5: [255.255.255.0]
Command line argument 6: [init]
Command line argument 7: []
```

```
Command line argument 8: []
Command line argument 9: []
COMSPEC=C:\Windows\system32\cmd.exe
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.JS;.WS;.MSC
PROMPT=PG
script_type=up
dev_type=tun
dev=Local Area Connection 2
link_mtu=1541
tun_mtu=1500
script_context=init
ifconfig_broadcast=192.168.200.255
ifconfig_netmask=255.255.255.0
ifconfig_local=192.168.200.2
common_name=openvpnserver
trusted_port=1194
trusted_ip=1.1.1.10
untrusted_port=1194
untrusted_ip=1.1.1.10
tls_serial_0=1
tls_digest_0=7a:26:e8:c1:ff:bf:43:7d:6a:33:a8:5c:b9:3c:21:1c:9e:60:2b:2f
tls_id_0=C=AZ, O=Itvpn, CN=openvpnserver, emailAddress=openvpn-ca@domain.lan
X509_0_emailAddress=openvpn-ca@domain.lan
X509_0_CN=openvpnserver
X509_0_O=Itvpn
X509_0_C=AZ
tls_serial_1=13728201484454112052
tls_digest_1=23:a9:58:6b:92:75:f5:f9:4b:78:ca:bb:cf:05:79:30:e2:11:28:48
tls_id_1=C=AZ, O=Itvpn, CN=Itvpn CA, emailAddress=openvpn-ca@domain.lan
X509_1_emailAddress=openvpn-ca@domain.lan
X509_1_CN=Itvpn CA
X509_1_O=Itvpn
X509_1_C=AZ
SystemRoot=C:\Windows
config=example6-1.ovpn
verb=1
daemon=0
daemon_log_redirect=1
daemon_start_time=1393590280
daemon_pid=1288
proto_1=udp
local_port_1=0
remote_1=openvpnserver.example.com
remote_port_1=1194
PATH=C:\Windows\System32;C:\WINDOWS;C:\WINDOWS\System32\Wbem
=== END 'up' script ===
```

Client serverdən disconnect elədikdə, script eyni əmr parametrlərilə yenidən çağırılır ancaq, ikinci **script\_type** down-dur.

Qeyd edin ki, ilk command line arqumenti TUN alətin adı olur. Bu Linux/UNIX və ya MAC sistemlərində avtomatik olaraq **tun0** yada **tun1** olacaq ancaq, windows maşınlarında bu **TAP-Win32** adapterdir.

**Bu necə işləyir...**

OpenVPN server ilə qoşulma uğurla başa çatdıqdan sonra OpenVPN client **up** scriptini çağırır. Əgər up scripti çıxış code-u sifıra bərabər olmayan kod qaytarırsa, qoşulma seansı kəsilir.

Uyğun olaraq qoşulma dayandırıldıqda da **down** scripti VPN qoşulması tam dayandırıldıqdan sonra yerinə yetirilir.

**Qeyd:** **up** və **down** direktivləri istifadə edildikdə, (\\) simvollarından istifadə edin: OpenVPN backslash simvollarını daxilən tərcümə edir və uyğun olaraq iki dəfə yazılması tələb edilir.

### Daha da ətraflı...

Bu başlıqda, biz up və down scriptlərini istifadə elədikdə çoxlu irəliləmiş üsulları görəceyik hansı ki, həmçinin test script ilə VPN serverin hostname-ni yoxlayacağıq.

### Mühit dəyişənləri

Bu misalda istifadə edilən script sadəcə bütün dəyişən mühitlərini kənar bir fayla yazır. Bu mühit dəyişənləri remote server haqqında istifadə edilə biləcək informasiyanı özündə təşkil edir. Məsələn sertifikatın **common\_name**-i. Bu scriptin ssenarisi ondan ibarətdir ki, yoxlayaraq görək remote hostname sertifikatın **common\_name**-inə uyğundur ya yox. Remote hostname-in IP ünvanı **trusted\_ip** kimi mövcuddur.

### Qoşulma kəsilsə 'down' scriptin çağırılması

Down scripti OpenVPN serverə gedən qoşulma dayandıqdan sonra yerinə yetirildi. Həmçinin mümkündür ki, scripti qoşulma kəsilməzdən öncə yerinə yetirə bilək. Bunu eləmək üçün clientin quraşdırma faylına aşağıdakı direktivi əlavə edin:

```
down-pre
```

### İrəliləmiş: remote hostname-in yoxlanılması

Up scriptin daha irəliləmiş istifadəsi o olar ki, yoxlayaraq görək remote IP onun hostname-inə uyğun gəlir ya yox. Misal üçün web browser ilə təhlükəsiz web saytları yoxlanışdır. UNIX/Linux maşınlarında bir çox asan olaraq up scripti ilə edilə bilər:

```
#!/usr/local/bin/bash
reverse DNS lookup
server_name=`host $untrusted_ip | sed -n 's/.*name pointer
\(.*\)\.\/\1/p'`
if ["$server_name" != "$common_name"]
then
 echo "Server certificate does not match hostname."
 echo "Aborting"
 exit 1
fi
```

Ancaq bu Windows-da bu işi "PowerShell" yada "Cygwin"-siz görmək çox çətindir.

## Windows login greeter

Bu misal öncəkinin davamıdır. Bu göstərəcək ki, necə OpenVPN serverdən, client qoşulması fazasında **push** ediləcək. Bu mesaj həm hüquqi tərəfdən və həm də xəbərdarlıq kimi istifadə edilə bilər. Bunu düzgün eləmək üçün biz **setenv-safe** direktivindən istifadə edəcəyik hansı ki, OpenVPN2.3 və daha böyük versiyalarında var. Bu direktiv çox istifadə edilən **setenv**-dən fərqli olaraq clientlərə də ötürülə bilər.

## Getting ready

OpenVPN2.3 ya da böyük versiyasını 2 məşində yükləyin. Əmin olun ki, maşınlar şəbəkə ilə bir-birlərini görürlər. 2-ci başlıqda client-server IP şəbəkələri üçün yaratdığınız client və server sertifikatlarını burda istifadə edin. Bu başlıqda server maşını FreeBSD9.2 x64 OpenVPN2.3-də, client isə Windows7-də OpenVPN2.3-də işləyəcək. Server quraşdırma faylını öncə istifadə elədiyimiz **example6-1-server.conf** faylını istifadə edin.

## Necə edək...

1. **example6-1-server.conf** faylını **example6-2-server.conf** faylına nüsxələyin və **example6-2-server.conf** faylının içinə aşağıdakı sətiri əlavə edin:  
**push "setenv-safe MSG 'This is a message from the OpenVPN server'"**

2. Serveri işə salın:  
root@siteA:/usr/local/etc/openvpn # **openvpn --config example6-2-server.conf**

3. Sonra client quraşdırma faylını yaradın:

```
client
proto udp
remote openvpnservers.example.com
port 1194

dev tun
nobind

ca "c:/program files/openvpn/config/ca.crt"
cert "c:/program files/openvpn/config/openvpnclient2.crt"
key "c:/program files/openvpn/config/openvpnclient2.key"
tls-auth "c:/program files/openvpn/config/ta.key" 1

script-security 2 system
up 'C:\\Windows\\System32\\wscript.exe c:\\openvpn\\example6-2.vbs'
```

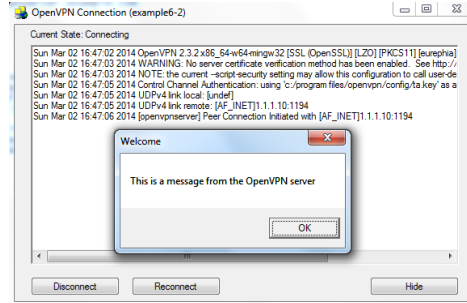
Yuxarıdakı sətirləri **example6-2.ovpn** faylında yadda saxlayın.

**Qeyd:** Unutmayın up scripti mütləq və mütləq tək dırnaq daxilində olmalıdır. Əks halda heçnə işləməyəcək.

4. Öncədən **c:\openvpn** qovluğunu yaradın. Sonra **example6-2.vbs** adlı Visual Basic Scripti **c:\openvpn** ünvanında yaradın:

```
Set oShell = CreateObject("WScript.Shell")
msg=oShell.ExpandEnvironmentStrings ("%OPENVPN_MSG%")
MsgBox msg, , "Welcome"
```

**example6-2.vbs** faylını **c:\openvpn** ünvanında saxlayın ya da digər qovluq saxlaya bilərsiniz ancaq, unutmayın ki, qovluğun adında boşluq heç bir halda olmalı deyil. OpenVPN client-i işə salın. Qoşulma fazasında mesaj ayrıca başlıqda ekrana çap ediləcək.



**Qeyd:** Bu tip qoşulma ilə bağlı daha ətraflı məlumatı <http://community.openvpn.net/openvpn/wiki/Openvpn23ManPage> linkdən əldə edə bilərsiniz.

### Bu necə işləyir...

Aşağıdakı server direktivi **setenv-safe MSG...** başlığını qoşulan clientə ötürür:  
**push "setenv-safe MSG 'This is a message from the OpenVPN server'"**

Client isə əgər direktiv aşağıdakı formadakı kimi olarsa yerinə yetirəcək:  
**setenv-safe MSG 'This is a message from the OpenVPN server'**

**setenv-safe** direktivi dəyişən əlavə elədikdə, sistem dəyişənləri ilə conflict yaratmamaq üçün onların hamısının önünə **OPENVPN\_** əlavə edir.

OpenVPN2.1-də aşağıdakı direktiv ona görə tələb edilir ki, biz Visual Basic Scripti birbaşa yerinə yetirmək istəyirik (Ancaq **OpenVPN2.3** versiyasından başlayaraq yerinə yetiriləcək exe faylının ünvanı tam olaraq göstərildiyindən, indi system yazmağa ehtiyac qalmadı. Sadəcə **script-security 2** yazmaq yetər.):

```
script-security 2 system
```

Visual Basic script yeni mühit dəyişəni elan edir və ekrana çap edilən mesajı bizə göstərir.

### Daha-da ətraflı...

Windows platformasında scriptləri yazdıqda bəzi önəmli hissələr vardır ki, yadda saxlamaq lazımdır.

### Fayl adlarında olan boşluq

OpenVPN-in script adlarının istifadəsində ciddi problemləri var və faylın adının ünvan olaraq işləmə yerini seçdikdə qovluq adlarında boşluq olarsa, bu sizə ciddi problemlər yaradacaq. Misal olaraq aşağıdakı sətirdə həm ünvana



daxil olub sonra scripti yerinə yetirmək üçün tək, tam ünvanı scripti yerinə yetirmək üçün isə cüt dırnaqdan istifadə etməlisiniz:

```
cd "c:\\program\ files\\openvpn\\scripts"
up 'C:\\Windows\\System32\\wscript.exe example6-2.vbs'
```

GUI istifadə elədikdə digər yol isə aşağıdakı kimi ola bilər. Çünki, **openvpnserver.exe** susmaya görə **bin** qovluğundan işə düşür və siz bir ünvan geridən bunu işə sala bilərsiniz.

```
up 'C:\\Windows\\System32\\wscript.exe ..\\scripts\\example6-2.vbs'
```

Digər yolu isə aşağıdakı kimi ola bilər.

```
up 'C:\\Windows\\system32\\wscript.exe C:\\Program\
Files\\OpenVPN\\scripts\\example6-2.vbs'
```

### **setenv ya da setenv-safe**

Adi halda aşağıdakı direktiv mühitin təyin edilməsi üçün istifadə edilir hansı ki, OpenVPN-in çağırma bildiyi istənilən script üçün mövcud olur:

```
setenv env-var value
```

Ancaq, **setenv** direktivi server tərəfdən client-ə ötürülə bilməz. **setenv-safe** dəyişəni isə hansı ki, OpenVPN2.1-dən başlayaraq işləyir təyin edilə bilər.

### **Təhlükəsizlik məntiqləri**

Bu misal VPN qoşulmasında müəllif hüquqları və ya xəbərdaredici mesaj kimi istifadə edilə bilər. Ancaq yenə də şərait yaradılmışdır ki, istifadəçi bundan istifadə etməsin hansı ki, quraşdırma faylını dəyişmək yetər. Əgər daha da böyük təhlükəsizlik işləri planlaşdırılırsa onda OpenVPN serverin özündə bu məqsədə uyğun programlaşdırılmalıdır.

### **client-connect/client-disconnect scriptlərinin istifadə edilməsi**

Bu misal göstərəcək ki, **client-connect** scriptini server tərəfdə client qoşulması anında yerinə yetirəcəyik. Eynilədə **client-disconnect** scriptini client öz qoşulmasını serverdən ayıranda server tərəfdə yerinə yetirəcəyik. **client-connect** və **client-disconnect** scriptlərini aşağıdakı hallarda istifadə edə bilərik:

- Əlavə autentifikasiya
- Firewall portlarının açılıb bağlanması
- Spesifik IP ünvanların spesifik clientlərə təyin edilməsində
- Client üçün connection-spesific sətirlərin yazılması

Bu misalda biz **client-connect** scriptini OpenVPN clientə seçilmiş mesajın ötürülməsi üçün istifadə edəcəyik hansı ki, **client-in** qoşulması gün vaxtlarına əsaslanır.

### **İşə başlayaq**

OpenVPN2.3-ü iki məşində yükləyək. Əmin olaq ki, məşinlər şəbəkədə bir-birlərini görürlər. 2-ci başlıqda yaratdığımız client və server sertifikatlarını burda istifadə edək. Bu başlıqda server məşini FreeBSD9.2

x64 OpenVPN2.3-də, client isə Windows7 OpenVPN2.3-də olacaq. Server quraşdırma faylını eynilə 1-ci misalımızda yaratdığımız **example6-1-server.conf**-dan istifadə edəcəyik.

### Necə edək...

1. **example6-1-server.conf** faylını **example6-3-server.conf** faylına nüsxələyin və **example6-3-server.conf** faylının sonuna aşağıdakı sətirləri əlavə edin:  
**script-security 2**  
**client-connect /usr/local/etc/openvpn/example6-3-connect.sh**

**Qeyd:** Öncədən (9.2: **pkg\_add -r bash** , **pkg install -y bash**) əmri ilə **bash-i** sisteme yükləməyi unutmayın.

2. İndi isə connect scriptini yaradaq(yəni **/usr/local/etc/openvpn/example6-3-connect.sh-i**):  

```
#!/usr/local/bin/bash
hour=`/bin/date +%H`
if [$hour -lt 6]
 then
 msg1="Hedden artıq yuxusuz qalirsiniz"
 elif [$hour -le 12]
 then
 msg1="Sabahınız xeyir"
 elif [$hour -lt 18]
 then
 msg1="Gunortanız xeyir"
 else
 msg1="Axshaminiz xeyir"
 fi

OPENVPN_MSG1="$msg1 $common_name"
OPENVPN_MSG2=`/bin/date +"VPN serverdəki hal-hazırki vaxt: %H:%M:%S"`

İndi əlavə quraşdırma sətirini parametr $1-ə ötürək
echo "push \"setenv-safe MSG1 '$OPENVPN_MSG1'\"" > $1
echo "push \"setenv-safe MSG2 '$OPENVPN_MSG2'\"" >> $1
```
3. Əmin olun ki, script yerinə yetiriləndir:  

```
root@siteA:/usr/local/etc/openvpn # chmod 755 example6-3-connect.sh
```
4. Serveri işə salın:  

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example6-3-server.conf
```
5. Client-in quraşdırma faylı isə öncə yaratdıqlarımıza çox oxşayır və aşağıdakı kimi olacaq:  

```
client
proto udp
remote openvpnserver.example.com
port 1194

dev tun
nobind
```

```
ca "c:/program files/openvpn/config/ca.crt"
cert "c:/program files/openvpn/config/openvpnclient2.crt"
key "c:/program files/openvpn/config/openvpnclient2.key"
tls-auth "c:/program files/openvpn/config/ta.key" 1
```

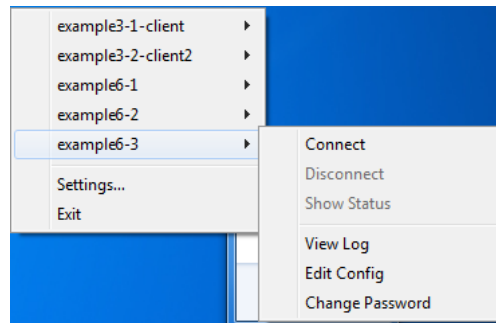
```
script-security 2
up 'C:\\Windows\\system32\\wscript.exe C:\\Program\\
Files\\OpenVPN\\scripts\\example6-3.vbs'
```

Faylı **example6-3.ovpn** adı ilə **c:\Program Files\OpenVPN\Config** ünvanında yadda saxlayın.

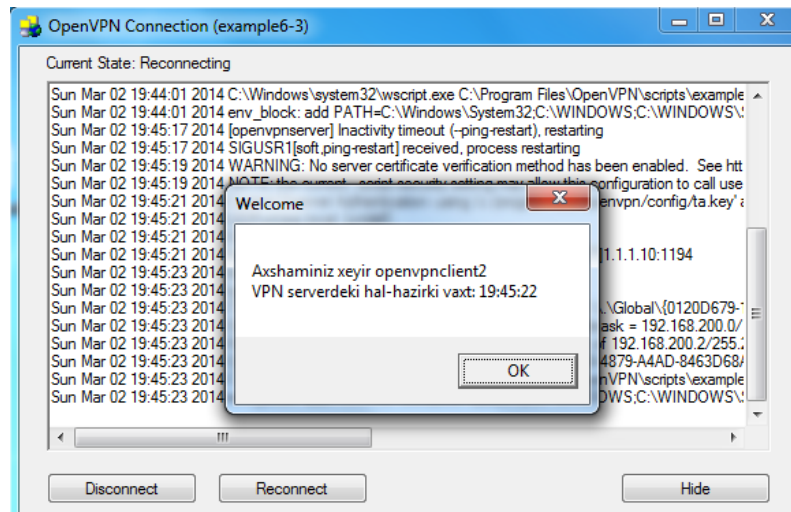
6. **C:\Program Files\OpenVPN\scripts\** ünvanında **example6-3.vbs** adlı scripti yaradıb içinə aşağıdakı sətirləri əlavə edək:

```
Set oShell = CreateObject("WScript.Shell")
msg1=oShell.ExpandEnvironmentStrings ("%OPENVPN_MSG1%")
msg2=oShell.ExpandEnvironmentStrings ("%OPENVPN_MSG2%")
MsgBox msg1 + vbCrLf + msg2, , "Welcome"
```

7. OpenVPN client-i işə salaq:



Qoşulma fazasında aşağıdakı xəbərdaredici səhifəsi ekranda çap ediləcək:



### **Bu necə işləyir...**

Client qoşulduqda OpenVPN server client-connect scriptini client qoşulması mühit dəyişənlərilə yerinə yetirir. Script qoşulma spesifikasiyalı quraşdırma faylına çıxışda iki sətiri yazır hansı ki, ilk olaraq keçir və yalnız client-connect script-i üçün parametr kimi istifadə edilir. Sonra bu quraşdırma faylı əgər normal quraşdırma faylıdırsa, OpenVPN server tərəfindən yerinə yetirilir. İstifadə edilən iki sətir isə aşağıdakılardır:

```
push "setenv-safe MSG1 '$OPENVPN_MSG1'"
push "setenv-safe MSG2 '$OPENVPN_MSG2'"
```

Bu o deməkdir ki, iki mühit dəyişəni client-ə ötürülmüşdür. Bu mühit dəyişənlərini OpenVPN client götürür və Windows VBS script vasitəsilə özündə dialog box-da ekrana çap edir.

### **Daha da ətraflı...**

Bu başlıqda biz daha çox diqqətimizi client-disconnect-ə və OpenVPN scriptlərində olan çoxlu dəyişən mühitlərinə yönləndirəcəyik.

#### **'client-disconnect' scriptləri**

Client-disconnect scripti aşağıdakı qaydadakı kimi göstərilə bilər:

```
client-disconnect /usr/local/etc/openvpn/disconnect.sh
```

Client qoşulması serverdən kəsildəndə bu script yerinə yetirilir. Bilin ki, client ilk olaraq qoşulmadan ayrılırsa və client tərəfdə **explicit-exit-notify** təyin edilməyibsə, onda OpenVPN server clientə dəfələrlə qoşulmağa çalışacaq. Əgər client bir neçə cəhd-dən sonra cavab vermirsə, onda **client-disconnect** scripti yerinə yetiriləcək. Server quraşdırmasından asılı olaraq, client-in real qoşulmadan kəsilməsi bir neçə dəqiqə ala bilər.

#### **Mühit dəyişənləri**

Client-connect və client-disconnect scriptlərində həddən artıq mühit dəyişənləri mövcuddur. Bu daha çox öyrədəcək əgər biz client scriptini aşağıdakı kimi yazsaq:

```
#!/usr/local/bin/bash
env >> /tmp/log
```

Həmçinin up və down scriptində olduğu kimi, mühit dəyişəni **script\_type**-da olur, script tipini təyin edir və server quraşdırma faylında təyin edilir. Bu server administrator opsiya verir ki, eyni script-də həm **client-connect** və həm də **client-disconnect** yazıla bilsin.

#### **Tam ünvan**

Qeyd edin ki, script üçün tam ünvan istifadə edilir. Asılı olan ünvanlarda təyin edilə bilər ancaq, əsasən OpenVPN serverin özündə. Tam ünvanın istifadə edilməsi daha təhlükəsizdir. OpenVPN serverin həmişə eyni ünvandan işə düşməsinə nəzərə alaraq bu pis praktikadır. Alternativ olaraq aşağıdakı sintaksis istifadə edilə bilər:

```
cd /usr/local/etc/openvpn
client-connect example6-3-connect.sh
```

## 'learn-address' scriptinin istifadə edilməsi

Bu misalda biz clientə qoşulmada ünvanında dəyişməsi halında learn-address scriptinin server tərəfdə istifadə edilməsini göstərəcəyik. Learn-address scriptləri seçilmiş clientlər üçün dinamik firewall rule-larının yada spesifik routing table-in yazılması üçün istifadə edilə bilər.

Bu misalda biz client üçün learn-address scriptini firewall-in açılması və masqueradingin istifadə edilməsi üçün istifadə edəcəyik. Client qoşulmadan kəsilən kimi firewall bağlanır və masquerade rule-u silinir.

### İşə hazırlaşaq

OpenVPN2.3 və ya daha böyük versiyasını 2 məşində yükləyin. Əmin olun ki, maşınlar şəbəkə ilə bir-birlərini görürlər. 2-ci başlıqda yaradılmış client və server sertifikatları burda da istifadə ediləcək. Bu misalda biz server maşını FreeBSD9.2 x64 OpenVPN2.3 və client maşını Windows7 x64 OpenVPN2.3-dən istifadə edəcəyik. Client üçün 2-ci başlıqda olan 'ifconfig-pool' block-da istifadə elədiyimiz **basic-udp-client.ovpn** quraşdırma faylından istifadə edəcəyik.

### Necə edək...

1. Server quraşdırma faylını yaradaq:

```
proto udp
port 1194
dev tun

server 192.168.200.0 255.255.255.0

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnsrvr.crt
key /usr/local/etc/openvpn/openvpnsrvr.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0

persist-key
persist-tun
keepalive 10 60

topology subnet

daemon
log-append /var/log/openvpn.log
script-security 2
learn-address /usr/local/etc/openvpn/example6-4-learn-address.sh
push "redirect-gateway def1"
```

Yuxarıda yazdığımız sətirləri **example6-4-server.conf** faylına əlavə edib yadda saxlayın. Nəzərə alın ki, yuxarıdakı quraşdırma sətirlərində **user nobody** və **group nobody** sətirləri yoxdur.

2. Sonra isə Linux üçün **/etc/openvpn/example6-4-learn-address.sh** scriptini yaradın:  
Linux Server üçün IPTABLES aşağıdakı kimi olacaq:

```
#!/bin/bash
$1 = action (add, update, delete)
$2 = IP or MAC
$3 = client_common name
if ["$1" = "add"]
then
 /sbin/iptables -I FORWARD -i tun0 -s $2 -j ACCEPT
 /sbin/iptables -I FORWARD -o tun0 -d $2 -j ACCEPT
 /sbin/iptables -t nat -I POSTROUTING -s $2 -o wlan0 -j
 MASQUERADE
elif ["$1" = "delete"]
then
 /sbin/iptables -D FORWARD -i tun0 -s $2 -j ACCEPT
 /sbin/iptables -D FORWARD -o tun0 -d $2 -j ACCEPT
 /sbin/iptables -t nat -D POSTROUTING -s $2 -o wlan0 -j
 MASQUERADE
fi
```

FreeBSD maşın üçün isə öncədən kernel aşağıdakı opsiyalarla quraşdırmalısınız.

```
PF FireWall
device pf
device pflog
device pfsync
```

Sonra isə startupa lazımı sətirləri əlavə etməlisiniz (Yeni `/etc/rc.conf-a`).

```
PF Service
pf_enable="YES"
pf_rules="/etc/pf.conf"
pflog_enable="YES"
pflog_logfile="/var/log/pflog"
```

Sonra isə `/etc/pf.conf` faylına aşağıdakı sətirləri əlavə etmək lazımdır.

```
clientlər cədvəlinə 10.100.100.100 IP ünvanı CLI-dan əlavə edirik.
pfctl -t clientlər -T add 10.100.100.100
pfctl -t clientlər -T show - Burda isə clientlər cədvəlinin içinə baxırıq
pfctl -t invpnusers -T delete 10.100.100.100
ext_if="em0"
ext_ip="1.1.1.10"
vpn_if="tun0"
table <clientlər> { }
table <invpnusers> { }
table <outvpnusers> { }

rdr on $vpn_if from <clientlər> to any -> $ext_if

pass in on tun0 from <invpnusers> to any
pass out on tun0 from <outvpnusers> to any
```

Ardınca da FreeBSD üçün `/usr/local/etc/openvpn/example6-4-learn-address.sh` scriptini yaradaq:

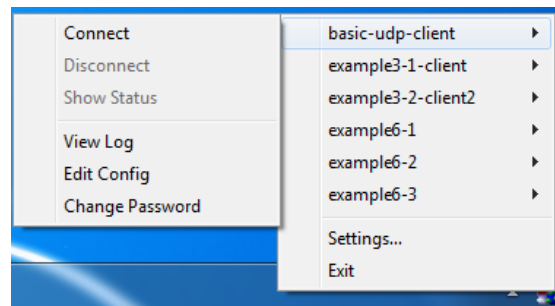
```
#!/usr/local/bin/bash
$1 = action (add, update, delete)
$2 = IP or MAC
$3 = client_common name
clientlər cədvəlinə 10.100.100.100 IP ünvanı əlavə CLI-dan
əlavə edirik.
pfctl -t clientlər -T add 10.100.100.100
pfctl -t clientlər -T show - Burda isə clientlər cədvəlinin
içinə baxırıq

if ["$1" = "add"]
then
 /sbin/pfctl -t invpnusers -T add $2
 /sbin/pfctl -t outvpnusers -T add $2
 /sbin/pfctl -t clientlər -T add $2
elif ["$1" = "delete"]
then
 /sbin/pfctl -t invpnusers -T delete $2
 /sbin/pfctl -t outvpnusers -T delete $2
 /sbin/pfctl -t clientlər -T delete $2
fi
```

3. `example6-4-learn-address.sh` scriptini yerinə yetirilən edib OpenVPN serveri işə salın.

```
root@siteA:/usr/local/etc/openvpn # chmod 755
/usr/local/etc/openvpn/example6-4-learn-address.sh
root@siteA:/usr/local/etc/openvpn # openvpn --config example6-4-
server.conf
```

4. Windows GUI-dən və `basic-udp-client.conf`-dan istifadə edərək client-i işə salaq:



5. Client serverə qoşulduqdan sonra həm Linux və həm də FreeBSD maşında statuslara baxaq:

```
Linux maşında:
[root@server]# iptables -L FORWARD -n -v
Chain FORWARD (policy ACCEPT 4612K packets, 1761M bytes)
pkts bytes target prot opt in out source
destination
0 0 ACCEPT all -- * tun0 0.0.0.0/0
192.168.200.2
```

```
0 0 ACCEPT all -- tun0 * 192.168.200.2
0.0.0.0/0
```

```
[root@server]# iptables -t nat -L POSTROUTING -n -v
Chain POSTROUTING (policy ACCEPT 336K packets, 20M bytes)
pkts bytes target prot opt in out source
destination
0 0 MASQUERADE all -- * wlan0 192.168.200.2
0.0.0.0/0
```

FreeBSD maşında:

TRANSLATION RULES:

```
rdr on tun0 inet from <clientler> to any -> 1.1.1.10
```

FILTER RULES:

```
pass in on tun0 from <invpnusers> to any flags S/SA keep state
pass out on tun0 from <outvpnusers> to any flags S/SA keep state
```

6. Client-in qoşulmasını ayırın, bir neçə dəqiqə gözləyin və sonra həm Linux-Iptables və həm FreeBSD-PF rulelarını yoxlayın ki, həqiqətən siliniblər (Misal üçün FreeBSD-də `/var/log/openvpn.log` faylında aşağıdakı sətirlər olacaq):  

```
1/1 addresses deleted.
1/1 addresses deleted.
1/1 addresses deleted.
```

### Bu necə işləyir...

Client OpenVPN-ə qoşulduqda və ya ayrıldıqda, OpenVPN server `learn-address` scriptini müxtəlif CLI arqumentləri ilə yerinə yetirir:

- **\$1**: İş (add, update, delete)
- **\$2**: IP yada MAC. Tun bazalı şəbəkələr üçün bu cliet IP ünvanıdır. Tap bazalı şəbəkələr üçün isə bu clientin MAC (virtual) ünvanıdır.
- **\$3**: `client_common` name

Bu başlıqda `learn-address` scripti istifadə edilir ki, qoşulmadan sonra client üçün firewall-da `access` və `nat` rule-larını əlavə edək və qoşulma bitdikdən sonra həmin rule-ları silək.

### Daha da ətraflı...

Aşağıdakı seksiyada `user nobody` direktivinin istifadəsi və `update` işinin `learn-address` scriptində istifadəsini açıqlayacağıq.

#### User 'nobody'

Öncə dediyimiz kimi server quraşdırmasında aşağıdakı sətirlər yox idi:

```
user nobody
group nobody
```



(bəzi Linux distributivlərində bu **nogroup** olur). Əgər biz bu sətirləri əlavə etmişiksə, onda OpenVPN server öz prosesini **nobody** istifadəçi adından işə salmağa çalışacaq. Ancaq bu istifadəçinin FireWall-da port açıb bağlamaq üçün yetkisi yoxdur(Hətta bu misalda rule-lar silindi).

### 'update' işinin görülməsi

OpenVPN server həmçinin client tərəfdə ünvan dəyişəndə onu yeniləyə bilər. Bu əsasən TAP bazalı şəbəkələrdə olur hansı ki, external DHCP serverdən istifadə edilir. Həmin halda learn-address scripti ya routing cədvəlini ya da firewall rule-larını qaydaya salır.

### 'tls-verify' scriptinin istifadə edilməsi

OpenVPN-in müxtəlif səviyyələri var hansı ki, qoşulan client-in verilənləri yoxlanılır. Həmçinin əlavə bir səviyyə mövcuddur ki, **tls-verify** scripti ilə başqa bir yoxlanış edə bilərsiniz. Bu misalda biz göstərəcəyik ki, script sayəsində seçilmiş sertifikat üçün girişə yetki verə bilərik.

### İşə başlayaq

OpenVPN2.3 ya da daha yüksək versiyasını iki maşında yükləyək. Əmin olun ki, maşınlar şəbəkə ilə bir-birlərini görürlər. 2-ci başlıqda yaratdığımız client və server sertifikatlarını burdada istifadə edək. Bu misalda da həmişə olduğu kimi server maşını FreeBSD9.2 x64 OpenVPN2.3-də client maşını isə Windows7 OpenVPN2.3-də olacaq. Client quraşdırma faylı **basic-udp-client.ovpn** olaraq saxlayın hansı ki, 2-ci başlıqda **'ifconfig-pool' block**-da istifadə eləmişdik.

### Necə edək...

1. Server quraşdırma faylını yaradaq:

```
proto udp
port 1194
dev tun

server 192.168.200.0 255.255.255.0

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnserver.crt
key /usr/local/etc/openvpn/openvpnserver.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0

persist-key
persist-tun
keepalive 10 60

topology subnet

user nobody
group nobody
daemon
log-append /var/log/openvpn.log

script-security 2
```

```
tls-verify "/usr/local/etc/openvpn/verify-cn
/usr/local/etc/openvpn/valid-CNs"
```

Faylı **example6-5-server.conf** adında yadda saxlayaq.

2. Sonra isə `tls-verify` üçün **verify-cn** scriptini `/usr/local/etc/openvpn` ünvanına nüsxələyək:

```
root@siteA:/usr/local/etc/openvpn # cp
/usr/local/share/examples/openvpn/sample-scripts/verify-cn .
```

**verify-cn** scriptinin tərkibi aşağıdakı kimi olacaq (Qeyd edin ki, perl öncədən sistemdə yüklənmiş olmalıdır):

```
#!/usr/bin/perl
```

```
die "usage: verify-cn cnfile certificate_depth subject" if (@ARGV !=
3);
($cnfile, $depth, $x509) = @ARGV;
```

```
if ($depth == 0) {
 if ($x509 =~ / CN=([^\,]+)/) {
 $cn = $1;
 open(FH, '<', $cnfile) or exit 1; # can't open, nobody
 authenticates!
 while (defined($line = <FH>)) {
 if ($line !~ /^[[:space:]]*(#|$)/o) {
 chop($line);
 if ($line eq $cn) {
 exit 0;
 }
 }
 }
 close(FH);
 }
 exit 1;
}
exit 0;
```

3. Sonda, izin verilən sertifikatların siyahısını yaradın (Öncə **openvpnclient2.crt** haqqında məlumat əldə edirsiniz və sonra da onu lazımi fayla əlavə edirsiniz):

```
root@siteA:/usr/local/etc/openvpn # openssl x509 -subject -noout -in
openvpnclient2.crt
subject= /C=AZ/O=Itvpn/CN=openvpnclient2/emailAddress=openvpn-
ca@domain.lan
```

Görünən öncəki sətirdə `client-in` CN-i **openvpnclient2-dir**

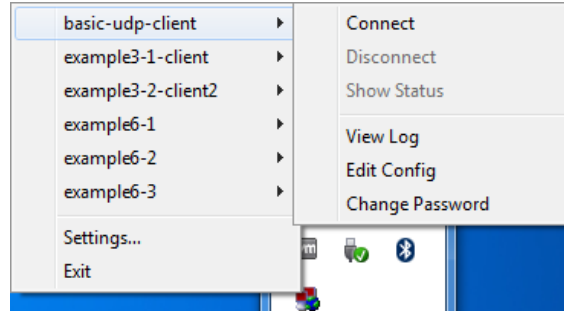
Lazımi CN-i inamlı faylımıza əlavə edək. CN-ləri ardıcıl hər sətirdə yazmaq lazımdır.

```
root@siteA:/usr/local/etc/openvpn # echo "openvpnclient2" >>
/usr/local/etc/openvpn/valid-CNs
```

4. OpenVPN server işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example6-5-server.conf
```

5. **basic-udp-client.ovpn** quraşdırmasını istifadə edərək client-i Windows GUI ilə işə salın:



Client problemsiz qoşulmalıdır.

6. İndi OpenVPN serverdə **/usr/local/etc/openvpn/valid-CNs** faylında **openvpnclient2**-nin qarşısına şərh qoşub aşağıdakı şəkildə edək və client ilə yenidən qoşulmağa çalışaq:

```
openvpnclient1
#openvpnclient2
```

Bu dəfə serverdə client üçün jurnallar aşağıdakı kimi olacaq:

```
Mon Mar 3 22:27:05 2014 2.2.2.10:63329 TLS_ERROR: BIO read
tls_read_plaintext error: error:140890B2:SSL
routines:SSL3_GET_CLIENT_CERTIFICATE:no certificate returned
Mon Mar 3 22:27:05 2014 2.2.2.10:63329 TLS Error: TLS object ->
incoming plaintext read error
Mon Mar 3 22:27:05 2014 2.2.2.10:63329 TLS Error: TLS handshake failed
```

### **Bu necə işləyir...**

Client, OpenVPN serverə qoşulmağa çalışanda **tls-verify** scripti bu sertifikat zəncirini bir neçə dəfə yoxlanış edir. Bizim misalda biz **openvpnclient2.crt** faylını yoxlanış edirdik hansı ki, sertifikatın CN-nini öncədən **/usr/local/etc/openvpn/valid-CNs** faylına əlavə etmişdik. Əgər lazımi CN faylda şərhə tapılarsa, istifadəçi uğurla qeydiyyatdan keçəcək. Digər hallarda bütün istifadəçilərə giriş qadağan olacaq. **tls-verify** scripti ilə bütövlükdə bir CA server üçün bütün sertifikatları bağlamaq olar. Elə hallar ola bilər ki, bir client sertifikatı bir neçə CA tərəfindən imzalana bilər. Bu hallarda müəyyən bir CA-nı siz bütövlükdə bağlaya bilərsiniz.

Misal olaraq aşağıdakı script istifadə edilə bilər:

```
#!/bin/bash
[$# -lt 2] && exit 1
CA=`echo $2 | sed -n 's/.*\|CN=(.*)\|.*\|1/p`
["$CA" = "Itvpn CA"] && exit 1
```

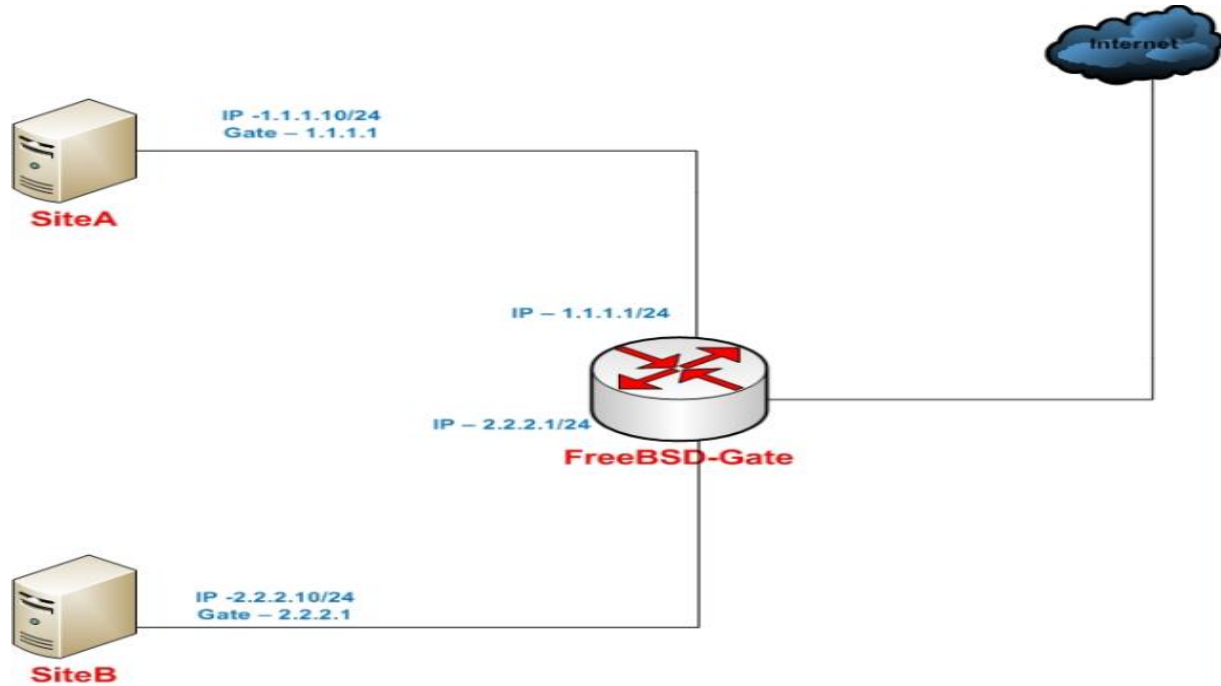
## 'auth-user-pass-verify' scriptinin istifadə edilməsi

Sertifikat və private açarlara əlavə olaraq, OpenVPN həmçinin clientlərin yoxlanışı üçün istifadəçi və şifrəli olan mexanizmi də dəstəkləyir. Bu misalda biz **auth-user-pass-verify** scriptinin necə mənimsədilməsini göstərəcəyik. Bu script istifadəçinin verilənlər bazasında ya da faylda yoxlanışı üçün istifadə edilə və həmçinin şifrənin düzgün təyin edilməsi üçün yoxlana bilər.

### İşə başlayaq

OpenVPN2.3-ü 2 məşində yükləyək. Əmin olun ki, maşınlar şəbəkə ilə bir-birlərini görürlər. 2-ci başlıqda olan client və server sertifikatlarını burdada istifadə edəcəyik. Bu misalımızda server və client maşınlarımız FreeBSD9.2 x64 OpenVPN2.3-də işləyəcək. Server quraşdırma olaraq Client tərəf up/down scriptlərində istifadə etdiyimiz **example6-1-server.conf** faylından burdada istifadə edəcəyik.

Topologiyamız aşağıdakı kimi olacaq:



Bunu necə edək...

1. **example6-1-server.conf** faylını **example6-6-server.conf** faylına nüsxələyin və **example6-6-server.conf** faylına aşağıdakı sətirləri əlavə edin:  

```
script-security 2
auth-user-pass-verify /usr/local/etc/openvpn/example6-6-aupv.sh via-file
```
2. **auth-user-pass-verify** scriptini yaradaq (Yəni `/usr/local/etc/openvpn/example6-6-aupv.sh` faylını yaradaq):  

```
#!/usr/local/bin/bash
```

```
the username+password is stored in a temporary file
pointed to by $1
username=`head -1 $1`
password=`tail -1 $1`

if grep "$username:$password" $0.passwd > /dev/null 2>&1
then
 exit 0
else
if grep "$username" $0.passwd > /dev/null 2>&1
then
 echo "auth-user-pass-verify: Wrong password entered for user
'$username'"
else
 echo "auth-user-pass-verify: Unknown user '$username'"
fi
 exit 1
fi
```

3. Çox təhlükəli olan şifrə faylını yaradaq:

```
root@siteA:/usr/local/etc/openvpn # echo "openvpn:openvpn" >
/usr/local/etc/openvpn/example6-6-aupv.sh.passwd
```

4. Əmin olun ki, **auth-user-pass-verify** scripti yerinə yetiriləndir və sonra serveri işə salın:

```
root@siteA:/usr/local/etc/openvpn # chmod 755
/usr/local/etc/openvpn/example6-6-aupv.sh
root@siteA:/usr/local/etc/openvpn # openvpn --config example6-6-
server.conf
```

5. Sonra isə SiteB-də clientin **/usr/local/etc/openvpn/example6-6-client.conf** quraşdırma faylını yaradın:

```
client
proto udp
remote openvpnsver.example.com
port 1194

dev tun
nobind

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnclient1.crt
key /usr/local/etc/openvpn/openvpnclient1.key
tls-auth /usr/local/etc/openvpn/ta.key 1

ns-cert-type server

auth-user-pass
```

6. Client-i işə salın:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config example6-6-
client.conf
```

```
Mon Mar 3 23:32:13 2014 OpenVPN 2.3.2 amd64-portbld-freebsd9.2 [SSL
(OpenSSL)] [LZO] [eurephia] [MH] [IPv6] built on Jan 9 2014
```

7. İlk olaraq OpenVPN client istifadəçi adı və şifrə istəyəcək(openvpn istifadəçi və şifrə ilə qoşulma uğurlu olmalıdır):

```
Enter Auth Username:openvpn
Enter Auth Password:openvpn
```

8. Sonra isə səhv login və şifrə ilə qoşulmağa çalışın:

```
Enter Auth Username:newuser
Enter Auth Password:newpass
```

Server jurnalı aşağıdakı kimi olmalıdır:

```
auth-user-pass-verify: Unknown user 'newuser'
Mon Mar 3 23:33:55 2014 2.2.2.10:55303 WARNING: Failed running command
(--auth-user-pass-verify): external program exited with error status: 1
Mon Mar 3 23:33:55 2014 2.2.2.10:55303 TLS Auth Error: Auth
Username/Password verification failed for peer
```

Nəzərə alın ki, bu metodla siz çoxlu istifadəçini həm sertifikat və üstündən login və şifrə ilə qeydiyyatla ala bilərsiniz.

### **Bu necə işləyir...**

OpenVPN client ilk qoşulmada istifadəçi adı və şifrəni daxil edir. Nəzərə alın ki, şifrə hər bir halda serverə şifrlənmiş kanalla ötürülür ancaq, şifrənin özü hash yada crypt edilmiş halda olmur. Server tərəfdə isə **auth-user-pass-verify** scripti isə istifadəçi adı və şifrəni bir sətirdə yoxlanışa yollayır. Script isə sonra istifadəçi və şifrə faylında istifadəçi adı və şifrənin düzgünlüyünü yoxlayır. Əgər düzdürsə, script **0** code ilə uğurla çıxış edir. Əks halda **1** code ilə çıxış edib client qoşulmasını kəsir.

### **Daha da ətraflı...**

Aşağıdakı seksiya ilə biz bəzi detalları görəcəyik ki, necə şifrə təyin edilə bilər və necə server tərəfdən **auth-user-pass-verify** scriptinə ötürülə bilər.

### **Client tərəfdə istifadəçi adı və şifrəni təyin edək.**

OpenVPN-in opsiyası vardır hansı ki, istifadəçi adı və şifrəni client tərəfdə təyin etmək olur. Bunun üçün, OpenVPN spesifik flagla kompilyasiya edilməlidir. Adi halda bu flag aktiv olmur və lazımi opsiyanı

**auth-user-pass /usr/local/etc/openvpn/password-file** kimi təyin elədikdə, OpenVPN client aşağıdakı sətirləri göstərərək qoşulmur:

```
... Sorry, 'Auth' password cannot be read from a file
... Exiting
```

Nəzərə alın ki, şifrənin client tərəfdə saxlanması təhlükəsiz deyil(açıq şəkildə). Ona görə də susmaya görə bu opsiyanın sönülü olması daha yaxşı üsuldür. Ancaq OpenVPN-in man səhifəsi bu imkanın olmasını bizə açıqlayır.

### **Mühit dəyişənləri ilə şifrənin keçidi**

Bu misalda biz istifadə etdik:

```
auth-user-pass-verify example6-6-aupv.sh via-file
```

Biz OpenVPN serverin istifadəçi adı və şifrə yoxlanışını adi fayl ilə elədik. Bu adi fayl yalnız server prosesi tərəfindən istifadə edilə bilər həmçinin, bu təhlükəsiz mexanizmdir ki, şifrəni şifrələnmiş kanalla **auth-user-pass-verify** scripti ilə ötürürük.

Həmçinin mümkündür ki, istifadəçi adı və şifrəni **auth-user-pass-verify** scripti ilə mühit dəyişənləri ilə ötürə bilək:

```
auth-user-pass-verify example6-6-aupv.sh via-env
```

Bunun üstünlüyü ondan ibarətdir ki, heç bir əlavə fayl yaratmağa gerek qalmır. Pis cəhəti odur ki, şifrəni mühit dəyişənləri ilə açıq formada ötürmək daha təhlükəlidir: Qismən digər prosesin mühitinə baxmaq, digər istifadəçi adına olan fayla baxmaqdan asandır.

### **Script ardıcılığı**

OpenVPN serverdə yerinə yetirilən scriptlərin içində vaciblik və üstünlük dərəcəsinə görə ardıcılıqla yerinə yetirilmənin öncədən təyin edilməsi çox önəmlidir. Bu misalda biz CLI-dan scriptlərin hər birinə aid olan ardıcılığı lazımı parametrlərlə edəcəyik.

### **İşə başlayaq**

OpenVPN2.3-ü 2 məşində yükləyək. Əmin olun ki, maşınlar şəbəkə ilə bir-birlərini görürlər. 2-ci başlıqda olan client və server sertifikatlarını burada da istifadə edəcəyik. Bu misalımızda server və client maşınlarımız FreeBSD9.2 x64 OpenVPN2.3-də işləyəcək. Server quraşdırma olaraq Client tərəf up/down scriptlərində istifadə etdiyimiz **example6-1-server.conf** faylından burdada istifadə edəcəyik. Client üçün quraşdırma olaraq, bundan öncəki misalımızda olanlardan istifadə edəcəyik.

### **Necə edəcəyik...**

1. **example6-1-server.conf** faylını **example6-7-server.conf** faylına nüsxələyin və **example6-7-server.conf** faylının sonuna aşağıdakı sətirləri əlavə edin:

```
script-security 2
cd /usr/local/etc/openvpn
up example6-7-script.sh
route-up example6-7-script.sh
down example6-7-script.sh
client-connect example6-7-script.sh
client-disconnect example6-7-script.sh
learn-address example6-7-script.sh
tls-verify example6-7-script.sh
auth-user-pass-verify example6-7-script.sh via-env
```
2. **/usr/local/etc/openvpn/example6-7-script.sh** scriptini yaradıb içinə aşağıdakı sətirləri əlavə edin:

```
#!/usr/local/bin/bash
exec >> /tmp/example6-7.log 2>&1
date +"%H:%M:%S: START $script_type script ==="
```

```
echo "argv = $0 $@"
echo "user = `id -un`/`id -gn`"
date +"%H:%M:%S: END $script_type script ==="
```

3. Əmin olun ki, script yerinə yetiriləndir və serveri işə salın:  
root@siteA:/usr/local/etc/openvpn # **chmod 755 example6-7-script.sh**  
root@siteA:/usr/local/etc/openvpn # **openvpn --config example6-7-server.conf**

4. Sonra clienti işə salın:  
root@siteB:/usr/local/etc/openvpn # **openvpn --config example6-6-client.conf**

Daxil olmaq üçün istifadəçi adı və şifrə istənilən istifadə edilə bilər(Çünki onlar istifadə edilmir ☺).

5. Serverə uğurlu qoşulduqdan sonra client-i şəbəkədən ayırın və bir neçə dəqiqə gözləyin o vaxtadək ki, server clientin qoşulmasının qırılmasını təyin edir. Sonra OpenVPN serveri dayandırın.

6. Jurnal faylı **/tmp/example6-7.log** ünvanında aşağıdakı göstərilədiyi kimi yaranacaq:

```
root@siteA:/usr/local/etc/openvpn # cat /tmp/example6-7.log
09:08:57: START up script ===
argv = example6-7-script.sh tun0 1500 1541 192.168.200.1 255.255.255.0
init
user = root/wheel
09:08:57: END up script ===
09:08:57: START route-up script ===
argv = example6-7-script.sh
user = root/wheel
09:08:57: END route-up script ===
10:44:57: START up script ===
argv = example6-7-script.sh tun0 1500 1541 192.168.200.1 255.255.255.0
init
user = root/wheel
10:44:57: END up script ===
10:44:57: START route-up script ===
argv = example6-7-script.sh
user = root/wheel
10:44:57: END route-up script ===
```

### **Bu necə işləyir...**

OpenVPN-ə əlavə edilmiş çox məqsədlərdə istifadə edilə bilən ssenarilər mövcuddur. OpenVPN işə düşəndə və sonra client ona qoşulub ayrılanda bu scriptlər ardıcıl olaraq işə düşür. OpenVPN2.3 üçün ardıcıl qaydalar aşağıdakılardır:

- **root** istifadəçi adından işə düş:
- **route-up** həmçinin root istifadəçi adından işə düşür; root yetkilər dayandırılır və OpenVPN nobody istifadəçi adından server quraşdırma faylından işə düşür.



- **tls-verify**. CA sertifikatı istifadə edilir ki, imzalanmış client sertifikatını yoxlanışdan keçirsin.
- **tls-verify**. Client sertifikatın özü ötürülür
- **user-pass-verify**.
- **client-connect**.
- **learn-address** isə **add** işi ilə.

Bu nöqtədə client uğurla VPN qoşulmasını edir. İndi isə client disconnect olanda:

- **client-disconnect**
- **learn-address** isə **delete** işi ilə

Və sonra server dayanır:

- **down;** qeyd bu **nobody!** İstifadəçi adından işə düşür

### Daha da ətraflı...

Scripti yazanda unutmayın ki, onu mütləq yerinə yetirilən eləmək lazımdır. OpenVPN-in dizaynı çox monolitikdir: hər şey (plugunləri çıxmaq şərtilə hansı ki, bu başlıqda birazdan danışacağıq) bir axının üstündə gedir. Bu o deməkdir ki, script yerinə yetiriləndə OpenVPN server müvəqqəti olaraq istənilən müştəri üçün çatılmaz olur: paketlərin routingi dayandırılır, digər clientlər qoşula, qırıla bilmir və hətta management interfeys cavab vermir. Ona görə də əmin olmaq lazımdır ki, bütün server tərəf scriptləri çox tez işə düşür.

Dizayn çatışmamazlığı qəbul edilib ancaq, OpenVPN3-ədək bunun düzəldiləcəyi güman edilmir.

### **Script təhlükəsizliyi və jurnallama**

OpenVPN-in 2.3 versiyasının əsas üstünlüyü odur ki, scriptlər işə düşəndə onun təhlükəsizliyinə çox baxır. OpenVPN2.0 versiyasında bütün scriptlər yerinə yetiriləndə **'system'** server olan bütün mühit dəyişənləri hər bir scriptlə götürülürdü. OpenVPN2.1-dən etibarən bu dəyişdi və **script-security** quraşdırma direktivi əlavə edilərək scriptlərin yerinə yetirilməsində **execv** istifadə edildi. Bundan əlavə təhlükəsizlik məqsədlərilə scriptinizin çıxışını jurnallamanız daha yaxşı olar. Script jurnallamasında çıxış həmçinin vaxt möhürlərini də əlavə edir ki, bu sizə problemin hansı vaxtda üzrə çıxmasına kömək edir.

Bu misalda biz script-security direktivinin çoxlu opsiyalarına baxacağıq və scriptin çıxışının daha tez jurnallama metodunu örgənəcəyik.

### **İşə hazırlaşaq**

OpenVPN2.3-ü 2 maşında yükləyək. Əmin olun ki, maşınlar şəbəkə ilə bir-birlərini görürlər. 2-ci başlıqda olan client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalımızda server və client maşınlarımız FreeBSD9.2 x64 OpenVPN2.3-də işləyəcək. Server quraşdırma olaraq Client tərəf up/down scriptlərində istifadə etdiyimiz **example6-1-server.conf** faylından

burdada istifadə edəcəyik. Client üçün quraşdırmaları isə bundan öncəki misalımızda olanlardan istifadə edəcəyik.

### Bunu necə etməliyik...

1. OpenVPN server ilk misalımızda istifadə elədiyimiz quraşdırma ilə işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example6-1-server.conf
```

2. Client quraşdırma faylını yaradın:

```
client
proto udp
remote openvpnservers.example.com

port 1194

dev tun
nobind

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnclient1.crt
key /usr/local/etc/openvpn/openvpnclient1.key
tls-auth /usr/local/etc/openvpn/ta.key 1
```

```
ns-cert-type server
```

```
up "/usr/local/etc/openvpn/example6-8-up.sh arg1 arg2"
```

Yuxarıdakı sətirləri `/usr/local/etc/openvpn/example6-8-client.conf`adında yadda saxlayın. Gördüyünüz kimi `script-security` sətiri yoxdur.

3. Up scriptini yaradın.

```
#!/usr/local/bin/bash
exec >> /usr/local/etc/openvpn/example6-8.log 2>&1
date +"%H:%M:%S: START $script_type script =="
echo "argv = [$0] [$1] [$2] [$3] [$4]"
/usr/local/bin/pstree $PPID
date +"%H:%M:%S: END $script_type script =="
```

Ancaq unutmayın script-də `pstree` əmri istifadə edilmişdir və bu susmaya görə FreeBSD-də olmur və onu yükləmək lazımdır.

```
cd /usr/ports/sysutils/pstree # Port ünvanına daxil oluruq
make install clean # Yükləyirik
```

Scripti `/usr/local/etc/openvpn/example6-8-up.sh` adında yadda saxlayın və əmin olun ki, script yerinə yetiriləndir. (`chmod +x /usr/local/etc/openvpn/example6-8-up.sh`)

4. OpenVPN client işə salın:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config example6-8-client.conf
```

Client qoşulmağa çalışacaq ancaq, scriptin OpenVPN tərəfindən yerinə yetirilməsinə izin verilmədiyinə görə aşağıdakı səhv çap ediləcək.

```
Tue Mar 4 22:56:24 2014 WARNING: External program may not be called
unless '--script-security 2' or higher is enabled. See --help text or
man page for detailed info.
Tue Mar 4 22:56:24 2014 WARNING: Failed running command (--up/--down):
external program fork failed
Tue Mar 4 22:56:24 2014 Exiting due to fatal error
```

5. Əgər biz clientin **example6-8-client.conf**-na **--script-security 2** parametri əlavə edib işə salsaq o uğurla qoşulacaq:
- ```
root@siteB:/usr/local/etc/openvpn # openvpn --config example6-8-
client.conf --script-security 2
```

Jurnal faylında aşağıdakı sətirləri görəəcəyik:

```
root@siteB:/usr/local/etc/openvpn # cat
/usr/local/etc/openvpn/example6-8.log
23:09:48: START up script ===
argv = [/usr/local/etc/openvpn/example6-8-up.sh] [arg1] [arg2] [tun0]
[1500]
-+- 01281 root openvpn --config example6-8-client.conf --script-
security 2
 \+- 01285 root /usr/local/bin/bash /usr/local/etc/openvpn/example6-8-
up.sh arg1 arg2 tun0 1500 1541 192.168.200.2 255.255.255.0 init
  \+- 01288 root /usr/local/bin/pstree 1281
    \--- 01289 root ps -axwwo user,pid,ppid,pgid,command
23:09:48: END up script ===
```

Əgər biz öncəki scripti **--script-security 2 system** və ya **--script-security 3** ile yerinə yetirsək eyni nəticə əldə etmiş olacağıq.

Bu necə işləyir...

Qayda ilə scriptləri client ya da serverdə yerinə yetirmək üçün script-security2 (ya da 3) direktivi təyin edilməlidir. Digər halda OpenVPN2.3 və daha yuxarı versiyalar qoşulmanın qarşısını almış olacaq. Aşağıdakı parametrlər **script-security** direktivi üçün təyin edilə bilər:

- **0:** External program çağırıla bilməz. Bu o deməkdir ki, OpenVPN uğurla başlaya bilməz, Microsoft Windows-u çıxmaq şərti ilə xüsusi şərtlərdə.
- **1:** Yalnız daxili proqramlar (Hansı ki, /sbin/ifconfig, /sbin/ip Linux-da, netsh.exe və route.exe Windows-da) çağırıla bilər.
- **2:** Daxili proqramlar və scriptlər çağırıla bilər.
- **3:** Eyni 2-dəki kimi, ancaq burda şifrələr mühit dəyişənlərinə scriptlərlə ötürülə bilər.

script-security direktivi üçün ikinci parametr:

- **execve:** Bu çağırılış ilə kənar proqram çağırılır. Susmaya görədir.
- **system:** System çağırılışından istifadə edərək kənar proqramlar çağırılır.

Bunlar arasında fərq böyük deyil ancaq, faylların ünvanlarının adlarında boşluqlar istifadə edilirsə ciddi fərqləri var və bizim köməyimizə çatır.

Daha da ətraflı...

Linux/BSD/MacOS və Windows scriptlərinin yerinə yetirilməsində müəyyən fərqlər var. Windows-da system çağırışı CreateProcess-i susmaya görə istifadə edir. Əgər **script-security 2 system** istifadə edilirsə, system çağırışından istifadə edilir. Böyük fərq dediyimiz kimi faylların ünvanlarının adlarında boşluqlar olanda hiss edilir. Misal üçün aşağıdakı script yalnız **--script-security 2** istifadə edilən halda işə düşəcək:

```
up "c:\\program\ files\\openvpn\\scripts\\example6-8-up.bat"
```

O halda ki, **--script-security 2 system** istifadə ediləcək onda aşağıdakı səhv çap ediləcək:

```
c:\program' is not recognized as an internal or external command
```

'down-root' pluginin istifadə edilməsi

OpenVPN plugin arxitekturu dəstəkləyir hansı ki, external pluginlər vasitəsilə OpenVPN-in funksionallığı artırıla bilər. Pluginlər spesifik modullar və ya kitabxanalardır hansı ki, OpenVPN plugin API ilə işləyirlər. Bu pluginlərdən biri **down-root**-dur hansı ki, Linux-da mövcuddur. Bu istifadəçiyə izin verir ki, OpenVPN dayananda spesifik əmrləri root istifadəçisi adından işə salmaq olsun. Normal halda OpenVPN prosesi root yetkiləri drop edir (əgər **-user** direktivi istifadə edilirsə) təhlükəsizlik səbəblərinə görə. Baxmayaraq ki, bu daha təhlükəsizdir ancaq, bəzi hallarda up scriptinin yerinə yetirilməsində bu bizim işimizi çox çətinləşdirə bilər (harda ki root adından nəse işə salmağa çalışsaq). Bundan ötrü down-root plugini hazırlanmışdır. Bu misal göstərəcək ki, necə down-root plugini istifadə edilə bilər ki, up scripti ilə yaradılan faylı silmək olsun.

İşə hazırlaşaq

2-ci başlıqda yaratdığınız server sertifikatlarını burda da istifadə edəcəyik. Bu misalda server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Client maşına ehtiyac yoxdur.

Necə edək...

1. Öncədən lazımı plugini lazımı ünvana nüsxələyək.

```
root@siteB:/usr/local/lib/openvpn/plugins # cp
/usr/local/lib/openvpn/plugins/openvpn-plugin-down-root.so
/usr/local/etc/openvpn/
```
2. Server quraşdırma faylını yaradaq:

```
proto udp
port 1194
dev tun

server 192.168.200.0 255.255.255.0

ca /usr/local/etc/openvpn/ca.crt
```

```
cert /usr/local/etc/openvpn/openvpnserver.crt
key /usr/local/etc/openvpn/openvpnserver.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0

persist-key
persist-tun
keepalive 10 60

topology subnet

user nobody
group nobody # nogroup on some distros

daemon
log-append /var/log/openvpn.log

script-security 2
cd /usr/local/etc/openvpn
up "example6-9.sh"
plugin ./openvpn-plugin-down-root.so "/usr/local/etc/openvpn/example6-9.sh --down"

suppress-timestamps
verb 5
```

Yuxarıdakı sətirləri `/usr/local/etc/openvpn/example6-9-server.conf` faylında yadda saxlayın.

3. Sonra up scripti yaradın hansı ki, biz həmçinin down-root plugini üçün istifadə edəcəyik:

```
#!/bin/sh
if [ "$script_type" = "up" ]
then
    touch /tmp/example6-9.tempfile
fi
if [ "$1" = "--down" ]
then
    rm /tmp/example6-9.tempfile
fi
```

Scripti `/usr/local/etc/openvpn/example6-9.sh` faylında yadda saxlayın və yerinə yetirən edin.

```
root@siteA:/usr/local/etc/openvpn # chmod +x
/usr/local/etc/openvpn/example6-9.sh
```

4. OpenVPN server işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example6-9-server.conf
```

Server jurnal faylı indi göstərəcək:

```
PLUGIN_CALL: POST ./openvpn-plugin-down-root.so/PLUGIN_UP status=0
example6-9.sh tun0 1500 1541 192.168.200.1 255.255.255.0 init
```

Bu onu göstərir ki, plugin işə düşdü. Səhv kodlarda səhvin çap edilməməsi o deməkdir ki, plugin uğurla işə düşdü.

5. Sonra `/tmp/example6-9.tempfile` faylın serverdə uğurla yaradılmasını yoxlayın.
6. Sonra OpenVPN serveri dayandırın və jurnal faylını yenidən yoxlayın:
`PLUGIN_CALL: POST ./openvpn-plugin-down-root.so/PLUGIN_DOWN status=0`
`PLUGIN_CLOSE: ./openvpn-plugin-down-root.so`
7. Əmin olun ki, `/tmp/example6-9.tempfile` faylı serverdən silindi.

Bu necə işləyir...

down-root plugini OpenVPN server prosesi root istifadəçi adından işləyəndə sistem startup-ında qeydiyyatdan keçir. Pluginlər ayrılmış axında işləyirlər yəni ki, OpenVPN server öz prosesini root yetkisindən ayırsa da, pluginlər yenə də root istifadəçi adından işləyirlər. OpenVPN dayandırılanda plugin çağırılır və root istifadəçi adından plugin tərəfindən yaradılmış fayl silinir.

Server jurnal fayllarının maraqlı tərəfi:

```
ifconfig: SIOCIFDESTROY: Operation not permitted
FreeBSD 'destroy tun interface' failed (non-critical): external program
exited with error status: 1
PLUGIN_CALL: POST ./openvpn-plugin-down-root.so/PLUGIN_DOWN status=0
PLUGIN_CLOSE: ./openvpn-plugin-down-root.so
```

Bu o deməkdir ki, OpenVPN prosesi həqiqətən də cli-dan `/sbin/ifconfig tun0 0.0.0.0` əmrini yerinə yetirə bilmədi ona görə ki, onun əlindən yetkilər alınmışdır. Plugin artıq onda root yetkiləri olmadığına çağırıldı və onun `/tmp`-dən root yetkisi olan faylı silmək hüququ yoxdur.

Qeyd edin ki, pluginin özü üçün `./` ilə ünvanı təyin etmək və yerinə yetirilən script üçün tam ünvanı təyin etmək çox önəmlidir. Əgər bu ünvanlar düzgün təyin edilməyə nə plugin və nə də script yerinə yetirilməyəcək. `./` hal-hazırkı ünvan deməkdir və **PATH** mühit dəyişənin hissəsi deyil.

Həmçinin qeyd edin ki, up scripti mühit dəyişəni olan **script_type** təyinatı ilə çağırılır ancaq, bu pluginlər üçün doğru deyil. Bunu aşmaq üçün isə əlavə parametr təyin edildi ki, eyni script həm **down-up** kimi istifadə edilsin.

Daha da ətraflı...

Pluginlər Linux, Net/FreeBSD və Windows-da dəstəklənir. Aşağıdakı script geri çağırışları pluginləri istifadə edərək ələ keçirilə bilər:

- Up
- down
- route-up
- ipchange
- tls-verify
- auth-user-pass-verify

- client-connect
- client-disconnect
- learn-address

Həmçinin baxın

- Növbəti başlıq hansı ki, PAM authentication pluginin istifadəsini açıqlayır. Burda OpenVPN pluginin istifadə edilməsi ilə uzaq clientlərin necə qeydiyyatdan keçilməsi göstərilir.

PAM authentication pluginin istifadə edilməsi

OpenVPN istifadəçi adının düzgünlüyün yoxlanılması üçün çox istifadə edilən plugin var hansı ki, Linux/UNIX PAM authentication sistemini istifadə edir. Pluggable Authentication Modules isə PAM üçün açıqlamadır və istifadəçilərin sistem resurslarından istifadəsini idarə etmək üçün məşhur sistemdir. Bu əksər LINUX/UNIX sistemlərində işləyir. Çox rahat, geniş imkanlıdır və istifadəçilərin autentifikasiyası və autorizasiyası üçün genişləndirilə bilər. Bu misalda biz PAM plugini **auth-user-pass-verify** plugininə əvəz olaraq istifadə edəcəyik ki, uzaq istifadəçi verilənlərini sistem PAM quraşdırmasında yenidən yoxlayaq.

İşə başlayaq

2-ci başlıqda yaratdığımız server və client sertifikatlarını burda da istifadə edəcəyik. Bu misalda server maşını FreeBSD9.2 x64 OpenVPN2.3 və client isə Windows7 x64 OpenVPN2.3-də olacaq.

Bunu necə edək...

1. Öncə Plugini lazımı ünvanı nüsxələyək:

```
root@siteA:/usr/local/etc/openvpn # cp
/usr/local/lib/openvpn/plugins/openvpn-plugin-auth-pam.so
/usr/local/etc/openvpn/
```

Server quraşdırma faylını yaradaq:

```
proto udp
port 1194
dev tun
```

```
server 192.168.200.0 255.255.255.0
```

```
ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnsrvr.crt
key /usr/local/etc/openvpn/openvpnsrvr.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0
```

```
persist-key
persist-tun
keepalive 10 60
```

```
topology subnet
```

```
user nobody
group nobody      # Bəzi distributivlərdə nogroup olur
```

```
daemon
log-append /var/log/openvpn.log
```

```
verb 5
suppress-timestamps
```

```
plugin /usr/local/lib/openvpn/plugins/openvpn-plugin-auth-pam.so "login
login USERNAME password PASSWORD"
```

Öncəki sətirləri `/usr/local/etc/openvpn/example6-10-server.conf` adında yadda saxlayaq.

2. OpenVPN serveri işə salaq:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example6-10-
server.conf
```

Server jurnal faylı indi göstərəcək:

```
AUTH-PAM: BACKGROUND: INIT service='login'
PLUGIN_INIT: POST /usr/local/lib/openvpn/plugins/openvpn-plugin-auth-
pam.so '['/usr/local/lib/openvpn/plugins/openvpn-plugin-auth-pam.so]
[login] [login] [USERNAME] [password] [PASSWORD]'
intercepted=PLUGIN_AUTH_USER_PASS_VERIFY
```

Bu o deməkdir ki, PAM pluginin arxa fona ötürülərək uğurla inisializasiya edilmişdir.

3. İndi işə client quraşdırma faylını yaradaq:

```
client
proto udp
remote openvpnserver.example.com
port 1194
```

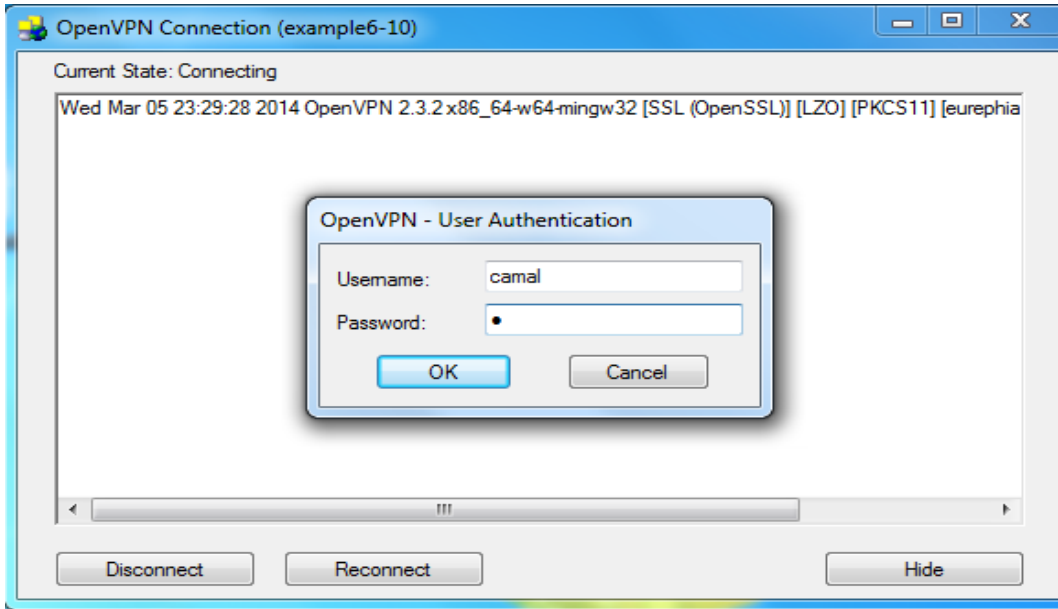
```
dev tun
nobind
```

```
ca "c:/program files/openvpn/config/ca.crt"
cert "c:/program files/openvpn/config/openvpnclient1.crt"
key "c:/program files/openvpn/config/openvpnclient1.key"
tls-auth "c:/program files/openvpn/config/ta.key" 1
```

```
auth-user-pass
```

Yuxarıdakı sətirləri `example6-10.ovpn` adında yadda saxlayın.

4. OpenVPN client işə salın. Windows-un OpenVPN GUI-si əlavə pəncərədə istifadəçi adı və şifrəni tələb edəcək:



VPN serverimizdə bu yoxlanış üçün **camal** adlı istifadəçi **1** şifrəsi ilə öncədən yaradılmışdır. İstifadəçi adı və şifrə daxil edildikdən sonra qoşulma uğurla sona çatır. OpenVPN serverin jurnalları aşağıdakı sətirləri göstərəcək:

```

AUTH-PAM: BACKGROUND: USER: camal
AUTH-PAM: BACKGROUND: my_conv[0] query='Login:' style=2
AUTH-PAM: BACKGROUND: name match found, query/match-string ['Login:',
'login'] = 'USERNAME'
AUTH-PAM: BACKGROUND: my_conv[0] query='Password:' style=1
AUTH-PAM: BACKGROUND: name match found, query/match-string
['Password:', 'password'] = 'PASSWORD'
2.2.2.10:49947 PLUGIN_CALL: POST
/usr/local/lib/openvpn/plugins/openvpn-plugin-auth-
pam.so/PLUGIN_AUTH_USER_PASS_VERIFY status=0
2.2.2.10:49947 TLS: Username/Password authentication succeeded for
username 'camal'

```

Bu onu göstərir ki, istifadəçi PAM istifadə edərək, uğurla qeydiyyatdan keçdi. Nəzərə alın ki, sistemdə root id 0 (sıfır) olanlardan başqa bütün login olma haqlı və shell mühiti olanların giriş hüququ olur.

Bu necə işləyir...

PAM authentication plugini auth-user-pass-verify qayıdışını tutur. OpenVPN client qoşulanda və istifadəçi adı ilə şifrəni ötürəndə plugin işə düşür. O PAM subsystem-ə müraciət yollayır və **“login”** modula baxır (Bu **openvpn-plugin-auth-pam.so** faylı üçün ilk parametrdir). Auth-pam plugin tərəfindən digər parametrlər ona görə istifadə edilir ki, bilinsin hansı girişi PAM subsistemindən gözləmək lazımdır.

login USERNAME password PASSWORD

PAM **“login”** altsistemi istifadəçi adı üçün köməkçi səhifədə **“login”** sözü və şifrə üçün köməkçi səhifədə **“password”** sözünü çap edəcək. **Auth-pam** plugini bu informasiyanı istifadə edir ki, bilə harda istifadəçi adı (**USERNAME**) və şifrə (**PASSWORD**)-ü təyin etməlidir.

Istifadəçi PAM subsystemlə uğurla qeydiyyatdan keçdikdən sonra qoşulma birləşir.

Daha da ətraflı...

Həmçinin imkan var ki, istifadəçiləri **'auth-user-pass-verify'** scripti sayəsile qeydiyyatdan keçirək hansı ki, PAM subsystem-e müraciət yollayır. Orda bunun üçün PAM pluginin istifadəsinə iki əsas üstünlük var:

- O istənilən halda **'script-security'** direktivinin istifadəsinə tələb etmir
- Plugin metodu çox sürətli və daha da çox əhatəlidir. Çoxlu istifadəçi OpenVPN serverə eyni anda qoşulmağa çalışdıqda, VPN serverin dayanıqlığı **auth-user-pass-verify** scriptin istifadəsindən asılı olacaq onda görə ki, hər istifadəçinin qoşulmasında uyğun prosesin startına ehtiyac var və bu qoşulmada OpenVPN-in əsas axınına təyin edilir.

Həmçinin baxın

- Öncəki misalda **'down-root'** pluginin istifadəsinə hansı ki, OpenVPN pluginlərin əsaslarını açıqlayır.

BÖLÜM 7

OpenVPN quraşdırmalarının problemlərinin araşdırılması

Bu başlıqda biz aşağıdakıları açıqlayacağıq:

- Chipher uyğunsuzluğu
- TUN-un TAP-a qarşı uyğunsuzluğu
- Kompresiya uyğunsuzluğu
- Açar uyğunsuzluğu
- MTU ve tun-mtu problemlərinin araşdırılma qaydaları
- Şəbəkə qoşulmasının problemlərinin araşdırılması
- Client-config-dir problemlərinin araşdırılması
- OpenVPN jurnal fayllarının oxunulması qaydaları

Giriş

Bu başlığın əsas məqsədi bütün OpenVPN problemlərinin araşdırılması qaydalarını öyrətməkdir. Bu başlıq OpenVPN-in səhv quraşdırılmasının araşdırılmasına ayrılmışdır hansı ki, növbəti başlıq tamamilə OpenVPN-in routinglə bağlı problemlərinin araşdırılmasına ayrılmışdır.

Ona görə də bu başlıqda istifadə edilən reseptlər ilk olaraq səhv olan işlərə diqqət yetirəcək. Sonra biz alətlər təqdim edəcəyik hansı ki, problemlərin necə tapılması və quraşdırma səhvlərinin düzəldilməsinə kömək edəcək. Bu başlıqda olan quraşdırma direktivlərinin bəziləri öncə heç istifadə edilməmişdir.

Cipher uyğunsuzluğu

Bu başlıqda biz OpenVPN istifadə edən cryptographic cipher-i dəyişdirəcəyik. İlk olaraq biz cipher-i client tərəfdə dəyişəcəyik hansı ki, VPN qoşulması inisializasiyasında səhv çıxaracaq. Bu misalın əsas məqsədi OpenVPN dəstəkləyən cipher metodlarını göstərmək yox, çıxan səhv mesajlarını görməkdir.

İşə hazırlaşaq

OpenVPN2.3-ü iki maşında yükləyin. Əmin olun ki, maşınlar şəbəkə ilə bir-birlərini görürlər. 2-ci başlıqda yaratdığınız client və server sertifikatlarını yaradın. Bu misalda client və server maşınları FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Server üçün 2-ci başlıqda istifadə etdiyimiz Server-side routing quraşdırma faylını **basic-udp-server.conf**-u istifadə edin. Client quraşdırma isə **basic-udp-client.conf** olacaq.

Necə edək...

1. **basic-udp-server.conf** faylını istifadə edərək server işə salın:
root@siteA:/usr/local/etc/openvpn # **openvpn --config basic-udp-server.conf**
2. Clientdə üçün **basic-udp-client.conf** faylını **example7-1-client.conf** faylına nüsxələyin və aşağıdakı sətiri **example7-1-client.conf** faylının sonuna əlavə edirik:
cipher CAST5-CBC
3. Client işə salın və sonra jurnallara baxın aşağıdakı sətir client jurnallarında göstəriləcək:
root@siteB:/usr/local/etc/openvpn # **openvpn --config example7-1-client.conf**
Thu Mar 6 13:37:31 2014 OpenVPN 2.3.2 amd64-portbld-freebsd9.2 [SSL (OpenSSL)] [LZO] [eurephia] [MH] [IPv6] built on Jan 9 2014
Thu Mar 6 13:37:32 2014 Control Channel Authentication: using '/usr/local/etc/openvpn/ta.key' as a OpenVPN static key file
Thu Mar 6 13:37:32 2014 UDPv4 link local: [undef]
Thu Mar 6 13:37:32 2014 UDPv4 link remote: [AF_INET]1.1.1.10:1194
Thu Mar 6 13:37:32 2014 WARNING: 'cipher' is used inconsistently, local='cipher CAST5-CBC', remote='cipher BF-CBC'
Thu Mar 6 13:37:32 2014 [openvpnserv] Peer Connection Initiated with [AF_INET]1.1.1.10:1194
Thu Mar 6 13:37:34 2014 TUN/TAP device /dev/tun0 opened
Thu Mar 6 13:37:34 2014 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0
Thu Mar 6 13:37:34 2014 /sbin/ifconfig tun0 192.168.200.2 192.168.200.2 mtu 1500 netmask 255.255.255.0 up
add net 192.168.200.0: gateway 192.168.200.2
add net 10.198.0.0: gateway 192.168.200.1
Thu Mar 6 13:37:34 2014 Initialization Sequence Completed
Thu Mar 6 13:37:44 2014 Authenticate/Decrypt packet error: cipher final failed

Və uyğun olaraq server tərəfdə:

```
root@siteA:/usr/local/etc/openvpn # tail -f /var/log/openvpn.log
```

```
Thu Mar 6 13:40:34 2014 openvpnclient1/2.2.2.10:11653
Authenticate/Decrypt packet error: cipher final failed
Thu Mar 6 13:40:46 2014 2.2.2.10:21671 WARNING: 'cipher' is used
inconsistently, local='cipher BF-CBC', remote='cipher CAST5-CBC'
Thu Mar 6 13:40:46 2014 2.2.2.10:21671 [openvpnclient1] Peer
Connection Initiated with [AF_INET]2.2.2.10:21671
Thu Mar 6 13:40:46 2014 MULTI_sva: pool returned IPv4=192.168.200.2,
IPv6=(Not enabled)
Thu Mar 6 13:40:49 2014 openvpnclient1/2.2.2.10:21671
send_push_reply(): safe_cap=940
Thu Mar 6 13:40:59 2014 openvpnclient1/2.2.2.10:21671
Authenticate/Decrypt packet error: cipher final failed
```

Qoşulma uğurlu olmayacaq və həmçinin cəld olaraq qoşulmadan kəsilməyəcək.

Bu necə işləyir...

Server və client qoşulması müddətində təhlükəsiz qoşulma üçün müəyyən parametrlər öz aralarında razılaşırlar. Bu fazanın əsas parametrlərindən biri şifrələnmə cipheridir hansı ki, bütün mesajların şifrələnməsi və deşifrələnməsi üçün istifadə edilir. Əgər client və server müxtəlif cipherlər istifadə edirsə, onda onlar bir-biri arasında danışma razılığına gəlməyəcəklər.

Aşağıdakı direktivi server quraşdırmasına əlavə etməklə server və client yenidən əlaqə qura biləcəklər:

```
cipher CAST5-CBC
```

Daha da ətraflı...

OpenVPN çox az ciphers dəstəkləyir, həmçinin dəstəklənən bezi cipherlərdə hələ sınaq müddətindədir. Dəstəklənən cipher-lərin siyahısına baxmaq üçün aşağıdakı əmrəndən istifadə edə bilərsiniz:

```
root@siteA:/usr/local/etc/openvpn # openvpn --show-ciphers
DES-CBC 64 bit default key (fixed)
RC2-CBC 128 bit default key (variable)
DES-EDE-CBC 128 bit default key (fixed)
DES-EDE3-CBC 192 bit default key (fixed)
DESX-CBC 192 bit default key (fixed)
BF-CBC 128 bit default key (variable)
RC2-40-CBC 40 bit default key (variable)
CAST5-CBC 128 bit default key (variable)
RC5-CBC 128 bit default key (variable)
RC2-64-CBC 64 bit default key (variable)
AES-128-CBC 128 bit default key (fixed)
AES-192-CBC 192 bit default key (fixed)
AES-256-CBC 256 bit default key (fixed)
CAMELLIA-128-CBC 128 bit default key (fixed)
CAMELLIA-192-CBC 192 bit default key (fixed)
CAMELLIA-256-CBC 256 bit default key (fixed)
```

Bu bütün cipherləri göstərir həm variable-larda və həm də fixed cipher uzunluğunda. **Cipher length** dəyişəni ilə olan cipherlər OpenVPN tərəfindən çox

yaxşı dəstəklənir, qalanlarıda işləyəcək ancaq, müəyyən anlar gözlənilməyən nəticələr ola bilər.

TUN və TAP alətlərinin uyğunsuzluğu

OpenVPN bazalı VPN istifadə edilməsində ən çox olan səhvlərdən biri seçilən adapterin tipinin təyin edilməsində olur. Yəni ki, server TUN tipli adapter üçün quraşdırılmış və client isə əksinə TAP tipli alətlə quraşdırılmışdır. Başlıqda əsas quraşdırma səhvi olanda baş verən yalnışlıqları görəcəyik.

İşə hazırlaşaq

OpenVPN2.3 ya da daha yuxarı versiyanı 2 məşində yükləyin. Əmin olun ki, maşınlar şəbəkə ilə bir-birlərini görürlər. 2-ci başlıqda istifadə edilən client və server sertifikatlarını burdada istifadə edəcəyik. Bu başlıqda server və client maşını üçün FreeBSD9.2 x64 və OpenVPN2.3 istifadə edəcəyik. Server quraşdırması üçün 2-ci başlıqda Server-side routing misalında istifadə elədiyimiz **basic-udp-server.conf** faylından istifadə edəcəyik. Client quraşdırma faylı isə **basic-udp-client.conf** olacaq.

Bu necə işləyir...

1. **basic-udp-server.conf** quraşdırma faylını istifadə edərək server işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-server.conf
```

2. Sonra client quraşdırma faylını yaradaq:

```
client
proto udp
remote openvpnsrver.example.com
port 1194

dev tap
nobind

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnclient1.crt
key /usr/local/etc/openvpn/openvpnclient1.key
tls-auth /usr/local/etc/openvpn/ta.key 1
```

ns-cert-type server

Yuxarıdakı sətirləri **example7-2-client.conf** adında client maşında yadda saxlayın.

3. Clienti işə salın:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config example7-2-client.conf
```

Client jurnalı aşağıdakıları göstərəcək:

```
Thu Mar 6 23:13:37 2014 WARNING: 'dev-type' is used inconsistently,
local='dev-type tap', remote='dev-type tun'
Thu Mar 6 23:13:37 2014 WARNING: 'link-mtu' is used inconsistently,
local='link-mtu 1573', remote='link-mtu 1541'
Thu Mar 6 23:13:37 2014 WARNING: 'tun-mtu' is used inconsistently,
local='tun-mtu 1532', remote='tun-mtu 1500'
Thu Mar 6 23:13:37 2014 WARNING: 'cipher' is used inconsistently,
local='cipher BF-CBC', remote='cipher CAST5-CBC'
Thu Mar 6 23:13:37 2014 [openvpnserver] Peer Connection Initiated with
[AF_INET]1.1.1.10:1194
Thu Mar 6 23:13:39 2014 TUN/TAP device /dev/tap0 opened
Thu Mar 6 23:16:44 2014 do_ifconfig, tt->ipv6=0, tt-
>did_ifconfig_ipv6_setup=0
Thu Mar 6 23:16:44 2014 /sbin/ifconfig tap0 192.168.200.2
192.168.200.2 mtu 1500 netmask 255.255.255.0 up
route: writing to routing socket: File exists
add net 192.168.200.0: gateway 192.168.200.2 fib 0: route already in
table
Thu Mar 6 23:16:44 2014 ERROR: FreeBSD route add command failed:
external program exited with error status: 1
add net 10.198.0.0: gateway 192.168.200.1
Thu Mar 6 23:16:44 2014 Initialization Sequence Completed
```

Bu hissədə siz serverə ping etməyə çalışa bilərsiniz ancaq, cavab error ilə olacaq:

```
root@siteB:~ # ping 192.168.200.1
ping: sendto: Host is down
ping: sendto: Host is down
ping: sendto: Host is down
ping: sendto: Host is down
```

Bu necə işləyir...

TUN stilli interfeys point-to-point qoşulmasına əsaslanır hansı ki, yalnız TCP/IP trafiki tunel edilə bilər. TAP stilli interfeys isə Ethernet interfeysin ekvivalentidir hansı ki, özünə əlavə başlıqları artırır. Bu istifadəçiyə imkan yaradır ki, digər tip trafiki interfeys üzərindən ötürə bilsin. Client və server səhv quraşdırılarda təyin edilən paket həcmi müxtəlif olacaq:

```
... WARNING: 'tun-mtu' is used inconsistently, local='tun-mtu 1532',
remote='tun-mtu 1500'
```

Bu onu göstərir ki, TAP stilli interfeys üzərindən keçən hər paket TUN stilli interfeysə baxanda 32 bayt daha genişdir. Client quraşdırılmasında **dev tap** əvəzinə **dev tun** yazsanız problem həll ediləcək.

Kompresiya uyğunsuzluğu

VPN tunel üzərindən gedən trafikın sıxılması üçün OpenVPN **on-the-fly** sıxılma alqoritmini dəstəkləyir. Zəif şəbəkə xətti olan yerlərdə davamlılığını artırır

bilər ancaq, bu əlavə başlıq artırır. Sıxılmamış datanın ötürülməsində (Misal üçün ZIP faylları) davamiyyət həqiqətən də kiçilir.

Əgər sıxılma serverdə aktivləşmiş və clientdə deyilsə, onda qoşulma qırılacaq.

İşə hazırlaşaq

OpenVPN2.3 ya da daha yuxarı versiyanı 2 məşində yükləyin. Əmin olun ki, maşınlar şəbəkə ilə bir-birlərini görürlər. 2-ci başlıqda istifadə edilən client və server sertifikatlarını burda da istifadə edəcəyik. Bu başlıqda server və client maşını üçün FreeBSD9.2 x64 və OpenVPN2.3 istifadə edəcəyik. Server quraşdırması üçün 2-ci başlıqda Server-side routing misalında istifadə elədiyimiz **basic-udp-server.conf** faylından istifadə edəcəyik. Client quraşdırma faylı isə **basic-udp-client.conf** olacaq.

Necə edək...

1. Server faylı **basic-udp-server.conf**-u **example7-3-server.conf** faylına nüsxələyin və **example7-3-server.conf** faylının sonuna aşağıdakı sətiri əlavə edin:

```
comp-lzo
```

2. Serveri işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example7-3-server.conf
```

3. Sonra isə clienti işə salın:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config basic-udp-client.conf
```

Qoşulma inisializasiya olacaq ancaq, VPN qoşulma üzərindən data gətdikdə aşağıdakı mesaj çap ediləcək:

```
Thu Mar 6 23:44:28 2014 WARNING: 'link-mtu' is used inconsistently, local='link-mtu 1541', remote='link-mtu 1542'
```

```
Thu Mar 6 23:44:28 2014 WARNING: 'comp-lzo' is present in remote config but missing in local config, remote='comp-lzo'
```

Bu necə işləyir...

Qoşulma fazası müddətində client və server arasında transfer informasiyasında sıxılma olmur. Parametrlərdən biri razılaşılır ki, compressiyanı VPN-in faktiki xeyirli yüklənməsi üçün istifadə edilsin.

Əgər client və server arasında olan quraşdırmada səhv olarsa, hər iki tərəf ötürülən trafiki qarışdıracaq.

Əgər şəbəkədə olan client və serverlərin hamısında OpenVPN2.3 istifadə edilirsə, aşağıdakı sətiri əlavə etməklə bütün problemlərinizi həll etmiş olacaqsınız:

```
push "comp-lzo"
```

Daha da ətraflı...

OpenVPN2.0-da imkan yoxdur ki, compressiya direktivini clientlərə təyin edə bilsin. Bu o deməkdir ki, OpenVPN2.0 server bu direktivi anlamır nəinki

client. Yəni ki, siz ən azı OpenVPN2.1 server və OpenVPN2.0 client istifadə etsəniz qoşulma olmayacaq.

Açar uyğunsuzluğu

OpenVPN özünün TLS kanalının idarə edilməsi üçün əlavə HMAC açarları təqdim edir. Bu açarlar 1-ci başlıqda Point-to-Point şəbəkələrdə olduğu kimi eyni olaraq, static "secret" açarlar istifadə edir. multi-client stilli şəbəkələr üçün bu əlavə qorunma **tls-auth** direktivinin istifadə edilməsi ilə aktivləşdirilə bilər. Əgər bu **tls-auth key** server və client-lə əlaqədə uyğunsuz olarsa, onda VPN qoşulma uğursuz olacaq və inisializasiya olmayacaq.

İşə hazırlaşaq

OpenVPN2.3 yada daha yuxarı versiyanı 2 məşində yükləyin. Əmin olun ki, maşınlar şəbəkə ilə bir-birlərini görürlər. 2-ci başlıqda istifadə edilən client və server sertifikatlarını burda da istifadə edəcəyik. Bu başlıqda server və client maşını üçün FreeBSD9.2 x64 və OpenVPN2.3 istifadə edəcəyik. Server quraşdırması üçün 2-ci başlıqda Server-side routing misalında istifadə elədiyimiz **basic-udp-server.conf** faylından istifadə edəcəyik. Client quraşdırma faylı isə **basic-udp-client.conf** olacaq.

Bunu necə edək...

1. **basic-udp-server.conf** faylını istifadə edərək serveri işə salın:
root@siteA:/usr/local/etc/openvpn # **openvpn --config basic-udp-server.conf**

2. Sonra client quraşdırmasını yaradın:

```
client
proto udp
remote openvpnsrver.example.com
port 1194

dev tun
nobind

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnclient1.crt
key /usr/local/etc/openvpn/openvpnclient1.key
tls-auth /usr/local/etc/openvpn/ta.key

ns-cert-type server
```

Qeyd edin ki, **tls-auth** direktivinin ikinci parametri təyin edilməyib. Yuxarıdakı sətirləri **example7-4-client.conf** adında **/usr/local/etc/openvpn** ünvanında yadda saxlayın.

3. Clienti işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example7-4-client.conf
```

Client jurnalında səhv görünməyəcək ancaq, qoşulma uğurlu olmayacaq.
Server jurnalda isə aşağıdakılar görünəcək:

```
Fri Mar 7 00:26:05 2014 Authenticate/Decrypt packet error: packet HMAC authentication failed
```

```
Fri Mar 7 00:26:05 2014 TLS Error: incoming packet authentication failed from [AF_INET]2.2.2.10:16987
```

Bu onu göstərir ki, `openvpnclient1` clienti `tls-auth` direktivində səhv parametr istifadə edir və qoşulma kəsilir.

Bu necə işləyir...

Qoşulma inisializasiyası olan anda ilk fazada client və server hər biri digər tərəfin HMAC açarlarını yoxlayır. Əgər HMAC açar düzgün quraşdırılmayıbsa, inisializasiya kəsilir və qoşulma dayandırılır. OpenVPN server müraciət edən şəxsin həqiqi client ya da pis niyyətli birisinin serveri yükləmək istədiyini təyin edə bilmədiyinə görə qoşulma sadəcə düşür.

Bu başlıqda çatışmayan səhv quraşdırma ondan ibarətdir ki, aşağıdakı sətirin sonunda `1` parametri yoxdur.

```
tls-auth /etc/openvpn/itvpn/ta.key
```

`tls-auth` direktivinin ikinci parametri açar üçün istiqamətdir. Normalda aşağıdakı razılaşma istifadə edilir:

```
0: serverden cliente  
1: clientden serverə
```

Bu parametr OpenVPN-i çağırır ki, `ta.key` faylının digər hissəsindən öz HMAC açarları çağırınsın. Əgər client və server qarşı tərəflərdən gələn HMAC açar hissələrinə razı deyilsə onda qoşulma kəsiləcək. Eyni olaraq deyə bilərik ki, client və server HMAC açarları fərqli `ta.key` faylından istifadə edirlərsə, qoşulma yenə də olmayacaq.

Həmçinin baxın

1-ci başlığın misalında, **Multiple secret keys** hansı ki, OpenVPN secret keylərin formatı və istifadəsini detallı açıqlayır.

MTU və tun-mtu problemlərinin araşdırılması

OpenVPN-nin əsas böyük üstünlüklərindən biri də şəbəkə parametrlərini həm TUN və həm də TAP adapterləri və şifrələnmiş linkləri öz istəklərimizə görə dəyişə bilməsi olmasıdır. Bu həmişə çıxan səhvdir ki, kiçik davamiyyətə ya da ümumiyyətlə VPN tunel üzərindən heç bir datanı ötürmək olmur. Bu misal client və server arasında istifadə edilən MTU-nun fərqli olması nəticəsində ortaya çıxan səhvləri və bu uyğunsuzluğun necə bəzi hallarda VPN tunelin düşməsinə gətirib çıxmasını açıqlayacağıq.

İşə hazırlaşaq

OpenVPN2.3 ya da daha yuxarı versiyanı 2 məşində yükləyin. Əmin olun ki, məşinlər şəbəkə ilə bir-birlərini görürlər. 2-ci başlıqda istifadə edilən client və server sertifikatlarını burda da istifadə edəcəyik. Bu başlıqda

server və client maşını üçün FreeBSD9.2 x64 və OpenVPN2.3 istifadə edəcəyik. Server quraşdırması üçün 2-ci başlıqda Server-side routing misalında istifadə elədiyimiz **basic-udp-server.conf** faylından istifadə edəcəyik. Client quraşdırma faylı isə **basic-udp-client.conf** olacaq.

Bunu necə edək...

1. Quraşdırma faylı **basic-udp-server.conf** istifadə edərək server işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-server.conf
```

2. Sonra client üçün **basic-udp-client.conf** faylını **example7-5-client.conf** faylına nüsxələyin və **example7-5-client.conf** faylının içinə aşağıdakı sətiri əlavə edin:

```
tun-mtu 1400
```

3. Clienti işə salın və sonra client jurnalına baxın:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config example7-5-client.conf
```

```
Sat Mar 8 13:12:23 2014 WARNING: normally if you use --mssfix and/or -  
-fragment, you should also set --tun-mtu 1500 (currently it is 1400)  
Sat Mar 8 13:12:23 2014 UDPv4 link local: [undef]  
Sat Mar 8 13:12:23 2014 UDPv4 link remote: [AF_INET]1.1.1.10:1194  
Sat Mar 8 13:12:23 2014 WARNING: 'link-mtu' is used inconsistently,  
local='link-mtu 1441', remote='link-mtu 1541'  
Sat Mar 8 13:12:23 2014 WARNING: 'tun-mtu' is used inconsistently,  
local='tun-mtu 1400', remote='tun-mtu 1500'  
Sat Mar 8 13:12:23 2014 [openvpnsrver] Peer Connection Initiated with  
[AF_INET]1.1.1.10:1194  
Sat Mar 8 13:12:26 2014 TUN/TAP device /dev/tun0 opened  
Sat Mar 8 13:12:26 2014 do_ifconfig, tt->ipv6=0, tt-  
>did_ifconfig_ipv6_setup=0  
Sat Mar 8 13:12:26 2014 /sbin/ifconfig tun0 192.168.200.2  
192.168.200.2 mtu 1400 netmask 255.255.255.0 up  
add net 192.168.200.0: gateway 192.168.200.2  
add net 10.198.0.0: gateway 192.168.200.1  
Sat Mar 8 13:12:26 2014 Initialization Sequence Completed
```

Gördüyünüz kimi tunel qalxdığı anda çoxlu warning-lər çıxır amma tunel yenə də qalxdı.

4. Həmçinin mümkündür ki, axını şəbəkə ilə ötürə biləsiniz, ping əmri ilə yoxlaya bilərsiniz:

```
root@siteB:~ # ping -c 2 192.168.200.1  
PING 192.168.200.1 (192.168.200.1): 56 data bytes  
64 bytes from 192.168.200.1: icmp_seq=0 ttl=64 time=1.312 ms  
64 bytes from 192.168.200.1: icmp_seq=1 ttl=64 time=1.321 ms
```

5. Ancaq böyük paketlər ötürmək istəyəndə, misal üçün aşağıdakı kimi:

```
root@siteB:~ # ping -s 1450 192.168.200.1
```

Aşağıdakı mesajlar client log faylında göstəriləcək:

```
Sat Mar 8 13:16:26 2014 Authenticate/Decrypt packet error: packet HMAC authentication failed
Sat Mar 8 13:16:27 2014 Authenticate/Decrypt packet error: packet HMAC authentication failed
```

Əgər client böyük həcmli faylı endirmək istəsə eyni nəticə olacaq.

Bu necə işləyir...

MTU yada Maximum Transfer Unit təyin edir ki, VPN tunelin üzərindən keçən bir paket heç bir bölgü olmadan (yeni fraqmentlərə bölünmədən) maksimal hansı həcmdə ola bilər. Əgər server və client bu MTU həcmi birgə qəbul etmirlərsə və server clientə böyük həcmli paket yollayarsa onda, HMAC səhvi çıxacaq (əgər bu misalda olduğu kimi, tls-auth istifadə edilirsə) ya da paketin böyük olan qalan hissəsi kənara atılacaq.

Daha da ətraflı...

Windows platformasında TAP-Win32 adapterlər üçün MTU həcmi dəyişmək çox çətinidir. Tun-mtu direktivini təyin edə bilərsiniz ancaq, Windows üzərində işləyən OpenVPN uyğun olan MTU həcmi dəyişə bilməyəcək (Çünki Windows Vista-yədək heç bir Windows bunu dəstəkləmirdi). Həmçinin Windows7-də də bu imkan yoxdur.

Həmçinin baxın

- 9-cu başlıqda Performance Tuning başlığında sizə göstərəcək ki, tun-mtu direktivi ilə necə optimizasiya işləri görə bilərsiniz.

Şəbəkə qoşulmasının problemlərinin araşdırılması

Bu misalda biz müəyyən jurnal tiplərini göstərəcəyik hansı ki, quraşdırma düz olduğu halda belə şəbəkədə problemlər olur. Əksər hallarda bu tip problem çıxanda, client və ya server tərəfdə firewall block edir.

Bu misalda biz özümüz server qulaq asdığı portu bağlayacağıq və sonra qoşulmağa çalışacağıq.

İşə hazırlaşaq

OpenVPN2.3 ya da daha yuxarı versiyanı 2 məşində yükləyin. Əmin olun ki, maşınlar şəbəkə ilə bir-birlərini görürlər. 2-ci başlıqda istifadə edilən client və server sertifikatlarını burdada istifadə edəcəyik. Bu başlıqda server və client maşını üçün FreeBSD9.2 x64 və OpenVPN2.3 istifadə edəcəyik. Server quraşdırması üçün 2-ci başlıqda Server-side routing misalında istifadə elədiyimiz **basic-udp-server.conf** faylından istifadə edəcəyik. Client quraşdırma faylı isə **basic-udp-client.conf** olacaq.

Necə edək...

1. **basic-udp-server.conf** faylını işə salaraq serveri işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-server.conf
```

2. Server tərəfdə IPFW, PF yada Linux-IpTables istifadə edərək OpenVPN portunu bağlayın:

FreeBSD üçün:

```
root@siteA:/usr/local/etc/openvpn # ipfw add 5000 deny udp from any to any dst-port 1194
```

Linux üçün:

```
root@siteA:/usr/local/etc/openvpn # iptables -I INPUT -p udp --dport 1194 -j DROP
```

3. Sonra isə **basic-udp-client.conf** faylını istifadə edərək client quraşdırma faylını işə salın:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config basic-udp-client.conf
```

Client çalışacaq ki, serverə **UDP** protocol ilə qoşula bilsin. Müəyyən vaxtdan sonra **timeout** olacaq ona görə ki, clientdən heç bir trafik getmir və client vpn-i restart edəcək:

```
Sat Mar 8 13:46:33 2014 TLS Error: TLS key negotiation failed to occur within 60 seconds (check your network connectivity)
```

```
Sat Mar 8 13:46:33 2014 TLS Error: TLS handshake failed
```

```
Sat Mar 8 13:46:33 2014 SIGUSR1[soft,tls-error] received, process restarting
```

Client-in qoşulmasını kəsin və serveri dayandırın.

Bu necə işləyir...

Əgər OpenVPN susmaya görə olan UDP protocol-un istifadə etməsi üçün quraşdırılıbsa, client serverin cavabı üçün **60** saniyə gözləyəcək. Əgər cavab yoxdursa, qoşulma restart edilir. Biz birdəfəlik UDP porta gələn trafiki bağladığımızı görə qoşulma vaxtı yaranır və client serverə heç bir zaman qoşula bilmir.

Client-in qoşulma üçün gözlədiyi vaxtı təyin etmək üçün aşağıdakı direktivi istifadə edə bilərsiniz:

```
hand-window N
```

Burda **N** qoşulma üçün gözlənilən vaxtı saniyələrlə təyin edir. Susmaya görə olan mənası **60** saniyədir.

Sözsüz ki, FireWall qaydasını silsəniz qoşulma işləyəcək.

Daha da ətraflı...

UDP protocol və TCP protocol istifadə edilməsinin əsas fərqi qoşulmanın uğurlu olmasındadır. Hər bir TCP qoşulması həm client və həm də server tərəfdə TCP əl sıxışmasını istifadə edərək işə düşür. Əgər razılaşma uğursuz olursa, onda qoşulma olmur. Orda heç bir tələb yoxdur ki, trafik serverdən qayıtmasını gözləyək ona görə ki, o **drop**(kəsilmə) edilir:

```
Sat Mar  8 14:03:11 2014 Attempting to establish TCP connection with
[AF_INET]1.1.1.10:1194 [nonblock]
Sat Mar  8 14:03:21 2014 TCP: connect to [AF_INET]1.1.1.10:1194 failed, will
try again in 5 seconds: Operation timed out
```

'client-config-dir' problemlerinin araştırılması

Bu misalda biz '**client-config-dir**' direktivi ilə bağlı çıxan problemlərin araştırılması qaydalarını öyrənəcəyik. Bu direktiv CCD faylların ünvanını göstərmək üçün istifadə edilir. CCD faylında xüsusi direktiv istifadə edilə bilər ki, clientin sertifikatına əsaslanaraq spesifik IP ünvan təyin eləmək imkanı olsun. Təcrübə göstərir ki, burda quraşdırma səhvləri əksər hallarda olur. Bu misalda biz əsas səhvlərdən birini edəcəyik və sonra necə bu problemin həll edilməsini sizə göstərəcəyik.

İşə hazırlaşaq

OpenVPN2.3 ya da daha yuxarı versiyayı 2 məşində yükləyin. Əmin olun ki, maşınlar şəbəkə ilə bir-birlərini görürlər. 2-ci başlıqda istifadə edilən client və server sertifikatlarını burda da istifadə edəcəyik. Bu başlıqda server və client maşını üçün FreeBSD9.2 x64 və OpenVPN2.3 istifadə edəcəyik. Server quraşdırması üçün 2-ci başlıqda Server-side routing misalında istifadə elədiyimiz **basic-udp-server.conf** faylından istifadə edəcəyik. Client quraşdırma faylı isə **basic-udp-client.conf** olacaq.

Necə edək...

1. **basic-udp-server.conf** faylını **example7-7-server.conf** faylına nüsxələyin və **example7-7-server.conf** faylının sonuna aşağıdakı sətiri əlavə edin:
client-config-dir /usr/local/etc/openvpn/clients
ccd-exclusive
2. Əmin olun ki, **/usr/local/etc/openvpn/clients** qovluğuna yalnız root istifadəçisinin yetkisi var:
root@siteA:/usr/local/etc/openvpn # **chown root**
/usr/local/etc/openvpn/clients/
root@siteA:/usr/local/etc/openvpn # **chmod 700**
/usr/local/etc/openvpn/clients/
3. Serveri işə salın:
root@siteA:/usr/local/etc/openvpn # **openvpn --config example7-7-server.conf**
4. **basic-udp-client.conf** quraşdırma faylını istifadə edərək clienti işə salın:
root@siteB:/usr/local/etc/openvpn # **openvpn --config basic-udp-client.conf**

```
Sonra client aşağıdaki səhvi çap edərək qoşula bilməyəcək:  
Sat Mar 8 14:25:46 2014 [openvpnserver] Peer Connection Initiated with  
[AF_INET]1.1.1.10:1194  
Sat Mar 8 14:25:48 2014 AUTH: Received control message: AUTH_FAILED
```

Server jurnal faylı qismən fərqlənir: ilk olaraq o deyir ki, CCD qovluğunda olan **openvpnclient1** quraşdırmasını oxuya bilmir və sonra isə VPN-in qalxdığı haqqında məlumat verir:

```
Sat Mar 8 14:25:45 2014 2.2.2.10:13269 TLS Auth Error: --client-  
config-dir authentication failed for common name 'openvpnclient1'  
file='/usr/local/etc/openvpn/clients/openvpnclient1'  
Sat Mar 8 14:25:45 2014 2.2.2.10:13269 [openvpnclient1] Peer  
Connection Initiated with [AF_INET]2.2.2.10:13269
```

Ancaq yenə də VPN qoşulması lazımı kimi olmadı.

Bu necə işləyir...

Aşağıdakı direktivlər OpenVPN tərəfindən istifadə edilir ki, **/usr/local/etc/openvpn/clients** qovluğuna baxıb client-in sertifikat **CN**-nə aid olan quraşdırma faylı oxuya bilsin:

```
client-config-dir /usr/local/etc/openvpn/clients  
ccd-exclusive
```

İkinci direktivin üstünlüyü ondan ibarətdir ki, **ccd-exclusive** yalnız **CDD** faylı olan müştərilərə girişə izin verəcək. Yeni **CCD** faylı olmayan istifadəçi qoşula bilməyəcək.

Client sertifikatın adı server jurnalında göstərilməmişdir:

```
Sat Mar 8 14:25:45 2014 2.2.2.10:13269 TLS Auth Error: --client-config-dir  
authentication failed for common name 'openvpnclient1'  
file='/usr/local/etc/openvpn/clients/openvpnclient1'
```

Ancaq siz bunu aşağıdakı əmrlərdə əldə edə bilərsiniz:

```
root@siteA:/usr/local/etc/openvpn # openssl x509 -subject -noout -in  
openvpnclient1.crt
```

/CN= ilə başlayan hissəyə baxın və boşluqları altdan xətt ilə convert edin

OpenVPN server nobody istifadəçi adından işə düşür ona görə ki, biz **/usr/local/etc/openvpn/clients** qovluğuna çox az hüquqlar təyin etmişdik və bu istifadəçi CCD qovluğunda heç bir faylı oxumaq yetkisinə malik deyil. Əgər client **openvpnclient1** CN-ilə qoşulsada belə OpenVPN CCD faylı oxuya bilməyəcək(hətta fayl orda olsa da belə). Ona görə ki, **ccd-exclusive** direktivi var və sonra heç bir client qoşula bilməz.

Daha da ətraflı...

Bu başlıqda biz jurnallamanın daha dərinləşməsinə və **client-config-dir** direktivində əksər hallarda edilən səhvləri açıqlayacağıq.

Daha da geniş jurnallama

Jurnalların səviyyəsinin artırılması adətən **client-config-dir** səhvini tapmağa çox kömək edir. **Verb 5** ilə və düzgün yetki hüquqları ilə siz aşağıdakı jurnal sətirlərini OpenVPN jurnal faylında görə bilərsiniz:

```
Sat Mar 8 15:09:46 2014 us=128255 openvpnclient1/2.2.2.10:25319
OPTIONS IMPORT: reading client specific options from:
/usr/local/etc/openvpn/clients/openvpnclient1
```

Əgər bu jurnal sətirini serverin jurnal faylında görə bilmirsinizsə onda sizin OpenVPN server clientə aid olan CCD faylı oxuya bilməmişdir.

Digər əksər olan client-config-dir səhvləri

Aşağıdakı əksər çıxan **client-config-dir** səhvləri açıqlanmışdır:

- **client-config-dir** üçün tam ünvan yazılmamışdır. Məsələn:

client-config-dir clients.

Bəzi hallarda bu işləyə bilər ancaq, unutmayın ki, əgər siz server quraşdırmanızda **-chroot** ya da **-cd** direktivlərini istifadə edirsinizsə, bu işləməyəcək. Ona görə ki, chroot direktivinin istifadə edilməsində bütün tam və yarımqıq direktivlər chroot ünvanına uyğun olaraq təyin edilir.

- CCD faylı dəqiq adlandırılmalıdır və genişlənməsi olmalı deyil. Bu adətən Windows istifadəçilərin səhvi olur. Bu halda server jurnal faylına baxıb görə bilərsiniz ki, server düşünür ki, **/CN=name** client sertifikatının adıdır. Həmçinin diqqətli olun ki, OpenVPN **/CN= name** olan yerlərdə bəzi simvolları boşluq kimi görə bilir. Tam simvolların siyahısı üçün man səhifəsinə baxın və **String Types** və **Remapping** bölümünü diqqətlə oxuyun.
- CCD faylı və onun üçün olan tam ünvan işə salınan OpenVPN prosesinin istifadəçisi tərəfindən oxunulan olmalıdır(adətən **nobody** olur).

Həmçinin baxın

- 2-ci başlıqda olan client-config-dir misalına baxın hansı ki, client config fayllarının əsaslarını öyrədir.

OpenVPN jurnal fayllarının oxunulması qaydaları

OpenVPN səhvinin tapılması adətən uzun müddət və düzgün jurnal fayllarının oxunulmasından sonra əldə edilir. Bu misalda OpenVPN-in heç bir yeni imkanı göstərilməyəcək ancaq jurnal fayllarını uzun və detallı araşdıracağıq. Öncəki misalımız olan Troubleshooting MTU və tun-mtu səhvlərinin araşdırılmasındakı quraşdırmaları istifadə edəcəyik.

İşə başlayaq

Biz Troubleshooting MTU və tun-mtu-da olan eyni quruluşdan istifadə edəcəyik. Bu misalda da server və client maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Server quraşdırma faylı 2-ci başlıqda olan Server-side routing misalında olan **basic-udp-server.conf** faylı olacaq. Client üçün isə Troubleshooting MTU və tun-mtu **example7-5-client.conf** faylından istifadə edəcəyik.

Bunu necə edək...

1. **basic-udp-server.conf** faylını istifadə edərək serveri işə salın:
root@siteA:/usr/local/etc/openvpn # **openvpn --config basic-udp-server.conf**
2. Sonra clienti dərin jurnallanma rejimində işə salın və **timestamp**-i yığışdırmaq şərtilə:
root@siteB:/usr/local/etc/openvpn # **openvpn --config example7-5-client.conf --verb 7 --suppress-timestamps**

Qoşulma uğurlu olacaq ancaq, siz böyük həcmli paketləri yollaya bilməyəcəksiniz:
3. Səhvin yaranması üçün aşağıdakı əmrlərdən istifadə edin:
root@siteB:~ # **ping -c1 192.168.200.1**
root@siteB:~ # **ping -c1 -s1450 192.168.200.1**
4. Clienti kəsin. Jurnal faylı qısa müddət ərzində çox böyüyəcək.
5. Jurnal faylı hansısa mətn redaktoru ilə açın və baxın. Jurnal faylının ümumi strukturunu növbəti başlıqda açıqlayacağıq.

Bu necə işləyir...

Jurnal faylın ilk hissəsi onun quraşdırma faylında təyin etdiyiniz quraşdırmaları və cli-dan götürdüyünüz parametrləri təşkil edir. Bu seksiya aşağıdakı kimi başlayır:

```
Current Parameter Settings:  
config = 'example7-5-client.conf'
```

Aşağıdakı sətirlə bitir:

```
OpenVPN 2.3.2 amd64-portbld-freebsd9.2 [SSL (OpenSSL)] [LZO] [eurephia] [MH]  
[IPv6] built on Jan 9 2014
```

Bu seksiya 250 sətirə yaxındır və bu sətirlərdə OpenVPN-in quraşdırma haqqında nə düşündüyünü göstərir. Bu sətirləri diqqətlə oxuyun və əmin olun ki, sizdə qəbul edirsiniz.

Növbəti maraqlı seksiya aşağıdakıdır:

```
Outgoing Control Channel Authentication: Using 160 bit message hash 'SHA1'  
for HMAC authentication  
Outgoing Control Channel Authentication: HMAC KEY: 48cf6934 c7596aec ae92f34c  
7c4ff0c9 25a03f3a  
Outgoing Control Channel Authentication: HMAC size=20 block_size=20  
Incoming Control Channel Authentication: Using 160 bit message hash 'SHA1'  
for HMAC authentication  
Incoming Control Channel Authentication: HMAC KEY: baac7c20 c95bb2f2 a85a953a  
b7970b87 76045e8b  
Incoming Control Channel Authentication: HMAC size=20 block_size=20
```

Jurnal faylının bu hissəsi **tls-auth** açarının oxunulmasını göstərir və bu iki individual HMAC açarı çatdırılmışdır. Açarlar həmçinin jurnal faylında çap

edilmişdir. Həmçinin siz bu açarları artıq server jurnal faylından əldə edə bilərsiniz. Serverə daxil olan açar və client-in çıxışında olan açar tam olaraq eyni olmalıdır eynilə də əksinə. Öncəki misalımızda olduğu kimi açarların uyğunsuzluğunu burda da etsəydik jurnallarda görə bilərdik.

Bu bölmədən sonra tez **warning** görürük hansı ki, əsas səbəbi bu başlığımızın misalı olan Troubleshooting MTU və tun-mtu üçün təyin elədiyimiz uyğunsuzluqdur.

WARNING: normally if you use --mssfix and/or --fragment, you should also set --tun-mtu 1500 (currently it is 1400)

WARNING başlığı ilə gələn jurnal sətirlərinə həmişə xüsusi diqqətlə baxmaq lazımdır. Bəzi hallarda ona məhəl qoyulmaya bilər ancaq, əksər hallarda VPN qoşulmasının əsas problemi bu sətirdə göstərilə bilər.

Bu warning-dən sonra aşağıdakı ardıcılığa oxşayaraq çoxlu mesaj ardıcılığı gəlir:

```
UDPv4 WRITE [42] to [AF_INET]1.1.1.10:1194: P_CONTROL_HARD_RESET_CLIENT_V2
kid=0 pid=[ #1 ] [ ] pid=0 DATA len=0
UDPv4 READ [54] from [AF_INET]1.1.1.10:1194: P_CONTROL_HARD_RESET_SERVER_V2
kid=0 pid=[ #1 ] [ 0 ] pid=0 DATA len=0
```

Bu mesajlar server və client arasında olan razılaşmalardır hansı ki, quraşdırma informasiyası bölgüsü, şifrələnmiş açarlar və digər informasiyaları təşkil edir ki, VPN qalxsın. Bundan sonra başqa bir başlıq çıxır ki, quraşdırmada uyğunsuzluq var:

```
WARNING: 'link-mtu' is used inconsistently, local='link-mtu 1441',
remote='link-mtu 1541'
WARNING: 'tun-mtu' is used inconsistently, local='tun-mtu 1400', remote='tun-
mtu 1500'
```

Biz **TLS_prf** mesajlarının çox hissəsini buraxaraq gələcəyik serverin **push** quraşdırmalarının emalına:

```
PUSH: Received control message: 'PUSH_REPLY,route 10.198.0.0
255.255.0.0,route-gateway 192.168.200.1,topology subnet,ping 10,ping-restart
60,ifconfig 192.168.200.2 255.255.255.0'
```

Bu diqqətlə yoxlanılması vacib olan önəmli sətirdir ona görə ki, məhz burda serverin client-ə **push** edildiği göstərilir. Diqqətlə yoxlayın görək bu həqiqətən də siz yolladığınız PUSH-durmu ya başqası deyil ki?

Bundan sonra Local TUN adapter açılır və inisializasiya edilir və ilk paket axına başlaya bilər.

İlk ping əmri uğurla işləyir, aşağıda görə bilərik:

```
TUN READ [84]
...
UDPv4 WRITE [125] to server-ip:1194: P_DATA_V1 kid=0 DATA len=124
UDPv4 READ [125] from server-ip:1194: P_DATA_V1 kid=0 DATA len=124
TLS: tls_pre_decrypt, key_id=0, IP=server-ip:1194
TUN WRITE [84]
```

Ping əmri TUN interfeysi oxumağa başlayır sonra da şifrələnmiş kanalla uzaq serverə yazmağa başlayır. Paket həcmində olan fərqə diqqətlə baxın: paket şifrələnmiş kanalla **125** bayt ötürür hansı ki, TUN interfeysin original paket oxuma bacarığından **41** bayt böyükdür. Bu dəqiqliklə öncəki **log** sətirlərimizdə olan **link-mtu** və **tun-mtu** arasında olan fərqləri göstərir.

Sonra isə **ping -s 1450** seksiyası gəlir. **1450** baytlıq ping oxuna bilməz ona görə ki, interfeysdə təyin edilən **MTU 1400**-dur uyğun olaraq, 2 TUN hesab edir ki, bütün verilənləri ələ keçirmək lazımdır.

```
TUN READ [1396]
```

```
...
```

```
UDPv4 WRITE [1437] to server-ip:1194: P_DATA_V1 kid=0 DATA len=1436
```

```
TUN READ [102]
```

```
...
```

```
UDPv4 WRITE [141] to server-ip:1194: P_DATA_V1 kid=0 DATA len=140
```

Nəzərə alın ki, verilənlər əslində server iki ayrı paket kimi yollanılır. Bu tamamilə normal hərəkətdir ona görə ki, paket fragmentlərə bölünməsi tələb edilir. Paket həcmələrinin hesablanması MTU həcmi ilə müqayisə edilir və böyük olan halda hissəyə ayrılır və ikinci paket tam IP paket olmur.

Server böyük ping əmri qaytarır və uyğun olaraq geniş cavab qaytarılır.

Serverdə **1500 MTU** təyin edildiyinə görə orda tələb edilmir ki, datanı fragmentlərə böləsiniz ona görə də, bu client-ə tək paket kimi qayıdır:

```
UDPv4 READ [1441] from [AF_INET]1.1.1.10:1194: P_DATA_V1 kid=0 DATA len=1440
```

```
TLS: tls_pre_decrypt, key_id=0, IP=[AF_INET]1.1.1.10:1194
```

```
Authenticate/Decrypt packet error: packet HMAC authentication failed
```

Client paketi maximal həcm ilə yeni uzağı **1400** bayt ilə gözləyir. Buna görə də onun imkanı yoxdur ki, böyük paketi decode eləsin və bu səbəbdən də çıxışı yazır ki, **paket HMAC authentication failed** səhvi oldu.

Sonra biz client-i abort elədikdə görəəcəyik ki, **interrupted system call** mesajı çıxdı (Bu hissədə **Ctrl+C** istifadə edilmişdir ki, client-in qoşulmasını kəsək):

```
event_wait : Interrupted system call (code=4)
```

```
...
```

```
TCP/UDP: Closing socket
```

```
/sbin/route delete -net 10.198.0.0 192.168.200.1 255.255.0.0
```

```
delete net 10.198.0.0: gateway 192.168.200.1
```

```
Closing TUN/TAP interface
```

```
/sbin/ifconfig tun0 destroy
```

```
PID packet_id_free
```

```
SIGINT[hard,] received, process exiting
```

Əgər client quraşdırmasına aşağıdakı sətir əlavə edilibsə:

```
user nobody
```

Onda biz həmçinin aşağıdakı jurnal hissəsini görə bilərik:

```
SIOSIFADDR: Permission denied
```

```
SIOSIFFLAGS: Permission denied
```

**Linux ip addr del failed: external program exited with error status:
255**

Bu hal zərərsizdir.

Daha da ətraflı...

UNIX bazalı əməliyyat sistemlərində həmçinin mümkündür ki, jurnalı sistemin syslog-na yollaya bilərsiniz. Bu inzibatçıya şərait yaradır ki, effektiv şəkildə böyük məşinlərin jurnallarının yığılmasını bir jurnal interfeysində idarə edə bilərsiniz. Jurnal mesajlarını syslog-a ötürmək üçün sadəcə **log-append** direktivini **syslog** direktivi ilə dəyişməyiniz yetər:

syslog [name]

Burda **name** unikal deyil və OpenVPN serverin adının jurnallarda təyin edilməsi üçün istifadə edilir. Bu o halda çox önəmli olur ki, sizin bir host üzərində bir neçə OpenVPN serveriniz işləyir və onların hamısı öz jurnallarını syslog serverə yollayır.

BÖLÜM 8

OpenVPN: Routing troubleshooting

Bu başlıqda biz aşağıdakı mövzuları açıqlayacağıq:

- Çatışmayan qayıdış kodu.
- **iroute** istifadə ediləndə çatışmayan qayıdış route-ları.
- OpenVPN son nöqtələrindən başqa bütün clientləri funksional etmək
- Source routing
- Windows üzərində routing və yetki
- **client-to-client traffic routing** problemlərinin həllinin araşdırılması
- **'MULTI: bad source'** xəbərdarlıqlarının başa düşülməsi
- **Default gateway** yönləndirməsində çıxan səhv

Giriş

Bu bölümün müzakirəsi və öncəki bölümün mövzusu OpenVPN-in problemlərinin araşdırılmasıdır. Bu başlıq tamamilə OpenVPN üzərində çıxan routing problemlərinə həsr edilmişdir. OpenVPN istifadəçilərinin OpenVPN rəsmi saytına yazdıqları səhvlərin 50%-i əksər hallarda routing ilə bağlı olur və məhz bu başlıqda biz o problemlərin çox qismini açıqlayacağıq.

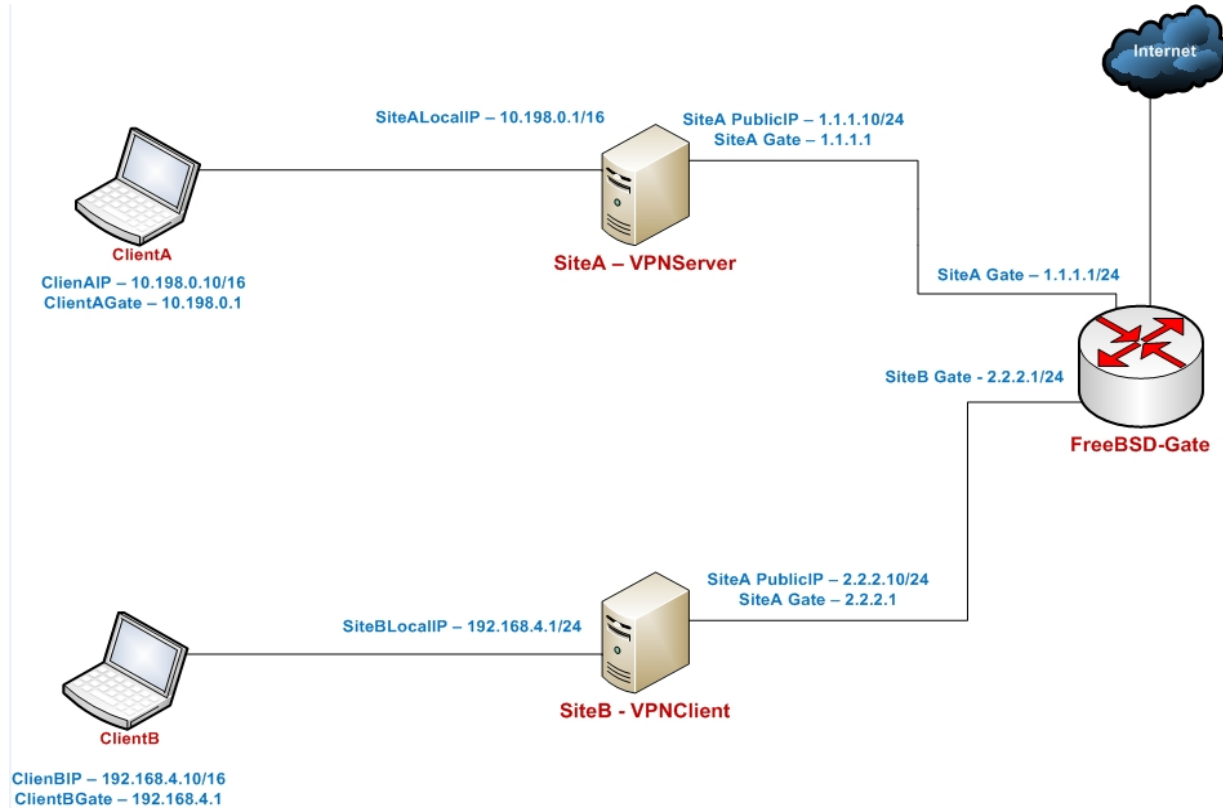
Ona görə də bu başlıqda istifadə elədiyimiz reseptlərdə öncə biz özümüz səhv quraşdırmalar edəcəyik və sonra problemin tapılması, səhv quraşdırmanın tapılması, aradan qaldırılması üçün lazımi alətləri göstərəcəyik.

Çatışmayan qayıdış kodu

OpenVPN-i ilk dəfə uğurla qurduqdan sonra əksər hallarda olur ki, OpenVPN şəbəkə route-ingləri səhv olsun. Bu misalda biz ilk olaraq 2-ci başlıqda Client-server IP şəbəkələrində olduğu kimi TUN stilli VPN quraşdıracağıq. İlk olaraq routing işləməyəcək o vaxtadək ki, düzgün routing əlavə edilməyəcək. Bu misalın əsas məqsədi routing səhvlərinin necə tapılmasının göstərilməsidir:

İşə hazırlaşaq:

Biz aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



OpenVPN2.3 ya da daha yuxarı versiyayı 2 məşində yükləyin. Əmin olun ki, maşınlar şəbəkə ilə bir-birlərini görürlər. 2-ci başlıqda istifadə edilən client və server sertifikatlarını burda da istifadə edəcəyik. Bu başlıqda server və client maşını üçün FreeBSD9.2 x64 və OpenVPN2.3 istifadə edəcəyik. Server quraşdırması üçün 2-ci başlıqda Server-side routing misalında istifadə elədiyimiz **basic-udp-server.conf** faylından istifadə edəcəyik. Client quraşdırma faylı isə **basic-udp-client.conf** olacaq.

Bunu necə edək...

1. **basic-udp-server.conf** faylını istifadə edərək serveri işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-server.conf
```
2. Sonra clienti işə salın:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config basic-udp-client.conf  
...  
Sun Mar 9 14:32:20 2014 Initialization Sequence Completed
```

3. Bu nöqtədə mümkündür ki, remote VPN IP ünvanı və bütün onda olan interfeyslərin IP ünvanlarını ping edə bilək:

```
root@siteB:~ # ping -c2 192.168.200.1  
PING 192.168.200.1 (192.168.200.1): 56 data bytes  
64 bytes from 192.168.200.1: icmp_seq=0 ttl=64 time=1.159 ms  
64 bytes from 192.168.200.1: icmp_seq=1 ttl=64 time=2.677 ms  
root@siteB:~ # ping -c2 10.198.0.10  
PING 10.198.0.10 (10.198.0.10): 56 data bytes  
64 bytes from 10.198.0.10: icmp_seq=0 ttl=127 time=1.716 ms  
64 bytes from 10.198.0.10: icmp_seq=1 ttl=127 time=3.214 ms
```

Əgər bu **'ping'**-lərdən hansısa biri uğursuz olarsa, onda qoşulma uğursuz olub və orda davam eləməyə ehtiyac yoxdur.

4. Əgər server tərəfin gateway-inə routing əlavə edilməmişsə, onda **10.198.0.0/16** şəbəkəsində olan bütün hostlar görünməyəcək (aşağıdakı kimi):

```
root@siteB:~ # ping 10.198.0.1  
PING 10.198.0.1 (10.198.0.1) 56(84) bytes of data.  
^C  
--- 10.198.0.1 ping statistics ---1 packets transmitted, 0 received,  
100% packet loss, time 764ms
```

5. Əgər siz LAN gateway üzərində route yazsanız ki, remote maşınları şifrələnmiş kanalla görə biləsiniz siz sadəcə serverə route yazmalısınız:

```
root@vpngate:~ # route add -net 192.168.200.0/24 1.1.1.10
```

Burda 1.1.1.1 VPN Gateway-imizin öz IP ünvanıdır. 1.1.1.10 maşını FreeBSD maşınıdır hansı ki, üstündə OpenVPN işləyir. Dəqiq routing yazmaq üçün sadəcə siz öz şəbəkənizi dəqiq bilməli və OS-nuzun routing əlavə etmək üçün sintaksisi bilməyiniz yəter.

6. Artıq VPN gateway üzərindən **192.168.200.0/24** şəbəkəsində olan bütün maşınları görə bilərik. Mütləq nəzərə alın ki, virtual TUN adapter işə düşdükdən sonra, onun üzərinə route yazdıqda işləmir.

Həmçinin 10.198.0.0/16 şəbəkəsini görmək üçün də geriye route yazmalıyıq:

```
root@vpngate:~ # route add -net 10.198.0.0/16 1.1.1.10
```

```
root@vpngate:~ # ping -c2 10.198.0.10  
PING 10.198.0.10 (10.198.0.10): 56 data bytes  
64 bytes from 10.198.0.10: icmp_seq=0 ttl=127 time=0.606 ms  
64 bytes from 10.198.0.10: icmp_seq=1 ttl=127 time=0.518 ms
```

Bu necə işləyir...

Client, Server tərəfdə olan hansısa host-a qoşulma cəhdi eləmək istədikdə paketlər source və destination IP ünvanla göndərilir:

- Source IP = 192.168.200.2: Bu VPN tunelin IP ünvanıdır
- Destination IP = Qoşulmaq istədiyimiz host-un IP ünvanı

Remote host isə müraciət edən paketə ünvanlarının source və destination ünvanlarını dəyişərək cavab vermək istəyir. Uzaq maşın paketi geri yollaya bilmir ona görə ki, **192.168.200.2** bizim VPN ünvanımızdır.

O sonra paketləri öz gateway üzərinə yönləndirməyə çalışır. Gateway-də cavab verməyəndə o onları default gateway üzərinə yönləndirməyə çalışır. Paketlər router-in üstünə çatdıqda isə, adətən router bütün paketləri drop edəcək ona görə ki, o göstərilən ünvanla çatmağa bilmir.

FreeBSD gateway üzərində route yazmaqla biz deyirik ki, **192.168.200.0/24** şəbəkəsi VPN serverin üzərinə yönləndirilməlidir - onda paketlər düzgün maşına qaydır. VPN server yenidən paketləri gateway maşına qaytara bildiyinə görə qoşulma uğurla başa çatmış olmuşdur.

Daha da ətraflı...

Bu hissədə biz diqqətimizi çıxan müxtəlif problemlərə yönləndirəcəyik.

Masquerading

Bu sualın tez və çirkli yolu 2-ci başlıqda Server-Side routing-də açıqlanmışdır. Orda masquerading istifadə edərək göstərdik ki, güya bütün trafik OpenVPN serverdən gəlir. Əgər siz uzaq gateway ünvanı idarə edə bilmirsinizsə, bu ideal variantdır. Bəzi program təminatları NAT edildikdə özünü yaxşı hiss edirlər. Həmçinin təhlükəsizlik baxımından NAT edilmənin üstünlüyü vardır.

LAN host-lara routing əlavə edilməsi

Gateway-in özünə route əlavə etməyin əvəzinə siz route-u remote maşınların özünə əlavə edə bilərsiniz ki, VPN clienti görə bilərsiniz. Bu əla seçimdir o halda ki, sizdən tələb edilir ki, VPN client lazımı hostlar tərəfindən görülebilsin.

Həmçinin baxın

- 2-ci başlıq Server-Side routing hansı ki, server tərəf routing trafikinin əsaslarını örgədir.

'iroute' istifadə ediləndə çatışmayan qayıdış route-ları

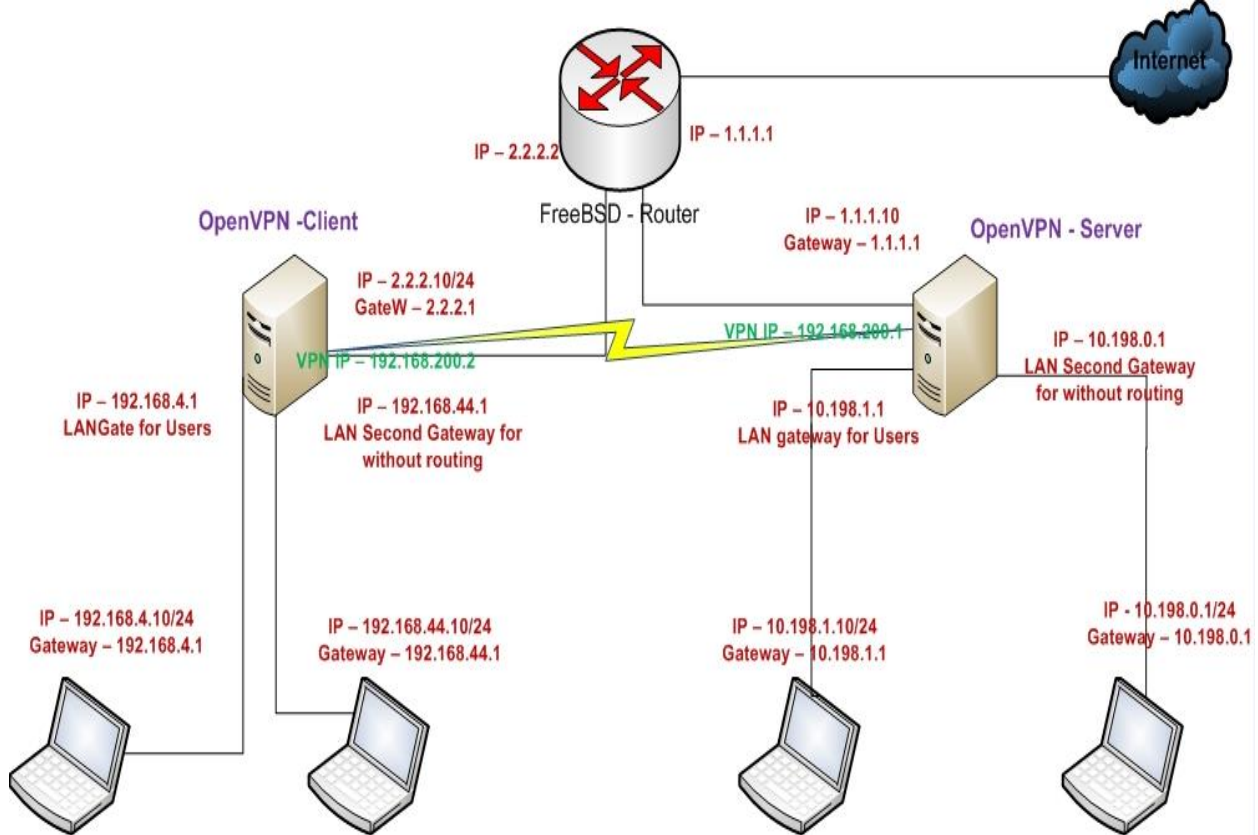
Bu misal öncəkinin davamıdır. Əmin olduqdan sonra ki, VPN clientin özü VPN server tərəfində olan bütün şəbəkələri görə bilər, indi vpn client tərəfdə olan local maşınlarında VPN server tərəfdə olan local şəbəkələri görməsinin yoxlanılmasının vaxtıdır.

Bu misalımızda biz öncə VPN-i 2-ci başlığın Routing: hər iki tərəfin subnetləri üçün misalına uyğun olaraq edəcəyik. Əgər route-lar yazılmayıbsa,

onda client tərəf lan-da olan maşınlar server tərəfdə olan maşınları və əksinə görə bilməyəcək. Lazımi route-ları əlavə etməklə problem həll ediləcək.

İşə hazırlaşaq

Biz aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



OpenVPN2.3-ü iki maşında yükləyin. Əmin olun ki, onlar bir birini şəbəkə ilə görürlər. 2-ci başlıqda istifadə edilən client və server sertifikatlarını burdada istifadə edin. Bu başlıqda client və server maşınları FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Server üçün 2-ci başlığın Routing: hər iki tərəfin subnetlərində misalında istifadə etdiyimiz **example2-5-server.conf** və client üçün isə 2-ci başlığın Server-side routing misalında istifadə etdiyimiz **basic-udp-client.conf** faylından istifadə edəcəyik.

Server üçün **example2-5-server.conf** quraşdırma faylının tərkibi:

```
proto udp
port 1194
dev tun
server 192.168.200.0 255.255.255.0
```

```
client-config-dir /usr/local/etc/openvpn/clients
```

```
ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnserver.crt
key /usr/local/etc/openvpn/openvpnserver.key
dh /usr/local/etc/openvpn/dh2048.pem
```

```
tls-auth /usr/local/etc/openvpn/ta.key 0

persist-key
persist-tun
keepalive 10 60

push "route 10.198.0.0 255.255.0.0"
topology subnet

user root
group wheel

daemon
log-append /var/log/openvpn.log

/usr/local/etc/openvpn/clients/openvpnclient1 clienti üçün xüsusi
quraşdırma fayli tərkibi:
iroute 192.168.4.0 255.255.255.0
iroute 192.168.44.0 255.255.255.0
```

Client maşınımızın **basic-udp-client.conf** client quraşdırması:

```
client
proto udp
remote openvpnsrvr.example.com
port 1194
dev tun
nobind

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnclient1.crt
key /usr/local/etc/openvpn/openvpnclient1.key
tls-auth /usr/local/etc/openvpn/ta.key 1

ns-cert-type server
```

Necə edəcəyik...

1. Serveri işə salaq:
root@siteA:/usr/local/etc/openvpn # **openvpn --config example2-5-server.conf**
2. Sonra client-i işə salın:
root@siteB:/usr/local/etc/openvpn # **openvpn --config basic-udp-client.conf**

...
... **Initialization Sequence Completed**
3. Bu nöqtədə mümkündür ki, remote VPN IP və VPN serverdə olan bütün interfeysləri ping edə bilək və əksinə server tərəfdən clientin VPN ip-si və bütün interfeyslərini ping edə bilək:
root@siteB:~ # **ping -c2 192.168.200.1**
PING 192.168.200.1 (192.168.200.1): 56 data bytes

```
64 bytes from 192.168.200.1: icmp_seq=0 ttl=64 time=0.878 ms
64 bytes from 192.168.200.1: icmp_seq=1 ttl=64 time=1.680 ms
root@siteB:~ # ping -c2 10.198.0.10
PING 10.198.0.10 (10.198.0.10): 56 data bytes
64 bytes from 10.198.0.10: icmp_seq=0 ttl=127 time=1.505 ms
64 bytes from 10.198.0.10: icmp_seq=1 ttl=127 time=2.450 ms
```

```
root@siteA:~ # ping -c2 192.168.200.2
PING 192.168.200.2 (192.168.200.2): 56 data bytes
64 bytes from 192.168.200.2: icmp_seq=0 ttl=64 time=0.807 ms
64 bytes from 192.168.200.2: icmp_seq=1 ttl=64 time=1.104 ms
```

```
root@siteA:~ # ping -c2 192.168.4.10
PING 192.168.4.10 (192.168.4.10): 56 data bytes
64 bytes from 192.168.4.10: icmp_seq=0 ttl=127 time=1.603 ms
64 bytes from 192.168.4.10: icmp_seq=1 ttl=127 time=2.409 ms
```

4. Server tərəfdə routing cədvəli göstərir ki, routing cədvəli düzgün route edilmişdir:

```
root@siteA:~ # netstat -rn | grep tun0
192.168.0.0/16      192.168.200.6      UGS          0          0      tun0
192.168.4.0/24    192.168.200.1      UGS          0          4      tun0
192.168.44.0/24   192.168.200.1      UGS          0          0      tun0
192.168.200.0/24  192.168.200.1      UGS          0          11     tun0
192.168.200.1     link#11             UH           0          0      tun0
```

5. Siz server tərəfdə olan host-a client tərəfdən ping yollamağa cəhd elədikdə qırılma olmayacaq çünki routing-lər VPN qoşulması səviyyəsində əlavə edilmişdir. Yeni ki, bütün routinglərin düzgün işləməsi üçün biz server quraşdırma faylında öncədən client üçün **192.168.4.0/24** və **192.168.44.0/24** şəbəkəsi üçün routing və serverin öz şəbəkəsinin **10.198.0.0/16**-nin client-ə route ilə ötürülməsi öncədən nəzərə alınmışdır.

Ancaq bu routing-ləri siz olduğunuz topologiyadan və ünvanından asılı olaraq özünüz də əlavə edə bilərsiniz. Problemin həlli üçün siz **tcpdump**, **ping** və **traceroute** əmrlərindən istifadə edə bilərsiniz.

Bu necə işləyir...

Bu qısa routing probleminin araşdırılması ilə məşğul olursunuzsa, vacibdir ki, öncə dərin şəbəkədən başlayasınız (Bizim halda VPN) və ondan sonra kənar trafik haqqında düşünəsiniz:

- Öncə əmin olun ki, VPN-in son nöqtələri bir-birlərini görürlər
- Əmin olun ki, VPN client maşını VPN server maşınının LAN ip ünvanını görür və əksinə eynilə Server clientin LAN IP-sini görür.
- Əmin olun ki, VPN client maşını VPN server maşınının LAN-ında olan maşını görür
- Əmin olun ki, serverin LAN tərəfində olan host VPN client-i görür.
- Əmin olun ki, client-in LAN tərəfində olan host VPN serveri görür
- Sonda əmin olun ki, client tərəfin LAN-ında olan host server tərəfin LAN-ında olan host-u görür və əksinə.

Həmçinin baxın

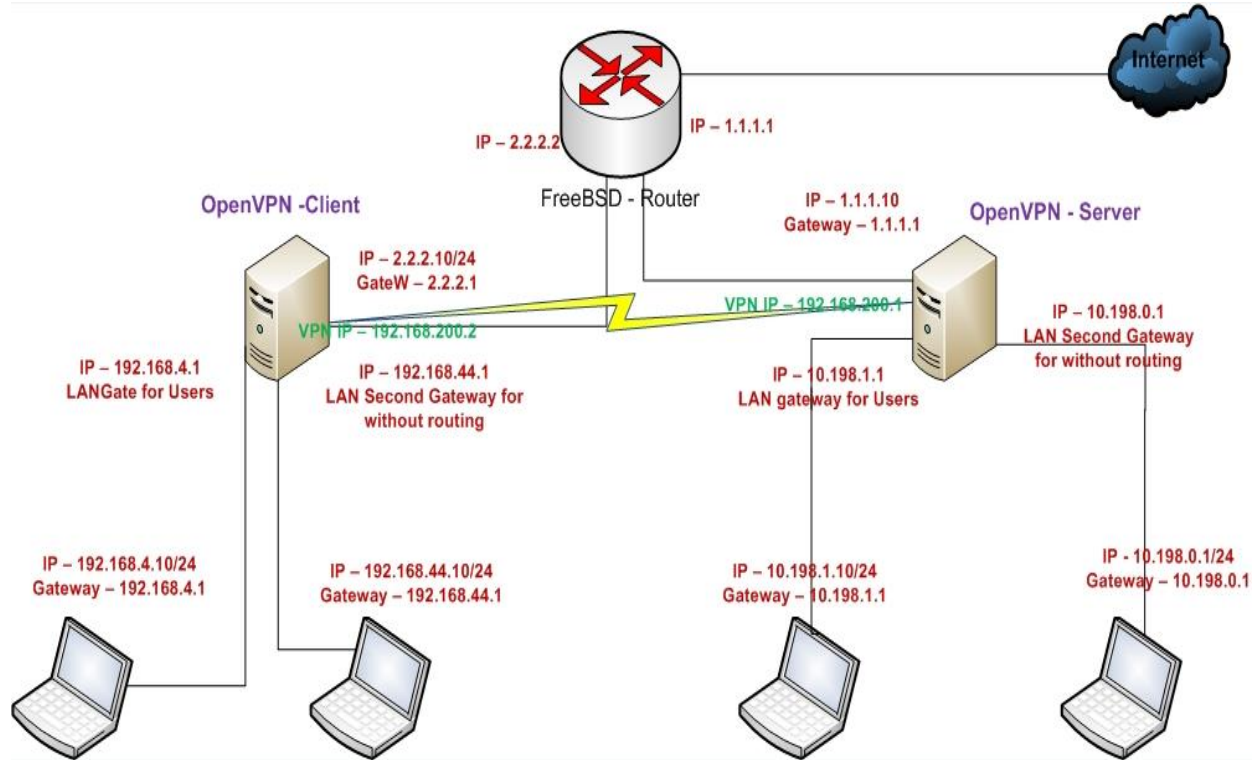
- 2-ci başlığın misalında, Routing: hər iki tərəfin subnetlərində hansı ki, hər iki tərəfdə olan routing işini detallı açıqlayır

OpenVPN-in son nöqtələrindən başqa bütün clientləri funksional etmək

Bu misal yenə də öncəkinin davamıdır. Öncəki misal açıqladı ki, client tərəf LAN(Yada subnet)-dan, server tərəf LAN-a qoşulduqda routing problemlərini necə həll etmək lazımdır. Ancaq öncəki misalda quraşdırma səhvləri qoyulmalı idi ki, səhv ortaya çıxsın(ancaq bu edilmədi). Bu misalda biz boşluğu qoyaraq problem araşdırmağa çalışacağıq.

İşə hazırlaşaq

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



OpenVPN2.3-ü iki maşında yükləyin. Əmin olun ki, onlar bir birini şəbəkə ilə görürlər. 2-ci başlıqda istifadə edilən client və server sertifikatlarını burda da istifadə edin. Bu başlıqda client və server maşınları FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Server üçün 2-ci başlığın Routing: hər iki tərəfin subnetlərində misalında istifadə etdiyimiz **example2-5-server.conf** və client üçün isə 2-ci başlığın Server-side routing misalında istifadə etdiyimiz **basic-udp-client.conf** faylından istifadə edəcəyik.

Necə edəcəyik...

1. Serveri işə salın:
root@siteA:/usr/local/etc/openvpn # **openvpn --config example2-5-server.conf**
2. Sonra clienti işə salın:
root@siteB:/usr/local/etc/openvpn # **openvpn --config basic-udp-client.conf**
...
... **Initialization Sequence Completed**
3. Nəzərə alsaq ki, həm client və həm server tərəfdə routinglər olmasa, onda nə client nədə server qarşı tərəfin hostlarını görməyəcək. Deyək ki, routing-lər yoxdur. Aşağıda göstərilən routingləri hər iki tərəfdə əlavə edin:

Öncə client tərəfdə edək:

```
root@siteB:~ # ifconfig tun0
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> metric 0 mtu 1500
      options=80000<LINKSTATE>
      inet 192.168.200.2 --> 192.168.200.2 netmask 0xffffffff00
      Opened by PID 1770
root@siteB:~ # route add -net 10.198.0.0/16 192.168.200.1
add net 10.198.0.0: gateway 192.168.200.1 fib 0
```

Sonra server tərəfdə edək:

```
root@siteA:/usr/local/etc/openvpn # ifconfig tun0
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> metric 0 mtu 1500
      options=80000<LINKSTATE>
      inet 192.168.200.1 --> 192.168.200.1 netmask 0xffffffff00
      Opened by PID 1809

root@siteA:/usr/local/etc/openvpn # route add -net 192.168.4.0/24 192.168.200.2
add net 192.168.4.0: gateway 192.168.200.2 fib 0
```

4. Hər iki tərəfin LAN-ında olan maşınları yoxlayaq:

```
root@siteB:~ # ping -c2 10.198.0.10
PING 10.198.0.10 (10.198.0.10): 56 data bytes
64 bytes from 10.198.0.10: icmp_seq=0 ttl=127 time=1.659 ms
64 bytes from 10.198.0.10: icmp_seq=1 ttl=127 time=2.324 ms
```

```
root@siteA:/usr/local/etc/openvpn # ping -c2 192.168.4.10
PING 192.168.4.10 (192.168.4.10): 56 data bytes
64 bytes from 192.168.4.10: icmp_seq=0 ttl=127 time=1.631 ms
64 bytes from 192.168.4.10: icmp_seq=1 ttl=127 time=2.117 ms
```

```
C:\Users\clientb>ping -n 2 10.198.0.1
Pinging 10.198.0.1 with 32 bytes of data:
Reply from 10.198.0.1: bytes=32 time=1ms TTL=63
Reply from 10.198.0.1: bytes=32 time=2ms TTL=63
```

```
C:\Users\ClientC>ping -n 2 192.168.4.10
Pinging 192.168.4.10 with 32 bytes of data:
Reply from 192.168.4.10: bytes=32 time=2ms TTL=126
Reply from 192.168.4.10: bytes=32 time=3ms TTL=126
```

Hamısı işləyir çünki, biz onun işləməsi üçün bütün routingləri artıq əlavə etmişik.

5. Linux/UNIX maşınlarında biz mənbəyə əsaslanaraq istənilən mənsəbə ping ata bilərik:

```
root@siteA:/usr/local/etc/openvpn # ping -S 10.198.0.1 -c2 192.168.4.10
PING 192.168.4.10 (192.168.4.10) from 10.198.0.1: 56 data bytes
64 bytes from 192.168.4.10: icmp_seq=0 ttl=127 time=1.623 ms
64 bytes from 192.168.4.10: icmp_seq=1 ttl=127 time=2.241 ms
```

Gördüyümüz kimi hər şey işləyir.

Bu necə işləyir...

Məqsəd o idi ki, VPN server və onun tərəfində olan host, client tərəfi və onun hostunu görə bilsin. Bu halda VPN server client-ə müraciət yolladıqda paket birbaşa VPN serverin interfeysi üzərindən keçmiş olacaq. Paket aşağıdakı quruluşda olacaq:

- Mənbə IP = 192.168.200.1 VPN Serverin özü
- Mənsəb IP = 192.168.4.10 SiteB LAN Host IP ünvanı

Eynilə paket VPN client-dən serverə tərəf qayıtdıqda o mənbə və mənsəbin ünvanı dəyişmiş şəkildə gedəcək.

Daha da ətraflı...

Bu misalda ən yaxşı istifadə üsulu NAT olardı. Biz bunu masquerading ilədə edə bilərik. Ancaq öncəki misallarında hər şey FreeBSD-yə aid elədiyim üçün aşağıda yalnız Linux masquerading misallarını göstərəcəm.

```
[root@server]# iptables -t nat -I POSTROUTING -i tun0 -o eth0 -s
192.168.200.0/24 -j MASQUERADE
[root@client]# iptables -t nat -I POSTROUTING -i tun0 -o eth0 -s
192.168.200.0/24 -j MASQUERADE
```

Gateway-lər üzərində əlavə routing-ə ehtiyac qalmır.

Bunu eləmək LINUX/UNIX-də həddən artıq asandır, Windows-a baxanda.

Həmçinin baxın

2-ci başlıqda olan Routing: hər iki tərəfin subnetlərində işini oxuyun hansı ki, detallı şəkildə routing işini açıqlayır.

Source routing

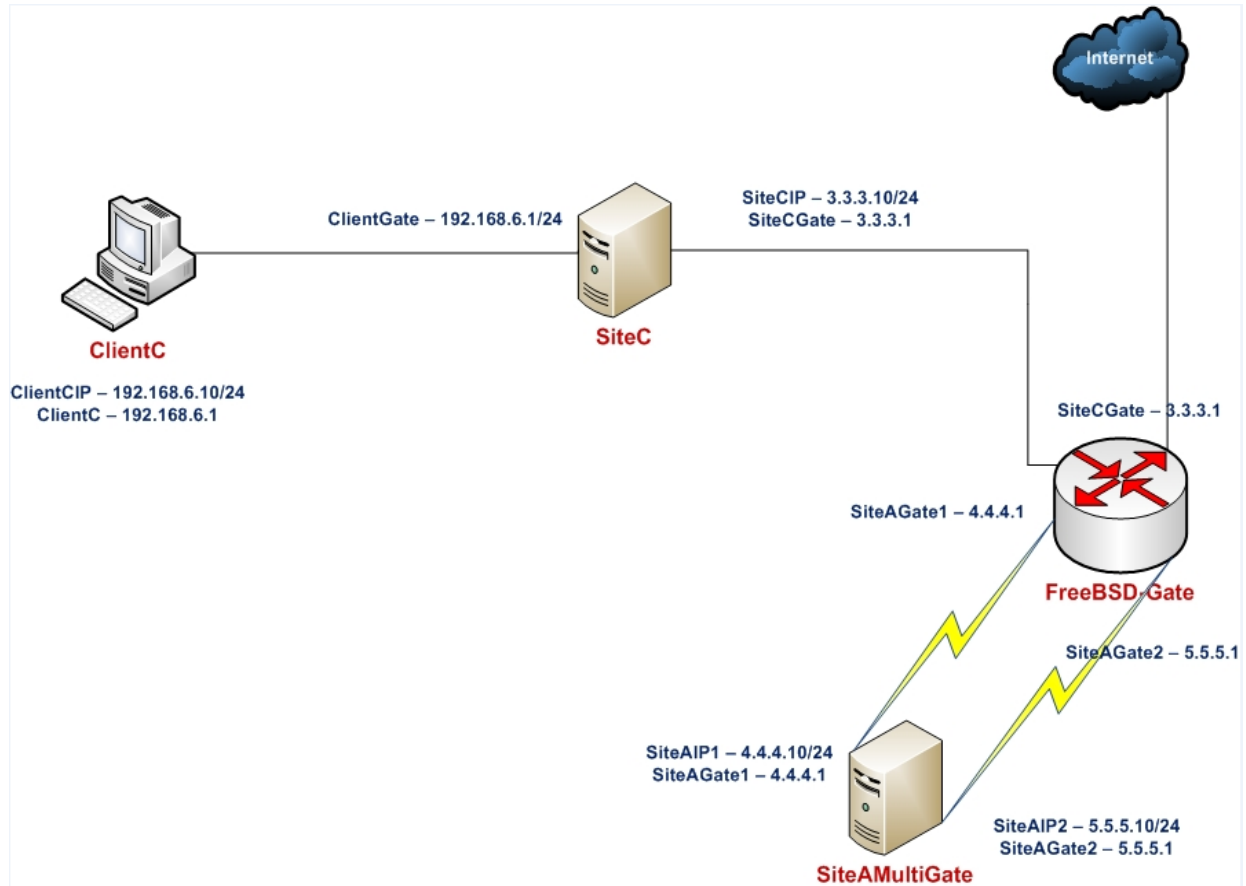
Şəbəkə quraşdırmaları həddən artıq çətinləşdiyinə görə, genişlənmiş imkanlara görə də tələblər böyüyür hansı ki, misal olaraq source routing imkanları.

Source routing o halda istifadə edilir ki, server şəbəkəyə iki şəbəkə kartı ilə qoşulmuşdur. Bu misalda önəmli hissə odur ki, əmin olasınız ki, qoşulmalar interfeysin birində işə düşmüşdür və həmin interfeysdə də saxlanılıb. Əgər qoşulma üçün gələn VPN trafiki ilk interfeysə gəlibse və qayıtmaq üçün 2-ci interfeysdən qayıdırsa, onda VPN qoşulma digərləri arasında kəsiləcək. Biz bunu bu misalda göstərəcəyik.

Mənbəyə görə routing edilməsi hal-hazırki əksər əməliyyat sisteminin bacarığıdır. Bu misalda biz Linux üzərində **iproute2** alətləri ilə mənbəyə görə routing göstərəcəyik. Həmçinin eyni işi FreeBSD və digər OS-lar üzərində də görə bilərik.

İşə hazırlaşaq

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik



OpenVPN2.3-ü iki maşına yükləyin. Əmin olun ki, maşınlar bir-birlərini şəbəkə ilə görürlər (Şəbəkəni aşağıda detallı şəkildə açıqlayacağıq). Server maşınımız FreeBSD9.2 x64 OpenVPN2.3 və client maşınımız isə Windows7 x64 OpenVPN2.3-də olacaq. Client maşının IP ünvanı 192.168.6.10-dur və onun üçün gateway IP ünvan 192.168.6.1-dir. Client maşın üçün gateway server sitec-nin PUBLIC IP ünvanı isə 3.3.3.10-dur hansı ki, üstünə gələn müraciətləri dünyaya 3.3.3.10 IP ünvan ilə NAT edir. Client Gateway maşının və OpenVPN server maşının gatewayi FreeBSD-Gate maşındır. SiteC üçün gateway 3.3.3.1-dir. OpenVPN server üçün isə 2 ədəd Gateway var 1-ci gateway IP 4.4.4.1 və ikinci

gateway IP 5.5.5.1-dir. OpenVPN serverin ilk IP ünvanı 4.4.4.10 və ikinci gateway IP-si isə 5.5.5.10-dur. OpenVPN Server üçün quraşdırma faylı **basic-tcp-server.conf** istifadə edəcəyik. Windows7 Client üçün isə **basic-tcp-client.ovpn** faylından istifadə edəcəyik.

Məqsədimiz OpenVPN serverin üzərində iki ədəd default gateway olarsa, o halda onun üzərinə bir public-dən gələn paket qayıdanda anlamayacaq ki, hansı şəbəkə kartı ilə geriye qayıtmalıdır. Bu halda bizim köməyimizə **PBR** çatır(Firewall rule-larına əsaslanan Policy Based Routing).

Necə edək...

1. Öncə Multigateway OpenVPN maşının kernelini lazımi opsiyalarla kompilyasiya etmək lazımdır:

```
# NUFFERS
maxusers      512
options       NBUF=4096
device        if_bridge

# routing MPATH and for multiple services
options       RADIX_MPATH
options       ROUTETABLES=15

# IPFW FireWall
options       IPFWALL
options       IPFWALL_VERBOSE
options       IPFWALL_VERBOSE_LIMIT=3
options       IPFWALL_NAT
options       LIBALIAS
options       DUMMYNET
options       IPFWALL_FORWARD

# PF FireWall
device        pf
device        pflog
device        pfsync
```

OpenVPN Serverimizin startup quraşdırma faylı yeni **/etc/rc.conf** aşağıdakı kimi olacaq:

```
hostname="siteA-MultiGate"
ifconfig_em0="inet 4.4.4.10 netmask 255.255.255.0"
ifconfig_em1="inet 5.5.5.10 netmask 255.255.255.0"
sshd_enable="YES"
dumpdev="NO"
firewall_enable="YES"
```

OpenVPN Serverimizin MultiDefault Route faylı yeni **/etc/rc.local** aşağıdakı kimi olacaq:

```
# default route-larımızı təyin edirik
setfib 0 route delete default
setfib 0 route add default 4.4.4.1
setfib 1 route delete default
setfib 1 route add default 5.5.5.1
```



```
# route table-larımızı interfeyslərimizə mənimsədirik
ipfw -f flush
ipfw add allow ip from any to any via lo0
ipfw add setfib 0 ip from any to any via em0
ipfw add setfib 1 ip from any to any via em1
ipfw add allow ip from any to any
```

Sonra şəbəkə servislərimizi yenidən yükləyirik.

```
/etc/rc.d/netif restart
/etc/rc.d/local restart
```

OpenVPN serverə **reboot** edirik və sonra route cədvəlimizə həm **setfib 0** və həm də **setfib 1** üçün baxırıq:

```
root@siteA-MultiGate:/usr/local/etc/openvpn # setfib 0 netstat -rn
Routing tables
```

Internet:

| Destination | Gateway | Flags | Refs | Use | Netif | Expire |
|----------------|----------------|------------|----------|------------|------------|--------|
| default | 4.4.4.1 | UGS | 0 | 621 | em0 | |
| 4.4.4.0/24 | link#2 | U | 0 | 0 | em0 | |
| 4.4.4.10 | link#2 | UHS | 0 | 0 | lo0 | |
| 5.5.5.0/24 | link#4 | U | 0 | 567 | em1 | |
| 5.5.5.10 | link#4 | UHS | 0 | 0 | lo0 | |
| 127.0.0.1 | link#9 | UH | 0 | 16 | lo0 | |

```
root@siteA-MultiGate:/usr/local/etc/openvpn # setfib 1 netstat -rn
Routing tables
```

Internet:

| Destination | Gateway | Flags | Refs | Use | Netif | Expire |
|----------------|----------------|------------|----------|----------|------------|--------|
| default | 5.5.5.1 | UGS | 0 | 4 | em1 | |
| 4.4.4.0/24 | link#2 | U | 0 | 0 | em0 | |
| 5.5.5.0/24 | link#4 | U | 0 | 0 | em1 | |
| 127.0.0.1 | link#9 | UH | 0 | 0 | lo0 | |

Qeyd: Nəzərə alın ki, bu quruluş öz praktikamda OpenVPN-in UDP ilə qoşulmasında işləmədi və **Connection reset by peer code=10054** səhvi verdi. Ona görə də həm client və həm də server tərəfdə **tcp** quraşdırmalarından istifadə edəcəyik.

Serverimizi işə salırıq:

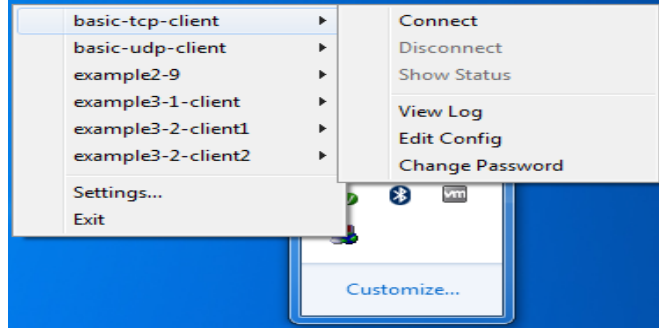
```
root@siteA-MultiGate:/usr/local/etc/openvpn # openvpn --config basic-  
tcp-server.conf
```

- Client-i işə salmazdan öncə açıqlama vermək istərdim ki, client həm **4.4.4.10** və həm də **5.5.5.10** IP ünvanlarına qoşulma eləsə problemsiz işləməlidir çünki, bu iki IP ünvanın hər biri eyni serverdə olsa da o server paketin hansı interfeysdən gəldiyini idarə edib o interfeyslə də geriye qaytaracaq (Yeni **PBR** işləyəcək). Hal-hazırda client maşınımızın **c:\windows\system32\drivers\etc\hosts** faylında aşağıdakı sətirlərimiz mövcuddur:

```
4.4.4.10          openvpnsrvr.example.com
#5.5.5.10        openvpnsrvr.example.com
```

Öncə 4.4.4.10 IP ünvanını test edin, nəticə əldə edildikdən sonra isə 4.4.4.10 IP sinin qarşısına şərh təyin edib, 5.5.5.10 IP ünvanın qarşısından şərh silin və faylı yadda saxlayıb, yenidən VPN qoşulmasını sınaqdan keçirin.

Sonra VPN client-i işə salaq:



Qoşulma uğurlu olduqdan sonra isə Windows7-nin hosts faylında dəyişiklik edirik ki, 5.5.5.10 IP-si tərəfdən qoşulma edək. Və ardınca VPN-i yenidən işə salıb test edirik.

Bu necə işləyir...

OpenVPN serverin adi halda 2 ədəd default gateway-i olduqda və ona hansısa bir gateway tərəfdən paket gəldikdə, FreeBSD OS dəqiq qərar verə bilmir ki, paketi hansı default gateway tərəfdən qaytarsın və əməlli qarışıqlıq baş verir. Bunun üçün OS üzərindən IPFW ilə PBR etdik ki, paket 4.4.4.1 tərəfdən gəldikdə həmin gateway ilə və 5.5.5.1 tərəfdən gələrsə həmin gateway ilə geriye qayıtsın.

Daha da ətraflı...

Daha da geniş routing idarə etmək istəsəniz, LINUX üzərində **LARTC** (Linux Advanced Routing and Traffic Control) istifadə edə bilərsiniz. Ən yaxşı üsulu interfeysə gələn paketləri qeydə alıb eyni interfeys ilə geriye qaytarmaqdır.

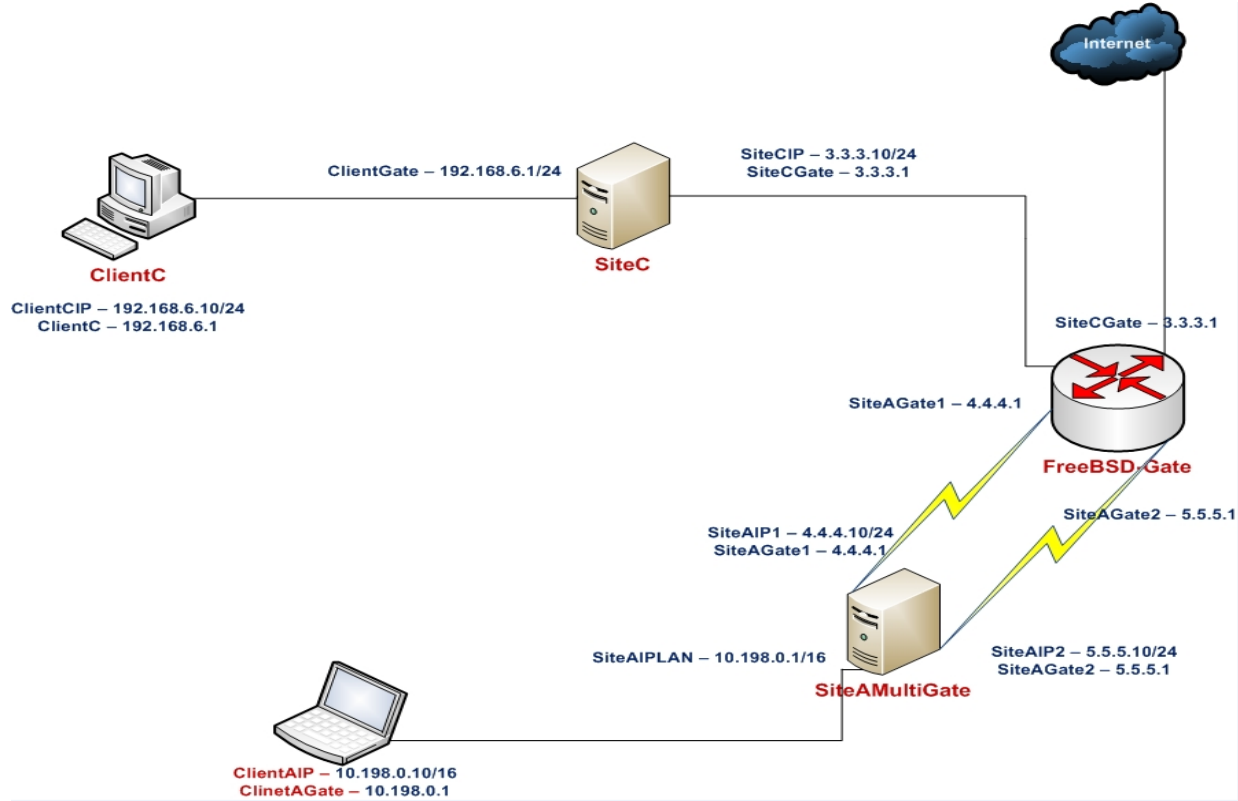
Windows üzərində routing və yetki

Bu misalımızda biz VPN client-in Windows maşında işləyəndə, yetkiləri olmaması üzündən praktikada görülən səhvlərdir. Bu halda OpenVPN server uğurla qoşulacaq ancaq, lazımi yetkilərin olmamasından routing sistemə əlavə edilməyəcək və VPN düzgün işləməyəcək. Bu başlıqda biz bu problemin tapılması və aradan qaldırılması işini görəcəyik. Bu səhv quraşdırma adətən Windows7/8 və Windows Server 2008-də olur.

İşə hazırlaşaq

Bu başlıqda biz OpenVPN2.3 versiyasını 2 maşında istifadə edəcəyik. Əmin olun ki, maşınlar şəbəkə ilə bir-birlərini görürlər. 2-ci başlıqda yaradılan client və server sertifikatlarını burda da istifadə edəcəyik. Server maşını 2 ayrı şəbəkə kartı ilə ayrı-ayrı Internet təchizatçısına qoşulmalıdır. Bu

misalda server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq və 2-ci başlıqda Server-side routing-də yaradılan **basic-udp-server.conf** faylından istifadə edəcək. Client maşın isə Windows7 x64 OpenVPN2.3-də olacaq. Client quraşdırma faylı isə 2-ci başlıqda **ifconfig-pool** bölümündə istifadə edilən **basic-udp-client.ovpn** quraşdırma faylı olacaq. Şəbəkə quruluşu aşağıdakı kimi olacaq:



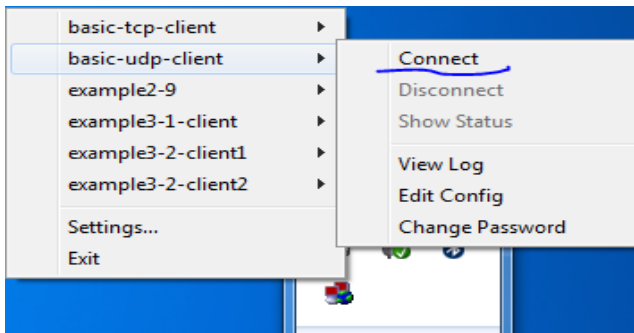
Necə edək...

1. Yetkisi olmayan istifadəçi adından sistemə daxil olun (Yəni ki **Administrator** olmayan istifadəçi adından).

2. **basic-udp-server.conf** faylını istifadə edərək serveri işə salın:

```
root@siteA-MultiGate:/usr/local/etc/openvpn # openvpn --config basic-udp-server.conf
```

3. Sonda client-i işə salın:



OpenVPN işə düşəcək və OpenVPN GUI-nin yaşıl işığı yanacaq. Ancaq client maşını aşağıdakı jurnalı çap edəcək. (Sözün düzü mənim halımda həm serverdə və

həm də client-də OpenVPN2.3 istifadə edilirdi. Ancaq məndə Windows7 client-də route adi istifadəçi adından (**Run as administrator** olduqda) problemsiz əlavə edildi.):

```
... C:\WINDOWS\system32\route.exe ADD 10.198.0.0 MASK 255.255.0.0
192.168.200.1
... ROUTE: route addition failed using CreateIpForwardEntry: Network
access is denied. [status=65 if_index=2]
... Route addition via IPAPI failed [adaptive]
Thu Aug 26 16:47:53 2010 us=187000 Route addition fallback to route.exe
```

Əgər siz server tərəfdə olan LAN şəbəkə karta çatmaq istəsəniz aşağıdakı səhvi görəcəksiniz:

```
[WinClient]C:\>ping 10.198.0.1
Pinging 10.198.0.1 with 32 bytes of data:
Request timed out.
Ping statistics for 10.198.0.1:
Packets: Sent = 1, Received = 0, Lost = 1 (100% loss),
```

Bu problemin həlli istifadəçiyə lazımi şəbəkə yetkilərin verilməsidir. Bunun üçün istifadəçini ya **Administrators** yada **Network Administrators** qrupuna əlavə etmək yetər.

Bu necə işləyir...

OpenVPN client TAP-Win32 adapter-i işə salmağa çalışır hansı ki, susmaya görə olan yüklənmədə izin verilmişdir. O halda ki server, client üçün aşağıdakı route sətirini ötürmək istəyir:

```
push "route 10.198.0.0 255.255.0.0"
```

Onda, OpenVPN client bu route-u öz routing cədvəlinə əlavə edə bilməyəcək. VPN client uğurla qoşulacaq və uğurlu qoşulma GUI-də görünəcək.

Qeyd: Nəzərə alın ki, hətta, **push route** sütunu olmasa da belə OpenVPN GUI-də yaşıll olacaq və qoşulma işə düşəcək.

Daha da ətraflı...

Həmçinin **Run As Administrator** imkanından istifadə edə bilərsiniz ki, OpenVPN servisini XP/Windows7/Windows8,8.1-də Administrator adından işə salasınız. Bu bütün problemlərin ən əsas həll üsuludur.

Həmçinin baxın

- 10-cu başlıq, Integrasiya, bu bölümdə olan misallarda OpenVPN-in Windows OS-a necə integrasiya edilməsi göstərilir.

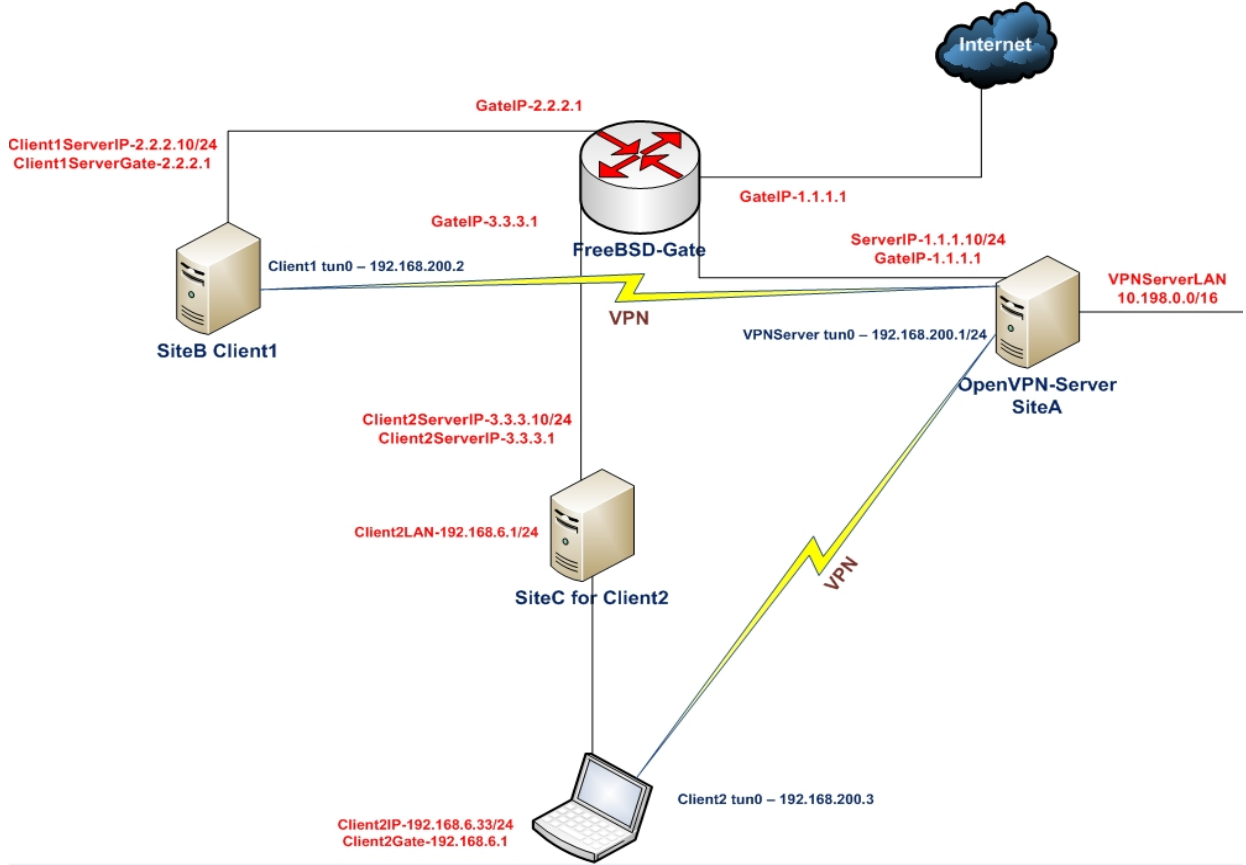
client-to-client traffic routing probleminin həllərinin araşdırılması

Bu misalımızda biz VPN quruluşunda client-dən client-ə trafikini ötürülməsini istəyən halda, VPN quraşdırmasının içində '**client-to-client**' direktivi olmadıqda çıxan problemlərin araşdırılması qaydasını öyrənəcəyik. TUN tipli şəbəkə kartlarında client-dən client-ə trafikini ötürülməsi bu direktivi

istifadə eləməsənzədə olacaq(Sözsüz ki, əgər server-in admin-i firewall-la bu trafiklərin bir-bilərini görməsinə izin vermişdirsə işləyəcək). Ancaq TAP stilli şəbəkə kartlarında bu mümkün deyil.

İşə hazırlaşaq

Biz aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



OpenVPN2.3-ü **3** məşində yükləyin. Əmin olun ki, məşinlər şəbəkə ilə bir-birini görürlər. 2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalımızda server məşinimiz FreeBSD9.2 x64 OpenVPN2.3-də olacaq. İlk client-imiz FreeBSD9.2 x64 OpenVPN2.3-də və ikinci client-imiz isə Windows7 x64 OpenVPN2.3-də olacaq. Server üçün 2-ci başlıqda server-side routing-də istifadə elədiyimiz **basic-udp-server.conf** faylından və eynilə FreeBSD client üçün orda istifadə edilən **basic-udp-client.conf**-dan istifadə edəcəyik. Windows7 client üçün isə 2-ci başlıqda **'ifconfig-pool'** misalında yaratdığımız **basic-udp-client.ovpn** faylından istifadə edəcəyik.

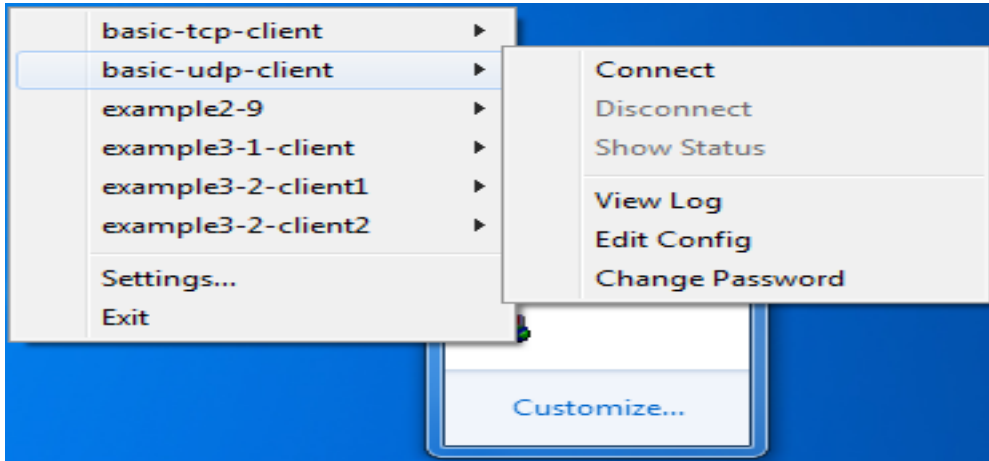
Bunu necə edək...

1. **basic-udp-server.conf** faylından istifadə edərək serveri işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-server.conf
```
2. FreeBSD məşini işə salaq:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config basic-udp-client.conf
```

3. Sonra Windows7 client-i adi istifadəçi adından işə salaq:



4. Sonra FreeBSD client maşından Windows7 client maşına ping atmağa çalışın. Əmin olun ki, heç bir firewall trafiki block eləmir. (Düzdür mənim halımda Windows7 maşında yenə də adi istifadəçi adından hər şey işlədi. Ancaq yazarın adından davam edək)

```
root@siteB:~ # ping -c 2 192.168.200.3  
PING 192.168.200.3 (192.168.200.3): 56 data bytes  
PING 192.168.200.3 (192.168.200.3) 56(84) bytes of data.  
--- 192.168.200.3 ping statistics ---2 packets transmitted, 0 received,  
100% packet loss, time 10999ms
```

Ola bilər ki, siz problemsiz digər hostu görəsiniz. Ancaq bu halda firewall portu block eləmişdi.

5. Bunun üçün VPN server-in firewall-unda jurnal rejimini aktivləşdirməyiniz düzgündür:

Linux üçün aşağıdakı kimi olacaq.
[root@server]# **iptables -I FORWARD -i tun+ -j LOG**

FreeBSD-də isə **/etc/rc.conf** faylına **firewall_logging="YES"** əlavə etməyiniz yetər.

Sonra yenidən ping edin. Siz sistemin **/var/log/messages** jurnal faylında nəticəni görəcəksiniz:

```
... openvpnsrver kernel: IN=tun0 OUT=tun0 SRC=192.168.200.2  
DST=192.168.200.3 LEN=84 TOS=0x00 PREC=0x00 TTL=63 ID=0 DF PROTO=ICMP  
TYPE=8 CODE=0 ID=40808 SEQ=1  
... openvpnsrver kernel: IN=tun0 OUT=tun0 SRC=192.168.200.2  
DST=192.168.200.3 LEN=84 TOS=0x00 PREC=0x00 TTL=63 ID=0 DF PROTO=ICMP  
TYPE=8 CODE=0 ID=40808 SEQ=2
```

İlk client 192.168.200.2 çalışır ki, ikinci client olan 192.168.200.3 IP ünvanına çatsın. Bu problemin həlli server quraşdırma faylına **client-to-client** direktivinin əlavə edilməsi və OpenVPN daemon-un restart edilməsidir.

Yada ki, tunel trafikinin yönləndirilməsi ilə həll edilə bilər. Hal-hazırda LINUX üçün göstərəcəyik ancaq UNIX PF-də öncəki başlıqlarımızda göstərmişdik:

```
[server]# iptables -I FORWARD -i tun+ -o tun+ -j ACCEPT  
[server]# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Bu necə işləyir...

İlk client, digərinə çatmaq istəyəndə, paketlər həmçinin serverin özünə də çatır. OpenVPN server onların necə emal edilməsini bilmir və kernel tərəfdən emal edilməsinin söndürülməyini bilmir. Kernel isə paketləri yönləndirir hansı ki, routing-ə və firewall rule-ların əsasında qəbul edilmişdir. Əgər qəbul edilməmişdirsə, onda paket sadəcə drop edilir və ikinci client-ə heç vaxt çatmaq olmur.

Aşağıdakı direktivin əlavə edilməsi ilə, OpenVPN server kernel trafikini yönləndirilməsi və firewall qaydalarını aşaraq paketi bir client-dən digərinə çatdırır.

client-to-client

Digər üsulu isə daha da təhlükəsiz olan UNIX/Linux kernel vasitəsilə routing işinin görülməsidir.

Daha da ətraflı...

TAP stilli şəbəkələrdə öncə göstərilən IPTABLES qaydası işləməyəcək. TAP stilli şəbəkələrdə olan bütün clientlər eyni broadcast domain-in üzvü olur. client-to-client trafiki yazılmayanda və bir client digərinə çatmaq istəyərsə, ilk olaraq o **'arp who has'** mesajı ilə digər client-in MAC ünvanını tapır. OpenVPN server bu müraciətləri iptables qaydasının olub olmadığından asılı olmayaraq məhəl qoymayacaq və onları digər clientlərə yönləndirməyəcək. Ardıcıl olaraq client asan yolla client-to-client direktivi olmadan digər client-ə çata bilməyəcək, əgər **proxy-ARP** istifadə edilmirsə.

Həmçinin baxın...

- 3-cü başlıqda olan, client-to-client trafikini aktiv edilməsi hansı ki, client-to-client trafikinin TAP stilli şəbəkələrdə açıqlanması göstərilir.

'MULTI: bad source' xəbərdarlıqlarının başa düşülməsi

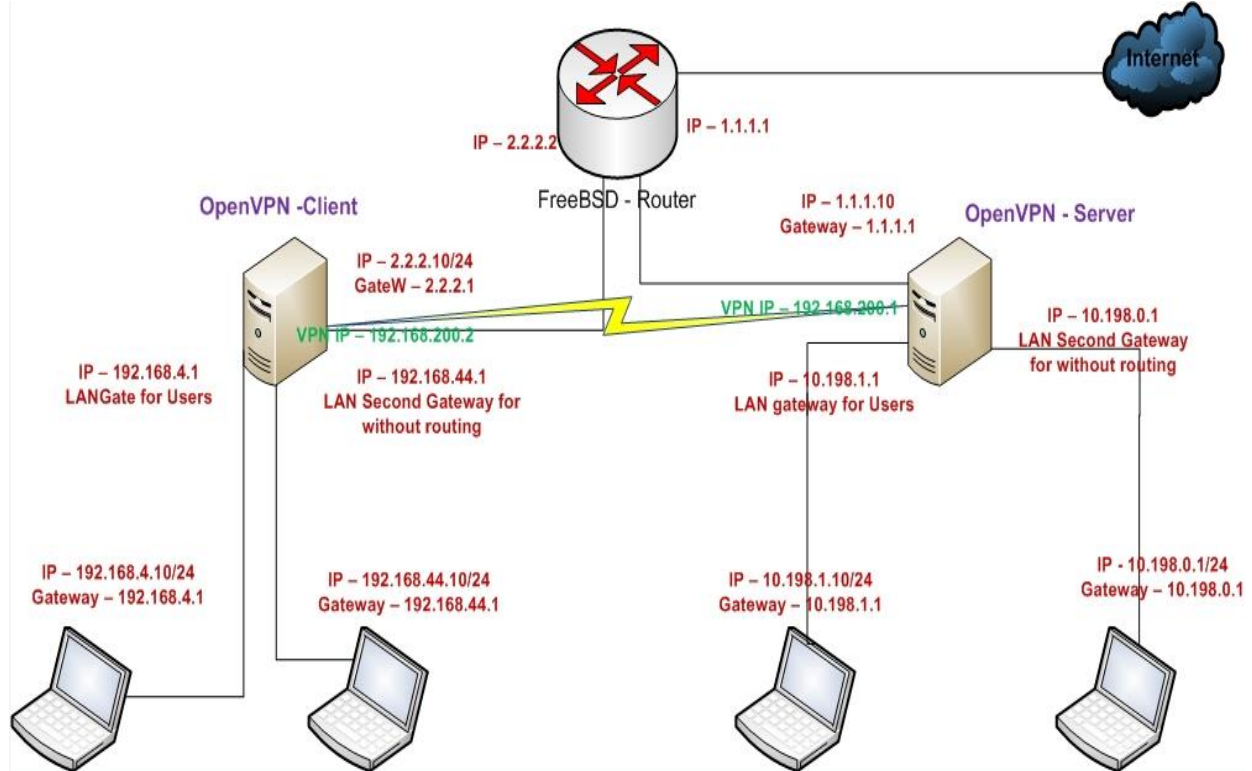
Bu misalda biz diqqətimizi VPN quraşdırmasının client-tərəf LAN-ın server-tərəf LAN-ına qoşulma hissəsinə ayıracağıq. Adi halda bu iş OpenVPN server quraşdırmasında client-config-dir direktivinə uyğun olan CCD faylının əlavə edilməsi ilə edilir. Əgər CCD fayl olmazsa və ya oxunma yetkisi olmazsa, VPN server yenə də normal işə düşəcək ancaq, client LAN şəbəkəsində olan istifadəçilər Server LAN şəbəkəsində olan istifadəçilərə normal çata bilməyəcək və eynilə də geriye. Bu misalda əgər **verbose** rejimi dərin

qoymuşuqsa, OpenVPN server jurnal faylı bizə **MULTI: bad source** sətirini göstərəcək.

Bu misalda biz VPN-i 2-ci başlıqda olan **Routing: Hər iki tərəfdə olan subnetlərə görə** quraşdıracağıq ancaq, client üçün CCD faylı olmadan. Sonra biz **MULTI: bad source** xəbərdarlıqlarını görəcəyik və bu problemin qarşısının necə alınmasını göstərəcəyik.

İşə hazırlaşaq...

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



OpenVPN2.3 serveri iki maşında yükləyin. Əmin olun ki, maşınlar şəbəkə ilə bir-birlərini görürlər. 2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalımızda server və client maşını FreeBSD9.2 x64 OpenVPN2.3-də işləyəcək. Eynilə 2-ci başlıqda yaratdığımız server üçün **basic-udp-server.conf** quraşdırma faylından və client üçün **basic-udp-client.conf** quraşdırma faylından istifadə edəcəyik.

Necə edək...

1. Əmin olaq ki, CCD faylı oxunulan deyil (Ancaq mən bu situasiyanın yaradılması üçün **/usr/local/etc/openvpn/clients** qovluğunda **openvpnclient1**-ə aid olan faylın adını dəyişib **openvpnclient2** qoymaqla yaratdım):


```
root@siteA:/usr/local/etc/openvpn # chmod 700 /usr/local/etc/openvpn/clients/
```


2. **example2-5-server.conf** faylından **verbose** səviyyəsi **5** istifadə edərək serveri işə salın(quraşdırma faylında istifadəçi və qrup **nobody** olmasından əmin olun):

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example2-5-server.conf --verb 5
```

```
Fri Mar 21 01:30:31 2014 OpenVPN 2.3.2 amd64-portbld-freebsd9.2 [SSL  
(OpenSSL)] [LZO] [eurephia] [MH] [IPv6] built on Jan 9 2014  
Fri Mar 21 01:30:31 2014 Control Channel Authentication: using  
'/usr/local/etc/openvpn/ta.key' as a OpenVPN static key file  
Fri Mar 21 01:30:31 2014 UDPv4 link local: [undef]  
Fri Mar 21 01:30:31 2014 UDPv4 link remote: [AF_INET]1.1.1.10:1194  
Fri Mar 21 01:30:31 2014 [openvpnsrver] Peer Connection Initiated with  
[AF_INET]1.1.1.10:1194  
Fri Mar 21 01:30:33 2014 TUN/TAP device /dev/tun0 opened  
Fri Mar 21 01:30:33 2014 do_ifconfig, tt->ipv6=0, tt-  
>did_ifconfig_ipv6_setup=0  
Fri Mar 21 01:30:33 2014 /sbin/ifconfig tun0 192.168.200.2  
192.168.200.2 mtu 1500 netmask 255.255.255.0 up  
add net 192.168.200.0: gateway 192.168.200.2 fib 0  
add net 10.198.0.0: gateway 192.168.200.1 fib 0  
Fri Mar 21 01:30:34 2014 Initialization Sequence Completed
```

Ancaq client-in LAN şəbəkəsindən hansısa bir istifadəçi server LAN tərəfdə olan bir istifadəçiyə paket yolladığı halda OpenVPN server məşının jurnalında aşağıdakı sətir elemel gələcək:

```
Fri Mar 21 01:31:00 2014 us=680376 openvpnclient1/2.2.2.10:10775 MULTI:  
bad source address from client [192.168.4.10], packet dropped
```

Bu misalın həlli sadəcə **/usr/local/etc/openvpn/clients** qovluğunun **root** istifadəçiyə yetkisinin **nobody**-ə dəyişməsidir(Mənim halımda isə sadəcə **/usr/local/etc/openvpn/clients** qovluğunda **openvpnclient2** faylının adını **openvpnclient1** edib, serveri yenidən işə salmaqdır).

Bu necə işləyir...

Düzgün qaydada remote LAN-ın OpenVPN serverə qoşulması üçün server məşında iki direktivin yazılmasına ehtiyac var:

```
route remote-lan remote-mask  
client-config-dir /usr/local/etc/openvpn/clients
```

Həmçinin client-in sertifikatının adı, onun CCD faylının adı ilə eyni olmalıdır. CCD faylında isə aşağıdakı sintaksisli sətir olmalıdır:

```
iroute remote-lan remote-mask
```

Bu sətirlər olmadan OpenVPN server bilmir ki, hansı VPN client hansı uzaq şəbəkəyə qoşulmuşdur. Əgər paket client-dən gələrsə, OpenVPN bu haqda bilmir və sonra paket drop edilir. '**verb 5**' loglanma səviyyəsi və daha böyüklərində xəbərdarlıq **MULTI: bad source** kimi çap edilir.

Daha da ətraflı...

Yuxarıda saydıqlarımızdan başqa, **MULTI: bad source** mesajının digər əsas bir səbəbi də var.

'MULTI: bad source' mesajının çıxmasının digər səbəbləri.

Bəzi hallar olur ki, OpenVPN serverin jurnal faylında **MULTI: bad source** mesajı çap edilir ancaq, bu halda yenə də client-in LAN tərəfindən serverin LAN-ına heç bir trafik getmir. Bu adətən Windows maşınlarında işləyən VPN clientlərdə olur. VPN qoşulmasında əgər **file sharing** varsa, Windows bəzi hallarda VPN interfeysində olmayan fərqli mənbə IP ünvandan paketlər yollayır. Paketlər OpenVPN server tərəfindən təyin edilə bilmir və xəbərdarlıq çap edilir. Bu səbəbin həlli yolu hələki məlum deyil.

Həmçinin baxın

- 2-ci başlıqda, Routing: hər iki tərəf üçün hansı ki, client-config-dir əsasları haqqında danışır.
- 7-ci başlıqda, 'client-config-dir' səbəblərinin araşdırılması hansı ki, client-config-dir direktivinin quraşdırılma səhvlərinin dərininə gedir.

Default gateway yönləndirilməsində çıxan səhv

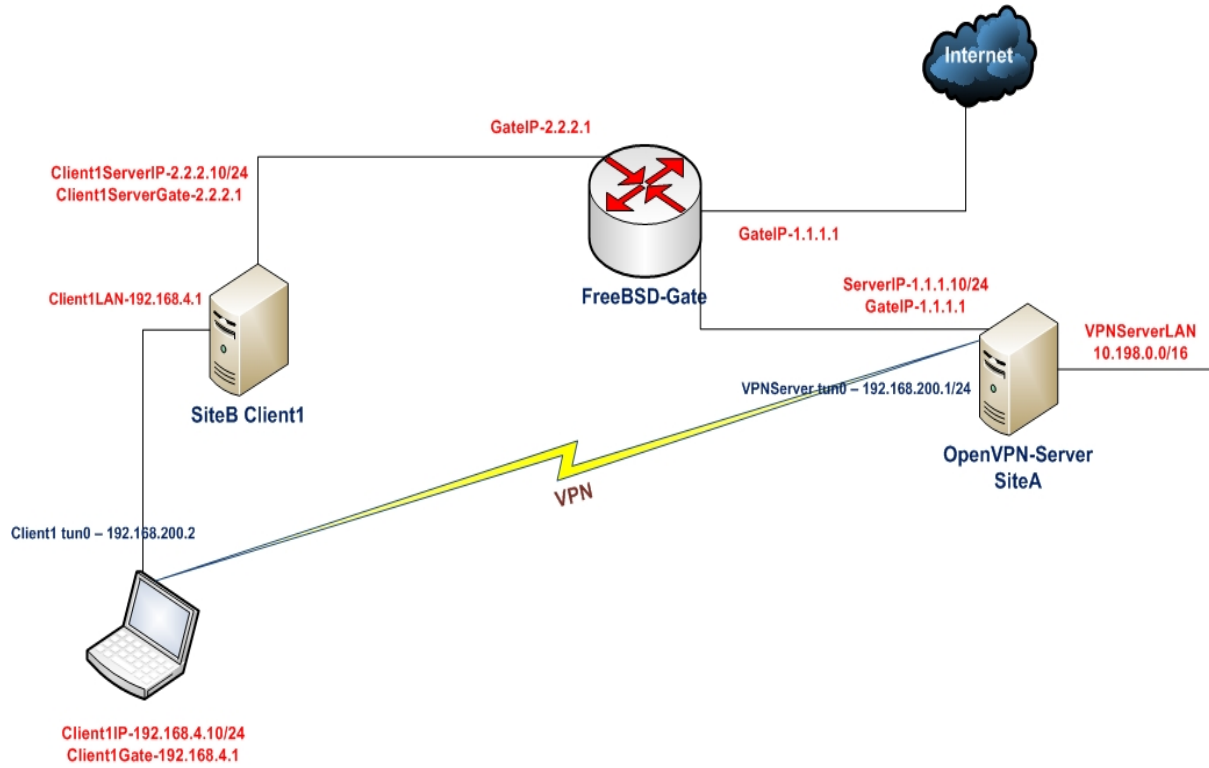
Bu misalda biz əksər hallarda çıxmayan bir səhvin qarşısını alacaq hansı ki, VPN quraşdırmasında yarana bilər. Əgər siz OpenVPN client-də default gateway-i yönləndirmək istəsəniz **redirect-gateway** direktivini istifadə edəcəksiniz hansı ki, bəzi hallarda client-in internet qoşulmalarını qırır. Bu adətən OpenVPN client-in işlədiyi maşında PPP bazalı Internet olanda olur (Misal üçün PPPoE, PPPoA, GPRS/UMTS)

Bu baş verəndə, OpenVPN bəzi hallarda yönləndirmədən öncə default gateway-i təyin edə bilmir. Ama default-gateway yönləndiriləndən sonra isə, bütün axın OpenVPN tunel üzərinə yönləndirilir. Bu halda həm şifrələnmiş həm də digər axınlar OpenVPN-ə düşdüyünə görə, nəticədə VPN düşür.

Bu misal bizə bu baş vermədən öncə problemin tapılması və həll edilməsini göstərəcək. Misalımızda GPRS/UMTS qoşulması yox, SSH üzərindən PPP istifadə edəcəyik.

İşə hazırlaşaq

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik.



OpenVPN2.3 ya da yuxarı versiyasını 2 məşında yükləyin. Əmin olun ki, məşınlar bir-birlərini şəbəkə ilə görürlər. Həmçinin client Internetə PPP istifadə edərək qoşulmalıdır ona görə ki, biz elə PPP problemini açıqlayırıq. Bu misal üçün biz SSH üzərindən PPP qoşulmasını istifadə edirik və default gateway ppp0 alətin üstünə yönləndirilmişdir.

2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda server və client məşını FreeBSD9.2 x64 və OpenVPN2.3-də olacaq. Server quraşdırması olaraq 2-ci başlıqda Server-tərəf routing üçün yaratdığımız **basic-udp-server.conf** faylından istifadə edəcəyik.

Necə edək...

1. Serveri işə salın və əlavə parametr artırın ki, default gateway-i yönləndirə biləsiniz:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-server.conf --push "redirect-gateway"
```

2. Client-in quraşdırma faylını yaradaq:

```
client  
proto udp  
# Aşağıda işə SSH istifadə edərək PPP üzərindən keçərək catılan VPN  
serverin IP-si göstərilir  
remote 1.1.1.10  
port 1194  
  
dev tun
```

```
nobind
```

```
ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnclient1.crt
key /usr/local/etc/openvpn/openvpnclient1.key
tls-auth /usr/local/etc/openvpn/ta.key 1
```

```
user nobody
verb 5
```

Faylı **example8-8-client.conf** adı ilə yadda saxlayın.

- Client-i işə salmazdan öncə system routinglərini yoxlayın:

```
root@siteB:/usr/local/etc/openvpn # netstat -rn
172.30.0.10 172.30.0.1 255.255.255.255 UGH 0 0 0 eth0
192.168.222.1 0.0.0.0 255.255.255.255 UH 0 0 0 ppp0
0.0.0.0 192.168.222.1 0.0.0.0 UG 0 0 0 ppp0
```

- Sonra client-i işə salın:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config example8-8-
client.conf
```

Qoşulma olacaq ancaq, bir neçə saniyədən sonra yenidən qırılacaq və jurnal faylında aşağıdakı mesaj yaranacaq (Mən öz halımda PPP qoşulma yaradıb test etmədim çünki, bunun üçün əlavə PPP server tələb edilirdi ancaq məntiqi quruluşun düzgün olmasına tam əmin olun. Həmçinin göstərilən jurnallar yazarın loglarıdır):

```
OpenVPN ROUTE: omitted no-op route: 192.168.222.1/255.255.255.255 ->
192.168.222.1
```

- System route-larını yenidən yoxlayın:

```
root@siteB:/usr/local/etc/openvpn # netstat -rn
172.30.0.19 172.30.0.1 255.255.255.255 UGH 0 0 0 eth0
192.168.222.1 0.0.0.0 255.255.255.255 UH 0 0 0 ppp0
192.16.186.192 0.0.0.0 255.255.255.192 U 0 0 0 eth0
192.168.200.0 0.0.0.0 255.255.248.0 U 0 0 0 tun0
10.198.0.0 192.168.200.1 255.255.0.0 UG 0 0 0 tun0
0.0.0.0 192.168.200.1 0.0.0.0 UG 0 0 0 tun0
```

Gördüyünüz kimi original default gateway silinmiş və onun yerinə VPN tunel **default gateway** olmuşdur. Client-də olan bütün qoşulmalar dayandı. Nə baş verdi, hətta OpenVPN clien-in prosesi **Ctrl+C** əmri ilə dayandırıldıqda belə, köhnə default gateway yerinə qayıtmadı.

```
TCP/UDP: Closing socket
/sbin/ip route del 10.198.0.0/16
RTNETLINK answers: Operation not permitted
ERROR: Linux route delete command failed: external program exited with
error status: 2
/sbin/ip route del 192.168.222.1/32
RTNETLINK answers: Operation not permitted
ERROR: Linux route delete command failed: external program exited with
error status: 2
/sbin/ip route del 0.0.0.0/0
```

```
RTNETLINK answers: Operation not permitted
ERROR: Linux route delete command failed: external program exited with
error status: 2
/sbin/ip route add 0.0.0.0/0 via 192.168.222.1
RTNETLINK answers: Operation not permitted
ERROR: Linux route add command failed: external program exited with
error status: 2
Closing TUN/TAP interface
```

Öncəki jurnallar client maşınında olan default gateway-in itməsini göstərir. Yeganə həll yolu bütün şəbəkə servislərinin restart-ıdır.

Bu problemin həllinin əsl yolu isə 2-ci başlıqda istifadə elədiyimiz

Redirecting default gateway-dir(Aşağıdakı sətir):

```
push "redirect-gateway def1"
```

Bu necə işləyir...

OpenVPN client inisializasiya elədikdə o həmişə çalışır ki, mövcud default gateway üzərindən OpenVPN server-ə birbaşa route yaratsın. Bəzi səbəblərdən bu alınmır. Əksər hallarda problem şəbəkə quraşdırmasında olur. Bu əksər hallarda default gateway dial-up, PPPoE qoşulması olanda olur hansı ki, ADSL, VDSL istifadə edilən yerlərdə və ya GPRS/VDSL istifadə edilən yerlərdə olur.

OpenVPN client-ə bütün trafikə VPN tunel üzərindən ötürülməsi öyrədiləndə, bu normal halda həmçinin şifrələnmiş axını birbaşa link üzərindən OpenVPN serverə ötürür. Siz düşünə bilərsiniz ki, şifrələnmiş VPN trafiki tuneldən kənarada gedir. Ancaq birbaşa routing olmayanda, bu kənara gedəcək trafik-də həmçinin tunelin içinə ötürülür və bunun nəticəsində **LOOP** yaranır hansı ki, VPN-də çökür.

Bu situasiyanı ağırlaşdıran client-in quraşdırmasında olan aşağıdakı direktivdir:

```
user nobody
```

Bu OpenVPN prosesinə deyir ki, işə düşəndən sonra bütün yetkiləri yıqışdır. Client-in qoşulması kəsildə və tunel normal işləməyəndə, client anlamır ki, necə original default gateway-i geri qaytarmaq lazımdır:

```
root@siteB:/usr/local/etc/openvpn # netstat -rn
194.171.96.27 192.16.186.254 255.255.255.255 UGH 0 0 0 eth0
192.168.222.1 0.0.0.0 255.255.255.255 UH 0 0 0 ppp0
192.16.186.192 0.0.0.0 255.255.255.192 U 0 0 0 eth0
```

Yalnız əlinizlə default gateway-in əlavə edilməsi ilə problem həll edə bilərsiniz.

Ən düzgün üsulu isə quraşdırmanızda aşağıdakı direktivin istifadəsidir:

```
push "redirect-gateway def1"
```

Öncəki sətirin sayəsində sizin default gateway silinməyəcək və əlavə onun üstünə gedən default route yazılacaq:

```
0.0.0.0 192.168.200.1 128.0.0.0 UGH 0 0 0 tun0
128.0.0.0 192.168.200.1 128.0.0.0 UGH 0 0 0 tun0
```

Gördüyünüz kimi iki ədəd marşrutun sayəsində hər şey işləyir.

Daha da ətraflı...

Bu problem əsasən OpenVPN2.0-da olan problem idi və OpenVPN2.1 və daha yüksək versiyalarda bu problem artıq həll edilmişdir. Ona görə də bütün client-lərinizi ən azı OpenVPN2.1 və daha yüksək versiyalarda istifadə etsəniz probleminizdə az olacaq

Həmçinin baxın

- 2-ci başlıqda olan default gateway-in yönləndirilməsi hansı ki, trafikə VPN tunelə yönləndirilməsini daha açıq şəkildə göstərir.

BÖLÜM 9

Performance tuning

Bu başlıqda biz aşağıdakı araşdırılma başlıqlarını açıqlayacağıq:

- **ping** istifadə edərək davamiyyətin optimallaşdırılması
- **iperf** istifadə edərək davamiyyətin optimallaşdırılması
- **OpenSSL** cipher-in sürəti
- Kompressiya sınaqları
- Axının boğulması
- **UDP** bazalı qoşulmaların təkmilləşdirilməsi
- **TCP** bazalı qoşulmaların təkmilləşdirilməsi
- **tcpdump** istifadə edərək davamiyyətin analiz edilməsi

Giriş

Bu başlıqda biz OpenVPN qurulmasında davamiyyətinin daha düzgün işləməsini təmin etmə işlərini görəcəyik. Həm client və həm də server tərəfdə gecikmənin az olması və daha sürətli işləməsi üçün fərqli parametrlərdən istifadə edəcəyik. Ancaq gecikmənin az olması üçün tələb edilən parametrlərin optimal quraşdırılması şəbəkə quruluşundan asılıdır. Bu başlıq məhz bu parametrlərin yerinə uyğun olaraq düzgün quraşdırılmasını bizə öyrədəcək.

'ping' istifadə edərək davamiyyətin optimallaşdırılması

Bu misalda biz aşağı səviyyədə olan ping əmrindən istifadə edəcəyik ki, OpenVPN quraşdırma edə bilməmiş üçün Maximum Transfer Unit-i təyin edək. MTU həcmi düzgün tapılması antenna və ya ADSL qoşulmaları istifadə elədikdə davamiyyət üçün çox önəmli olur. Çünki elə PPPoE qoşulmasının qeyri standart MTU həcmi olur. Adi şəbəkələrin MTU həcmində demək olar ki, heç bir vaxt problemi çıxmır çünki, OpenVPN-in susmaya görə olan MTU həcmi elə adi halda olan MTU-ya çox yaxın olur.

İşə hazırlaşaq

Əmin olun ki, client və server maşınlar şəbəkə ilə bir-birlərini görürlər. Bu misalda client və server maşınları FreeBSD9.2 x64 və OpenVPN2.3-də olacaq. Ancaq hər bir halda Windows maşınlar üçündə instruksiya göstərilir.

Necə edək...

1. Öncə əmin olaq ki, client maşından server maşını görə bilirik:

```
root@siteB:/usr/local/etc/openvpn # ping -c2 1.1.1.10
PING 1.1.1.10 (1.1.1.10): 56 data bytes
64 bytes from 1.1.1.10: icmp_seq=0 ttl=63 time=1.591 ms
64 bytes from 1.1.1.10: icmp_seq=1 ttl=63 time=2.150 ms
```

```
--- 1.1.1.10 ping statistics ---
```

```
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.591/1.870/2.150/0.280 ms
```

Bu serverə ICMP paketlər yollayacaq və iki cavab qayıtmalıdır. Əgər qayıtmırsa onda Firewall ICMP trafiki block edir. İşə başlamazdan öncə əmin olun ki, ping vasitəsilə serverə çatmaq olur.

2. Sonra isə client-dən serverə böyük paketlə ping yollamağa çalışın hansı ki, **Don't Fragment (DF)** biti təyin edilmişdir. Linux-da bunu **-M** parametri ilə etmək yetir.

```
[root@siteb ~]# ping 1.1.1.10 -M do -c2 -s 1600
```

```
Adi halda ping uğursuz olub aşağıdakı mesajı çap etməlidir.
From 2.2.2.10 icmp_seq=1 Frag needed and DF set (mtu = 1500)
From 2.2.2.10 icmp_seq=1 Frag needed and DF set (mtu = 1500)
```

Eyni işi FreeBSD maşında eləsək:

```
root@siteB:~ # ping -c2 -D -s 1600 1.1.1.1
```

Bu o deməkdir ki, şəbəkə kartı üzərindən keçə biləcək paketin maksimal həcmi **1500 bayt** ola bilər. Bu halda Ethernet başlıqları (header)-da nəzərə alınmalıdır (adi halda **28 bayt** olur). Yəni ki, normal **1500 baytdan 28 bayt** çıxarılmalıdır və dəqiq paket həcmi hesablanmalıdır. Bu halda **1472** alınır.

FreeBSD üçün:

```
root@siteB:~ # ping -c2 -D -s 1472 1.1.1.1
PING 1.1.1.1 (1.1.1.1): 1472 data bytes
1480 bytes from 1.1.1.1: icmp_seq=0 ttl=64 time=0.333 ms
```



```
1480 bytes from 1.1.1.1: icmp_seq=1 ttl=64 time=1.039 ms
```

Linux üçün:

```
[root@siteB ~]# ping 1.1.1.1 -M do -c2 -s 1472
PING 1.1.1.1 (1.1.1.1) 1472(1500) bytes of data.
1480 bytes from 1.1.1.1: icmp_seq=1 ttl=255 time=0.900 ms
1480 bytes from 1.1.1.1: icmp_seq=2 ttl=255 time=0.996 ms
```

3. Windows client-lər üçün isə ping sintaksisi biraz fərqlidir:

```
C:\Users\ClientC>ping -f 10.198.0.1 -l 1600
```

Əslində paketlər fragmentasiya edilməli idi ancaq, biz **-f** flagi ilə onun edilməməsini demişdik.

```
Pinging 10.198.0.1 with 1600 bytes of data:
```

```
Packet needs to be fragmented but DF set.
```

```
Packet needs to be fragmented but DF set.
```

```
Packet needs to be fragmented but DF set.
```

```
Packet needs to be fragmented but DF set.
```

Və:

```
C:\Users\ClientC>ping -f 10.198.0.1 -l 1472
```

```
Pinging 10.198.0.1 with 1472 bytes of data:
```

```
Reply from 10.198.0.1: bytes=1472 time<1ms TTL=64
```

```
Reply from 10.198.0.1: bytes=1472 time<1ms TTL=64
```

Beləliklə Ethernet şəbəkələrində və ADSL2+ şəbəkəsini çıxmaq şərti ilə xeyirli həcm **1472** bayt olur. Buna **payload** deyilir. Yəni ki, **1500** baytdan Ethernet header-ə aid olan həcm **28** bayt hər dəfə çıxıldığına görə xeyirli yükləmə (payload) **1472** bayt qalır.

OpenVPN üçün **tun-mtu** imkanının düzgün təyinatı maksimal xeyirli yüklənmə həcmi (**payload**) yeni adı halda **1472 bayt** və onun üstünə **28 byte** gəlmək ilə düzgün hesablınsa düzgündür. Ancaq bu optimal məna deyil və gələcək misallarımızda biz bunu açıqlayacağıq.

Bu necə işləyir...

ping əmri tərəfindən istifadə edilən ICMP protokolu **Don't fragment** opsiyasını istifadə edir. Bu bitin təyinatı ilə ICMP mənsəbinə çatanadək paketi hissələrə bölməmək yetkisinə malik olmur. Əgər paket mənsəbinə çatmazdan öncə Router tərəfindən hissələrə bölünməlidirsə, o drop edilir və ICMP səhvi çap ediləcək. Serverdən client-ə və client-də serverə gedən paketin ən geniş həcmnin tapılması üçün ən yaxşı üsuldur. Adətən çox kiçik olan şəbəkələrdə çox önəmli olur ki, paketlərin minimum həcmində ötürməyinə ehtiyac olur.

Beləliklə ping əmrindən istifadə edərək paketin maksimal həcmi təyin edirik ki, sonra bu həcmi OpenVPN-in davamiyyətli işləməsində istifadə edək.

Daha da ətraflı...

Bəzi hallar olur ki, firewall tərəfindən ICMP trafik block edilir onda, öncəki misalımız tam yararsız olur. Əgər sizin OpenVPN serverə yetkiniz varsa, onda tunel üzərindən də maksimal xeyirli həcmi tapa bilərsiniz.

- OpenVPN serveri əlavə olaraq aşağıdakı flaglarla işə salın:

cipher none
auth none

Eyni quraşdırmaları OpenVPN client üçün də edin. Əmin olun ki, compressiya söndürülüdür və fragment opsiyasi istifadə edilmir. Bu açıq şəkildə olan tuneli işə salacaq hansı ki, onun üzərindən ICMP paketlərin göndərilməsi ilə maksimal yararlı həcmi örgənmək olacaq.

- Remote VPN Serverin IP-sinə aşağıdakı sintaksislə ping edirik:
Linux üçün:

```
[client]$ ping 1.1.1.10 -M do -c2 -s 1472
```

FreeBSD üçün:

```
[client]$ ping -c2 -D -s 1472 1.1.1.1
```

ICMP paket çox böyük olsa, trafik yolda olan Router tərəfindən drop ediləcək. Ping uğurla qayıtmayanadək ICMP paketin həcmi kiçildərək test edin. Nəticədə uyğun olan MTU-nu əldə edəcəksiniz.

Həmçinin baxın

- UDP bazalı qoşulmaların təkmilləşdirilməsi hansı ki, UDP bazalı qoşulmaların təkmilləşdirilməsi detallarını tam açır.
- TCP bazalı qoşulmaların təkmilləşdirilməsi hansı ki, TCP bazalı qoşulmaların təkmilləşdirilməsi detallarını tam açır və həmçinin server adapterinin MTU-sunun təyin edilməsinin bəzi xırdalıqlarını açıqlayır.

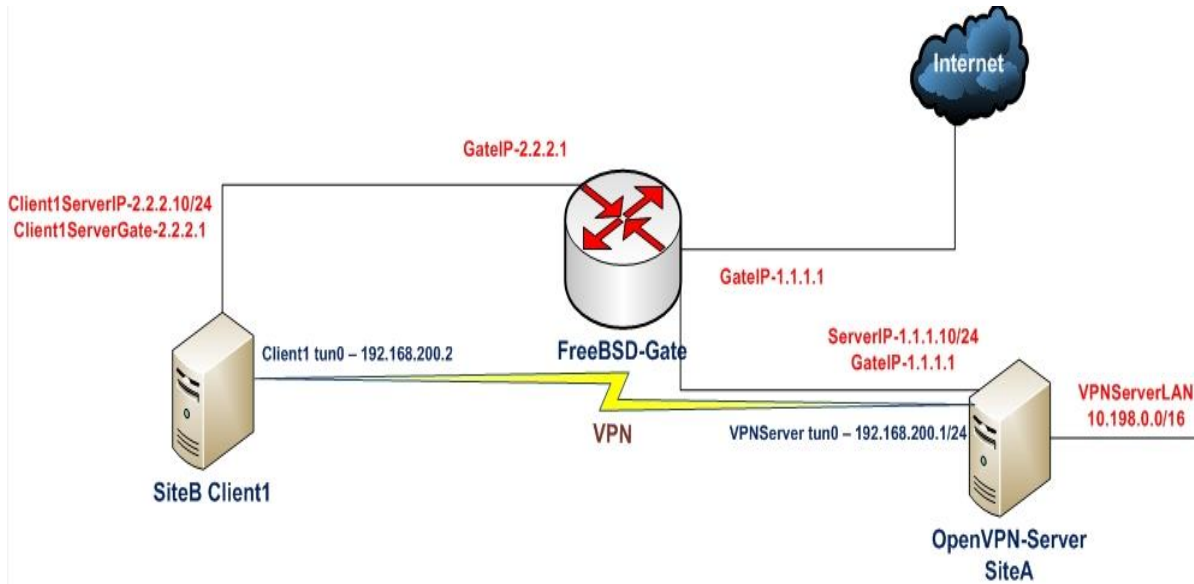
'iperf' istifadə edərək davamiyyətin optimallaşdırılması

Bu misal əsasən OpenVPN-ə aid deyil və daha çox şəbəkənin davamlılığının yoxlanılması üçün olan alət **iperf** haqqındadır. iperf utiliti <http://sourceforge.net/projects/iperf/> linkindən endirilə bilər. Linux/FreeBSD və Windows versiyaları mövcuddur.

Bu misalda biz iperf-i OpenVPN-dən kənar və VPN tunelin içində istifadə edəcəyik hansı ki, fərqi gözlərimizlə görəəcəyik.

İşə hazırlaşaq

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



OpenVPN2.3-ü iki maşında yükləyin. Əmin olun ki, maşınlar bir-birlərini şəbəkə ilə görürlər. 2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda client və server maşınları FreeBSD9.2 x64 və OpenVPN2.3-də işləyəcək. Server quraşdırma üçün 2-ci başlıqda Server-tərəf routing üçün yaratdığımız **basic-udp-server.conf** istifadə edəcəyik. Eynilə 2-ci başlıqda server-side routing misalında yaratdığımız **basic-udp-client.conf** client quraşdırma faylını client-imizdə istifadə edəcəyik.

Bunu necə edək...

1. Serveri işə salaq:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-server.conf
```
2. Sonra client-i işə salın:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config basic-udp-client.conf --daemon
```

Qoşulma aşağıdakı sətirlə uğurlu olduğunu bildirməlidir.

Sat Mar 22 14:35:44 2014 Initialization Sequence Completed

3. Sonra isə OpenVPN serverdə **iperf**-i işə salın:

```
root@siteA:/usr/local/etc/openvpn # iperf -s
```

```
-----  
Server listening on TCP port 5001  
TCP window size: 128 KByte (default)  
-----
```

4. İlk olaraq tunelin davamiyyətini yoxlayaq:

```
root@siteB:/usr/local/etc/openvpn # iperf -c 1.1.1.10
```

```
-----  
Client connecting to 192.168.200.1, TCP port 5001  
TCP window size: 129 KByte (default)  
-----
```

```
[ 3] local 192.168.200.3 port 15465 connected with 192.168.200.1 port
5001
[ ID] Interval      Transfer      Bandwidth
[ 3]  0.0-10.0 sec  82.4 MBytes  69.0 Mbits/sec
```

Gördüyümüz kimi Ethernet şəbəkəsində **10** saniyə müddətində **69 megabit** sürət transfer elədik.

Beləliklə siz IPerf sayəsində həm client və həm də server tərəfdə TCP və UDP protokolları ilə şəbəkə davamiyyətini yoxlaya bilərsiniz.

Gigabitlik şəbəkələrdə

Nəzərə alın ki, gigabitlik şəbəkələrdə TCP qoşulması iperf testində 900 megabitədək qalxa bilər. Ancaq OpenVPN şəbəkələri 320 megabit-dən yuxarı qalxa bilmir.

OpenSSL cipher-in sürəti

OpenVPN bütün şifrələnmə işləri üçün OpenSSL-dən istifadə edir. Bu o deməkdir ki, OpenVPN client və serverin sürəti ona gələn və gedən datanın şifrələnmə sürətindən asılıdır. Bu client üçün heç vaxt problem yaratmayacaq ancaq, OpenVPN server üstündə 100-lərlə client işləyən məşında həmişə problem olacaq.

Bu misalda biz OpenSSL cryptoqrafik modulunun davamiyyət sürətinin ölçülməsini və bu ölçülmə nəticəsindən necə səmərəli istifadə ediləcəyini göstərəcəyik ki, çoxlu client-lər eyni serverə qoşula bilsinlər.

İşə hazırlaşaq

Misal bir neçə məşında edilmişdir:

- FreeBSD9.2 x64 Core2 Duo olan 2.5GHZ məşın.
- Server məşını FreeBSD9.2 x64 2.8GHZ.
- Windows7 x64 1.5GHZ.

Hər bir məşında OpenVPN2.3 və OpenSSL kitabxanaları ilə yüklənmişdir.

Necə edək...

Hər bir məşında aşağıdakı əmrləri yerinə yetirin:

```
openssl speed -evp bf-cbc
openssl speed -evp aes-128-cbc
openssl speed -evp aes-256-cbc
```

İlk əmr OpenVPN-i BlowFish kriptografik cipher-i ilə edəcək. Sonrakı iki test isə **128 bit** və **256 bit**lik cipher-lərdə ediləcək hansı ki, daha çox web saytların təhlükəsizliyi üçün istifadə edilir.

Nəticə aşağıdakı cədvəldə çap ediləcək. Cədvəldə göstərilən bütün rəqəmlər, hansısa bir datanın şifrələnməsi üçün istifadə edilən 1 saniyə ərzində olan baytları göstərir. Data blokunun həcmi sütünlərdə göstərilir.

Blowfish cipher üçün aşağıdakı nəticələr çap edildi:

| type | 16 bytes | 64 bytes | 256 bytes | 1024 bytes | 8192 bytes |
|--------|------------|------------|------------|------------|------------|
| bf-cbc | 118748.92k | 127716.60k | 130120.24k | 130516.73k | 130250.37k |
| bf-cbc | 117757.83k | 127127.99k | 128675.45k | 130088.53k | 130668.32k |
| bf-cbc | 106326.68k | 124201.73k | 128904.80k | 129580.09k | 130115.00k |

AES128 cipher üçün nəticələr aşağıdakı kimi çap edildi:

| type | 16 bytes | 64 bytes | 256 bytes | 1024 bytes | 8192 bytes |
|-------------|------------|------------|------------|------------|------------|
| aes-128-cbc | 184173.55k | 203123.68k | 205958.45k | 208837.10k | 194852.87k |
| aes-128-cbc | 184420.06k | 202912.22k | 206805.27k | 209462.62k | 210086.50k |
| aes-128-cbc | 185529.24k | 202549.50k | 206816.32k | 208201.15k | 206364.90k |

AES256 bit üçün nəticələr aşağıdakı kimi çap edildi:

| type | 16 bytes | 64 bytes | 256 bytes | 1024 bytes | 8192 bytes |
|-------------|------------|------------|------------|------------|------------|
| aes-256-cbc | 147661.81k | 156870.67k | 159150.42k | 161345.67k | 161311.75k |
| aes-256-cbc | 147212.53k | 158340.29k | 159318.40k | 153164.28k | 127918.32k |
| aes-256-cbc | 145454.05k | 157834.94k | 161025.96k | 161933.70k | 158107.36k |

Bu necə işləyir...

openssl speed əmrinin çıxışı şifrələnmə və deşifrələnmə davamiyyətini göstərir. Bu nəticə avadanlıq və şifrələnmə açarından çox asılıdır. OpenVPN-in əksər paketləri **1500 bayt** olur və **1024 bayt**lıq sütün çox maraqlıdır.

Blowfish algoritmi prosessor sürətindən çox asılıdır. Blowfish daha dəqiq desək prosessorun böyük herz-də işləməsindən daha çox asılıdır. **AES128** və **AES256** bitlik şifrələnmədə əgər server istifadə edilirsə, bu əsasən görə CPU gücü çox core saylı və sürətli olmalıdır.

Daha da ətraflı...

OpenVPN davamiyyətinə görə 1 client-i üçün cipher seçimini edək. Bu hesablamaları aparmazdan öncə deyək ki, əksər client-lərdə lap yüklənmiş VPN tunellərdə də belə bu faiz nisbəti client prosessorunda maksimum 8% olub. Ancaq hesablamalar göstərdi ki, daha köhnə kompyuterlərdə əgər blowfish-dən **AES256** cipherə keçid etsəniz emal nəticəsi **760** kilobitdən **720** kilobitədək düşə bilər.

Həmçinin baxın

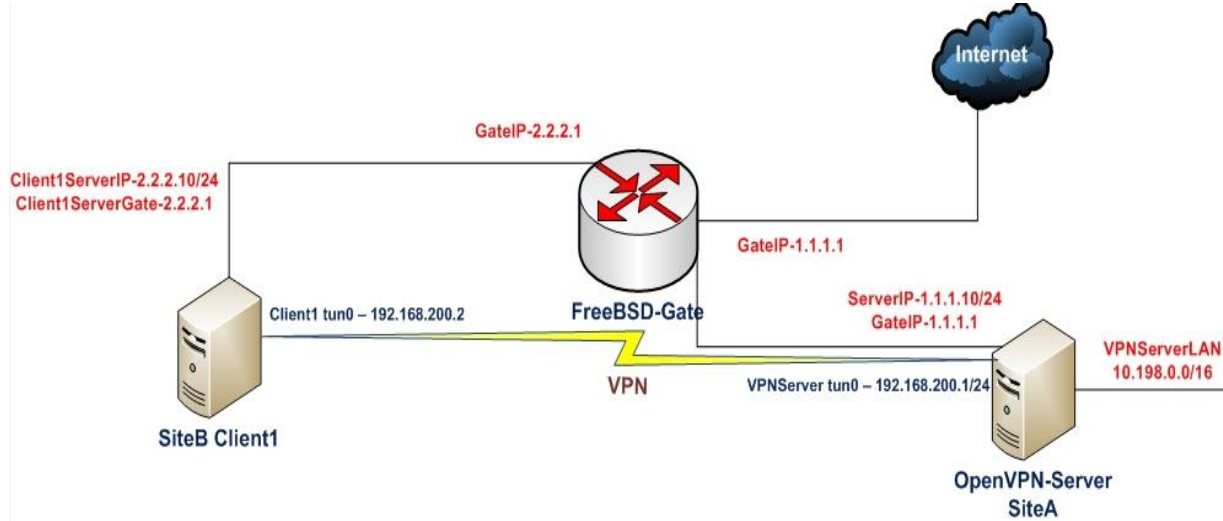
- 7-ci başlıqda, cipher uyğunsuzluğu hansı ki, client və server qoşulmalarında cipher səhvlərinin detallarını dahada ətraflı açıqlayır.

Kompressiya sınaqları

OpenVPN-in LZO kompressiya imkanı vardır. Bütün windows versiyalarında LZO versiyalar susmaya görə olur. Bu misalda biz LZO kompressiyasını istifadə edərək web səhifə və ya txt datanın clientdən serverə ötürülməsində davamiyyətə təsirini və adi şifrələnməyən data video və ya şəkilin ötürülməsində olan davamiyyəti göstərəcəyik.

İşə hazırlaşaq

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



OpenVPN2.3-ü iki maşında yükləyin. Əmin olun ki, maşınlar bir-birlərini şəbəkə ilə görürlər. 2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda client və server maşınları FreeBSD9.2 x64 və OpenVPN2.3-də işləyəcək. Server quraşdırma üçün 2-ci başlıqda Server-tərəf routing üçün yaratdığımız **basic-udp-server.conf** istifadə edəcəyik. Eynilə 2-ci başlıqda server-side routing misalında yaratdığımız **basic-udp-client.conf** client quraşdırma faylını client-imizdə istifadə edəcəyik. Ancaq client olaraq həmçinin Windows7 x64-dən istifadə ediləcək və quraşdırma faylı 2-ci başlıqda yaratdığımız **ifconfig-pool** misalındaki **basic-udp-client.ovpn** faylı olacaq.

Bunu necə edək...

1. **basic-udp-server.conf** faylını **example9-4-server.conf** adlı fayla nüsxələyin və **example9-4-server.conf** faylının sonuna aşağıdakı sətiri əlavə edin:
comp-lzo
2. Serveri işə salın:
root@siteA:/usr/local/etc/openvpn # **openvpn --config example9-4-server.conf**
3. Uyğun olaraq FreeBSD client maşında **basic-udp-client.conf** faylını **example9-4-client.conf** faylına nüsxələyin və **example9-4-client.conf** faylının sonuna aşağıdakı sətiri əlavə edin:
comp-lzo
4. Client-i işə salın:
root@siteB:/usr/local/etc/openvpn # **openvpn --config example9-4-client.conf --daemon**
5. Sonra serverdə iperf-i işə salın:

```
root@siteA:/usr/local/etc/openvpn # iperf -s
```

6. İlk olaraq tuneldən kənarında ötürülən datanın davamiyyətinə baxaq:

```
root@siteB:/usr/local/etc/openvpn # iperf -c 192.168.200.1
```

```
-----  
Client connecting to 192.168.200.1, TCP port 5001
```

```
TCP window size: 128 KByte (default)  
-----
```

```
[ 3] local 192.168.200.2 port 19020 connected with 192.168.200.1 port 5001  
[ ID] Interval      Transfer      Bandwidth  
[ 3] 0.0-10.0 sec  85.2 MBytes  71.4 Mbits/sec
```

Bu nəticə Ethernet şəbəkəsində **10** saniyədə **85.2** megabayt və **71** megabit oldu.

7. Sonra isə sıxılmamış data yaradaq və onu yollayaq:

```
root@siteB:/usr/local/etc/openvpn # dd if=/dev/urandom bs=1024k  
count=60 of=random
```

```
root@siteB:/usr/local/etc/openvpn # iperf -c 192.168.200.1 -F random
```

```
-----  
Client connecting to 192.168.200.1, TCP port 5001
```

```
TCP window size: 128 KByte (default)  
-----
```

```
[ 4] local 192.168.200.2 port 65395 connected with 192.168.200.1 port 5001  
[ ID] Interval      Transfer      Bandwidth  
[ 4] 0.0- 9.1 sec  60.1 MBytes  55.5 Mbits/sec
```

Öncə **60MB** həcmində **random** data fayl yaratdıq sonra isə həmin faylı şəbəkə ilə **iperf** vasitəsilə digər maşına transfer elədik. **9** saniyədə **55** megabit

8. Və sonda sıxılmış data(Sıfırlarla doldurulmuş fayl):

```
root@siteB:/usr/local/etc/openvpn # dd if=/dev/zero bs=1024k count=60  
of=zeros
```

```
root@siteB:/usr/local/etc/openvpn # iperf -c 192.168.200.1 -F zeros
```

```
-----  
Client connecting to 192.168.200.1, TCP port 5001
```

```
TCP window size: 128 KByte (default)  
-----
```

```
[ 4] local 192.168.200.2 port 35549 connected with 192.168.200.1 port 5001  
[ ID] Interval      Transfer      Bandwidth  
[ 4] 0.0- 7.2 sec  60.1 MBytes  69.9 Mbits/sec
```

Gördüyünüz kimi **7** saniyədə **60** megabayt və **70** megabit sürətində. Davamiyyət fərqlidir.

9. Eyni fərqi Windows7 maşında yoxlamaq üçün isə **basic-udp-client.ovpn** faylını **example9-4.ovpn** faylına nüsxələyin və **example9-4.ovpn** faylının sonuna aşağıdakı sətiri əlavə edin:

```
comp-lzo
```

10. Sonra client-i işə salın.

iperf sınaqlarının nəticəsi gördüyümüz kimi fərqli oldu. Ancaq bizim test elədiyimiz CPU-lar güclü olduğuna görə nəticədə olan fərq o qədər böyük olmadı.

Bu necə işləyir...

Kompresiya aktiv olanda tunel üzərindən keçən paketlər öncə sıxılır və sonra şifrələnib digər tərəfə ötürülür. Kompresiya LZO kitabxanası ilə edilir hansı ki, öncədən OpenVPN-ə inteqrasiya edilmişdir.

Daha da ətraflı...

Kompresiyanın istifadə edilməsində bəzi önəmli hissələr vardır ki, siz nəzərə almalısınız.

Kompresiya opsiyalarının ötürülməsi

OpenVPN vasitəsilə mümkündür ki, aşağıdakı direktivdən istifadə edərək serverdən client-ə kompresiya opsiyalarını ötürə bilərsiniz:

```
comp-lzo  
push "comp-lzo"
```

push direktivi OpenVPN2.1-dən başlayaraq əgər onda aşağıdakı sətir mövcuddursa, o halda client-də işləyəcək:

```
comp-lzo {yes|no|adaptive}
```

Əgər client tərəfdə aşağıdakı direktiv mövcud olmazsa qoşulma uğurlu olmayacaq.

Adaptive kompresiya

Əgər aşağıdakı directive istifadə edilirsə, onda adaptive kompresiya susmaya görə işləyir:

```
comp-lzo
```

Əgər siz OpenVPN üzərindən keçən bütün dataların hamısının tamamilə sıxılmasını istəyirsinizsə, onda həm client və həm də serverdə aşağıdakı direktivi yazmanız yetər:

```
comp-lzo yes
```

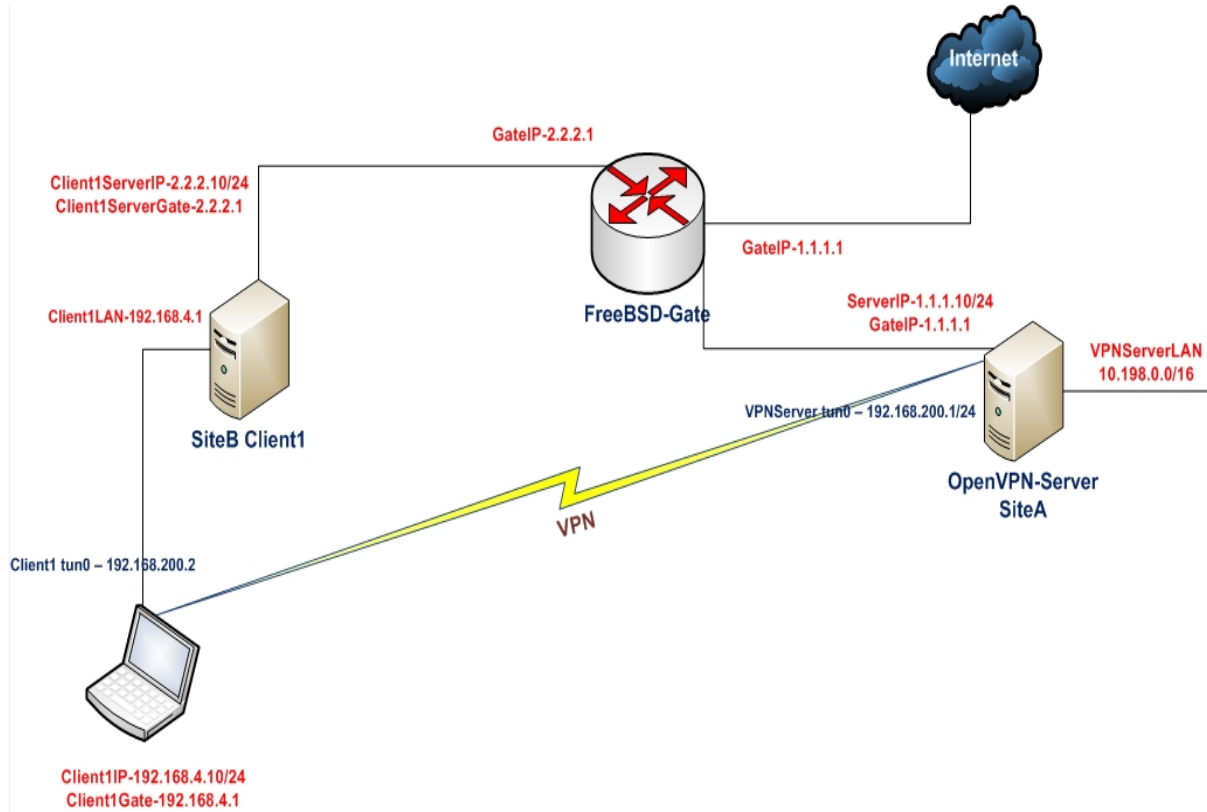
Ötürülən datanın tipindən asılı olaraq davamiyyət fərqli olacaq.

Axının boğulması

Bu misedə biz OpenVPN client-in upload sürətini sıxacayıq. Bu server və ya client-in şəbəkə sürətini boğmaq üçün istifadə edilir. Ancaq OpenVPN axının boğulmasında clientlərin endirim sürətini kiçildə bilmir o yalnız upload sürətini boğa bilir. Endirim sürətini öz firewall-ınızla edə bilərsiniz.

İşə hazırlaşaq

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



OpenVPN2.3-ü iki maşında yükləyin. Əmin olun ki, maşınlar bir-birlərini şəbəkə ilə görürlər. 2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda server maşını FreeBSD9.2 x64 və OpenVPN2.3-də işləyəcək. Server quraşdırma üçün 2-ci başlıqda Server-tərəf routing üçün yaratdığımız **basic-udp-server.conf** istifadə edəcəyik. Client olaraq həmçinin Windows7 x64 OpenVPN2.3-dən istifadə ediləcək və quraşdırma faylı 2-ci başlıqda yaratdığımız **ifconfig-pool** misalındaki **basic-udp-client.ovpn** olacaq.

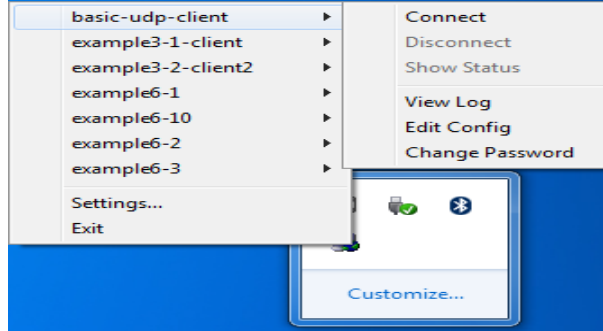
Necə edəcəyik.

1. Server maşında **basic-udp-server.conf** faylını **example9-5-server.conf** nüsxələyin və **example9-5-server.conf** faylının sonuna aşağıdakı sətiri əlavə edin:

```
push "shaper 100000"
```

Bu VPN client-lər üçün upload sürətini saniyədə **100000** bayt edir (**100kbps**).

2. Serveri işə salın:
root@siteA:/usr/local/etc/openvpn # **openvpn --config example9-5-server.conf**
3. Client-i işə salın:



4. Sonra iperf-i serverdə işə salaq:
root@siteA:/usr/local/etc/openvpn # **iperf -s**
5. Sonra iperf-i Windows maşında işə salıb sürətə baxıb görək ki, **100KB/s-**
də yuxarı çıxır:

```

C:\>cd iperf
C:\iperf>iperf.exe -c 192.168.200.1 -fK
-----
Client connecting to 192.168.200.1, TCP port 5001
TCP window size: 64.0 KByte (default)
-----
[ 3] local 192.168.200.3 port 49159 connected with 192.168.200.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0-12.5 sec  1024 KBytes  82.2 KBytes/sec
C:\iperf>

```

Bu necə işləyir...

Client serverə qoşulan kimi, server ona opsiya ötürür ki, sənin VPN tunel ilə çıxış trafikini **100KB/s** olacaq. Və bundan sonra client tunel ilə data ötürmək istəyəndə özü həmin şəbəkə sürətini boğur sonra ötürür. Endirim sürətini belə boğmaq olmur, həmçinin unutmayın ki, aşağıdakı direktiv heç bir vaxt OpenVPN serverin özündə istifadə edilə bilməz.

shaper 100000

Endirim sürətini boğmaq üçün isə UNIX/Linux-un öz imkanlarından istifadə edə bilərsiniz.

Daha da ətraflı...

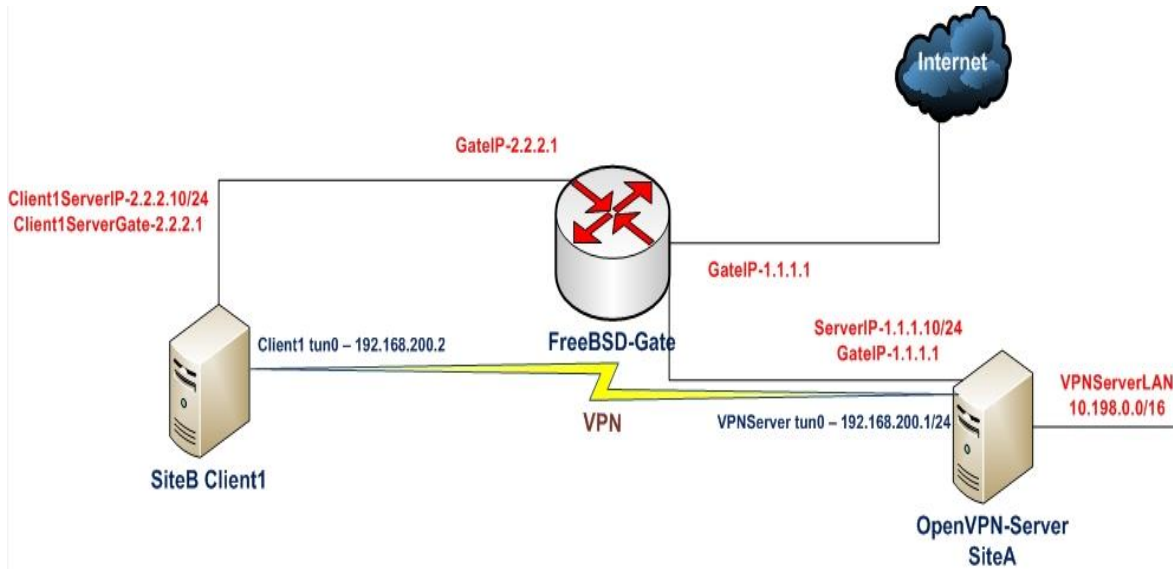
Nəzərə alın ki, UNIX/LINUX bazalı clientlərdə OpenVPN traffic shaper normal işləmir və bug olaraq hələ də qeydiyyatdadır.

UDP bazalı qoşulmaların təkmilləşdirilməsi

Bu başlıqda biz UDP bazalı VPN tunellərin təkmilləşdirilməsi haqqında danışacağıq ancaq istifadə etdiyimiz imkanlarda ehtiyatla istifadə etmək lazımdır. Çünki hər elədiyiniz dəyişiklik şəbəkədə tamamilə dəyişiklik edəcək.

İşə hazırlaşaq

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



OpenVPN2.3-ü iki maşında yükləyin. Əmin olun ki, maşınlar bir-birlərini şəbəkə ilə görürlər. 2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda client və server maşınları FreeBSD9.2 x64 və OpenVPN2.3-də işləyəcək. Server quraşdırma üçün 2-ci başlıqda Server-tərəf routing üçün yaratdığımız **basic-udp-server.conf** istifadə edəcəyik. Eynilə 2-ci başlıqda server-side routing misalında yaratdığımız **basic-udp-client.conf** client quraşdırma faylını client-imizdə istifadə edəcəyik.

Necə edək...

1. **basic-udp-server.conf** faylını **example9-6-server.conf** faylına nüsxələyin və **example9-6-server.conf** faylının sonuna aşağıdakı sətiri əlavə edin:
fragment 1400

2. Serveri işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example9-6-server.conf
```

3. Uyğun olaraq **basic-udp-client.conf** faylını **example9-6-client.conf** faylına nüsxələyin və **example9-6-client.conf** faylının içine aşağıdakı sətiri əlavə edin:

fragment 1400

4. Client-i işə salın:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config example9-6-client.conf --daemon
```

5. Sonra serverdə iperf-i işə salaq:

```
root@siteA:/usr/local/etc/openvpn # iperf -s
```

6. İlk olaraq tunelin kənarında davamlılığını yoxlayaq:

```
root@siteB:/usr/local/etc/openvpn # iperf -c 1.1.1.10
```

```
-----  
Client connecting to 1.1.1.10, TCP port 5001
```

```
TCP window size: 129 KByte (default)
```

```
-----
[ 3] local 2.2.2.10 port 22022 connected with 1.1.1.10 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0-10.0 sec   150 MBytes   125 Mbits/sec
```

7. Sonra tunelin içində olan davamlılığını hesablayaq:

```
root@siteB:/usr/local/etc/openvpn # iperf -c 192.168.200.1
```

```
-----
Client connecting to 192.168.200.1, TCP port 5001
TCP window size: 128 KByte (default)
```

```
-----
[ 3] local 192.168.200.2 port 34587 connected with 192.168.200.1 port
5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0-10.0 sec   61.0 MBytes   51.0 Mbits/sec
```

Gördüyümüz kimi tuneldən kənarında nə qədər həcmdə və tunelin içində nə qədər həcmdə fragmentasiya fərqi oldu.

8. Fərqli fragment mənası mənimsədilib yoxlamaq üçün client-dən serverə ping yollayın:

```
root@siteB:/usr/local/etc/openvpn # ping -c10 192.168.200.1
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.064/2.438/4.176/0.723 ms
```

Nəticə aşağıdakı cədvəldə göstərilir:

| Fragmentation size | Ping result |
|---------------------------|--------------------|
| Default (1500) | 41 +/- 1 ms |
| 1400 | 43 +/- 1 ms |
| 400 | 47 +/- 1 ms |

Gördüyünüz kimi, hal-hazırkı şəbəkə quruluşunda server quraşdırmasına fragment opsiyasının əlavə edilməsi davamiyyətin gücünə heç bir təsir göstərmir. Ona görə də, bu opsiyanı əlavə etdikdə şəbəkə quruluşunu nəzərə almalısınız.

Bu necə işləyir...

OpenVPN quraşdırma direktivi:

```
fragment 1400
```

Bu o deməkdir ki, şifrələnmiş kanalla gələn və həcmi **1400** baytdan böyük olan bütün paketlər fragmentləşdiriləcək. Əgər şəbəkəyinizdə gecikmələr çox olursa, bu elədə effekt verməyəcək. Fragment həcmi kiçiltməklə paketlər daha da kiçik həcmlərə bölünür. Bu gedib-qayıdan paketlərin böyüdülmə vaxtına gətirib çıxarır. Əgər şəbəkədə gecikmə böyükdürsə, bu daha böyük gecikmələrə gətirib çıxaracaq. Ona görə də **fragment** və **mssfix** opsiyalarını ehtiyatla istifadə etmək lazımdır.

Daha da ətraflı...

fragment direktivi əksər hallarda **mssfix** direktivi ilə birlikdə istifadə edilir:

mssfix [maximum-segment-size]

Bu direktiv tunel üzərindən keçən TCP sessiyalarına məlumat ötürür ki, onlar OpenVPN tərəfindən enkapsulyasiya edildikdən sonra ötürükləri paketin həcmi limitləməlidirlər. Qəbul edilən UDP paket həcmi hansı ki, OpenVPN öz clientlərinə ötürəcək və o maximum segment həcmi aşmalı deyil. Bu həmçinin OpenVPN tərəfindən çıxış paketlərinin ötürülməsində istifadə edilir. Əgər maximum segment size təyin edilməyibsə, onda fragment direktivindəki məna istifadə ediləcək.

İdeal olaraq həmişə **mssfix** və **fragment** direktivləri birgə istifadə edilir hansı ki, **mssfix** çalışır ki, ilk addımda götürdüyü paketin fragmentində TCP-ni saxlasın. İlk növbədə əgər böyük paketlər gələrsə, **fragment** direktivi onları fragmentasiya edəcək.

Həmçinin baxın

- Növbəti misalımızda TCP bazalı qoşulmaların təkmilləşdirilməsinə baxacağıq

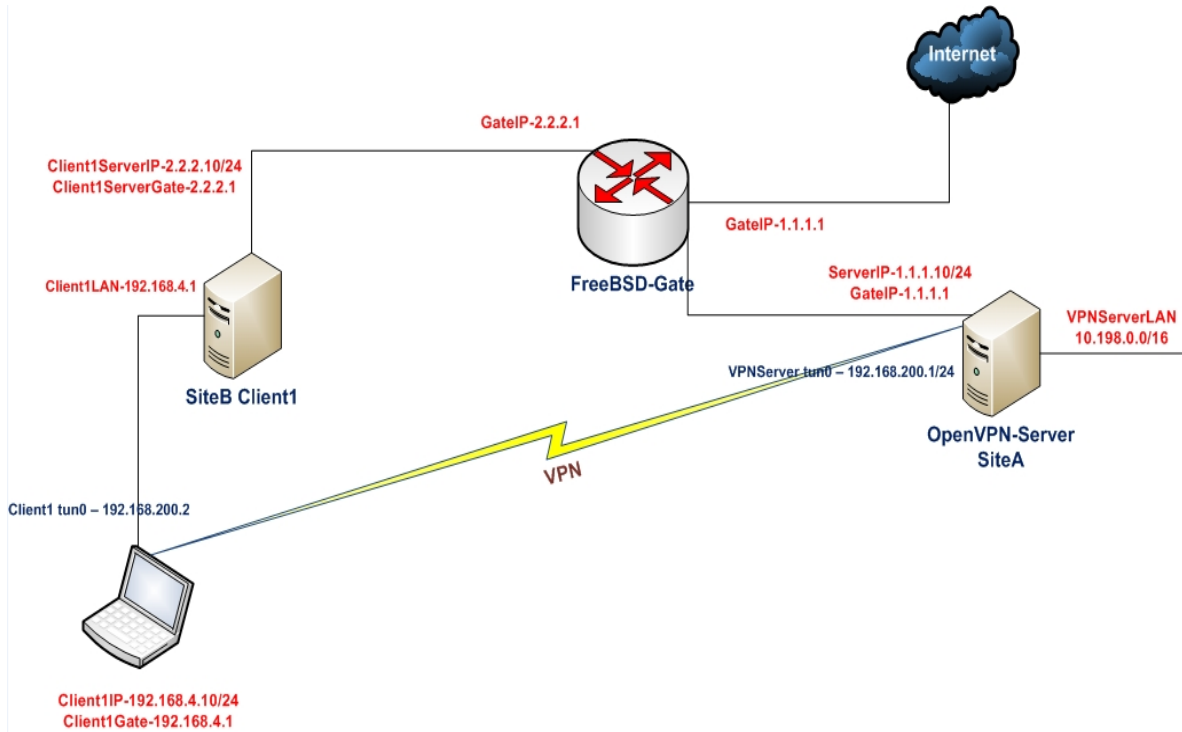
TCP bazalı qoşulmaların təkmilləşdirilməsi

Bu misalımızda biz OpenVPN tunellərdə TCP bazalı qoşulmaların təkmilləşdirilməsinə baxacağıq. TCP bazalı qoşulmalarında VPN son nöqtələri arasında istifadə edilən protocol TCP olur. Bunun üstünlükləri və çatışmamazlıqları var. Əsas üstünlüyü ondan ibarətdir ki, əksər hallarda TCP qoşulmanı quraşdırmaq UDP-dən asandır. Ancaq TCP qoşulmalarında şəbəkə sürətiniz kiçikdirsə, çox böyük gecikmələrə gətirib çıxara bilər. Buna TCP üzərindən TCP sindromu deyilir. TCP protokol təminat verir ki, əgər göndərilən paketin hansısa bir hissəsi itərsə onu yenidən yollayacaq. Elə ona görə də, TCP üzərindən TCP tunelin edilməsinin pis cəhəti odur ki, hər iki səviyyədə TCP protocol təminat verir. Bu şəbəkəni boş yerə yükləməyə başlayır.

Əgər düzgün təkmilləşdirilmə edilibsə, onda OpenVPN tuneli UDP-də edə bildiyimiz kimi, elə TCP-də də davamlı edə bilərik. Bu misalda biz elə TCP bazalı OpenVPN qoşulmalarının təkmilləşdirilməsi üsullarını araşdıracağıq.

İşə hazırlaşaq

Biz aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



OpenVPN2.3-ü iki maşında yükləyin. Əmin olun ki, maşınlar bir-birlərini şəbəkə ilə görürlər. 2-ci başlıqda yaratdığımız client və server sertifikatlarını burdada istifadə edəcəyik. Bu misalda server maşını FreeBSD9.2 x64 və OpenVPN2.3-də işləyəcək. Client isə Windows7 x64 OpenVPN2.3-də işləyəcək.

Necə edək...

1. **example9-7-server.conf** adında server quraşdırma faylı yaradıb içine aşağıdakı sətirləri əlavə edək:

```

proto tcp
port 1194
dev tun
server 192.168.200.0 255.255.255.0

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnsrver.crt
key /usr/local/etc/openvpn/openvpnsrver.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0

persist-key
persist-tun
keepalive 10 60

topology subnet

user nobody
group nobody

```

```
daemon
log-append /var/log/openvpn.log
```

```
tcp-nodelay
```

2. Serveri işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example9-7-server.conf
```

3. Sonra işə client maşında **C:\Program Files\OpenVPN\config** ünvanında **example9-7.ovpn** adlı quraşdırma faylını yaradıb içinə aşağıdakı sətirləri əlavə edək(**c:\windows\system32\drivers\etc\hosts** faylında **1.1.1.10 openvpnsrver.example.com** sətiri mövcuddur):

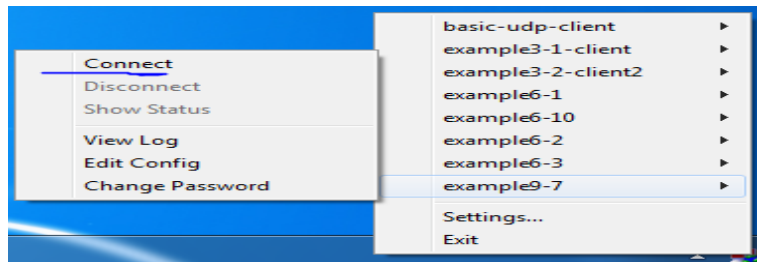
```
client
proto tcp
remote openvpnsrver.example.com
port 1194
```

```
dev tun
nobind
```

```
ca "c:/program files/openvpn/config/ca.crt"
cert "c:/program files/openvpn/config/openvpnclient2.crt"
key "c:/program files/openvpn/config/openvpnclient2.key"
tls-auth "c:/program files/openvpn/config/ta.key" 1
```

```
ns-cert-type server
```

4. Sonra client-i işə salın:



5. Sonra serverdə iperf-i işə salın:

```
root@siteA:/usr/local/etc/openvpn # iperf -s
```

6. Tunelimizin davamiyyətini ölçək:

```
C:\iperf>iperf -c 192.168.200.1
```

```
-----
Client connecting to 192.168.200.1, TCP port 5001
TCP window size: 64.0 KByte (default)
-----
```

```
[ 3] local 192.168.200.2 port 49160 connected with 192.168.200.1 port 5001
```

```
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0-10.1 sec  34.6 MBytes  28.9 Mbits/sec
```

Bu şəbəkəmizdə biz aşağıdakı cədvələ uyğun olaraq testlərimizi apardıq:

| Protocol | Nəticə |
|-------------------------------------|----------------|
| UDP | 42.7 Mbits/sec |
| TCP | 9.88 Mbits/sec |
| TCP və <code>tcp-nodelay</code> ilə | 28.9 Mbits/sec |

Gördüyünüz kimi, TCP ilə `--tcp-nodelay` opsiyası istifadə edilən halda, UDP ilə istifadə edilən nəticəyə çox yaxın olur.

Bu necə işləyir...

OpenVPN TCP protocol istifadə elədikdə, bütün paketlər TCP qoşulması üzərindən gedir. Susmaya görə, TCP qoşulmaları Nagle alqoritmi istifadə edir hansı ki, bütün kiçik paketlər göndərilməzdən öncə bir yere yığılır. OpenVPN üçün bu əksər hallarda çox pis gecikmələrə gətirib çıxarır və buna görə də Nagle alqoritmini `--tcp-nodelay` direktivi ilə söndürürük. Sonra da fərqi sınaqlarımızda görürük.

Daha da ətraflı...

TCP bazalı qoşulmalarda 2 vacib parametr dəyişdirilə bilər:

- `--tcp-nodelay` directive
- **TUN/TAP-Win32** adapter-in MTU həcmi hansı ki, `--tun-mtu` ya da `--link-mtu` direktivləri ilə təyin edilir.

UNIX/Linux maşınlarında TUN/TAP adapterlərin MTU həcmi dəyişdirilməsi çox asandır ancaq, Windows maşınlarında bu nisbətən çətinidir. OpenVPN serverdə öncədən planlı şəkildə MTU təyin edilməlidir ki, sonra client-lərdə həmin MTU həcmindən istifadə eləsin.

Windows7 maşında MTU-nu `netsh` əmri ilə dəyişək:

- Düzgün interfeys və onların MTU həcmələrinə baxaq:

```
C:\iperf>netsh interface ipv4 show subinterfaces
```

| MTU | MediaSenseState | Bytes In | Bytes Out | Interface |
|------------|-----------------|----------|-----------|------------------------------|
| 4294967295 | 1 | 0 | 21757 | Loopback Pseudo-Interface 1 |
| 1500 | 1 | 9630966 | 113043930 | Local Area Connection |
| 1500 | 5 | 0 | 0 | Bluetooth Network Connection |
| 1500 | 1 | 3044472 | 105914341 | Local Area Connection 2 |

- Sonra sub-interfeys üçün MTU həcmi dəyişək:

```
C:\iperf>netsh interface ipv4 set subinterface "1" mtu=1400
```

Ok.

Qeyd edin ki, öncəki əmrlər admin yetkisi ilə işə düşə bilər.

Əgər Windows TAP-Win32 adapter-in MTU həcmi OpenVPN-də quraşdırılan MTU həcmindən böyük olsa, aşağıdakı sətirlər OpenVPN2.3 jurnal faylında çap ediləcək:

```
read from TUN/TAP [State=AT?c Err=[c:\src\21\tap-win32\tapdrv.c/2447] #O=4 Tx=[29510,0] Rx=[15309,0] IrpQ=[0,1,16] PktQ=[0,22,64]
```

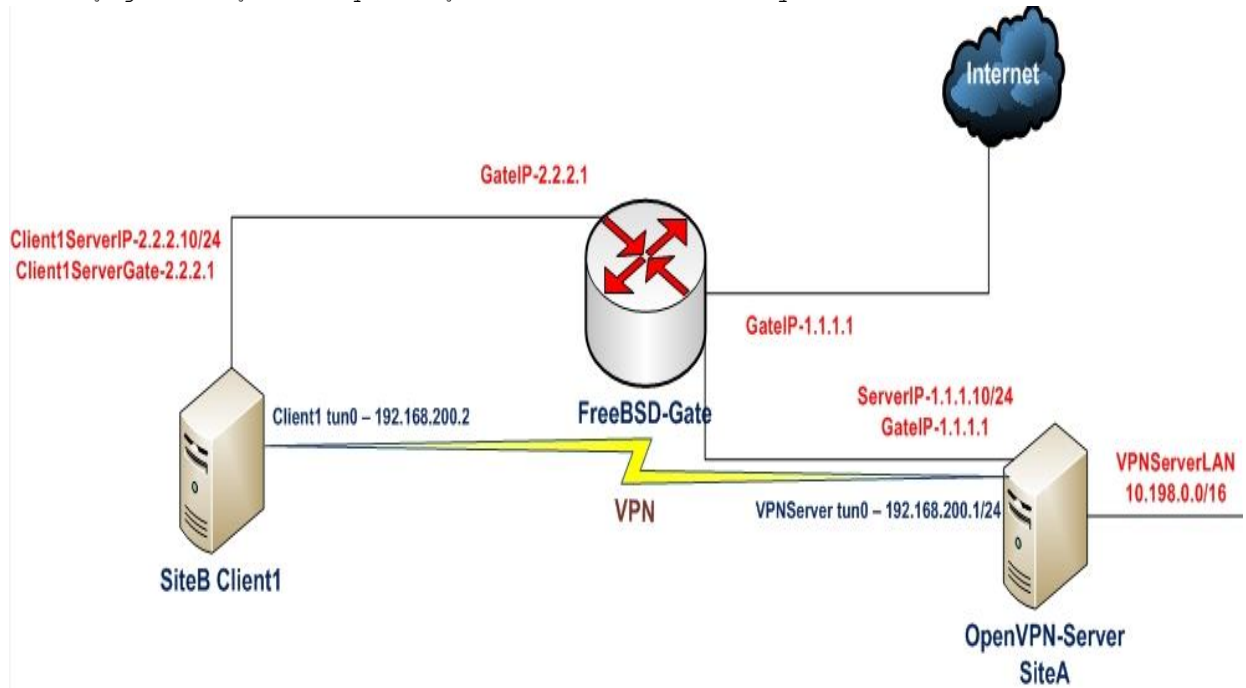

InjQ=[0,1,16]]: More data is available. (code=234)

tcpdump istifadə edərək davamiyyətin analiz edilməsi

Bu misalda biz OpenVPN quruluşumuzda davamiyyətin yoxlanılması üçün **tcpdump** utilitindən istifadə edəcəyik. Həmçinin mümkündür ki, Linux, Windows, MacOS üçün olan OpenSource Wireshark utilitindən istifadə edəsiniz.

İşə başlayaq

Biz aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



OpenVPN2.3-ü iki maşında yükləyin. Əmin olun ki, maşınlar bir-birlərini şəbəkə ilə görürlər. 2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda server və client maşınları FreeBSD9.2 x64 və OpenVPN2.3-də işləyəcək. Server üçün UDP bazalı qoşulmaların təkmilləşdirilməsi misalında olan **example9-6-server.conf** faylından istifadə ediləcək. Eynilə client üçün eyni misalda olan **example9-6-client.conf** faylını client üçün istifadə edəcəyik.

Necə edək...

1. Serveri işə salaq:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example9-6-server.conf
```
2. Sonra client-i işə salaq:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config example9-6-client.conf --daemon
```

3. Server tərəfdə tcpdump ilə tunel interfeysin özünə yox ancaq public şəbəkə kartımızda daxil olan paketlərə baxaq:

```
root@siteA:/usr/local/etc/openvpn # tcpdump -nnl -i em0 udp port 1194
```

Öncəki əmrlə **tcpdump** deyir ki, **em0** şəbəkə kartında **UDP** trafikə **1194**-cü port üçün qulaq as. Bu port-da OpenVPN qulaq asır.

4. OpenVPN client-dən serverə 2 fərqli MTU həcm ilə ping paketləri yollayın:

```
root@siteB:/usr/local/etc/openvpn # ping -c2 -s 1300 192.168.200.1
```

Serverdə olan tcpdump-in **1300** bayt üçün cavabı:

```
16:00:00.407687 IP 2.2.2.10.55056 > 1.1.1.10.1194: UDP, length 1373
16:00:00.408808 IP 1.1.1.10.1194 > 2.2.2.10.55056: UDP, length 1373
16:00:01.428980 IP 2.2.2.10.55056 > 1.1.1.10.1194: UDP, length 1373
16:00:01.430175 IP 1.1.1.10.1194 > 2.2.2.10.55056: UDP, length 1373
```

```
root@siteB:/usr/local/etc/openvpn # ping -c2 -s 1400 192.168.200.1
```

Server-də olan tcpdumpin 1400 bayt üçün cavabı:

```
16:00:38.734587 IP 2.2.2.10.55056 > 1.1.1.10.1194: UDP, length 757
16:00:38.734640 IP 2.2.2.10.55056 > 1.1.1.10.1194: UDP, length 757
16:00:38.735121 IP 1.1.1.10.1194 > 2.2.2.10.55056: UDP, length 757
16:00:38.735206 IP 1.1.1.10.1194 > 2.2.2.10.55056: UDP, length 757
16:00:39.747225 IP 2.2.2.10.55056 > 1.1.1.10.1194: UDP, length 757
16:00:39.747308 IP 2.2.2.10.55056 > 1.1.1.10.1194: UDP, length 757
16:00:39.748315 IP 1.1.1.10.1194 > 2.2.2.10.55056: UDP, length 757
16:00:39.748517 IP 1.1.1.10.1194 > 2.2.2.10.55056: UDP, length 757
```

İlk ICMP paketlər fragmentasiya edilmədən yollandı ona görə ki, onlar **1400** bayt-dan kiçikdir. İkinci ICMP paketləri isə şifrələnmiş paketləri 1400 bayt-dan çox olduğu üçün onları iki hissəyə bölüb ötürdü.

Bu necə işləyir...

OpenVPN quraşdırma direktivi:

```
fragment 1400
```

Bunun səbəbi odur ki, bütün şifrələnmiş paketlər **1400** baytdan yuxarı olduğuna görə fragmentlərə bölünməlidir. Şifrələnmiş axına baxmaq istədikdə isə OpenVPN serverə ping edərək əldə edə bilərik.

Qeyd: Fragmentləşməyə ehtiyacı olan bütün paketlər bərabər olaraq fragmentlənir.

Şifrələnmiş paketlər **1400** baytdan çox olur çünki, təhlükəsiz tunel paketin önünə əlavə başlıqlar artırır. Məhz buna görə VPN tunellərdə şəbəkənin şifrələnmədən ötürülməsi yüksək davamiyyətli olur.

Həmçinin baxın

- 9-cu başlıqda UDP bazalı qoşulmaların təkmilləşdirilməsi hansı ki, bu başlıqda **fragment** direktivi açıqlanır.

BÖLÜM 10

OS inteqrasiyası

Bu başlıqda biz aşağıdakıları açıqlayacağıq:

- Linux: NetworkMaganer-in istifadə edilməsi
- Linux: pull-resolv-conf istifadə edilməsi
- Mac OS: Tunnelblick istifadə edilməsi
- Windows7: yetkilərin artırılması
- Windows: CryptoAPI yığılmasının istifadəsi
- Windows: DNS cache-in yenilənməsi
- Windows: OpenVPN-in servis kimi işə düşməsi
- Windows: PUBLIC ya da Private şəbəkə kartları
- Windows: routing metodları

GİRİŞ

Bu başlıqda biz OpenVPN-i əksər əməliyyat sistemlərində həm client və həm də server kimi istifadə edilməsinin üsullarını açıqlayacağıq. Misallar tam şəkildə OpenVPN-in quraşdırmasının özünə əsaslanır. Yeni şəbəkə quruluşu və yüklənməsi deyil yalnız quraşdırmaya əsaslanır.

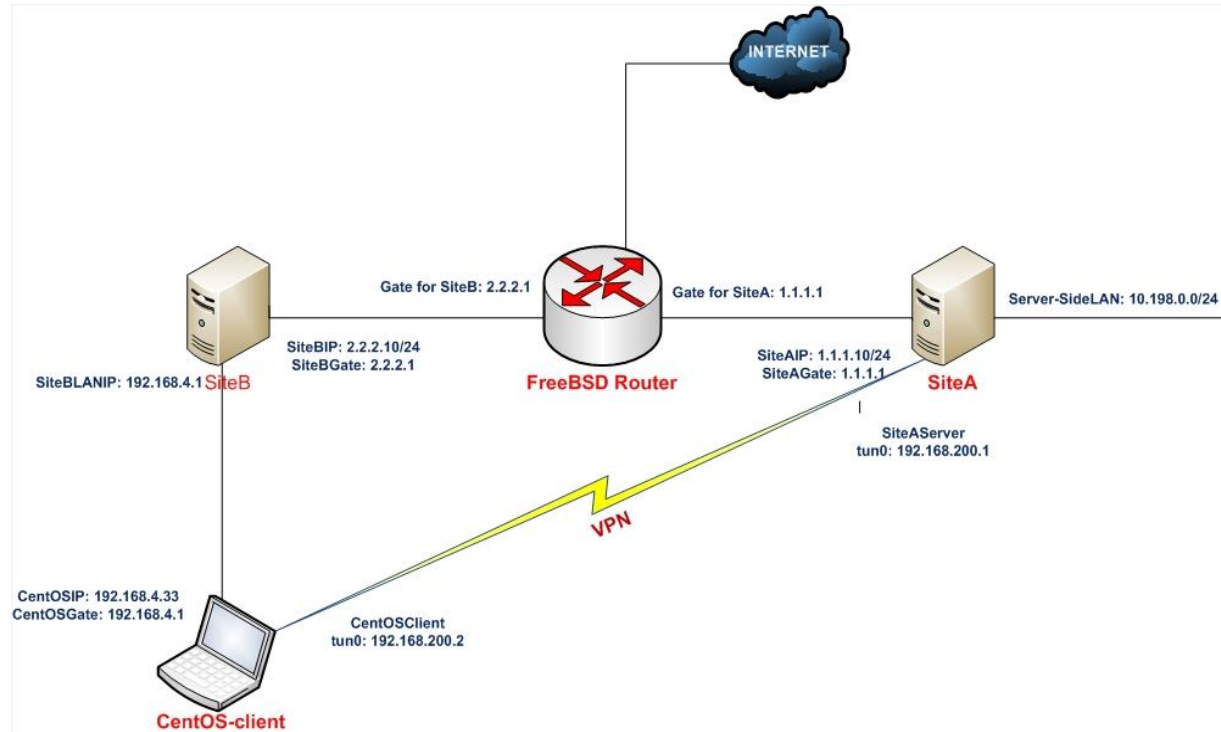
Linux: NetworkManager-in istifadə edilməsi

Əgər Linux bazalı Desktop OS istifadə edilirsə, şəbəkə quraşdırması əksər Linux-larda NetworkManager tərəfindən edilir. Bu paketin sayəsində root olmayan istifadəçi şəbəkə qoşulmalarını start/stop, wireless şəbəkələrində qoşulma/ayrılma və həmçinin OpenVPN-ə qoşulma/ayrılma işlərini edə bilər. Bu misalda biz OpenVPN qoşulmasının GNOME desktopla necə ediləcəyini göstərəcəyik.

İşə hazırlaşaq

2-ci başlıqda istifadə elədiyimiz client və server sertifikatlarını bu başlıqda da istifadə edəcəyik. Bu misalda server maşını FreeBSD 9.2 x64 OpenVPN2.3-də işləyəcək. Client maşını isə CentOS6.5-də olacaq. Ancaq CentOS6.5 maşında OpenVPN-i GUI-dən adi istifadəçi adından quraşdırmaq üçün siz **openvpn.x86_64** və **NetworkManager-openvpn.x86_64** paketlərini sisteme yükləməlisiniz. Həmçinin qeyd eləmək istəyirəm ki, CentOS-un susmaya görə olan anbar siyahısında bu paketlər olmur və siz EPEL reposlarından bu paketi yükləməlisiniz. Server quraşdırması olaraq isə 2-ci başlıqda server-tərəf routingdə yaratdığımız **basic-udp-server.conf** faylından istifadə edəcəyik. CentOS client üçün tələb edilən client sertifikatını, açarını, CA sertifikatını və **ta.key** açarını bu maşına öncədən köçürün. Həmçinin CentOS client maşının **/etc/hosts** faylına öncədən **1.1.1.10**

openvpnservers.example.com sətirinə əlavə edin ki, ad ilə qoşula bilsin. Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



Necə edək...

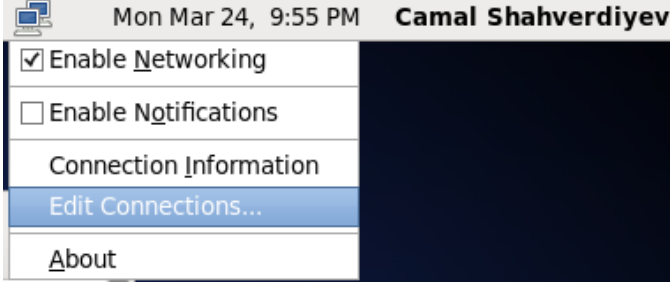
1. Öncə EPEL anbarı yükləyirik və sistemdə olan paketləri yeniləyirik:


```
wget http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
rpm -Uvh epel-release-6-8.noarch.rpm
yum update -y
```

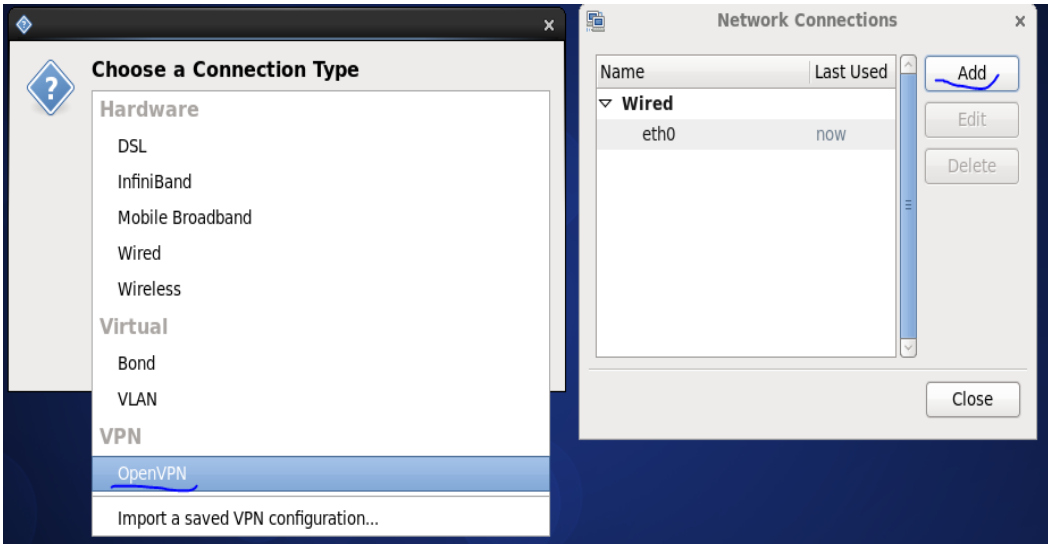
2. Sonra OpenVPN-ə aid olan paketləri yükləyirik:

```
yum -y install `yum search openvpn | grep -v Matched | grep openvpn | awk '{ print $1 }'`
```

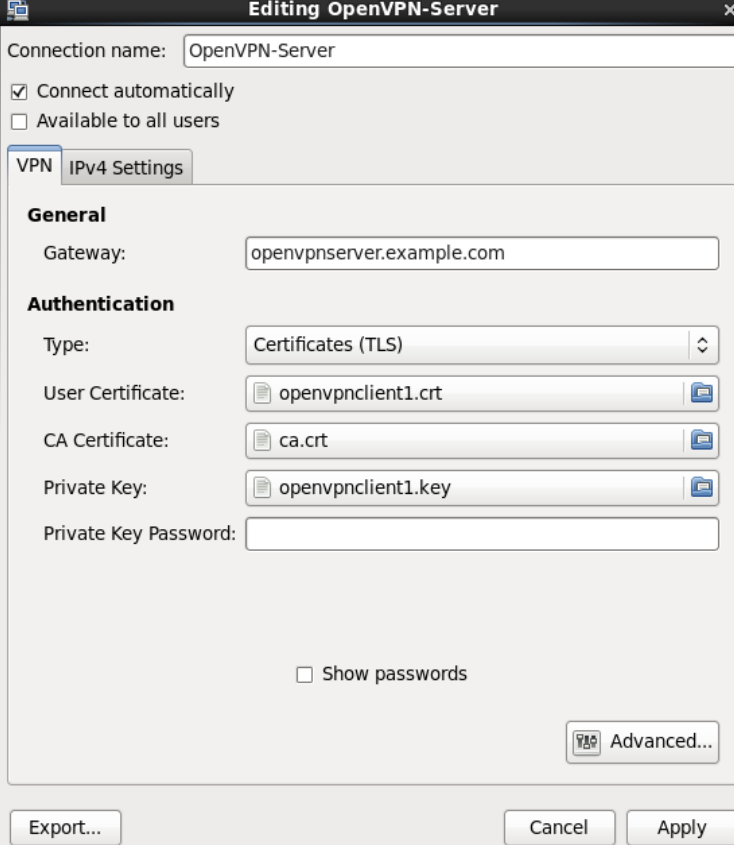
3. Sonra CentOS client maşında adi istifadəçi adından, OpenVPN qoşulmasını yaratmaq üçün şəbəkə kartı işarəsinin üstündə sağ düyməni sıxırıq və **Edit connections** düyməsinə sıxırıq.



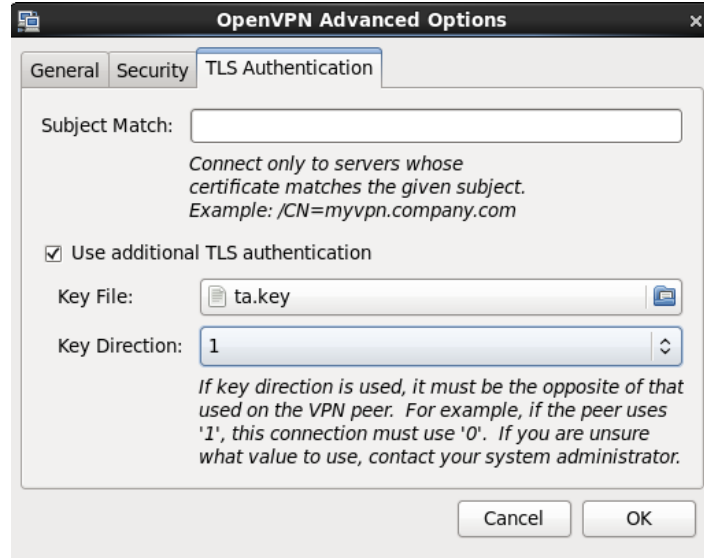
4. Açılan **Network Connections** səhifəsində **Add** düyməsinə sıxırıq, **OpenVPN** seçirik və **Create** düyməsinə sıxırıq:



5. VPN üçün açılan səhifədə isə lazımı verilənləri əlavə edin(Şəkildə görüldüyü kimi bütün sertifikatların ünvanları dəqiq təyin edirik):



6. **Gateway** yazılan ünvanda OpenVPNserver-in adını yazırıq(/etc/hosts faylında əlavə etmişik). Authentifikasiya **Type**-ında **Certificates (TLS)** seçirik çünki sertifikatla qoşulacayıq. **User Certificate**-də **openvpnclient1.crt**, **CA certificate**-də öncədən yaratdığımız **ca.crt**, **Private Key**-də isə **openvpnclient1.key** client açarımız göstəririk. Client sertifikatı üçün şifrə daxil etmirik ona görə ki, **openvpnclient1** üçün generasiya edəndə biz öncədən şifrə təyin etməmişdik. Sonra **Apply** düyməsini sıxmada **Advanced** bölümünə keçin.
7. **Advanced** düyməsini sıxdıqdan sonra isə **TLS Authentication** tab-ına keçirik:

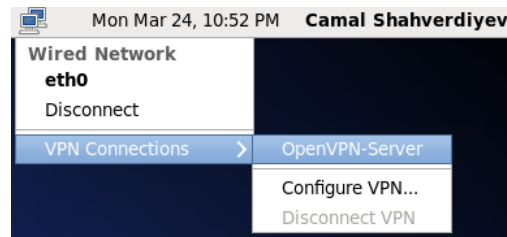


Use additional TLS authentication-a işarə təyin edirik və **ta.key** faylının ünvanını göstəririk. **Key Direction**-da isə **1** seçirik. Sonra **OK** və **Apply** düyməsinə sıxırıq.

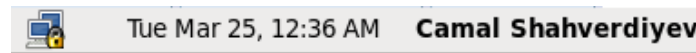
8. Və serveri işə salırıq:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-server.conf
```

9. Və sonda şəbəkə qoşulmasının üstündə sol düyməni sıxıb **VPN connections** bölümünə keçib **OpenVPN-Server**-i seçirik:



OpenVPN qoşulmasının uğurlu olmasını serverin VPN IP-sinə ping atmaqla yoxlaya bilərsiniz ancaq, hər bir halda qoşulma ikonu aşağıdakı şəkildəki kimi olmalıdır:



Biz nə etdik...

Biz Network-Manager-OpenVPN GUI sayəsində qrafik interfeys ilə OpenVPN client-i quraşdırdıq.

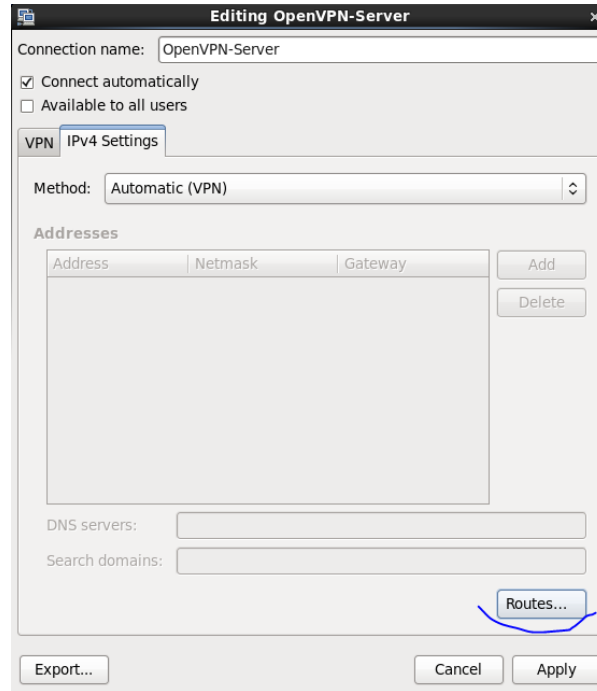
Qeyd: Nəzərə alın ki, köhnə versiya Network-Manager-lər OpenVPN plugin-i dəstəkləmir.

Daha da ətraflı...

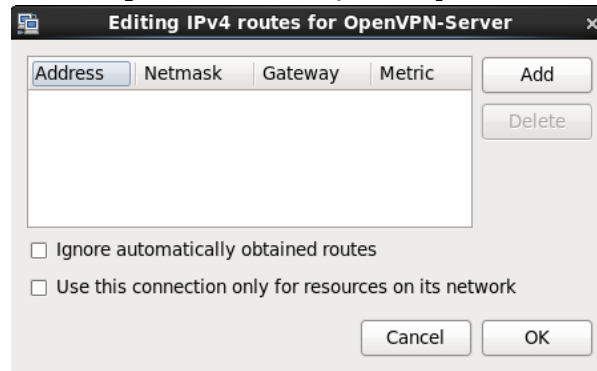
NetworkManager-openvpn plugin-i daha çox quraşdırma imkanlarına malikdir.

NetworkManager istifadə edərək route-ların əlavə edilməsi

NetworkManager-openvpn plugin-i həmçinin VPN-ə uyğun olan route-ların əlavə edilməsində istifadə edilir. VPN-in əsas quraşdırma səhifəsini açın və **IPv4 Settings** bölümünə keçid edin. Sonra isə **Routes** düyməsinə sıxın.



Aşağıdakı şəkildəki kimi yeni səhifə açılacaq:



Server tərəfində göndərilən routing-lər **Ignore automatically obtained routes** istifadə edilərək məhəl qoyulmaya bilər. **'redirect-gateway'** direktivi isə **Use this connection only for resources on its network** seçilməsi ilə rədd edilə bilər.

DNS quraşdırmaları

NetworkManager-openvpn plugin-i həmçinin **/etc/resolv.conf** faylından yenilənə bilər. OpenVPN server isə aşağıdakı direktivi istifadə edərək DNS quraşdırmasını client-ə göndərir:


```
push "dhcp-option DNS a.b.c.d"
```

Scripting

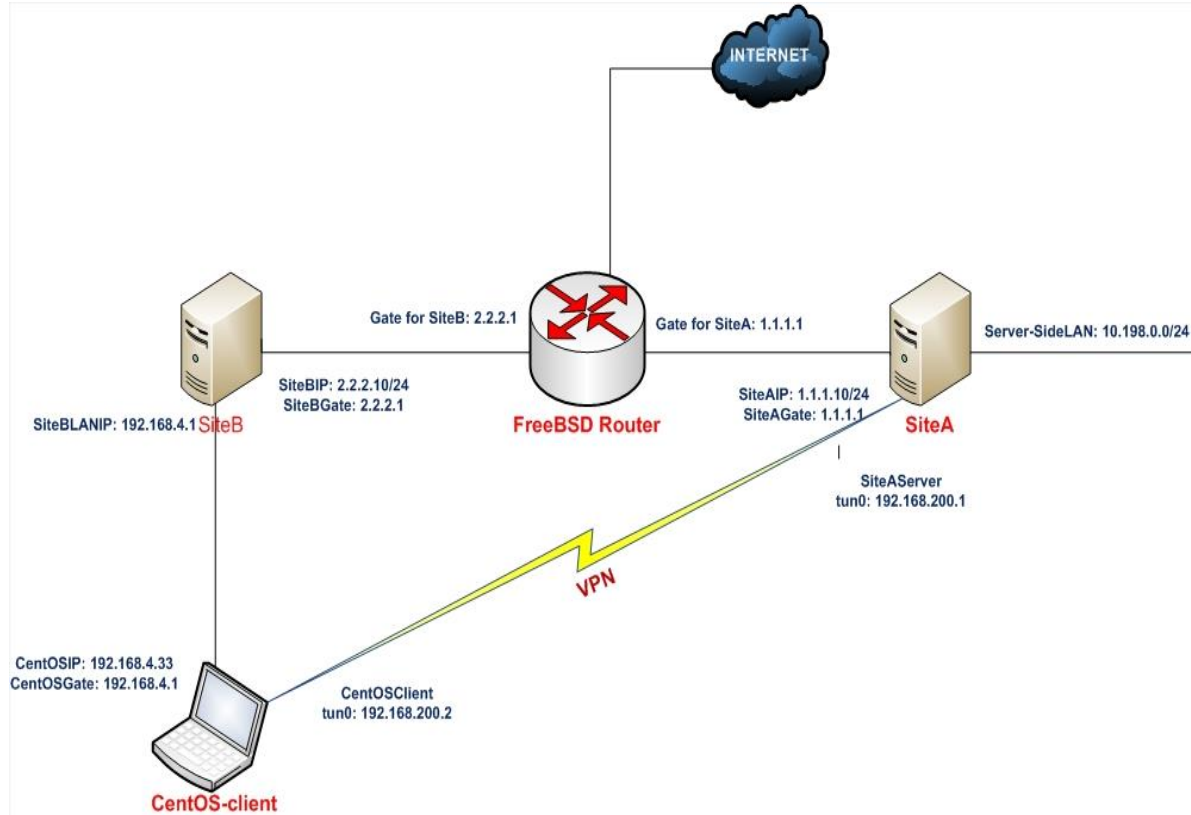
Nəzərə alın ki, **NetworkManager** client tərəfdə script yazma və plugindən istifadəyə izin vermir ona görə ki, bu təhlükəsizlik baxımından çox risklidir.

Linux: "pull-resolv-conf" istifadə edilməsi

Linux üzərində VPN quraşdırılmasının pis cəhətlərindən biri odur ki, OpenVPN server öz DNS quraşdırmalarını clientə oturur. Öncəki misalımızda gördük ki, **NetworkManager-openvpn** plugin-i həmçinin sistemin tutduğu **/etc/resolv.conf** faylında olan DNS quraşdırmalarını da dəyişirdi. Əgər CLI istifadə edilirsə, bu avtomatik olmayacaq. Susmaya görə OpenVPN iki scriptlə gəlir ki, **/etc/resolv.conf** faylına DNS serverləri əlavə edib silə bilsin. Bu misal həmçinin scriptlərin necə istifadə edilməsini göstərəcək.

İşə hazırlaşaq

Biz aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



2-ci başlıqda istifadə elədiyimiz client və server sertifikatlarını bu başlıqda da istifadə edəcəyik. Bu misalda server maşını FreeBSD 9.2 x64 OpenVPN2.3-də işləyəcək. Client maşını isə CentOS64-də olacaq.

Ancaq CentOS6.5 maşında OpenVPN-i GUI-dən adi istifadəçi adından quraşdırmaq üçün siz **openvpn.x86_64** və **NetworkManager-openvpn.x86_64** paketlərini sisteme yükləməlisiniz. Həmçinin qeyd etmək istəyirəm ki, CentOS-un susmaya görə olan anbar siyahısında bu paketlər olmur və siz EPEL anbarlarından bu paketi yükləməlisiniz. Server quraşdırması olaraq isə 2-ci başlıqda server-tərəf routingdə yaratdığımız **basic-udp-server.conf** faylından istifadə edəcəyik. Həmçinin client üçün 2-ci başlıqda Server-tərəf routing üçün yaratdığımız **basic-udp-client.conf** faylından istifadə edəcəyik. CentOS client üçün tələb edilən client sertifikatını, açarını, CA sertifikatını və **ta.key** açarını bu maşına öncədən köçürün. Həmçinin CentOS client maşının **/etc/hosts** faylına öncədən **1.1.1.10 openvpnsrver.example.com** sətirinə əlavə edin ki, ad ilə qoşula bilsin.

Necə edək...

1. **basic-udp-server.conf** faylını **example10-2-server.conf** faylına nüsxələyin və **example10-2-server.conf** faylının sonuna aşağıdakı sətiri əlavə edin:

```
push "dhcp-option DNS 8.8.8.8"
```

Burda 8.8.8.8 GOOGLE DNS serverin IP ünvanıdır.
2. Serveri işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example10-2-server.conf
```
3. Uyğun olaraq **basic-udp-client.conf** faylını CentOS client maşına köçürün və bu faylı **example10-2-client.conf** faylına nüsxələyin. Sonra **example10-2-client.conf** faylının sonuna aşağıdakı sətirləri əlavə edin:

```
script-security 2
up "/home/camal/Desktop/client.up"
down "/home/camal/Desktop/client.down"
```
4. **client.up** və **client.down** fayllarının OpenVPN2.3-ün **contrib** qovluğundan quraşdırma faylında təyin etdiyimiz ünvanı nüsxələyin və onları yerinə yetirilən edin. CentOS6.5-də bu qovluq **/usr/share/doc/openvpn-2.3.2/contrib/pull-resolv-conf** ünvanında yerləşir.

```
[root@openvpn-centos pull-resolv-conf]# cp /usr/share/doc/openvpn-2.3.2/contrib/pull-resolv-conf/client.* /home/camal/Desktop/
```



```
[root@openvpn-centos pull-resolv-conf]# chmod 755 /home/camal/Desktop/client.*
```
5. Sonda isə client-i işə salın (Ancaq CLI-dan root istifadəçi adından işə salın çünki, adi istifadəçinin tun aləti yaratmaq yetkisi hələ yoxdur):

```
[root@openvpn-centos Desktop]# openvpn --config example10-2-client.conf
```

VPN qoşulması uğurlu olduqdan sonra isə, **/etc/resolv.conf** faylına baxsanız aşağıdakı sətirləri görməlisiniz:

```
# resolv.conf autogenerated by /home/camal/Desktop/client.up (tun0)
nameserver 8.8.8.8
domain
```

VPN qoşulası kəsildikdən sonra isə bu sətirlər avtomatik olaraq silinəcək.

Bu necə işləyir...

Bu scriptlər OpenVPN ilə mühit dəyişənləri mənisədir **foreign_option_***, **DOMAIN** və **DNS** quraşdırmaları. Bu quraşdırmalar sonra **/etc/resolv.conf** faylının önünə yazılır. Bu ona görədir ki, OpenVPN Server tərəfindən ötürülən DOMAIN və DNS quraşdırmaları sistemin özündə olanlardan üstün olsun.

VPN qoşulması kəsildikdən sonra isə, eyni quraşdırmalar **/etc/resolv.conf** faylından silinir.

Daha da ətraflı...

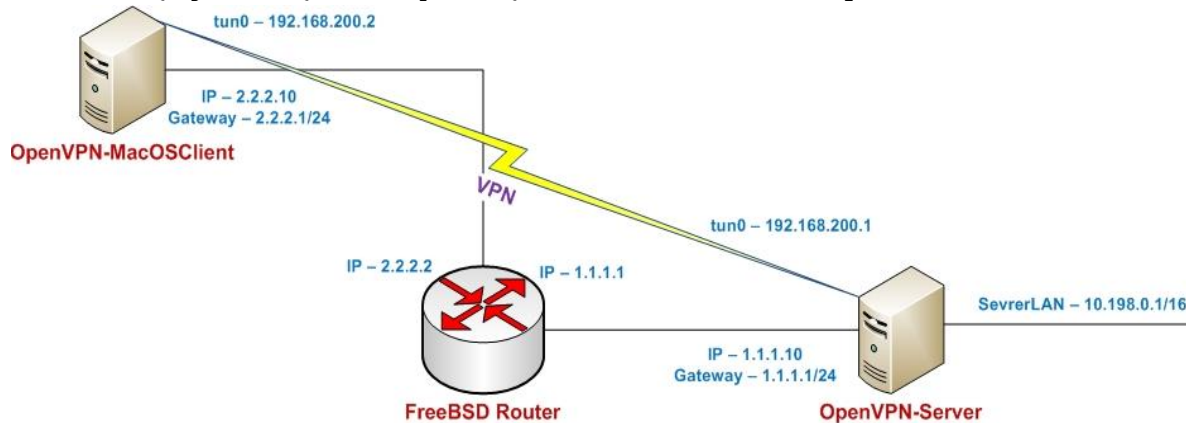
Nəzərə alın ki, NetworkManager-plugin istifadə edildikdə, öncə göstərilən scriptlərə ehtiyac yoxdur ona görə ki, NetworkManager-plugin-i avtomatik olaraq **/etc/resolv.conf** faylını yeniləyir.

MacOS: Tunnelblick istifadə edilməsi

Bu misal OpenVPN-in client olaraq MacOS X maşınlarında necə quraşdırılmasını göstərir. MacOS X üçün çoxlu OpenVPN GUI proqramları mövcuddur. Bu misalda biz onlardan birini göstərəcəyik. Tunnelblick (<http://code.google.com/p/tunnelblick/>)

İşə hazırlaşaq

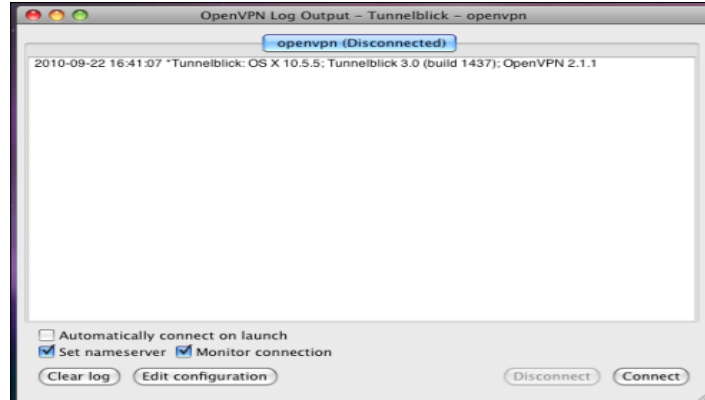
Yenə də aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Server maşın FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Client maşını isə MacOS X "Leopard", Tunnelblick3.3 və OpenVPN2.3-də olacaq. Server tərəfdə quraşdırma olaraq **example10-2-server.conf** yenə də istifadə edəcəyik. MacOS maşına **ca.crt**, **openvpnclient1.crt**, **openvpnclient1.key** və **ta.key** fayllarını köçürün. MacOS-un **/etc/hosts** faylına **1.1.1.10** **openvpnsrvr.example.com** sətirini əlavə etməyi unutmayın.

Necə edək...

1. Əgər tunnelblick işləmirsə işə salın.
2. Tunnelblick pəncərəsi işə düşdükdən sonra tunel ikonkasına sıxın:



3. **Edit configuration** düyməsinə sıxın ki, **Text Editor**-u susmaya görə olan quraşdırma faylı ilə açasınız. Client quraşdırmasının adını **openvpn.conf** edin və aşağıdakı sətirləri içinə əlavə edin:

```
client
proto udp
remote openvpnservers.example.com
port 1194

dev tun
nobind

ca /etc/openvpn/ca.crt
cert /etc/openvpn/openvpnclient1.crt
key /etc/openvpn/openvpnclient1.key
tls-auth /etc/openvpn/ta.key 1

ns-cert-type server
```

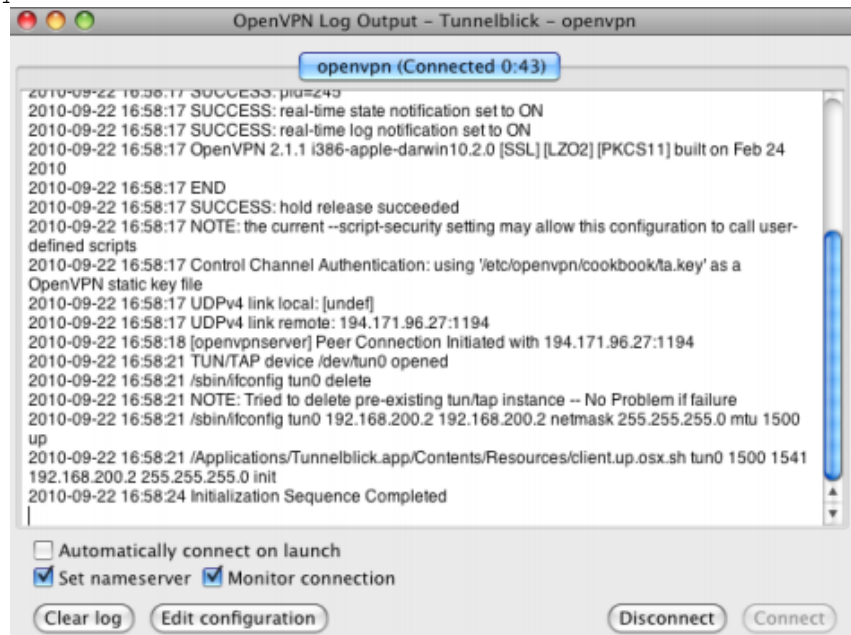
Quraşdırma faylını yadda saxlayın və mətn redaktorundan çıxın.

4. Əgər Tunnelblick xəbərdarlıq etsə ki, fayl müdafiədədir **Unprotect and modify** düyməsinə sıxın ki, faylda dəyişiklik edə bilərsiniz.
5. Sonra serveri işə salın:

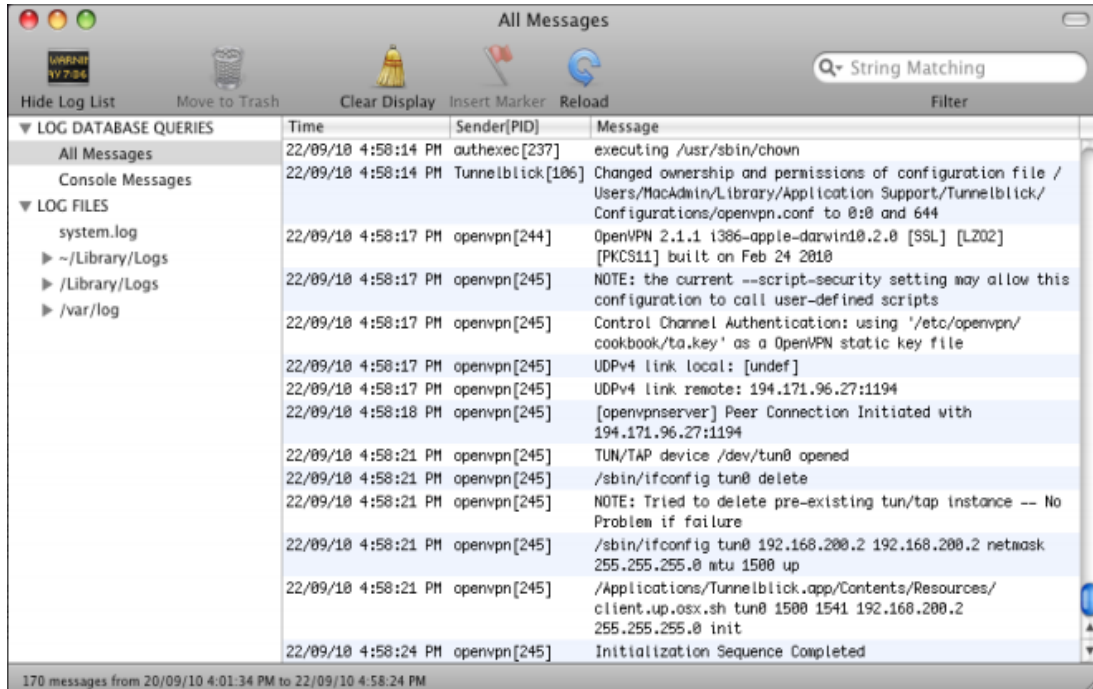

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example10-2-server.conf
```
6. Sonra client-də əsas **Tunnelblick** səhifəsində **Connect** düyməsinə sıxın. OpenVPN qoşulması uğurlu olmazdan öncə **Tunnelblick** yeni səhifə açacaq. Ona görə ki, faylda dəyişiklik edilib və **Tunnelblick MacAdmin** şifrəsini soruşacaq ki, lazımi yetkini əldə edə bilər:



Mac Admin şifresi daxil edildikdən sonra OpenVPN qoşulması uğurlu olacaq:



Əgər hansısa bir fazada sizdə problem yaranırsa, **Tunnelblick** və **OpenVPN**-ə aid olan bütün mesajlar **Console** utilitinin **All Messages** səhifəsində tapıla bilər. **Console** utilit adətən sistem diskində **Utilities** qovluğunda tapılır:



| LOG DATABASE QUERIES | Time | Sender[PID] | Message |
|----------------------|---------------------|------------------|--|
| All Messages | 22/09/10 4:58:14 PM | authexec[237] | executing /usr/sbin/chown |
| Console Messages | 22/09/10 4:58:14 PM | Tunnelblick[106] | Changed ownership and permissions of configuration file /Users/MacAdmin/Library/Application Support/Tunnelblick/Configurations/openvpn.conf to 0:0 and 644 |
| LOG FILES | | | |
| system.log | 22/09/10 4:58:17 PM | openvpn[244] | OpenVPN 2.1.1 i386-apple-darwin0.2.0 [SSL] [LZO2] [PKCS11] built on Feb 24 2010 |
| ~/Library/Logs | 22/09/10 4:58:17 PM | openvpn[245] | NOTE: the current --script-security setting may allow this configuration to call user-defined scripts |
| /Library/Logs | 22/09/10 4:58:17 PM | openvpn[245] | Control Channel Authentication: using '/etc/openvpn/cookbook/ta.key' as a OpenVPN static key file |
| /var/log | 22/09/10 4:58:17 PM | openvpn[245] | UDPv4 link local: [undef] |
| | 22/09/10 4:58:17 PM | openvpn[245] | UDPv4 link remote: 194.171.96.27:1194 |
| | 22/09/10 4:58:18 PM | openvpn[245] | [openvpnserver] Peer Connection Initiated with 194.171.96.27:1194 |
| | 22/09/10 4:58:21 PM | openvpn[245] | TUN/TAP device /dev/tun0 opened |
| | 22/09/10 4:58:21 PM | openvpn[245] | /sbin/ifconfig tun0 delete |
| | 22/09/10 4:58:21 PM | openvpn[245] | NOTE: Tried to delete pre-existing tun/tap instance -- No Problem if failure |
| | 22/09/10 4:58:21 PM | openvpn[245] | /sbin/ifconfig tun0 192.168.200.2 192.168.200.2 netmask 255.255.255.0 mtu 1500 up |
| | 22/09/10 4:58:21 PM | openvpn[245] | /Applications/Tunnelblick.app/Contents/Resources/client.up.osx.sh tun0 1500 1541 192.168.200.2 255.255.255.0 init |
| | 22/09/10 4:58:24 PM | openvpn[245] | Initialization Sequence Completed |

Bu necə işləyir...

Tunnelblick spesifik GUI program təminatıdır ki, openvpn əmrlərinə CLI-dan müraciət edir. Və client üçün CLI-dan tələb edilən istənilən işləri GUI vasitəsilə yerinə yetirə bilər.

Digər sözlə MacOS X üzərində işləyən OpenVPN versiyası heçnəyi ilə UNIX/LINUX-da olanlardan geri qalmır. Həmçinin UNIX/Linux üzərində işləyən istənilən scriptləri asanlıqla MacOS üzərindədə işlədə bilərsiniz.

Dahada ətraflı...

UNIX və MacOS OpenVPN versiyaları arasında iki əsas fərq var və aşağıda onları açıqlayırıq.

Name resolution

Əgər Tunnelblick-də nameserver qutucuğunda seçim etmisinizsə, onda o UP scriptini istifadə edərək OpenVPN server tərəfindən ötürülən DNS serveri istifadə edəcək. Həmçinin **/etc/resolv.conf** faylında DNS informasiyasını yeniləyəcək.

Scripting

Tunnelblick client tərəfdə scriptinge izin vermir və yalnız özündə olan up və down scriptlərindən istifadə edir.

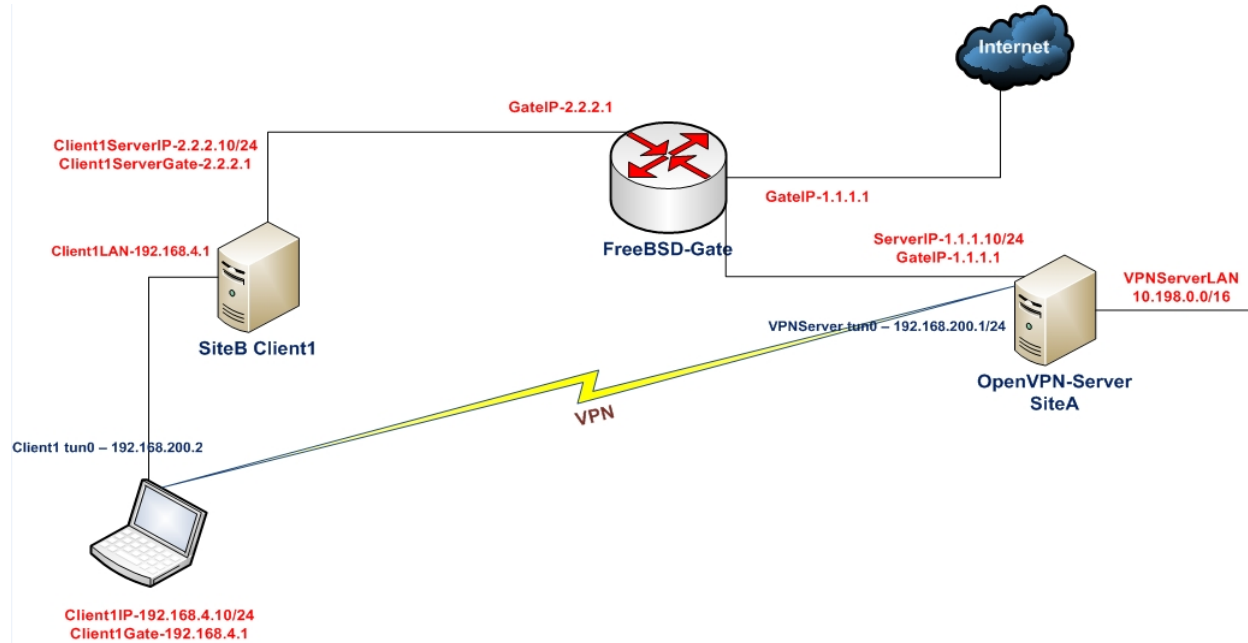
Windows7: yetkilərin artırılması

Windows Vista-dan başlayaraq Microsoft yeni imkan **UAC(User Access Control)** yaratdı. UAC istifadəçilər tərəfində sistemi dəyişdirə biləcək programların işə salınmasının qarşısını alır. İstənilən bir programın işə salınması üçün mütləq yetkilərin artırılması tələb edilir hətta, istifadəçi Administrator olsa da belə. Bu halda dialog pəncərəsi çıxacaq ki, istifadəçi yerinə yetirilmədən öncə ona sızlamlıdır. OpenVPN-in işə düşməsi üçün artırılmış yetkilər ona görə lazım olur ki, OpenVPN VPN qoşulmasını yaradanda sistemə route əlavə etmək istəyir.

Bu misal Windows7 maşında yetkilərin necə artırılmasını və **up, down** scriptlərinin necə istifadə edilməsini göstərəcək.

İşə hazırlaşaq

2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Client maşını isə Windows7 x64 OpenVPN2.3-də olacaq. Server üçün 2-ci başlıqda server tərəf routing üçün yaratdığımız **basic-udp-server.conf** quraşdırma faylından istifadə edəcəyik. Client üçün isə 6-cı başlıqda **up/down** scriptlər üçün istifadə elədiyimiz **example6-1.ovpn** quraşdırma faylından istifadə edəcəyik. Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



Necə edək...

1. Serveri işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-server.conf
```
2. Sonra client maşında **example6-1.ovpn** faylını **example10-4.ovpn** faylına nüsxələyirik və **example10-4.ovpn** faylının içində aşağıdakı dəyişiklikləri edirik:
 Bu sətirləri:

```
script-security 2 system
```

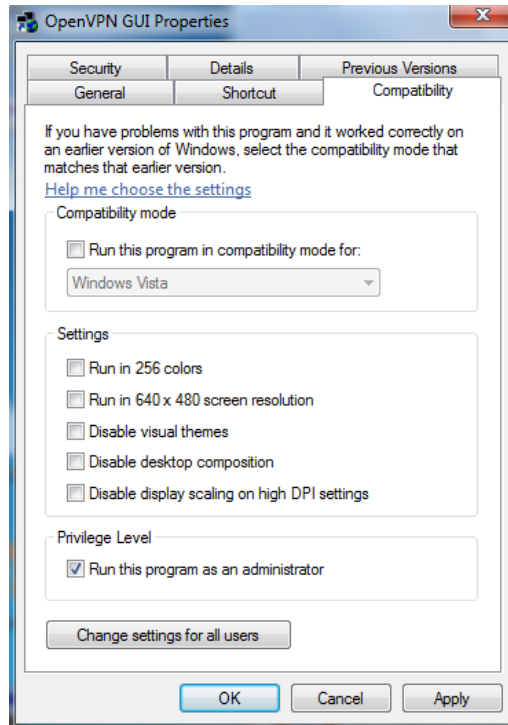
```
up "C:\\updownfold\\updown.bat"
down "C:\\updownfold\\updown.bat"
```

Dəyişiklik aşağıdakılara:

```
script-security 2 system
cd "c:\\program\\files\\openvpn\\scripts"
up "%windir%\\system32\\cmd.exe /c updown.bat"
down "%windir%\\system32\\cmd.exe /c updown.bat"
```

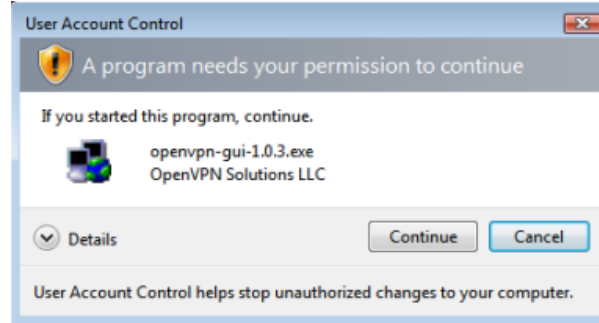
Bu yerinə yetirilən **batch** fayllar üçün tələb edilir hansı ki, OpenVPN tərəfindən yerinə yetiriləcək.

3. OpenVPN-i Windows maşına yüklədikdən sonra, Windows7 Desktop-da yaranan OpenVPN GUI-nin üstündə sağ düyməni sıxın.
4. **Properties** daxil olduğdan sonra, **Compability**-ə sıxın və **Run this program as an administrator** qutucuğunda seçim edin:



Sonra **OK** düyməsinə sıxın.

5. Ola bilər ki, növbəti dəfə programı işə saldıqda aşağıda göstərilən şəkildə sizdə göstəriləcək ki, yetkilərin artırılması izni alsın:



6. **Continue** düyməsinə sıxın. Bu ardıcılığımız o deməkdir ki, OpenVPN sistemdə tam maksimal yetki ilə işlədi.
7. Client maşında **example10-4.ovpn** quraşdırma faylı ilə vpn-ə qoşulun və VPN statusuna baxın ki, qoşulma uğurlu oldu. (Yalnız uğursuz nəticə əldə etsəniz: tək dırnaq, cüt dırnaq, geriye slash, və adında boşluq olamayan ünvanlarla yoxlanış edin. Bu tip problemlər OpenVPN-də çox çıxır)

Necə işləyir...

OpenVPN GUI işə düşəndə istifadəçi təsdiqləməlidir ki, o Administrator yetkiləri ilə işləyəcək. Bundan sonra isə OpenVPN GUI həm yerinə yetirilən faylları, həm adapterin yaradılmasını və ya routing cədvəlinin əlavə edilməsini problemsiz edəcək. Bir işi edə bilməyəcək hansı ki, batch faylını birbaşa işə sala bilməyəcək.

Windows: CryptoAPI yığımının istifadə edilməsi

OpenVPN-də imkan var ki, qoşulma üçün tələb edilən public və private açarları CryptoAPI anbarında saxlasın. Bu təhlükəsizliyi xeyli artırır ona görə ki, **CryptoAPI** anbarı adi plaintext-də saxlanılan **key** və **crt** fayllarından xeyli təhlükəsizdir. Bu misalda biz OpenVPN-i elə quracağıq ki, qoşulma üçün tələb edilən məlumatları CryptoAPI-dən alsın ki, serverə normal qoşula bilsin. Bu testi Windows7 üzərində edəcəyik.

İşə hazırlaşaq

2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Client maşını isə Windows7 x64 OpenVPN2.3-də olacaq. Server üçün 2-ci başlıqda server tərəf routing üçün yaratdığımız **basic-udp-server.conf** quraşdırma faylından istifadə edəcəyik:

Necə edək...

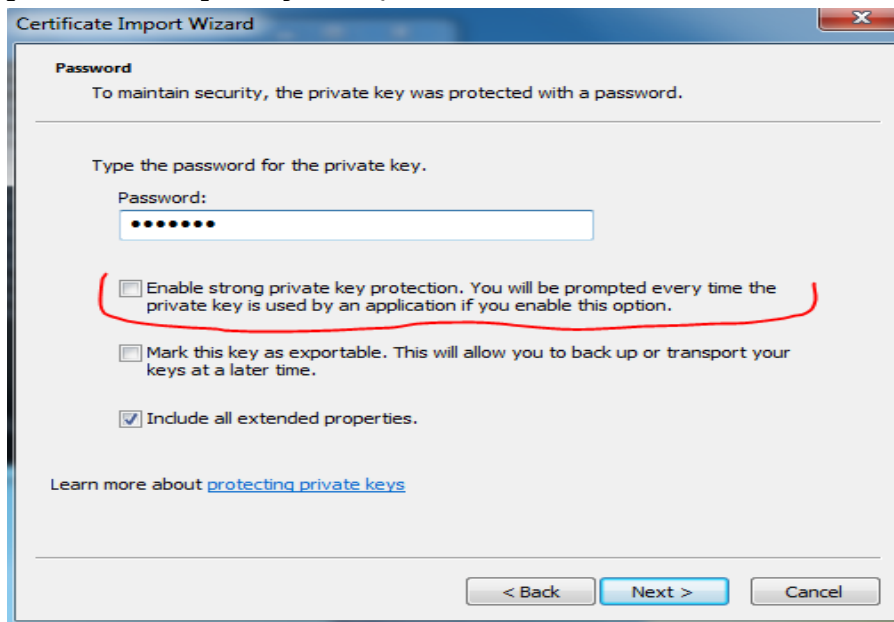
1. Öncə biz client sertifikatını CryptoAPI anbarda saxlamalıyıq. Qayda ilə bunu eləmək üçün biz öncə **openvpnclient2.crt** və **openvpnclient2.key** fayllarını PKCS12 formatına convert eləməliyik. Windows maşınıınızda CLI-dan daxil olun və həmin fayllar yerləşən ünvana daxil olun:
C:\Users\clientb>cd C:\Program Files\OpenVPN\config

```
C:\Program Files\OpenVPN\config>..\bin\openssl pkcs12 -export -in
openvpnclient2.crt -inkey openvpnclient2.key -out openvpnclient2.p12
WARNING: can't open config file: /etc/ssl/openssl.cnf
Loading 'screen' into random state - done
Enter pass phrase for openvpnclient2.key:
Enter Export Password:
Verifying - Enter Export Password:
```

2. Sonra PKCS12 faylını Windows CryptoAPI anbarına import edək:
C:\Program Files\OpenVPN\config>**start openvpnclient2.p12**

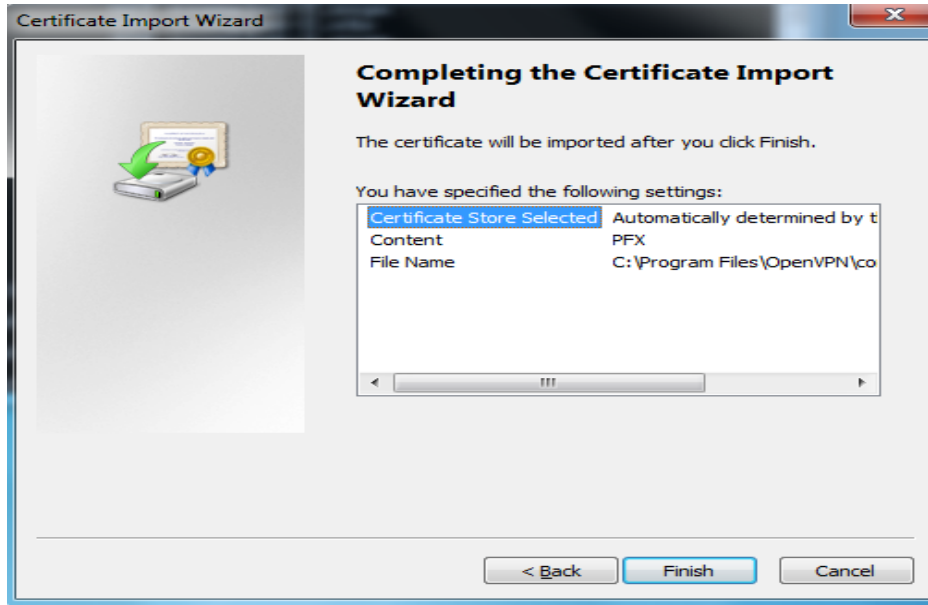
Sertifikatın importu üçün köməkçi ekrana gələcək.

3. Birinci və ikinci pəncərələrdə **Next** düyməsinə sıxın. Sonra isə **Export password**-da yazdığınız şifrəni daxil etməlisiniz:



Əgər siz **Enable strong private key protection** qutucuğunu seçsəniz, sertifikat faylı və key faylı çox güclü qorunacaq ancaq, siz hər dəfə OpenVPN işə düşdükdə şifrəni daxil etməlisiniz.

4. **Next** düyməsini sıxın və növbəti səhifədə **Automatically select the certificate store based on the type of certificate** susmaya görə olan opsiyanı seçin və **Next** düyməsini sonadək ardıcıl sıxın. Sondakı səhifədə **Finish** düyməsini sıxmaqla artıq sertifikatın import işini bitirmiş olacaq:



5. Windows7 maşında **example10-5.ovpn** adlı client quraşdırma faylını yaradın və içine aşağıdakı sətirləri əlavə edin:

```

client
proto udp
remote openvpnserver.example.com
port 1194

dev tun
nobind

ca "c:/program files/openvpn/config/ca.crt"
tls-auth "c:/program files/openvpn/config/ta.key" 1
cryptoapicert "SUBJ:OpenVPNClient2"

ns-cert-type server

```

6. Sonra serveri işə salın:
- ```

root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-
server.conf

```

7. Sonra client-i işə salın.

VPN qoşulması rahat olmalıdır və qoşulma anı client şifrəsi sizdən soruşulmalı deyil. Əgər CryptoAPI opsiyası **Enable strong private key protection** aktivdirsə, onda uyğun pəncərə çıxıb sizdən şifrə istəyəcək.

### **Bu necə işləyir...**

Windows-da olan OpenVPN program təminatının imkanı vardır ki, Windows CryptoAPI-dən sertifikat və public açarı açsın. Əgər göstərilən sertifikatın subject adı **SUBJ:** yada fingerprinti **THUMB:** açar sözlə göstərilibse bunu eləmək mümkün olacaq. CryptoAPI anbardan sertifikat və private key faylı əldə edildikdən sonra işə VPN qurulma qoşulur.

## Daha da ətraflı...

Windows CryptoAPI istifadə edildikdə bir neçə önəmli nöqtələr var:

### CA sertifikat faylı

Mütləq yenə də CA sertifikat faylının tam ünvanını göstərmək lazımdır (aşağıdakı sətirdəki kimi):

```
ca c:/program files/openvpn/config/ca.crt
```

Nəzəriyyəyə əsaslanaraq CryptoAPI-dən həmçinin CA sertifikatını-da almaq olar ancaq, OpenVPN hələki bunu dəstəkləmir.

### Certificate fingerprint

Yerinə istifadə edilir:

```
cryptoapicert SUBJ:<subject name>
```

Həmçinin aşağıdakı kimi təyin etmək mümkündür:

```
cryptoapicert THUMB:<fingerprint>
```

Thumbprint yada fingerprint-i siz x509 sertifikatından əldə edə bilərsiniz.

Bunu Windows sertifikat anbarından OpenSSL əmri ilə əldə edə bilərsiniz:

```
C:\Program Files\OpenVPN\config>..\bin\openssl x509 -fingerprint -noout
-in openvpnclient2.crt
WARNING: can't open config file: /etc/ssl/openssl.cnf
SHA1
Fingerprint=7D:90:AE:AA:44:9A:10:8D:1F:90:E9:4B:57:9F:E7:33:CC:D3:BF:59
```

## Windows: DNS cache-in yenilənməsi

OpenVPN-in mail list suallarında əksər zaman DNS-lə bağlı şikayətlənirlər. Client qoşulmanı etdikdən sonra OpenVPN-dən aldığı DNS həmişə həmin anda da işləmir çünki, köhnə DNS-lər hələdə client-in cache-ində durur. Bunu OpenVPN-də və birazda Windows DNS servisdə kiçik dəyişikliklə həll etmək olar. OpenVPN2.1.3-dən başlayaraq yeni direktiv yarandı **register-dns**. Bu direktivin sayəsində OpenVPN həm Windows DNS cache-ində yenilənmə edir və həm də VPN IP ünvanını Windows DNS cədvəlində qeydiyyat alır. Bu misalımızda biz VPN qoşulması yarandıqdan sonra scriptin istifadəsilə Windows DNS cache-ində yenilənmə edəcəyik. Bəzi istifadəçilər birdəfəlik DNS cache-lənməni söndürürlər ancaq, bu davamiyyətin gücünü nisbətən alır.

## İşə hazırlaşaq...

2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Client maşını isə Windows7 x64 OpenVPN2.3-də olacaq. Server üçün Linux: **pull-resolv-conf** istifadəsi üçün yaratdığımız **example10-2-server.conf** quraşdırma faylından istifadə edəcəyik. Client üçün isə 2-ci başlıqda **'ifconfig-pool'** blockunun istifadəsi üçün yaratdığımız **basic-udp-client.ovpn** faylından istifadə edəcəyik.

## Necə edək...

1. Serveri işə salaq:

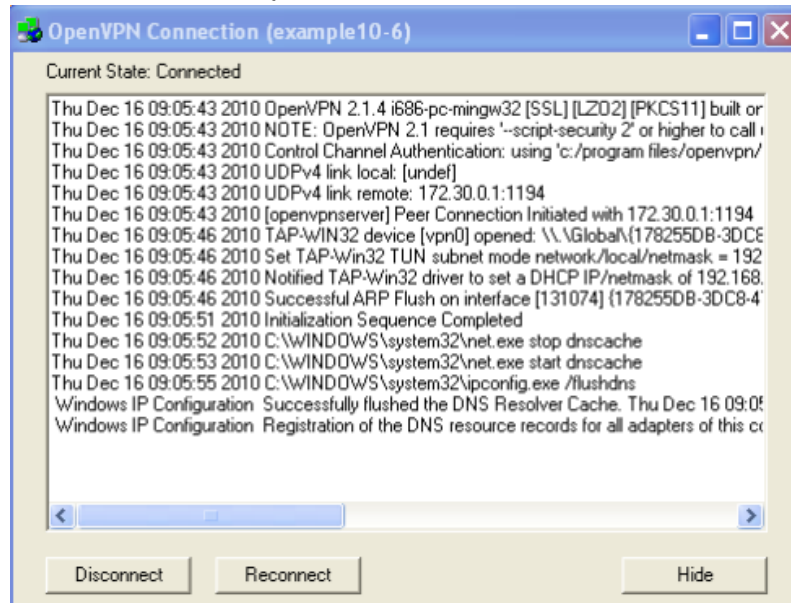
```
root@siteA:/usr/local/etc/openvpn # openvpn --config example10-2-server.conf
```

2. **basic-udp-client.ovpn** quraşdırma faylını **example10-6.ovpn** faylına nüsxələyin və **example10-6.ovpn** faylının sonuna aşağıdakı sətiri əlavə edək:

```
register-dns
```

3. OpenVPN client-i işə salın.

OpenVPN GUI status pəncərəsi göstərəcək ki, Windows-in dnscache servisi restart edildi (Aşağıdakı şəkildə windowsXP-dir ancaq Windows7-də servislərin restart-ı çıxmadı):



4. VPN qoşulması uğurlu olduqdan sonra nslookup əmri ilə VPN-dən əldə edilən DNS-i yoxlayın.

### **Bu necə işləyir...**

VPN qoşulma uğurlu olduqdan sonra, OpenVPN client proqramı DHCP paketlərini TAP-Win32 adapterinə IP ünvan, default gateway və digər şəbəkəyə aid olan məlumatlarla yollayır. DNS serverdə onların içində olur. Məlumat clientə düzgün çatır ancaq DNS cache servisi bu məlumatın həmin anda da işə düşməsi üçün qərar vermək imkanına malik olmur. Əlavə elədiyimiz **register-dns** direktivi bu işi aşağıdakı əmrlərlə görür:

```
net stop dnscache
net start dnscache
ipconfig /flushdns
ipconfig /registerdns
```

DNS servisini sərt olaraq restart elədiyimiz üçün DNS servisi yeni DNS məlumatlarını həmin anda da əldə edir.

### **Daha da ətraflı...**

OpenVPN2.1.3-ədək isə bu işi UP scripti ilə görmək lazım idi. Client-in quraşdırma faylına aşağıdakı sətirlər əlavə edilirdi:

```
script-security 2 system
cd "c:\\program\ files\\openvpn\\config"
up "%windir%\\system32\\cmd.exe /c example10-6.bat"
```

Və **example10-6.bat** bat faylında aşağıdakı sətirlər olacaq:

```
@echo off
net stop dnscache
net start dnscache
```

### **Windows: OpenVPN-in servis kimi işə düşməsi**

OpenVPN-in Windows versiyasına olan üstünlüklərindən biri də odur ki, onu Windows servis kimi istifadə etmək olur. Yəni ki, system qalxdıqda istifadəçi öz maşınına daxil olmadan VPN avtomatik olaraq servisini işə salır və VPN qoşulmasını da avtomatik olaraq edir. OpenVPN öz servisini susmaya görə Windows-a yükləyir ancaq avtomatik işə salınması aktiv olmur.

Bu misalda biz OpenVPN GUI-ni istifadə edərək öz servisinin necə idarə edilməsini göstərəcəyik.

### **İşə hazırlaşaq**

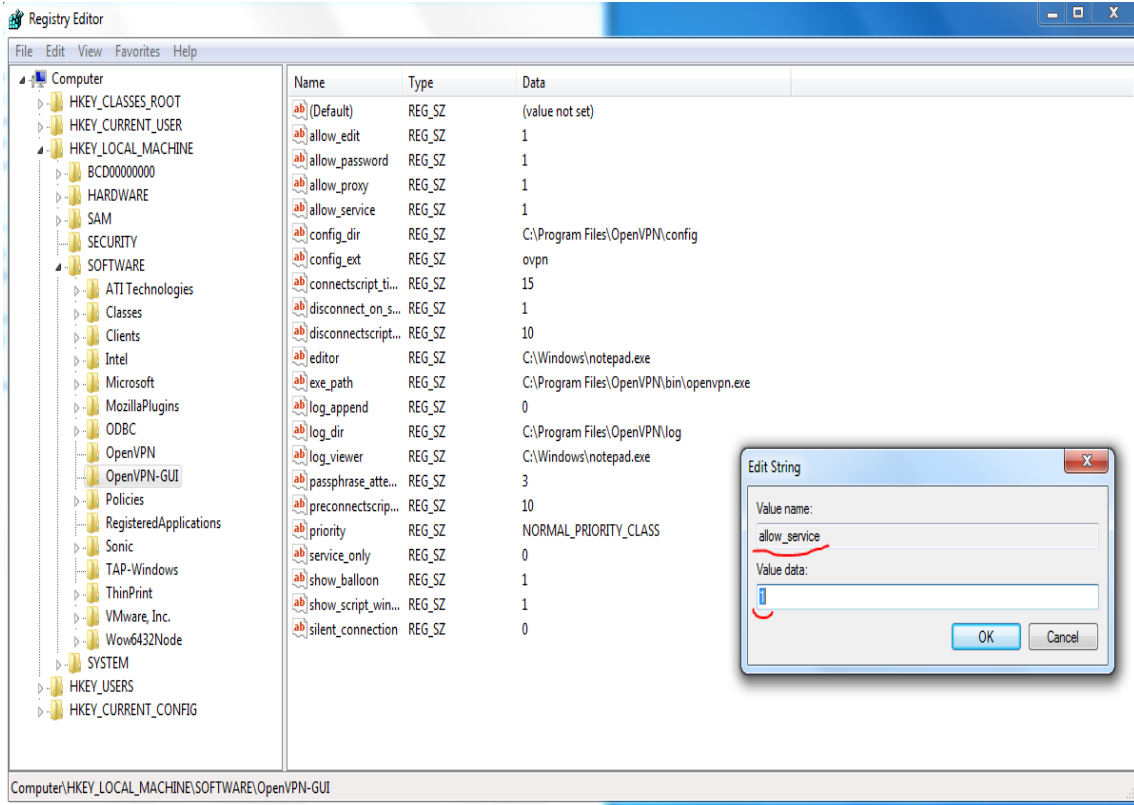
2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Client maşını isə Windows7 x64 OpenVPN2.3-də olacaq. Server üçün 2-ci başlıqda server tərəf routing üçün yaratdığımız **basic-udp-server.conf** quraşdırma faylından istifadə edəcəyik. Client üçün də həmçinin 2-ci başlıqda yaratdığımız **basic-udp-client.ovpn** quraşdırma faylından istifadə edəcəyik.

### **Bunu necə edək...**

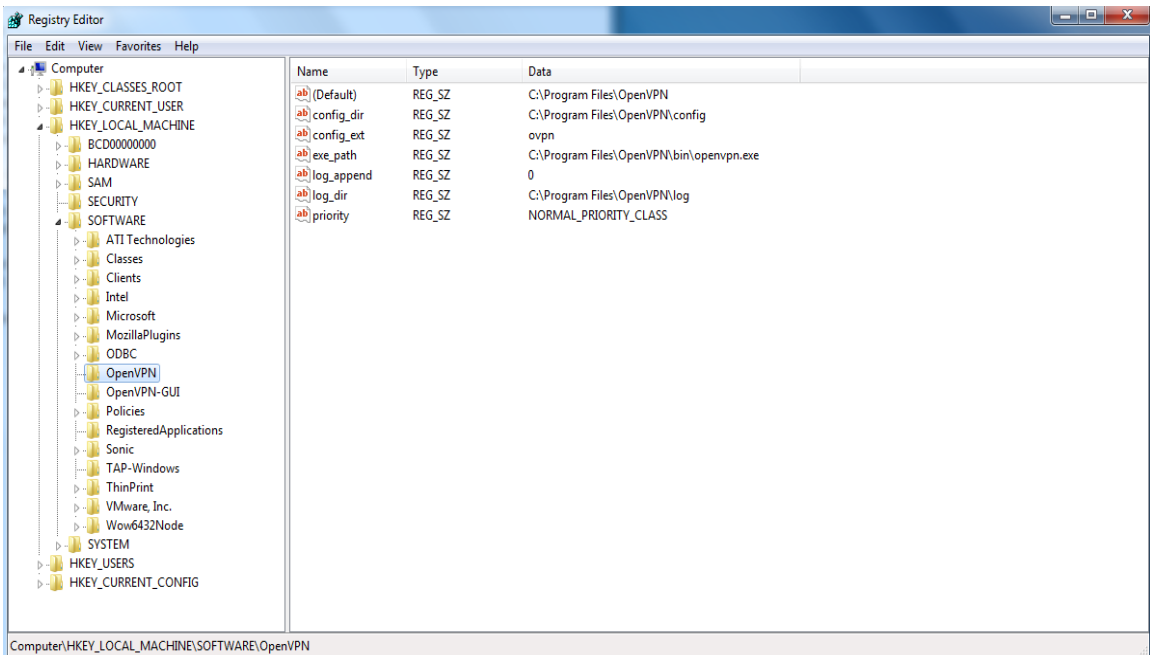
1. Serveri işə salaq:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-server.conf
```

2. OpenVPN GUI-ni işə salmazdan öncə **Windows -> RUN -> regedit** bölməsinə keçin və Windows Reestrində **HKEY\_LOCAL\_MACHINE\SOFTWARE\OpenVPN-GUI** hissəsinə keçin. Diqqətlə baxın və bir yerə qeyd edin ki, **config\_dir**-də registry key **C:\Program Files\OpenVPN\config-dir**:



3. **allow\_service**-in **registry key**-ni **1** edin. Həmçinin **log\_dir**-in registry key-ni bir yərə qeyd edin ki, **C:\Program Files\OpenVPN\log**-dur.
4. Sonra **HKEY\_LOCAL\_MACHINE\SOFTWARE\OpenVPN** register ünvanını yoxlanış edin və əmin olun ki, **config\_dir** və **log\_dir** registry key-ləri öncə OpenVPN GUI-də olduğu kimidir:



5. Sonra regisry editor-u bağlayın.
6. **basic-udp-client.ovpn** quraşdırma faylını **example10-7.ovpn** quraşdırma faylına nüsxələyin və **example10-7.ovpn** quraşdırma faylının içində aşağıdakı dəyişiklikləri edin:

Aşağıdakıları:

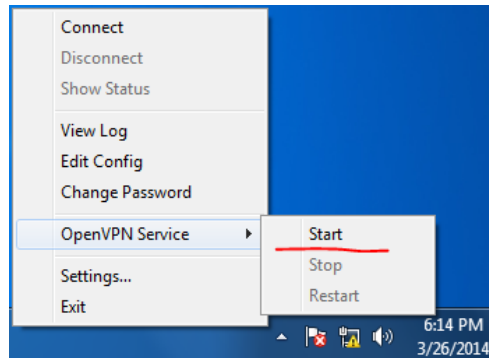
```
cert "c:/program files/openvpn/config/openvpnclient2.crt"
key "c:/program files/openvpn/config/openvpnclient2.key"
```

Dəyişiklik aşağıdakı sətirlərə:

```
cert "c:/program files/openvpn/config/openvpnclient1.crt"
key "c:/program files/openvpn/config/openvpnclient1.key"
```

Çünki, **openvpnclient2**-nin sertifikat şifrəsi mövcuddur ancaq, bizə lazımdır ki, servis avtomatik işə düşəndə şifrə tələb edilməsin. Ona görə də **openvpnclient1**-dən istifadə edəcəyik.

7. Bütün **.ovpn** genişlənməli faylları **config** qovluğundan başqa yerə köçürün ki, OpenVPN servisi **1** ədəd quraşdırma faylı görsün və ona qoşula bilsin.
8. Sonra OpenVPN-i sevis kimi işə salın. Müəyyən vaxtdan sonra həm client və həm də serverdə jurnal fayllarına baxa bilərsiniz ki, qoşulma uğurlu olmuşdur.



### **Bu necə işləyir...**

Windows servisi istifadəçi sistemə giriş etməzdən öncə işə düşür. OpenVPN servisi **HKEY\_LOCAL\_MACHINE\SOFTWARE\OpenVPN\config\_dir** registr ünvanında olan açarı axtarış edir.

Bu OpenVPN prosesini həmin qovluqda olan hər bir **.ovpn** genişlənməli fayl üçün işə salır. Hər bir bu fayllar üçün çıxış aşağıdakı göstərilən registry key qovluğunda qeydə alınır:

```
HKEY_LOCAL_MACHINE\SOFTWARE\OpenVPN\log_dir
```

Burda jurnal faylın adı quraşdırma faylının adı ilə başlayır və **.log** genişlənməsi ilə bitir. Bu misalımız üçün quraşdırma faylı **C:\Program Files\OpenVPN\config\example10-7.ovpn** olduğu üçün jurnal faylı da **C:\Program Files\OpenVPN\log\example10-7.log** olacaq.



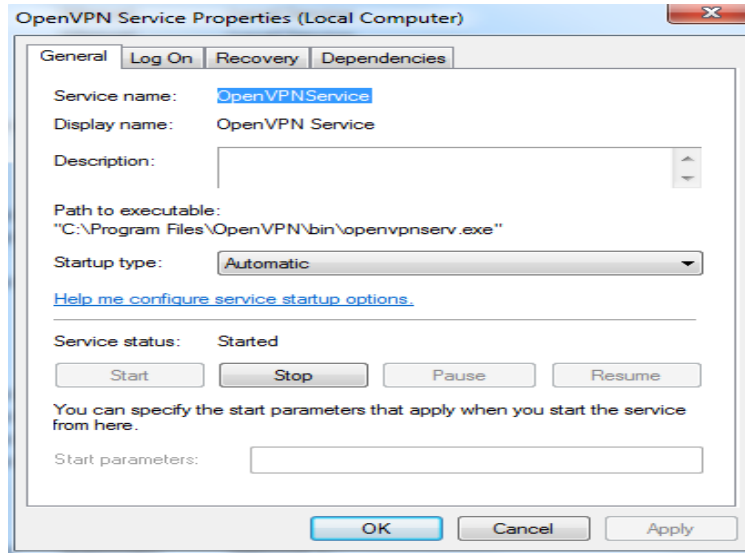
Gördüyümüz kimi, OpenVPN GUI üçün registry-ə sadəcə **allow\_service** üçün 1 əlavə eləməklə OpenVPN GUI-ni servis kimi istifadə elədik.

### Daha da ətraflı...

OpenVPN servisi istifadə elədikdə bəzi önəmli məqamlar var ki, biz nəzərə almalıyıq.

### Avtomatik servis startup

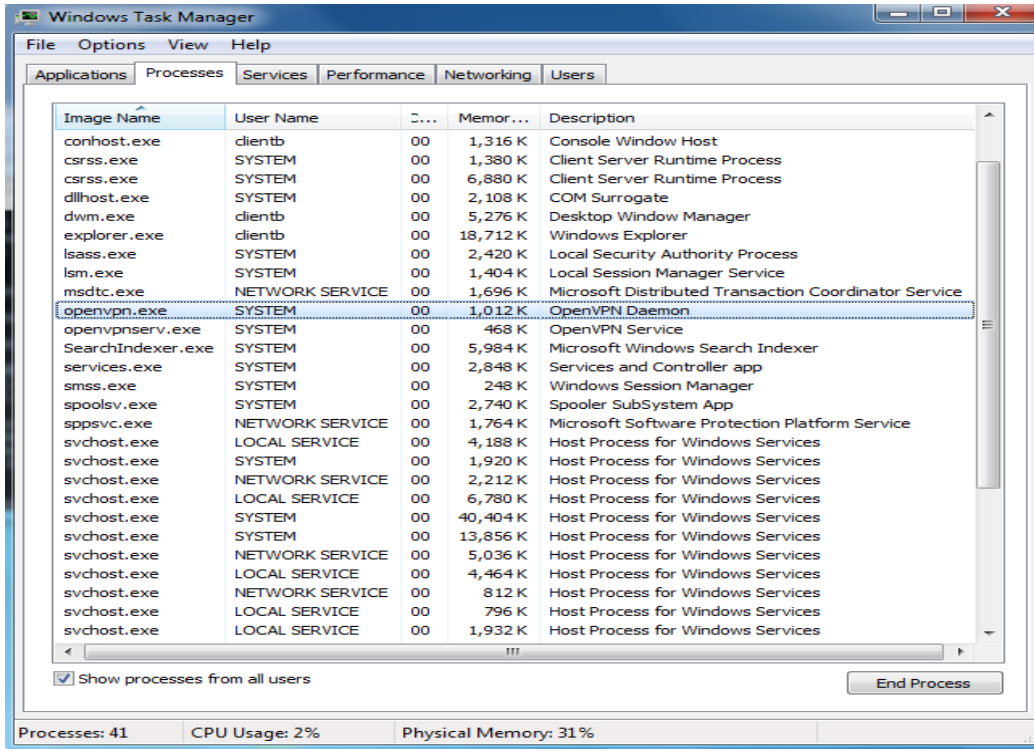
OpenVPN-i sistem startup-ında avtomatik işə salınması üçün administrative control panel olan **Services**-i açmaq lazımdır. **Control Panel | Administrative Tools | Services** ünvanında **OpenVPN Service**-nin üstündə iki dəfə sıxın ki, servisinin **Startup type**-ni **Automatic** edəsiniz.



**OK** düyməsinə sıxın və **Services** administrative control panel-i bağlayın. Windows-u reboot edin və sonra yoxlayıb ki, VPN avtomatik qoşulub ya yox.

### OpenVPN İstifadəçi adı

OpenVPN prosesi işə düşən kimi normal halda **SYSTEM** istifadəçi adından sistemdə proses yaranır. Şəkildə görə bilərik:



Bu halda quraşdırma fayllarında bəzi çatışmazlıqlar yaranır. Əgər **cryptoapicert** direktivi istifadə edilirsə onda sertifikat anbarında olan sertifikatlar **SYSTEM** hesabı adından işləmədiyinə görə yetkilərdə problem çıxacaq. Bu halda sistemə import edilən sertifikat **User certificate** tipi ilə yox **LOCAL MACHINE** kimi yüklənməlidir.

### Həmçinin baxın

- Öncə öyrəndiyimiz Windows: CryptoAPI anbarının istifadə edilməsi hansı ki, Windows-da CryptoAPI anbarının istifadəsinin detallarını açıqlayır.

### Windows: PUBLIC ya da PRIVATE şəbəkə kartları

Windows 7 istifadəsinə başlayanda Microsoft şəbəkə kartlarının klasslara bölünməsinə yeni şərait yaratdı. Şəbəkə kartları PUBLIC və ya PRIVATE aralığının üzvü ola bilərlər. OpenVPN-in istifadə edilməsində bu klasın seçilməsindən ehtiyatlı olmaq lazımdır. Susmaya görə OpenVPN-in TAP-Win32 adapteri PUBLIC şəbəkədə yerləşdirilir və problemə səbəb olur ki, ümumi fayl resurslarından istifadə edə bilərsiniz. Bu misalda biz göstərəcəyik ki, necə şəbəkə tipini dəyişməklə VPN qoşulma üzərindən file sharing-ə izin veriləcək. Ancaq bu misalın OpenVPN-in özü ilə demək olar ki, heç bir əlaqəsi yoxdur.

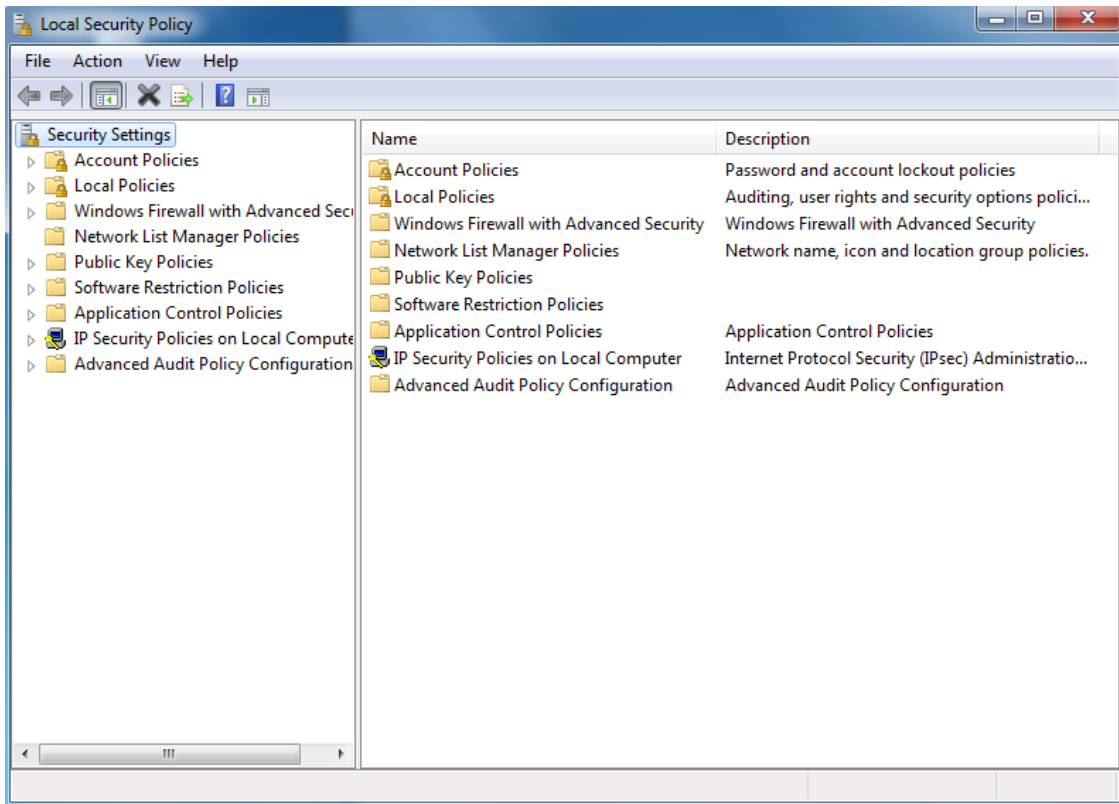
### İşə hazırlaşaq

2-ci başlıqda yaratdığımız client və server sertifikatlarını burada da istifadə edəcəyik. Bu misalda server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Client maşını isə Windows7 x64 OpenVPN2.3-də olacaq. Server üçün 2-ci başlıqda server tərəf routing üçün yaratdığımız **basic-udp-server.conf**

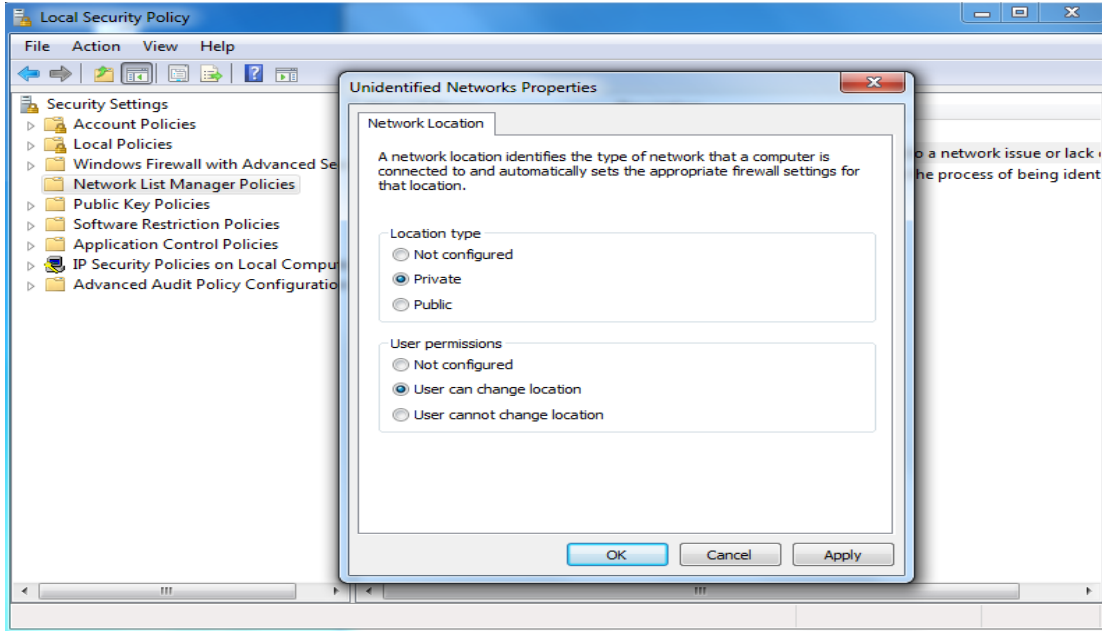
quraşdırma faylından istifadə edəcəyik. Client üçün də həmçinin 2-ci başlıqda yaratdığımız **basic-udp-client.ovpn** quraşdırma faylından istifadə edəcəyik.

### Necə edək...

1. Serveri işə salaq:  
root@siteA:/usr/local/etc/openvpn # **openvpn --config basic-udp-server.conf**
2. Windows7 maşında OpenVPN GUI proqramını maksimal yetkilərlə işə salın və clienti işə salın.
3. VPN qoşulması uğurlu olduğdan sonra **Control Panel\All Control Panel Items\Administrative Tools** ünvanından **Local Security Policy**-ni şəkildə görüldüyü kimi açın:



4. Sonra **Network List Manager Policies**-ə daxil oluruq və bizim TAP alətinə aid olan şəbəkə kartını seçirik. Yeni şəkildə göstərilən kimi, **Unidentified Networks**-un üstündə iki dəfə mousun sol düyməsini sıxırıq.
5. Açılan pəncərədə **Location type** olaraq **Private** və **User permissions**-da **User can change location** seçirik:



6. Sonda **OK** düyməsini sıxırıq və bütün pəncərələri bağlayırıq.

### **Bu necə işləyir...**

Windows7,8 maşınlarında şəbəkə tipinin fərqli yetkiləri mövcuddur. Aşağı şəbəkə yetkilərinə malik olan şəbəkə tipi **PUBLIC**-dir hansı ki, proqramların TCP/IP qoşulmalarına izin verir ancaq, **PRIVATE** şəbəkələrində olan resurslara yetkiyə izin vermir (Local DISK, printer və.s). OpenVPN client ilə eyni şəbəkədə olan resursu paylaşsanız adi halda bu problemə çevriləcək. OpenVPN şəbəkə kartının tipini **Private** eləməklə siz bu problem həll etmiş olacaqsınız.

### **Həmçinin baxın**

- Öncə həll etdiyimiz misalda Windows7: yetkilərin artırılması hansı ki, OpenVPN GUI-nin artırılmış yetkilərlə işə salınmasının detallarını açıqlayır.

### **Windows: routing metodları**

VPN server öz clientinə əlavə ediləcək route cədvəlini yolladıqda, iki üsul vardır ki, onları client-in routing cədvəlinə əlavə edəsiniz:

- **IPAPI helper** funksiyalarından istifadə eləməklə (susmaya görə)
- **ROUTE.exe** proqramı istifadə eləməklə

Əksər hallarda **IPAPI** metodları əla işləyir ancaq, bəzi hallarda tələbat onun imkanlarını aşır. Bu misalda onun işləməsini göstərəcəyik və client-in jurnal faylında hansı düzgün metodun seçilməsini göstərəcəyik

### **İşə hazırlaşaq**

2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Client maşını isə Windows7 x64 OpenVPN2.3-də olacaq. Server üçün 2-ci başlıqda server tərəf routing üçün yaratdığımız **basic-udp-server.conf** quraşdırma faylından istifadə edəcəyik. Client üzündə həmçinin 2-ci başlıqda yaratdığımız **basic-udp-client.ovpn** quraşdırma faylından istifadə edəcəyik.

### Necə edək...

1. Serveri işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-server.conf
```

2. Client maşında **basic-udp-client.ovpn** quraşdırma faylını **example10-9.ovpn** quraşdırma faylına nüsxələyin və **example10-9.ovpn** faylının sonuna aşağıdakı sətirləri əlavə edin:

```
verb 5
route-method ipapi
```

3. OpenVPN client-i işə salın.

4. Və VPN qoşulması uğurlu olduqdan sonra, VPN GUI-nin üstündə **Show status** düyməsinə sıxıb jurnalların son sətirinə baxın. Aşağıdakı sətir sizdə də olmalıdır:

```
Thu Mar 27 12:43:54 2014 C:\Windows\system32\route.exe ADD 10.198.0.0
MASK 255.255.0.0 192.168.200.1
Thu Mar 27 12:43:54 2014 ROUTE: CreateIpForwardEntry succeeded with
dwForwardMetric1=30 and dwForwardType=4
Thu Mar 27 12:43:54 2014 Route addition via IPAPI succeeded
Thu Mar 27 12:43:54 2014 Initialization Sequence Completed
```

Gördüyünüz kimi route metod olaraq IPAPI istifadə edilib və Windows-un **route.exe** əmrinə müraciət edilmişdir.

5. Sonra isə client-in **example10-9.ovpn** faylında dəyişiklik edərək route metod olaraq exe seçin. Aşağıda kimi:

```
verb 5
route-method exe
```

6. OpenVPN clienti yenidən işə salın.

7. Qoşulma bitdikdən sonra jurnalı yenidən yoxlayın. Artıq qoşulma tipi olaraq IPAPI çıxmayacaq və jurnallar aşağıdakı şəkildə olacaq:

```
Thu Mar 27 12:54:54 2014 C:\Windows\system32\route.exe ADD 10.198.0.0 MASK
255.255.0.0 192.168.200.1
Thu Mar 27 12:54:54 2014 env_block: add
PATH=C:\Windows\System32;C:\WINDOWS;C:\WINDOWS\System32\Wbem
Thu Mar 27 12:54:54 2014 Initialization Sequence Completed
```

### Bu necə işləyir...

**route-method** direktivinin 3 opsiyası vardır:

- **adaptive**: İlk olaraq, IPAPI metoda müraciət edir. Əgər IPAPI-dən səhv cavab qayıdırsa, route.exe-yə müraciət edəcək.

- **ipapi:** Hər bir halda routinglərin cədvələ əlavə edilməsi üçün IPAPI helper-dən istifadə edilir.
- **exe:** Hər bir halda external route.exe-dən istifadə edilir.

Əksər hallarda bütün istifadəçilər bildirirlər ki, route-method tipi exe olanda OpenVPN2.1-də başlayaraq problemsiz işləyir. Qeyd edin ki, əgər OpenVPN Windows-un route cədvəlinə route əlavə edə bilmirsə, o qoşulmanı kəsməyəcək. Hal-hazırkı OpenVPN GUI bunun səhv olmasını da belə göstərmir və hər bir halda yaşıl rəngdə olacaq.

#### **Daha da ətraflı...**

OpenVPN susmaya görə **C:\WINDOWS\system32** ünvanında olan **route.exe** proqramına müraciət edir. Əgər windows fərqli qovluğa yüklənibsə, **win-sys** direktivindən istifadə edə bilərsiniz. **win-sys** direktivinin iki opsiyası vardır:

- Deyək ki, Windows OS-un yükləndiyi ünvan D:\Windows ünvanıdır.
- Spesifik opsiya **env** sayəsində OpenVPN client mühit dəyişəni istifadə edəcək hansı ki, **windir** dəyişəni ilə Window OS-un ünvanını təyin edir. Bu mühit dəyişəni adi halda istənilən Windows üzərində olur.

## BÖLÜM 11

### Genişlənmiş quraşdırma

Bu başlıqda biz aşağıdakıları açıqlayacağıq:

- Quraşdırma fayllarınının quraşdırma fayllarına include(əlavə) edilməsi
- Multiple remote və remote-random
- ifconfig-pool-persist detalları
- SOCKS proxy istifadə edərək qoşulma
- HTTP proxy istifadə edərək qoşulma
- autentifikasiyası olan HTTP proxy ilə qoşulma
- dyndns-in istifadə edilməsi
- IP daha az olan quruluşlar(**ifconfig-noexec**)

#### **Giriş**

Bu başlıqda olan ilk və növbəti misallar OpenVPN-in genişlənmiş quraşdırmasını açıqlayacaq. Bu başlıq əksər hallarda OpenVPN-də görünməyən quraşdırmalara əsaslanır. Bu başlıqda siz DYNDNS-in və Proxy-nin istifadə edilməsi ilə VPN-ə qoşulma quraşdırmalarını ətraflı şəkildə öyrənəcəyik.

### **Quraşdırma fayllarının quraşdırma fayllarına include(əlavə) edilməsi**

OpenVPN-in az tanınmış imkanlarından biridə odur ki, bir quraşdırma faylının içinə bir neçə əlavə quraşdırma fayllarını artırmaq olar. Bu daha çətin OpenVPN serverin quraşdırılmasında lazım olur hansı ki, eyni vaxtda bir neçə OpenVPN servisi işləyəcək. Global quraşdırma direktivləri bir faylda saxlana bilər ancaq, qoşulma detallarını açıqlayan quraşdırmalar hərəsi ayrı-ayrı fayllarda saxlana bilər. Bu misalda biz OpenVPN-i müxtəlif servislərdə(instance) işlədəcəyik. Eyni vaxtda UDP istifadə edilməsi üçün və TCP istifadə edilməsi üçün quraşdırılacaq.

Ancaq nəzərə alın ki, fərqli OpenVPN instance-lar arasında VPN IP aralığının sharing-inə izin verilmir.

### **İşə hazırlaşaq**

2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalımızda da həmişə olduğu kimi, server maşınımız FreeBSD9.2 x64 OpenVPN2.3-də olacaq.

### **Necə edək...**

1. **example11-1-common.conf** adlı **Global** quraşdırma faylını yaradaq və içinə aşağıdakı sətirləri əlavə edək:

```
dev tun

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnserver.crt
key /usr/local/etc/openvpn/openvpnserver.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0
```

```
persist-key
persist-tun
keepalive 10 60
```

```
push "route 10.198.0.0 255.255.0.0"
topology subnet
```

```
user nobody
group nobody
```

```
daemon
```

Gördüyünüz kimi, bu quraşdırma faylında protocol təyin etmə sətiri və ya server direktivi spesifikasiyası yox idi.

2. UDP bazalı qoşulmalar üçün **example11-1-server1.conf** adlı server quraşdırma faylını yaradaq:

```
config example11-1-common.conf

proto udp
port 1194
server 192.168.100.0 255.255.255.0
```



```
log-append /var/log/openvpn-udp.log
```

3. Həmçinin TCP bazalı qoşulmalar üçün **example11-1-server2.conf** adlı server quraşdırma faylını yaradaq:  
**config example11-1-common.conf**

```
proto tcp
port 443
server 192.168.200.0 255.255.255.0
```

```
log-append /var/log/openvpn-tcp.log
```

Gördüyümüz kimi burda 443(https) portu ona görə istifadə edilmişdir ki, bezi firewall-lar bu portu bağlamır.

4. Hər iki serveri işə salaq:  
root@siteA:/usr/local/etc/openvpn # **openvpn --config example11-1-server1.conf**  
root@siteA:/usr/local/etc/openvpn # **openvpn --config example11-1-server2.conf**

**/var/log/openvpn-tcp.log** və **/var/log/openvpn-udp.log** jurnal fayllarını analiz edin və əmin olun ki, serverlər həqiqətən də işləyirlər.

### **Bu necə işləyir...**

OpenVPN quraşdırma faylları uyğun olaraq CLI-dan daxil edilən opsiyalarla eynidir. Yəni ki, siz CLI-dan **-config** əmri ilə bir neçə ayrı OpenVPN prosesini işə saldığınız kimi də, uyğun olaraq bir neçə OpenVPN quraşdırma faylını birdən işlədə bilərsiniz. Bu sizə şərait yaradır ki, hər kəsə aid olan ümumi quraşdırma direktivlərini ayrıca faylda və hər spesifik VPN istifadəçisinə aid olan direktivləri ayrı faylda saxlaya bilərsiniz. Gördüyünüz kimi, server direktivi daha kiçik quraşdırma fayllarında saxlana bilər və sizin işinizi daha da rahatlaşdırır. Bu ən çox böyük həcmli VPN serverlər üçün istifadə edilir.

OpenVPN-in özünün müdafiə sistemi vardır ki, eyni quraşdırma faylını rekursiv olaraq oxumasın.

### **Multiple remote və remote-random**

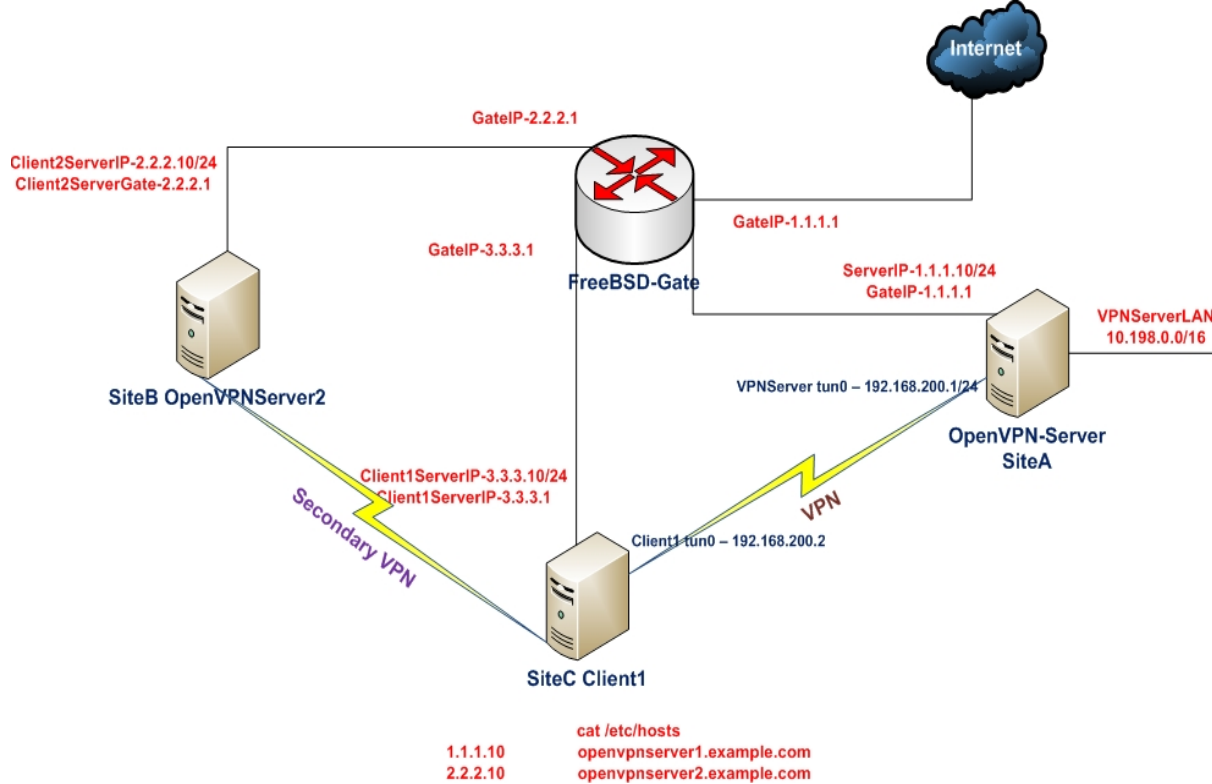
OpenVPN-in eyni zamanda avtomatik failover və load-balancing-i dəstəkləyir. Yəni ki, əgər bir OpenVPN serverə qoşulma kəsilsə, onda növbəti OpenVPN server avtomatik olaraq işləyəcək. **remote-random** direktivi sayəsində isə çoxlu OpenVPN clienti, bir neçə OpenVPN server arasında bölüşdürmək üçün istifadə edilir. Bu misalda biz 2 ədəd OpenVPN server quraşdıracağıq və sonra **remote-random** direktivini istifadə edəcəyik ki, o iki serverdən biri ilə həmişə işləyə bilsin.

Qeyd edin ki, OpenVPN transparent keçidi dəstəkləmir. Yəni bir server üzərində olan real işlək qoşulmaları avtomatik rejimdə digərinə miqrasiya

etmir. Transparent failoveri VPN qoşulmasında etmək çox çətindir ona görə ki, təhlükəsizlik sessiya açarları bir serverdən digərinə miqrasiya edilməlidir. Hal-hazırda bunu OpenVPN ilə eləmək mümkün deyil.

## İşə hazırlaşaq

Biz aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalımızda hər iki server maşını və client maşını üçün FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Hər iki server maşınında 2-ci başlıqda yaratdığımız **basic-udp-server.conf** quraşdırma faylından istifadə edəcəyik.

## Necə edək...

1. Hər iki OpenVPN serverdə NAT quraşdırmasını edək. Hər iki maşında **/etc/rc.conf** faylına aşağıdakı sətirləri əlavə edək və sonra **reboot** edək ki, NAT işə düşsün.

```
natd_enable="YES"
natd_interface="em0"
firewall_enable="YES"
firewall_type="OPEN"
```

2. Hər iki serveri işə salaq:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-server.conf
root@siteB:/usr/local/etc/openvpn # openvpn --config basic-udp-server.conf
```

Hər iki serverin jurnal fayllarını yoxlayın ki, vpn-in uğurlu olmasından əmin olun. Eyni quraşdırmanı hər iki serverdə istifadə edə bilərik.

3. Sonra client maşında **example11-2-client.conf** adlı quraşdırma faylı yaradıb içinə aşağıdakı sətirləri əlavə edək:

```
client
proto udp
remote openvpnserver1.example.com 1194
remote openvpnserver2.example.com 1194
remote-random
dev tun
nobind

ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnclient1.crt
key /usr/local/etc/openvpn/openvpnclient1.key
tls-auth /usr/local/etc/openvpn/ta.key 1

ns-cert-type server
```

Ancaq client maşınının **/etc/hosts** faylına aşağıdakı sətirləri əlavə etməyi unutmayın:

```
1.1.1.10 openvpnserver1.example.com
2.2.2.10 openvpnserver2.example.com
```

4. Sonra clienti işə salın:

```
root@siteC:/usr/local/etc/openvpn # openvpn --config example11-2-
client.conf
```

OpenVPN client təsadüfi olaraq seçilən bir serverə qoşuldu. Qoşulma uğurlu olduqdan sonra işə, jurnal faylında ilk qoşulmanı bəlli etdiyiniz OpenVPN serverin prosesini dayandırın:

```
root@siteA:/usr/local/etc/openvpn # killall openvpn
```

Sonra gözləyin ilk **timeout** bitdikdən sonra işə client ikinci serverə qoşulacaq.

### **Bu necə işləyir...**

OpenVPN client-i işə düşdükdən sonra, qeyd edilmiş **remote-random** işə düşür və öz siyahısında olan VPN serverlərdən birini təsadüfi seçib qoşulur. Əgər bu serverə qoşulma kəsilsə o özündə susmaya görə təyin edilmiş vaxt intervalından sonra qoşulduğu VPN serverdən cavab almazsa, növbəti VPN serverlərdən birinə qoşulacaq. Vaxt aralığını 2-ci başlıqda Server-tərəf routing-də örgəndiyimiz **keepalive** opsiyası ilə təyin edə bilərsiniz.

### **Daha da ətraflı...**

OpenVPN-i failover rejimində quraşdırma edərkən bəzi məqamlar var ki, onları burda açıqlayırıq.

### TCP və UDP bazalı quruluşun birgə istifadə edilməsi

Həmçinin mümkündür ki, TCP və UDP bazalı quruluşun protocol tipini təyin edərək quraşdırasınız:

```
remote openvpnserver1.example.com 1194 udp
remote openvpnserver2.example.com 1194 tcp
```

OpenVPN2.1-dən başlayaraq yeni imkan yarandı hansı ki, **connection blocks** imkanı yaradır. Növbəti başlıqda biz bunu detallı danışacağıq.

### TCP bazalı qoşulmaların üstünlükləri

TCP bazalı qoşulmanın bir üstünlüyü var ki, onu **failover** kombinasiyasında istifadə etmək olur. Əgər client qoşulmuş OpenVPN server-də problem yaranarsa, adətən TCP bazalı qoşulmalar həmin anda da ayrılır.

Bu çox qısa **timeout** period yaradır hansı ki, bundan sonra OpenVPN client yenidən qoşulmağa çalışır. UDP bazalı qoşulmalarda isə, client o qədər də tez anlaya bilmir ki, serverə çatmaq mümkün deyil və **keepalive** müddətini gözləyir.

### Avtomatik olaraq ilk OpenVPN serverə qayıtma

OpenVPN-ə əksər verilən suallardan biridə o olur ki, OpenVPN client-in ilk düşən OpenVPN serverə avtomatik olaraq geriye qaytarmaq olarmı? Yeni OpenVPN client ilk qoşulduğu OpenVPN server dayandıqdan sonra o avtomatik ikinci OpenVPN serverə keçir. Həmçinin lazımdır ki, 1-ci OpenVPN server normal işlək vəziyyətə gətirildikdən sonra da, OpenVPN client avtomatik olaraq ona qayıdış edə bilsin. Hal-hazırkı vaxtda client-in yenidən 1-ci serverə qayıdışı üçün 2-ci serverin prosesini dayandırmaq lazım olur. Biz bunu script ilə edə bilərik ancaq, o yenə də qoşulma tipindən asılıdır. Bu uzaq serverin UP olmamasının təyin edilməsi üçün müəyyən vaxt alır. Client-lərin yenidən 1-ci serverə qısa müddətdə qaytarılmasının ən yaxşı yolu 2-ci serverdə management interfeysin olmasıdır və 2-ci məşində olan client-lərin hamısının sərt olaraq ayrılması ilə siz hamısını bütöv şəkildə 1-ciyə qaytara bilərsiniz.

### Həmçinin baxın

- Server-tərəf routing-ə hansı ki, OpenVPN qoşulmasının əsaslarını açıqlayır.
- 12-ci başlıq, connection block-ların istifadə edilməsi hansı ki, tək client-də bir neçə OpenVPN serverin dəstəklənməsini alternativ üsulunu göstərir.

### **ifconfig-pool-persist** detalları

OpenVPN quraşdırmasında ən çox qarışıqlığa gətirib çıxaran opsiyalardan biridə **ifconfig-pool-persist**-dir. Bu direktivin sayəsində siz client-lər üçün təyin edilmiş IP ünvan siyahısını öncədən tutursunuz və client yenidən qoşulduqda o öncə istifadə etdiyi IP ünvanı yenidən də əldə edəcək. Bu istifadəçi üçün static IP ünvan mənimsətmək üçün 3 usuldan biridir. Digər 2 üsul isə aşağıdakılardır:

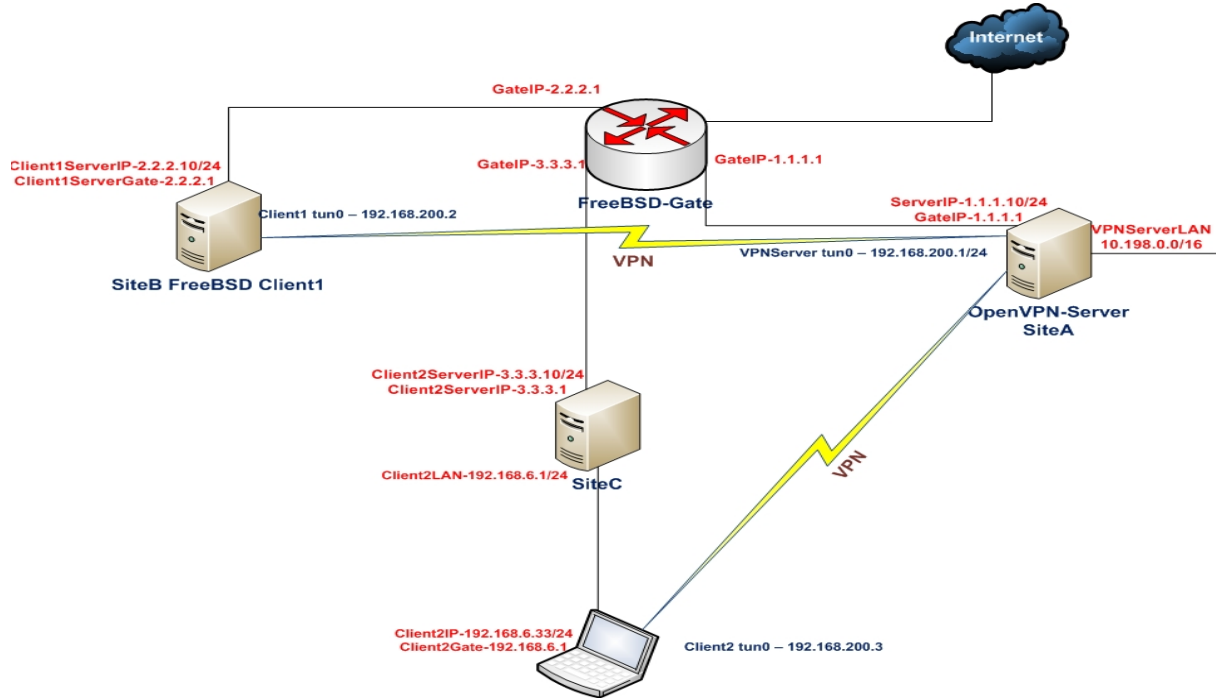
- **client-connect** scriptində **ifconfig-push** istifadə edilməsi
- **client-configuration** faylında **ifconfig-push** istifadə edilməsi

Praktika göst rir ki, OpenVPN quruluşu d zg n iřlem y nd  **ifconfig-pool-persist** faylın istifad sini s nd rm k daha d zg nd r.

Bu misalda biz ifconfig-pool-persist-in istifad sini g st r c y k v  orda hansı t l l rin olduėunu a ıqlayacaıyıq.

## İř  hazırlařaq

Ařaėıdaki ř b k  quruluşundan istifadə ed c y k:



2-ci b şlıqda yaratdıėımız client v  server sertifikatlarını burda da istifadə ed c y k. Bu b şlıqda server mařını FreeBSD9.2 x64 OpenVPN2.3-d  olacaq v  qurařdırma faylı olaraq 2-ci b şlıqda yaratdıėımız **basic-udp-server.conf** faylından istifadə ed c y k. Client mařının biri FreeBSD9.2 x64 OpenVPN2.3-d  v  qurařdırma faylı olaraq **basic-udp-client.conf** istifadə edil c k. Client mařının dig ri is  Windows7 x64 OpenVPN2.3-d  v  qurařdırma faylı is  **basic-udp-client.ovpn** olacaq.

## Nec  ed k...

1. Server  c n qurařdırma faylı yaradaq. **basic-udp-server.conf** qurařdırma faylını **example11-3-server.conf** faylına n sx l y k v  **example11-3-server.conf** faylının sonuna ařaėıdaki s tiri  lav  ed k:  
**ifconfig-pool-persist /usr/local/etc/openvpn/ipp.txt**

2. Sonra serveri iř  salaq:  

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example11-3-server.conf
```

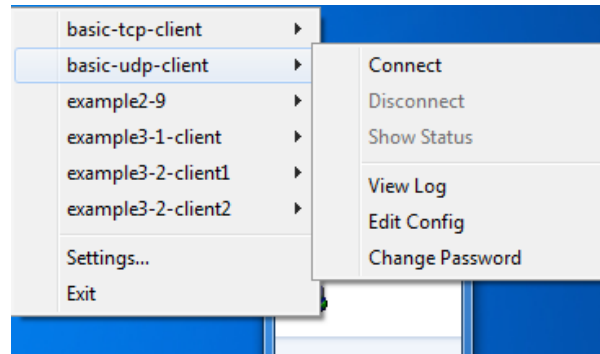
Server iř  d ř n kimi, **/usr/local/etc/openvpn/ipp.txt** adlı boř fayl yaranacaq.

3. FreeBSD clienti işə salaq(SiteB /etc/hosts faylında 1.1.1.10 **openvpnserver.example.com** yazmağı unutmayın):  
 root@siteB:/usr/local/etc/openvpn # **openvpn --config basic-udp-client.conf**

Normalda, clientə server direktivi aralığında təyin edilən IP-dən ilk mümkün ola bilən IP **192.168.200.2** mənimsədiləcək.

4. Client və serveri dayandırın və serverdə faylının içinə baxın:  
 root@siteA:/usr/local/etc/openvpn # **cat ipp.txt**  
**openvpnclient1,192.168.200.2**

5. Sonra OpenVPN serveri yenidən işə salın və Windows7 clienti işə salın ancaq, fərqli sertifikat ilə(Misal üçün: **openvpnclient2**):



6. Artıq müştəriyə **192.168.200.3** IP ünvanı mənimsədiləcək. Ancaq, **ifconfig-pool-persist** opsiyası olmasaydı client-ə ilk mövcud olan IP yeni **192.168.200.2** mənimsədiləcəkdi.

### **Bu necə işləyir...**

OpenVPN server işə düşəndə o, ilk öncə **ipp.txt** faylını oxuyur və çalışır ki, faylda tapdığı sertifikatlarla əsaslanaraq IP ünvanı yenidən həmin client-ə mənimsətsin. Beləliklə hər dəfə OpenVPN client yenidən qoşulduqda o **ipp.txt** faylında onun sertifikatının common name-inə aid olan IP ünvan taparsa, yenidən həmin client-ə eyni IP ünvanı qaytaracaq.

Qoşulan ilk client ala biləcəyi ilk IP ünvanı serverdən alır yeni 192.168.200.2. OpenVPN server dayandırıldıqdan sonra isə, client haqqında olan informasiya **ipp.txt** faylına yazılır. İkinci dəfə OpenVPN server işə düşdükdə bu informasiya yenidən oxunur və **192.168.200.2** IP ünvanı öncəki **openvpnclient1** common name ilə olan client üçün rezerv edildi. Növbəti **openvpnclient2** common name ilə qoşulan client isə növbəti mövcud olan **192.168.200.3** IP ünvanı aldı. OpenVPN Server yenidən dayandırıldıqda **ipp.txt** faylında **openvpnclient2** common name olan client haqqında da məlumatlar əlavə edildi. Bu o deməkdir ki, artıq OpenVPN serverimiz dayansa və yenidən işə düşsədə həmişə eyni olaraq **openvpnclient1** üçün **192.168.200.2** və **openvpnclient2** üçün isə **192.168.200.3** IP ünvanı mənimsədəcək. Ancaq çoxlu OpenVPN clientlər olduğu halda elə hallar olur ki, hər kəsə dəqiq IP ünvanlar mənimsədilmir.

Əksər hallarda VPN IP ünvanı bitməyəndə bu tip problemlər olmur. Tam təminat üçün isə **client-config-dir** direktivindən istifadə etsəniz daha yaxşı olar.

### Daha da ətraflı...

ifconfig-pool-persist direktivinin istifadəsində bəzi məqamlar vardır ki, onları mütləq nəzərə almalıyıq.

### Yenilənmə intervalının təyin edilməsi

Biz vaxt intervalı təyin eləmədiyimiz üçün **ipp.txt** faylı susmaya görə hər **600(10 dəq)** saniyədən bir özünü yeniləyəcək. Bunu siz həmçinin **ipp.txt** faylında görə bilərsiniz ki, client tez müddətdə çıxış eləsə **ipp.txt** faylının için yenilənməyəcək. Bu ona görə ki, ilk yenilənmə müddəti sona çatmayıb ya da OpenVPN serverin prosesi özü heç dayanmayıb.

Həmçinin yenilənmə müddətinin intervalını **0** təyin edə bilərsiniz hansı ki, **ipp.txt** faylının heç bir zaman yenilənməməsini deyir. Bu ondan ötürüdür ki, OpenVPN server işə düşən kimi yalnız **ipp.txt** faylında olan IP ünvanlarla clientləri sət olaraq qeydə alsın ancaq, server işə düşdükdən sonra bu fayl heç bir zaman yenilənməyəcək.

### Ehtiyat: duplicate-cn opsiyasından qorunma

**duplicate-cn** opsiyası ilə eyni client sertifikatının eyni anda bir neçə yerdən qoşulmasına izin verir. Əgər bu opsiya **istifadə edilirsə**, **ifconfig-pool-persist** opsiyası **yararsız olur** ona görə ki, eyni sertifikat ilə bir neçə dəfə qoşulma imkanı olur. Bu halda hər bir Common Name olan sertifikat üçün fərqli IP ünvan verilməsinə gerek olur və **ipp.txt** faylı yararsız olur.

### 'topology net30' istifadə ediləndə

Əgər server opsiyası olan **topology net30** istifadə edilirsə (hansı ki, susmaya görə OpenVPN2.0 üçündür), **ipp.txt** faylının formatı biraz dəyişir. **net30** topologiyası rejimində hər bir clientə **/30** şəbəkəsi mənimsədir və o da öz növbəsində 4 IP ünvanının istifadə edilməsi anlamına gəlir. Network address, VPN serverin son nöqtə ünvanı, real VPN IP ünvanı və **/30** şəbəkəsi üçün broadcast ünvan. Aşağıda **ipp.txt** faylını bunların ilk ikisi üçün yazmışıq:

```
openvpnclient1,192.168.200.4
openvpnclient2,192.168.200.8
```

### **SOCKS proxy istifadə edərək qoşulma**

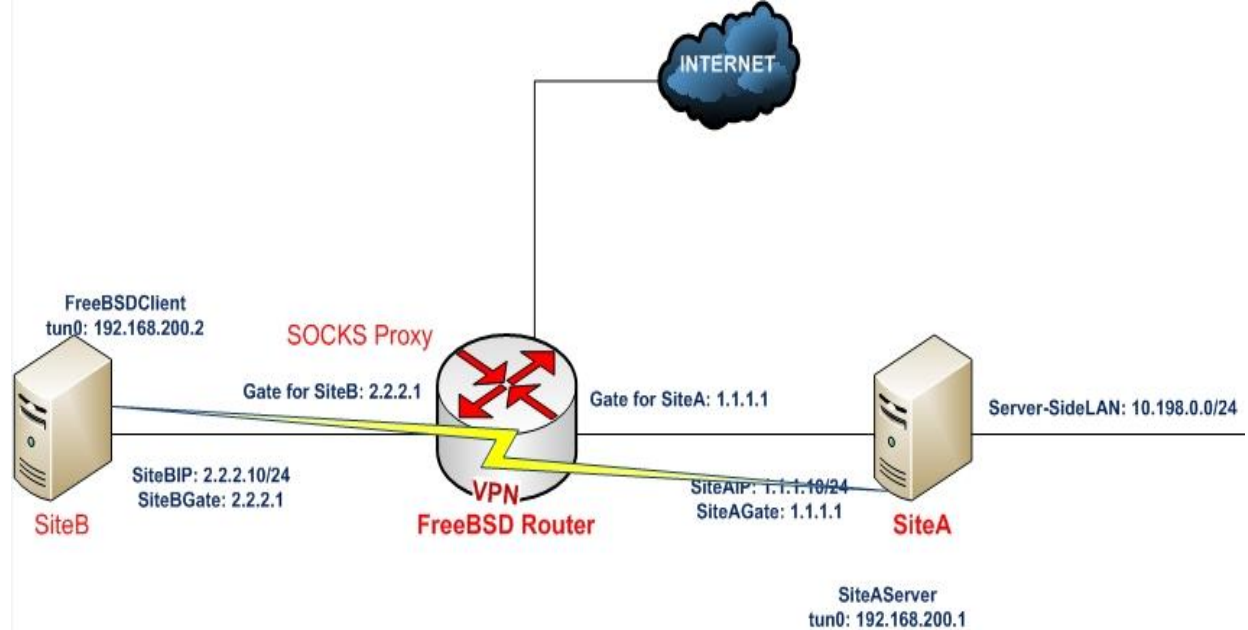
Elə hallar olur ki, client-i UDP trafik ilə qoşmaq mümkün olmur çünki yolda UDP trafiki kəsən Firewall olur. Bu hallarda arada olan Proxy maşınlar sayəsində VPN-ə qoşulmağa gerek olur. OpenVPN SOCKS və HTTP proxy tiplərinə qoşulmanı dəstəkləyir. Bu qoşulmaların hər ikisi TCP ilə işləyir. Bu misalımızda SOCKS proxy serverə qoşulmanı və növbətisini isə HTTP proxy serverə qoşulmanı açıqlayacağıq. Bu qoşulmaların hər ikisində autentifikasiya olmadan işləyir.

SOCKS proxy-ini çox asan yolla istənilən SSH server olan serverin üzərindən eləmək olar. Məsələn istənilən LINUX/UNIX maşının üzərindən eləmək olar. Yeni

həmçinin Windows Client maşının özündən SSH client PUTTY ilə qoşulub SOCKS proxy yarada bilərik. Yada əgər client UNIX maşın olarsa, onun üzərindən qoşulub eləməkdə asan olacaq.

## İşə hazırlaşaq

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda server və client maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Server üçün 9-cü başlıqda TCP bazalı qoşulmaların təkmilləşdirilməsi üçün yaratdığımız **example9-7-server.conf** quraşdırma faylından istifadə edəcəyik. Client üçün də həmçinin 2-ci başlıqda yaratdığımız **basic-tcp-client.conf** quraşdırma faylından istifadə edəcəyik. Client maşının **/etc/hosts** faylında **1.1.1.10 openvpnserver.example.com** yazmağı unutmayın. Ancaq yuxarıdakı şəkildə gördüyünüz **FreeBSD Router** maşının **/etc/hosts** faylına da mütləq VPN serverin adını əlavə etmək lazımdır çünki, bizim halda FreeBSD server Socks proxy server olacaq. Yeni client öz quraşdırmasında **remote** direktivində **openvpnserver.example.com** yazdığına görə socks proxy server bu adı özündə tanımasa yönləndirmə işini görməyəcək.

**FreeBSD Router** maşınının **/etc/hosts** faylına aşağıdakı sətiri əlavə edək:

```
root@vpngate:~ # cat /etc/hosts
127.0.0.1 localhost
1.1.1.10 openvpnserver.example.com
```

## Necə edək...

1. Serveri işə salaq:  

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example9-7-server.conf
```
2. **basic-tcp-client.conf** quraşdırma faylını **example11-4-client.conf** quraşdırma faylına nüsxələyin və **example11-4-client.conf** faylın sonuna aşağıdakı sətiri əlavə edin:  

```
socks-proxy 127.0.0.1 1080
```



3. Client maşında SSH qoşulması ilə SOCKS proxy yaradaq:  
root@siteC:/usr/local/etc/openvpn # **ssh -D 1080 3.3.3.1**
  
4. Clientin digər terminal pəncərəsində isə qoşulmanı edin:  
root@siteC:/usr/local/etc/openvpn # **openvpn --config example11-4-client.conf**  
Sat Mar 29 15:28:07 2014 OpenVPN 2.3.2 amd64-portbld-freebsd9.2 [SSL (OpenSSL)] [LZO] [eurephia] [MH] [IPv6] built on Jan 12 2014  
Sat Mar 29 15:28:07 2014 Control Channel Authentication: using '/usr/local/etc/openvpn/ta.key' as a OpenVPN static key file  
Sat Mar 29 15:28:07 2014 Attempting to establish TCP connection with [AF\_INET]127.0.0.1:1080 [nonblock]  
Sat Mar 29 15:28:08 2014 TCP connection established with [AF\_INET]127.0.0.1:1080  
Sat Mar 29 15:28:08 2014 TCPv4\_CLIENT link local: [undef]  
Sat Mar 29 15:28:08 2014 TCPv4\_CLIENT link remote: [AF\_INET]127.0.0.1:1080  
Sat Mar 29 15:28:08 2014 [openvpnsrver] Peer Connection Initiated with [AF\_INET]127.0.0.1:1080  
Sat Mar 29 15:28:10 2014 TUN/TAP device /dev/tun0 opened  
Sat Mar 29 15:28:10 2014 do\_ifconfig, tt->ipv6=0, tt->did\_ifconfig\_ipv6\_setup=0  
Sat Mar 29 15:28:10 2014 /sbin/ifconfig tun0 192.168.200.2 192.168.200.2 mtu 1500 netmask 255.255.255.0 up  
add net 192.168.200.0: gateway 192.168.200.2  
Sat Mar 29 15:28:10 2014 Initialization Sequence Completed

Gördüyümüz kimi client ilk olaraq 127.0.0.1 IP və 1080-ci porta qoşulur. Məhz bundan sonra OpenVPN serverə qoşulma baş verir.

### **Bu necə işləyir...**

SOCKS proxy server OpenVPN client və OpenVPN server arasında aralıq qoşulma rolunu oynayır. Əksər WEB browserlərdə SOCKS qoşulmasını dəstəkləyir. Client öncə SOCKS proxy serverə qoşulma edir və sonra SOCKS proxy server üzərindən OpenVPN serverə qoşulma edir. Əgər bu qoşulmaya SOCKS server izin verirsə VPN sessiya yaranacaq.

### **Daha da ətraflı...**

Proxy host-un VPN qoşulmasında istifadə edilməsindən öncə bəzi məqamlar vardır ki, biz diqqətə almalıyıq.

### **Davamiyyət**

Əksər hallarda proxy host-lara qoşulmalarda şəbəkə sürətində gecikmə olur. Yeni əgər bir neçə proxy host üzərindən keçid edirsinizsə, nezərə alın ki, şəbəkəniz xeyli kiçilə bilər.

### **SSH üzərindən keçən SOCKS proxy-lər Qeyd #1**

OpenVPN qoşulması üçün SOCKS proxy-nin qurulmasında SSH çox rahat alət ola bilər. OpenVPN server özü onsuzda şəbəkəni şifrələyir. Ancaq SOCKS proxy SSH

üzərindən olduğuna görə də şəbəkə ilk dəfədə SSH ilə şifrələnir və sonra OpenVPN tərəfindən şifrələnir. Bu tip qoşulma davamiyyəti aşağı salır.

### SSH üzərindən keçən SOCKS proxy-lər Qeyd #2

OpenVPN2.2-dən başlayaraq SOCKS proxy qoşulmasına həmçinin autentifikasiyada əlavə edilib. SOCKS özü autentifikasiyada login və şifrəni açıq şəkildə yollasa belə yenə də təhlükəsiz olacaq çünki, SOCKS özü SSH tunel üzərindən şifrələnərək keçir.

### Həmçinin baxın

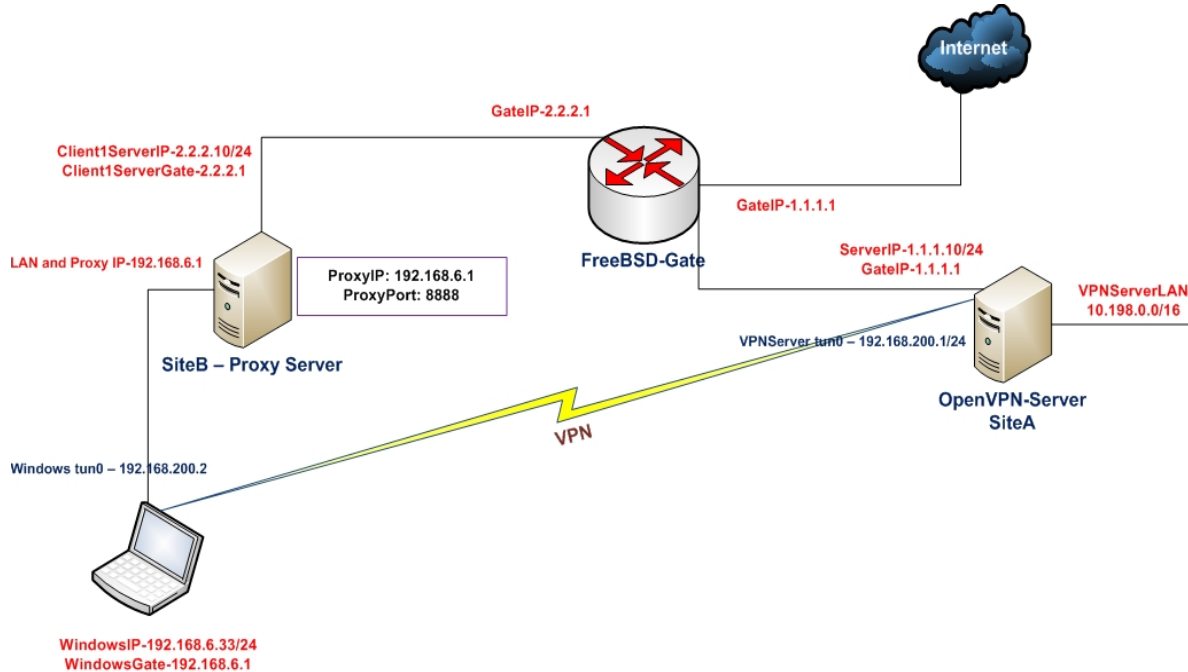
- Növbəti iki misalda HTTP proxy-nin istifadə qaydaları.

### HTTP proxy istifadə edərək qoşulma

Bu misalımızda biz OpenVPN serverə qoşulmanı HTTP proxy üzərindən edəcəyik. Misalda istifadə edəcəyimiz HTTP proxy UNIX maşınlarında əksər istifadə edilən apache WEB serverin **mod\_proxy** modulundan istifadə ediləcək. Bu modulu **CONNECT** müraciətlərin qəbulu kimi də istifadə edə bilərik. **CONNECT** metodlu qoşulma tipi təhlükəsiz WEB server tələb edir (yəni HTTPS) və bu da OpenVPN serverdir. Əgər **CONNECT** müraciətinə izin verilməyibsə, onda OpenVPN qoşulmasında HTTP proxy istifadə edilə bilməyəcək.

### İşə hazırlaşaq

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Server üçün 9-cu başlıqda TCP bazalı qoşulmaların təkmilləşdirilməsi



```

x [x] MIME mod_mime
x [x] MIME_MAGIC mod_mime_magic
x [x] NEGOTIATION mod_negotiation
x [x] REWRITE mod_rewrite
x [x] SETENVIF mod_setenvif
x [x] SPELLING mod_spelling
x [x] STATUS mod_status
x [x] UNIQUE_ID mod_unique_id
x [x] USERDIR mod_userdir
x [x] USERTRACK mod_usertrack
x [x] VHOST_ALIAS mod_vhost_alias
x [x] FILTER mod_filter
x [] SUBSTITUTE mod_substitute
x [x] VERSION mod_version
x [x] SSL mod_ssl
x [] SUEXEC mod_suexec
x [] SUEXEC_RSRCLIMIT suEXEC rlimits based on login class
x [] SUEXEC_USERDIR suEXEC UserDir support
x [x] REQTIMEOUT mod_reqtimeout
x [x] PROXY mod_proxy
x [] IPV4_MAPPED Allow IPv6 socket to handle IPv4
x [] BUCKETEER mod_bucketeer
x [] CASE_FILTER mod_case_filter
x [] CASE_FILTER_IN mod_case_filter_in
x [] EXT_FILTER mod_ext_filter
x [] LOG_FORENSIC mod_log_forensic
x [] OPTIONAL_HOOK_EXPORT mod_optional_hook_export
x [] OPTIONAL_HOOK_IMPORT mod_optional_hook_import
x [] OPTIONAL_FN_IMPORT mod_optional_fn_import
x [] OPTIONAL_FN_EXPORT mod_optional_fn_export
x [] PROXY_AJP mod_proxy_ajp
x [x] PROXY_BALANCER mod_proxy_balancer
x [x] PROXY_CONNECT mod_proxy_connect
x [] PROXY_FTP mod_proxy_ftp
x [x] PROXY_HTTP mod_proxy_http
x [x] PROXY_SCGI mod_proxy_scgi
m

```

```

root@siteC:/usr/ports/www/apache22 # make -DBATCH install #
Yükləyək

```

/usr/local/etc/apache22/httpd.conf faylına aşağıdakı sətirləri əlavə edirik:

```

Listen 192.168.6.1:8888
<VirtualHost *:8888>
 ProxyRequests On
 <Proxy *>
 Order deny,allow
 Deny from all
 Allow from all
 </Proxy>

 ProxyVia Off
 <IfModule mod_headers.c>
 Header set P3P "policyref=\"/w3c/p3p.xml\"",
CP=\"NOI DSP COR NID CUR ADM DEV OUR BUS\"
 </IfModule>
 AllowCONNECT 443 1129 1194 7934 8080 993 # Nəzərə
 alın ki,
 1194 əlavə
 edilmişdir
</VirtualHost>

```

```

Startup-a əlavə edirik:
root@siteC:/usr/ports/www/apache22 # echo 'apache22_enable="YES"'
>> /etc/rc.conf
root@siteC:/usr/ports/www/apache22 # /usr/local/etc/rc.d/apache22
start # İşə salırıq

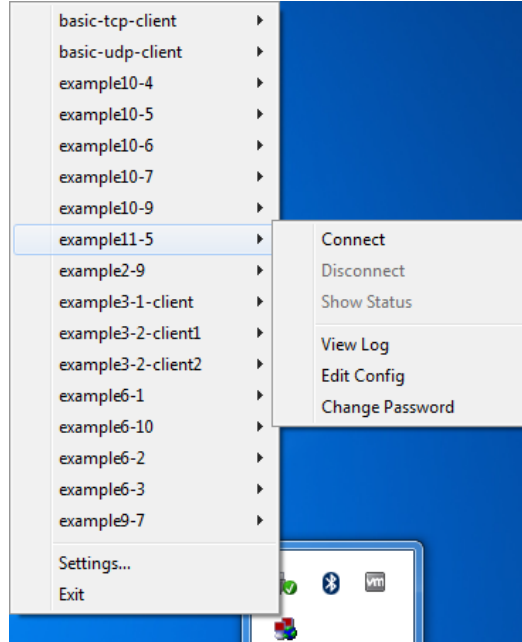
```

3. Client üçün **example9-7.ovpn** quraşdırma faylını **example11-5.ovpn** quraşdırma faylına nüsxələyin və **example11-5.ovpn** faylının sonuna aşağıdakı sətirləri əlavə edin:

```
http-proxy 192.168.6.1 8888
verb 4
```

**192.168.6.1** IP ünvanı client maşınının Gateway-dir və biz apache22 proxy serveri orda qaldırmışıq. HTTP proxy server **8888** portunda işləyir.

4. Client-i işə salaq:



Qoşulma jurnalında görməlisiniz ki, OpenVPN client ilk olaraq HTTP proxy hostuna qoşulur və sonra HTTP '**CONNECT**' metodu ilə müraciətini OpenVPN serverə qoşulmaq üçün göndərir:

```

OpenVPN Connection (example11-5)
Current State: Connected
Sat Mar 29 21:02:32 2014 Expected Remote Options hash (VER=V4): 'c413e92e'
Sat Mar 29 21:02:32 2014 Attempting to establish TCP connection with [AF_INET]192.168.6.1:8888
Sat Mar 29 21:02:32 2014 MANAGEMENT: ->STATE:1396112562,TCP_CONNECT...
Sat Mar 29 21:02:32 2014 TCP connection established with [AF_INET]192.168.6.1:8888
Sat Mar 29 21:02:32 2014 Send to HTTP proxy: 'CONNECT 1.1.1.10:1194 HTTP/1.0'
Sat Mar 29 21:02:32 2014 HTTP proxy returned: 'HTTP/1.0 200 Connection Established'
Sat Mar 29 21:02:34 2014 TCPv4_CLIENT link local: [undef]
Sat Mar 29 21:02:34 2014 TCPv4_CLIENT link remote: [AF_INET]192.168.6.1:8888
Sat Mar 29 21:02:34 2014 MANAGEMENT: ->STATE:1396112564,WAIT...
Sat Mar 29 21:02:34 2014 MANAGEMENT: ->STATE:1396112564,AUTH...
Sat Mar 29 21:02:34 2014 TLS: Initial packet from [AF_INET]192.168.6.1:8888, sid=544b5291d4dec649
Sat Mar 29 21:02:34 2014 VERIFY OK: depth=1, C=NL, O=Cookbook, CN=Cookbook CA, emailAddress=openvpn-ca@atl.az
Sat Mar 29 21:02:34 2014 VERIFY OK: nsCertType=SERVER
Sat Mar 29 21:02:34 2014 VERIFY OK: depth=0, C=NL, O=Cookbook, CN=openvpnsrvr, emailAddress=openvpn-ca@atl.az
Sat Mar 29 21:02:34 2014 Data Channel Encrypt: Cipher 'BF-CBC' initialized with 128 bit key
Sat Mar 29 21:02:34 2014 Data Channel Encrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Sat Mar 29 21:02:34 2014 Data Channel Decrypt: Cipher 'BF-CBC' initialized with 128 bit key
Sat Mar 29 21:02:34 2014 Data Channel Decrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Sat Mar 29 21:02:34 2014 Control Channel: TLSv1, cipher TLSv1/SSLv3 DHE-RSA-AES256-SHA, 2048 bit RSA
Sat Mar 29 21:02:34 2014 [openvpnsrvr] Peer Connection Initiated with [AF_INET]192.168.6.1:8888
Sat Mar 29 21:02:36 2014 MANAGEMENT: ->STATE:1396112566,GET_CONFIG...
Sat Mar 29 21:02:37 2014 SENT CONTROL [openvpnsrvr]: 'PUSH_REQUEST' (status=1)
Sat Mar 29 21:02:37 2014 PUSH: Received control message: 'PUSH_REPLY:route-gateway 192.168.200.1,topology subnet,ping 10,ping-restart 60,socket-flags TCP_NODELAY,ifconfig 192.168.200.2 255.255.255.0'
Sat Mar 29 21:02:37 2014 OPTIONS IMPORT: timers and/or timeouts modified
Sat Mar 29 21:02:37 2014 OPTIONS IMPORT: --socket-flags option modified
Sat Mar 29 21:02:37 2014 OPTIONS IMPORT: --ifconfig-lup options modified
Sat Mar 29 21:02:37 2014 OPTIONS IMPORT: --route-related options modified
Sat Mar 29 21:02:37 2014 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0
Sat Mar 29 21:02:37 2014 MANAGEMENT: ->STATE:1396112567,ASSIGN_IP...:192.168.200.2,
Sat Mar 29 21:02:37 2014 open_tun, tt->ipv6=0
Sat Mar 29 21:02:37 2014 TAP-WIN32 device [Local Area Connection 2] opened: \\.\Global\{B835D57D-F453-48B3-A987-077A7A6E65DC}.tap
Sat Mar 29 21:02:37 2014 TAP-Windows Driver Version 9.9
Sat Mar 29 21:02:37 2014 TAP-Windows MTU=1500
Sat Mar 29 21:02:37 2014 Set TAP-Windows TUN subnet mode network/local/netmask = 192.168.200.0/192.168.200.0/255.255.255.0 [SUCCESSFUL]
Sat Mar 29 21:02:37 2014 Notified TAP-Windows driver to set a DHCP IP/netmask of 192.168.200.2/255.255.255.0 on interface {B835D57D-F453-48B3-A987-077A7A6E65DC} [DHCP-srv: 192.168.200.254, lease-time: 31536000]
Sat Mar 29 21:02:37 2014 Successful ARP Flush on interface [16] {B835D57D-F453-48B3-A987-077A7A6E65DC}
Sat Mar 29 21:02:42 2014 TEST ROUTES: 0/0 succeeded len=0 ret=1 a=0 u/d=up
Sat Mar 29 21:02:42 2014 Initialization Sequence Completed
Sat Mar 29 21:02:42 2014 MANAGEMENT: ->STATE:1396112562,CONNECTED,SUCCESS:192.168.200.2,192.168.6.1

```

Şəkilə gördüyünüz kimi **HTTP** proxy server **200** cavab kodu qaytarmışdır və mənası **OK** deməkdir. Yəni VPN uğurla qoşuldu.

### Bu necə işləyir...

HTTP proxy host-u OpenVPN client və serveri arasında aralıq bir yol rolunu oynayır. HTTP proxy serverləri bütün web browserlər tərəfindən quraşdırıla bilər və əksər korporativ şirkətlərdə yetkilərin quraşdırılması üçün istifadə edilir. Client öz mənəsinə çatmaq üçün öncə HTTP proxy serverə qoşulur və onun üzərindən HTTP **'CONNECT'** metodu istifadə edərək öz mənəsinə çatır. Əgər HTTP proxy serverinin **CONNECT** müraciətinə izin verirsə, **HTTP code 200** qayıdacaq və OpenVPN qoşulması uğurlu olacaq.

### Daha da ətraflı...

HTTP proxy istifadə edərkən bəzi məqamlar vardır ki, onlardan özümüzü qorunmalıyıq:

#### http-proxy options

HTTP proxy host-a qoşulmada OpenVPN quraşdırılmasında bəzi opsiyalar var hansı ki, aşağıda onları açıqlayırıq:

- **http-proxy-timeout [n]**: HTTP proxy host-una qoşulduqda timeout-un [n] saniyələrlə təyin edilməsi. Susmaya görə olan mənası **5** saniyədir.
- **http-proxy-option AGENT [string]**: HTTP proxy hostuna qoşulduqda HTTP agenti [string] təyin elə. Bəzi proxy-lər yalnız tanınmış browserlərə qoşulmağa izin verir.
- **http-proxy-option VERSION 1.1**: HTTP protocol versiyasını 1.1 təyin edin. Susmaya görə HTTP/1.0 olur. OpenVPN2.1-də HTTP/1.1 proxy-ə

qoşulanda uyğun olmur ona görə ki, bəzi browserlər qoşulmanı qəbul etmir. OpenVPN2.2-dən sonra bu problem tamamilə həll edilmişdir.

### Firewall-dan keçid

Nəzərə alın ki, OpenVPN heç vaxt öz trafikini firewall-dan gizlətmir və bunun heç bir mənası belə yoxdur çünki, hal-hazırkı firewall-ların hamısı paketin dərinliyinə qədər analiz edib onun tipini heç bir çətinlik çəkmədən təyin edə bilirlər. Və təyinat olduqdan sonra isə OpenVPN-i bağlamaq heçdə problem yaratmayacaq.

### Davamlılıq

SOCKS proxy-də olduğu kimi eynilərdə HTTP proxy-də həmçinin davamlılığın aşağı düşməsinə səbəb ola bilər.

### Həmçinin baxın

- Bundan öncəki misal SOCKS proxy və növbəti olan HTTP proxy autentifikasiya ilə.

## Authentifikasiya olan HTTP proxy ilə qoşulma

Öncəki misalımızda OpenVPN serverə adi HTTP proxy üzərindən qoşulduğumuz kimi, indidə eyni HTTP proxy server üzərindən qoşulacayıq. Sadəcə bu HTTP proxy-də bizdən giriş üçün istifadəçi adı və şifrə tələb ediləcək (Yeni autentifikasiya).

Bu misalımızda da HTTP proxy olaraq UNIX/Linux bazalı olan, **mod\_proxy** modulu ilə apache22 httpd serverdən istifadə edəcəyik. Authentifikasiya metodu olaraq **Basic** istifadə edəcəyik.

### İşə hazırlaşaq

2-ci başlıqda yaratdığımız client və server sertifikatlarını burdada istifadə edəcəyik. Bu misalda server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Server üçün 9-cu başlıqda TCP bazalı qoşulmaların təkmilləşdirilməsi üçün yaratdığımız **example9-7-server.conf** quraşdırma faylından istifadə edəcəyik. Client maşını isə Windows7 x64 OpenVPN2.3-də olacaq. Client üçündə 9-cu başlıqda TCP bazalı qoşulmaların təkmilləşdirilməsi üçün yaratdığımız **example9-7.ovpn** quraşdırma faylından istifadə edəcəyik.

Proxy maşınının **/etc/hosts** faylında **1.1.1.10** **openvpnsrver.example.com** yazmağı unutmayın. Client maşınının **C:\Windows\System32\Drivers\etc\hosts** faylında **1.1.1.10** **openvpnsrver.example.com** yazmağı unutmayın.

### Necə edək...

1. Serveri işə salın:  
root@siteA:/usr/local/etc/openvpn # **openvpn --config example9-7-server.conf**

2. Basic autentifikasiya metodunun dəstəkləyən HTTP proxy server qurun. Öncəki misalımızda olduğu kimi, apache22-ni **mod\_proxy** və digər modullarla birgə kompilyasiya edib sistmə yükləyin və sonra aşağıdakı quraşdırmaları **/usr/local/etc/apache22/httpd.conf** faylına əlavə edin:

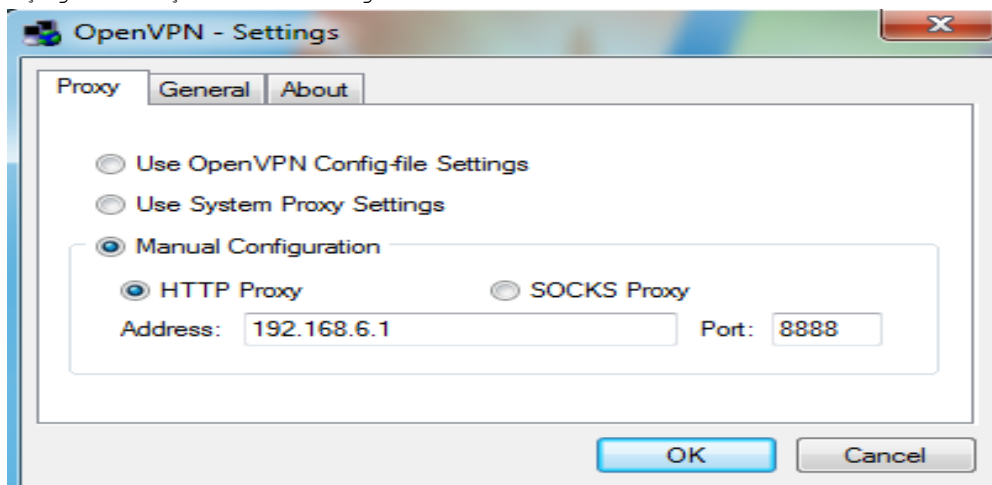
```
<VirtualHost *:8888>
 ProxyRequests On
 ProxyVia On
 KeepAlive On
 <Proxy *>
 Order deny,allow
 Deny from all
 Allow from all
 Require user openvpn # Mütləq tələb edilən
 istifadəçi openvpn-dir

 AuthType Basic
 AuthName "Password Required"
 AuthUserFile /usr/local/etc/apache22/vpnpass-file
 </Proxy>
 <IfModule mod_headers.c>
 Header set P3P "policyref=\"/w3c/p3p.xml\"", CP=\"NOI DSP
COR NID CUR ADM DEV OUR BUS\""
 </IfModule>
 AllowCONNECT 443 1129 1194 7934 8080 993
</VirtualHost>
```

**openvpn** adlı yeni istifadəçini yaradaq və login/parol bazası olaraq **/usr/local/etc/apache22/vpnpass-file** faylından istifadə edək.

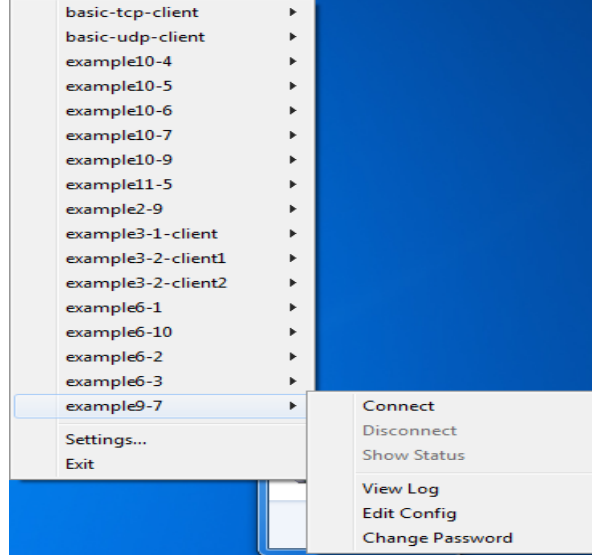
```
root@siteC:/usr/ports/www/apache22 # htpasswd -c
/usr/local/etc/apache22/vpnpass-file openvpn
New password: Şifrəni_yazaq
Re-type new password: Şifrəni_təkrar_yazaq
```

3. OpenVPN GUI-ni quraşdıraraq ki, HTTP proxy-ni dəstəkləsin: OpenVPN GUI-nin üstündə sağ düyməni sıxın və **Settings**-ə daxil olun. Quraşdırmanı aşağıdakı şəkildə olduğu kimi edin:

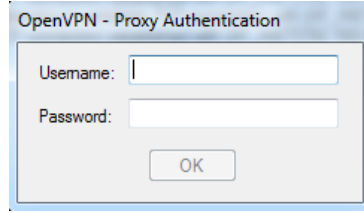




4. Artıq OpenVPN GUI ilə **example9-7.ovpn** quraşdırması ilə OpenVPN serverə qoşulmağa çalışın:



Əgər hər şeyi düzgün quraşdırmışıqsa, aşağıdakı şəkildə göstərildiyi kimi, sizə login/password səhifəsi çıxacaq:



İstifadəçi adı və şifrəni düzgün daxil etsəniz HTTP proxy server sizə OpenVPN serverə qoşulmağa izin verəcək. Sonda da qoşulma aşağıdakı şəkildəki kimi uğurlu olacaq:

```

OpenVPN Connection (example9-7)
Current State: Connected
Sat Mar 29 23:15:34 2014 Send to HTTP proxy: 'CONNECT openvpnserver.example.com:1194 HTTP/1.0'
Sat Mar 29 23:15:34 2014 Attempting Basic Proxy-Authentication
Sat Mar 29 23:15:34 2014 HTTP proxy returned: 'HTTP/1.0 200 Connection Established'
Sat Mar 29 23:15:36 2014 TCPv4_CLIENT link local: [undef]
Sat Mar 29 23:15:36 2014 TCPv4_CLIENT link remote: [AF_INET]192.168.6.1:8888
Sat Mar 29 23:15:36 2014 MANAGEMENT: >STATE:1396120536.WAIT...
Sat Mar 29 23:15:36 2014 MANAGEMENT: >STATE:1396120536.AUTH...
Sat Mar 29 23:15:36 2014 TLS: Initial packet from [AF_INET]192.168.6.1:8888, sid=8655875e 4dfceae
Sat Mar 29 23:15:36 2014 VERIFY OK: depth=1, C=NL, O=Cookbook, CN=Cookbook CA, emailAddress=openvpn-ca@atl.az
Sat Mar 29 23:15:36 2014 VERIFY OK: nsCertType=SERVER
Sat Mar 29 23:15:36 2014 VERIFY OK: depth=0, C=NL, O=Cookbook, CN=openvpnserver, emailAddress=openvpn-ca@atl.az
Sat Mar 29 23:15:37 2014 Data Channel Encrypt: Cipher 'BF-CBC' initialized with 128 bit key
Sat Mar 29 23:15:37 2014 Data Channel Encrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Sat Mar 29 23:15:37 2014 Data Channel Decrypt: Cipher 'BF-CBC' initialized with 128 bit key
Sat Mar 29 23:15:37 2014 Data Channel Decrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Sat Mar 29 23:15:37 2014 Control Channel: TLSv1, cipher TLSv1/SSLv3 DHE-RSA-AES256-SHA, 2048 bit RSA
Sat Mar 29 23:15:37 2014 [openvpnserver] Peer Connection Initiated with [AF_INET]192.168.6.1:8888
Sat Mar 29 23:15:38 2014 MANAGEMENT: >STATE:1396120538.GET_CONFIG...
Sat Mar 29 23:15:39 2014 SENT CONTROL [openvpnserver]: 'PUSH_REQUEST' (status=1)
Sat Mar 29 23:15:39 2014 PUSH: Received control message: 'PUSH_REPLY,route-gateway 192.168.200.1,topology subnet,ping 10,ping-restart 60,socket-flags TCP_NODELAY,ifconfig 192.168.200.2,255.255.255.0'
Sat Mar 29 23:15:39 2014 OPTIONS IMPORT: timers and/or timeouts modified
Sat Mar 29 23:15:39 2014 OPTIONS IMPORT: --socket-flags option modified
Sat Mar 29 23:15:39 2014 OPTIONS IMPORT: --ifconfig/up options modified
Sat Mar 29 23:15:39 2014 OPTIONS IMPORT: route-related options modified
Sat Mar 29 23:15:39 2014 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0
Sat Mar 29 23:15:39 2014 MANAGEMENT: >STATE:1396120539.ASSIGN_IP,,192.168.200.2.
Sat Mar 29 23:15:39 2014 open_tun, tt->ipv6=0
Sat Mar 29 23:15:39 2014 TAP-WIN32 device [Local Area Connection 2] opened: '\\Global\{B835D57D-F453-48B3-A987-077A7A6E65DC}.tap
Sat Mar 29 23:15:39 2014 TAP-Windows Driver Version 9.9
Sat Mar 29 23:15:39 2014 TAP-Windows MTU=1500
Sat Mar 29 23:15:39 2014 Set TAP-Windows TUN subnet mode network/local/netmask = 192.168.200.0/192.168.200.2/255.255.255.0 [SUCCEEDED]
Sat Mar 29 23:15:39 2014 Notified TAP-Windows driver to set a DHCP IP/netmask of 192.168.200.2/255.255.255.0 on interface {B835D57D-F453-48B3-A987-077A7A6E65DC} [DHCP-serv: 192.168.200.254, lease-time: 31536000]
Sat Mar 29 23:15:39 2014 Successful ARP Flush on interface [16] {B835D57D-F453-48B3-A987-077A7A6E65DC}
Sat Mar 29 23:15:44 2014 TEST ROUTES: 0/0 succeeded len=0 ret=1 a=0 u/d=up
Sat Mar 29 23:15:44 2014 Initialization Sequence Completed
Sat Mar 29 23:15:44 2014 MANAGEMENT: >STATE:1396120544.CONNECTED,SUCCESS,192.168.200.2,192.168.6.1

```

Client quraşdırmasında **verb 4** rejimdə qoşulsanız yuxarıda jurnalda gördüyümüz kimi, OpenVPN client HTTP Proxy serverə **Basic Proxy-Authentication** rejimində qoşulmağa çalışır. Əgər autentifikasiya uğurlu olarsa, HTTP proxy server yetki verir ki, OpenVPN serverə qoşulma davam etsin.

### Bu necə işləyir...

Bu misalımız eynilə öncədə olduğu kimi, HTTP Proxy üzərindən Basic Authentication ilə qoşulmağa çalışır. OpenVPN GUI-də **HTTP proxy settings**-də istifadəçi adı və şifrə quraşdırılır. Nəzərə alın ki, bu misalımızda quraşdırma faylında heç bir dəyişikliyə ehtiyac yoxdur çünki, bu işi quraşdırmanın əvəzinə OpenVPN GUI edəcək. Uğurlu qeydiyyatdan sonra, client **HTTP 'CONNECT'** müraciətini Serverə yollayır. Bu hissə artıq adi TCP bazalı qoşulma ilə eynidir.

### Daha da ətraflı...

OpenVPN clienti qoşulma üsulu olaraq HTTP Proxy istifadə etdikdə, çoxlu autentifikasiya metodundan istifadə edə bilər.

### NTLM Proxy authorization

OpenVPN həmçinin HTTP proxy üzərindən NTLM autentifikasiya metodunu da dəstəkləyir (NTLM - **NT Lan Manager**). Bu autentifikasiya metodu Microsoft

Windows istifadəçi bazası ilə işləmək üçün istifadə edilir. Ancaq OpenVPN NTLM ilə müəyyən limitlə işləyir. Düzgün **NTLMSSP** mesajlarını yollaya bilmir və məhdud proxy serverlərlə işləyə bilər. Bu proxy-nin işləməsi üçün aşağıdakı sətirlərdən birini istifadə edilən NTLM versiyasından asılı olaraq, quraşdırma faylınıza əlavə etməyiniz yetər:

```
http-proxy proxyhost proxyport stdin ntlm
http-proxy proxyhost proxyport stdin ntlm2
```

stdin olan yer isə OpenVPN-ə başa salır ki, istifadəçi adı və şifrəni cli-a çap elə. Bu tip authorization metodu Windows OpenVPN GUI ilə yaxşı işləmir.

### **OpenVPN2.2-dən başlayaraq yeni imkanlar**

OpenVPN2.2-dən başlayaraq HTTP digest autentifikasiya metodu dəstəklənməyə başladı hansı ki, açıq şəkildə gedən plain-text metoddan daha təhlükəsizdir. Həmçinin **http-proxy** autentifikasiya üçün **auto-nct** adlı yeni opsiya yaratdı ki, zəif olan autentifikasiya metodunu özü bağlaya bilsin.

### **Həmçinin baxın**

- Bu başlıqda olan öncəki misala harda ki, HTTP proxy heç bir autentifikasiyasız qoşulurdu.

### **DynDNS-in istifadə edilməsi**

Bəzi hallar olur ki, OpenVPN dinamik IP ünvanı istifadə eləmək məcburiyyətində qalır. Bu o deməkdir ki, OpenVPN client hər dəfə fərqli IP ünvanına qoşulmalı olur çünki, hər dəfə serverin IP ünvanı dəyişir. Bu o halda ola bilər ki, OpenVPN server Internetə ADSL ilə qoşulur. Bu başlıqda biz sizə OpenVPN üçün Dynamic DNS adının necə quraşdırılmasını və client-in necə Dynamic DNS adından istifadə edilməsi üçün quraşdırılmasını göstərəcəyik. Dinamik DNS provider kimi, **dyndns.org** istifadə edə bilərsiniz.

### **İşə hazırlaşaq**

2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Client maşını isə Windows7 x64 OpenVPN2.3-də olacaq. Server üçün 2-ci başlıqda server tərəf routing üçün yaratdığımız **basic-udp-server.conf** quraşdırma faylından istifadə edəcəyik. Client üçün də həmçinin 2-ci başlıqda yaratdığımız **basic-udp-client.ovpn** quraşdırma faylından istifadə edəcəyik.

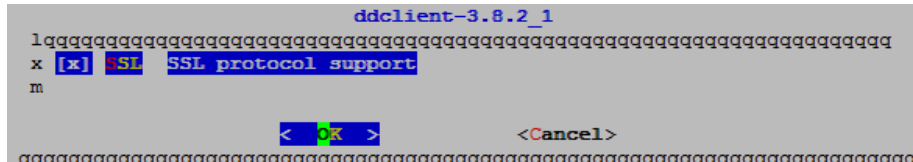
### **Necə edək...**

1. **dyndns.org** ünvanında qeydiyyatdan keçin ki, dinamik dns istifadə edə bilərsiniz.
2. Hansısa bir hostname ilə hal-hazırkı VPN serverinizin IP-sini qeydiyyata salın. Misal üçün **openvpn.dyndns.org**
3. DNS adı **openvpn.dyndns.org** yoxlayın ki, həqiqətən də sizin təyin etdiyiniz IP ünvan qayıdırımı:  
host openvpn.dyndns.org  
openvpn.dyndns.org has address 1.1.1.10

4. **ddclient** alətini istifadə edin ki, **dyndns.org**-a IP ünvanı tez-tez və asan yeniləyə bilsin. Bunun üçün **ddclient.conf** faylından istifadə edəcəyik. Öncə FreeBSD maşına paketi yükləyək:

```
root@siteC:/usr/ports/www/apache22 # cd `whereis ddclient | awk '{
print $2 }'`
```

```
root@siteC:/usr/ports/dns/ddclient # make config # Lazımı
modulları seçirik
```



```
root@siteC:/usr/ports/dns/ddclient # make install
root@siteC:/usr/ports/dns/ddclient # cp
/usr/local/etc/ddclient.conf.sample /usr/local/etc/ddclient.conf
```

**/usr/local/etc/ddclient.conf** faylını aşağıdakı şəkildə quraşdıraraq:

```
daemon=0
syslog=yes
mail-failure=root
pid=/var/run/ddclient/ddclient.pid
ssl=yes
use=web, web=checkip.dyndns.org/, web-skip='IP Address'
login=dyndns-username
password=dyndns-password
server=members.dyndns.org, \
protocol=dyndns2 \
openvpn.dyndns.org
```

Startupa əlavə edək və işə salaq:

```
root@siteC:/usr/ports/dns/ddclient # echo 'ddclient_enable="YES"' >>
/etc/rc.conf
```

```
root@siteC:/usr/ports/dns/ddclient # /usr/local/etc/rc.d/ddclient start
```

5. Serveri işə salaq:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-
server.conf
```

Sonra isə client üçün **basic-udp-client.ovpn** quraşdırma faylını **example11-7.ovpn** adlı fayla nüsxələyin və **example11-7.ovpn** faylında aşağıdakı dəyişiklikləri edin:

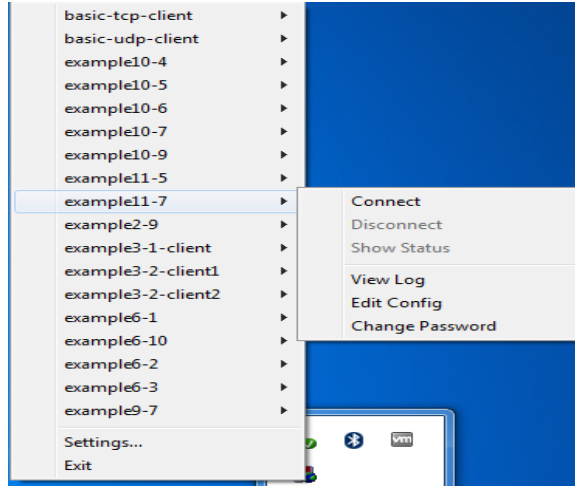
Aşağıdakı sətiri:

```
remote openvpnserver.example.com
```

Bu sətirlərə dəyişin:

```
remote openvpn.dyndns.org
resolv-retry 300
```

6. Client-i işə salın:



OpenVPN client **openvpn.dyndns.org** adını hal-hazırkı IP ünvanına resolve edəcək və OpenVPN serveri tapacaq.

7. **openvpn.dyndns.org** üçün yeni IP ünvanı ya WEB intefeys ilə ya da **ddclient** aləti ilə edin:

```
root@siteC:/usr/ports/dns/ddclient # ddclient --verbose
```

8. Sonra openvpn.dyndns.org adınının yeni IP ünvanla resolv olmasını yoxlayın.

9. OpenVPN serveri eyni quraşdırma ilə yenidən işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-server.conf
```

10. Sonra original OpenVPN serveri dayandırın. Müəyyən vaxtdan sonra OpenVPN client dayanacaq və yenidən yeni adı resolve edəcək. Yeni **openvpn.dyndns.org** adı yeni yeni IP ilə tanınacaq. Ardınca ilə yenidən qoşulmaq olacaq.

### **Bu necə işləyir...**

**dyndns** servisi istifadəçiyə izin verir ki, pulsuz olaraq dns adını qeydiyyatı ala bilsin. Bu dns adı seçdiyimiz IP-yə resolve ediləcək ancaq, DNS üçün **TTL(Time-To-Live)** flag çox qısadır. Bu o deməkdir ki, təyin edilən DNS adına yazılmış IP ünvan dünyada yalnız çox qısa müddət üçün yayımlanacaq. Bu ona görə yaxşıdır ki, əgər IP ünvan dəyişərsə, DNS adının resolve edilməsində tez baş verir.

**ddclient** alətinin istifadəsi ilə, DNS adınının təyin edilmiş host üçün yeni IP ünvanına mənimsədir. ddclient-i **--verbose** rejimdə işə salın ki, dyndns web servis haqqında səhvləri /var/log/messages ünvanında görə bilərsiniz. OpenVPN client tərəfdə biz əlavə direktiv yazdıq:

```
resolv-retry 300
```

Bu ondan ötrüdür ki, OpenVPN client çalışır ki, remote direktivində tapdığı DNS adını **300** saniyelik resolve eləsin. Əgər bu direktiv əlavə edilməzsə,

onda dns adının yenidən yoxlanılma cəhdi heç bir zaman olmayacaq. Bu o deməkdir ki, bu halda dyndns hostname-i heç resolve edilməyə bilər və bu halda client dayanacaq.

### **Daha da ətraflı...**

**dyndns** istifadəsilə OpenVPN serverin dynamic IP ünvanla işləməsinin imkanlarını genişlədirik. Ancaq bu işin tamda avtomatlaşdırılması mənası demək deyil. Linux maşınlarında bu heç NetworkManager ilə inteqrasiya edilməyib.

### **Failover**

Yadda saxlayın ki, hətta dnydns servisin özü olsa da belə, OpenVPN client yenə də restart edilməlidir ki, serverə yenidən qoşulma edə bilsin. Həmçinin yadda saxlayın ki, OpenVPN hələki transparent failoveri dəstəkləmir. Yeni ki, mövcud olan qoşulmaları bir serverdən digərinə miqrasiya edə bilmir. Biz buna mövcud başlığın **Multiple remotes & remote-random** misallarında baxmışdıq.

### **NetworkManager və 'ddclient'**

Linux üçün olan NetworkManagerin yeni versiyalarında ddclient üçün dispatcher plugini var. Bu plugin şəbəkə kartı yeni IP ünvan alan kimi işə düşür. Bu plugin dyndns qeydiyyatının yenilənməsi üçün quraşdırıla bilər. Bu o deməkdir ki, siz DHCP-dən aldığınız IP ünvanı lazım olan Hostname-ə mənimsədilib resolv edilmə işini avtomatik edə bilərsiniz.

Unutmayın, bu plugin o zaman həqiqətən işlək vəziyyətdə olur ki, əgər sizin şəbəkə kartınız NetworkManager ilə yalnız PUBLIC IP ünvan alır.

### **Həmçinin baxın**

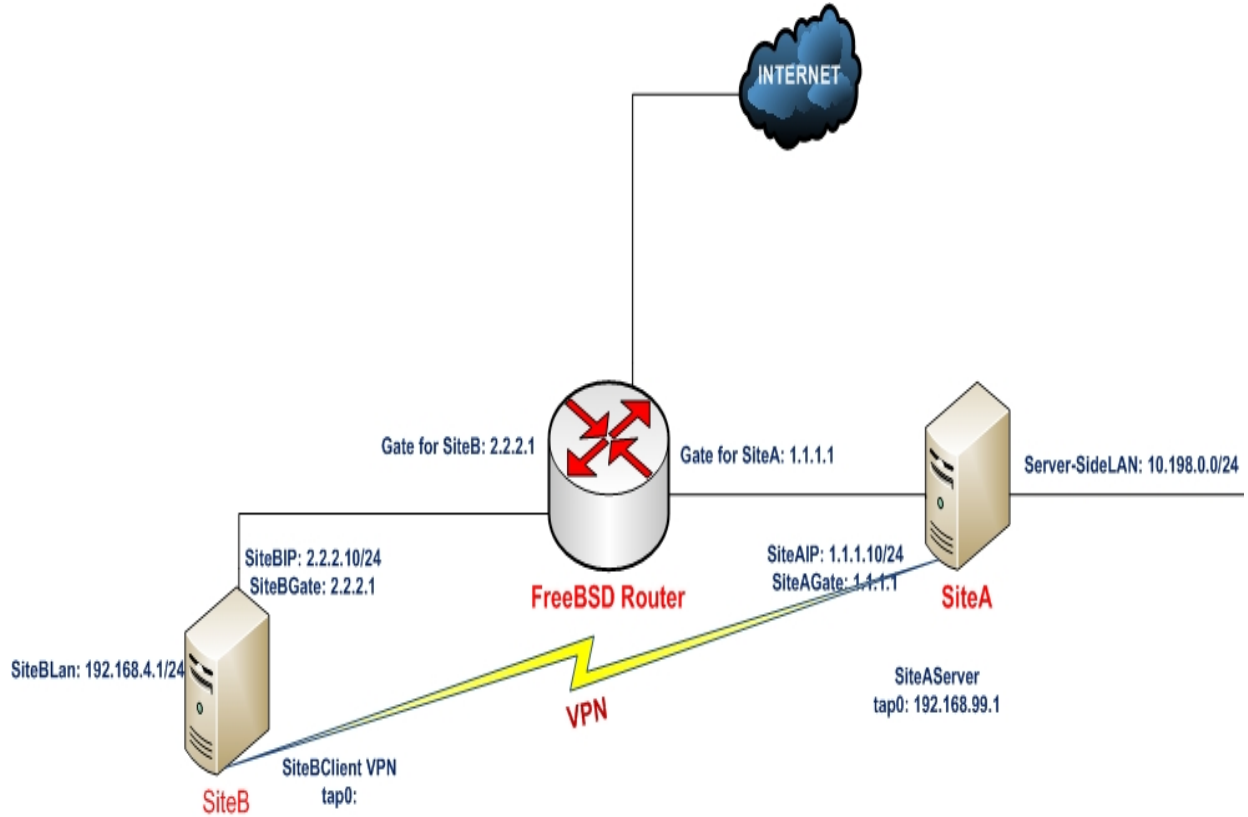
- Bu başlığın əvvəlində olan Multiple **remotes & remote-random** misalına

### **IP daha az olan quruluşlar(ifconfig-noexec)**

Bu misalın məqsədi, OpenVPN-in son nöqtələrinə IP ünvanı vermədən tunelin qaldırılmasıdır. Route edilmiş şəbəkələrdə bu təminat verir ki, tunelin son nöqtələri öz üzərlərindən bir-birlərinə çata bilməyəcəklər hansı ki, təhlükəsizliyi müəyyən dərəcədə artırır və həmçinin də route cədvəlini nisbətən kiçildə bilir. OpenVPN quraşdırma faylında IP ünvan təyin edilməlidir ancaq, heç bir vaxt tunel interfeysinə mənimsədilməli deyil.

### **İşə hazırlaşaq**

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda server və client maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Server üçün 3-cü başlıqda olan **non-bridge** üçün yaratdığımız **example3-1-server.conf** quraşdırma faylından istifadə edəcəyik.

### Necə edək...

1. Server üçün **example3-1-server.conf** faylını **example11-8-server.conf** faylına nüsxələyin və **example11-8-server.conf** faylının içinə aşağıdakı sətiri əlavə edin.

```
route 192.168.4.0 255.255.255.0 192.168.99.1
```

2. Serveri işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example11-8-server.conf
```

3. Client üçün client maşında **example11-8-client.conf** adlı quraşdırma faylını yaradın və içinə aşağıdakı sətirləri əlavə edin:

```
client
proto udp
remote openvpnservers.example.com
port 1194
```

```
dev tap
nobind
```

```
ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnclient1.crt
key /usr/local/etc/openvpn/openvpnclient1.key
tls-auth /usr/local/etc/openvpn/ta.key 1
```

```
ns-cert-type server
```

```
script-security 2
ifconfig-noexec
up /usr/local/etc/openvpn/example11-8-up.sh
```

```
route-noexec
route-up /usr/local/etc/openvpn/example11-8-route-up.sh
```

4. Sonra `/usr/local/etc/openvpn/example11-8-up.sh` faylını yaradaq:
- ```
#!/usr/local/bin/bash
```

```
/sbin/ifconfig $1 0.0.0.0/0 up
# TAP alətləri üçün tələb edilir
sysctl net.link.ether.inet.proxyall=1
```

5. Uyğun olaraq `/usr/local/etc/openvpn/example11-8-route-up.sh` scriptini yaradaq:

```
#!/usr/local/bin/bash
# VPN son nöqtəsinə açıq routun əlavə edilməsi
/sbin/ip route add $route_vpn_gateway/32 -interface $dev
n=1;
while [ $n -le 100 ]
do
    network=`env | sed -n
"/^route_network_${n}=/s/^route_network_${n}=//p"`
    netmask=`env | sed -n
"/^route_netmask_${n}=/s/^route_netmask_${n}=//p"`
    if [ -z "$network" -o -z "$netmask" ]
    then
        break
    fi
    /sbin/ip route add $network/$netmask -interface $dev
    let n=n+1
done
```

6. Scriptləri yerinə yetirən edin və `client-i` işə salın:

```
root@siteB:/usr/local/etc/openvpn # chmod 755
/usr/local/etc/openvpn/example11-8*.sh
root@siteB:/usr/local/etc/openvpn # openvpn --config example11-8-
client.conf
```

7. Client uşurla OpenVPN serverə qoşulduqdan sonra işə `tap0` alətini, routing cədvəlini və serverə ping getməsinə yoxlayın:

```
root@siteB:~ # ifconfig tap0
tap0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
    options=80000<LINKSTATE>
    ether 00:bd:3c:df:00:00
    inet 0.0.0.0 netmask 0xff000000 broadcast 0.255.255.255
```



```
media: Ethernet autoselect
status: active
Opened by PID 1715
```

```
root@siteB:~ # netstat -rn | grep tap
0.0.0.0/8          link#11          U              0          0    tap0 =>
10.198.0.0/31     00:bd:3c:df:00:00 US              0          0    tap0
192.168.99.1/32   00:bd:3c:df:00:00 US              0          0    tap0
```

```
root@siteB:~ # ping -c 2 192.168.99.1
PING 192.168.99.1 (192.168.99.1): 56 data bytes
64 bytes from 192.168.99.1: icmp_seq=0 ttl=63 time=0.537 ms
64 bytes from 192.168.99.1: icmp_seq=1 ttl=63 time=0.844 ms
```

Bu necə işləyir...

OpenVPN server client üçün IP ünvan aralığını əlavə edir ancaq, bu o demək deyil ki, client həmişə bu IP ünvanı özünə mənimsətməlidir. **example11-8-up.sh** scripti məhz bu işi görür.

Bəzi köhnə UNIX/Linux distributivlərin kernelləri şəbəkə kartında IP ünvan olmadan routing əlavə etmək imkanına malik olmurlar. Ona görə də biz **tap0** alətinə **0.0.0.0** IP ünvanı mənimsətdik. Server tərəfindən ötürülən routinglərin əlavə edilməsi üçün isə, **example11-8-route-up.sh** adlı spesifik **route-up** scripti istifadə edilir ki, bütün routingləri UP edsin.

Daha da ətraflı...

IP olmayan quraşdırmaları etdikdə xahiş edirik aşağıdakıları nəzərə alınız:

Point-to-Point və TUN stilli şəbəkələr

Bu misal həmçinin point-to-point stilli mühitlərdə də istifadə edilə bilər hansı ki, iki şəbəkəyə qoşulmaqdan ötrü static açarlardan istifadə edilir. Həmçinin uyğun olaraq, TUN stilli şəbəkələrdə də istifadə edilə bilər.

Routing və firewallama

İlk baxışdan bu misal biraz qərribə gələ bilər. Ancaq üstünlüyü ondan ibarətdir ki, OpenVPN clientə digər clientlər çata bilməyəcəklər. Bu o halda gərəkli olur ki, OpenVPN serverə çoxlu client qoşulur ancaq, bəzi clientlər öz arxalarında olan şəbəkələr üçün gateway rolunu oynayır. Remote office-ə gateway təyin edilməməsilə gateway maşına heç bir risk qalmır. Hətta VPN tərəfin arxasında da hücum edilərsə də belə. Həmçinin firewall-la elə qayda yazmaq olar ki, OpenVPN client-də VPN mənbəli IP ünvan gələrsə onu DROP eləsin. Məhz bu səbəbdən OpenVPN serverin client tərəfə birbaşa marşrutu həmişə olur:

```
route 192.168.4.0 255.255.255.0 192.168.99.1
```

BÖLÜM 12

OpenVPN 2.2-nin yeni imkanları

Bu başlıqda biz aşağıdakıları açıqlayacağıq:

- Sətir arası sertifikatlar
- Qoşulma blockları
- HTTPS server ilə portun yayımlanması
- Routing bacarıqları: **redirect-private**, **allow-pull-fqdn**
- PUBLIC IP ünvanların mənimsədilməsi
- OCSP dəstəklənməsi
- OpenVPN2.2-də yenilik: **x509_user_name** parametri

Giriş

Bu başlıqda biz OpenVPN2.2-də olan yeni imkanlara diqqətimizi ayıracayıq.

Başlığımızın misallarında biz yeni imkanlar olan **inline sertifikatlar**, **qoşulma blockları** və **port-sharing** istifadəsi ilə məşğul olacağıq.

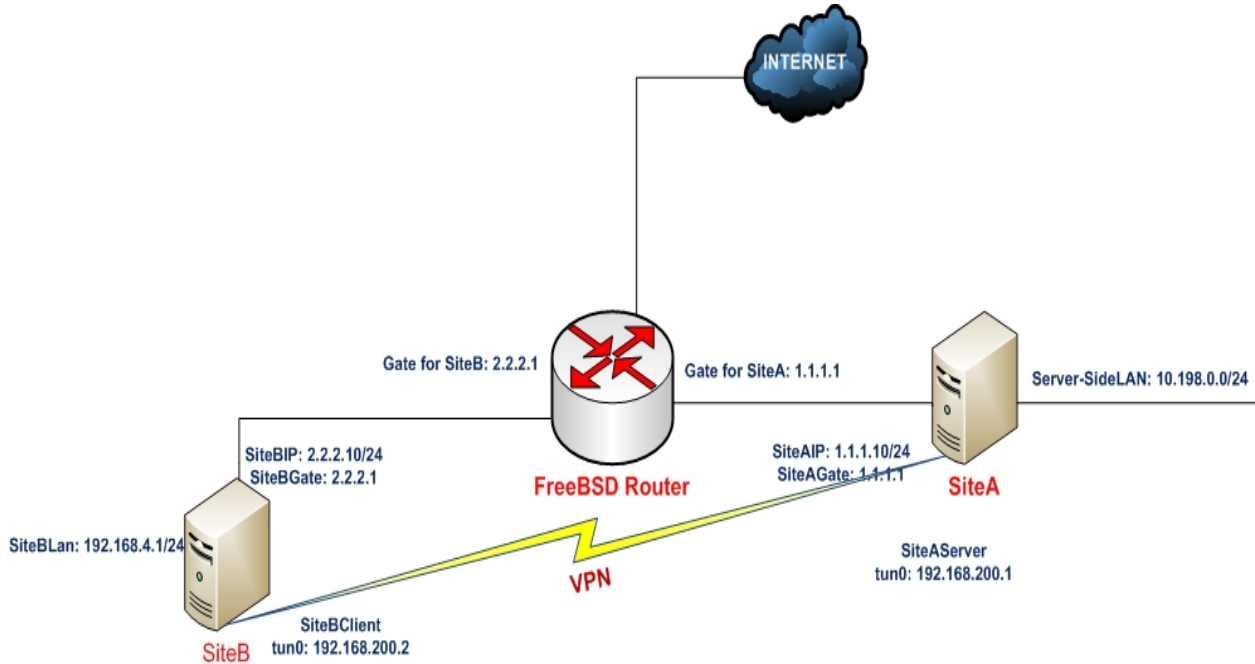
Başlığımızın sonunda isə lap yeni imkan olan OCSP ilə OpenVPN-i inteqrasiya edəcəyik.

Sətir arası sertifikatlar

OpenVPN quraşdırılma işini asanlaşdırmaq üçün yeni imkanla, public və private açarları bir faylda istifadə etmək olur. Bu iş client quraşdırma faylında **ca,cert,key** daxilində olan tərkibin **tls-auth** faylıyla inteqrasiyası etməklə olur. Bu misalımızda biz bu tip faylı yaradıb standart OpenVPN serverə qoşulacayıq.

İşə hazırlaşaq

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda server və client maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Server üçün 2-ci başlıqda olan server-tərəf routing üçün yaratdığımız **basic-udp-server.conf** quraşdırma faylından istifadə edəcəyik.

Necə edək...

1. İlk olaraq serveri işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-server.conf
```

2. **example12-1-client.conf** adlı client quraşdırma faylını yaradın və içine aşağıdakı sətirləri əlavə edin:

```
client  
proto udp  
remote openvpnserver.example.com  
port 1194  
dev tun  
nobind
```

```
ca [inline]  
cert [inline]
```

```
key [inline]
tls-auth [inline] 1

<ca>
-----BEGIN CERTIFICATE-----
# ca.crt sertifikatının base64 formatında olan içini bura əlavə edin
-----END CERTIFICATE-----
</ca>

<cert>
-----BEGIN CERTIFICATE-----
# openvpnclient1.crt sertifikatının base64 formatında olan içini bura
əlavə edin
-----END CERTIFICATE-----
</cert>

<key>
-----BEGIN PRIVATE KEY-----
# openvpnclient1.key keyinin base64 formatında olan içini bura əlavə
edin
-----END PRIVATE KEY-----
</key>

<tls-auth>
-----BEGIN OpenVPN Static key V1-----
# ta.key-in içini burda əlavə edin.
-----END OpenVPN Static key V1-----
</tls-auth>

ca.crt, openvpnclient1.crt, openvpnclient1.key və ta.key fayllarının
tərkibini quraşdırma faylına əlavə edin.
```

3. Sonra clienti işə salın:

```
root@siteB:/usr/local/etc/openvpn # openvpn --config example12-1-
client.conf
```

Bu necə işləyir...

OpenVPN-də **ca**, **cert**, **key** və **tls-auth** quraşdırma direktivlərini açdıqda və onların mənasını **[inline]** hissəsində tapdıqdan sonra quraşdırmanın davamında **XML** block hissəsi mütləq olmalıdır. XML block-nun tərkibi sonradan oxunur və mənimsədir. Əgər quraşdırmada bütün tələb edilən XML blocklar olarsa, qoşulma uğurlu olacaq.

Ancaq onu da deyək ki, öncə göstərilən blockların hamısını birdən göstərməyə ehtiyac yoxdur. Həmçinin mümkündür ki, yalnız CA sertifikatının **[inline]** blokunu göstəresiniz.

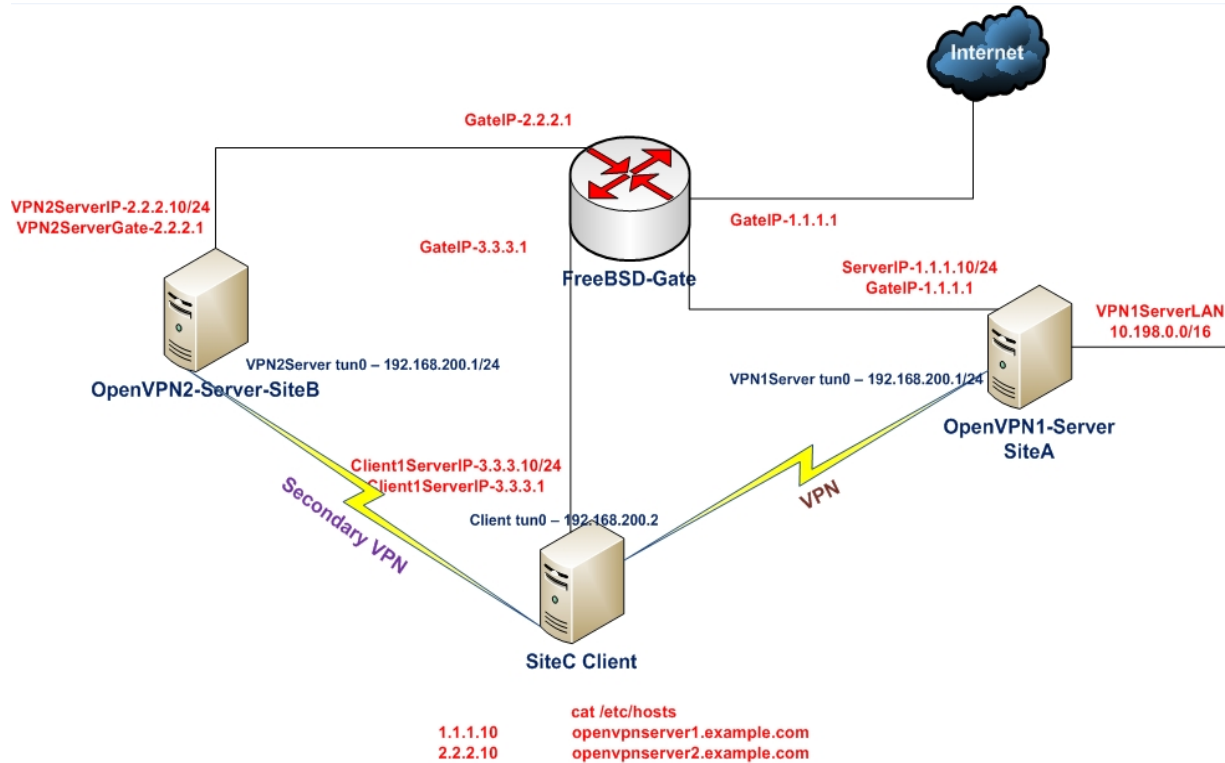
Qoşulma blokları

Öncəki misalımızda göstərdiyimiz inline certificates kimi həmçinin, qoşulma blocklarını da göstərmək olar. Bu qoşulma blockları remote serverlər üçün fərqli təyinatlara ayrılırlar və onlar VPN qoşulması olanadək qayda ilə çalışırlar. Connection blockun istifadə edilməsinin üstünlüyü ondan ibarətdir ki, hər bir remote server üçün xüsusi spesifik parametrlər təyin edilə bilər (Məsəl üçün protocol UDP və ya TCP, və ya uzaq port harda ki, proxy server istifadə edilməlidir və.s).

Bu misalda biz iki server quraşdıracağıq. Bir server TCP-də qulaq asır və digəri isə UDP-də qulaq asır. Sonra biz OpenVPN clienti elə quracağıq ki, ilk olaraq UDP serverə qoşulmağa çalışsın. Əgər qoşulma uğurlu olmazsa, onda TCP ilə qoşulmağa çalışacaq.

İşə hazırlaşaq

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



2-ci başlıqda yaratdığımız client və server sertifikatlarını burdada istifadə edəcəyik. Bu misalda 2 ədəd server və client maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. İlk server üçün 2-ci başlıqda olan server-tərəf routing üçün yaratdığımız **basic-udp-server.conf** quraşdırma faylından istifadə edəcəyik. İkinci server üçün isə 9-cu başlıqda olan TCP bazalı qoşulmaların təkmilləşdirilməsi üçün yaratdığımız **example9-7-server.conf** quraşdırma faylından istifadə edəcəyik.

Necə edək...

1. Hər iki serveri işə salaq:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config basic-udp-  
server.conf
```

```
root@siteB:/usr/local/etc/openvpn # openvpn --config example9-7-  
server.conf
```

2. Hər iki serverin jurnallarını yoxlayın və əmin olun ki, uğurla qalxdılar.
3. Client maşın üçün **/etc/hosts** faylında aşağıdakı sətirlər olmalıdır:

```
1.1.1.10          openvpnsver1.example.com  
2.2.2.10          openvpnsver2.example.com
```

Client üçün **/usr/local/etc/openvpn/example12-2-client.conf** quraşdırma faylını yaradın və içinə aşağıdakı sətirləri əlavə edin:

```
client  
dev tun
```

```
<connection>  
remote openvpnsver1.example.com  
proto udp  
port 1194  
</connection>
```

```
<connection>  
remote openvpnsver2.example.com  
proto tcp  
port 1194  
</connection>
```

```
ca /usr/local/etc/openvpn/ca.crt  
cert /usr/local/etc/openvpn/openvpnclient1.crt  
key /usr/local/etc/openvpn/openvpnclient1.key  
tls-auth /usr/local/etc/openvpn/ta.key 1
```

```
ns-cert-type server
```

4. Clienti işə salın:

```
root@siteC:/usr/local/etc/openvpn # openvpn --config example12-2-  
client.conf
```

5. Qoşulma uğurla başa çatdıqdan sonra, ilk serverin prosesini dayandırın ki, client digər maşına qoşulsun.

```
root@siteA:/usr/local/etc/openvpn # killall openvpn
```

Müəyyən vaxt gözləyin ki, client yenidən qoşulma cəhdi eləsin. Susmaya görə olan vaxt periodundan sonra, client ikinci serverə TCP protokolu ilə yenidən qoşulacaq

Bu necə işləyir...

OpenVPN client işə düşdükdə o çalışır ki, ilk **<connection>** blockunda olan serverə qoşulsun. Əgər qoşulmada səhv olarsa o timeout-dan sonra ikinci **<connection>** blockunda olan serverə qoşulacaq. Və bu ardıcillıq 4 ədəd

serverədək gedir. Əgər OpenVPN server dayanarsa və ya görünməzsə, client avtomatik olaraq özünü restart edəcək və ilk görünən serverə qoşulmağa yenidən cəhd edəcək.

OpenVPN client ilk öncə global direktivləri oxuyur hansı ki, **<connection>** blockundan kənarında yazılır. Hər bir block üçün isə block-spesifikasiyalı quraşdırmalar, global quraşdırmalardan daha prioritetli olur. Məhz buna görə də **<connection>** direktivini hər serverə uyğun olan spesifikasiya ilə təyin etmək çox asan olur.

Daha da ətraflı...

Connection blockları elə inline certificates kimidir. Bu imkanın çatışmamazlığı ondan ibarətdir ki, quraşdırma faylının CLI-dan istifadəsi xeyli çətinləşir. Connection block-ların istifadə edilməsində bəzi nöqtələr vardır ki, biz yadımızda saxlamalıyıq.

Connection blockların daxilində istifadə edilə bilinəcək direktivlər

Connection block daxilində yalnız aşağıdakı direktivlər istifadə edilə bilər:

- bind
- connect-retry, connect-retry-max, connect-timeout
- float
- http-proxy, http-proxy-option, http-proxy-retry, http-proxytimeout
- local lport
- nobind
- port
- proto
- remote, rport
- socks-proxy, socks-proxy-retry

Bütün digər direktivlər global kimi təyin edilə bilər və connection block daxilində istifadə edilə bilməz.

TCP və UDP bazalı quraşdırmanın birgə istifadə edilməsində çatışmamazlıq

TCP və UDP bazalı qoşulmanın birgə quraşdırılması çox asandır ancaq, global parametrlər bütün serverlər üçün eyni olmalıdır. Məhz buna görə siz **fragment** direktivini və digər tuning direktivlərini global parametrlərdə istifadə edə bilmirsiniz. Bu tip quraşdırmalar **<connection>** blockun daxilində istifadə edilə bilmir ancaq, gələcək versiyalarda təkmilləşdirilə bilər.

Həmçinin baxın

- 11-ci başlıqda olan **Multiple remotes & remote-random** misalı hansı ki, connection block olmadan eyni nəticəni əldə edir.

HTTPS server ilə portun yayımlanması

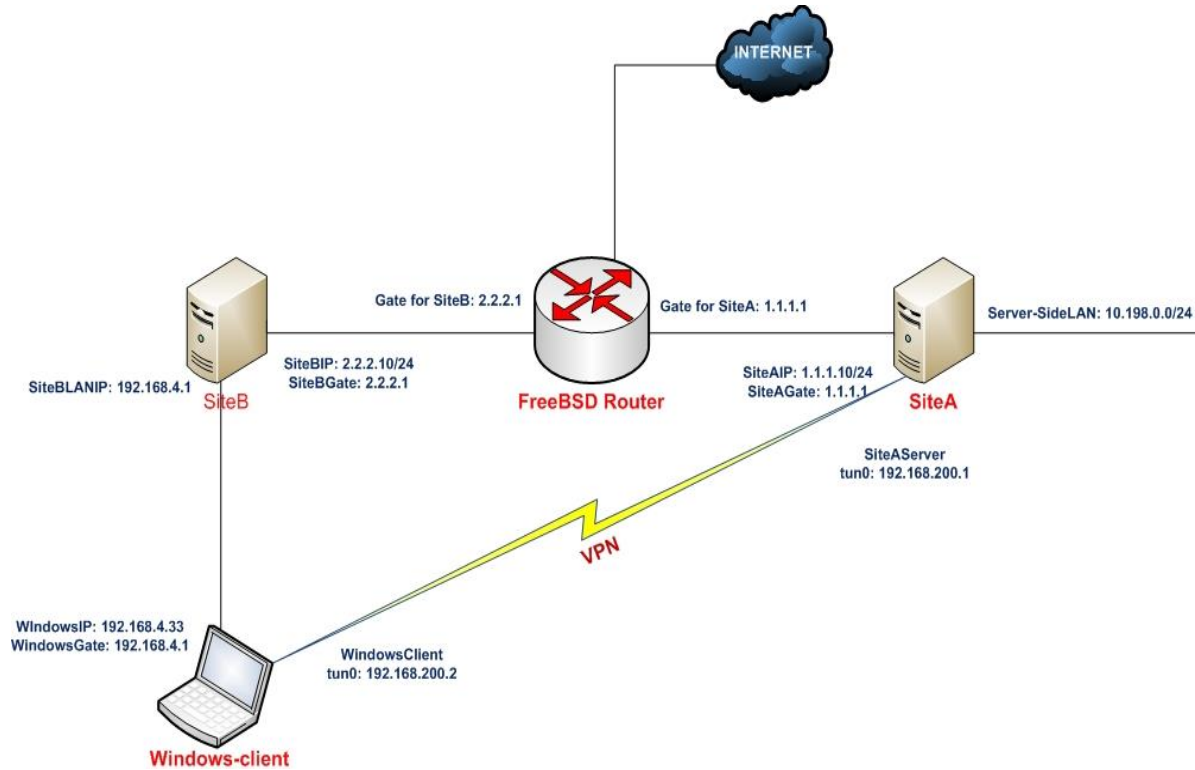
OpenVPN quraşdırması bizə şərait yaradır ki, kiçik ofis quraşdırmasında OpenVPN-in HTTPS 443-cü portda qulaq asa bilsin. Ancaq problem onda yaranır ki, bu port HTTPS üçün məşğul olduğu üçün OpenVPN bu portu istifadə edə bilmir. OpenVPN2.1-dən imkan yarandı ki, TCP portu cütləşdirib yayımlaya

bilsin. Yəni bütün paketlər OpenVPN serverə gəlir. OpenVPN özü iz qoyduğu paketlər ayrılır və OpenVPN-ə ötürülür, qalan paketlər isə digər maşın və porta yönləndirilir.

Bu misalımızda OpenVPN server-i elə quracağıq ki, 443-cü portda olan WEB server və OpenVPN server birgə işləyəcək.

İşə hazırlaşaq

Aşağıdakı şəbəkə quruluşundan istifadə edə biləcəyik:



2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Client maşını isə Windows7 x64 OpenVPN2.3-də olacaq. Server quraşdırması olaraq 9-cu başlıqda yaratdığımız TCP bazalı qoşulmaların təkmilləşdirilməsi üçün yaratdığımız **example9-7-server.conf** faylından istifadə edəcəyik. Həmçinin eyni başlıqda yaratdığımız **example9-7.ovpn** quraşdırma faylıni client üçün istifadə edəcəyik.

Bizim misalda OpenVPN serverin özündə apache22 server 8443-cu port listen edir. Apache22 server HTTPS ilə quraşdırılıb.

Apache22 serveri https ilə yükləyib quraşdıraraq.

```
cd /usr/ports/www/apache22          # Port ünvanına daxil olaq
make -DBATCH install                # Susmaya görə olan modullarla
                                     yükləyək.
```

/etc/rc.conf faylına aşağıdakı sətirləri əlavə edirik:
apache22_enable="YES"


```
apache2ssl_enable="YES"

/usr/local/etc/rc.d/apache22 start           # Apache22-ni işə salırıq

Sertifikatlar üçün lazımı ünvanları yaradaq və lazımı yetkiləri verək:
mkdir /usr/local/etc/apache22/ssl.key
mkdir /usr/local/etc/apache22/ssl.crt
chmod 0700 /usr/local/etc/apache22/ssl.key
chmod 0700 /usr/local/etc/apache22/ssl.crt

WEB HTTPS Server üçün tələb edilən açar və sertifikatı yaradaq.
cd /root
openssl genrsa -des3 -out server.key 1024      # Şifrəni daxil
                                              edirik
openssl req -new -key server.key -out server.csr # CSR
                                              yaradıırıq

# Yaratdığımız sertifikatı özümüz imzalayırıq.
openssl x509 -req -days 365 -in /root/server.csr -signkey
/root/server.key -out /root/server.crt

Sertifikatları lazımı ünvanlarına nüsxələyirik:
cp /root/server.key /usr/local/etc/apache22/ssl.key/
cp /root/server.crt /usr/local/etc/apache22/ssl.crt/

Sonra onlara düzgün yetkiləri veririk:
chmod 0400 /usr/local/etc/apache22/ssl.key/server.key
chmod 0400 /usr/local/etc/apache22/ssl.crt/server.crt

ee /usr/local/etc/apache22/extra/httpd-ssl.conf # Faylda
                                              aşağıdakı dəyişiklikləri edirik.
Listen 8443
ServerName openvpnsrvr.example.com:8443
SSLCertificateFile "/usr/local/etc/apache22/ssl.crt/server.crt"
SSLCertificateKeyFile
"/usr/local/etc/apache22/ssl.key/server.key"

/usr/local/etc/rc.d/apache22 restart         # WEB daemonu restart
                                              edirik
```

Necə edək...

1. Server üçün **example9-7-server.conf** quraşdırma faylını **example12-3-server.conf** faylına nüsxələyin və **example12-3-server.conf** faylının içində aşağıdakı sətirlərdə dəyişiklik edin:

Aşağıdakı sətiri:

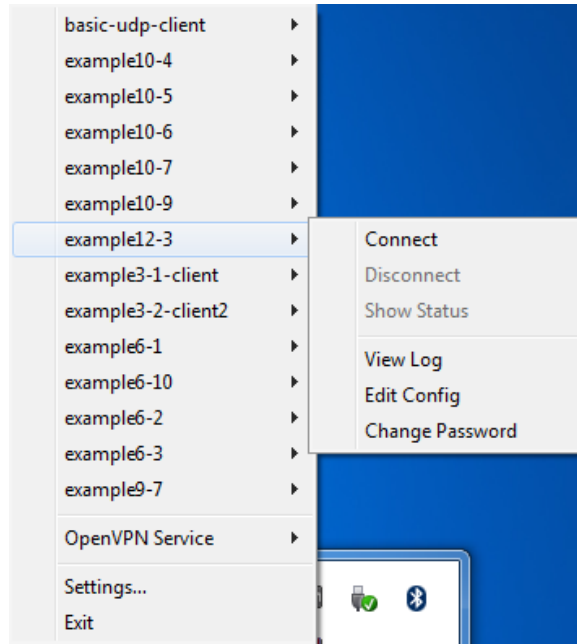
```
port 1194
```

Aşağıdakılara dəyişin:

```
port 443
```

port-share localhost 8443

2. Serveri işə salın:
root@siteA:/usr/local/etc/openvpn # **openvpn --config example12-3-server.conf**
3. Client üçün quraşdırma faylı **example9-7.ovpn**-i **example12-3.ovpn** adlı fayla nüsxələyin və **example12-3.ovpn** faylının içində **port-u 443** edin:
4. Sonra client-i işə salın:



5. Yoxlayın görək client OpenVPN serverə qoşula bilirmi. Client qoşulduqdan sonra browser-i açın və yoxlayın görək aşağıdakı linkə daxil ola bilərsinizmi (Sözsüz ki, **c:\windows\system32\drivers\etc\hosts** faylına uyğun sətir olmalıdır):
https://openvpnserver.example.com

OpenVPN serverin jurnalı aşağıdakı sətirləri göstərməlidir:

```
Tue Apr 1 15:04:59 2014 2.2.2.10:49168 TCP connection established with [AF_INET]2.2.2.10:49169
Tue Apr 1 15:04:59 2014 2.2.2.10:49168 Non-OpenVPN client protocol detected
Tue Apr 1 15:04:59 2014 2.2.2.10:49169 Non-OpenVPN client protocol detected
```

Bu necə işləyir...

port-shared istifadə edildikdə, OpenVPN onun **443**-cü portuna gələn axını analiz eləməyə başlayacaq. Əgər axın OpenVPN sessiyasının hissəsidirsə ya da bu OpenVPN handshake inisializasiyalıdırsa onda, OpenVPN server bu axını özünə götürəcək. Əgər bu axın OpenVPN tərəfindən təyin edilmirsə, onda bu axın **port-share** direktivində göstərilmiş IP və portun üzərinə yönləndiriləcək.

Beləliklə, OpenVPN server prosesi həmişə 443-cü portda qulaq asır. WEB server isə fərqli port və Host-da qulaq asmalıdır. Bu misalda eyni portu fərqli servislər üçün istifadə edə bilərsiniz.

Daha da ətraflı...

OpenVPN yönləndirdiyi WEB server təhlükəsiz WEB(HTTPS) server olmalıdır. Bu sizin OpenVPN serverdə olan SSL trafikə oxşamasından asılıdır. Əgər trafik apache üzərində işləyən 80-ci porta yönləndirilsə onda, aşağıdakı səhv çap ediləcək:

```
[error] [client 127.0.0.1] Invalid method in request \x16\x03\x01
```

Routing bacarıqları: `redirect-private`, `allow-pull-fqdn`

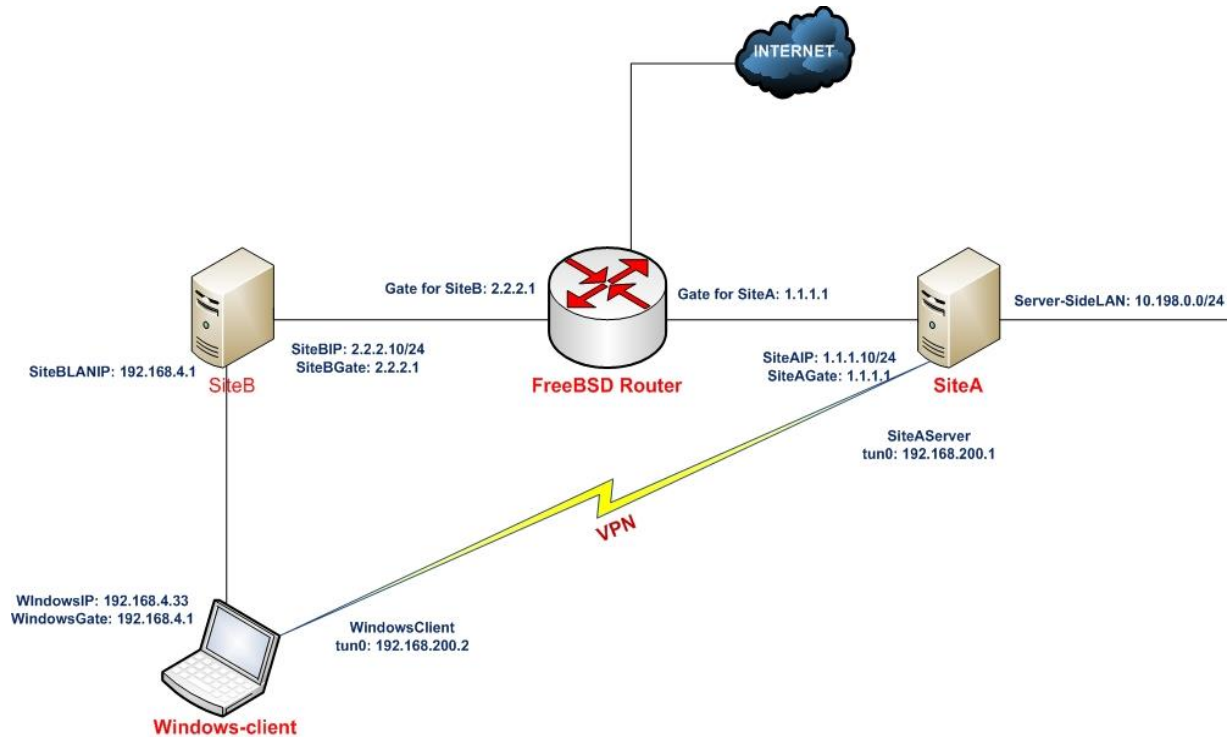
OpenVPN2.3-də bəzi routing imkanları açıqlanılır. **`redirect-gateway`** üçün yeni opsiyalar və yeni routing direktivləri mövcuddur:

- **`redirect-private`**: Bu opsiya **`redirect-gateway`**-ə çox oxşayır. Xüsusən də yeni parametrlər istifadə ediləndə. Ancaq bu yenə də default gateway demək deyil.
- **`allow-pull-fqdn`**: Clientə imkan yaradır ki, DNS adlarını OpenVPN-dən götürsün. Öncə yalnız IP ünvanlar yalnız götürülə vəya ötürülə bilərdi. Bu opsiya client-in quraşdırma faylına 'push' (ötürülə) vəya əlavə edilə bilməz.
- **`route-null`**: Routing opsiyalarından başqa, client tərəfindən serverə ötürülən bütün opsiyalar. Bu adətən OpenVPN-i troubleshoot edəndə lazım olur.
- **`max-routes n`**: Routing-in maximum rəqəmindən asılı olaraq bu ya serverdə təyin edilə bilər ya da remote server tərəfindən ötürülə bilər.

Bu misalda, biz diqqətimizi `redirect-private` direktivinə və onun parametrlərinə ayıracayıq. Misal olaraq, **`allow-pull-fqdn`** parametri.

İşə başlayaq...

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Client maşını isə Windows7 x64 OpenVPN2.3-də olacaq. Server quraşdırması olaraq 2-ci başlıqda server-tərəf routing üçün yaratdığımız **basic-udp-server.conf** faylından istifadə edəcəyik. Həmçinin eyni başlıqda yaratdığımız **basic-udp-client.ovpn** quraşdırma faylını client üçün istifadə edəcəyik.

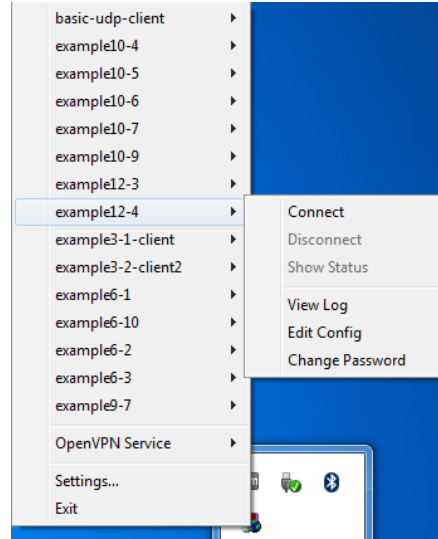
Necə edək...

1. **basic-udp-server.conf** quraşdırma faylını **example12-4-server.conf** quraşdırma faylına nüsxələyin və **example12-4-server.conf** faylının sonuna aşağıdakı sətirləri əlavə edin:

```
push "redirect-private bypass-dhcp bypass-dns"
push "route server.example.com"
```
2. Serveri işə salın:

```
root@siteA:/usr/local/etc/openvpn # openvpn --config example12-4-server.conf
```
3. Client üçün isə **basic-udp-client.ovpn** quraşdırma faylını **example12-4.ovpn** quraşdırma faylına nüsxələyin və **example12-4.ovpn** faylın sonuna aşağıdakı sətiri əlavə edin:

```
allow-pull-fqdn
```
4. Sonra clienti işə salın:



5. Qoşulma bitdikdən sonra routing cədvəlinə baxın:

- o Əgər DHCP ya da DNS server client olduğu şəbəkədən fərqli subnet-də yerləşirsə, onda yeni route əlavə ediləcək. Bu DHCP müraciətlərin VPN tunel üzərindən yox hələ də local DHCP serverə getməsinə əmin edir.
- o server.example.com hostuna routing həmçinin əlavə ediləcək.

Bu necə işləyir...

bypass-dhcp və **bypass-dns** opsiyaları, **redirect-gateway** və **redirect-private** direktivləri üçündür ona görə ki, OpenVPN client DNCP və DNS serverlər başqa şəbəkədə olsalar, onların görülməsi üçün əlavə route-lar yazacaq. Geniş şəbəkələrdə DNS server əksər hallarda client qoşulu olan local şəbəkədə tapılmaz. Əgər client qoşulduqdan sonra, DNS serveri görmək üçün route yazılırsa, bu dayanıqlığın ciddi azalmasına gətirib çıxara bilər. Tam demək olar ki, DNS Server tamamilə dayanacaq.

allow-pull-fqdn direktivi izin verir ki, route təyin elədikdə, DNS adını IP ünvanının yerinə istifadə eləmək olsun. Əgər hosta seçilmiş marşrut dinamik IP ünvanla işləyirsə onda, bu bizim köməyimizə çox çatacaq.

Daha da ətraflı...

Öncəki direktivlərin açıqlamasının hissəsindən sonra, orda çoxlu routing direktivləri mövcuddur ki, client-ə əlavə edilən routingin necə olması və idarə edilməsini göstərir.

route-nopull direktivi

route-nopull direktivi client-i çağırır ki, marşrut-dan başqa bütün informasiyanı serverdən alsın. Bu serverdə olan problemin tapılması üçün çox yaxşı ola bilər. Bu o demək deyil ki, OpenVPN client tərəfindən heç bir route əlavə edilmir. Yalnız **'route'** istifadə edilən ötürülmüş routinglər istifadə edilməyəcək.

'max-routes' direktivi

OpenVPN2.1-dən başlayaraq əmələ gəldi. Clientin çoxlu routinglərlə yığılmasının qarşısını almaq üçün **max-routes** direktivinə tələb yaranır. Opeşiya **max-routes n** ilə təyin edilir. **n** client quraşdırma faylında təyin edilə biləcək maksimal route sayını təyin edir. Susmaya görə olan mənası **100**-dür.

PUBLIC IP ünvanların mənimsədilməsi

OpenVPN2.1-dən başlayaraq **topology subnet** imkanı ilə şərait yarandı ki, clientlərə qoşulmaq üçün public IP ünvanlardan istifadə edilə bilinsin. Bu misalda biz belə quruluşun necə olmasını göstərəcəyik. Biz 2-ci başlıqda istifadə etdiyimiz '**proxy-arp**' texnikası ilə, əgər clientlər remote şəbəkənin bir hissəsidirsə, onları mövcud edəcəyik. **proxy-arp** istifadəsinin üstünlüyü ondan ibarətdir ki, ayrılmış public şəbəkə hissəsini həm client və həm də serverdə istifadə edə bilək.

İşə hazırlaşaq

Bu misalda server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Client maşını isə Windows7 x64 OpenVPN2.3-də olacaq. Client quraşdırması olaraq 2-ci başlıqda '**ifconfig-block**'-da yaratdığımız **basic-udp-client.ovpn** istifadə edin.

Bu misalı test etmək üçün public IP ünvan blocku olaraq 16 ədəd IP istifadə eləmişik. Ancaq PUBLIC IP ünvan əvəzinə local aralıqdan istifadə edəcəyik(10.0.0.0/255.255.255.240).

- 10.0.0.18: Bu serverin VPN ünvanı kimi istifadə ediləcək
- 10.0.0.19: Götürülə bilməz
- 10.0.0.20-10.0.0.25: VPN clientlər üçündür
- 10.0.0.26: Götürülə bilməz
- 10.0.0.27: OpenVPN serverin özünün LAN ünvanıdır
- 10.0.0.28-10.0.0.29: Götürülə bilməz
- 10.0.0.30: Remote LAN-da olan router

Necə edək...

1. **example12-5-server.conf** adlı server quraşdırmasını yaradaq və içinə aşağıdakı sətirləri əlavə edək:

```
mode server
tls-server
proto udp
port 1194
dev tun

ifconfig 10.0.0.18 255.255.255.240
ifconfig-pool 10.0.0.20 10.0.0.25
push "route 10.0.0.27 255.255.255.255 net_gateway"
push "route-gateway 10.0.0.30"
push "redirect-gateway def1"
```

```
ca /usr/local/etc/openvpn/ca.crt
cert /usr/local/etc/openvpn/openvpnsrvr.crt
key /usr/local/etc/openvpn/openvpnsrvr.key
dh /usr/local/etc/openvpn/dh2048.pem
tls-auth /usr/local/etc/openvpn/ta.key 0

persist-key
persist-tun
keepalive 10 60

topology subnet
push "topology subnet"

script-security 2
client-connect /usr/local/etc/openvpn/proxyarp-connect.sh
client-disconnect /usr/local/etc/openvpn/proxyarp-disconnect.sh

user root
group wheel

daemon
log-append /var/log/openvpn.log
```

2. Sonra `/usr/local/etc/openvpn/proxyarp-connect.sh` scriptini yaradın və içinə aşağıdakı sətirləri əlavə edin:

```
#!/usr/local/bin/bash
/usr/sbin/arp -s $ifconfig_pool_remote_ip auto pub
/sbin/route add ${ifconfig_pool_remote_ip}/32 -interface tun0
```

3. Uyğun olaraq `/usr/local/etc/openvpn/proxyarp-disconnect.sh` scriptini yaradın içinə aşağıdakı sətirləri əlavə edin:

```
#!/usr/local/bin/bash
/usr/sbin/arp -d $ifconfig_pool_remote_ip
/sbin/route del ${ifconfig_pool_remote_ip}/32 -interface tun0
```

4. Əmin olun ki, hər iki script yerinə yetiriləndir və sonra serveri işə salın:

```
root@siteA:/ # cd /usr/local/etc/openvpn
root@siteA:/usr/local/etc/openvpn # chmod 755 proxyarp-connect.sh
proxyarp-disconnect.sh
root@siteA:/usr/local/etc/openvpn # openvpn --config example12-5-
server.conf
```

5. Sonra client-i işə salın. Client-ə mənimsədilən IP ünvan **10.0.0.20** olmalıdır.

6. Sonda <http://www.whatismyip.com> linkinə daxil olun və öz IP ünvanınızı yoxlanış edin.

Bu necə işləyir...

Server quraşdırma faylında olan direktivlərin bəzilərini açıqlayaq:

```
ifconfig 10.0.0.18 255.255.255.240
ifconfig-pool 10.0.0.20 10.0.0.25
```

Client-lər istifadə edəcəyi PUBLIC IP ünvanların aralığının təyinatı. Ona görə ki, /28 block-unda bu aralıqda olan bütün IP ünvanlar istifadə edilə bilməz:

```
server 10.0.0.18 255.255.255.240
```

Növbəti sətir həmçinin VPN serverin adi şəbəkə üzərindən görünməsinin əminliyi üçün istifadə edilir (VPN tunelin üzərindən yox):

```
push "route 10.0.0.27 255.255.255.255 net_gateway"
```

Qayda ilə bütün trafikə VPN tunel üzərindən ötürülməsi üçün bizim ehtiyacımız vardır ki, açıq şəkildə yeni susmaya görə olan marşrutu və **redirect-gateway**-i göstərək:

```
push "route-gateway 10.0.0.30"
```

```
push "redirect-gateway def1"
```

Normal halda bu quruluşda aşağıdakı sətir həmişə tələb olunur ki, clientlərə ötürək:

```
topology subnet
```

Ancaq, biz server direktivi istifadə etmədiyimizə görə, bu avtomatik baş vermir. Ona görə də açıq şəkildə topology direktivinin istifadəsilə biz əmin oluruq ki, clientlər doğru quraşdırmaları əldə ediblər.

client-connect və **client-disconnect** scriptləri 2-ci başlıqda istifadə edilən client-server IP şəbəkələrində istifadə edilən Proxy-ARP misalında istifadə edilənlərə oxşayır.

Daha da ətraflı...

topology subnet imkanı OpenVPN2.1-də yaradılmışdır. Bu imkan olmadan hər bir client /30 şəbəkəsində olardı hansı ki, hər bir client 4 IP ünvan istifadə edəcəkdə. Bu hər bir client üçün IP ünvanının istifadəsində çox baha qiymətə gətirib çıxarır.

Həmçinin baxın

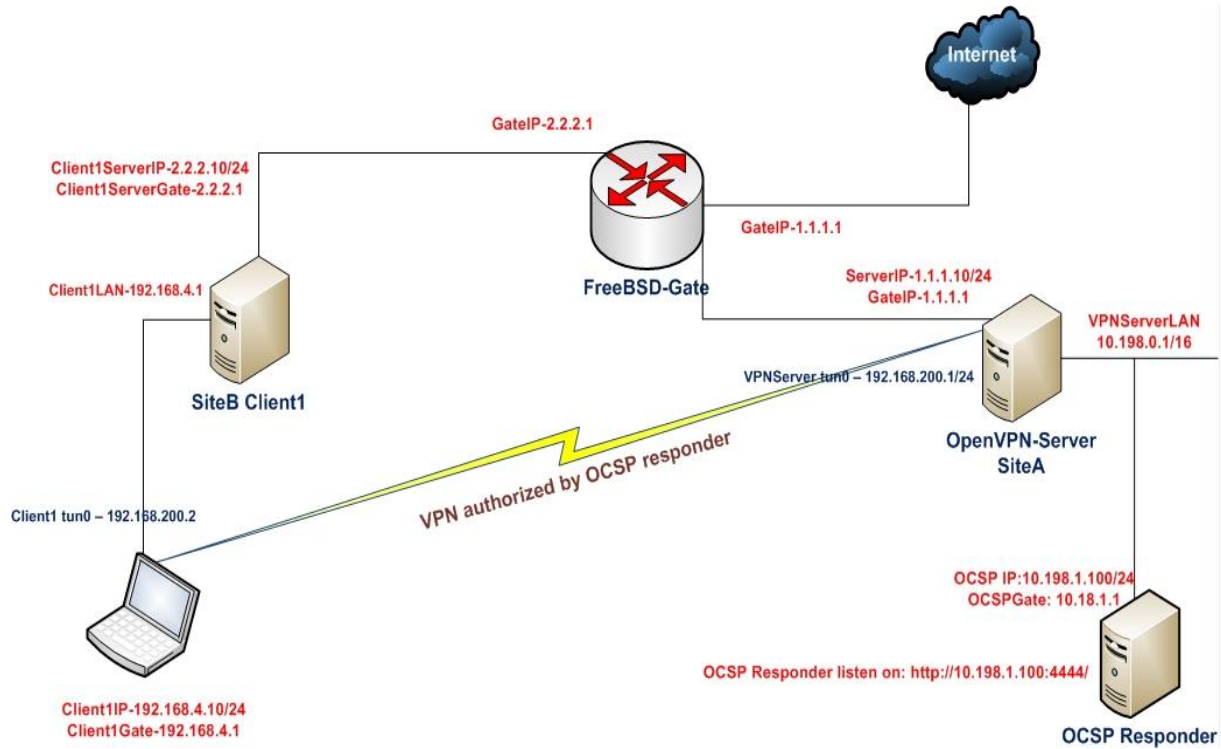
- 2-ci başlıqda Proxy-ARP misalı hansı ki, UNIX şəbəkələrində proxy-arp-in istifadəsinin detallarını açıqlayırdı.

OCSP dəstəklənməsi

Kiçik fakt odur ki, client sertifikatının serial rəqəmi scriptlərin mühit dəyişənlərində mövcud olur və OpenVPN-in Online Certificate Status Protocol (OCSP)-u ilə işləməsinə şərait yaradır. Bu misal OpenVPN serverdə OCSP-nin necə qurulub istifadə edilməsini göstərəcək.

İşə hazırlaşaq

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda CA işləyən serverdə CA sertifikatı və CA key-i özündə saxlayır və adı **ocsp.example.com** olacaq. Bu misalda server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Client maşını isə Windows7 x64 OpenVPN2.3-də olacaq. Server quraşdırması olaraq 2-ci başlıqda server-tərəf routing üçün yaratdığımız **basic-udp-server.conf** faylından istifadə edəcəyik. Həmçinin eyni başlıqda yaratdığımız **basic-udp-client.ovpn** quraşdırması faylını client üçün istifadə edəcəyik.

Necə edək...

1. Öncə 2-ci başlıqda client-server IP şəbəkələri üçün yaratdığımız PKI-ləri istifadə edərək OCSP serveri işə salaq. OCSP serverin IP ünvanı **10.198.1.100**-dür və Gateway-i OpenVPN serverdir. OpenVPN serverdən OCSP serverə lazımı faylları nüsxələyək. **ocsp.example.com** maşınında aşağıdakı əməlləri işə salın (OCSP responder serverin yüklənilməsi haqqında daha ətraflı **OpenSSL-OCSP-Responder.docx** sənədindən oxuya bilərsiniz):

```
root@siteA:~ # cd /usr/local/etc/openvpn/
root@siteA:/usr/local/etc/openvpn # scp itvpn/keys/index.txt
10.198.1.100:/root/certs/
root@siteA:/usr/local/etc/openvpn # scp itvpn/keys/ca.crt
itvpn/keys/ca.key 10.198.1.100:/root/certs/
```

Sonra OCSP serverə gedib OCSP-ni işə salaq:

```
root@ocsp:~/certs # cd /root/certs/
root@ocsp:~/certs # openssl ocspp -index index.txt -port 4444 -CA ca.crt
-rsigner ca.crt -rkey ca.key -resp_text
Enter pass phrase for ca.key:
```

Waiting for OCSP client connections...

2. Sonra OpenVPN serverdə **basic-udp-server.conf** quraşdırma faylını **example12-6-server.conf** quraşdırma faylına nüsxələyin və **example12-6-server.conf** faylının sonuna aşağıdakı sətirləri əlavə edin:

```
script-security 2  
tls-verify /usr/local/etc/openvpn/example12-6-ocsp.sh
```

OpenVPN Server server maşınının **/etc/hosts** maşınına aşağıdakı sətiri öncədən əlavə edirik:

```
10.198.1.100 ocsp.example.com
```

3. **tls-verify** scriptini **/usr/local/etc/openvpn/example12-6-ocsp.sh** yeni yaradaq. Öncədən demək istərdim ki, FreeBSD-də OpenVPN portlardan yükləndikdən sonra portun kompilyasiya etdiyi dataları silməyin çünki, orda OCSP haqqında **script** yüklənir ki, indi bizə lazım olacaq. Və hal-hazırda həmin ünvandan o scripti OpenVPN quraşdırma qovluğuna nüsxələyəcəyik:

```
root@siteA:/usr/local/etc/openvpn # cp  
/usr/ports/security/openvpn/work/openvpn-  
2.3.2/contrib/OCSP_check/OCSP_check.sh example12-6-ocsp.sh
```

Faylın tərkibi aşağıdakı kimi olacaq:

```
root@siteA:/usr/local/etc/openvpn # cat example12-6-ocsp.sh  
#!/bin/sh
```

```
# OCSP Serverin qulaq asdığı ad və port  
ocsp_url="http://ocsp.example.com:4444/"
```

```
# CA Sertifikatın ünvanı  
issuer="/usr/local/etc/openvpn/ca.crt"  
nonce="-nonce"
```

```
# CA Sertifikatın ünvanı  
verify="/usr/local/etc/openvpn/ca.crt"  
check_depth=0  
cur_depth=$1  
common_name=$2
```

```
err=0  
if [ -z "$issuer" ] || [ ! -e "$issuer" ]; then  
    echo "Error: issuer certificate undefined or not found!" >&2  
    err=1  
fi  
  
if [ -z "$verify" ] || [ ! -e "$verify" ]; then  
    echo "Error: verification certificate undefined or not found!" >&2  
    err=1  
fi  
  
if [ -z "$ocsp_url" ]; then  
    echo "Error: OCSP server URL not defined!" >&2
```

```
err=1
fi

if [ $err -eq 1 ]; then
    echo "Did you forget to customize the variables in the script?" >&2
    exit 1
fi

if [ $check_depth -eq -1 ] || [ $cur_depth -eq $check_depth ]; then

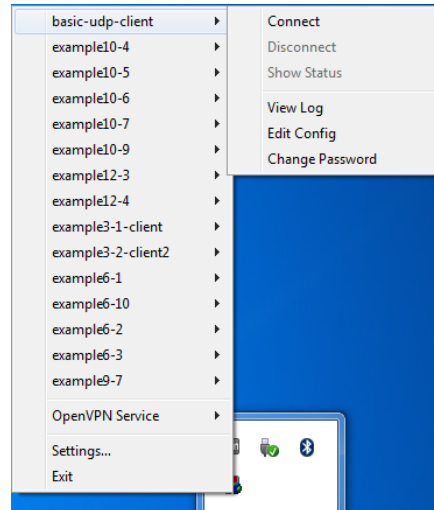
    eval serial="\$tls_serial_${cur_depth}"

    if [ -n "$serial" ]; then
        status=$(openssl ocsp -issuer "$issuer" \
            "$nonce" \
            -CAfile "$verify" \
            -url "$ocsp_url" \
            -serial "0x${serial}" 2>/dev/null)

        if [ $? -eq 0 ]; then
            # check that it's good
            if echo "$status" | grep -Fq "0x${serial}: good"; then
                exit 0
            fi
        fi
        exit 1
    fi
fi
```

Bu script **OCSP_verify.sh** scriptinə əsaslanır və bu UNIX maşının **openvpn-2.3.2**-sinin **contrib** qovluğundan **example12-6-ocsp.sh** adında scriptə nüsxələnmişdir.

4. Əmin olun ki, script yerinə yetiriləndir və sonra serveri işə salın:
root@siteA:/usr/local/etc/openvpn # **chmod 755 example12-6-ocsp.sh**
root@siteA:/usr/local/etc/openvpn # **openvpn --config example12-6-server.conf**
5. **basic-udp-client.ovpn** quraşdırma faylından istifadə edərək Windows client-i işə salın:



6. Serverdə `/var/log/openvpn.log` jurnal faylında yoxlanış edin və yoxlayın, aşağıdakı sətir əmələ gəlməlidir:
- ```
Fri Apr 4 09:58:57 2014 us=778780 2.2.2.10:51269 VERIFY OK: depth=1,
C=AZ, O=Itvpn, CN=Itvpn CA, emailAddress=openvpn-ca@domain.lan
Fri Apr 4 09:58:57 2014 us=813449 2.2.2.10:51269 VERIFY SCRIPT OK:
depth=0, C=AZ, O=Itvpn, CN=openvpnclient2, emailAddress=openvpn-
ca@domain.lan
Fri Apr 4 09:58:57 2014 us=813519 2.2.2.10:51269 VERIFY OK: depth=0,
C=AZ, O=Itvpn, CN=openvpnclient2, emailAddress=openvpn-ca@domain.lan
```

7. OCSF serverdə console-da aşağıdakı sətirlər çap edilməlidir:

```
OCSF Response Data:
 OCSF Response Status: successful (0x0)
 Response Type: Basic OCSF Response
 Version: 1 (0x0)
 Responder Id: C = AZ, O = Itvpn, CN = Itvpn CA, emailAddress =
openvpn-ca@domain.lan
 Produced At: Apr 4 03:23:11 2014 GMT
 Responses:
 Certificate ID:
 Hash Algorithm: sha1
 Issuer Name Hash: C75292942ADFED35D708E1138F6BDC2DEDFA7069
 Issuer Key Hash: B41F428AB4C39AB53ACBC8D391D0FDB65FDCE6A4
 Serial Number: 03
 Cert Status: good
 This Update: Apr 4 03:23:11 2014 GMT

Response Extensions:
 OCSF Nonce:
 0410D244AF4C5F212714922602FC6058A867
 Signature Algorithm: sha1WithRSAEncryption
 6b:d3:38:13:1d:a5:bd:60:8e:67:7d:0d:34:84:32:bc:f7:93:
 c3:66:77:c6:54:09:a1:a3:9a:aa:ff:69:6d:7d:b4:94:94:a7:
 1b:a4:10:d9:a9:15:cf:b2:a9:60:01:4f:01:58:ef:db:a4:79:
 81:36:9c:47:db:9d:44:6a:aa:10:32:5b:79:66:1a:f7:01:ca:
 fe:f2:f5:59:fa:0e:32:9e:37:8b:b1:a6:a3:bb:7a:ba:06:bc:
```

```
f2:70:93:1c:4f:02:9b:db:de:e8:27:14:65:95:7b:d8:1a:2c:
7b:51:83:b8:c2:34:ff:ec:74:bf:44:62:65:24:17:36:9b:8e:
f5:fc:c0:a8:81:cf:e9:6a:eb:71:d4:70:33:e8:79:0c:81:8e:
e0:5b:72:a5:db:64:86:ef:25:32:30:21:c9:b2:8a:45:8a:20:
c2:01:c1:c1:50:36:16:99:ca:64:53:84:b0:37:57:bd:04:eb:
43:63:aa:5b:02:17:38:b7:fa:cf:80:d0:2a:ac:f0:4b:90:78:
b1:38:f5:7a:1f:54:5f:27:37:fc:1e:60:d3:f9:c2:a5:3b:39:
68:11:f9:99:b8:01:2c:fc:ee:54:34:42:27:5e:11:2f:5c:36:
0e:d8:37:c6:28:cd:a4:ce:3e:76:b5:58:30:18:45:45:46:28:
b9:1d:28:1d:1c:6c:e0:c8:fa:2d:b3:74:4e:d5:24:aa:7a:99:
99:69:ca:f4:47:eb:b0:40:f9:35:2c:98:3d:bd:b9:19:d9:f1:
28:d8:15:8b:ed:83:53:7a:37:3e:77:40:60:a0:fa:b5:df:2d:
30:1d:63:1e:6e:a9:60:66:a8:43:eb:0c:bb:b4:f3:0f:3a:fc:
e4:d2:a9:e3:96:1b:8e:85:cd:2b:ec:70:9e:53:a8:30:60:20:
5c:dc:e6:a2:e8:7b:d8:3d:f5:0e:cc:31:4a:58:16:6e:e6:23:
bc:cd:d2:51:08:39:fb:4d:00:d2:ed:5d:97:45:dd:32:7e:75:
a6:30:99:53:d9:e6:6b:ec:7b:bb:97:18:cf:dc:aa:57:6a:be:
45:9c:c5:a2:0f:58:24:c0:90:84:0c:14:b2:76:e2:d4:c6:73:
2c:88:1e:21:7b:7c:0b:a4:ec:ac:e4:a4:08:0d:37:5d:74:e2:
51:9a:1e:c2:b6:79:a0:de:85:0c:27:15:61:ef:14:72:d7:f4:
b0:06:01:bf:c7:40:70:88:63:68:ad:25:a9:d7:11:08:00:0a:
ee:b1:4c:67:f2:a6:55:6f:30:2b:02:4e:be:89:5a:47:9a:dc:
8d:0c:23:70:59:df:63:f9:76:0a:5c:5f:04:3d:77:d7:1a:b8:
1c:4d:bc:6f:26:84:33:ae
```

Certificate:

Data:

Version: 3 (0x2)

Serial Number:

be:84:5e:83:d5:e0:ab:34

Signature Algorithm: sha1WithRSAEncryption

Issuer: C=AZ, O=Itvpn, CN=Itvpn CA/emailAddress=openvpn-ca@domain.lan

Validity

Not Before: Jan 16 04:05:40 2014 GMT

Not After : Jan 14 04:05:40 2024 GMT

Subject: C=AZ, O=Itvpn, CN=Itvpn CA/emailAddress=openvpn-ca@domain.lan

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (4096 bit)

Modulus (4096 bit):

```
00:c8:12:15:81:3f:80:cc:46:50:13:3d:1c:cd:01:
1b:0d:79:10:2f:95:42:68:9a:8b:a5:4e:ec:62:63:
8f:9b:37:b9:db:f4:59:dc:c6:e1:60:5d:2c:5f:32:
1b:52:93:af:eb:9e:42:d6:c8:a7:f6:2f:01:f0:01:
43:8a:fd:99:9a:ed:3a:0b:ee:70:0e:30:8b:86:5b:
32:74:5a:e3:b7:f7:e2:1f:58:f5:3d:3c:d9:5b:89:
cc:09:9c:29:60:10:09:f7:ca:20:49:0d:52:97:80:
99:0c:6f:35:f7:c8:fb:9e:ad:99:f0:ed:53:23:5d:
e7:1b:81:36:0c:54:45:37:da:4d:4a:eb:c1:99:53:
fc:54:77:b6:79:70:02:45:1f:69:ba:0c:a7:5d:8a:
68:ce:b1:13:6f:30:a9:c0:14:d5:ad:10:2a:60:04:
16:1b:e8:53:ac:1b:df:5a:95:da:20:1f:b9:a1:3d:
```

```
42:04:35:e7:04:b5:62:a3:ea:89:42:d7:b1:00:4d:
26:bf:23:b8:f4:86:71:3a:91:d7:c0:44:99:7a:c1:
04:d7:d2:a1:b6:99:c2:10:61:e2:26:83:e7:f5:e8:
39:90:9f:24:2c:6a:49:8b:41:df:81:e2:0b:0b:ef:
d9:81:a8:52:8e:f9:98:b8:33:03:9e:3e:9e:eb:6d:
e2:fd:35:56:50:ea:ca:ab:db:13:9c:85:68:1e:8f:
84:fe:7f:6e:e7:91:cc:41:02:58:db:96:65:23:fe:
2b:0d:9a:3f:d9:1f:04:d7:48:7e:6e:d5:e9:83:55:
ea:58:8d:bc:ac:3f:2e:5a:5b:2a:5c:a8:8c:81:db:
e4:57:bb:6a:21:11:9b:e1:4e:ed:54:bc:ff:4e:7a:
46:bf:0e:32:27:0d:50:53:94:30:f9:ec:d2:87:a9:
1f:dd:df:29:03:7c:30:e1:01:94:e7:1d:9d:90:29:
fa:81:7d:5a:bb:36:31:7c:59:de:96:3d:c3:b3:06:
d5:71:8f:88:6d:09:ae:62:4a:5b:53:e2:7f:d9:bb:
dc:17:d1:ec:0c:1e:e1:be:fc:82:74:e5:ba:c1:97:
10:d5:29:5a:66:6f:2b:ad:8a:02:7c:ba:33:85:6c:
b3:70:44:62:53:d9:3a:4c:d2:fc:a1:1f:2c:61:b0:
0c:21:ae:0e:a5:32:b9:dc:2c:28:9a:e5:a9:3b:c4:
68:20:1b:77:97:44:a5:e2:69:31:3a:31:f3:92:02:
59:a8:62:f1:cd:7e:0c:1a:ec:e5:76:b2:2c:5f:27:
fb:fe:be:e5:74:84:25:b9:49:13:6c:db:99:d2:05:
9f:21:db:e4:34:9e:e2:fe:14:8b:7c:1a:cc:e7:75:
88:35:31
```

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Subject Key Identifier:

B4:1F:42:8A:B4:C3:9A:B5:3A:CB:C8:D3:91:D0:FD:B6:5F:DC:E6:A4

X509v3 Authority Key Identifier:

keyid:B4:1F:42:8A:B4:C3:9A:B5:3A:CB:C8:D3:91:D0:FD:B6:5F:DC:E6:A4

DirName:/C=AZ/O=Itvpn/CN=Itvpn CA/emailAddress=openvpn-

ca@domain.lan

serial:BE:84:5E:83:D5:E0:AB:34

X509v3 Basic Constraints:

CA:TRUE

Signature Algorithm: sha1WithRSAEncryption

```
41:dd:2c:60:9d:32:d5:1d:46:63:fa:00:00:9d:22:89:d5:82:
3d:7d:c6:48:f1:27:cc:3f:44:fa:e0:af:d6:05:a7:06:e0:8a:
ea:1d:48:7a:ee:82:72:fd:e7:3a:2d:17:ba:9f:13:0a:5d:f0:
72:b6:12:ec:fd:0b:eb:02:0a:30:c1:c1:00:d9:83:8b:89:08:
4e:1e:e2:3f:08:ff:bf:3c:2b:b3:0c:7c:da:ea:07:8d:70:88:
cc:95:14:79:79:7f:a1:51:33:f0:e6:ff:ff:5c:3b:09:f2:76:
78:7c:69:16:3a:52:51:78:5a:6e:b7:8e:ca:b3:93:b6:38:c6:
c8:b2:e7:2c:a2:5e:8d:a3:a5:72:2b:4b:50:06:78:33:ca:ac:
7e:cf:1e:1d:51:e7:7c:d8:ca:c0:02:59:5e:6f:e6:2c:87:f4:
05:eb:01:68:ae:be:04:bb:22:26:55:6f:75:10:c7:5c:42:70:
c2:41:db:f6:55:0b:48:cd:40:27:d7:1d:0d:8c:01:31:e8:f5:
c8:b2:96:5d:e0:1e:b4:1f:1f:0b:05:bf:2f:60:1c:be:a4:a9:
ac:9f:d6:db:e4:07:6e:d3:22:da:9d:d3:7b:74:2d:42:19:09:
71:bd:4f:9e:27:32:43:d2:d4:d3:1a:5e:94:b7:ce:b6:27:37:
da:8c:34:33:fd:15:8b:2f:1a:40:80:1e:64:09:2f:5b:59:36:
```

43:cb:7f:f1:68:f7:c8:04:8a:7c:10:69:2d:47:fd:30:f7:2b:  
97:f7:46:82:a4:4d:cc:0c:5e:32:c8:01:18:ad:ab:78:be:2c:  
70:9d:2b:7b:ac:ac:26:64:1a:d3:2f:9c:d3:42:b7:ae:2a:78:  
27:20:b4:c2:35:38:a7:7a:f1:3c:08:9c:16:b6:9d:09:35:d1:  
00:a5:57:3f:18:cb:4c:db:a3:d7:70:47:5c:87:02:9a:f6:33:  
bd:b4:71:af:a2:2f:51:26:6a:8d:81:9c:99:34:f2:52:8d:c1:  
85:a2:42:4b:68:48:d1:6f:b4:93:ba:f7:25:a6:3b:38:f0:af:  
28:ea:63:8f:57:1f:76:fc:3e:55:88:1f:85:0b:f8:43:20:b2:  
3c:d5:fa:66:a5:37:cc:54:e2:45:d3:97:7e:ab:67:e5:aa:e7:  
f1:2d:97:65:92:dc:94:b3:b7:ab:62:53:01:f4:06:11:6e:58:  
6e:ff:e9:30:34:3c:ec:51:40:fb:76:f6:9c:62:48:25:a0:46:  
bd:48:a7:74:b8:96:10:ff:a5:4c:38:b8:72:4c:c6:1d:de:e0:  
c3:c8:d0:a9:62:a3:9c:59:16:b0:23:40:0a:b0:c9:1a:1a:11:  
f7:89:73:11:37:12:c1:76

-----BEGIN CERTIFICATE-----

MIIF8DCCA9igAwIBAgIJAL6EXoPV4Ks0MA0GCSqGSIb3DQEBBQUAMFgx CzAJBgNV  
BAYTAK5MMREwDwYDQOKEwhDb29rYm9vazEUMBIGA1UEAxMLQ29va2Jvb2sgQ0Ex  
IDAeBgkqhkiG9w0BCQEWEW9wZW52cG4tY2Y2FAYXRSLmF6MB4XDTE0MDExNjA0MDU0  
MFoXDTE0MDExNDA0MDU0MFowWDELMAkGA1UEBhMCTkwETAPBgNVBAoTCENvb2ti  
b29rMRQwEgYDQOQDEwtDb29rYm9vayBDQTEgMB4GCSqGSIb3DQEJARYRb3B1bnZw  
biljYUBhdGwYXowggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDIEhWB  
P4DMRlATPRzNARsNeRAvlUJomoulTuxiY4+bN7nb9FncxuFgXSxfMhtSk6/rnkLW  
yKf2LwHwAUOK/Zma7ToL7nAOMIuGWzJ0Wu039+IfWPU9PNlbi cwJnClgEAn3yiBJ  
DVKXgJkMbzX3yPuerZnw7VMjXecbgTYMVEU32k1K68GZU/xUd7Z5cAJFH2m6DKdd  
imjOsRNvMKnAFNWtECpgBBYb6FOsG99aldogH7mhPUIENecEtWKj6olC17EATSa/  
I7j0hnE6kdfARJl6wQTX0qG2mcIQYeImg+f16DmQnyQsakmLQd+B4gsL79mBqFKO  
+Zi4MwOePp7rbeL9NVZQ6sqr2xOchWgej4T+f27nkcxBALjblmUj/isNmj/ZHwTX  
SH5ulemDVepYjbyPy5aWypcqIyB2+RXu2ohEZvhTu1UvP9Oeka/DjInDVBT1DD5  
7NKHqR/d3ykDfDDhAZTnHZ2QKfqBfVq7NjF8Wd6WPcOzBtVxj4htCa5iSlT4n/Z  
u9wX0ewMHuG+/IJ05brBlxDVKVpmbbyutigJ8ujOfbLNwRGJT2TpM0vyhHyxhsAwh  
rg6lMrncLCia5ak7xGggG3eXRKXiaTE6MfOSAlmoYvHNfgwa7OV2sixfJ/v+vuV0  
hCW5SRNs25nSBZ8h2+Q0nuL+Fit8GszndYg1MQIDAQABo4G8MIG5MB0GA1UdDgQW  
BBS0H0KKtMOatTrLyNOR0P22X9zmpDCBiQYDVR0jBIGBMH+AFLQfQoq0w5q1OsvI  
05HQ/bZf3OakoVykWjBYMQswCQYDQOGEwJOTDERMA8GA1UEChMIQ29va2Jvb2sx  
FDASBgNVBAMTC0Nvb2tib29rIENBMSAwHgYJKoZIhvcNAQkBFhFvcGVudnBuLWNh  
QGf0bC5heoIJAL6EXoPV4Ks0MAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEFBQAD  
ggIBAeHdLGCdMtUdRmP6AACdIonVgj19xkxj8w/RPrgr9YFpwbgiuodSHrugNL9  
5zotF7qfEwPd8HK2Euz9C+sCCjDBwQDZg4uJCE4e4j8I/788K7MMfNrqb41wiMyV  
FHl5f6FRM/Dm//9cOwnydnh8aRY6UlF4Wm63jsqzk7Y4xsiy5yyiXo2jpxIrS1AG  
eDPKRh7PHh1R53zYysACWV5v5iyH9AXrAWiuvgs7IiZVb3UQx1xCcMJb2/ZVC0jN  
QCfXHQ2MATHo9ciyl13gHrQfHwsFvy9gHL6kqayf1tvkB27TItqd03t0LUIZCXG9  
T54nMkPS1NMaXpS3zrYnN9qmNDP9FYsvGkCAHmQJL1tZNkPLf/Fo98gEinwQaS1H  
/TD3K5f3RoKkTcwmXjLIARitq3i+LHCdK3usrCZkGtMvnnNct64qeCcgTMI1OKd6  
8TwInBa2nQk10QC1Vz8Yy0zbo9dwr1yHApr2M720ca+iL1Emao2BnJk081KNwYWi  
QktoSNFvtJO69yWmOzjwryjqY49XH3b8P1WIH4UL+EMgsjzV+malN8xU4kXT136r  
Z+Wq5/Et12WS3JSzt6tiUwH0BhFuWG7/6TA0POxRQPt29pxiSCWgRr1Ip3S41hD/  
pUw4uHJMxh3e4MPI0Klio5xZFrAjQAqwyRoaEfeJcxE3EsF2

-----END CERTIFICATE-----

### Bu necə işləyir...

Client sertifikatının seriya nömrəsi hal-hazırda mühit dəyişəni `tls_serial_0-` da mövcuddur və bizə kömək edir ki, OCSP dəstəklənsin. Yoxlanış müraciətini OCSP serverə göndərdikdə, biz əmin olmaq istəyirik ki, həqiqətən qoşulmaq istəyən istifadəçi bizim CA tərəfindən imzalanıb və keçərlidirmi (yeni **revoked** deyil ki?) Biz bunu həmçinin **Certification Revocation List (CRL)** ilə də edə bilərik ancaq, OCSP-nin gözəlliyi ondan ibarətdir ki, heç bir client-ə CRL list ötürülmür.

### Həmçinin baxın

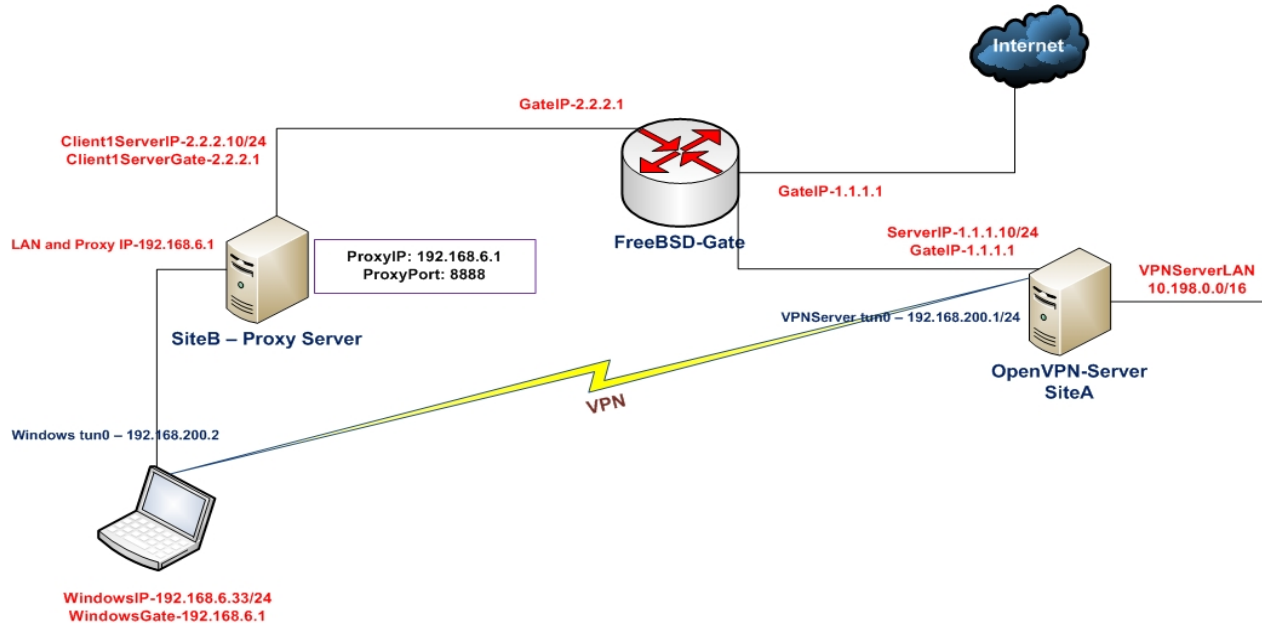
- 4-cü başlıqda CRL-lərin istifadəsi hansı ki, Certificate Revocation List (CRL)-i açıqlayır. Əksər standart metodlarda hələ də sertifikatın yoxlanışı üçün CRL istifadə edilir.

### 'x509\_user\_name' parametri

`x509_user_name` parametrinin gözəlliyi ondan ibarətdir ki, bu bizə izin verir ki, x509 sertifikatların istifadəsində bizə `/CN=` elementinin istifadə edilməsinə şərait yaradır. Bu əksər hallarda kənar üçüncü tərəfin sertifikatının istifadəsi və ya hansısa digər avtorizasiya sistemlərində olan inteqrasiya vaxtı tələb edilir.

### İşə hazırlaşaq

Aşağıdakı şəbəkə quruluşundan istifadə edəcəyik:



2-ci başlıqda yaratdığımız client və server sertifikatlarını burda da istifadə edəcəyik. Bu misalda server maşını FreeBSD9.2 x64 OpenVPN2.3-də olacaq. Client maşını isə Windows7 x64 OpenVPN2.3-də olacaq. Server quraşdırması olaraq 2-ci başlıqda server-tərəf routing üçün yaratdığımız **basic-udp-server.conf** faylından istifadə edəcəyik. Həmçinin eyni başlıqda



yaratdığımız **basic-udp-client.ovpn** quraşdırma faylını client üçün istifadə edəcəyik.

### Necə edək...

İlk olaraq OpenSSL-in asan əmrləri yeni sertifikat generasiya edək. Bu ona görədir ki, OpenVPN-in istifadə elədiyi **easy-rsa** scriptləri bizə asan imkan yaratmır ki, /CN= hissəsi olmadan script yaradaq.

1. Specific subject adı ilə yeni sertifikat müraciətini generasiya edək:  

```
root@siteA:/usr/local/etc/openvpn # openssl req -new -nodes -keyout
openvpnclient6.key -out openvpnclient6.csr -newkey rsa:2048 -subj
"/C=AZ/O=ATL/UID=atl"
```
2. Boş olan OpenSSL genişlənmə faylı yaradıb əmin olaq ki, sertifikatın nəticəsi X.509 v3-dür:  

```
root@siteA:/usr/local/etc/openvpn # touch openssl-ext.conf
```
3. Və sonda CA key istifadə edərək sertifikat müraciətini imzalayaq:  

```
root@siteA:/usr/local/etc/openvpn # openssl x509 -req -CA ca.crt -CAkey
ca.key -in openvpnclient6.csr -set_serial 0xAA -sha1 -days 1000 -
extfile openssl-ext.conf -out openvpnclient6.crt
```
4. Server üçün **basic-udp-server.conf** quraşdırma faylını **example12-7-server.conf** quraşdırma faylına nüsxələyirik və **example12-7-server.conf** quraşdırma faylının sonuna aşağıdakı sətirləri əlavə edirik:  

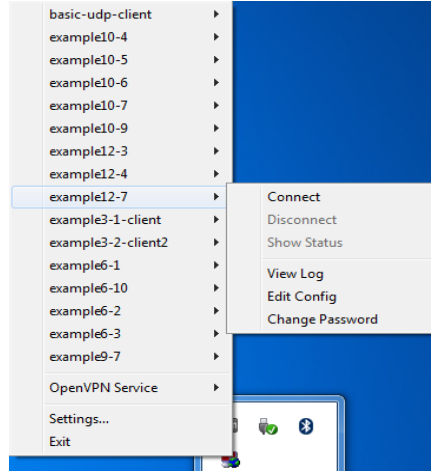
```
script-security 2
client-connect /usr/local/etc/openvpn/example12-7-client-
connect.sh

verify-x509-name "UID"
```
5. Sonra **client-connect** üçün **/usr/local/etc/openvpn/example12-7-client-connect.sh** scriptini yaradıb içinə aşağıdakı sətirləri əlavə edək:  

```
#!/usr/local/bin/bash
echo "common_name = [$common_name]"
```
6. Əmin olun ki, script yerinə yetiriləndir və serveri işə salın:  

```
root@siteA:/usr/local/etc/openvpn # chmod 755 example12-7-client-
connect.sh
root@siteA:/usr/local/etc/openvpn # openvpn --config example12-7-
server.conf
```
7. **openvpnclient6.crt** və **openvpnclient6.key** fayllarını təhlükəsiz kanalla Windows7 maşına nüsxələyin (WinSCP yada pscp).
8. Client üçün **basic-udp-client.ovpn** quraşdırma faylını **example12-7.ovpn** quraşdırma faylına nüsxələyək və **example12-7.ovpn** faylının içində sertifikatlar və açarın ünvanı aşağıdakılarla əvəz edək:  

```
cert /usr/local/etc/openvpn/openvpnclient6.crt
key /usr/local/etc/openvpn/openvpnclient6.key
```
9. Sonra client-i işə salın:



10. Sonra OpenVPN server maşında `/var/log/openvpn.log` faylını yoxlayın aşağıdakı sətir olmalıdır:

```
common_name = [itvpn]
```

### **Bu necə işləyir...**

Yeni **verify-x509-name** direktivi ilə OpenVPN client qoşulan client-in istifadəçi adını susmaya görə olan `/CN=` sütunu yox digər sütundan əldə edə bilər. Bu misalda `/UID=` istifadə edildi. Client sertifikat ilə qoşulduqda onun, `/UID=` adlı subject seksiyası olur hansı ki, OpenVPN server client adını bu sütundan açır. Client-in adi mühit dəyişeni kimi **common\_name** elan edilib necə ki, server-tərəf scriptlərdə və pluginlərdə **client-connect**, **client-disconnect**, **learn-address** və həmçinin **ifconfig-pool-persist** faylında istifadə edildiyi kimi.

**Qeyd:** Ancaq X509 sertifikat-ın istənilən `/<field>=<name>` seksiyası istifadə edilə bilməz. **<field>** BÖYÜK hərflərlə yazılmalıdır və o OpenSSL tərəfindən tanınılmalıdır, digər halda sertifikatın generasiyası mümkün olmayacaq.

### **Daha da ətraflı...**

#### **OpenVPN2.3-də olan bug-da özünün aparması**

Qeyd edim ki, öncə yazdığım misalı OpenVPN2.3-də etdiyim üçün həmin halda BUG var idi və nəticəni OpenVPN2.2-də normal aldım. Ancaq hər hal üçün əgər siz aşağıdakı səhvi görərsinizsə ya sertifikatı düzgün seçməmişiz ya da OpenVPN versiyasında BUG var:

```
Fri Apr 4 22:17:56 2014 us=105846 2.2.2.10:55319 VERIFY ERROR: could not extract CN from X509 subject string ('C=AZ, O=Itvpn, UID=itvpn') -- note that the username length is limited to 64 characters
```

## **İstifadə olunmuş ədəbiyyat siyahısı**

1. OpenVPN Building and Integrating Virtual Private Networks - Markus Feilner
2. OpenVPN 2 Cookbook - Jan Just Keijser
3. Build and integrate Virtual Private Networks using OpenVPN - Markus Feilner
4. <https://openvpn.net/index.php/open-source/documentation.html>
5. <https://www.google.ru/>
6. <https://www.howtoforge.com/>
7. <https://www.freebsd.org/>
8. <http://tldp.org/>

## Kitab haqqında

İstənilən özəl ya da dövlət müəssisəsində baş ofis və ona tabe olan bir neçə filiallar mövcud olarsa, VPN qurmağa ehtiyac yaranacaq. Bu kitabla siz gündəmdə olan VPN tələblərinin qarşılığını açıq qaynaqlı proqram təminatı ilə əldə etmiş olacaqsınız. Kitab OpenVPN açıq qaynaqlı proqram təminatının bacarıqlarını detallarla incələyir.

## Müəllif haqqında

Açıq qaynaqlı proqram təminatlarının qurulması və sistemlərə tətbiqi üzrə geniş təcrübəyə malikdir. Başlıca məqsədi, gənc İT mühəndisləri ilə təcrübi biliklərini bölüşmək, ana dilimizdə bu sahə üzrə ədəbiyyatların hazırlanmasına öz tövhəsini verməkdir. Maraq dairəsi yalnız open source(açıq qaynaqlı) proqram təminatlarının tədqiq edilməsidir.

ISBN 978-9952-8302-4-8



9 789952 830248