VTun

Kapcsoljuk össze otthonunkat és munkahelyünket virtuális magánhálózaton keresztül!

A dotcom-korszak boldog, szép napjaiban egy P2P-programot fejlesztő, induló cég legelső alkalmazottja voltam. Mivel az intranetet és a fejlesztőkörnyezetet az alapjaitól kellett felépítenünk, mindenhol használhattuk a Linuxot. Mint tudjuk, a világ megváltozott, a dotcomok a dodó madár sorsára jutottak. Induló vállalkozásom is így járt, felvásárolta egy nagyobb vállalat, amely Windows-alapon fejlesztett. Bár az új cég elég engedékeny volt, így továbbra is Linuxszal dolgozhattam, de az ezzel kapcsolatos rendszergazdai feladatok is teljesen rám hárultak.

Egyetlen terület volt, ahol komoly nehézséggel kellett szembenéznem: a VPN beállítása. A régi munkahelyemen minden fejlesztőnek volt bemenő SSH-kapcsolata a saját munkaállomására. Az új munkahelyemen nemcsak az SSH-kapukat zárták le, de a rendszeresített VPN-megoldás sem volt Linux-barát. A tehetetlenség törvényéből fakadóan várható volt, hogy a több felületen is működőképes megoldások, például a FreeS/WAN, nem fognak egyhamar bevezetésre kerülni. Szerencsére a VTun, a régi munkahelyemen használt VPN-megoldás elég rugalmas volt ahhoz, hogy ebben a barátságtalan környezetben is helytálljon.

Hogyan működik?

A VTun úgy működik, hogy az IP-alagutazást észrevétlenül a meglévő csomagtovábbító programokon keresztül valósítja meg. A unixos moduláris szemléletnek megfelelően a VTun közvetlenül csak a csomagok alagutaztatásáért felel a két rendszer között, és a meglévő hálózati segédeszközöket használja a teljes VPN-megoldás kialakítására. A hasonlat kedvéért képzeljük el, hogy az otthoni és a munkahelyi hálózat két különálló vasúti hálózat. Minden számítógépnek egy állomás felel meg. A Linux-rendszermag vezérli a váltókat, így határozza meg, hogyan érik el a vonatok egyik állomásról a másikat. Ezeket a lehetőségeket a `route` programmal befolyásolhatjuk, így a végfelhasználó is adhat hozzá vagy távolíthat el állomásokat. A Linux-rendszermag képes a vonatok útvonalát megváltoztatni. Vasutaspéldánkhoz adjuk hozzá az internetet, a hatalmas vasúthálózatot. Az otthoni és a munkahelyi hálózatok aprócska nyúlványok ebben a rendszerben. Általában egyetlen állomás, a tűzfal vagy az átjáró rendelkezik közvetlen hozzáféréssel az internet vasúthálózatához. Ha egy másik állomás az otthoni vágányokról vonatot akar küldeni az internetre, először az átjáróállomásra kell továbbítania. Ezt az útvonal-módosítást, ami műszakilag az IP-álcázásnak vagy a hálózati címfordításnak felel meg, az `iptables` program vezérli. Az `iptables` a 2.4-es Linux-rendszermagban lévő Netfilter tűzfalkód felhasználói térben futó fele.

Hogyan illeszthető be a VTun a példába? Emlékezzünk, hogy az otthoni és a munkahelyi hálózatok egymástól elszigetelt vasúthálózatok. Otthonról általában nem mehet vonat a munkahelyre, mert a munkahelyi tűzfalállomás nem engedi át. A VTun segítségével virtuális sínpárt építhetünk két elkülönült hálózaton lévő állomás – például egy otthoni és egy munkahelyi gép – között. A sínpár lefektetése után az állomásokon be kell állítani az `iptables`-t és a `routed`-et, hogy az otthonról érkező vonatok szabadon elérhessék a munkahelyi rendszert, mintha csak egy munkahelyi gépről érkeztek volna.

Szabályok és kikötések

Most, hogy áttekintettük a VTun VPN-összetevőit, készen állunk a teljes megvalósítás vizsgálatára. A leghétköznapibb eset, amikor egyetlen távoli munkaállomást (az otthoni gépünket) kötjük össze a munkahelyi belső hálózattal a munkahelyi gépünkön keresztül. Az egyszerűség kedvéért feltesszük, hogy lehetséges SSH-kapcsolatot létesíteni otthonról a munkahelyi géppel, de az a gép egyébként nem érhető el az internetről. Tegyük fel, hogy az otthoni hálózat a 192.168.1.0/24 alhálózatra, a munkahelyi hálózat a 192.168.5.0/24 és a 192.168.100.0/24 alhálózatokra van beállítva.

A VTun kiszolgálóalapú rendszer. A kiszolgálógép a megadott kapukon figyeli a bejövő kapcsolatokat. Az ügyfél kezdeményezi az alagút létrehozását a kiszolgáló kapujára csatlakozva – ebben a példában az otthoni gép az ügyfél és a munkahelyi gép a kiszolgáló.

A VPN létrehozása azt jelenti, hogy a munkahelyi hálózat onnantól fogva csak annyira biztonságos, mint az otthoni hálózat. Emiatt az otthoni gépeket kötelező tűzfallal védeni, amire mindig fel kell rakni az összes biztonsági foltot, és behatolásvédelmi szempontból rendszeresen ellenőrizni kell. A másik nagyon fontos szabály, hogy soha ne hozzunk létre VPN-t a munkahelyi rendszergazdák tudta és beleegyezése nélkül.

Telepítés

A VTun programot az ügyfélre és a kiszolgálóra is telepíteni kell, azaz az itt leírt folyamatot mindkét rendszeren végre kell hajtani. A folyamatot a Red Hat Linux újabb változatain próbálták ki. Ha a telepítés nem sikerül a terjesztéseden, küldj levelet a ryan@ryanbreen.com címre. Ezeket a válaszokat beépítem a ➲ http://www.ryanbreen.com/vtun honlapon fenntartott terjesztésfüggő hibákat felsoroló állományba. Néhány terjesztés eleve tartalmazza a VTun-csomagot, úgyhogy elképzelhető, hogy megtakaríthatunk egy lépést, ha a csomagkezelővel telepítjük a VTun programot.

A legtöbb VPN-megoldáshoz hasonlóan a VTun is igényli a rendszermag támogatását, ebben az esetben a TUN pont–pont hálózati illesztőprogramot. A `TUN` modul része az alaprendszermagnak, úgyhogy valószínűleg nem kell rendszermagot fordítani hozzá. Tegyünk egy próbát az `insmod tun` parancscsal (rendszergazdaként), amely megkísérli az illesztőprogram betöltését. Ha a modul nem található, töltsük le a legújabb változatot (jelenleg ez a tun-1.1) a ➲ http://vtun.sourceforge.net/tun/index.html címről. Telepítsük:

```
tar xzf tun-1.1.tar.gz
cd tun-1.1
su -c ´make install´
```

Ha önműködően szeretnénk betölteni a TUN modult, amikor egy folyamat megkísérli elérni a virtuális alagúteszközt, akkor a következő sort adjuk a */etc/modules.conf* állományhoz:

```
alias char-major-10-200 tun
```

Ezután állítsuk be és telepítsük a felhasználói térben futó vtund programot. A legújabb VTun-csomag a ➲ http://vtun.sourceforge.net/download.html oldalról érhető el. Most forrásból telepítünk, de ha a terjesztésünk támogatja az RPM- vagy DEB-csomagokat, akkor nyugodtan telepítsük az előre lefordított változatot. A legújabb forráscsomag a cikk nyomdába adásának pillanatában a vtun-2.5.tar.gz.
A fordítás a szokásos módon zajlik:

```
tar xzf vtun-2.5.tar.gz
cd vtun-2.5
./configure
make
su -c ´make install´
```

Egyes terjesztéseknél a beállítólépés sikertelen lehet, mert az *LZO* nincs telepítve. Az *LZO* egy tömörítő programkönyvtár, amelyet a VTun használ.
A ➲ http://www.oberhumer.com/opensource/lzo/download weboldalról tölthető le. Fordítsuk le és telepítsük az *LZO*-t, majd próbáljuk újra a VTun telepítését.
A telepítés során a VTun a beállítóállományát a */usr/local/etc/vtund.conf* helyre teszi. Ez nagyon zavaró lehet, mert az ügyfélnek és a kiszolgálónak mást-mást kell beállítani az alagútleíró részben. A félreértések elkerülése érdekében a *vtund.conf* állományt nevezzük át *vtund-client.conf*, illetve *vtund-server.conf* névre. Ezt követően az indításkor kézzel adjuk meg a megfelelő beállítóállomány elérési útját. A következőkben e javaslat szerint járunk el.

A VTun beállítóállománya

A VTun beállítóállománya viszonylag egyszerű (lásd az *1.* és a 2. listát – az utóbbi megtalálható az 51. CD Magazin/ VTun könyvtárában). Az állomány három különálló részből áll. Az elsőben általános beállítások szerepelnek, például a kiszolgáló kapuszáma és a segédprogramok elérési útja. A másodikban az alapértelmezett munkamenet beállításai vannak, amelyek az alagút hálózati tulajdonságait írják le. Ezeket a tulajdonságokat szükség esetén egy adott alagút beállításánál felül lehet bírálni.
Van egy alagútbeállító kapcsoló, amely különleges figyelmet igényel: ez a keepalive. A vállalati rendszergazdák gyakran alacsony tétlenségi időt engedélyeznek a tűzfalon keresztül a működő kapcsolatok számára. Ha az alagút a megadott időnél tovább tétlen, a kapcsolat időtúllépéssel megszakad. A keepalive engedélyezésével a VTun úgy kerüli meg ezt a gondot, hogy rendszeresen csomagokat küld az ügyféltől a kiszolgálóra, így meggyőzi a tűzfalat arról, hogy a kapcsolat él.
A beállítások utolsó csoportja az adott alagút beállításait tartalmazza. A beállítóállomány tetszőleges számú ilyen beállítást tartalmazhat, így több VPN kialakítása is lehetővé válik az ügyfelek és a kiszolgálók számára. Minden alagútbeállító csoport egy névvel kezdődik. Én a my_tunnel nevet választottam, de bármit megadhatunk. Minden alagút jelszóval védhető, de ezzel általában nem foglalkozunk, ha az alagút SSH-n keresztül jön létre. Az up és down részek az alagút felépülésekor, illetve lebontásakor végrehajtandó parancsokat tartalmazzák.
Az 1. és 2. listán látható egyszerű beállítóállományok arra utasítják a VTun programot, hogy kapcsolatfelvételkor hozza létre az alagút csatolóit. A beállítóállományokban a %% minta jelenti az alagútcsatolót, így több alagutat is létrehozhatunk tetszőleges sorrendben. Az alagútcsatoló tényleges neve a „tun” előtagból és egy számjegyből áll. Az első létrehozott alagút neve tun0.

1. lista Egyszerű vtund-client.conf

```
options {
    port 5000;

    # Különféle programok elérési útja
    ifconfig    /sbin/ifconfig;
}

# Munkamenet alapértékei
default {
    compress no;    # Nincs tömörítés
    encrypt no;     # Az SSH úgyis titkosít
    speed 0;        # Legnagyobb sebesség
alapértelmezettként
    keepalive yes;
    stat yes;
}


my_tunnel {
    pass   XXXXXXXX;     # Jelszó
    type   tun;           # IP-alagút
    proto tcp;            # TCP protokoll

up {
    # 10.3.0.1 = hamis alagútcsatoló (otthon)
    # 10.3.0.2 = hamis alagútcsatoló
                ➥(munkahelyen)
    # 192.168.5.0/24 = az 1. munkahelyi
                      ➥hálózat
    # 192.168.100.0/24 = a 2. munkahelyi
                       ➥hálózat
    ifconfig
      "%% 10.3.0.1 pointopoint 10.3.0.2 mtu
       ➥1450";
};
down{
    ifconfig "%% down";
};
}
```

VTun VPN létrehozása

Próbáljuk ki a gyakorlatban a VTun beállításáról tanultakat. Az 1. és 2. lista felhasználásával hozzunk létre egy egyszerű alagutat. A listák az ➲ ftp.ssc.com/pub/lj/listings/issue112/6675.tgz címen megtalálhatók, ha nem szeretnénk begépelni őket. Mentsük a *vtund-server.conf* állományt a munkahelyi gép */usr/local/etc* könyvtárába, illetve a *vtund-client.conf* állományt az otthoni gép */usr/local/etc* könyvtárába. Miután a beállítóállományok a helyükre kerültek, mindkét gépen indítsuk el a VTun-folyamatokat. Rendszergazdaként indítsuk el a kiszolgálót a munkahelyi gépen:

```
vtund -f /usr/local/etc/vtund-server.conf -s
```

3. lista Teljes vtund-client.conf

```
options {
    port 5000;

    # Különféle programok elérési útja
    ifconfig    /sbin/ifconfig;
    firewall    /sbin/iptables;
    route       /sbin/route;
}

# Munkamenet alapértékei
default {
    compress no;    # Nincs tömörítés
    encrypt no;     # Az SSH úgyis titkosít
    speed 0;        # Legnagyobb sebesség
                    # alapértelmezettként
    keepalive yes;
    stat yes;
}

my_tunnel {
    pass  XXXXXXXX;   # Jelszó
    type  tun;        # IP-alagút
    proto tcp;        # TCP protokoll

up {
    # 10.3.0.1 =  hamis alagútcsatoló
                  ➥(otthon)
    # 10.3.0.2 =  hamis alagútcsatoló
                  ➥(munkahelyen)
    # 192.168.5.0/24 = az 1. munkahelyi
                       ➥hálózat
    # 192.168.100.0/24 = a 2. munkahelyi
                         ➥hálózat
    ifconfig
      "%% 10.3.0.1 pointopoint 10.3.0.2 mtu
       ➥1450";
    route "add -net 192.168.5.0 netmask
    ➥255.255.255.0 gw 10.3.0.2";
    route "add -net 192.168.100.0 netmask
    ➥255.255.255.0 gw 10.3.0.2";
};
down{
    ifconfig "%% down";
    route "del -net 192.168.5.0 netmask
    ➥255.255.255.0 gw 10.3.0.2";
    route "del -net 192.168.100.0 netmask
    ➥255.255.255.0 gw 10.3.0.2";
    };
}
```

A `-s` kapcsoló hatására a `vtund` kiszolgálóként indul, a kapcsolatokra az 5000-es kapun vár. A kiszolgáló eléréséhez el kell tudnunk érni az 5000-es kaput a munkahelyi gépen. Emlékezzünk rá, hogy példánkban feltettük, hogy a munkahelyi gép csak SSH-n keresztül érhető el, ezért az SSH kaputovábbító képességét kell használnunk a munkahelyi gép 5000-es kapujához való alagutazásra. Otthonról adjuk ki a következő parancsot:

```
ssh mydesktop.work.com -L 5000:localhost:5000
```

A `-L` kapcsoló hatására az OpenSSH az otthoni gép 5000-es kapuját a munkahelyi gép 5000-es kapujára irányítja. Az otthoni gép 5000-es kapujára irányított kapcsolatok SSH-n keresztül észrevétlenül átalagutaznak a munkahelyi gép 5000-es kapujára. Az elrendezésnek további előnye az, hogy a VPN-es forgalom titkosítva van.
Miután a munkahelyi gépen futó kiszolgáló elérhető az otthoni gépről, csak az marad hátra, hogy elindítsuk az ügyfelet. Rendszergazdaként az otthoni gépen futtassuk a következő parancsot:

```
vtund -f /usr/local/etc/vtund-client.conf
my_tunnel localhost
```

A `my_tunnel` kapcsoló megadja az ügyfélnek és a kiszolgálónak, hogy milyen alagutat kell létrehozni. Mindkét rendszer lekéri és futtatja a megfelelő beállítóállományokból a `my_tunnel` részhez tartozó `up` bejegyzésben megadott parancsokat. Az utolsó kapcsoló, a `localhost`, megadja a VTun kiszolgáló gépnevét. Ebben az esetben a VTun kiszolgáló a `localhost`, mert az 5000-es kaput átirányítottuk az otthoni gépről a munkahelyi gépre.
Ha az alagút sikeresen létrejött, akkor mindkét gépen megjelenik az `ifconfig` kimenetében a `tun0` csatoló. Az otthoni gépnek az IP-címe 10.3.0.1 a `tun0`-n, a munkahelyi gépnek az IP-címe 10.3.0.2. A vonatos példára visszatérve, létrejött egy sínpár az otthoni és a munkahelyi gépek között, és most már irányíthatunk rá vonatokat. Bemutatásként hozzunk létre egy SSH-kapcsolatot az otthoni gépről a 10.3.0.2-re.

Beüzemelés

Van már egy működő alagutunk otthonról a munkahelyre. A következő lépésben a `route` és `iptables` programok segítségével be kell állítanunk, hogy az otthonról jövő csomagok a munkahelyi gépen keresztül álcázódjanak a munkahelyi hálózat felé. Szerencsére ez egyszerűen megtehető, csak egy pár sort kell hozzáadni beállítóállományokhoz az ügyfélen és a kiszolgálón, és újra kell indítani a `vtund` folyamatokat.
A VTun a kapcsolat létrejöttekor végrehajtja a megfelelő `route` és `iptables` parancsokat.
A vonatos példával elmagyarázva ez azt jelenti, hogy az otthoni állomásról minden munkahelyi hálózatra irányuló vonatot az újonnan létrehozott VTun-sínpáron keresztül kell küldeni. Kézzel ez így állítható be:

```
route add -net 192.168.5.0 netmask
➥255.255.255.0 gw 10.3.0.2
route add -net 192.168.100.0 netmask
➥255.255.255.0 gw 10.3.0.
```

A másik megoldás, hogy a *3. listán* látható módon adjuk hozzá a parancsokat a ***vtund-client.conf*** állományhoz. A parancsok hatására az `iptables` minden csomagot továbbít a `tun` csatolóról, és úgy álcázza őket, mintha a munkahelyi gépről indulnának. Megtehetjük azt is, hogy a *4. listán* látható parancsokat adjuk hozzá a ***vtund-server.conf*** állományhoz, és újraindítjuk a kiszolgálót.
A `route` és az `iptables` beállítása után az egész vállalati belső hálózat látszani fog otthonról. Böngészhetjük a belső webkiszolgálókat, kapcsolódhatunk a forráskód-kiszolgálóhoz, és megpróbálhatunk grafikus elemeket (például egy `xterm`-et) exportálni. Ezekre a célokra több mint elegendő a teljesítmény, és az SSH-alagút biztosítja, hogy a forgalom a kíváncsi szemek elől rejtve marad.
Ha a kiszolgálót esetleg önműködően szeretnénk elindítani,

4. lista Teljes vtund-server.conf

```
options {
    port 5000;

    # Különféle programok elérési útja
    ifconfig    /sbin/ifconfig;
    firewall    /sbin/iptables;
    route       /sbin/route;
}

# Munkamenet alapértékei
default {
    compress no;    # Nincs tömörítés
    encrypt no;     # Az SSH úgyis titkosít
    speed 0;        # Legnagyobb sebesség
                    # alapértelmezettként
    keepalive yes;
    stat yes;
}

my_tunnel {
    pass  XXXXXXXX;    # Jelszó
    type  tun;         # IP-alagút
    proto tcp;         # TCP protokoll

up {
    # 10.3.0.1 = hamis alagútcsatoló (otthon)
    # 10.3.0.2 = hamis alagútcsatoló
                 ➥(munkahelyen)
    # 192.168.1.0/24 = az otthoni hálózat
    ifconfig
      "%% 10.3.0.2 pointopoint 10.3.0.1 mtu
       ➥1450";
    route "add -net 192.168.1.0 netmask
    ➥255.255.255.0 gw 10.3.0.1";
    firewall "-t nat-A POSTROUTING -o %%
    ➥-j MASQUERADE";
    firewall "-AFORWARD -i %% -j ACCEPT";
};
down{
    ifconfig "%% down";
    route "del -net 192.168.1.0 netmask
    ➥255.255.255.0 gw 10.3.0.1";
};
}
```

akkor a használt terjesztéstől függ a teendőnk. A VTun-forráscsomag tartalmaz néhány indítóparancsfájlt a különböző terjesztésekhez – olvassuk el a ***Readme*** állományt, majd válasszuk ki a legmegfelelőbbet.

Beállítások haladóknak

Észrevehettük, hogy csak egyetlen otthoni gépnek volt hozzáférése a munkahelyi intranethez. Az otthoni hálózat más állomásairól induló vonatok nincsenek átirányítva. Ez a megoldás egy hajszálnyival biztonságosabb, mert csökkenti a munkahelyi hálózatot fenyegető, az otthoni hálózatról eredő hibákból adódó veszélyeket. Ha az otthoni hálózat más gépeiről is el szeretnénk érni a munkahelyi hálózatot, akkor egyszerűen adjuk hozzá a megfelelő `iptables`-szabályokat a ***vtund-client.conf*** állomány `up` részéhez. Ezt az érdeklődő olvasó gyakorlásképpen elvégezheti.

A fenti megoldás tökéletesen működik, ha bármelyik munkahelyi gép elérhető SSH-n keresztül. Sajnos sok munkahelyen egyáltalán nem hagynak nyitva egy bejövő kaput sem. Pontosan ez volt a helyzet az új munkahelyemen is, de a rugalmassága átsegített ezen az akadályon is. A megoldás a szerepek felcserélése: a munkahelyi gép lesz az ügyfél, és az SSH-alagutat a munkahelyről kezdeményezzük.

A megoldás akkor működőképes, ha az otthoni gépet a munkahelyünkről el tudjuk érni. A legtöbb széles sávú szolgáltató viszont dinamikus IP-címet oszt. A gond a dinamikus IP-kre kifejlesztett DNS-szolgáltatások egyikével áthidalható, például a Dyndns.org által nyújtottal.

A legnagyobb hátrány a viszonylagos törékenység. Biztonságos környezetben az ügyfél nem indul el önműködően, mert az SSH-kapcsolathoz hitelesítés szükséges. Egy munkahelyi áramszünet miatt könnyen kizáródhatunk. Ha kevésbé aggódunk a biztonság miatt, önműködővé tehetjük a bejelentkezést az SSH nyilvános kulcsos hitelesítése segítségével, ha nincs a kulcsnak jelszava, illetve az `expect` program parancsállományba építésével. Egyik módszert sem javaslom.

Ha a munkahelyi gép szünetmentes tápegységre van kötve, akkor ritkán jön elő ez a gond. Az elmúlt hat hónapban, amióta ezt a megoldást használom, csak egyszer tartott olyan hosszú ideig az áramszünet, hogy a VPN-em ügyféloldala meghaljon tőle. A megoldás az otthoni hálózat oldalán is megbízható. A számítógép napokra is hálózati kapcsolat nélkül maradhat, a VPN újraindul, ahogy elindítjuk a `vtun` kiszolgálót, hála az ügyfél okos életbentartó és újrapróbálkozó képességeinek.

Összefoglalás

A VTun képességeinek teljes ismertetése jóval meghaladná e cikk kereteit. Az itt bemutatott egyszerű megoldásokon kívül a VTun lehetővé teszi IP-n kívüli protokollok alagutaztatását is etherneten, PPP-n vagy SLIP-en keresztül. A VTun saját maga is képes titkosításra, tömörítésre és sávszélesség-formázásra, úgyhogy minden elképzelhető összeköttetéshez alkalmazható.

Köszönetnyilvánítás

Nagyon köszönöm ***Jennifer Edwardsnak*** és ***James Manningnak***, hogy lektorálták a cikket.

A cikkhez kapcsolódó listák megtalálhatóak az 51. CD Magazin/VTun könyvtárában.

Linux Journal 2003. augusztus, 112. szám

Ryan Breen (ryan@ryanbreen.com)
A Duke Egyetemen szerzett számítástechnikai és közgazdasági diplomát. Bostonban él a barátnőjével és a kutyájával. Nagy áteresztőképességű böngészőszimulációkat épít, elkötelezett KDE-felhasználó, és néha fejleszt is KDE alá.

KAPCSOLÓDÓ CÍMEK

VTun ➲ http://vtun.sourceforge.net
Ryan VTun lapja ➲ http://www.ryanbreen.com/vtun
VTun TUN/TAP meghajtó ➲ http://vttun.sourceforge.net/tun