WLAN-ok védelme WPA és FreeRADIUS alkalmazásával – III. rész

Új, a korábbiaknál biztonságosabb vezeték nélküli hálózatunk üzembe helyezésének végső lépéseként fel kell készítenünk néhány nem linuxos ügyfelet az új szabvány kezelésére.

Előző két írásomban áttekintettem, hogy a ***WPA (Wi-Fi protected access, Wi-Fi védett elérés)*** segítségével hogyan védhetjük meg a ***vezeték nélküli (wireless)*** helyi hálózatokat, röviden ***WLAN***-okat a jogosulatlan hozzáférésektől és a lehallgatásoktól. Elkezdtem ismertetni, hogy a ***FreeRADIUS*** segítségével hogyan valósíthatjuk meg a ***WPA***-t saját ***WLAN***-unkon. Az eddigiek során a ***FreeRADIUS*** telepítéséről, a ***hitelesítő szervezet (certificate authority, CA)*** létrehozásáról, valamint a ***WPA***-val történő használatra szánt digitális tanúsítványok előállításáról és aláírásáról volt szó. Ebben a hónapban megnézzük, hogy hova kell elhelyezni ezeket a tanúsítványokat, hogyan kell beállítani a ***FreeRADIUS***-t, a vezeték nélküli hozzáférési pontot és az ügyfeleket. Mindezek alapján, úgy vélem, bárki nekiláthat saját ***WLAN***-ja biztonságának megerősítéséhez.

Rövid ismétlés

Azok számára, akik csak most kapcsolódnak be a cikksorozatba, esetleg fel kell frissíteniük az emlékezetüket, hogy pontosan mit is próbálunk elérni, röviden tekintsük át céljainkat és lehetőségeinket. A ***WPA*** erőteljes hitelesítési eljárásokkal bővíti a régebbi, kriptográfiailag megtört ***WEP*** protokollt; teszi ezt a ***802.1x*** protokoll és alprotokolljai révén, mint az ***EAP***, a ***PEAP*** és az ***EAP-TLS***. A ***WPA*** a ***TKIP*** protokoll révén képes a munkamenetkulcsok dinamikus egyeztetésére és a kulcsok önműködő megújítására is. Ha vezeték nélküli ügyfelünk támogatja a ***WPA***-t – vagyis rendelkezik ***WPA*** kérvényezővel –, továbbá a vezeték nélküli hozzáférési pontunk is rendelkezik ilyen képességgel, akkor kétharmad részben már sikerrel jártunk. Ha viszont teljes mértékben ki akarjuk használni a ***802.1x*** által kínált lehetőségeket, akkor szükségünk lesz egy ***RADIUS*** háttérkiszolgálóra is – itt lép be a képbe a ***FreeRADIUS***.
A múlt alkalommal kialakított példakörnyezetben egy ***FreeRADIUS*** kiszolgálót helyezünk üzembe, amelynek feladatául a tetszőleges ***WPA***-képes vezeték nélküli hozzáférési ponthoz csatlakozó ***Windows XP*** ügyfelek hitelesítését tűztük ki. ***802.1x*** eljárásunk az ***EAP-TLS*** lett. Az ***EAP-TLS***, mint talán néhányan még emlékeznek rá, a ***TLS*** protokollt használja a vezeték nélküli kérvényezők (ügyfelek) és a hozzáférési pontok kölcsönös hitelesítésére; illetve mindehhez ***X.509*** digitális tanúsítványokat alkalmaz.
Még hátralévő feladataink a következők:

- A múlt alkalommal létrehozott kiszolgáló és ***CA*** tanúsítványok telepítése a ***FreeRADIUS*** kiszolgálóra
- A ***FreeRADIUS*** beállítása ezeknek a tanúsítványoknak a használatára, ***EAP-TLS*** felett, a hozzáférési pont felhasználóinak hitelesítése céljából
- A hozzáférési pont beállítása a hitelesítés átirányítására a ***FreeRADIUS*** kiszolgálóra
- Az ügyfélprogram és a múlt alkalommal létrehozott ***CA*** tanúsítványok telepítése egy ***Windows XP*** alapú ügyfélre, illetve beállítása úgy, hogy a ***WLAN***-hoz történő csatlakozáskor ***WPA***-t használjon.

A FreeRADIUS kiszolgáló előkészítése

A ***WPA***-ról szóló sorozat második részében létrehoztunk három ***X.509*** digitális tanúsítványt: a ***hitelesítő szervezetét*** (***cacert.pem***), egy ***kiszolgálótanúsítványt*** (***kiszolgalo_kulcstanusitvany.pem***) és egy ***ügyféltanúsítványt*** (***ugyfel_tanusitvany.p12***). A kiszolgáló és az ügyfél fájlja a tanúsítványt és annak titkos kulcsát egyaránt tartalmazza, ezért mindkettő telepítését kellő körültekintéssel kell elvégezni. A ***CA*** tanúsítványt ugyanakkor a kulcstól elkülönítve tároljuk, vagyis a ***cacert.pem***-et szabadon terjeszthetjük.
A ***FreeRADIUS*** beállító fájljai a ***/etc/raddb/*** vagy a ***/usr/local/etc/raddb/*** könyvtárban találhatók, terjesztéstől függően. A könyvtárnak van egy ***certs/*** nevű alkönyvtára, ide kell bemásolnunk a ***CA*** tanúsítványát, valamint a kiszolgálótanúsítványt és kulcsát. Ellenőrizzük, hogy a cacert.pem tulajdonosa a root felhasználó-e, a fájlra vonatkozó engedélyek pedig a következők legyenek:

```
-r--r--r--
```

1. kódrészlet A raddb/certs könyvtárban található tanúsítványokra megadott tulajdonosok és engedélyek

```
-r--r--r--  1 root   users 1294 2005-02-10 01:05
➥cacert.pem
-r--------  1 nobody users 1894 2005-02-10 01:00
➥kiszolgalo_kulcstanusitvany.pem
```

2. kódrészlet A radiusd.conf két módosítandó beállítása

```
user = nobody
group = nobody
```

A ***kiszolgalo_kulcstanusitvany.pem*** fájlnak ugyanakkor a ***nobody*** felhasználó tulajdonába kell tartoznia, `-r--------` engedéllyel. Az *1. kódrészlet* ennek a két fájlnak a hosszú könyvtárlistáját szemlélteti.
Ha már a fájlok tulajdonosaival foglalkozunk, ellenőrizzük, hogy a */var/log/radius/radius.log* fájlra és a */var/run/radiusd/* könyvtárra van-e írási engedélye a ***nobody*** felhasználónak. Ha a ***FreeRADIUS***-t forrásból fordítottuk, akkor lehetséges, hogy az előbbiek helyett a */usr/local/var/log/radius/radius.log* fájllal és a */usr/local/var/srun/radiusd/* könyvtárral kell dolgoznunk. A ***radius.log*** és a ***radiusd/*** tulajdonosa egyaránt a ***nobody*** legyen.
Mielőtt beleásnánk magunkat a ***FreeRADIUS*** beállító fájljaiba, létre kell hoznunk további két, a ***FreeRADIUS*** által a ***TLS*** használatához igényelt fájlt. Az első a ***Diffie-Hellman (dh)*** átadott értékeket tartalmazza, ezekre a ***TLS*** munkamenet-kulcsok egyeztetésekor van szükség. A ***dh*** fájlt úgy hozhatjuk létre, hogy átváltunk a ***FreeRADIUS raddb/certs/*** könyvtárába, majd kiadjuk a következő parancsot:

```
# openssl dhparam -check -text -5 512 -out dh
```

A második fájl egy véletlenszerű bitfolyamot tartalmazó, szintén a ***TLS*** műveleteknél szükséges adatfájl, ***random*** névvel. Itt hívnám fel rá a figyelmet, hogy a véletlenszerűnek szánt tartalmak létrehozását ***nem szabad*** egyszerűen az aktuális időbélyeg vagy valamilyen hasonló, a legkevésbé sem véletlenszerű karakterlánc beírásával letudni; még akkor sem, ha a ***WPA***-val kapcsolatos, az interneten fellelhető leírások némelyikében ezt javasolják. Ehelyett a rendszermag kiváló minőségű véletlenszám-előállítóját kell használni. A ***raddb/certs*** könyvtárból tehát a következő parancsot adjuk ki:

```
# dd if=/dev/urandom of=random count=2
```

Mindkét fájlra olvasási jogot kell adni a ***nobody*** felhasználónak, az írási jogot pedig mindenkitől meg kell vonni.

A FreeRADIUS beállítása

Végre készen állunk a ***FreeRADIUS*** beállításainak megadására. Elég ijesztő lehet a */raddb* könyvtárban sorakozó fájlok sokasága, de riadalomra semmi ok. Az ***EAP-TLS*** alapú ***WPA*** használatához csak a következő három fájlt kell módosítanunk: ***radiusd.conf***, ***eap.conf*** és ***clients.conf***.
A ***radiusd.conf*** fájlban csupán azt kell megadnunk, hogy a ***radiusd*** folyamat melyik felhasználó és csoport jogaival fusson. A jogok alapesetben a démont indító felhasználótól öröklődnek. Ha a ***radiusd***-t parancsfájlból indítjuk, akkor a root jogait örökli – nyilván ezt el szeretnénk kerül. Felhasználóként és csoportként tehát a ***radiusd.conf***-ban egyaránt ***nobody***-t adjunk meg, ahogy ezt a *2. kódrészlet* is szemlélteti. Természetesen a ***nobody/nobody*** helyett más, kiemelt jogokkal nem rendelkező felhasználót és csoportot is választhatunk, ám ha így teszünk, akkor a korábban említett tanúsítványfájlok tulajdonosait és a vonatkozó jogokat is módosítanunk kell. Bárhogy is döntünk, ellenőrizzük, hogy a kiválasztott felhasználóhoz a */etc/password* fájlban tartozó bejegyzés szerint a felhasználó nem kaphat héjhozzáférést (a bejegyzés például */bin/false* vagy */bin/true* lehet); a fiókot ***SSH***, ***telnet*** és hasonló programok nem használhatják. Mondanom sem kell, azt sem árt ellenőrizni, hogy a felhasználó és a csoport egyáltalán létezik-e, és ha nem, létre kell hozni őket.
A ***radiusd.conf*** további beállításokat is tartalmaz, ám csak a fenti kettő módosítása az, ami igazán lényeges. További tudnivalókat a ***radiusd.conf(5) man*** oldalon vagy ***Jonathan Hassell RADIUS*** című könyvében találni.
A következő átírandó fájl az ***eap.conf*** – itt merülünk a sűrűjébe. A *3. kódrészlet* az ***eap.conf*** módosítandó sorait tartalmazza. A 3. kódrészletben a `private_key_password` átadott értékkel megadtam egy kiszolgáló-kulcs jelszót. Ez valójában üres, feltéve, hogy a kiszolgáló tanúsítványát és kulcsát az ***OpenSSL*** `-nodes` kapcsolójával hoztuk létre. Sajnos a múlt hónapban magam is ezt javasoltam, ám most, talán még időben, szeretném ezt visszavonni. A jelszó nélküli ***X.509*** kulcsok használata ront a biztonságon, még ha a kulcsot nyílt szöveges beállító fájlban tároljuk is, mint például az ***eap.conf***. Bizony, ha a ***FreeRADIUS*** kiszolgálón egy behatoló root jogokhoz jut, akkor – köszönhetően az ***eap.conf*** fájlnak – még egy jelszóval védett tanúsítvány bizalmassága is sérülhet. Ha viszont a tanúsítványt/kulcsot út közben – például, amikor a ***CA*** állomásról a ***FreeRADIUS*** kiszolgálóra másoljuk – hallgatják le, akkor, ha jelszóval védett, a támadó nem tud mit kezdeni vele.
Bármelyik megoldást is válasszuk, ellenőrizzük, hogy az ***eap.conf*** a root tulajdonában van-e, illetve csak ő rendelkezik-e hozzá írási joggal, és nem a ***radiusd.conf*** fájlban megadott felhasználó. Furcsa, igaz? A ***nobody***-nak nem kellene jogot adni a beállító fájlok olvasására? A válasz nem, hiszen ha a ***radiusd***-t rootként indítjuk, akkor először beolvassa a beállító fájlokat (***radiusd.conf***, ***eap.conf*** és ***clients.conf***), csak ezután vált át a ***nobody*** jogosultságaira.
Végül létre kell hoznunk egy bejegyzést a hozzáférési pont számára a clients.conf fájlban. A *4. kódrészlet* erre mutat példát.
A 4. kódrészletben a `client` (ügyfél) utasítás adja meg a hozzáférési pont IP-címét. A hozzá tartozó `secret` (titok) átadott érték adja meg azt a karakterláncot, amelyet a hozzáférési pont a ***FreeRADIUS*** kiszolgálónak küldött kérésekben titkosítási kulcsként használ. A `shortname` (rövid név) egyszerűen csak egy álnév a hozzáférési ponthoz, például a naplóbejegyzésekben találkozhatunk vele.

3. kódrészlet Az eap.conf fájl módosításai

```
eap {
  # Itt számos általános EAP átadott érték
  # megadására van lehetőség,
  # ám számunkra most csak
  # a default_eap_type fontos:
  default_eap_type = tls
  # Ezután következnek az egyes
  # EAP-típusok beállításai.
  # Mivel EAP-TLS-t akarunk használni,
  # csak a tls{} szakasszal
  # kell foglalkoznunk:
  tls {
    # Az alábbi értékek azt adják meg
    # a radiusd-nek, hogy hol
    # találja a tanúsítványokat és a kulcsokat,
    # illetve a dh és a random fájlokat:
    private_key_password =
    ➥ide_jon_a_kulcs_jelszava
    private_key_file =
    ➥${raddbdir}/certs/bt_keycert.pem
    certificate_file =
    ➥${raddbdir}/certs/bt_keycert.pem
    CA_file = ${raddbdir}/certs/cacert.pem
    dh_file = ${raddbdir}/certs/dh
    random_file = ${raddbdir}/certs/random
    }
  }
```

4. kódrészlet Hozzáférési ponthoz tartozó bejegyzés a clients.conf fájlban

```
client 10.1.2.3/32 {
      secret          = felhasznalojelszo
      shortname       = hozzaferesi_pont
}
```

A ***radiusd*** készen áll az indításra, amit az ***rc.radiusd*** parancsfájllal tehetünk meg:

```
rc.radiusd start
```

Az újraindítás az `rc.radiusd restart` paranccsal történik. Ha a ***radiusd*** hiba nélkül elindult, továbbléphetünk.

A hozzáférési pont beállítása

A következő lépés az egész folyamat legkönnyebb része: a vezeték nélküli hozzáférési pont beállítása ***WPA*** használatára és a ***FreeRADIUS*** kiszolgáló címének megadása. Mindehhez csupán kétféle adatra van szükség, a ***FreeRADIUS*** kiszolgáló ***clients.conf*** fájljában megadott ***RADIUS titokra*** (***secret*** átadott érték), valamint a kiszolgáló ***IP***-címére. Az, hogy ezeket az adatokat ténylegesen hogyan kell megadni a hozzáférési pontnak, az alkalmazott eszköztől és a rajta futó szoftvertől függ. Az én hozzáférési pontom egy ***WLAN***-hozzáférés biztosítására is képes ***Actiontec DSL*** forgalomirányító. Ennek webes felületén a ***Setup›Advanced Setup›Wireless Settings (Beállítások›Speciális beállítások›Vezeték nélküli beállítások)*** pontra kattintottam, majd a ***Security (Biztonság)*** beállításnál a ***WPA***-t választottam. Ezután előre megosztott kulcs helyett átállítottam ***802.1x*** használatára. Kellett adnom neki egy kiszolgálócímet, ez 10.1.2.3 lett, továbbá be kellett írnom a ***FreeRADIUS*** kiszolgáló ***IP***-címét, valamint a 4. kódrészletben már látott titkot (felhasznalojelszo). A kapuszámot az alapértelmezett 1812-n hagytam.
Ha már szóba került a téma: ha a hozzáférési pontot és a ***RADIUS*** kiszolgálót tűzfal választja el egymástól, akkor lehetővé kell tennünk, hogy a hozzáférési pont elérhesse a ***RADIUS*** kiszolgáló 1812-es és 1813-as kapuját. Ekkor egyben a ***RADIUS*** kiszolgáló is módot kap válaszainak ezeken a kapukon keresztül történő továbbítására.

A Windows XP alapú ügyfelek beállítása

Végre elérkeztünk oda, hogy a ***Windows XP*** alapú, vezeték nélküli ügyfeleket beállíthassuk a ***WPA*** használatára képessé tett hozzáférési ponthoz való csatlakozásra. Tudom, hogy linuxos magazinba írok, ezért nem akarok túlságosan sokat rágódni a témán, akit részletesebben is érdekel, az olvassa el ***Ken Roser HOGYAN***-jának 4.3-as szakaszát (lásd a forrásokat). Teendőink röviden:

1. ***Start›Futtatás (Start›Run)***, majd az `mmc` parancs futtatása.

2. A ***Microsoft Management Console***-ban válasszuk a ***Fájl›Beépülő modul hozzáadása/eltávolítása (File›Add/Remove Snap-in)*** parancsot, válasszuk ki a ***Tanúsítványok (Certificates)*** beépülő modult, majd válasszuk a saját fiókunkhoz tartozó tanúsítványok helyi gépre vonatkozó kezelését.

3. Másoljuk át ***CA*** tanúsítványunkat (***cacert.pem***) a windowsos rendszer merevlemezére, például ***C:\cacert.pem*** névvel.

4. Az ***MMC***-ben bontsuk ki a ***Kezelőpultgyökér (Console Root)*** és a ***Tanúsítványok - Aktuális felhasználó (Certificates - Current User)*** csomópontot, majd kattintsunk az egér jobb gombjával a ***Megbízható legfelső szintű hitelesítésszolgáltatók (Trusted Root Certification Authorities)*** elemre. A helyi menüből válasszuk az ***Összes feladat›Importálás (All Tasks›Import)*** parancsot. A varázslóban válasszuk ki a ***C:\cacert.pem*** fájlt, majd mentsük el a megbízható legfelső szintű hitelesítésszolgáltató alá.

5. Másoljuk át az ügyfél tanúsítvány/kulcs fájlját a windowsos rendszerre, például ***C:\ugyfel_tanusitvany.p12*** névvel.

6. A kezelőpulton kattintsunk az egér jobb gombjával a ***Tanúsítványok (Certificates)*** csomópont ***Személyes (Personal)*** ágára. A helyi menüből válasszuk az ***Összes feladat›Importálás (All Tasks›Import)*** parancsot. A megjelenő varázslóban importáljuk be a ***C:\ugyfel_tanusitvany.p12*** fájlt.

7. Az importáló varázsló bekéri tőlünk a tanúsítvány jelszavát, illetve ugyanezen a párbeszédpanelen felkínálja a titkos kulcs erőteljes védelmét is. Sajnos ennek az engedélyezése megakadályozza a *WPA* működését, vagyis a használatától el kell tekintenünk. A kulcs exportálását engedélyező négyzetet is hagyjuk érintetlenül, jobban járunk ugyanis, ha a jelszóval védett fájlról készítünk biztonsági mentést, mintha az importált, jelszóval nem védett változat exportálását engedélyeznénk.

8. A következő képernyőn hagyjuk, hogy a varázsló magától kiválassza a tanúsítványtárolót.

Ezzel a *Windows XP* alapú rendszer készen áll, már csak egy vezeték nélküli hálózati profilt kell létrehoznunk. Ennek módja a vezeték nélküli kártya illesztőprogramjaitól és attól függően változik, hogy a *Windows XP* melyik szervizcsomagját telepítettük. Nálam *Windows XP SP1* fut, *Centrino* lapkakészleten, és az *XP* saját *WPA* kérvényezőjével hoztam létre vezeték nélküli hálózati profilt, megadva saját *WLAN*-om *SSID*-jét. *Hálózati hitelesítésre (Network Authentication) WPA*-t, *adattitkosításra (Data encryption) TKIP*-t választottam, az *EAP típusa (EAP type)* pedig *intelligens kártya vagy más tanúsítvány (Smart Card or other Certificate)* lett. A *Windows* magától felismerte, hogy milyen ügyféltanúsítványt akarok használni – ez annak köszönhető, hogy a múlt alkalommal külön lépésekkel gondoskodtunk arról, hogy az ügyféltanúsítványunk tartalmazza a *Windows XP* kiterjesztett jellemzőit.

Miután megtörtént a vezeték nélkül hálózati profil összeállítása, a *Windowsnak* önműködően kapcsolódnia kell a hozzáférési ponthoz, és végre kell hajtania a *WPA*-kapcsolat egyeztetését. Amennyiben ez sikerrel jár, a *Hálózati kapcsolatok (Network Connections)* között olyan jelzésnek kell megjelennie, mely szerint a vezeték nélküli hálózati kapcsolat hitelesítése sikeresen megtörtént.

Összefoglalás

Messzire jutottunk, remélem, mindenki követni tudott, és a továbbiakban senki számára nem okoz gondot a *WPA* használata. Bár a *WPA* távolról sem tökéletes – valójában a jelszóval védett ügyféltanúsítványoknak a jelszavak nyílt szövegben való tárolása nélküli kezelésére is képes *WPA* kérvényezőkre volna szükség –, azért elmondhatjuk, hogy a vezeték nélküli hálózatok végre elindultak a biztonság irányába.

Linux Journal 2005. június, 134. szám

A cikkhez tartozó források elérhetősége:
➲ *www.linuxjournal.com/article/8200*

Mick Bauer (mick@visi.com)
Biztonsági szakember, a Linux Journal biztonsági témákkal foglalkozó szerkesztője, biztonsági tanácsadó a Minnesota állambeli Minneapolisban található Upstream Solutions LLC Inc.-nél.