Sistemi za detekcijo napadov

Domen Kožar, Andraž Brodnik

2. januar 2013

Kazalo

1 Povzetek

Teoretični del zajema razlago sistemov za detekcijo napadov (IDS) in preprečevanje napadov (IPS) ter njihove podskupine. Osredotočili se bomo na sisteme, ki opazujejo ves promet na omrežnem vmesniku (NIDS/NIPS).

V praktičnem delu pa smo namestili in nastavili sistem Snort.

2 Uvod

Namen seminarske naloge se je seznaniti z sistemi za zaznavanje vdorov ter sistemi za preprečevanje vdorov. Kako sestaviti osnovno politiko (policy) za taksistem, kakšne napade lahko detektiramo ter so priporočljivi obrambni mehanizmi.

Dandanes se srecujemo z novicami o nepoblascenih vdorih v informacijske sisteme. Taksni vdori lahko unicijo podjetje, kar pomeni, da je racionalno investirati nekaj tehnicnih ur v postavitev sistema, ki bi (ni pa nujno) taksen vdor preprecil ali pa zaznal poskus vdora. Ce pa je prepozno pa vsaj omili posledice. To nam lahko koristi, da vidimo na kaksen nacin je napadalec napadel nas sistem, ter kaj je storil.

Kljub temu, da je nasa varnostna politika popolna (menjava gesel, dvonivojska avtentikacija v sisteme, pozarni zidovi, varne aplikacije, tuneliranje prometa, utrjeni strezniki), ne smemo biti prevec zadovoljni z sami sabo ter postavit se sistem za detekcijo in/ali preprecevanje napadov. Taksno razmislanje nam koristi tudi, pri razvoju aplikacij, zato pisimo teste, saj noben ni popoln.

Naj bralca opozorimo tudi na zakonodajo, 'nepravilno' konfigurirani IDS ali IPS sistemi lahko globoko posezejo tudi v zasebnost posameznika, kar ni v skladu z ustavo Republike Slovenije in drugimi pravnimi akti. IDS in IPS sistemi se lahko uporabljajo tudi DPI (deep packet inspection), kar pomeni, da ne gledamo samo glav paketnih protokolov ampak tudi aplikacijski nivo (aplikacijski protokol ali vsebino), zato nastavljajmo IDS in IPS sisteme odgovorno, podatke pa shranjujmo z najvecjo skrbjo.

3 Teoretični del

3.1 Pregled tipov sistemov

3.1.1 IDS

Slika 1: MAC flooding napad

Sistemi za detekcijo napadov (intrusion detection system), krajse IDS. So sistemi, katera naloga je analizirati podatke na omrezju ali sistemu samem ter zaznati poskuse udora ali pa udor sam.

Naj omenimo, da so tej sistemi namenjeni ponudnikom storitev (podjetjem, institucijam, posameznikom) v vecini niso namenjeni omreznim operaterjem, razen ce zelimo preprecevati napade na nase omrezne elemente. Ne moremo pa vsiliti pravil za vse nase uporabnike. Vcasih pa je tega prometa za analizo prevec, a vendar so se casi spremenili in to ni glavna omejitev.

Delimo jih na dve glavni skupini:

- - NIDS (network intrusion detection system)
- - HIDS (host intrusion detection system)

3.1.2 NIDS

Sistemi, katerim je glavni vir podatkov za analizo omrezje samo se imenujejo NIDS sistemi (network intrusion detection system).

Bralcu bo po vsej verjetnosti poznan program WireShark ali pa tcpdump. NIDS ponavadi delujejo podobno kot zgoraj omenjena programa, zajema vse paketke, ki jih vidi na omreznem vmesniku, nato jih premerja z pravili v svoji bazi, sumljive pakete ali niz paketov pa zabelezi ali pa si ustrezno napise informacije o njih. Sistemi se ponavadi zavedajo tudi prejsnih paketov, spremljajo transportni protokol. Torej je v njih kar nekaj logicnih avtomatov z pomnilnikom in razlicnimi stanji.

3.1.3 Primeri napadov

3.1.4 HIDS

3.1.5 IPS

3.1.6 NIPS

3.1.7 HIPS

Overitelj (authenticator) deluje kot posrednik v komunikaciji med odjemalcem in avtentikacijskim strežnikom. Nadzira fizični dostop do omrežja glede na podatke avtentikacije odjemalca. Uporabnikom katerim želijo dostopati, do omrežja vsiljuje predhodno overjanje. Podatke prenese

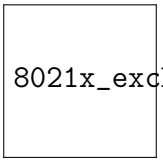
avtentikacijskemu strežniku in reagira glede na odziv strežnika (port odpre verificiranemu odjemalcu). V žičnem ethernet omrežju je to fizičen port na stikalu, na brezžičnem ethernet omrežju pa je overitelj logičen LAN port na brezžični dostopovni točki.

Odjemalec (supplicant) je sistem, ki zahteva dostop do omrežja. Ta potrebuje software, ki podpira protokol IEEE 802.1x (windows 7, ubuntu, ... ali namensko programje).

Avtentikacijski strežnik overja poverilnice odjemalca, ki jih posreduje overitelj, nato odgovori overitelju ali je odjemalec avtoriziran za dostop do omrežja. Overitelj posreduje odjemalčevo zahtevo po overjanju do avtentikacijskega strežnika.

Omrežno stikalo pred uspešno avtentikacijo dovoljuje samo EAPOL promet, ki ga preusmeri v avtentikacijskemu strežniku.

Postopek avtentikacije:



Slika 2: 802.1x potek avtentikacije

1. Če se odjemalec pridruži žičnemu omrežju, prejme od stikala poziv za avtentikacijo – paket »EAP-Request/Identity«. V kolikor se odjemalec prridruži brezžični dostopovni točki, odjemalec sam prične s pošiljanjem paketa »EAP-Start« za avtentikacijo.
2. Odjemalec pošlje stikalu paket »EAP-Request/Identity«, ki vsebuje identiteto fizične naprave (računalnika), ta pa ga posreduje strežniku
3. RADIUS strežnik odgovori odjemalcu zahtevo po pridružitvi - RADIUS sporočilo »Access-Challenge«, ki vsebuje avtentikacijsko sporočilo »RADIUS EAP-Request«
4. Odjemalec odgovori na zahtevo po pridružitvi z »EAP-Response« s poverilnico. To sporočilo je odjemalčev pozdrav avtentikacijskemu strežniku. Overitelj posreduje EAP sporočilo do RADIUS strežnika v obliki »RADIUS Access-Request«
5. RADIUS odda sporočilo »Access-Challenge«, ki vsebuje sporočilo »EAP-Request« ter digitalni certifikat RADIUS strežnika. Overitelj nato posreduje to sporočilo do odjemalca
6. Odjemalec pošlje »EAP-Response« ter svoj digitalni certifikat. Overitelj posreduje sporočilo do RADIUS strežnika v obliki »RADIUS Access-Request«.
7. Radius pošlje »EAP-Request« ter šifrirni material.
8. Odjemalec pošlje strežniku »EAP-Response«. Overitelj nato posreduje sporočilo do RADIUS strežnika v obliki«RADIUS Access-Request«
9. RADIUS odgovori odjemalcu s sporočilom o uspešnosti vzpostavitve seje »EAP-Success«

3.1.8 Napadi na 802.1x

802.1x podpira različne variante EAP za avtentikacijo. Glede na tip protokola, so možni različni napadi:

Tabela 1: EAP nacini avtentikacije

	EAP-MD5	LEAP	EAP-TLS	EAP-TTLS	PEAP
Server Authentication	None	Password Hash	Public Key (Certificate)	Public Key (Certificate)	Public Key (Certificate)
Supplicant Authentication	Password Hash	Password Hash	Public Key (Certificate or Smart Card)	CHAP, PAP, MS-CHAP(v2), EAP	Any EAP, like EAP-MS-CHAPv2 or Public Key
Dynamic Key Delivery	No	Yes	Yes	Yes	Yes
Security Risks	Identity exposed, Dictionary attack, Man-in-the-Middle (MitM) attack, Session hijacking	Identity exposed, Dictionary attack	Identity exposed	MitM attack	MitM attack; Identity hidden in Phase 2 but potential exposure in Phase 1

Napad na 802.1x naj bi bil mogoč preko brezžičnega dostopa – Man in the Middle napad, izvršen na EAP avtentikacijo. Napadalec pošlje odjemalcu lažno »EAP-Success« sporočilo, tako da zgleda, kot da je poslano s strani overitelja. To sporočilo ne vsebuje celovitosti ohranjanja informacij? in tako povzroči, da se oba – overitelj in odjemalec postavita v overjeno stanje. To naj bi napadalcu omogočilo da se postavi vmes.

Druga ranljivost je napad na obstoječo sejo. Po tem ko se je odjemalec avtenticiral pri avtentikacijskem serverju, se overitelj in klient postavita v overjeno stanje. Takrat lahko napadalec pošlje odjemalcu lažni ločitveni ovkir z MAC naslovom overitelja. Odjemalca tako loči od omrežja, ter pusti overitelja v povezanem stanju. Napadalec dobi dostop do omrežja z zbiranjem sistemskih MAC naslovov z naslovom pravkar ločenega sistema.

Orodje za izvajanje napadov na 802.1x preko kabla se imenuje Marvin [1]

3.1.9 IEEE 802.1x in port security

Možna je nastavitev port security-ja na portu z IEEE 802.1x. Ko omogočimo port security in IEEE 802.1x avtentikacijo na portu, 802.1x avtenticira port, port security pa nadzira omrežni dostop za vse MAC naslove. Nastavimo lahko število klientov, ki imajo dostop do omrežja preko IEEE 802.1x porta.

802.1x že sam po sebi lahko vodi evidenco in število prijavljenih MAC naslovov. Administrator lahko tako izbira kje (ali na 802.1x avtentikacijskem strezniku ali pa z port-security na portu samem) bo poskrbel za obrambo pred tovrstnimi napadi.

[1] http://www.gremwell.com/marvin-mitm-tapping-dot1x-links

4 Praktični del

4.1 802.1x

4.1.1 Nastavitev Cisco Catalyst 2960 omrežnega stikala

V laboratoriju sva dobila v roke omrežno stikalo **Cisco Catalyst 2960**. Naprava je imela privzete tovarniške nastavitve, tako sva jo preko konzole oz. serijskega porta priklopila na PC. Na Ubuntu 10.10 sistemu sva namestila **minicom** program in ga uporabila za komunikacijo z omrežnim stikalom:

```
sudo apt-get install minicom
minicom -s
```

Potrebno je bilo nastaviti minicom na pravilno pot */dev/ttyS0* in ze sva bila v terminalu. Prvi korak je bila uspesna postavitev DHCP strežnika, čeprav nisva bila prepričana, če stikalo to sploh podpira (saj je to naloga 3. nivoja). Še preden sva lahko nastavila karkoli, je bilo potrebno vzdigniti privilegije in preiti v tako imenovani administratorski nivo:

```
enable
```

Nato sva nastavila in pognala dhcp strežnik na omrežju *10.1.1.0* z masko */24* ter privzetim usmerjevalnikom *10.1.1.1*:

```
configure terminal
ip dhcp pool pool1
network 10.1.1.0 255.255.255.0
default-router 10.1.1.1
exit
service dhcp
```

Zanimivo je bilo odkriti, da je potrebno nastaviti IP naslov vlan interface-u, tako da se obnaša kot nekakšen usmerjevalnik (za mangement):

```
interface vlan 1
ip address 10.1.1.1 255.255.255.0
```

Naslednji ukaz vklopi vlan 1 interface, kar pomeni da je sedaj omogočen:

```
no shutdown
```

Naslednji ukaz nastavi spremembe na omrežnem stikalu in cez par sekund sva priklopila laptop, ki je dobil IP 10.1.1.2 preko DHCP-ja:

```
end
```

Naslednji korak je nastavitev 802.1x protokla in namestitev podatkov za FreeRadius strežnik. Vklopimo "aaa"funkcionalnost:

```
configure terminal
aaa new-model
```

Nastavive parametrov za radius strežnik je potrebno spraviti v skupino, v kateri je lahko več kot eden strežnik zaradi redundančnosti:

```
radius-server host 10.1.1.2 auth-port 1812 acct-port 1813 key testing123
aaa group server radius radius1
server 10.1.1.2 acct-port 1813 auth-port 1812
exit
```

Zelo pomembno je nastaviti *source-interface* na interface-u, torej informacija radius strežniku od kod prihajajo zahtevki. Freeradius strežnik na podlagi IP naslova izbere pravilne nastavitve in preveri vnesene podatke uporabnika. V našem primeru se uporabi vlan1 interface:

"testing123"je geslo, ki si ga delita NAS ter Radius strežnik za enkripcijo izmenjave podatkov.

```
ip radius source-interface vlan1
```

Povežemo še 802.1x autentikacijo z radius skupino, ki smo jo ustvarili in omogocimo 802.1x na omrežnem stikalu:

```
aaa authentication dot1x default group radius group radius1
dot1x system-auth-control
```

Še kot zadnji korak je potrebno omogočiti *port-based security* na samih fizičnih portih, in sicer tokrat samo na mestih od 5 do 10. Pravtako je potrebno nastaviti porte na *accces mode*, torej ne podpirajo *trunking* načina:

```
interface range FastEthernet0/5-10
switchport mode access
dot1x port-control auto
end
```

Z naslednjim ukazom prikažemo trenutne nastavitve stikala in jih nato prenesemo v boot nastavitve:

```
show running-config
copy running-config startup-config
```

4.1.2 Nastavitev FreeRadius strežnika na Ubuntu 11.04 laptop

Na enega iz klientov omrežnega stikala sva namestila FreeRadius strenižnik z naslednim ukazom:

```
sudo apt-get install freeradius
```

V datoteko /etc/freeradius/client.conf vključimo:

```
client 10.1.1.0/24 {
        ipaddr = 10.1.1.1
        secret  = testing123
        shortname = *
        require_message_authenticator = no
        nastype = other
}
```

Datoteka *client* dovoli omrežnemu stikalu z IP-jem 10.1.1.1 možnost avtentikacije.

V datoteko /etc/freeradius/users vključimo:

```
foobar  Cleartext-Password := "test"
        Service-Type = NAS-Prompt-User
```

Datoteka *users* je podatkovna baza dovoljenih uporabnikov, v našem primeru smo dovolili dostop uporabniku *foobar* z geslom *test*.

Po vnosu novih parametrov je potrebno strežniku še sporočiti, naj jih na novo naloži:

```
sudo /etc/init.d/freeradius reload
```

4.1.3 Nastavitev Ubuntu 10.10 klienta

Ugotovila sva, da v Ubuntu 10.10 orodje za upravljanje z omrežnimi povezavami (Network Manager 8.1) avtomatsko ne prepozna 802.1x protokola, kar sva tudi prijavila [1] na Ubuntu bug tracker.

Tako sva ročno za dano povezavo vnesila podatke:

- Use 802.1x security
- Authentication: Protected EAP (PEAP)
- PEAP version: Automatic
- Inner Authentication: MSCHAPv2
- Username: foobar
- Password: test

4.1.4 Testiranje protokola

Testiranje sva izvaja sproti, ker sva omrežje postavljala prvic, nama je to dalo povratno informacijo kje se trenutno zatakne.

Že pred uporabo laboratorija sva doma postavila FreeRadius strežnik in ga potestirala s pomočjo ukaza radtest, katerega parametri so naslednji:

```
# radtest <uporabniško ime> <geslo> <naslov strežnika> <port izvora zahtevka> <geslo med N
```

Po uspešni authentikaciji dobimo naslednji odgovor:

```
root@kaki:~# radtest foobar test localhost 1813 testing123
Sending Access-Request of id 50 to 127.0.0.1 port 1812
        User-Name = "test"
        User-Password = "test"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 1813
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=50, length=26
        Service-Type = NAS-Prompt-User
```

Sedaj sva imela delujoč Radius strežnik. Vklopila sva najvecji nivo izposovanja podatkov v loge, zato da sva lahko spremljala nadaljne zahtevke, ki bodo prihajali iz omrežnega stikala.

Precej težav nama je povzročal FreeRadius, saj je zavračal zahtevke omrežnega stikala. Kasneje sva dognala, da je potrebno določit šource-interface"parameter, da omrežno stikalo lahko pove iz kategera omrežja je prišel zahtevek.

Pri nastavljanju klienta je bilo kar nekaj težav glede podprtih protokolov za avtentikacijo. Po nekaj branja in probavanja sva izbrala najlažje, torej tiste z čimmanj komplikacij (zelo običajno se postavi Radius strežnik v proxy način, tako da glede na domeno pošilja zahtevke naprej drugim Radius strežnikom).

Nazadnje uspešna avtentikacija izgleda v logih Radius strežnika takole:

```
Thread 1 got semaphore
Thread 1 handling request 8, (2 handled so far)
[<thread>] # Executing section authorize from file /etc/freeradius/sites-enabled/default
[<thread>] +- entering group authorize {...}
++[preprocess] returns ok
++[chap] returns noop
++[mschap] returns noop
++[digest] returns noop
[suffix] No '@' in User-Name = "test", looking up realm NULL
[suffix] No such realm "NULL"
++[suffix] returns noop
[eap] EAP packet type response id 8 length 80
[eap] Continuing tunnel setup.
++[eap] returns ok
Found Auth-Type = EAP
# Executing group from file /etc/freeradius/sites-enabled/default
+- entering group authenticate {...}
[eap] Request found, released from the list
[eap] EAP/peap
[eap] processing type peap
[peap] processing EAP-TLS
[peap] eaptls_verify returned 7
[peap] Done initial handshake
[peap] eaptls_process returned 7
[peap] EAPTLS_OK
[peap] Session established.  Decoding tunneled attributes.
[peap] Peap state phase2
```

```
[peap] EAP type mschapv2
  PEAP: Setting User-Name to test
# Executing section authorize from file /etc/freeradius/sites-enabled/inner-tunnel
+- entering group authorize {...}
++[chap] returns noop
++[mschap] returns noop
[suffix] No '@' in User-Name = "test", looking up realm NULL
[suffix] No such realm "NULL"
++[suffix] returns noop
++[control] returns noop
[eap] EAP packet type response id 8 length 6
[eap] No EAP Start, assuming it's an on-going EAP conversation
++[eap] returns updated
[files] users: Matched entry test at line 205
++[files] returns ok
++[expiration] returns noop
++[logintime] returns noop
[pap] WARNING: Auth-Type already set.  Not setting to PAP
++[pap] returns noop
Found Auth-Type = EAP
# Executing group from file /etc/freeradius/sites-enabled/inner-tunnel
+- entering group authenticate {...}
[eap] Request found, released from the list
[eap] EAP/mschapv2
[eap] processing type mschapv2
[eap] Freeing handler
++[eap] returns ok
Login OK: [test/<via Auth-Type = EAP>] (from client * port 0 via TLS tunnel)
  WARNING: Empty post-auth section.  Using default return values.
# Executing section post-auth from file /etc/freeradius/sites-enabled/inner-tunnel
[peap] Tunneled authentication was successful.
[peap] SUCCESS
++[eap] returns handled
Finished request 8.
```

Jedrnat povzetek dogajanja:

- strežnik dobi zahtevek in ga začne obdelovati
- pogleda za katero vrsto avtentikacije gre in jo izvede
- vzpostavi TLS varno povezavo
- znotraj varnega tunela preveri ujemanje uporabniškega imena in gesla ter vrne OK

Log Radius strežnika ob napačnem geslu:

```
Thread 5 handling request 17, (4 handled so far)
```

```
[<thread>] # Executing section authorize from file /etc/freeradius/sites-enabled/default
[<thread>] +- entering group authorize {...}
++[preprocess] returns ok
++[chap] returns noop
++[mschap] returns noop
++[digest] returns noop
[suffix] No '@' in User-Name = "test", looking up realm NULL
[suffix] No such realm "NULL"
++[suffix] returns noop
[eap] EAP packet type response id 17 length 144
[eap] Continuing tunnel setup.
++[eap] returns ok
Found Auth-Type = EAP
# Executing group from file /etc/freeradius/sites-enabled/default
+- entering group authenticate {...}
[eap] Request found, released from the list
[eap] EAP/peap
[eap] processing type peap
[peap] processing EAP-TLS
[peap] eaptls_verify returned 7
[peap] Done initial handshake
[peap] eaptls_process returned 7
[peap] EAPTLS_OK
[peap] Session established.  Decoding tunneled attributes.
[peap] Peap state phase2
[peap] EAP type mschapv2
  PEAP: Setting User-Name to test
# Executing section authorize from file /etc/freeradius/sites-enabled/inner-tunnel
+- entering group authorize {...}
++[chap] returns noop
++[mschap] returns noop
[suffix] No '@' in User-Name = "test", looking up realm NULL
[suffix] No such realm "NULL"
++[suffix] returns noop
++[control] returns noop
[eap] EAP packet type response id 17 length 63
[eap] No EAP Start, assuming it's an on-going EAP conversation
++[eap] returns updated
[files] users: Matched entry test at line 205
++[files] returns ok
++[expiration] returns noop
++[logintime] returns noop
[pap] WARNING: Auth-Type already set.  Not setting to PAP
++[pap] returns noop
Found Auth-Type = EAP
# Executing group from file /etc/freeradius/sites-enabled/inner-tunnel
+- entering group authenticate {...}
```

```
[eap] Request found, released from the list
[eap] EAP/mschapv2
[eap] processing type mschapv2
[mschapv2] # Executing group from file /etc/freeradius/sites-enabled/inner-tunnel
[mschapv2] +- entering group MS-CHAP {...}
[mschap] Creating challenge hash with username: test
[mschap] Told to do MS-CHAPv2 for test with NT-Password
[mschap] FAILED: MS-CHAP2-Response is incorrect
++[mschap] returns reject
[eap] Freeing handler
++[eap] returns reject
Failed to authenticate the user.
Login incorrect: [test/<via Auth-Type = EAP>] (from client * port 0 via TLS tunnel)
[peap] Tunneled authentication was rejected.
[peap] FAILURE
++[eap] returns handled
Finished request 17.
```

Na koncu klient za pinga nek drug client na omrežnem stikalu, da potrdimo vzpostavljeno povezavo.

4.2 Port security

4.2.1 Nastavitev Cisco Catalyst 2960 omrežnega stikala

Podpore za 802.1x vključno z port security na najinem omrežnem stikalu ni blo, saj je podprta šele v višjih verzijah operacijskega sistema (to nazorno piše v navodilih in pravtako sva sama preizkusila). Včasih kombinacija ni bila podprta, saj 802.1x protokol lahko sam izvaja nadzor nad številom MAC naslovov glede na port. Zato sva pretestirala protokol ločeno in nastavila stikalo:

```
configure terminal
interface FastEthernet0/2
switchport port-security
switchport port-security maximum 1
switchport port-security violation shutdown
```

Nastavila sva vse skupaj na stikalu 2, z maksimalno enim MAC naslovom in povedala stikalu naj ga izklopi v primeru prekoračitve dovoljenega števila.

4.2.2 Testiranje protokola

Namestitev programa na Ubuntu 10.10 za izvajanje MAC flooding napada je zelo enostavna, kakor je tudi poganjanje le-tega:

```
sudo apt-get install dsniff
sudo macof -i eth0
```

To pomeni, da čez interface eth0 pošilja ogromno lažnih paketkov, ki napolnijo MAC tabelo omrežnega stikala. Ko je bil port na omrežnem stikalu zaščiten in nastavljen z port security, sva na konzoli opazila naslednje opozorilo in port se je izklopil:

```
01:12:45: %PM-4-ERR_DISABLE: psecure-violation error detected on Fa0/2, putting Fa0/2 in e
01:12:45: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC a
01:12:46: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2, changed state t
01:12:47: %LINK-3-UPDOWN: Interface FastEthernet0/2, changed state to down
```

To dokazuje uspešno obrambo pred napadom na 2. nivoju.

5 Rezultati

802.1x protokol uspešno omejuje dostop do omrežja. Prav tako port security uspesno nadzira število MAC naslovov na posamezen port in tako ščiti pred več napadi.

Nažalost protokolov nisva mogla testirat in uporabljat skupaj, ker firmware verzija na omrežnem stikalu tega ni dopuščala.

Literatura

[1] Prijavljen predlog izboljšave avtomatskega prepoznavanja 802.1x protokola na Ethernet-u, https://bugs.launchpad.net/bugs/783560, 21.05.2011

[2] Navodila za uporabo yersenia paketa, http://manpages.ubuntu.com/manpages/hardy/man8/yersinia.8.html, 21.05.2011

[3] Domača stran paketa dsniff, http://monkey.org/~dugsong/dsniff/, 21.05.2011

[4] Tabela 1, http://www.wi-fiplanet.com/tutorials/article.php/3075481/Deploying-8021X-for-WLANs-EAP-Types.htm, 21.05.2011

[5] Slika 1, http://www.ciscozine.com/wp-content/uploads/mac_flooding_attack.png, 21.05.2011

[6] Slika 2, http://media.packetlife.net/media/blog/attachments/332/vlan_hopping_attack.png, 21.05.2011

[7] Slika 3, http://www.skapadmin.net/uploads/1/MAC_Spoofing.JPG, 21.05.2011

[8] Slika 4, http://upload.wikimedia.org/wikipedia/commons/1/1f/802.1X_wired_protocols.png, 21.05.2011

[9] Slika 5, http://www.cisco.com/en/US/i/100001-200000/100001-110000/101001-102000/101228.jpg, 21.05.2011

[10] Video posnetek predavanja na defcon, https://media.defcon.org/dc-16/video/Defcon16-Figueroa-Figueroa-Williams-VLANs_Layer2_Attacks.m4v, 21.05.2011

[11] Specifična navodila za Cisco naprave za FreeRadius strežnik, http://wiki.freeradius.org/Cisco, 21.05.2011

[12] Predstavitev napadov, http://www.blackhat.com/presentations/bh-europe-05/BH_EU_05-Berrueta_Andres/BH_EU_05_Berrueta_Andres.pdf, 21.05.2011

[13] Navodila za Cisco Catalyst 2960, http://www.cisco.com/en/US/docs/switches/lan/catalyst2960/software/release/12.2_46_se/configuration/guide/2960SCG.pdf, 21.05.2011

[14] Predstavitev layer 2 napadov, http://www.cisco.com/warp/public/cc/so/cuso/epso/sqfr/sfblu_wp.pdf, 21.05.2011

[15] Predstavitev napadov, http://www.iphelp.ru/faq/12/ch03lev1sec8.html, 21.05.2011

[16] Predstavitev napadov, http://www.javvin.com/networksecurity/NetworkSecurity.html, 21.05.2011

[17] Predstavitev obrambnih mehanizmov layer 2, http://www.sanog.org/resources/sanog7/yusuf-L2-attack-mitigation.pdf, 21.05.2011