LINUX - VPN

VPN; genel ag alt yapisini kullanarak sifreli protokol uzerinden guvenli data transferi yapabilmek icin gelistirilmistir. VPN'de en onemli nokta 2 uc arasinda kurulan tunel yapisidir. Bu tunelden gececek tum paketlerin sifrelenmis olmasi gerekir. Bir ucta sifrelenen paket diger ucta desifre edilerek kullanilir. Gercek zamanli protokol uygulamalarinda; bu olay yavasliga neden olabilir. Bu isler icin gelistirilmis ozel hatlar kullanilmalidir (Lease Line gibi) fakat maliyet ve kolaylik acisindan VPN tercih nedeni olmaktadir.

VPN icin kullanilabilecek yazilimlarin birkaci asagida listelenmistir.

. OpenVPN
. OpenSWAN
. IPSec
. L2TP
. PPTP

bunlardan bazilaridir.

VPN 2 cesit yapilandirilabilir. Ilk olarak bir VPN sunucusu uzerinde VPN istemciler barindirarak; ikinci olarakda Uctan-uca VPN tunelleme yaparak kullanilabilir. Ikinci yontemde Sunucu-Istemci yapisi bulunmaz!

En genel itibari ile; Linux uzerinde gelen Dial-UP baglantilar icin vazgecilmez olan "pppd" ve uzak baglanti icin olan "ssh" yontemlerini kullanarak VPN alt yapisi olusturacagiz. Ilk olarak bir kullanici-grup yaratalim ve kullanici icin bir sudo belirtelim. Daha sonra bu kullanici uzerinden "pppd" protokolunu calistiracagiz.

LOCAL_IP --> 101.101.101.101
REMOTE_IP --> 202.202.202.202

```
server #  cd /etc
server #  groupadd vpn
server #  useradd vpn -g vpn
server #  passwd vpn
server #  echo "vpn ALL=NOPASSWD: /usr/sbin/pppd" >> sudoers
```

"pppd" programi icin root yetkilerine sahip “vpn” isminde bir kullanici olusturduk. “Client” olarak davranacak makina uzerinden; “Server” gibi davranan makinamiza “ssh” yapabilmemiz gerekir. VPN icin kullanacagimiz sifreli protokol; uzak makina uzerinde guvenli oturum acmak icin kullandigimiz “ssh” kabugumuz olacak.

Client makina uzerinde de “vpn” isminde bir kullanici olusturalim.. “vpn” kullanicisi icin “pppd” kullanma iznide vermemiz gerekir.

Yalniz bu kullanici icin “Server” tarafinda oldugu gibi “NOPASSWD” kullanmayacagiz. Cunku iki makina uzerinde de kisitlama yapmaz isek; “vpn” kullanicisi haklarina sahip olan birisi, iki makina arasinda VPN baslatabilir.

Aslinda burada “Server” tarafinda sifre sormamiz gerekir, fakat “pppd” daemon'lari haberlesirken sizin sifreyi girmenizi saglayacak konsolu size yansitmayacak ve anlamsiz bir sekilde iki uc birim arasinda oturum sonlanacaktir. Server tarafinda sifre soramiyorsak, Client tarafinda “root” sifresini isteriz bizde.. Bunu da “sudo” ya “PASSWD” ekleyerek gerceklestirebiliriz.

Simdi “vpn” kullanicisini olusturalim.

```
client #   groupadd vpn
client #   useradd vpn -g vpn
client #   passwd vpn
client #   echo "vpn ALL=PASSWD: /usr/sbin/pppd" >> /etc/sudoers
```

Client makinadan Server makinaya sifresiz bir ssh oturumu yapilandiralim. Erisim haklarinida ayarlayalim..

```
client #   ssh-keygen -t rsa -b 1024
```

Olusturulan “key” icin bir sifre belirtmeyin.. Bos birakarak geciniz.

```
client #   mkdir /home/vpn/.ssh
client #   chown root.vpn /home/vpn/.ssh
client #   chmod 755 /home/vpn/.ssh
client #   cp /root/.ssh/id_rsa  /home/vpn/.ssh/
```

Client makina uzerinde olusturdugumuz "publickey" i Server makina uzerindeki "vpn" kullanicisinin ev dizinine aktaralim. (Guvenli aktarma konusu size kalmistir.)

```
server #   cd /home/vpn/.ssh/
server #   mv  id_rsa.pub authorized_keys
server #   chmod 644 authorized_keys
server #   ls -la ./authorized_keys
-rw-r--r--   1  vpn  root  221  Mar  8  02:49   authorized_keys
```

Olusturdugumuz Client -> Server sifresiz oturumunu kontrol edin.

```
client  #  ssh vpn@{server_makina_ip}
server #  logout
```

Client makina, Server makina uzerinde ssh ile sifresiz oturum acabilecektir. Simdi "pppd" daemon'unu kullanarak Server makina uzerinde bir “ssh” kabugu cagiralim ve bu kabuga bagli “pppd” daemon'ini calistiralim.. Client makina uzerindeki “pppd” daemon'ina “pty” parametresini verecegiz. Iki uc uzerindeki "pppd" daemon'larini bu sekilde konusturmus olacaz.

"pppd" daemon'inin yeteneklerini gorme zamani. “vpn” kullanicisi olarak sisteme login olduktan sonra;

```
client $ sudo  /usr/sbin/pppd  updetach  noauth  passive mru 16384 mtu 16384 \
        pty  "/usr/bin/ssh  -P  vpn@{server_makina_ip} -o  Batchmode=yes \
        sudo  /usr/sbin/pppd  nodetach  notty  noauth  passive" \
        ipparam VPN  192.168.1.2:192.168.1.1

Password:
```

“sudo”da belirttigimiz “PASSWD” parametresinden dolayi, “root” sifresini soracak, dogru verdiginiz takdirde;

```
Using interface ppp0
Connect: ppp0 <--> /dev/pts/0
Deflate (15) compression enabled
local  IP address 192.168.1.2
remote IP address 192.168.1.1
```

Ilk sudo icin bizden "Password:" istedi. Burdaki yetkilendirme client uzerindeki vpn kullanicisi icindir. Buradan yetkiyi aldiktan sonra; publickey kullanarak sunucu uzerinde ssh oturumu sagliyoruz. “sudo” ile Server makina uzerinde "pppd" daemon'ini calistiriyoruz (en onemli nokta burasidir!) Client ustunde calisan "pppd" daemon'ina ait “tty” olarak Server makina uzerindeki “ssh” kabugunu kullanmis oluyoruz..

Ilk once baglantimizi kontrol edelim, sonradan hayal kirikligina ugramayalim.

```
server #  ifconfig ppp0
ppp0    Link encap:Point-to-Point Protocol
        inet addr:192.168.1.1  P-t-P:192.168.1.2  Mask:255.255.255.255
        UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
        RX packets:17 errors:0 dropped:0 overruns:0 frame:0
        TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:3
        RX bytes:236 (236.0 b)  TX bytes:224 (224.0 b)

client #  ifconfig ppp0
ppp0    Link encap:Point-to-Point Protocol
        inet addr:192.168.1.2  P-t-P:192.168.1.1  Mask:255.255.255.255
        UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
        RX packets:17 errors:0 dropped:0 overruns:0 frame:0
        TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:3
        RX bytes:236 (236.0 b)  TX bytes:224 (224.0 b)
```

Server makina ve Client makina uzerinde dis ag bacaklarini “eth0” kabul ediyoruz. Dis ag (internet) uzerinde Sanal IP'ler gezemeyecegi icin testimiz biraz daha kolay olacaktir . Test icin “tcpdump” kullanacagiz.

```
server #  tcpdump -i eth0 -nn host 192.168.1.1      # Konsollardan biri
server #  tcpdump -i ppp0 -nn                       # Baska bir konsol

client #  tcpdump -i eth0 -nn host 192.168.1.1      # Konsollardan biri
client #  tcpdump -i ppp0 -nn                       # Baska bir konsol
```

Basit bir ping paketi yollayalim ve gelen-giden trafigi bacaklarda izleyelim.

```
client #  ping -c1 192.168.1.1
```

Eger ping cekebildiyseniz; Server makina ve Client makina uzerindeki eth0 bacaklarinda (yani internete cikan bacaklar) 192.168.1 ip blogunu da goremiyorsaniz iki nokta arasinda VPN kurulumunuz basarili olmustur.

- Peki tum bu trafik nerden gelip gidecek ?

Butun iletisim Client makinada calistirdigimiz "pppd" daemon'i icin belirttigimiz; ssh oturumu uzerinden tasinacakcaktir. Iki uc arasinda ssh oturumu kaybolmadigi sure boyunca VPN'niniz aktif olarak calisacaktir.

- VPN icin Bandwidth ayarlamasi yapabilir miyim ?

“iproute” paketini kullanarak bandwidth sinirlamasi yapabilirsiniz. Fakat bu normal olarak kullandiginiz “ssh” oturumlarinin da etkilenmesine neden olacaktir!!

- SSH Server'i baska porta bind edip ayni anda birden fazla calistirabilir miyim ?

Tabii ki, “sshd_config” dosyasinin bir yedegini cikartin. “Port” kismini sizin istediginiz sekilde degistirin. Yeni bir tane “sshd” daemon'ina konfigurasyon dosyasi olarak belirterek extradan farkli bir portta “sshd” calistirmis olursunuz.

```
server #  cd /etc/ssh/
server #  cp sshd_config  sshd_config_vpn
server #  vi  sshd_config_vpn   (Port degistirme)
server #  /usr/sbin/sshd -f /etc/ssh/sshd_config_vpn
```

Client uzerinde “pppd” daemon'ini calistirirken “ssh” parametrelerine portu belirtmelisiniz.

```
client $ sudo  /usr/sbin/pppd  updetach  noauth  passive mru 16384 mtu 16384 \
        pty  "/usr/bin/ssh  -P  vpn@{server_makina_ip} -p PORT -o  Batchmode=yes  \
        sudo  /usr/sbin/pppd  nodetach  notty  noauth  passive" \
        ipparam VPN  192.168.1.2:192.168.1.1
```

- Baglantiyi nasil sonlandirabilirim ?

client # kill `cat /var/run/pppX.pid`

seklinde sonlandirabilirsiniz. Server makina uzerinde sonlandirmak isterseniz; bu komut sadece “pppd” daemon'ini sonlandiracaktir. Client makina tarafindan baslatilan “ssh” oturumu sonlanMAmis olacaktir. Client makina uzerinde “pppd” timeout'a dusene kadar calisiyor olarak algilanacaktir. Baglantilarinizda “interface” e bagla kontrol yaparsaniz, Client makina tarafinda baglantilariniz hala aktif sekilde devam ediyor gorunecektir !!

ACIKLAMA: Dokuman da; elimden geldigince minumum kurulmus Linux uzerinde; herhangi bir extra araca veya yazilima ihtiyac olmadan nasil VPN yapilabilecegini anlatmaya calistim.. Kullanimdan kaynakli dogacak sorunlar, tamamen uygulayan sahis veya kurumlara aittir. Konu ile ilgili bilgi ve yardim icin mail atabilirsiniz..

Mehmet CELIK
bsd_daemon@msn.com