Hálózatok építése és üzemeltetése

EAP – RADIUS : Gyakorlati útmutató

EAP-RADIUS gyakorlat

- A gyakorlat alatt egy, az IEEE 802.1X szerint működő hozzáférés vezérlést fogunk megvalósítani. A hozzáférés 3 szereplője:
  - Supplicant: Ő szeretne csatlakozni
  - Authenticator: Ő vezérli a hozzáférési portot és kéri a hitelesítést
  - Authenticaton Server (AS): Ő hitelesíti a felhasználót

- Enterprise WiFi környezetben szinte mindig ezzel találkozunk, de vezetékes környezetben is használatos és hasznos.

Hozzávalók

- A 3 szereplőhöz 3 külön gépet rendelünk
  - Lehet a BME Cloudon 3 gépet indítani
  - Lehetséges, hogy 3 hallgató együtt dolgozik

- A hálózati forgalom megfigyeléséhez szükséges lesz egy WireShark program a lokális számítógépen. Ezt telepíteni kell. Viszonylag kevés mellékhatással jár a hordozható verzió (portable) telepítése.

BME Cloud sablon

- A gyakorlathoz szinte tetszőleges, ami Ubuntu vagy Debian sablon használható. De készült egy sablon kifejezetten a méréshez is EAP-RADIUS néven
- A BME Cloud virtuális gépén nem szükséges grafikus felületet futtatni, a parancssoros elérés elegendő
- A BME Cloud géphez ssh elérés kell. Ajánlott a putty program használata.
- A BME Cloud Ubuntu gépein alapértelmezésben tűzfal fut, amelyet konfigurálni kell. Most az egyszerűség kedvéért akár le is tilthatjuk a tűzfalat, tudva, hogy nem ez a helyes hozzáállás

```
$ sudo ufw disable
```

1. feladat

RADIUS szerver életre keltése

Teszt kapcsolat a RADIUS szerverrel

Jelszavas hitelesítés

RADIUS Authentication Server

- A RADIUS szerver végzi a felhasználók hitelesítést, ő lesz az AS szereplő
- A freeradius programot használjuk, amely nagy tudású, nyílt forrású és ingyenesen elérhető.
- Amennyiben nincs telepítve a freeradius, úgy azt telepíteni kell. A telepítéshez a következő parancsot használjuk:

```
$ sudo apt-get update
$ sudo apt-get install freeradius
```

Freeradius konfigurálása

- A RADIUS szervert sokféleképpen lehet konfigurálni, most csak egy egyszerű konfigurációt hajtunk végre, hogy működjön a hitelesítés. A valóságban ennél bonyolultabbat érdemes használni, mellőze a nyílt jelszavakat.
- A RADIUS szerverhez fel kell venni egy új felhasználót
  - Ehhez a /etc/freeradius/3.0/users fájlt kell módosítani
  - A fájlba a következő sort kell beírni:

```
test    Cleartext-Password := "password"
```

  - Hatására egy *test* nevű felhasználó jön létre *password* jelszóval.

Freeradius konfigurálása (folyt.)

- A RADIUS szerverhez engedélyezni kell az Authneticator kapcsolódását, a kapcsolat jelszóval van hitelesítve és védve
  - Ehhez a /etc/freeradius/3.0/clients.conf fájlt kell módosítani
  - A fájlba a következő sort kell beírni, a megfelelő IP cím helyettesítésével:

```
client 10.9.0.36/32 {
        secret = testing123
        shortname = test
}
```

  - Hatására a megadott IP címről a megadott jelszóval lehet csatlakozni.

Freeradius tesztelése

- A felhasználó és Authenticator helyes beállítását a *radtest* paranccsal tudjuk tesztelni.
- Először érdemes lokálisan testelni, azaz a RADIUS szerver gépen.
  - Állítsuk le a freeradius daemon-t és indítsuk el úgy, hogy jobban lássuk mi történik, azaz maradjon előtérben és írjon ki sokmindent. Ezt hagyjuk egy külön terminálban.

```
$ sudo /etc/init.d/freeradius stop
$ sudo freeradius -X
```

Freeradius tesztelése (folyt.)

- Indítsuk el a radtest programot arról a gépről, amelyen a RADIUS is fut.
  - Paraméternek a felhasználónév, jelszó, RADIUS szerver cím, NAS port és a RADIUS kapcsolat jelszó kerül
  - Amennyiben itt hibát kapunk, akkor valószínűleg a felhasználó nincs beállítva rendesen
- Majd indítsuk el a radtest programot egy távoli gépről és használjuk a RADIUS szerver IP címét
  - Amennyiben itt hibát kapunk, de a lokális teszt működött, akkor a RADIUS szerveren nincs beállítva a kliens rendesen
  - Előfordulhat, hogy a kapcsolat nem hozható létre. Ellenőrizzük ping-gel a másik gép elérését, illetve ellenőrizzük a tűzfal beállításokat

Freeradius tesztelése (folyt)

Lokális kliens

```
$ radtest test password localhost 1
testing123
Sent Access-Request Id 11 from 0.0.0.0:36785
to 127.0.0.1:1812 length 74
        User-Name = "test"
        User-Password = "password"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 1
        Message-Authenticator = 0x00
        Cleartext-Password = "password"
Received Access-Accept Id 11 from
127.0.0.1:1812 to 0.0.0.0:0 length 20
```

RADIUS szerver

```
Ready to process requests
radyt   (19) Received Access-Request Id 11
from 127.0.0.1:36785 to 127.0.0.1:1812
length 74
(19)    User-Name = "test"
(19)    User-Password = "password"
(19)    NAS-IP-Address = 127.0.1.1
(19)    NAS-Port = 1
(19)Message-Authenticator =
   0xbd4c15518765b6652081460f49922880
…
(19) Sent Access-Accept Id 11 from
127.0.0.1:1812 to 127.0.0.1:36785 length 0
(19) Finished request
Waking up in 4.9 seconds.
(19) Cleaning up request packet ID 11 with
timestamp +6724
```

Freeradius tesztelése (folyt)

Távoli kliens

```
$ radtest test password 10.9.0.36 1
testing123
Sent Access-Request Id 52 from 0.0.0.0:41859
to 10.9.0.36:1812 length 74
        User-Name = "test"
        User-Password = "password"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 1
        Message-Authenticator = 0x00
        Cleartext-Password = "password"
Received Access-Accept Id 52 from
10.9.0.36:1812 to 0.0.0.0:0 length 20
```

RADIUS szerver

```
Ready to process requests
(1) Received Access-Request Id 52 from
10.9.0.172:41859 to 10.9.0.36:1812 length 74
(1)   User-Name = "test"
(1)   User-Password = "password"
(1)   NAS-IP-Address = 127.0.1.1
(1)   NAS-Port = 1
(1)Message-Authenticator =
   0xfc55fa47b24c8755ebf11e2b3b0c68f1
…
(19) Sent Access-Accept Id 11 from
127.0.0.1:1812 to 127.0.0.1:36785 length 0
(19) Finished request
Waking up in 4.9 seconds.
(19) Cleaning up request packet ID 11 with
timestamp +6724
```

RADIUS forgalom megfigyelése

- A hitelesítés során a távoli RADIUS kliens és a RADIUS szerver között RADIUS forgalom megy az 1812 UDP porton. Ezt a forgalmat megfigyelhetjük úgy, hogy az IP csomagokat meglessük és eltároljuk.
- A forgalom megfigyelésére a *tcpdump* parancsot használjuk. Amennyiben nem létezik az adott gépen, akkor telepíteni kell (apt-get install).

RADIUS forgalom megfigyelése (folyt)

- A megfigyelésnél a RADIUS szerveren egy külön terminált indítunk, onnan futtatjuk a *tcpdump* parancsot
- Paraméterként az interfészt és a fájl nevét adjuk meg, valamint a szűrési feltételt
- A parancsot a mérés végén a *CTRL+C* billentyűzetkombinációval állítjuk meg

```
$ sudo tcpdump -i eth0 -w file.pcap port 1812
```

RADIUS forgalom megfigyelése (folyt)

- Hajtsuk végre az előző távoli mérést úgy, hogy közben megfigyeljük a RADIUS szerver forgalmát!
- A lementett forgalom a file.pcap fájlban lesz a megfigyelő gépen. Ezt a fájlt kell eljuttatni a saját gépünkre, ahol a WireShark majd elemezni tudja.
  - A fájlt át tudjuk másolni az linuxos *scp*, illetve a putty mellett érkező *pscp* paranccsal. De egyszerű lehet az is, ha egy webszervert indítunk és egy böngészővel töltjük le a fájlt. Ez utóbbi megoldáshoz a BME Cloud kezelőfelületén meg kell nyitni a 80-as portot a web forgalom számára. A megfigyelő gépen a következő paranccsal indíthatjuk a web szervert (Ha nincs busybox, telepíteni kell.)

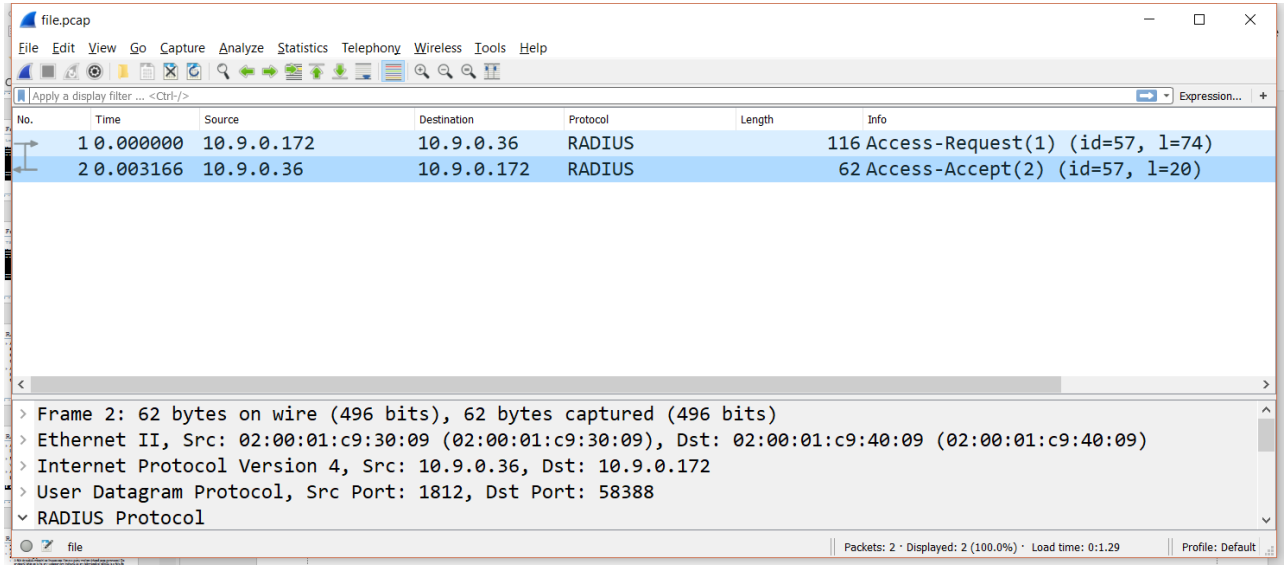
```
$ sudo busybox httpd
```

  - A böngészőből a következő URL használható, a megfelelő port számot beírva

```
http://http://vm.smallville.cloud.bme.hu:11553/file.pcap
```

WireShark

- A WireShark segítségével be kell tölteni a lementett csomagokat, amely ezután kijelzi azokat.



2. Feladat

CHAP hitelesítés

PAP, CHAP, MSCHAP, EAP-MD5

- A *radtest* parancs segítségével az alapértelmezett PAP hitelesítés mellett más hitelesítéseket is tesztelhetünk.
- A CHAP teszteléséhez a következő paranccsal történik:

```
$ radtest -t chap test password 10.9.0.36 1 testing123
```

- Vizsgáljuk meg a különböző hitelesítések üzeneteit a WireShark segítségével!
  - Ehhez a korábbi lépések ismétlése szükséges

CHAP tesztelése

Távoli kliens

```
$ radtest -t chap test password 10.9.0.36 1
testing123
Sent Access-Request Id 61 from 0.0.0.0:55837
to 10.9.0.36:1812 length 75
        User-Name = "test"
        CHAP-Password =
0x979674421828f27f8875846a16d4aca5d8
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 1
        Message-Authenticator = 0x00
        Cleartext-Password = "password"
Received Access-Accept Id 61 from
10.9.0.36:1812 to 0.0.0.0:0 length 20
```

RADIUS szerver

```
Ready to process requests
(2) Received Access-Request Id 61 from
10.9.0.172:55837 to 10.9.0.36:1812 length 75
(2)   User-Name = "test"
(2)   CHAP-Password =
0x979674421828f27f8875846a16d4aca5d8
(2)   NAS-IP-Address = 127.0.1.1
(2)   NAS-Port = 1
(2)Message-Authenticator =
   0x5d1db7f15d84f064d2911765d937f98e
…
(2) Sent Access-Accept Id 61 from
10.9.0.36:1812 to 10.9.0.172:55837 length 0
(2) Finished request
Waking up in 4.9 seconds.
(2) Cleaning up request packet ID 61 with
timestamp +16103
```

3. Feladat

802.1X környezet összerakása

Supplicant – Authenticator – Authentication server

EAP-MD5 hitelesítés

IEEE 802.1X

- A folyamat a Supplicant, az Authenticator és az Authentication Server között zajlik. Így most mint a három szerepre szükség lesz.
- A Supplicant –et és az Authenticator –t egy lokális link köti össze.
- Az Authenticator szerepét a *hostapd* program látja el. Az erre szánt virtuális gépen ezt telepíteni kell. (hostapd csomag)
  - A megoldás nem tökéletes, de a hitelesítés szempontjából jól működik
- A Supplicant szerepét a wpa_supplicant program látja el. Az erre szánt gépen ezt telepíteni kell (wpasupplicant csomag)
- Mindkét program elsősorban a vezetéknélküli környezethez lett kialakítva, de ismerik a vezetékes hálózatokat is

Lokális link

- A BME Cloud gépei esetében most nincsen lokális hálózat beállítva, így a lokális linket IP alapon kell felépíteni a két távoli gép között. (Persze valójában egy hoston futnak)
- A link kialakításához következő parancsok szükségesek, a megfelelő címek átírásával:

```
$ sudo ip link add tunnel0 type gretap remote <Távoli cím>
local <Lokális cím>
$ sudo ip addr add 10.0.0.1 dev tunnel0
$ sudo ip link set tunnel0 up
$ sudo route add -net 10.0.0.0/24 dev tunnel0
```

Lokális link (folyt.)

- A linket mindkét gépen be kell állítani. A példában így a 10.0.0.1 címet 10.0.0.2 címre kell cserélni, illetve természetesen megcserélődnek a local és remote címek is.
- A sikeres beállítást érdemes a *ping* paranccsal ellenőrizni.
  - Amennyiben nem válaszol a távoli gép, érdemes ellenőrizni a tűzfalat, illetve az IP cím beállításokat.

Hostapd

- A hostapd program kapcsolódik a RADIUS szerverhez, illetve vezérli a supplicant portot, amely jelen esetben a létrehozott lokális link.
- A hostpad program paramétereit egy konfigurációs fájl segítségével állítjuk be
  - A fájlban a RADIUS szerver paramétereit és a vezetékes működési módot állítjuk be
  - A RADIUS IP címeket be kell állítani a valóságnak megfelelően

Hostapd konfigurációs fájl

```
interface=tunnel0
driver=wired
use_pae_group_addr=1
ieee8021x=1
eap_reauth_period=3600
#own_ip_addr=127.0.0.1
#nas_identifier=demo.ap
auth_server_addr=127.0.0.1
auth_server_port=1812
auth_server_shared_secret=testing123
acct_server_addr=127.0.0.1
acct_server_port=1813
acct_server_shared_secret=testing123
logger_stdout=1
logger_stdout_level=4
```

Hostapd futtatása

- A hostapd futtatása a következő paranccsal történik

```
$ sudo hostapd -dd -t hostapd.conf
```

- Amennyiben megfelelően van konfigurálva, nincs komolyabb hibaüzenet

```
1543962072.430604: tunnel0: interface state UNINITIALIZED-
>ENABLED
1543962072.430718: tunnel0: AP-ENABLED
1543962072.430795: tunnel0: Setup of interface done.
1543962072.430871: ctrl_iface not configured!
1543962072.431024: RADIUS message: code=5 (Accounting-
Response) identifier=0 length=20
```

Supplicant

- A Supplicant az EAPoL protokoll segítségével fog kapcsolatot kezdeményezni a port felnyitásáért az Authenticatornál
- A Supplicant paramétereit egy konfigurációs fájlba írjuk
- Jelen esetben a RADIUS szerverünk nagyon sok hitelesítést támogat, így a Supplicant konfigurációja fogja eldönteni, hogy melyik EAP metódus legyen

Wpa_supplicant konfigurációs fájl példa

- Az EAP-MD5 módszerhez használt konfigurációs fájl

```
ctrl_interface=/var/run/wpa_supplicant
network={
   proto=WPA
   key_mgmt=WPA-EAP
   pairwise=CCMP
   eap=MD5
   identity="test"
   password="password"
}
```

Supplicant indítása

- A Supplicant a következő paranccsal indul
  - Az utolsó paraméter a konfigurációs fájl neve

```
$ sudo wpa_supplicant -Dwired -itunnel0 -c client_PEAP-
MSCHAPv2.conf
```

- Sikeres indulásnál megtörténik a hitelesítés, majd a következő válasszal tér vissza:

```
tunnel0: CTRL-EVENT-EAP-SUCCESS EAP authentication
completed successfully
```

Supplicant forgalma

- A Supplicant az EAPoL protokollal kommunikál. A *tcpdump* parancs képes ezt a forgalmat is elmenteni
- Az elmentéshez használt parancs:

```
$ sudo tcpdump -i any -A -w file.pcap -e ether proto
0x888e or port 1812
```

- A fájl átmásolása és vizsgálata korábban leírtak alapján történik

4. Feladat

MSCHAPv2 Hitelesítés

Wpa_supplicant konfigurációs fájl

- Az MSCHAPv2 módszerhez használt konfigurációs fájl

```
ctrl_interface=/var/run/wpa_supplicant
network={
   proto=WPA
   key_mgmt=WPA-EAP
   pairwise=CCMP
   eap=MSCHAPV2
   identity="test"
   password="password"
}
```

5. Feladat

EAP-TTLS-PAP hitelesítés

Wpa_supplicant konfigurációs fájl

- Az EAP-TTLS-PAP módszerhez használt konfigurációs fájl

```
ctrl_interface=/var/run/wpa_supplicant
network={
   proto=WPA
   key_mgmt=WPA-EAP
   pairwise=CCMP
   eap=TTLS
   identity="test"
   anonymous_identity="senki@sehol.hu"
   password="password"
   phase2="auth=PAP"
}
```

6. Feladat

PEAP-MSCHAPV2 hitelesítés

Wpa_supplicant konfigurációs fájl

- A PEAP-MSCHAPv2 módszerhez használt konfigurációs fájl:

```
ctrl_interface=/var/run/wpa_supplicant
network={
   proto=WPA
   key_mgmt=WPA-EAP
   pairwise=CCMP
   eap=PEAP
   identity="test"
   anonymous_identity="senki@sehol.hu"
   password="password"
   phase2="auth=MSCHAPV2"
}
```