SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1762

Procjena ranjivosti mrežnih sustava

Krešimir Valdec

Voditelj: *doc. dr. sc. Marin Golub*

Zagreb, listopad 2008.

Sažetak

U ovom radu je analizirana automatizirana procjena ranjivosti računalnih mreža kao tehnički i organizacijski proces. Proučene su poznate vrste prijetnji i osnovni načini obrane od svake zasebno. Izložena je i teorijska osnova za sistematičan način prevencije sigurnosnih incidenata i prilagodbu procesa procjene ranjivosti za različite mrežne sustave. Kao potpora teorijskoj razradi, opisana su i korištena tri alata za procjenu ranjivosti te je prikazan primjer primjene opisane metode na pravoj produkcijskoj mreži.

Abstract

This diploma thesis analyzes automated vulnerability assessment of computer networks as a tehnical, as well as an organizational process. Various threats and related ways of defense are discussed. The paper also describes a theoretical basis for a systematic approach to prevent security incidents and to adapt the process to different network systems. To support the theoretical analysis, three vulnerability assessment tools are described and used in an example of applying the described method in a real-life, production network.

Sadržaj

1. Uvod

Razvoj elektroničkih računala, kao što je dobro poznato, uzrokovao je najdublje i najbrže promjene u načinu života cijelog čovječanstva. Nova otkrića i nova postignuća na području računarstva imala su višestruki lavinski efekt – osim hranjenja vlastitog ubrzanog napretka, vjerojatno nema grane ljudske djelatnosti koja nije pozitivno reagirala na nove mogućnosti ubrzanog istraživanja i razvoja. No, digitalna pohrana, obrada tekstualnih, grafičkih i numeričkih podataka, mada revolucionarni pomaci sami za sebe, bili su tek uvod za najveći pojedinačni skok u povijesti računarstva, a to su mreže računala. Tek s umrežavanjem je povećana moć procesiranja podataka dobila posljednji potreban sastojak – ubrzani protok i distribuciju informacija i znanja. Krajnji rezultat je današnji svijet računarstva: računala su svugdje, neizostavni su dio naše dnevne rutine, a mreže su posvuda i neprestano ih koristimo. Mreže računala i Internet, "mreža svih mreža", postali su medij koji koristimo u svakom poslu, svaki put kada nas bilo što zanima, kada nešto učimo, kada trebamo prenositi podatke za bilo koju namjenu ili se jednostavno želimo družiti i opustiti. I taj medij nam je postao *neophodan*, na njega *računamo*.

Danas se gotovo svaka organizacija u većoj ili manjoj mjeri oslanja na uporabu Interneta u nekom segmentu poslovanja. Dapače, to više nije trend, već uobičajeni zahtjev korisnika koji to očekuju. Dodatno, korporacijski intraneti postali su osnova funkcioniranja većine firmi i sa svakim ispadom interne mreže se rad usporava ili čak zaustavlja. Drugim riječima, od navedenih tehnologija očekuje se da nas neumorno i bez puno problema služe, jer bez njih ne možemo.

No, "mreža svih mreža" nije sigurno mjesto i ponekad bez svega navedenog jednostavno moramo. Na Internet su od početka spojeni svi koji žele, uključujući i one zlonamjerne, a distribucija znanja, dislociranost i anonimnost jednako su pogodovale i njima. U najširem smislu i ne razmatrajući motive, njihov je cilj drugima uskratiti lijepa svojstva navedena ranije, u toj nakani su sve vještiji kako vrijeme prolazi, a kako je računarstvo sve više orijentirano prema mrežama, može se reći da je ta opasnost s vremenom sve ozbiljnija. Nasuprot njima stoji također veliko nastojanje da se Internet održi što sigurnijim, odnosno da se zaštite interne mreže svih korisnika, i to pomoću razne programske podrške, sigurnosnih politika i edukacije korisnika. Ipak, činjenica da su sigurnosne prijetnje došle prve, a zaštita tek nakon njih, duboko utječe na današnje stanje i načine obrane.

Današnje mreže su kompleksne. Raznovrsne su i mogu biti izuzetno velike, struktura im je ponekad vrlo složena, a u zadnjih par godina je i broj različitih vrsta naprava koje se mogu naći na mrežama jako porastao. Osim toga, razlikuju se po namjeni, po važnosti i cijeni, tehnologiji izvedbe i nebrojenim drugim osobinama. Zato je njihova zaštita zasebna industrija i već poduže je zasebno polje istraživanja. Postoje različite metodologije zaštite, koje u osnovi dijele namjeru da se zaštita sustava organizira kao višeslojan i višefazan proces. Početni korak svake metodologije je upravo tema ovog rada – procjena ranjivosti mrežnog sustava.

Procjena ranjivosti je složen i važan korak u osiguravanju mrežnih sustava. Njegova složenost dijelom leži i u tome što se nalazi na rubu poslovnog i tehničkog procesa – ima tehničke i netehničke ulazne faktore, zahtjeva međudisciplinarno znanje iz oba svijeta i nema "jednog pravog načina" za izvedbu. Cilj ovog rada je predložiti i demonstrirati jedan način izvedbe, imajući u vidu više od isključivo tehničkih aspekata sigurnosti. Naredna poglavlja prikazati će s kakvim se problemima susrećemo, što nam prijeti, kako se obraniti od pojedinih prijetnji i kako se sustavno obraniti, ovisno o karakteristikama ciljnog sustava. I, dakako, što je točno procjena ranjivosti, koje su njezine vrijednosti i što nam znači u procesu zaštite sustava.

2. Informacijska sigurnost mrežnih sustava

Ovo poglavlje počinje prikazom prijetnji s kojima se susrećemo u osiguranju mrežnih sustava. Naime, kako su stvarne mreže različite na mnogo načina, to još više vrijedi za ranjivosti i same napade kao glavne i najgore posljedice ranjivosti. Raznovrsni propusti koji se mogu naći u opremi ili programskoj potpori čine potencijal da se napad dogodi, a dodatna je komplikacija to što se načini iskorištavanja ranjivosti u pojedinim napadima mogu razlikovati, uzrokujući različitu razinu štete. Dodatno, osim tehničkih propusta koji rezultiraju sigurnosnom rupom u sustavu, isti se rezultat može postići na netehnički način - manpulacijom korisnicima, njihovom neopreznošću ili neznanjem. Mnogo je različitih izvora ranjivosti i oni rezultiraju velikim brojem načina, odnosno vektora napada (eng. *attack vector*), a njihovo razumijevanje, tj. razumijevanje prirode različitih prijetnji, daje nam bolju šansu za obranu i pomaže nam u podjeli posla kod procjene ranjivosti. Zato prvi dio ovog poglavlja donosi pregled i podjelu prijetnji prema mjestu nastanka ranjivosti. U nastavku poglavlja bit će opisani glavni napadi i stvarne prijetnje koje ugrožavaju mreže, a poglavlje završava opisom načina zaštite.

2.1 Ranjivosti uzrokovane programskom podrškom

Ovo je osnovna i najraširenija grupa ranjivosti. Statistički, najveći broj dokumentiranih i pronađenih rupa koje otvaraju mogućnosti proboja u neki sustav spada u ovu grupu. O čemu se točno radi?

Ranjivost u programskoj podršci predstavlja bilo kakav propust u dizajnu, arhitekturi ili izvedbi nekog programa (ili dijela programa), koji može dovesti do sigurnosnog proboja u računalni sustav [18]. Također, u istu grupu spadaju sigurnosni problemi uzrokovani uobičajenom uporabom određenih programa koji sami po sebi, ovisno o situaciji, mogu biti opasni ili sadrže neki izvršni dio koji može predstavljati prijetnju sigurnosti. Drugim riječima, programska podrška uvodi velik broj različitih vrsta ranjivosti u mreže, a prema tim razlikama grade se metode i načini obrane. Interesantno je i to što se osnovne grupe ranjivosti poklapaju s glavnim cjelinama u ciklusu života aplikacije. Naime, ranjivosti programske podrške se mogu podijeliti ovako:

- propusti u dizajnu aplikacije ili mrežnog protokola;
- propusti u programskoj izvedbi;
- propusti u konfiguraciji;
- namjena, način rada i skriveni dijelovi aplikacije.

Svaka od ovih točaka zahtijeva podrobniji opis.

2.1.1 Propusti u dizajnu aplikacije ili mrežnog protokola

Na početku treba reći da se ne radi uvijek strogo o propustu, pogotovo kada je riječ o mrežnim protokolima, već do određenih sigurnosnih problema može doći iskorištavanjem pozitivnih svojstava. Dobar primjer su određene vrste napada uskraćivanjem usluge (eng. *Denial of Service, "DoS"*), koje koriste temeljna svojstva protokola TCP (i o kojima će više riječi biti kasnije).

Do ove vrste ranjivosti došlo je ponajviše iz razloga koji je spomenut u uvodu – ranjivosti i napadi su došli prvi, a misao o sigurnosti tek nakon toga. Prvi protokoli razvijani su u okolini koja nije bila opasna, činili su je uglavnom dobronamjerni i entuzijastični istraživači, a glavni interes bio je razvoj protokola koji će osiguravati pouzdan prijenos informacija. Proboj

računala u svakodnevnu uporabu, pojava Interneta i zlonamjernih korisnika – i na kraju, ogroman porast broja sigurnosnih incidenata, doveli su do toga da se sigurnost kao tehničko rješenje uvodi *naknadno*, u već gotov "proizvod". Takva situacija rezultirala je, dakako, puno kompleksnijim rješenjima, a gotovo sigurno bi tehnička kakvoća bila na višoj razini, da se o sigurnosti razmišljalo u početku. Kao primjer se može navesti mrežni datotečni sustav NFS, koji u prvim verzijama nije imao autentifikaciju korisnika (što je kasnije dodano) [1].

Primjer - *idlescan*

Kao zanimljiviji primjer se može navesti "*idlescan*" ranjivost [2]. Naime, prvi korak svakog napada je otkrivanje cilja, odnosno stanice "žrtve" (dalje u tekstu će izraz "stanica" predstavljati svaku implementaciju TCP/IP protokolnog sloga koja se može pojaviti na ciljnoj mreži i uključuje klijente, poslužitelje, usmjernike, IP telefone itd.). U tu se svrhu koriste razni alati kojima je osnova rada snimanje (dalje u tekstu: skeniranje) otvorenih TCP ili UDP pristupa (eng. *port*) na način da se na pristup šalje neka vrsta zahtjeva (za TCP, to može biti zahtjev za uspostavu veze, prekid veze, itd.) i promatra odgovor. No, današnje mreže često su zaštićene sustavima za otkrivanje napada koji ovakvo skeniranje odmah prepoznaju i, ovisno o važnosti sustava i drugim parametrima, sposobni su brzo poduzeti mjere prema izvoru skeniranja. Isto tako, ovakvi napadi lako se primjećuju na meti skeniranja – radi se o klasičnom skeniranju i današnja programska podrška nudi velike mogućnosti prepoznavanja. Osim zatvaranja pristupa adresi (ili mreži) s koje skeniranje dolazi, moguća je i prijava davatelju internetskih usluga napadača, čime napadač može biti ugrožen. Ova ranjivost pomaže skrivanju tragova i očuvanju anonimnosti napadača, dozvoljavajući napadaču da za napad (tj. inicijalno skeniranje) okrivi drugoga. Riječ je, dakle, o relativno bezopasnom napadu, čija je primarna svrha zamaskirati vlastiti identitet kod skeniranja. S druge strane, može se reći da je ovaj propust ozbiljan jer otežava praćenje napada kao i forenziku, za slučaj da se napad doista ostvari.

Idlescan koristi uobičajena svojstva protokola IP i TCP i temelji se na "Identification" polju u zaglavlju IP paketa (dalje u tekstu: IP ID polje, vidi dodatak A). Riječ je o 16-bitnom polju namijenjenom da sadrži svojevrstan "serijski broj" paketa, pomoću kojeg bi se fragmentirani datagram sastavljao na odredištu. Svi fragmenti bi trebali imati isti taj broj, čime bi činili grupu. Ono što protokolom nije definirano (i time nije krivnja implementacije), je kako mijenjati taj broj između dva datagrama, tj. dva paketa ako do fragmentiranja nije došlo. Mnoge implementacije TCP/IP protokolnog sloga ovo rješavaju jednostavnim inkrementom za jedan i upravo je to temelj ovog napada u njegovoj originalnoj verziji, a za općenitiji slučaj je dovoljno da je sekvencioniranje tog polja predvidljivo, makar se radilo o kompliciranijem postupku od inkrementa.

Za takvo skeniranje potreban je napadač (na dijagramu označen s "E"), cilj (meta, na dijagramu "B") i jedna TCP/IP implementacija koja se koristi kao "zombi" (na dijagramu označen sa "A"). Cilj napada je skenirati otvorene pristupe na meti, a da pritom sa strane mete izgleda kao da je skenira zombi računalo. To zombi računalo mora imati opisanu implementaciju protokola IP i mora se odlikovati inače malim prometom. Primjer zombija koji zadovoljavaju oba preduvjeta su mrežni pisači, IP telefoni i brojne druge naprave.

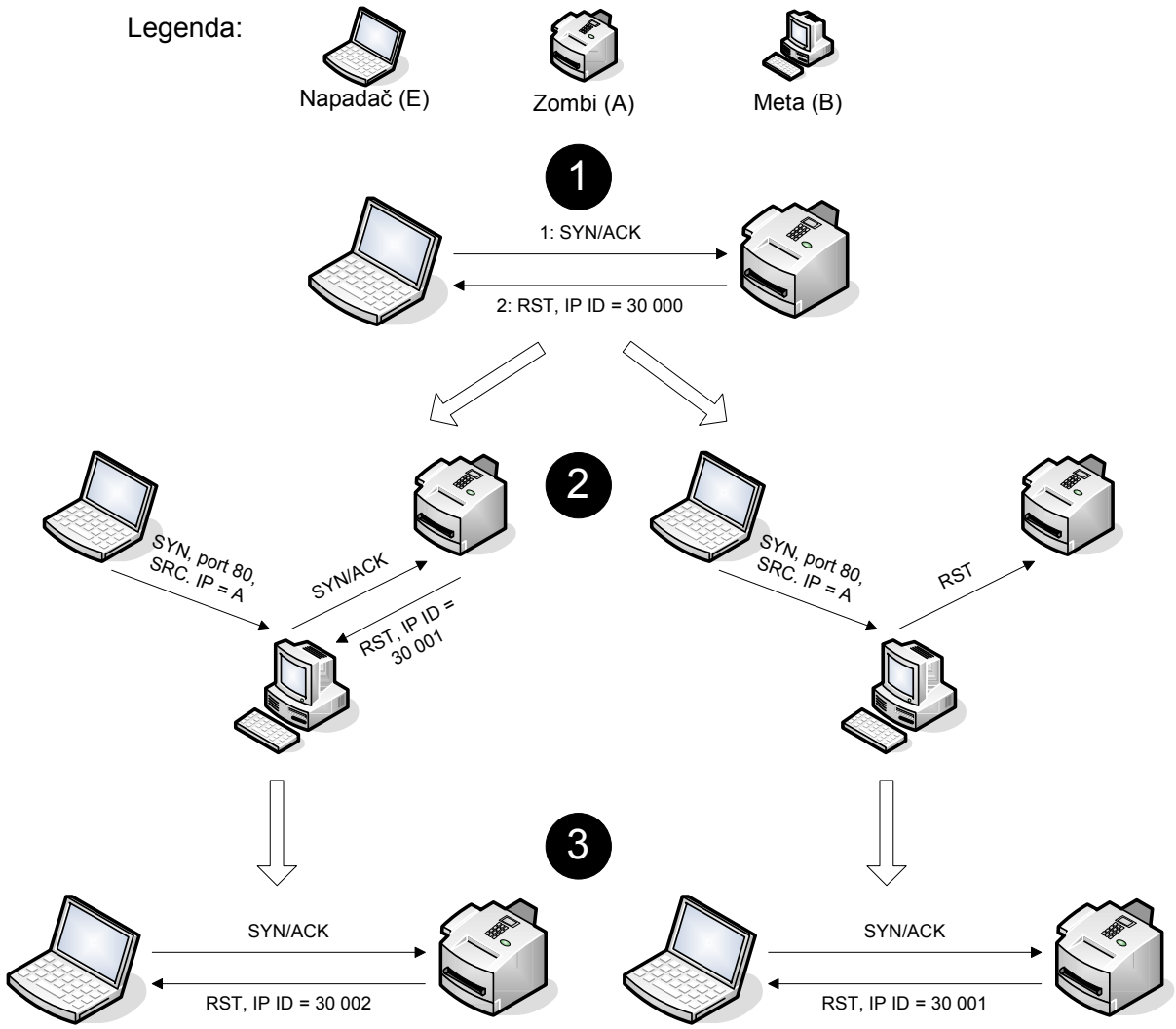
Napad se temelji na sljedećim činjenicama:

1. Ako TCP implementacija na otvoreni TCP pristup primi segment s postavljenom SYN zastavicom (a takav segment predstavlja zahtjev za uspostavljanjem sjednice), ona odgovara segmentom s postavljenim SYN i ACK (od eng. *acknowledge*) zastavicama, čineći drugi korak u standardnom uspostavljanju TCP veze. Ako je pristup zatvoren, odgovara segmentom s postavljenom RST (od eng. *reset*) zastavicom. Odgovori idu na IP adresu koja piše u izvorišnom polju paketa na koje odgovara.

2. Ako TCP implementacija primi SYN/ACK segment, a da nije prethodno poslala SYN segment, odgovara RST segmentom. Ako primi RST segment, ne odgovara nikako i ignorira RST (ne raskida vezu koja nije ni započeta).

Imajući na umu zahtjeve navedene ranije, napad se izvodi u sljedećim koracima (vidi sliku 2.1.):

1. Napadač šalje SYN/ACK segment zombiju, dobiva odgovor i pamti IP ID polje iz dobivenog IP paketa.
2. Koristeći (umjesto svoje) IP adresu zombija kao izvorište, napadač formira SYN segment s odredišnim pristupom postavljenim na pristup od interesa (pristup koje želi provjeriti). Ovo će uzrokovati ili SYN/ACK, ili RST segment poslan na adresu zombija.
3. Napadač ponovo šalje SYN/ACK segment zombiju, dobiva odgovor i promatra IP ID polje.



Slika 2.1. Iskorištavanje ranjivosti *Idlescan*

Prema stanju IP ID polja može se zaključiti jesu li odabrani pristupi na meti dostupni ili nisu. Ako je u drugom koraku odaslan SYN/ACK segment prema zombiju (a to znači da je pristup otvoren), IP ID na zombi implementaciji će se povećati za jedan, što se neće dogoditi ako je odaslan RST segment (tj. ako je pristup zatvoren). Treći korak uzrokuje još jedno povećanje IP ID i zaključak je jednostavan: ako je pristup otvoren, IP ID je povećan za 2, u protivnom je povećan za 1. Ako se IP ID razlikuje više od ovoga, to najvjerojatnije znači da zombi računalo istovremeno obavlja neku drugu komunikaciju i tada su rezultati nepouzdani.

Uz ovaj primjer vrijedi još napomenuti da prebacivanje krivnje za skeniranje na nekog drugog također može otežati rad sigurnosnih stručnjaka, trošeći vrijeme na određivanje pravog izvora.

Obrana od ove vrste napada ovisi odakle se napad pokreće. Ako je unutar vlastite mreže, može se koristiti sustav za otkrivanje napada na način da reagira na promet s lažiranim izvorišnim IP adresama. Ako je napad izvana, vjerojatno će biti uočen, no treba biti oprezan pri zaključku odakle skeniranje dolazi. Sa strane prevencije korištenja vlastitih resursa kao zombi računala, potrebno je posebno obratiti pozornost na TCP entitete koji su kandidati za zombije, odnosno zaštiti ih dizajnom mreže, a za to je potrebno znati o kojim se točno IP adresama radi. Procjena ranjivosti, detaljno opisana u zasebnom poglavlju, izvrsno rješava probleme katalogizacije resursa prema proizvoljnim kriterijima, pa tako i ovom. Primjerice, niti jedna takva naprava ne bi smjela biti vidljiva s Interneta, a ako je mreža opremljena sustavom za otkrivanje napada (eng. *Intrusion Detection System, IDS*), treba sustav konfigurirati tako da uzima ove naprave u obzir.

2.1.2 Propusti u izvedbi aplikacije ili protokola

Ova grupa ranjivosti programske podrške najbliže je čisto tehničkom problemu. Radi se o programerskim greškama u izradi programa koje mogu izazvati neki od brojnih poznatih problema, od kojih je daleko najpoznatiji preljev spremnika (eng. *buffer overflow* ili, rjeđe, *buffer overrun*). Osim preljeva spremnika, česti propusti su stanja utrke, propusti u upravljanju memorijom, nepravilan višeprocesni/višedretveni rad i slično [4].

Primjer – preljev spremnika

Preljevi spremnika mogu se podijeliti na preljeve spremnika na stogu te preljeve spremnika na hrpi i nisu slučajno najpoznatija ranjivost ove vrste. Do njih dolazi kada program pokuša u neki spremnik upisati veću količinu podataka nego što je predviđeno deklaracijom spremnika. Drugim riječima, podaci se pokušaju upisati u premali spremnik. To redovito rezultira modifikacijom podataka "susjednog" spremnika, tj. dolazi do preljeva podataka preko ruba spremnika, i upravo o važnosti (i ulozi) tog susjednog podatka ovisi koliko ovakva ranjivost može biti opasna. Dodatno, izvedba ovisi o arhitekturalnim detaljima ciljne platforme poput načina iskorištavanja memorije, načina adresiranja (*big/little endian*) itd.

Memorijska struktura "stog" sadrži argumente potprograma, lokalne varijable, i vrijednosti određenih registara. Kod spremnika na stogu, tipično se ova opasnost odnosi na povratnu adresu iz potprograma (adresu s koje se nastavlja izvođenje nakon završetka, tj. adresu sljedeće instrukcije nakon posljednje u potprogramu), odnosno napadač u toj varijanti može odrediti proizvoljnu povratnu adresu. Napad se tada realizira postavljanjem vlastitog zlonamjernog izvršnog teksta programa na povratnu adresu, efektivno izvršavajući proizvoljni izvršni tekst, i to s ovlastima koje je imao program koji je napadnut. Ovo spada u najveće i, nažalost, vrlo brojne prijetnje – iskorištavanje poznatih ranjivosti ovog tipa događa se svakodnevno, a priljev novih rupa je konstantan i velik. S druge strane, razina opasnosti i proširenost ove ranjivosti prouzrokovali su istraživanja i implementacije različitih načina obrane [5].

1. Opreznije programiranje i analiza izvornog teksta programa

   Ova ranjivost isključivo je programerska greška i mogla bi se u ogromnoj mjeri spriječiti kvalitetnijim programiranjem. Cijela se ranjivost zapravo temelji na izostanku provjere ulaznih podataka na duljinu, čemu pogoduje ogromna količina izvornog teksta napisana u programskom jeziku C, koji nema ugrađenu provjeru duljine polja. Pisanje kvalitetnijeg izvornog teksta iziskuje upoznavanje programera s ovim problemom i promjenu načina pisanja. Kao pomoć tome postoje i alati koji prolaze kroz izvorni tekst programa i prepoznaju tipične "sumnjive" pozive, npr. funkcija "strcpy", "sprintf" itd., mada se mora napomenuti da ti alati u određenoj mjeri griješe, najčešće upozoravajući na legitimne pozive. Temelj tog opreznijeg programiranja je detaljnija provjera (veličine) korisničkog ulaza u ranjive funkcije.

2. Zaštita stoga od izvršnog teksta

   Ova metoda se temelji na konfiguriranju stoga tako da ne može sadržavati izvršni kôd, čime se eliminira cijela klasa napada. Za operacijske sustave temeljene na UNIX-u (i slične) postoje zakrpe za jezgru, dok se ova zaštita kod Windows operacijskih sustava zove "Data execution prevention".

3. Dinamička zaštita

   Ova se vrsta zaštite provodi za vrijeme izvođenja (eng. *runtime*) i ima više tehnika:

   - obrana pomoću "kanarinaca":

     Izraz "kanarinac" odnosi se na nekadašnje iskorištavanje tih ptica u rudnicima kako bi *upozorile* na prisustvo otrovnih plinova. Ovdje je riječ o umetanju brojeva uz povratnu adresu. Prije povratka, kontrolira se stanje tog broja i ako je promijenjen, došlo je do pokušaja iskorištavanja preljeva spremnika. Ove mogućnosti dodaju se u prevodioce i njihovo se korištenje implementira prevođenjem. Kanarince se može umetati na više mjesta, čime se postiže veća sigurnost jer postaje puno teže promijeniti samo jedan, ciljani podatak, a da se pritom ne poremete druge važne informacije na stogu, čime i sam napad ne uspijeva.

   - obrana kopiranjem povratne adrese:

     U ovom se slučaju sama povratna adresa sprema odvojeno i vraća na originalnu u slučaju promjene.

   - provjera duljine polja:

     Ovdje je zapravo riječ o dodavanju dodatne provjere duljine povrh gotovog izvršnog teksta i ova tehnika bitno usporava rad programa.

Što se tiče preljeva spremnika na hrpi, treba imati na umu da je hrpa memorijska struktura na kojoj mjesto ne alocira operacijski sustav, već aplikacija i da se na njoj nalaze dinamički podaci koji pripadaju aplikaciji. Preljev spremnika na hrpi, dakle, neće uzrokovati mijenjanje ikakve povratne adrese. No, koristeći druge tehnike (prepisivanjem internih struktura aplikacije), jednako može rezultirati izvođenjem proizvoljnog izvršnog teksta napadača, kako je vidljivo iz [3]. Preljevi spremnika na hrpi nešto se rjeđe uspijevaju pretočiti u pravu ranjivost i zato se zaštiti od njih povijesno pristupalo manje sistematično. Današnja borba protiv tih propusta veže se uz pojedine incidente, i to krpanjem ranjivog programa.

Vrijedi još naglasiti da korisnički ulazni podaci nisu obavezno neki tekst koji će biti predugačak za alocirano polje programa pisanog u programskom jeziku C. Preljev spremnika

temelji se na takvoj jednostavnoj tehnici, odnosno koncept je jednostavan, ali načini realizacije su napredovali. Primjerice, preljev spremnika postignut je u popularnim i rasprostranjenim programskim paketima poput Microsoft Excela ili Winampa – jednostavnim otvaranjem posebno konstruirane datoteke. Korisniku se može dogoditi da jednostavnim otvaranjem Excel tablice, odnosno mp3 dokumenta ili priključka (eng. *plugin*) koji modificira Winampov izgled zapravo uzrokuje preljev spremnika s izvođenjem proizvoljnog zlonamjernog izvršnog kôda, time ugrožavajući prvu stanicu na nekoj mreži. U još gorem slučaju, nesiguran izvršni tekst može se naći u dinamičkoj, dijeljenoj biblioteci za, primjerice, JPEG kompresiju grafičkih datoteka – čime je ugroženo više od samo jednog programa [3]. Ovakvi scenariji zamijenili su puno manje opasne prethodnike, poput slučajnog pokretanja virusa primljenog u elektroničkoj pošti, i danas čine veliku opasnost u svakodnevnom korištenju računala.

No, kontrola duljine korisničkih ulaznih podataka samo je jedna stvar koja može uzrokovati probleme. Sasvim drugu vrstu napada na sasvim drugoj razini može izazvati sadržaj podataka. Najbolji primjer te, vrlo rasprostranjene vrste napada je SQL umetanje (eng. *SQL code injection*) [6].

Treba reći da je upravo izloženo razmatranje zaštite od obje vrste preljeva spremnika zapravo nepotpuno. Tehnički odgovori su ispravni, no s organizacijske strane oni nisu dovoljni i nije zapravo jasno kako se u stvarnim situacijama rješava ovakav proboj. Mada će o tim aspektima zaštite više biti rečeno kasnije, treba napomenuti da većina stvarnih, produkcijskih situacija ne dozvoljava intervencije poput promjene konfiguracije stoga, bez prethodne analize utjecaja. Također, postavlja se pitanje koliko stanica je ugroženo? O analizi izvornog teksta programa za vrijeme incidenta se i ne razmišlja (jer reakcija mora biti hitra), a metode dinamičke zaštite – ako nisu uvedene ranije, opet zahtijevaju prethodnu analizu – samo isključivanje pa uključivanje sustava ponekad je problematično, pa to još više vrijedi za postupak koji uključuje ponovno prevođenje programa, ako to uopće dolazi u obzir. U slučaju korištenja zatvorene programske podrške, uvid u izvorni tekst nije niti moguć pa ni ponovno prevođenje nije opcija. Može se uočiti da je postizanje sigurnosti izuzetno složen proces koji je u naravi jednako organizacijski, koliko i tehnički.

Primjer – SQL umetanje

Do ove ranjivosti dolazi na sustavima koji imaju slabu ili nikakvu kontrolu podataka koji se koriste u pozadinskim SQL upitima. Ovo je tipičan problem u Web aplikacijama, koje općenito glavninu ranjivosti nude kroz obradu korisničkih podataka ili zahtjeva.

Cilj napada je izmijeniti SQL upit kroz nepredviđenu konstrukciju ulaznih podataka. Uzmemo li kao primjer formu za prijavu korisnika, koja prima korisničko ime i lozinku, pozadinski SQL upit za autentifikaciju korisnika bi, konceptualno, u izvornom tekstu forme mogao izgledati ovako:

```
SELECT status_aktivnosti_korisnika
FROM tabela_auth
WHERE korisničko_ime = '$ime' AND lozinka = '$lozinka'
```

... gdje **`$ime`** i **`$lozinka`** predstavljaju varijable koje sadrže tekst pročitan iz ulaznih polja forme. Dohvaćeni `"status_aktivnosti_korisnika"` u nastavku izvornog teksta sadrži rezultat autentifikacije korisnika Ako se nad tim varijablama ne provodi nikakva (ili se provodi nedovoljno detaljna) provjera sadržaja, tada je ovakav sustav autentifikacije trivijalno zaobići i potpuno je nesiguran. Sadržaj tih varijabli može biti takav da u potpunosti promijeni smisao SQL upita, i to na više načina. Primjerice, unese li korisnik kao korisničko ime tekst:

```
eve'; DROP TABLE tabela_auth; --
```

... tada upit izgleda ovako:

```
SELECT status_aktivnosti_korisnika
FROM tabela_auth
WHERE korisničko_ime = 'eve'; DROP TABLE tabela_auth; --'
AND lozinka = '$lozinka'
```

Sadržaj varijable **$ime** uključuje jednostruki navodnik nakon riječi "eve" i time zatvara taj SQL upit. Znak ";" (točka-zarez) označava graničnik između dvije naredbe (kod mnogih implementacija SQL-a) i nakon toga počinje sljedeća naredba, u ovom slučaju razorna za tabelu u pitanju. Nakon brisanja tabele, postavljeni su znakovi koji započinju linijski komentar u upitu i time je zapravo zanemaren ostatak linije pa su upiti formalno ispravni i SQL parser će ih prihvatiti.

Kako bi i ovakav pojednostavljeni primjer proradio, napadač mora, primjerice, saznati ime tabele, no i to ponekad nije previše teško. Do mnogih informacija može se doći i slanjem loše formatiranih SQL upita (npr. vrste i verzije SQL poslužitelja), ugniježđenih upita prema nepostojećim atributima tabele ili jednostavnim nagađanjem. Osim toga, web poslužitelj morao bi imati ovlasti za brisanje tabela, što najčešće nije slučaj, no jednako opasan učinak moglo bi imati i, primjerice, ažuriranje tabela. Tehnički detalji realističnijih napada najviše ovise o vrsti SQL poslužitelja, odnosno napadači koriste posebnosti pojedinih poslužitelja kako bi realizirali napad. Može se pritom raditi o poznavanju predinstaliranih pohranjenih procedura (eng. *stored procedure*), sistemskih tabela ili bilo kojem detalju koji omogućava napad.

I protiv ove ranjivosti ima više načina obrane. Općenito prihvaćeni stav je da niti jedna tehnika sama po sebi ne garantira sigurnost i da ih svakako treba kombinirati. Prva tehnika koja se primjenjuje je obrada ulaznih podataka, tako da ne sadrže specijalne znakove, poput jednostrukog ili dvostrukog navodnika, povlake ("-"), dvotočke itd. Prve obrane temeljile su se na uklanjanju ovih znakova iz ulaza, ali ta je tehnika ograničeno učinkovita. Nadalje, pokazalo se efikasno provjeravati odgovara li očekivani ulaz nekoj zahtijevanoj formi – poput adrese elektroničke pošte (čija je struktura definirana), korisničkog imena (koje smije sadržavati samo određen skup znakova) itd., za što su pogodni regularni izrazi. Razumno je dopustiti samo dobro oblikovani ulaz. Treće na što treba obratiti pozornost je ispravna konfiguracija ovlasti web poslužitelja. Ovisno o ovlastima, poslužitelj može obavljati razne operacije nad bazom podataka i datotečnim sustavom i potrebno je konfigurirati ovlasti tako da one budu minimalne, a dovoljne za obavljanje poslužiteljskog zadatka. Na kraju, isplati se izbrisati/onemogućiti sve predinstalirane pohranjene procedure koje nisu potrebne aplikaciji. One, primjerice, mogu omogućiti napadaču poziv programa iz ljuske, dajući mu efektivno pristup operacijskom sustavu s ovlastima koje ima poslužitelj i pozivima programa koje nudi OS. S takvim pristupom napadač može učiniti veliku štetu na tom sustavu ili, pomoću njega, trećem sustavu.

Vidljivo je kako je i u ovom primjeru misao o sigurnosti potrebna već u fazi planiranja i razvoja aplikacije te postavljanja web poslužitelja. Sigurnost se opet postiže opreznim programiranjem i kvalitetnim dizajnom. No, ostaje problem postojećih aplikacija, kao i novootkrivenih tehnika napada. U stvarnoj situaciji, postojeće aplikacije je teško mijenjati jer to predstavlja težak (skup) organizacijski zadatak – od aplikacije koja je jednom puštena u produkciju očekuje se da radi besprijekorno i do dozvole za modifikaciju aplikacije je ponekad teško doći. Za takvu odluku, potrebno je dobro odvagnuti čimbenike poput stvarne

opasnosti, cijene intervencije u izvorni tekst te cijene i trajanja popravka potencijalne štete – a takva se odluka ne može donijeti bez kvalitetnih informacija, odnosno bez procjene stvarne ranjivosti.

Propusti u implementaciji aplikacije ili protokola, kao što je izloženo, dio obrane nalaze u kvalitetnoj konfiguraciji okoline u kojoj se koriste i upravo o konfiguraciji govori sljedeće poglavlje.

2.1.3 Propusti u konfiguraciji

Popularan zahtjev nad današnjom programskom (i sklopovskom) podrškom je fleksibilnost. Dobar primjer su web poslužitelji. U početku, njihov je zadatak bio jednostavno posluživanje klijenata statičkim HTML stranicama, što je neusporedivo s današnjim zahtjevima. Danas web poslužitelji poslužuju kompleksne, dinamičke stranice, imaju funkcionalnosti poslužitelja datoteka, nude kompleksno skriptiranje na strani poslužitelja, a sve to (i više) ponekad moraju nuditi za više različitih vrsta i verzija klijentskih aplikacija. Dodatno, za ostvarivanje dinamičkih stranica postoje mnoge tehnologije pa je implementacija (i sigurno održavanje) web poslužitelja utoliko kompliciranija, jer web poslužitelj mora znati raditi s njima i mora nuditi mogućnosti za njihovu specifičnu konfiguraciju. Uzevši u obzir da web poslužitelji s jedne strane imaju cijeli Internet kojem su na raspolaganju (za dobre i zlonamjerne korisnike jednako), a s druge pristup privatnom sustavu čije su vlasništvo, jasno je da su često meta svakakvih vrsta napada jer napadaču nude potencijalna vrata u ciljni sustav.

Propusti u konfiguraciji sustava danas čine neiscrpan izvor sigurnosnih problema. Konfiguracijom se može posve kvalitetno programiran softver u trenu pretvoriti u najslabiju kariku i uzrok proboja u mrežu [10]. S druge strane, kao što je navedeno ranije, konfiguracija sustava može predstavljati i obranu od nekih programskih ranjivosti, ili brzo rješenje nekog gorućeg problema – ako je izvedena kvalitetno. Nekvalitetna konfiguracija se, zbog prirode današnjih napadača, često lako uoči i tada čini poziv na napad sustava koji je napadaču možda do te spoznaje bio posve nezanimljiv.

Dio problema leži upravo u preciziranju što bi značila kvalitetna konfiguracija. Konfiguracija sustava je, gledajući izvore ranjivosti, na pola puta između tehničkog posla i niza organizacijskih, poslovnih odluka pa je zato nemoguće imati jasne upute za općenitu "kvalitetnu konfiguraciju". Mada za svaku vrstu naprave i za svu konfigurabilnu programsku podršku postoje naputci i uobičajene najbolje prakse, konfiguracija je uvijek ovisna o namjeni sustava i uvijek je na kraju riječ o svojevrsnoj ravnoteži između iskoristivosti i sigurnosti. Naime, iz perspektive sigurnosti, konfiguriranje sustava je zapravo niz odgovora na pitanje "Dozvoliti ovu mogućnost ili ne?" i upravo o potrebi za nekim mogućnostima ovisi koliko će konfiguracija biti sigurna. Najveći dio problema nije u dopuštanju potrebnih stavki, niti u zabrani opasnih – često postoji niz pitanja na koja nema jasnog odgovora jer neke mogućnosti možda olakšavaju rad korisnicima, a nisu neophodne. No, ako nisu opasne, vrijedi li ih zabraniti? Također, sigurno konfiguriranje sustava u stvarnoj okolini s vremenom bez iznimke naiđe na problem iznimki – uvijek se postavi pitanje vrijede li pravila jednako za sve i mora li administrator nekome dozvoliti neku opciju, koju općenito sigurnosne postavke brane.

Konfiguracija mrežnog sustava odnosi se na sljedeće dijelove:

- konfiguracija pojedinih stanica i poslužitelja;
- konfiguracija mrežne opreme;
- konfiguracija programske podrške.

Za sigurnu konfiguraciju stanica postoje brojni naputci i njezina sigurnost (i strogoća) ovisi o politici organizacije. Za poslužitelje također postoje naputci, ali oni su primjenjivi samo onoliko koliko ne ometaju poslužitelj u obavljanju posla. Ti su naputci često dio, odnosno temelj sigurnosne politike (o kojima će nešto više biti riječi kasnije) i radi se o dokumentima koji detaljno pokrivaju brojne mogućnosti koje nudi operacijski sustav te daju naputke kako koji dio konfigurirati.

Primjerice, uzmemo li u obzir stanicu koja ima Windows XP operacijski sustav, ranjivosti se u mrežu ponekad unesu lako. Ako uobičajeni korisnik ima administratorske ovlasti i ima slobodu instalacije programa, što je izuzetno česta situacija, sustav će s vremenom gotovo sigurno postati ranjiv na mnogo načina, čak i uz razmjerno opreznog korisnika. Ovo će biti primjerima potkrijepljeno u kasnijim poglavljima. Nadalje, uzmimo kao primjer poslužitelj elektroničke pošte (protokol SMTP, recimo). Takvi poslužitelji mogu se konfigurirati da prosljeđuju poruke elektroničke pošte neovisno o domeni primatelja i bez konfiguracije korisnika – čime se na Internet zapravo postavlja poslužitelj za neželjene poruke (eng. *spam e-mail*). Ovo su samo dva primjera, no kad se razmišlja o konfiguraciji sigurnog sustava, razmatraju se raznovrsna pitanja poput politike lozinki, automatskog čitanja sadržaja CD ili DVD medija, dijeljenih tvrdih diskova pa sve do zabrane desnog klika mišem na radnu površinu. Dobar primjer može biti naputak o sigurnoj konfiguraciji računala s Windows XP operacijskim sustavom, ovisno o namjeni računala, kako ga u suradnji sa Microsoftom propisuje američka agencija za nacionalnu sigurnost (NSA – *National Security Agency*) [9].

Konfiguracija mrežne opreme, i mreže općenito, može u velikoj mjeri doprinjeti sigurnosti sustava. Postavke današnjih sklopovskih i programskih sigurnosnih zaštitnih stijena (eng. *firewall*), usmjernika i ostale opreme, danas mogu biti toliko stroge i profinjene, da sakriju (i time donekle ublaže) opasnosti uvedene programskom podrškom. S druge strane, vrijedi i obratno – loša konfiguracija opet uvodi probleme. Kod mrežne opreme mora se pripaziti na opterećenje pojedinih naprava, pogotovo ako neka od njih predstavlja neizostavni dio mreže, jer u protivnom napadač može ugroziti cijeli sustav napadom na jednu točku (eng. *single point of failure*).

Tipičan problem nesigurne konfiguracije mreže su bežične pristupne točke (eng. *Wireless Access Point, WAP*). Čest propust je ostavljanje pristupne točke nezaštićene, odnosno zanemarivanje sigurnosti kod priključivanja i izostanak enkripcije podataka. Uz redovito ispunjeni uvjet da je bežična mreža spojena s glavnom mrežom organizacije (preko virtualne privatne mreže ili izravno), ovakva ranjivost može napadaču osigurati pristup mreži organizacije, neovisno o ostatku sigurnosne politike ili inače kvalitetnoj obrani od napada (Slika 2.2.).

Slika 2.2. Pristup kroz nezaštićenu točku pristupa

Slika 2.2. mogla bi sugerirati da je zaštita pomoću sigurnosne zaštitne stijene neprobojna, što, dakako, nije slučaj. I takve stanice na mreži imaju svoje dobre i loše strane te ranjivosti specifične za određene verzije proizvoda, ali u općenitom slučaju predstavljaju povećanje sigurnosti. Njihova korisnost ponajviše ovisi upravo o kvaliteti konfiguracije, odnosno o vrsti prometa kakvu sigurnosna zaštitna stijena propušta u i iz mrežnog sustava.

Ranjivosti izazvane lošom konfiguracijom ostaju ranjivosti – bila ta konfiguracija opravdano ili neopravdano kakva jest i ponekad se na određenu ranjivost i pristaje, ovisno o parametrima procesa upravljanja rizikom. No, kao i za ostale izvore ranjivosti, važno je i ovaj imati pod kontrolom, odnosno poznavati koliko prijetnju nam predstavlja trenutno stanje postavki na sustavu. Zaštita od loše konfiguracije manje je jasna od npr. zaštite od preljeva spremnika na stogu – prvenstveno zato jer se radi o procesu. Osnovna karakteristika svake produkcijske mreže je njena dinamičnost, kako u pogledu vlastite konfiguracije, tako i u pogledu svake pojedine promjene na svakoj stanici pa se može reći da se konfiguracija mijenja u nepredvidivim trenucima u životu jedne mreže. Iz tog razloga je i utvrđivanje sigurnosnog stanja mreže proces koji traje stalno ili se redovito ponavlja. Dobar početak je imati jasne upute o konfiguraciji svakog servisa koji će biti dostupan na mreži – od elektroničke pošte, do vlastitih aplikacija, specifičnih za mrežu organizacije. Također, sve promjene u konfiguraciji moraju se dokumentirati, kako bi se normalan rad mogao brzo rekonstruirati u slučaju prekida, što nam je važno za implementaciju procesa procjene ranjivosti (koja povremeno može izazvati ili zahtijevati prekide rada sustava). Ovime bi se trebale izbjeći glavne ranjivosti uzrokovane lošom konfiguracijom. No, kao i ranjivosti izazvane propustima u implementaciji, dobra priprema nije dovoljna obrana i potrebno je redovito ispitivati vlastiti sustav kako bi se osigurala što veća razina sigurnosti. Podsjetimo, ravnoteža između dopuštenog i sigurnog, odnosno između pravila i lakoće korištenja se lagano naruši i s vremenom svaka mreža teži olakšavanju sigurnosnih pravila.

2.1.4 Namjena, način rada i skriveni dijelovi aplikacija

Zadnji izvor ranjivosti izazvanih programskom podrškom čine sami programi koji se mogu pojaviti na nekoj mreži, i to korišteni kako je i zamišljeno. Često izvor ranjivosti nije pogreška u izvedbi programa, već program sam po sebi. Programe koji u mrežni sustav unose ranjivosti možemo ovako podijeliti:

- zlonamjerni programi
    - virusi
    - crvi
    - trojanski konji
    - špijunski programi
- neželjeni programi
- željeni programi

Zlonamjerni programi, poglavito virusi, povijesno spadaju u ranije oblike ranjivosti. Virusi, prvi zlonamjerni programi, umetali su svoj zlonamjerni izvršni tekst u druge, legitimne programe i svojevremeno su činili pravu pošast na računalima. Njihovo širenje bilo je vrlo efikasno još i prije pojave Interneta, dobrim dijelom kao rezultat povjerenja korisnika. Virusi su uzrokovali i porast u sigurnosnoj industriji, jer su svojom razornošću potaknuli brojna ulaganja u računalnu sigurnost. Radi se, dakle, o zlonamjernom izvršnom tekstu koji je naseljen u legitimni izvršni tekst nekog programa. Za njegovo pokretanje je odgovoran korisnik, odnosno za njegovo širenje potrebni su korisnik i legitimni program-domaćin. Nakon pokretanja, virus može zaraziti druge programe i obavljati bilo kakve druge operacije nad datotečnim sustavom, uključivo s brisanjem datoteka.

Obrana od virusa većinom se svodi na antivirusnu programsku podršku. Antivirusni programi rade na dva osnovna principa - prepoznavanja potpisa virusa i prepoznavanja ponašanja virusa. Prepoznavanje potpisa se svodi na usporedbu pronađenog izvršnog teksta s bazom poznatog virusnog izvršnog teksta i reakcijom na pozivan rezultat usporedbe. Dakako, efikasnost ovog pristupa ovisi o ažurnosti baze s kojom antivirusni program radi. Dodatan problem stvaraju polimorfni virusi, odnosno virusi koji mijenjaju svoj binarni potpis sa svakom replikacijom. Ovo se, primjerice, može postići enkripcijom dijela izvršnog teksta. Tada samo manji dio izvršnog teksta ostaje nekriptiran (inače ne bi bilo izvedivog kôda) i taj služi za pokretanje kriptiranog dijela. Opet, antivirusni programi tada moraju ciljati nekriptirani dio. Prepoznavanje ponašanja, s druge strane, pristupa problemu tako da promatra sve programe i reagira na sumnjivo ponašanje, poput pisanja po drugim izvedivim datotekama, ili naprosto korištenje drugih programa. Kako i ovo može biti sasvim legitimna radnja (kao i svaka druga heuristika koju koriste takvi antivirusi), korisnika se mora upozoriti i pitati treba li dozvoliti potencijalno opasnu radnju, čime s vremenom opada osjetljivost korisnika na ovakve prijetnje i može se dogoditi i da korisnik greškom dozvoli djelovanje virusa. S organizacijske strane, obrana od virusa može se potpomoći politikom rastavljanja kritičnih podataka od ranjivih segmenata mreže. S korisničke strane, ispad nekog servisa je puno manja opasnost s zanemarivim posljedicama u odnosu na gubitak osobnih ili korporativnih podataka.

Crvi su u mnogo toga nalik virusima i često se miješaju. Radi se o novijoj ranjivosti u odnosu na viruse, a osnovna razlika im je u načinu širenja – crvu ne treba program domaćin i korisnik ne mora imati upliva na njegovo širenje, tj. korisnik ga ne mora uzrokovati. Crv je, dakle, samoreplicirajući samostalni program koji ima sposobnost samostalno se širiti kroz mrežu. Dodatno, crvi *imaju* mogućnost skrivanja u drugim datotekama. Radi tolike samostalnosti, očito je da se radi o ozbiljnijoj prijetnji od virusa i, doista, crvi su sa širenjem Interneta zauzeli glavno mjesto u svijetu zloćudnih programa.

Opasnosti koje uvode crvi su dosta ozbiljne. Osim iste razorne sposobnosti koje imaju virusi, crvi često ne mijenjaju računalo na kojem se pojave na niti jedan očiti način i ponekad je bilo teško razaznati zaraženo računalo. Primjerice, uobičajeni učinak zaraze crvom je otvaranje "stražnjih vrata" (eng. *backdoor*) na zaraženom računalu, čime ono predaje određenu razinu kontrole onome tko je poslao crva. Takva situacija može rezultirati s više vrsta napada (o kojima će više biti rečeno kasnije).

Za obranu od crva preporuča se slična programska podrška kao i za viruse, odnosno antivirusi su danas jednako programi za obranu od virusa, kao i od crva. Dodatno, preporuča se uporaba sigurnosnih zaštitnih stijena, koje mogu ukazati na postojanje crva ili čak spriječiti njegovo daljnje širenje. Također, crvi za svoje širenje često koriste propuste u programskoj podršci i redovito pregledavanje i krpanje korištenih programa je dio zaštite.

Trojanski konj je izraz koji se odnosi na program koji naoko nudi neku poželjnu funkcionalnost, dok istovremeno skriva zlonamjerni izvršni tekst. Ta skrivena funkcionalnost može primjerice skupljati adrese elektroničke pošte s napadnutog računala (ili druge podatke), instalirati druge viruse, ometati rad ili omogućavati udaljeni pristup napadaču. Klasa prijetnji koje uvode trojanski konji slična je virusima i crvima, a isto vrijedi za obranu. Treba napomenuti i da je trojanski konj često implementiran u poslužitelj-klijent arhitekturi, gdje se poslužiteljska strana nalazi na napadnutom računalu i nudi usluge klijentu – napadaču.

Špijunski program (eng. *spyware*) je vrsta programa koja špijunira korisnika. Pritom podaci koji su cilj špijuniranja variraju od sociooloških i privatnih informacija, iskoristivih u željenom i neželjenom marketingu, do kritičnih tajnih informacija, poput brojeva kreditnih kartica. Osim toga, takvi programi mogu pratiti što korisnik piše npr. u elektroničkim porukama ili što korisnik ima u osobnim datotekama. Za obranu od ovakvih programa koriste se opet isti (ili slični, *anti-spyware*) programi kao za viruse, crve i trojanske konje.

Neželjeni programi su legitimni programi koji na neki način mogu ugroziti sigurnost mreže. U poslovnoj okolini postoji niz programa čije se korištenje ne potiče, odnosno čije se korištenje i brani. Ovisno o razini tehničke zaštite i o odabranoj strogoći sigurnosne politike, te zabrane su ponekad propisane usmeno, ponekad pismeno, a ponekad korisnici nemaju mogućnost samostalne instalacije programa. No, potonje je najozbiljniji i najrjeđi slučaj – najčešće korisnici imaju veće ovlasti i, zapravo, ne mare mnogo za nepisana pravila.

Kao i kod konfiguracije mreže, dosta je široka "siva zona" programa, za koje je teško odrediti trebaju li biti zabranjeni ili ne. Dobar primjer je Winamp, najpopularniji program za reprodukciju glazbe. Diskusija o tome treba li zabraniti Winamp ne može biti isključivo tehnička i doista ovisi o okolini. Do sasvim drukčijih zaključaka se dolazi ako je riječ o studentskom računalnom laboratoriju, knjižnici ili uredu u kojem rade, npr., razvojni inženjeri. Često je slična situacija s programima za mrežno druženje i razgovor, poput Skypea ili ICQ-a, s tim da je ponekad politika organizacije dozvolila uporabu jednog, a zabranila uporabu drugog, upravo na temelju povijesti ranjivosti i sigurnosnih incidenata.

Primjer – Hamachi i eMule

Hamachi je besplatni programski paket koji omogućava virtualno privatno umrežavanje preko Interneta, praktički bez napora oko konfiguracije. Računala koja ga koriste mogu biti iza sigurnosnih zaštitnih stijena i usmjernika koji koriste translaciju mrežnih adresa, što Hamachi čini idealnim programom za izbjegavanje sigurnosne politike organizacije, ako takva postoji.

eMule je također besplatni programski paket iz grupe *peer to peer* programa za dijeljenje datoteka. Korisnici eMulea mogu dijeliti svoje datoteke preko Interneta, a sučelje nudi raspodijeljeno povlačenje datoteka s više izvora, pretraživanje itd. Glavni nedostatak je što ponekad za dohvat neke velike ili rijetke datoteke treba jako puno vremena, a pritom se troši

mrežni pristup koji se, primjerice, naplaćuje po potrošenom prometu. No, ako korisnik instalira eMule na računalo u organizaciji u kojoj je zaposlen, onda potrošnju resursa ne plaća on, već organizacija. Problem dohvata datoteka kada su one u potpunosti preuzete može se riješiti pomoću programa Hamachi.

Osim krađe resursa, najveći problem je legalnost preuzetih datoteka. Radi li se o zaštićenim materijalima, poput glazbe, filmova, knjiga ili programa, organizacija može biti djelomično odgovorna za kršenje važećih zakona.

Mada niti jedan od ovih programa nije protuzakonit sam po sebi, njihove se funkcionalnosti mogu zlouporabiti i dobar su primjer programa koji bi bili zabranjeni sigurnosnom politikom organizacije. I opet se postavlja pitanje kako osigurati da korisnici nemaju instaliran neki program? Kako će se pokazati kasnije, procjena ranjivosti i za ovo ima rješenje.

I na kraju, uobičajena je situacija da se negdje unutar mreže iz nekog razloga mora koristiti program koji ima poznate ranjivosti. Uobičajeni primjeri su interne aplikacije koje organizacija koristi kao podršku redovnom poslovanju – aplikacije za knjigovodstvo, fakturiranje, praćenje rada i slično. Takve se aplikacije često ne održavaju, odnosno budu napisane, krenu u uporabu i na njihovu sigurnost i održavanje se dalje ne misli. S vremenom se u njima mogu i ne moraju pronaći greške, ili se može ispostaviti da su ranjive na neku novu prijetnju. Također, često se radi o programima pisanim u jeziku Java, a koji nisu kompatibilni s verzijama Jave nakon one za koju su originalno pisani. Kako je program važan, pristaje se na poznatu ranjivost, Java se na tom poslužitelju ne ažurira i za osiguranje cijelog sustava se mora pronaći netehničko rješenje.

Od ovakvih programa se, dakle, ne branimo, već njih moramo dodatno *zaštititi*. To se prvenstveno radi kontrolom pristupa, ako je to moguće. S druge strane, ako je pristup aplikaciji s Interneta neophodan, tada moramo biti svjesni da će aplikacija sigurno biti napadnuta i potrebno je ograničiti ugroženo područje, kao i osigurati sve što je s navedenom aplikacijom u dodiru (poput baza podataka). Također, ovo mora biti privremeno rješenje i ponovno pisanje aplikacije (ili dorada) mora se istaknuti kao zadatak visokog prioriteta.

Ovime su opisane sve vrste ranjivosti izazvane programskom podrškom. Može se zaključiti da vrsta ranjivosti ima dosta, da su raznolike i treba stalno imati na umu da su vrlo brojne. Trenutno nema razloga očekivati popravak ovog stanja – propusti u dizajnu, kao i propusti u izvedbi i dalje pate od potrebe za ubrzanim razvojem. Dapače, kako vrijeme ide, rokovi su sve kraći i ubrzani razvoj (eng. *rapid development*) postaje činjenica o kojoj se vodi računa već u dizajnu novih programskih jezika i integriranih razvojnih okolina. Što se propusta u konfiguraciji tiče, ovdje se mora naglasiti poboljšanje. Sigurnost više nije zanemarena i pri postavljanju sustava postoji već prirodno nastojanje da se sustav postavi na siguran način. U tipičnom slučaju, propusti u konfiguraciji često su rezultat informirane odluke i prihvaćanja rizika [12] nego neznanja i takve konfiguracije se prate s većom pozornošću. Na području aplikacija, naprotiv, sigurnosni trendovi su negativni. Crva, virusa, trojanskih konja i špijunskog softvera sve je više i postaje sve "pametniji" u zavaravanju današnjih antivirusnih programa.

Svemu navedenom treba, nažalost, dodati još jednu, sasvim drugačiju grupu ranjivosti, a to su one uzrokovane korisnicima. U kombinaciji s programskim ranjivostima, ranjivosti uzrokovane korisnicima čine plodno tlo za napadače i za konačnu realizaciju svih prijetnji, a to su napadi na sustave.

2.2 Ranjivosti uzrokovane korisnicima

Korisnici u svijet ranjivosti unose još malo "ljudskog faktora", dajući napadima dodatnu raznovrsnost i dajući napadačima brojne nove vektore napada. Zbog određenih vrsta ponašanja korisnika, napadi dobivaju novu, netehničku dimenziju. Naime, cilj napadača ponekad se ne mora ostvariti isključivo iskorištavanjem tehničkih nedostataka, kako je opisano u prethodnom poglavlju. Dapače, često je realizacija napada puno lakša zaobilaznim putem – prijevarom, predviđanjem ponašanja korisnika ili iskorištavanjem nekih "otvorenih vrata" koja korisnici neoprezno ostavljaju iza sebe. Napadačev primarni cilj postaje skupljanje tajnih informacija od korisnika, i to na način da te informacije preda sam korisnik. Drugu vrstu opasnosti unose neoprezni ili nezadovoljni, zlonamjerni korisnici sustava.

Ova se vrsta ranjivosti može podijeliti na sljedeći način:

- nezadovoljni korisnici/zaposlenici;
- neoprezni korisnici;
- socijalni inžinjering.

2.2.1 Nezadovoljni korisnici/zaposlenici

Nezadovoljni korisnici već imaju pristup sustavu i njihova je pozicija zato vrlo opasna. Ovisno o sigurnosnoj konfiguraciji sustava, ulozi i postojećim ovlastima korisnika, oni mogu predstavljati veću ili manju prijetnju sustavu. Korisnik možda posjeduje pristup tajnim informacijama koje bi poželio otkriti trećim osobama, ili ima znanje i mogućnosti onesposobiti važan dio sustava. Važno je znati da se takav korisnik nalazi u situaciji koja je prvi veliki cilj napadača – ima pristup sustavu. Daljnje napredovanje prema većim ovlastima i, time, većoj razornoj moći, ovisi o znanju i upornosti tog napadača.

Zaštita se temelji na kontroli pristupa podacima i podsustavima na osnovi potreba posla. Svaki korisnik trebao bi imati ograničena prava i područje pristupa i na taj se način ova vrsta korisničke ranjivosti može kontrolirati – teško se može spriječiti.

2.2.2 Neoprezni korisnici

Neoprezni korisnici čine veliku štetu u kombinaciji s konfiguracijskim propustima i korištenjem pojedinih aplikacija (vidi opis u poglavljima 2.1.3 i 2.1.4). Primjeri aktivnosti ili navika neopreznih korisnika su:

- korištenje iste lozinke na mnogo mjesta i izostanak periodičke promjene lozinke – što je i konfiguracijski propust; zatim loše lozinke, predvidljive lozinke, dijeljenje lozinke s kolegama;
- instalacija privatnih bežičnih pristupnih točaka unutar mreže organizacije;
- ostavljanje radne stanice otključane kod odlaska s radnog mjesta;
- isprobavanje nepouzdanih programa s Interneta, protuzakonito korištenje programa;

... i ovaj popis se može nastaviti još dugo. Osim toga, neoprezni korisnici ranjivi su na metode socijalnog inžinjeringa.

Obrana od ranjivosti koje donose neoprezni korisnici je samo dijelom u edukaciji. Potrebno je inzistirati na jasnim uputama za korištenje službene opreme, redovito pregledavati vlastiti

sustav i periodički ga ispitivati na ranjivosti iz ovog područja. Drugim riječima, potrebna su pravila u kombinaciji s redovitim provjerama koliko se korisnici drže tih pravila.

2.2.3 Socijalni inžinjering

Socijalni inžinjering podrazumijeva vještinu manipuliranja ljudima s ciljem otkrivanja povjerljivih informacija. Metode i tehnike realizacije se dosta razlikuju, ali temelj im je u uspostavi odnosa povjerenja između napadača i žrtve. Napadačev cilj može, primjerice, biti nabavljanje broja kreditne kartice ili neke lozinke i pritom se koristi pažljivo konstruiranim lažima kako bi došao do željenih informacija. Pažljiva konstrukcija laži u ovom slučaju podrazumijeva pripremno proučavanje žrtve, okoline u kojoj ona radi i skupljanje svih mogućih informacija kako bi napadač i njegova laž djelovali autentično.

U implementaciji, ovi napadi se često izvršavaju telefonom ili elektroničkom poštom, a rjeđe osobno. Napadač se u pravilu predstavlja kao netko drugi i u kontaktu sa žrtvom, kritičan trenutak predstavlja zahtijevanje nekih osobnih podataka – u najgorem slučaju, lozinke ili broja kreditne kartice.

Primjer - *phishing*

Među poznatije vrste socijalnog inžinjeringa spada takozvani *phishing*. Naziv dolazi od iskrivljenog pisanja engleske riječi "*fishing*", koja u prijevodu znači pecanje i dobro opisuje koncept napada.

Primjerice, napadač pošalje *e-mail* poruku na velik broj adresa u kojoj se predstavlja kao neka legitimna prodajna organizacija i u toj poruci traži od korisnika da potvrdi broj svoje kreditne kartice na predviđenoj web stranici, na koju ponudi poveznicu (eng. *link*) – u protivnom će se korisnički račun izbrisati. Web stranica o kojoj je riječ može izuzetno dobro imitirati izgled originalne web stranice odabrane organizacije, no zapravo se radi o napadačevoj zamci i svi ostavljeni podaci zapravo su predani napadaču. U određenom postotku će primatelji takve poruke doista biti korisnici te organizacije, a neki od njih neće uočiti prijevaru i doći će do otkrivanja vlastitih podataka.

Sličan scenarij može lako ugroziti korporativni intranet. Odlučnom napadaču dovoljno je par dana intenzivnog proučavanja rada organizacije da se sakupi velika količina informacija i njih iskoristi kao mamac za ciljano osoblje.

Primjer – igranje scenarija

Napadači, ako je to moguće, često posjete fizičku lokaciju organizacije koja im je meta. Na taj način znanja sakupljena s Interneta i eventualno iz medija mogu nadopuniti novim informacijama. Tada puno lakše konstruiraju priču, scenarij koji će odigrati pred ciljanom žrtvom iz organizacije, a koji može biti vrlo uvjerljiv.

Posjet nekom uredu može otkriti puno detalja koji su naoko nevažni, ali pridodaju autentičnosti pristupa napadača. Primjerice, napadač može uočiti mrežni pisač neke marke i proizvođača te kasnije nazvati nekoga iz administrativnog osoblja, i to u ulozi predstavnika tog proizvođača. Pozvavši se na neki viši autoritet, napadač možda tražiti korisničko ime i lozinku žrtve, radi izmišljene dogovorene dijagnostike uređaja i eventualno na taj način dobiti određene korisničke ovlasti na ciljnoj mreži.

Napadači u ovom slučaju redovito ciljaju osoblje koje slabije poznaje računala i računalnu sigurnost, odnosno osoblje kojem je računalo isključivo alat, a ne predmet interesa. Takvi korisnici češće imaju predispoziciju shvatiti računalne lozinke olako i nisu svjesni opasnosti, na sličan način kao što ljudi ostavljaju ključeve automobila mehaničaru, ili u autopraonici. Takvo ponašanje nije obavezno rezultat manjka edukacije već uobičajene sklonosti ljudi da

vjeruju *autoritetima* za neku tematiku, bez obzira na rang unutar organizacije. Ovakvo otkrivanje privatnih informacija će se, dakle, prije dogoditi osobi iz višeg menadžmenta, nego administratoru sustava, razvojnom inženjeru ili programeru – i učinit će utoliko veću štetu. Zato se obrana opet temelji na organizacijskim odlukama. Osjetljive informacije, primjerice, moraju se pohranjivati samo u kriptiranom obliku, a ključevi se ne smiju nalaziti nigdje na mreži. Brojevi kreditnih kartica ne bi se smjeli držati na računalima, a lozinke bi se trebale držati strogom tajnom.

2.3 Napadi na mrežne sustave

S prikazom socijalnog inžinjeringa prikazana je i zadnja grupa ranjivosti s kojima se susrećemo na današnjim mrežama računala. Prikazano je kako su ranjivosti danas raznovrsne, kako dolaze iz puno različitih vrsta propusta i neopreznosti i, što je u konačnici najvažnije, prikazano je kako je za obranu od njih potrebno razmišljati multidisciplinarno i napredno, organizacijski i tehnički. Obrana nije isključivo tehnička, kao što to nisu ni napadi, već nadilazi te granice – jer u njima ostaje isključivo teorijska.

Sve opisane ranjivosti, kako je spomenuto u uvodu ovog poglavlja, čine potencijal da se dogodi prava opasnost, a to je uspješan napad na sustav. Mada je vrsta ranjivosti mnogo, one u pravilu vode do neke od svega par vrsta napada. Naime, uspjeli proboj redovito rezultira nekom od sljedećih posljedica:

- krađa, zlouporaba, uništenje podataka ili resursa
- izostanak usluge

2.3.1 Krađa, zlouporaba, uništenje podataka ili resursa

Napadači najčešće, neovisno o motivaciji, za cilj imaju krađu podataka. To mogu biti osobni podaci, poput životnih navika, ali i medicinskih podataka. Nadalje, cilj napadača može biti krađa novca (brojeva kartica), identiteta ili intelektualnog vlasništva. Krađa resursa odnosi se na korištenje resursa organizacije kako bi se napalo neku drugu organizaciju; mada prva organizacija nije krajnji cilj napadača, ona se svejedno može smatrati napadnutom jer je probijena njezina zaštitna politika.

Za ove napade koriste se sve vrste ranjivosti koje napadač nađe, od snimanja otvorenih pristupa, preko iskorištavanja (ne)poznate ranjivosti u pronađenom servisu i uzrokovanja preljeva spremnika na stogu, do socijalnog inžinjeringa i manipulacije korisnika. Postoji izuzetno mnogo načina da se neki sustav koristi pogrešno i omogući neki od ovih ciljeva.

Ova grupa čini najozbiljniju prijetnju današnjoj informacijskoj sigurnosti i bez sumnje je najveći uteg u daljnjem razvoju informatike. Dobar dio organizacija, posebno financijskih institucija, ne mogu prihvatiti postojeće rizike u nekim segmentima poslovanja i zato se manje oslanjaju na tehničke mogućnosti koje nudi Internet. To je posve opravdano, uzmemo li u obzir današnje stanje operacijskih sustava i nove programske podrške. Kada je riječ o ljudskoj imovini i egzistenciji, (pretjerani) oprez je očekivan. U tek nešto boljoj varijanti, uspješan proboj i realizacija krađe ili uništenja podataka rezultira gubitkom kredibiliteta i nepovratno narušenom javnom slikom o stradaloj organizaciji.

2.3.2 Izostanak usluge

Kada je riječ o izostanku usluge, treba imati na umu da usluga (koju nudi mrežni informacijski sustav organizacije) može izostati prema unutra i prema van.

Izostanak usluge prema unutra rezultira usporavanjem rada organizacije. Tek kada informacijski sustav prestane raditi dobro, postaje vidljivo koliko uobičajeno poslovanje o njemu ovisi – a to je ponekad iznenađujuće mnogo. Bez intraneta, interne usluge prosljeđivanja poruka, internih aplikacija i, u zadnje vrijeme radi uvođenja IP telefonije sve češće bez telefona, rad i razvoj u današnjim tvrtkama jednostavno staje. Dodatni problem je period ponovnog uspostavljanja normalnog rada, jer sve aktivne, "žive" mreže danas mogu imati puno posebnosti koje otežavaju ponovno pokretanje.

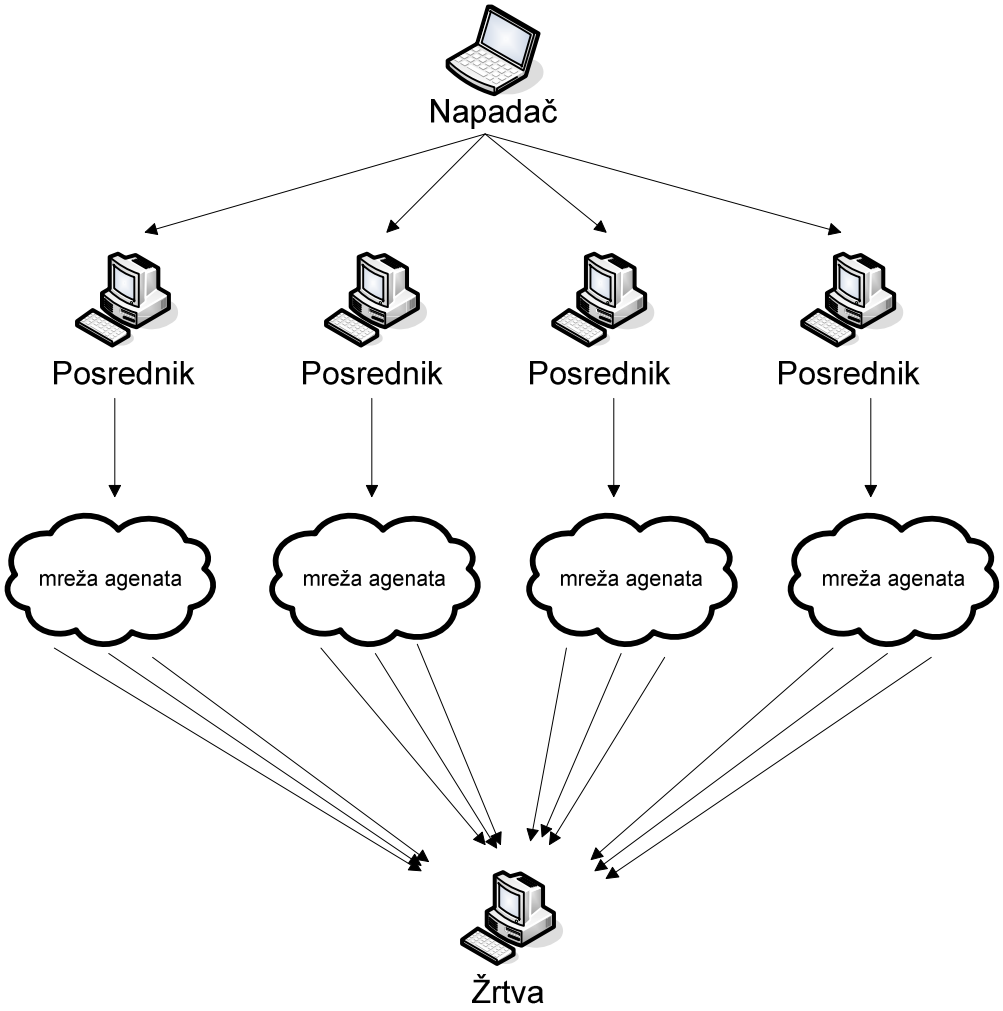
Mnoge organizacije upravo temelje svoje poslovanje na Internetu i izostanak usluge prema van u njihovom slučaju predstavlja izravan gubitak novca. Ti gubici mogu biti ogromni i prvenstveno ovise o trajanju ispada. Također, ovisno o uzrocima i javnoj percepciji napada, moguće je narušavanje imidža organizacije, a koje u konačnici opet predstavlja financijske gubitke.

Napadi uskraćivanjem usluge - *DoS* napadi

DoS napadi predstavljaju najozbiljniju grupu napada na dostupnost neke usluge prema van. Napadi se relativno lako pokreću, izuzetno su efikasni i obrana od njih praktički ne postoji, ili graniči s umjetnošću.

U DoS napadu, napadačev cilj je spriječiti legitimne korisnike u pristupanju nekoj usluzi, odnosno internetskom servisu. Konceptualno, napadač ovo ostvaruje iskorištavanjem uobičajenih karakteristika TCP/IP protokolnog sloga u svrhu masovne potrošnje resursa na strani servisa koji nudi uslugu. Kada su resursi poslužitelja dovoljno zauzeti, korisniku usluga postaje nedostupna, ili izuzetno spora – dovoljno da postane neiskoristiva. I tada je napad uspješan, odnosno usluga je doista uskraćena. Također, isti se efekt može postići bilo kakvom dostupnom metodom rušenja ciljnog servisa, odnosno napad se može izvršiti i na aplikacijskom sloju (ne samo na mrežnom ili transportnom) OSI referentnog modela. U tom slučaju, riječ je o iskorištavanju ranjivosti u ciljnoj aplikaciji i taj slučaj nipošto nije rijedak. Mnogi servisi su se pokazali ranjivi čak na elementarno primanje posebno konstruiranog paketa na odabrane TCP ili UDP pristupe.

Ako intenzivno zauzimanje resursa provodi jedan napadač (a za to postoje razne tehnike, od nepravilnog uspostavljanja lažnih veza koje zagušuju metu do slanja ciljano neispravnih paketa), riječ je o klasičnom DoS napadu. No, ako napadač na raspolaganju ima više računala – koja se i u ovom slučaju nazivaju zombiji, može pokrenuti puno ozbiljniji, distribuirani napad uskraćivanja usluge (eng. *Distributed Denial of Service, DDoS*). Tipično, ovakvu situaciju napadač gradi kroz dulje vrijeme, iskorištavajući neku poznatu ranjivost. Primjerice, iskorištavanjem preljeva na stogu, napadač može otvoriti stražnja vrata i instalirati trojanskog konja na žrtvu. Pomoću svakog zaraženog računala, napadač može lakše zaraziti sljedeće računalo i nakon nekog vremena, može pod svojom kontrolom imati na tisuće zombija. Radi lakše kontrole, neka od zaraženih računala se odrede za posrednike, koji će kontrolirati vlastite grupe računala i na kraju napadač ima na raspolaganju višeslojnu strukturu, prikazanu dijagramom na slici 2.3.



Slika 2.3. Dijagram *DDoS* napada

Napadač pod svojom izravnom kontrolom ima računala posrednike, koji će na njegovu komandu pokrenuti napad na žrtvu. Napad će izvršavati pojedine mreže agenata, koje su pod kontrolom posrednika, a te mreže mogu sadržavati stotine, čak i tisuće zombija.

Glavni problem kod ovakve vrste ranjivosti ukazuje na možda najopasnije svojstvo današnjih napada, a to je njihova automatizacija. Danas su svima dostupni besplatni alati koji lako i automatski skeniraju velike dijelove Interneta u potrazi za ranjivim računalima, što je dovelo do sasvim nove prijetnje, a to su slučajni napadi. Riječ je o napadima do kojih dolazi isključivo zato što se prilika ukazala, odnosno sama mreža možda i ne bi bila ciljana, da na njoj nije uočena neka ranjivost koja ju je učinila zanimljivom napadačima. Primjer prikupljanja zombija dobro ilustrira taj problem, a treba imati na umu da onog trenutka kada napadač pribavi novog zombija – ništa ga ne sprječava da na njemu pogleda i ostale "zanimljivosti". Postojanje automatiziranih alata također je pogodovalo svojevrsnoj deevoluciji zlonamjernih korisnika. U prošlosti, za iskorištavanje ranjivosti bila je potrebna daleko veća razina tehničkog znanja, domišljatosti i inspiracije nego danas, kada te karakteristike kvalitetno mijenjaju programi i algoritmi. No, to ne znači da su oni išta manje opasni, dapače, to samo znači da ih ima više i da je skup motiva za napad proširen poticajnim karakteristikama poput djetinjaste osvete i čistog vandalizma.

Uzevši u obzir brojne načine nastanka ranjivosti, od kojih su neki duboko ukorijenjeni u današnju informacijsku industriju te raznolikost vrsta i načina njihovog iskorištavanja, a sve to u kombinaciji sa sve raširenijom uporabom mreža i sve većim priljevom zlonamjernih korisnika, postalo je jasno kako je potreban sustavan oblik zaštite. Zanemarivanje sigurnosti ne dolazi u obzir, *ad hoc* sigurnost naprosto nije dovoljno pouzdana, a potpuno osiguranje svakog djelića sustava je preskupo i nepraktično. Ono što je u današnjim mrežama neophodno, je teorijski utemeljen, sistematičan pristup osiguravanju mrežnih sustava koji će, osim pregleda trenutnog stanja dati prijedloge za poboljšanje i odgovore na važna pitanja koja tek omogućavaju upravljanje rizikom. Sljedeće potpoglavlje govori o metodama zaštite mreža, čemu koja služi, koje su im prednosti i nedostaci, kako se nadopunjuju i kolika se sigurnost pomoću njih postiže.

2.4 Metode zaštite mrežnih sustava

2.4.1 Elementarni tehnički postupci

Antivirusni programi

U ovu grupu ubrajaju se svi programi protiv virusa, crva, trojanskih konja i programa koje svrstavamo u špijunske. Zbog većeg povjerenja prema autoriziranim korisnicima i ubrzanog protoka dijeljenih informacija putem elektroničke pošte, intraneta ili USB prijenosnih memorijskih uređaja, zloćudni programi se vrlo lako šire unutar organizacijske mreže. Zato se zaštita svake pojedine stanice smatra obaveznom, a kako postoje kvalitetni i besplatni antivirusni programi, njihovo prisustvo na pojedinim stanicama i poslužiteljima već se uzima zdravo za gotovo.

Sigurnosne zaštitne stijene

Osnovnu sigurnost organizacije moraju postići kroz uporabu elementarnih sigurnosnih uređaja na mreži. Postavljanje sigurnosnih zaštitnih stijena (eng. *firewall*) smatra se obavezom barem na svim točkama gdje se organizacija spaja na uvijek nesigurni Internet, a pogodno je imati po jedan paketni filter na svakoj podmreži, kako bi se postigla fleksibilnost i rasteretilo glavne sigurnosne zaštitne stijene. Osim toga, u elementarne postupke može se svrstati i politika sigurnosnih stijena na pojedinim stanicama, značilo to njihovu konfiguraciju i korištenje, ili gašenje.

Osim elementarne funkcije filtriranja prometa prema karakteristikama IP paketa koji kroz njih prolaze, sigurnosne zaštitne stijene evoluirale su u fleksibilne i vrlo moćne komade programske ili sklopovske podrške (postoje obje izvedbe). Njihova funkcionalnost već godinama uključuje usmjeravanje paketa, praćenje stanja TCP veze, modifikaciju i preusmjeravanje prometa te mnoge druge funkcionalnosti, na gotovo svim slojevima OSI modela.

Radi kasnijeg razmatranja, vrijedi odmah uočiti da se radi o potencijalno vrlo opterećenoj opremi. Spomenuta fleksibilnost i svestranost često je uzrok popriličnog naprezanja ove vrste opreme, kao i činjenica da kroz ove uređaje prolazi velika količina prometa.

Sustavi za otkrivanje napada (IDS)

Drugi koristan tehnički postupak je instalacija sustava za otkrivanje napada. Radi se o posebnoj vrsti programske podrške koja snima sav promet na mreži i reagira na sumnjivi promet. Takav promet može biti generiran unutar vlastite mreže i tada se radi o nekoj vrsti proboja iznutra, odnosno može dolaziti izvana, što znači da je napadač (još uvijek) izvan sustava. Osim obavještavanja odgovornih osoba o uočenim sumnjivim aktivnostima, sustavi

za otkrivanje napada mogu obavljati i ponešto aktivnije zadatke, poput ograničenog mijenjanja konfiguracije mreže, kako bi se sustav obranio od napada.

IDS programska podrška radi na principu prepoznavanja određenog prometa kao štetnog. Općenito, najčešće i s najmanje grešaka IDS reagira na skeniranje sustava i DoS napade. Osim toga, relativno dobro se prepoznaju zlonamjerni pokušaji postizanja većih ovlasti udaljenih ili lokalnih korisnika. Ovisno o pravilima rada, IDS može poduzeti aktivne mjere na mreži, poput blokiranja prometa s određenih lokacija ili promjene konfiguracije nekog drugog uređaja na mreži, no, kako otkrivanje nije uvijek pouzdano, to se radi rjeđe, samo za vrste prometa za koje smo sigurni da su nedopušteni. Može se dogoditi i da IDS alarmantnim označi i sasvim legitiman promet pa je aktivnija reakcija od same dojave nešto čemu treba pristupiti s puno opreza. Također, vrijedi napomenuti da IDS proizvodi veliku količinu zapisa jer je u naravi pomalo paranoičan sustav. Kako bi se tako velika količina podataka razbila u manje cjeline, senzori sustava (a to je dio sustava koji prikuplja podatke), postavljaju se na strateški odabrana mjesta, poput svake podmreže te svakog ulaza u cijelu mrežu. Tako razdijeljene podatke lakše je pregledavati i koristiti.

Sigurnosne zaštitne stijene i IDS čine vrlo snažnu prvu liniju obrane, dajući mreži istovremeno određenu razinu prevencije i reakcije na proboj. Sigurnosna zaštitna stijena još daje mogućnost privremenog zatvaranja prometa na lakši i elegantniji način, dok se kao sekundarna pogodnost IDS-a može smatrati velika pomoć njegovih zapisa pri eventualnoj forenzici. No, primarna pozitivna posljedica korištenja IDS-a je svakako sprječavanje eskalacije incidenta, osiguravajući bržu reakciju na iskorištavanje neke ranjivosti.

2.4.2 Složeniji organizacijsko-tehnički postupci i procesi

Postizanje informacijske sigurnosti mreže je nešto teži zadatak. Današnje mreže, osim što su na mnogo načina ranjive – što je opisano ranije – dodatno kompliciraju situaciju svojom heterogenošću. Na njima se uz radne stanice, pisače i poslužitelje danas nalaze IP telefoni, raznovrsna mrežna oprema s konfiguracijskim sučeljima, bežične pristupne točke koje uvode razna ručna računala, pametne mobilne naprave/telefone i slično. Osim toga, na mrežama se i same stanice pale, gase i mijenjaju, postaju nesigurne i proizvode svakakav promet, odnosno mreže se mijenjaju. S vremenom se ukazala potreba za sistematičnim i skalabilnim rješenjima za informacijsku sigurnost mreža, čija je osnova dovoljno generička i prilagodljiva raznim organizacijama, neovisno o veličini i području djelovanja. Kao odgovor na tu potrebu, razvijeni su sljedeći zaštitni procesi:

- procjena ranjivosti sustava;
- penetracijsko ispitivanje;
- cjeloviti pregled informacijske sigurnosti organizacije.

Procjena ranjivosti

Procjena ranjivosti sustava (eng. *Vulnerability Assessment, VA*) spada u prvi i osnovni korak u sustavnom postizanju sigurnosti mrežnog sustava. Spada u preventivne aktivnosti, namijenjene sprječavanju incidenta, a osim povećanja sigurnosti, nudi veliku vrijednost u mrežno-administratorskim poslovima, katalogizaciji resursa i kao pomoć u procesu planiranja i dimenzioniranja mrežne nadogradnje. Kao alat, automatizirana procjena ranjivosti nudi odgovore na mnoge probleme izložene do sada, a kao proces predstavlja složen postupak podijeljen u više faza, koji zahtijeva koordinaciju više ljudi iz više sfera poslovanja, često iz različitih tvrtki. Treće poglavlje ovog rada bavi se raščlanjivanjem procjene ranjivosti na faze

koje uključuju planiranje, izvedbu i analizu rezultata te alatima koji su namijenjeni za te poslove.

Penetracijsko ispitivanje

Penetracijsko ispitivanje (eng. *Penetration Testing, "pen-testing"*) je metoda osiguravanja sustava koja zapravo simulira napad [11]. Svrha ove vrste ispitivanja sustava je pronaći i doista iskoristiti neku ranjivost kako bi se dospjelo na ciljanu mrežu te pritom pokazati kako je ranjivost omogućila napadaču uzrokovanje nekakve štete. Ovisno o dogovoru s vodstvom organizacije koja naručuje ovu uslugu, penetracijsko ispitivanje može se obavljati i bez znanja ostalih zaposlenika, čime kvalitetan test uključuje i različite pokušaje socijalnog inžinjeringa. Ispitivanje se tada obavlja u neodređeno vrijeme i to je najbolji način za provjeru koliko je mreža sigurna u uobičajenoj, svakodnevnoj uporabi. Također, ispitivanje se može obavljati uz različitu razinu poznavanja ciljne mreže. Opet, u dogovoru s naručiocem, izvršitelj ispitivanja može i ne mora imati detaljne podatke o mreži koju "napada". Ovisno o ovoj odluci, sigurnosni stručnjak može i ne mora znati arhitekturu mreže, raspon dostupnih adresa, popis servisa i druge pojedinosti.

Svi čimbenici koji određuju vrstu penetracijskog ispitivanja zapravo služe za simulaciju određene vrste napada. Ako izvršitelj testa nema detaljne podatke o mreži, tada zauzima ulogu hakera i u tom slučaju kvaliteta rezultata najviše ovisi o stručnosti i upornosti izvršitelja ispitivanja, u usporedbi s istim karakteristikama kod hakera. Ovakvo ispitivanje je pogodno ako organizacija dio svoje sigurnosti temelji na skrivanju tehničkih detalja pozadinske mreže i želi vidjeti koliko se može postići s informacijama koje izvršitelj testa može nabaviti iz javnih izvora. S druge strane, za postizanje sigurnosti je često pogodniji test uz poznavanje detalja mreže – tada izvršitelj penetracijskog testa zna gdje se ranjivosti nalaze i može bolje iskoristiti vlastito umijeće u realizaciji napada.

U svim varijantama penetracijskog ispitivanja, procjena ranjivosti čini važan prvi korak. U prvom slučaju, rezultati govore vlasniku mreže gdje se nalaze poznate ranjivosti i kako osigurati sustav, dok u drugom slične informacije daje "napadaču", čime stvarne razina opasnosti dolazi u pitanje. Primjerice, kritična ranjivost u nekom servisu koji nije na Internetu predstavlja mali problem u odnosu na javno dostupno računalo s lošom politikom lozinki (npr. takvom da lozinke ne ističu i da se broj pokušaja prijava ne ograniči). Procjena ranjivosti navest će obje ranjivosti, ali tek kod (pokušaja) iskorištavanja postaje jasno koja ranjivost čini veću prijetnju.

Penetracijsko ispitivanje dijeli mnoge osobine procjene ranjivosti, koja je detaljno opisana kasnije. U oba se slučaja radi o dogovorenim testovima koji mogu biti i pomalo opasni za ciljne sustave i čija priprema čini velik dio ukupnog posla, u što osim tehničke pripreme spadaju i pitanja poput zaštite podataka i opsega ispitivanja te što činiti s rezultatima.

Cjeloviti pregled informacijske sigurnosti organizacije

Cjeloviti pregled informacijske sigurnosti organizacije (eng. *Information Technology Security Audit*) je najširi i najkompleksniji postupak osiguranja kompletne informatičke potpore koju organizacija posjeduje. Točnije, radi se o formalnoj, pisanoj provjeri sigurnosnog stanja i usklađenosti sa sigurnosnom politikom. Ovakav pregled služi za osiguranje provođenja odluka vezanih uz informacijsku sigurnost i služi kao potpora sigurnosnoj politici – koju mora imati svaka velika organizacija. Upravo u kombinaciji sigurnosne politike i redovite provjere usklađenosti s njom se postiže potrebna skalabilnost sigurnosnih rješenja, i tek na ovaj način organizacije postižu zadovoljavajuću razinu sigurnosti mreže.

Ovakav pregled prvenstveno zahtijeva grupiranje svih relevantnih dobara organizacije u resurse. Promatrati isključivo računala kao resurse je pojednostavljenje, odnosno resursima se smatra više toga, primjerice:

- prijenosna i stolna računala,
- poslužitelji,
- pisači, telefoni, ostala mrežna oprema,
- digitalni i analogni fotoaparati s osjetljivim podacima,
- snimke telefonskih razgovora, snimke konferencija,
- intranetske i internetske web stranice, interne aplikacije,
- sigurnosne kamere,
- pristupne pametne kartice zaposlenika, itd.

Za svaku stavku koja se klasificira kao resurs se detaljno i kritički promatraju postupci uporabe, prijenosa ili prava pristupa. Općenito, i prijetnje se (za ovu vrstu testova) moraju klasificirati, kako bi međusobno referenciranje i povezivanje prijetnji s resursima bilo unificirano. Ispitivanje se izvršava koristeći automatiziranu procjenu ranjivosti, "ručnu" inspekciju i analizu te intervjue i promatranje ponašanja korisnika.

Kao i kod procjene ranjivosti, u jednom trenutku se mora pojedinim stavkama pridati određeni prioritet, ovisno o procjeni opasnosti i tada se kreće u implementaciju rješenja svih pronađenih problema, i to u fazama, imajući u vidu te prioritete.

Za sve tri navedene metode mora se naglasiti da su međusobno povezane, da se često isprepliću i da su najjače kada se koriste zajedno. Također, važna im je karakteristika privremenost. Svaki oblik zaštite ima svoj rok trajanja i kako bi se održala neka razina sigurnosti, testove treba redovito ponavljati. Osim stalnog priljeva novih ranjivosti i metoda napada, slabljenju sigurnosti doprinose i korisnici svakodnevnom uporabom, prirodno zanemarivanje pravila koje se pojačava s vremenom i iznimke koje se uvode po potrebi i zaboravljaju. Iz tog razloga, sigurnost mreže mora se shvatiti kao cilj kojem se teži, od kojeg se s vremenom udaljava i kojem se povremeno opet treba približiti. Drugim riječima, sigurnost nije stanje koje se na bilo koji način *postigne*, nego je samo privremenog karaktera, a osiguravanje mrežnog sustava je u naravi višeslojan proces koji mora trajati – a ne posao koji se jednokratno obavi. Od navedenih metoda zaštite, procjena ranjivosti bi se morala obavljati najčešće, penetracijsko ispitivanje rjeđe, a cjelokupni pregled sustava najrjeđe.

Sljedeće poglavlje govori o prvom koraku u zaštiti mrežnih sustava – procjeni ranjivosti. Bit će izložen teorijski pristup i dijelovi koji sačinjavaju taj proces, i bit će izloženi ulazni parametri, kao i rezultat cijelog postupka.

3. Sustavni pristup procjeni ranjivosti

Procjena ranjivosti mrežnog sustava je proces kojem je glavni cilj automatizirano skeniranje mreže u potrazi za poznatim ranjivostima te grupiranje, dodjela prioriteta i katalogizacija rezultata prema skeniranim resursima i drugim proizvoljnim kriterijima. Riječ je o sveobuhvatnom procesu koji bi morao biti primjenjiv na sve vrste i veličine mreža. Kako bi to bilo moguće, izvedba procjene ranjivosti (kao svojevrstan projekt ili posao) podijeljena je u više faza, a svrha svake od njih je upravo prilagođavanje generičkog procesa pojedinoj mreži. Ovo poglavlje opisuje sve faze procjene ranjivosti i probleme koji se u pojedinim fazama rješavaju, od definiranja što točno treba napraviti do definiranja izlaza iz cijelog procesa.

3.1 Dogovaranje procjene ranjivosti

Ugovaranje procjene i definiranje osnovnih parametara procjene ranjivosti može se zapravo smatrati nultom fazom cijele priče. To se ovdje spominje jer zapravo ugovor i njegovi detalji imaju dubok utjecaj na tehničku izvedbu, doslovce se manifestirajući u izboru opcija koje nudi programska podrška pri skeniranju.

Osim određivanja cijene posla i rokova, osnovna pitanja ove faze su:

- Koji je opseg procjene, na koje se mreže odnosi?

  Organizacija se može sastojati od manjih, dobro razdvojenih podtvrtki koje dijele infrastrukturu ili može željeti procjenu ranjivosti samo nekog dijela svoje mreže. U takvim slučajevima mora se definirati opseg ispitivanja u vidu uključenih poslužitelja i ostale opreme. Potrebno je, dakle, odrediti granice mreže (imajući u vidu posebnosti poput VPN-a).

- Kakve testove želimo, kakve ne želimo?

  Ranije je napomenuto da testovi mogu biti opasni po normalno funkcioniranje mreže, što se doista i pokazalo u praksi. Korištenje alata za procjenu ranjivosti može uzrokovati probleme i naručioc ima puno pravo zaobići takve testove – poput DoS napada koje se može pokrenuti kao dio procjene. Osim toga, postoje i testovi koji traže osjetljive podatke na stanicama i naručioc mora na takve biti upozoren, kako bi očuvao željenu razinu privatnosti.

- Kako kontrolirati tajnost podataka?

  Rezultati ispitivanja su zapravo određen skup dokumenata. Otkrivanje njihovog sadržaja se mora kontrolirati prema željama naručioca.

- Kakve izvještaje trebamo?

  Sve akcije koje se poduzimaju nakon procjene ranjivosti ovise o izlaznim izvještajima, a njihov oblik ovisi o namijenjenoj publici.

- Koja će biti frekvencija procjene ranjivosti?

  Učestalost skeniranja mora se dogovoriti na početku posla, a ovisi o mjestu procjene ranjivosti u sigurnosnoj politici organizacije.

Sljedeća faza, u kojoj se donose prve tehničke odluke i postavlja programska podrška pod najvećim je utjecajem odluka iz ove faze. Osim općenite naklonosti višeg menadžmenta cijelom procesu skeniranja (na što se može računati, s obzirom da se posao ugovara), važnu ulogu ima i izvođač, i to u kvalitetnom informiranju naručilaca o svim bitnim karakteristikama sigurnosnog skeniranja – od prednosti, do potencijalne štete.

3.2 Pripremne radnje

Promatrajući procjenu ranjivosti isključivo kao tehnički proces, ovo se može nazvati prvom fazom procjene. Pripremne radnje uključuju pretežito tehničke odluke poput izbora lokacije korištenog alata za automatiziranu procjenu ranjivosti (dalje u tekstu se izraz "skener" koristi za pretraživač aktivnih pristupa i sigurnosni analizator) i izbora testova (iz kompletne ponude koju nudi korišteni alat). Također, u ovoj fazi se provodi diferencijacija testova prema karakteristikama resursa koji se skeniraju. Kao pomoć u ovim pitanjima, računa se na administratore ciljne mreže – općenito, kroz cijelu ovu fazu se u velikoj mjeri surađuje s tehničkim osobljem organizacije naručioca i treba naglasiti da ovaj odnos, bez obzira na ugovoren posao i obavezu suradnje, mora biti što pozitivniji. U protivnom, izvedba procjene ranjivosti može na vrlo suptilne i netehničke načine puno izgubiti na kvaliteti.

Postoji par osnovnih čimbenika koji utječu na proces više nego ostali. To su:

- izbor gledišta,
- svojstva ciljne mreže,
- svojstva i namjena pojedinih računala u mreži.

3.2.1 Izbor gledišta

Kada razmatramo sigurnost neke mreže računala, moramo se (između ostalog) odlučiti iz koje perspektive promatramo mrežu, a različite perspektive impliciraju različite uvjete i oružja koja imamo na raspolaganju. Tipično postoji par zanimljivih uloga u koje se vrijedi postaviti (valja napomenuti kako ovo nipošto nije kompletna lista i kako mogućnosti ima više):

- vanjski napadač,
- korisnik,
- poslovni partner.

U ulozi vanjskog napadača sustav promatramo izvana, kao potpuno nepoznat. Izbor ove pozicije implicira primjerice promatranje ciljne mreže s neke druge, odvojene mreže i postojanje sigurnosne zaštitne stijene između nas i sustava. Iz ove perspektive, može nam biti želja ostvariti neovlašteni pristup sustavu, pokušati na neki način onesposobiti sustav (npr. uskratiti uslugu) ili slično. U procjeni sigurnosti, ovo stajalište je dosta važno i ovaj oblik ispitivanja je poželjno izvršiti, budući da je smisleno očekivati ovu vrstu razmišljanja od strane pravog napadača, kojem je to doista situacija u kojoj se nalazi. Donekle, ova se funkcionalnost preklapa s penetracijskim ispitivanjem iste vrste, ali u izvedbi nije toliko temeljita. Procjena ranjivosti ne iskazuje svoju najveću vrijednost na ovaj način.

Klasična paradigma napada na sustav započinje s pridobivanjem pristupa sustavu s ovlastima običnog korisnika, kako bi u nastavku aktivnosti osigurali veće ovlasti ili zlorabili tuđi identitet. Iz tog razloga je zanimljivo pretpostaviti da je napadaču to uspjelo i proučiti što tada može učiniti. Zato možemo odabrati gledište običnog korisnika. U ovom slučaju, naša je pozicija na ciljnoj mreži i time puno pogodnija za napad. Valja imati na umu da nisu sve ranjivosti isključivo zlonamjerni napadi izvana. Na primjer, slučajne korisničke greške mogu biti jednako razorne kao i vanjski napadi.

Ova pozicija, ovisno o ovlastima korisnika, daje najveću moć procjeni ranjivosti i dopušta uobičajenim postupcima otkrivanje najveće količine informacija. Ovo donekle ovisi o vrsti mreže i metodama autentifikacije legalnih korisnika, no u najčešćem slučaju, situacija je

pogodna – legitimni korisnik ima pristup većem broju računala i autentifikacija korisnika je centralizirana. Ovakva konfiguracija omogućava vrlo elegantno skeniranje velikog broja računala na najbolji mogući način, a to je ovlašteno skeniranje. Ako korisnik ima ovlasti prijaviti se na neku stanicu i pristupiti njezinim vlastitim informacijama, sakupljanje informacija postiže vrlo visoku efikasnost.

Pozicija poslovnog partnera je nešto kompleksnija i zapravo uključuje pomalo od obje prethodne grupe. Poslovni partner se nalazi na odvojenoj mreži, ali ima neke ovlasti ili neki oblik povlaštenog pristupa ciljnoj mreži, poput VPN-a. Ovo nam je gledište zanimljivo jer nije uvijek jasno definirano te implicira određenu količinu iznimki od uobičajenih pravila. Potrebno je odlučiti kako se postaviti prema poslovnom partneru i prema mogućnosti da njegova mreža bude kompromitirana. S druge strane, ako je naša mreža kompromitirana, moramo biti svjesni utjecaja koji bi to moglo imati na poslovnog partnera radi očuvanja poslovnog odnosa koji bi u slučaju nemara mogao biti narušen.

3.2.2 Svojstva ciljne mreže i stanica na mreži

Veličina, segmentacija, eventualne posebnosti te način korištenja mreže i pojedinih podmreža spadaju u svojstva mreže koja utječu na parametre skeniranja. Ukupan proces skupljanja svih potrebnih informacija naziva se mapiranje mreže.

Mapiranje ciljne mreže može se promatrati kao svojevrstan popis inventara i nezaobilazan je dio procesa procjene ranjivosti. Na kraju ovog koraka moramo poznavati kompletnu strukturu mreže. Ključne informacije koje treba sakupiti u ovoj fazi uključuju:

- Ima li mreža "uska grla" i ako da, gdje se nalaze?
- Gdje su granice mreže, od kojih podmreža se sastoji?
- Gdje se nalaze sigurnosne zaštitne stijene i IDS senzori?
- Koji protokoli za usmjeravanje se koriste, gdje su usmjernici?
- Gdje su izlazi na Internet? Gdje su izlazi prema mrežama poslovnih partnera?
- Gdje se nalaze kritična računala na mreži (npr. računala koja u slučaju prestanka rada uzrokuju nedostatak povezivosti ostalim računalima)?
- Gdje su uopće ikakva računala/stanice na mreži? Koji operacijski sustav je na kojem računalu? Koji servis se nudi na kojem računalu?

Procjena ranjivosti koja započne bez odgovora na ova pitanja će u najboljem slučaju biti površna, a često beskorisna. Dobro poznavanje sustava može drastično ubrzati cijeli proces te povećati pouzdanost i čitljivost rezultata.

Dobar dio mreža u produkciji odlikuje se kompleksnom strukturom, što pogotovo vrijedi za mreže koje su u pogonu duže vrijeme i koje su kroz to vrijeme doživjele više manjih i, primjerice, slabo dokumentiranih izmjena. Nije rijedak slučaj da niti administrator mreže ne zna detaljno odgovoriti na sva navedena pitanja (koja nisu sva koja se u ovoj fazi postavlja). Valja napomenuti da razlozi za ovu mogućnost prelaze običnu lijenost ili slično – odgovornom administratoru to može biti samo jedna od više desetaka mreža koje održava ili je mreža "naslijeđena" sa slabom dokumentacijom. Postoji, dakako, i sasvim suprotna mogućnost, a to je da organizacija ima dobro definiran sustav s kvalitetnom dokumentacijom i/ili da ima (i implementira) striktnu sigurnosnu politiku. U tom slučaju su odgovori na većinu gornjih pitanja lako dostupni, a ovaj izuzetno važan korak je tada brzo gotov.

Uska grla i kritični dijelovi mreže

"Usko grlo" može biti bilo koji segment sustava koji u normalnom radu koristi potpuni vlastiti kapacitet i predstavlja najslabiji, najopterećeniji odnosno najsporiji dio sustava. U slučaju potrebe za povećanjem produktivnosti, takav segment nije u stanju prilagoditi se težim uvjetima rada i ostatak sustava bi zato bio usporen. Ako je takav dio mreže od velike važnosti, a uzmemo li u obzir kako sigurnosno skeniranje često implicira veliko opterećenje mreže, neoprezno opterećenje takvog segmenta može uzrokovati prekid rada sustava. Iz tog razloga je važno poznavati sadrži li sustav takve dijelove i gdje se nalaze, kako bi se proces prilagodio.

Slično razmišljanje vrijedi za pojedine stanice od velike važnosti. "Napadnemo" li, i uspješno srušimo važan usmjernik na početku skeniranja, ostali rezultati će vjerojatno biti posve neupotrebljivi. Treba obratiti posebnu pažnju na takve točke u sustavu, pogotovo ako o njima ovisi povezivost ostalih stanica. U slučaju postojanja sigurnosne politike, popis kritičnih stanica vjerojatno već postoji i treba ga iskoristiti. U protivnom, zadatak mapiranja mreže je saznati gdje su takve stanice.

Gdje su aktivne stanice?

Nadalje, mreže mogu biti jako velike, u smislu da imaju mogućnost spajanja ogromnog broja stanica, a da se stvarne adrese raspodjeljuju nasumično i dinamički (recimo, pomoću servisa DHCP). U tom slučaju, poseban je problem pronaći adrese aktivnih stanica u potencijalno ogromnom broju mogućih adresa. Postoje kvalitetni alati koji pomažu u rješavanju ovog problema i o njima će biti više riječi kasnije.

Suradnja s ostalim uređajima na mreži

Sigurnosne zaštitne stijene, IDS senzori i usmjernici također zahtijevaju posebnu pažnju jer oni određuju gdje će se postavljati skeneri za procjenu ranjivosti. Mada će o njima više riječi biti u kasnijim poglavljima, takvi skeneri generiraju velike količine prometa i taj promet na različite načine djeluje na svaku od navedenih mrežnih naprava.

Sigurnosne zaštitne stijene će gotovo sigurno blokirati skeniranje. Rezultati su tada nepouzdani i pretjerano optimistični jer zapravo nedostaju. S druge strane, ako smo odabrali gledište vanjskog napadača, upravo je rad zaštitne stijene meta ovakvog skeniranja, odnosno ono se koristi kao provjera kvalitete zaštitne stijene.

Svako skeniranje u određenoj mjeri simulira napad na temelju ranjivosti koju ispituje. Logično, promet koji se generira će sustav za otkrivanje napada protumačiti kao napad i reagirati ovisno o vlastitoj konfiguraciji za taj pojedini okidač. Ako je reakcija aktivna, ostatak procjene ranjivosti može biti upitan i zato je potrebno ove sustave prilagoditi ili privremeno onesposobiti za trajanja procjene. Opet, procjena ranjivosti kao sekundarnu pogodnost nudi upravo i ispitivanje kvalitete IDS sustava – ako IDS propusti reagirati na neku vrstu napada tokom procjene ranjivosti, mora se imati na umu da tako neće reagirati i u slučaju pravog napada. Dodatno, vrijedi napomenuti i sljedeći potencijalni problem. IDS sustavi osluškuju promet, eventualno povezuju događaje, a osim što obavještavaju administratore o svemu, oni jako puno podataka zapisuju u dnevnike (eng. *log*). Uz automatizirano skeniranje iznutra, događalo se da ti zapisi jako porastu i zapune sav diskovni prostor na poslužitelju.

Nakon temeljitog upoznavanja ciljne mreže, morali bi raspolagati mapom mreže - popisom svih podmreža, adresama svih aktivnih stanica i relevantnim informacijama o svakoj stanici, poput operacijskog sustava koji je na stanici, namjene stanice i razine kritičnosti (važnosti).

Vrijedi napomenuti da se u stvarnoj situaciji neke od tih informacija, poput operacijskog sustava stanice ili servisa koje nudi mogu dobiti kasnije, brzim inicijalnim skeniranjem.

Kada poznajemo sve ove informacije, možemo nadopuniti prethodni korak i ponuditi nešto konkretnije mogućnosti za lokaciju s koje ćemo skenirati mrežu:

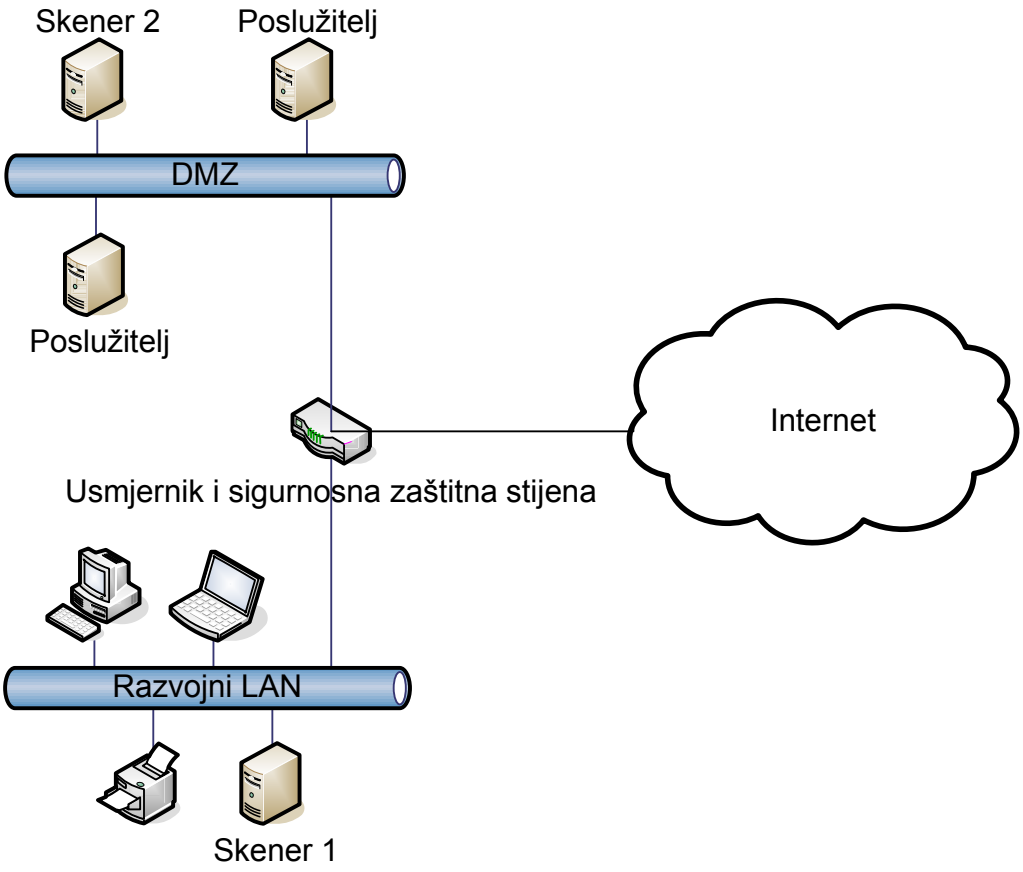
- Skeniranje kritičnih stanica s lokacije unutar ciljne mreže: Kritične stanice zaslužuju nešto veću pažnju i želimo ih dobro provjeriti uz pretpostavku da su dostupne napadaču.
- Skeniranje stanica spojenih na Internet s lokacije izvan ciljne mreže: Uglavnom nisu sve stanice spojene izravno na Internet, nego to vrijedi samo za neke. Takva procjena ranjivosti takvih stanica je vrlo uobičajena jer one predstavljaju prve mete za napadača s Interneta.
- Skeniranje stanica spojenih na Internet s lokacije koja je na njihovoj (istoj) domeni razašiljanja (*brodcast domain*): Pretpostavimo li da je jedna od stanica koje su dostupne na Internetu uspješno kompromitirana, zanima nas kakav to utjecaj ima na ostale takve stanice. Dakle, zauzimamo ulogu napadača koji ima pristup domeni razašiljanja stanica koje su spojene na Internet.
- Skeniranje svih stanica s lokacije unutar ciljne mreže: Velika je vjerojatnost da će ovakvo skeniranje rezultirati s izuzetno puno informacija, pogotovo ako je to prvi izbor. U hrpi informacija se teško snaći i lako se dogodi da vrijeme potrošeno samo na dodjelu prioriteta svakoj od stavki izvještaja postane znatno. Omjer uloženog i dobivenog za ovakvo skeniranje je nešto povoljniji ako želimo proučiti izuzetno važan sustav koji je već dobro osiguran. No, ovakvo skeniranje daje najpotpuniju informaciju o cjelokupnom sustavu.

Vezano uz lokaciju skenera, vrijedi posebno temeljito razmotriti usmjernike. Usmjernici stoje na granicama podmreža i kroz njih prolazi sav međumrežni promet. Povećanjem tog prometa možemo preopteretiti usmjernik i onemogućiti (ili dosta usporiti) normalan rad mreže.

Iz navedenog se može zaključiti kako je općenito pogodno postaviti po jedan skener na svaku zasebnu podmrežu, kako je prikazano na slici 3.1.

Slika 3.1. prikazuje jednostavan sustav od dvije podmreže usmjernikom spojene međusobno i na Internet. U gornjem dijelu nalazi se demilitarizirana zona (DMZ) koja sadrži poslužitelje vidljive s Interneta, a niže je privatna razvojna podmreža, koju sačinjavaju radne stanice tipične za ured. Za ovakvu konfiguraciju su optimalna dva skenera, po jedan na svakoj podmreži. Uporaba samo jednog bi uzrokovala velik promet na usmjerniku, a osim toga, mogla bi biti otežana činjenicom da je usmjernik ujedno i sigurnosna zaštitna stijena. Rezultat takve konfiguracije bi bila lošija i teža procjena ranjivosti, koja bi opteretila usmjernik i smanjila dostupnost resursa interno i izvana.

Smještanjem skenera unutar svake podmreže, postiže se više od zaštite drugih naprava – drastično se povećava brzina skeniranja. Kako će biti objašnjeno kasnije, skener može zauzeti puno resursa na računalu na kojem je pokrenut (ovisno o konfiguraciji), a dodatna pogodnost je paralelizacija skeniranja. Podjelom posla između dva servera, osim olakšanja problema resursa, ta se faza može odvijati istovremena za dvije podmreže.



Slika 3.1. Smještaj skenera po podmrežama

3.2.3 Ovlasti

U slučaju da se organizacija odluči za temeljito skeniranje iznutra, korisno je dozvoliti izvršitelju pristup skeniranim računalima. Ovisno o dopuštenoj razini pristupa, rezultati skeniranja mogu izuzetno varirati, od posve optimističnih i pogrešnih, prema realističnijima. Preciznost skenera izravno ovisi o ovlastima koje skener ima na ciljanom sustavu; uz veće ovlasti, skener ima mogućnost ispitivanja verzija programa, prava pristupa datotekama i čak provjere verzije binarne izvršne datoteke koja je u izvođenju, i koja se može razlikovati od verzije programa navedene na drugim "pouzdanim" mjestima.

U pripremne radnje, dakle, spada i dodjela ovlasti. Ako se radi o mreži koja ima centralnu autentifikaciju, najpogodnije je dodati korisnika samo za potrebe procjene ranjivosti i modificirati mu ovlasti prema potrebi. Za računala koja se autentificiraju odvojeno, što je čest slučaj kod izdvojenih poslužitelja (internih i na DMZ-u jednako), možda je potrebno "ručno" dodati korisnika.

3.2.4 Zaključno o prvoj fazi

Ovaj korak u procjeni ranjivosti ima presudan utjecaj na kvalitetu cijelog procesa. Slijedeći korak je samo skeniranje, koje je automatizirano i koje prethodi pregledu rezultirajućih izvještaja – što znači da se greške uvedene u ovom koraku propagiraju do kraja i ne mogu se ispraviti nikako osim povratkom u prvu fazu. Iz tih razloga je ova priprema, u stvarnoj

okolini, daleko najdugotrajniji dio procesa. Uključuje istraživanje i upoznavanje ciljne mreže, suradnju s novim ljudima, brojne odluke i, na kraju, instalaciju određene količine softvera na neodređen broj lokacija unutar mreže. Također, prvo skeniranje – ono vezano uz pretragu za aktivnim stanicama – obavlja se u ovoj fazi, kao korak u mapiranju mreže.

Nakon ove faze trebali bi biti ispunjeni sljedeći preduvjeti za nastavak:

- Instalirana programska podrška

  Za automatiziranu procjenu ranjivosti potreban je određen broj alata, o kojima će biti riječi u nastavku. Na kraju ove faze, kao zadnji (i zapravo prijelazni) korak mora se pripremiti programska podrška – svi skeneri moraju biti instalirani na odabrana mjesta, prema preporukama opisanim ranije.

- Spremne prilagođene karakteristike mreže

  Mapa mreže mora biti prikazana u obliku popisa podmreža, spremna za unos u programsku podršku. Također, za potrebe prilagodbe testova pojedinim podmrežama, sve relevantne posebnosti moraju biti sažeto navedene. U sljedećoj fazi se ne smije trošiti vrijeme na planiranje.

- Spreman plan ispitivanja

  Ukoliko je moguće, zgodno je imati raspored skeniranja po danima, uz određenu vremensku toleranciju u predviđenom trajanju (skeniranje zna potrajati dugo). No, što će se ispitivati na kojoj vrsti resursa i plan upravljanja ovlastima skenera – to se mora dogovoriti prije skeniranja. Potrebno je predvidjeti i moguće ispade sustava i biti na njih spreman.

3.3 Otkrivanje aktivnih stanica i njihovih ranjivosti

Sigurnosno skeniranje (otkrivanje stanica i ranjivosti na njima) je druga faza izvedbe i u ovoj fazi je na raspolaganju najveća pomoć računala, kroz uporabu specifičnih alata. Ranije je analizirana okolina u kojoj rade današnje mreže – brojne ranjivosti i velik broj korisnika koji na automatizirani način traže ranjiva računala. Logičan način borbe protiv toga je automatizacija sustava zaštite. Danas sigurnosni stručnjaci također raspolažu alatima koji na automatiziran i brz način pomažu u otklanjanju poznatih propusta. O samim alatima bit će više riječi kasnije, no ovdje treba napomenuti da za njihovo korištenje moramo "znati što želimo", odnosno moramo imati dobru podlogu iz prethodne faze. Naime, od velikog broja poznatih ranjivosti, samo se neke mogu pojaviti u određenim uvjetima. Primjerice, želimo li skenirati Windows ili Linux računalo, nema smisla isprobavati je li ranjivo na propuste specifične za, npr., Cisco mrežnu opremu. Ovdje valja podsjetiti na važnost mapiranja mreže jer se kvalitetnom mapom može izbjeći niz ovakvih besmislenih testova. Ranjivosti, osim što su brojne, uvijek su čvrsto vezane uz vrstu mete, što nam olakšava posao. Zanemarimo li to, mogli bi primjerice koristiti logiku "isprobaj sve testove za sve moguće IP adrese", misleći kako ćemo tako postići fantastičnu razinu sigurnosti! Primijenimo li taj pristup u slučaju s 254 različite adrese i 10000 različitih testova, dobijemo 2 540 000 operacija (testova) koje treba obaviti. U posve optimističnim, idealnim uvjetima (u pogledu mrežne propusnosti i sklopovskih zahtjeva skenera), ovakva bi procjena trajala dugo. Izvještaj kakav bi proizvelo ovakvo skeniranje bio bi sasvim sigurno neupotrebljiv.

U ovom koraku, dakle, imajući na umu sve informacije skupljene do sada, određujemo koje testove želimo obaviti za koju stanicu i, na kraju, pokrećemo skeniranje. Uz kvalitetno upravljanje, alati za ovu namjenu obavit će najveći dio posla. Ipak, bez obzira na svu

pripremu, ovaj korak ima svoja nova pitanja te se moramo pripremiti na neke teže odluke i kompromise.

Ranije je spomenuto kako neki testovi ranjivosti mogu biti opasni. U nekim slučajevima mogu postići isti efekt kao i ranjivost za čije ispitivanje služe (što može značiti i dugotrajan prestanak određene usluge), a zabilježeni su i slučajevi ozbiljnijih oštećenja na opremi. Uzevši u obzir kako sve alate koji imaju ovakve mogućnosti posve legalno mogu nabaviti i napadači, očito rješenje nije u tome da takve testove jednostavno preskočimo, no o njihovom pokretanju se odlučuje kod potpisivanja ugovora. Nadalje, pojedini testovi mogu imati možda ne poguban, ali dubok i primjetan utjecaj na funkcioniranje mreže budući da su zbog prirode ranjivosti koju ispituju ozbiljan potrošač resursa. Neki testovi jako dugo traju i mogu znatno produljiti vrijeme skeniranja. Činjenica da svaki test možemo, ali i ne moramo pokrenuti dovodi do niza odluka koje su u ranijim fazama bile općenite i nespecificirane.

Kada govorimo o kompromisima, zapravo se radi o balansiranju sljedeća tri poželjna svojstva skeniranja:

- brzina,
- preciznost (točnost),
- stabilnost.

Ta su svojstva najčešće u konfliktu.

U ovom kontekstu, brzina ima dva aspekta - brzina procesa skeniranja i brzina (propusnost) ciljne mreže. Sam proces skeniranja može po prirodi biti brz, kao primjerice pronalazak aktivnih računala u nekoj podmreži, ali može biti i spor, poput ispitivanja sadržaja *web* poslužitelja koje zahtijeva replikaciju svake pojedinačne stranice koju poslužuje. Mada na prvi pogled brzina može izgledati kao pogodnost koje se možemo odreći, u praksi ipak igra važnu ulogu. Preciznost, odnosno točnost procjene ranjivosti odnosi se kako na istinitost rezultata, tako i na potpunost. Na točnost rezultata utječu faktori poput detaljnosti testova, vrijeme ispitivanja i kvaliteta mape mreže. Stabilnost se prvenstveno odnosi na stabilnost cijelog ciljnog sustava.

Bilo je dosta riječi o izboru lokacije za skeniranje, a vrijeme skeniranja je do sada bilo nevažno. Međutim, za sustav koji je u pogonu, vrijeme koje odaberemo (kao i vrijeme koje smo voljni potrošiti na skeniranje) može utjecati na tri opisana svojstva. Primjerice, odlučimo li skenirati sustav po noći, ili općenito izvan radnog vremena, nastaje mogućnost zanemarivanja ugašenih računala, koja su inače u pogonu. Pogodnost je veća sloboda, ne prijetimo stabilnosti sustava i uživamo bolju propusnost mreže, ali preciznost je na gubitku. Pokrenemo li skeniranje za radnog vremena, moramo paziti na stabilnost i brzinu sustava, kako ne bi smanjili produktivnost naručioca. Očito postižemo bolju preciznost, ali nauštrb brzine.

Kada isprobati eventualno opasne testove, ako su oni dogovoreni? Kada dozvoliti jaka opterećenja na mrežu? Koliku detaljnost testova dozvoliti? Opet, nema univerzalnog odgovora, ali mogu se pratiti neke smjernice. Opasne testove valja izvoditi u prisutnosti administratora, koji će sustav vratiti u normalno stanje ako dođe do problema. Ovo pogotovo vrijedi za poslužitelj koji je u produkciji. S druge strane, najpogodnije vrijeme za opasne testove je prije puštanja novog sustava u pogon – kao i kod ostalih testova, ovo vrijeme ne opterećuje produkcijskim zahtjevima i stvara bolji temelj za kasniji rad i sigurnost.

Ostatak posla u ovoj fazi svodi se na konfiguraciju skenera i njegovo pokretanje. Od pokretanja nadalje je skeniranje automatizirano i većinu posla obavlja računalo bez ljudske intervencije – u idealnom slučaju. Uloga administratora i naručioca je pritom paziti na cijelu

mrežu za vrijeme skeniranja i reagirati prekidom skeniranja, ako se pokaže potrebno. Vrijedi napomenuti da opasni testovi "samo" sruše ciljani servis, ako mogu. Oporavak u većini slučajeva zapravo znači ponovno pokretanje problematičnog servisa i najčešće ne predstavlja problem.

Izlaz iz ove faze je izvještaj koji generira skener.

3.4 Analiza rezultata i rad na izvještajima

Analiza i rad na izvještajima predstavljaju treću, konačnu fazu procjene ranjivosti. Svi danas dostupni alati za procjenu ranjivosti implementiraju više ili manje napredne mogućnosti izrade izvještaja i pomoći pri analizi. Iako je uredan i pregledan izvještaj neophodan za daljnju analizu rezultata, tumačenje rezultata složen je i osjetljiv posao ostavljen isključivo sigurnosnom stručnjaku, ponajviše radi nesavršenosti alata. Dodatno, vrijedi imati na umu da su za različite ljude potrebni različiti izvještaji.

Dostupni alati su posljednjih godina podosta napredovali na svaki način, nudeći u području izrade izvještaja primjerice rangiranje pronađenih ranjivosti. No upravo je ta mogućnost dvosjekli mač i upućuje na prvu opasnost u tumačenju rezultata.

Nesavršenost alata manifestira se na sljedeća dva načina:

- pogrešna prijava ranjivosti;
- neuspjeh u otkrivanju ranjivosti.

3.4.1 Pogrešne prijave ranjivosti

Pogrešna prijava ranjivosti (*false positive*) je smetnja koja uzrokuje gubitak vremena i uzrok je čestih kritika na račun automatiziranih sigurnosnih analizatora. Mada je šteta koju ova nesavršenost uzrokuje relativno neznatna, kritike koje izaziva su često su vrlo strastvene i ova se nesavršenost nerijetko koristi kao dokaz tehničke inferiornosti tih alata. Tipičan scenarij koji dovodi do takve situacije je: predamo veliki, automatski generirani izvještaj (kojeg nismo pročitali) prezaposlenom administratoru; administrator čitajući izvještaj uoči par ranjivosti označenih kao izuzetno opasne, a za koje zna da ne mogu postojati na određenoj stanici (npr., upozorenje o nezakrpanom Microsoft Internet Exploreru na stanici koja uopće nema instaliran taj preglednik, kao ni MS Windows OS, već primjerice Opera preglednik na Linux OS-u). Nakon par očito pogrešnih prijava, posve prirodno se javlja sumnja u valjanost ostalih stavki izvještaja i uopće korisnost alata.

Do takvih problema dolazi iz dva razloga - nedovoljna ili nepostojeća analiza rezultata i nepotpune pripremne radnje prije skeniranja (nepoznavanje ciljne mreže). Treba prihvatiti da su automatski sigurnosni analizatori, bez obzira na dosadašnji napredak, ipak nesavršeni i da se ne treba slijepo pouzdati u njihove rezultate. Mada dizajneri tih alata ulažu velik trud da predvide moguće reakcije sustava na skeniranje i pojedine testove, *sve* mogućnosti nije lako pokriti. Kao i u drugim djelatnostima, alat je pomagalo, a ne rješenje problema sam po sebi.

Razlozi ovakve pogreške najčešće mogu biti opravdani tehničkim razlozima, odnosno postoje poznate konfiguracije koje mogu uzrokovati pogrešne prijave. Primjerice, ako skenirana stanica zapravo proslijeđuje sav promet stanici u pozadini ili to vrijedi za promet na određenim vratima, pogrešne prijave su česte. Česta je konfiguracija u kojoj možemo skenirati jednu IP adresu koja je zapravo virtualna adresa za veći broj stanica u pozadini, s istim rizikom pogrešnih rezultata. Sustavi mogu koristiti *proxy* programsku podršku, mogu biti namjerno izmijenjeni kako bi zavarali analizatore itd. Ovdje valja još napomenuti kako neki sigurnosni analizatori pretpostavljaju mogućnost vlastite pogreške i u određenim

sumnjivim situacijama to jasno navode u izvještajima. Uz određenu stavku izvještaja može se nalaziti napomena kako je prijava potencijalno pogrešna, ovisno o implementaciji testa.

3.4.2 Neuspjeh u otkrivanju ranjivosti

Neuspjeh u otkrivanju ranjivosti je, s druge strane, puno ozbiljniji problem. Naime, programska podrška ograničena je onime što je do nekog trenutka poznato. Sve poznate ranjivosti su pojedina otkrića ili incidenti koji su vezani uz pojedinu verziju nekog programa ili određenu konfiguraciju nekog programa. Analizatori koriste prilagođenu bazu takvih potpisa kako bi uočili ranjivost. Dok se nalazimo u domeni poznatih ranjivosti, današnji sigurnosni analizatori su u načelu zadovoljavajuće pouzdani. Međutim, ranije spomenuta raznolikost prijetnji implicira da je izuzetno teško osloniti se na bilo kakvu heuristiku za prepoznavanje do sada nepoznatih prijetnji. Iako i na tom području postoje pomaci, analizatori su ograničeni do sada poznatim. Također, procjena ranjivosti, za razliku od penetracijskog ispitivanja, ne bavi se povezivanjem informacija i to znači da ako kombinacija dviju manjih ranjivosti na dva poslužitelja, npr. FTP i neke baze podataka, skupa tvori jednu veliku rupu, npr. otkrivanje lozinki korisnika – automatizirana procjena ranjivosti je neće obavezno uočiti. To je izvan granica procjene ranjivosti i ne bi se smjela računati u slabost procesa, osim ako je procjena ranjivosti jedini sigurnosni proces koji se implementira (što je greška samo po sebi).

Osim kvalitetnog izvještaja, na kraju procesa procjene ranjivosti moramo raspolagati još nekim znanjima – moramo predložiti načine da se sigurnosno stanje popravi. Ovaj zahtjev postavljen je pred sigurnosne analizatore u prošlosti i kroz godine izlaženja novih inačica, dosta je kvalitetno zadovoljen u većini slučajeva. Gotovo svaki proizvođač programske potpore danas na neki javni način vodi bitku sa sigurnosnim propustima, što pogotovo vrijedi za proizvođače operacijskih sustava. Tipično je na *web* stranicama proizvođača neki segment posvećen izvještavanju javnosti o poznatim ranjivostima i načinima njihovog otklanjanja. Sigurnosni analizatori se jako oslanjaju na tu uslugu i koriste je u vlastitim izvještajima. Tako se uz gotovo svaku stavku izvještaja može pronaći (uz kratki opis same ranjivosti) referenca na način otklanjanja prijetnje ili izravan opis unutar same stavke, ako je prikladno.

3.4.3 Modifikacija izvještaja

Izvještaj kojeg proizvede alat za procjenu ranjivosti ne mora (i često ne može) biti dovoljan za svu publiku. Izvještaj gotovo uvijek sadrži neki sažetak ukupnog skeniranja i sažetak skeniranja svakog pojedinog resursa – i taj dio čini dobar temelj za izvještaj kakav je potreban višem menadžmentu. Detaljna specifikacija pojedinih ranjivosti s opisima načina popravka nije potrebna na toj razini.

S druge strane, administratori sustava preferiraju tabelarni prikaz rezultata, grupiran po pojedinom resursu i po mogućnosti s dodijeljenim prioritetom. Na taj način praktički imaju popis problema i onda u vlastitoj organizaciji, prema ustaljenim pravilima koja vrijede u organizacija, mogu interno distribuirati probleme u obliku zadataka pojedinim zaposlenicima. Izrada svih izvještaja se ponajviše dogovara na početku procesa, no praksa pokazuje kako se taj dio posla mijenja ovisno o rezultatima.

S opisom izvještaja je zaključen teorijski opis procjene ranjivosti mrežnog sustava i u nastavku rada bit će opisani neki dostupni alati. Izložena metoda odlikuje se općenitošću i fleksibilnošću, riječ je o generičkom opisu procesa koji je prilagodljiv, prateći naputke koji su također izloženi za svaku fazu. Nakon opisa alata, bit će prikazan i praktični dio rada, u kojem je demonstrirano kako se teorijski pristup preslikava u konkretnu izvedbu na produkcijskoj mreži.

4. Alati

Ovo poglavlje govori o alatima koji se koriste u procesu procjene ranjivosti. Bit će izložen jedan izuzetno popularni alat koji se posebno iskazuje u području mapiranja mreže i dva alata za automatiziranu procjenu ranjivosti, oba izuzetno cijenjena na tom području.

4.1 Nmap

Nmap je vrlo popularan alat u svijetu računalne sigurnosti. Ime mu dolazi od *network mapper*, što bi se moglo prevesti kao *alat za mapiranje mreže*. Jasno je kako bi nam taj alat mogao biti koristan u ranije opisanoj prvoj fazi procesa procjene ranjivosti.

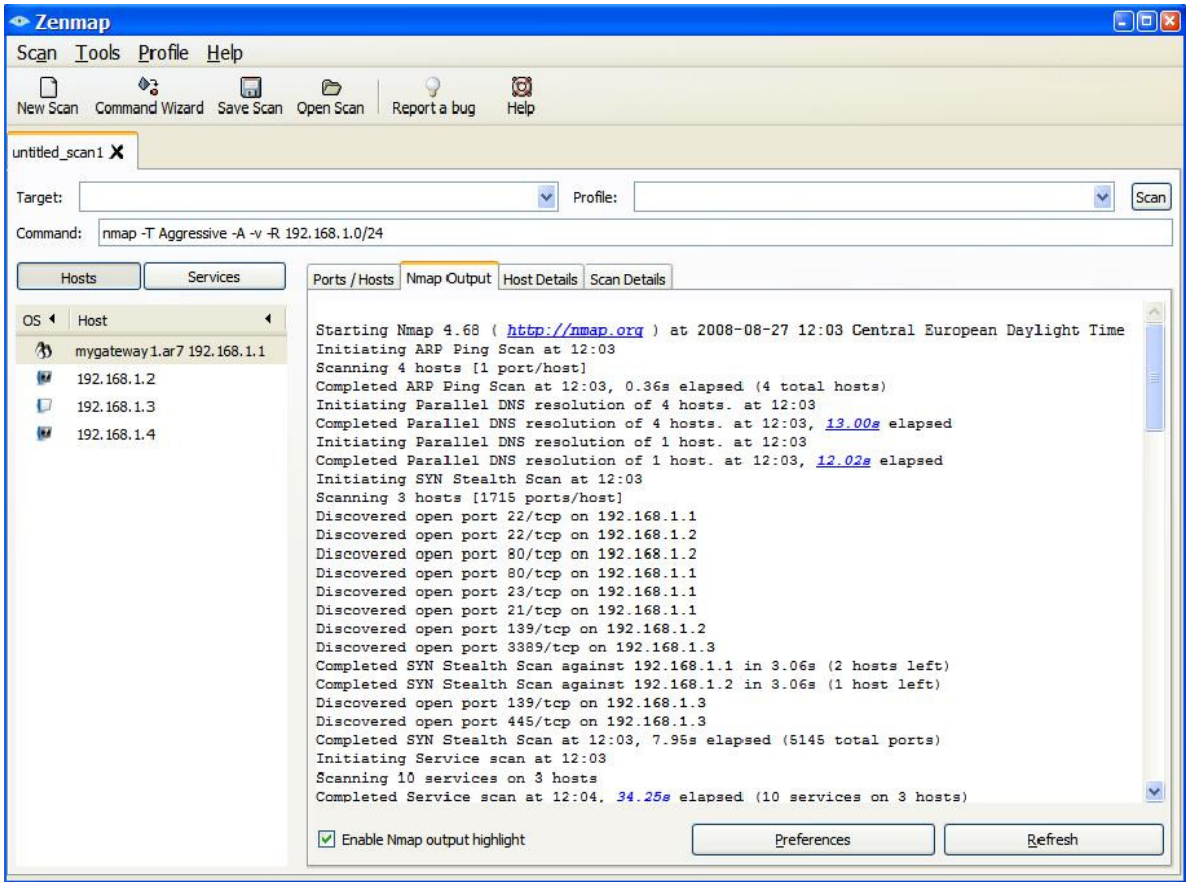
Mada Nmap ima pomalo "hakersku" reputaciju, radi se o poznatom i zrelom alatu otvorenog izvornog teksta (objavljen pod GNU GPL licencom) koji služi za istraživanje mreže i, donekle, procjenu sigurnosti sustava. Nmap koristi svojstva TCP i IP protokola kako bi saznao koje su stanice na mreži aktivne i koji pristupi su aktivni te nude li uslugu za koju su namijenjeni. To mu je primarni zadatak a uz tu, osnovnu funkcionalnost, Nmap je sposoban izvršiti niz testova za utvrđivanje dodatnih informacija o mreži i pojedinim stanicama, poput tipa i verzije operacijskog sustava na stanici, postojanje (i eventualno vrstu) sigurnosne zaštitne stijene i slično, koristeći brojna saznanja o ponašanju pojedinih operacijskih sustava i aplikacija te poznavanju njihove reakcije na pravilne i nepravilne "podražaje".

Nmap odlikuju brojne opcije, kojima je moguće izvršiti skoro svaku vrstu skeniranja proizvoljnog broja računala. U svom radu, Nmap nudi kontrolu i nad brzinom skeniranja, modifikaciju izvorišne adrese, ranije opisani *idlescan*, slanje lažnih paketa s više lažnih izvorišta i mnoge druge metode kako bi prikrio svoj trag, nudeći tako mogućnosti zavaravanja sustava za otkrivanje napada. Također, Nmap je vrlo efikasan u pronalasku aktivnih računala na istoj lokalnoj mreži (eng. *Local Area Network, LAN*) korištenjem protokola ARP, neovisno o postojanju sigurnosne zaštitne stijene koja otežava otkrivanje na višim slojevima. Dodatno, jedna od ljepših sposobnosti programa je razmjerno pouzdano otkrivanje operacijskog sustava stanice. Kombinacijom ovih (i drugih) mogućnosti, Nmap je velika pomoć pri mapiranju mreže, kako će biti prikazano u opisu praktičnog rada, gdje je korišten za otkrivanje i verifikaciju aktivnih stanica.

Mada tradicionalno komandno-linijski alat, Nmap je s vremenom imao više grafičkih sučelja dostupnih na Internetu. U zadnjih par godina se ipak jedno istaknulo – Zenmap – i razvilo više od ostalih te je danas "službeno" grafičko sučelje i dostupno je na stranicama Nmapa. Primjer je vidljiv na slici 4.1.

Sučelje nudi klasično upisivanje komande za pretragu u predviđeno polje, čime Nmap radi isto kao prije, odnosno korištenje "čarobnjaka" za izgradnju pretrage, gdje se pretraga gradi u koracima, korištenjem izbornika na prilično intuitivan način. Također, može se vidjeti rezultantna komanda koju izgradi čarobnjak. Rezultati skeniranja mogu se grupirati prema pronađenim stanicama ili servisima i mogu se sačuvati. Grafičko sučelje, dakle, ne nudi nove funkcionalnosti u skeniranju, ali pregled rezultata je puno ugodniji i rezultati se elegantno mogu čuvati. Također, Zenmap nudi izradu (i par predinstaliranih) profila koji se koriste kao predlošci za tipične potrebe skeniranja, poput otkrivanja operacijskog sustava ili brzog otkrivanja stanica.

Osim grafičkog sučelja, Nmap je krajem 2006. godine uveo i dugo očekivanu mogućnost skriptiranja. Trenutno je opremljen podsustavom za tumačenje skripti pisanih u programskom jeziku Lua (podsustav se zove *Nmap Scripting Engine*) i s time si je Nmap otvorio vrata u svijet sigurnosnih skenera, kakvi su opisani kasnije.



Slika 4.1. Zenmap grafičko sučelje

U kojoj mjeri će Nmap doista biti konkurencija alatima poput Nessusa, vrijeme će pokazati, ali svakako će preduvjet biti postojanje mnogih skripti koje provjeravaju mnoge poznate ranjivosti.

4.2 Nessus

Nessus spada u sigurnosne analizatore (*Vulnerability Analyzer*) i jedan je od najpopularnijih programa te vrste. Dostupan je za sljedeće operacijske sustave: Linux, FreeBSD, Solaris, Mac OS X 10.4 i 10.5 i Microsoft Windows. Radi se o alatu čija je osnovna namjena automatizacija procesa otkrivanja poznatih sigurnosnih propusta pri analizi ranjivosti. Postoji već duži niz godina i trenutno je aktualna verzija 3, točnije 3.2.1.1 za MS Windows. Kroz to vrijeme, točnije na prijelazu između verzija 2.x u verzije 3.x promijenjena je licenca pod kojom se Nessus izdaje. Do verzije 3.x Nessus je bio potpuno otvorenog izvornog teksta, što više nije. Ipak, i od verzije 3.x se može besplatno skinuti i koristiti za osobnu uporabu, kao i za potrebe edukacije. Uz nadoplatu, koja je prema licenci obvezna za svaku komercijalnu uporabu, dolaze neke pogodnosti o kojima će nešto riječi biti kasnije. Tvrtka Tenable, koja razvija Nessus, paralelno s razvojem verzije 3, i dalje nudi podršku za otvorenu verziju 2 te je verzija 2 i dalje dostupna za skidanje s Interneta. Pri instalaciji verzije 3, korisnik mora pristati na određene uvjete korištenja, odnosno prihvatiti aktualnu Nessus licencu, dostupnu na stranicama proizvođača [13]. Nessus je u treunutku pisanja bio dostupan svim korisnicima, osim stanovnicima Kube, Irana, Sjeverne Koreje, Sudana i Sirije.

4.2.1 Arhitektura Nessusa

Nessus je fleksibilan sustav za procjenu ranjivosti čiji dizajn dozvoljava elegantno praćenje smjernica iz drugog poglavlja ovog rada. Izveden je u klijent-poslužitelj arhitekturi u kojoj poslužitelj obavlja sve poslove skeniranja koje zahtijevamo preko klijenta. Ovaj pristup ne uvodi poteškoće u radu, izuzev eventualno potrebe za nešto pažljivijom instalacijom, nego dapače – donosi niz pogodnosti. Sama arhitektura dopušta mogućnost postavljanja proizvoljnog broja poslužitelja na strateške lokacije unutar/izvan ciljne mreže i istovremeno upravljanje svakim od njih pomoću klijenta, kojih opet može biti više i na različitim lokacijama. Ovo dopušta prilagodbu lokacije skeniranja svakom mogućem zahtjevu.

Primjerice, pretpostavimo da tvrtka ima određen broj javnih adresa na Internetu i vlastite, privatne podmreže s adresama oblika 192.168.x.y/24. Prvih 24 bita označava adresu mreže, a kako je svaki segment 8-bitni pozitivan broj, mijenjanjem polja označenog s "x" u granicama od 0 do 255 dobivamo 256 mogućih podmreža, od kojih na svakoj mogu postojati 254 stanice. Neka je takvih podmreža svega nekoliko i brzo prelazimo tisuću mogućih adresa na kojima mogu postojati aktivne stanice. Kod većih organizacija, poput fakulteta, sveučilišta ili velikih korporacija, doista se lako može raditi o velikom broju stanica. Klijent-poslužitelj arhitektura Nessusa dopušta, primjerice, postavljanje po jednog poslužitelja za svaku podmrežu od 254 stanice, a uz dodatnu podršku se skeniranje elegantno može odvijati paralelno, s tim da se kasnije izvještaj gradi agregatno i može promatrati kao cjelina. Ovu mogućnost (između ostalog) Tenable nudi kroz isključivo komercijalni proizvod Security Center, no ništa nas ne sprečava da sami instaliramo veći broj skenera, pokrećemo ih po volji, a izvještaje kasnije smisleno spojimo.

Vrijedi još napomenuti da ovakva arhitektura dopušta korištenje (ili pisanje) drugih klijenata osim onog koji dolazi u paketu. Na Internetu je dostupno više takvih programa, od kojih se mali broj njih ističe kvalitetom. Jedan takav zove se Nessconnect – vrlo dobar klijent, dapače, po nekim karakteristikama i bolji od originalnog koji nudi Tenable. Riječ je o besplatnom klijentu, ali u njegovom slučaju se ne može računati na trajnu dostupnost, odnosno njegov razvoj ovisi o dobroj volji autora ili neke druge osobe, koja eventualno preuzme projekt.

4.2.2 Instalacijski detalji i način rada

Instalacija Nessusa na Windows računalu rezultira s pet programa koji čine paket. Dostupni su preko Start menija i oni su:

- Plugin Update – program za osvježavanje baze ranjivosti;
- User Management – program za upravljanje korisnicima Nessus skenera;
- Product Registration – program zas registraciju skenera;
- Nessus Server Configuration – program za osnovnu konfiguraciju poslužitelja;
- Nessus Client – klijentska aplikacija.

Plugin Update

Skeniranje i provjeru svih ranjivosti Nessus implementira kao zasebne module (ili "priključke", eng. *plugin*). Ti moduli su tekstualne datoteke koje sadrže skripte pisane u NASL skriptnom jeziku (eng. *Nessus Attack Scripting Language*), o kojem će više riječi biti kasnije, a primjer NASL skripte je dostupan u dodatku B. Sve poznate provjere (trenutno se radi o čak 23 497 provjera) moraju biti dostupne poslužiteljskoj komponenti i za to se brine program Plugin Update. Njegovim pokretanjem započinje ažuriranje baze ranjivosti.

User Management

Program za upravljanje korisnicima služi za dodavanje korisničkih računa koji će imati pristup skeneru. Tu se određuje korisničko ime i lozinka, odnosno certifikat – ovisno o odabranoj metodi autentifikacije korisnika. Vrijedi napomenuti smiješan propust: korisnički račun se može stvoriti i izbrisati, ali ne može modificirati. To ne predstavlja velik problem, jer postavljanje korisničkog računa traje doista kratko i nije problem ponoviti postupak ako se, primjerice, zaboravi lozinka.

Product Registration

Program za registraciju skenera prima registracijski kôd i obavlja registraciju skenera na strani proizvođača. Ovisno o modelu, ova registracija otključava ili zaključava neke mogućnosti Nessus skenera. Besplatna, kućna, odnosno edukacijska verzija je oslabljena, tako što joj nisu aktivne neke napredne mogućnosti – poput provjere usklađenosti stanice sa sigurnosnom politikom (eng. *policy compliance checks*) i nisu joj dostupne neke, strogo definirane grupe provjera, poput provjera namijenjenih kontroli SCADA sustava. Srećom, za elementarnu i umjereno složenu procjenu ranjivosti, ove mogućnosti nisu potrebne.

Nessus Server Configuration

Osnovna konfiguracija poslužitelja je isto jednostavna – riječ je o postavljanju IP adrese i pristupa na kojima će se poslužitelj odazivati i opcije da poslužitelj automatski svakih 24 sata ažurira bazu ranjivosti.

O klijentu će više riječi biti u nastavku, jer se kroz klijentsku aplikaciju zapravo konfigurira i pokreće proces skeniranja.

4.2.3 Baza znanja

Način rada skenera svodi se na tumačenje i izvođenje NASL skripti i prijavu rezultata koje skripte vraćaju nakon izvođenja. To vrijedi za sve Nessusove operacije, čak i za vlastitu konfiguraciju. To, u kombinaciji s izuzetno velikim (ukupnim i prosječnim) brojem provjera koje treba izvršiti, ukazuje na potencijalni problem s međuovisnostima pojedinih skripti.

Nessus taj problem vrlo efikasno rješava višestruko korisnom organizacijom određenih podataka u strukturu koja se zove "Knowledge Base", odnosno, prevedeno – u bazu znanja. U prvim verzijama Nessusa to je bila memorijska struktura građena dinamički tokom skeniranja, a kasnije je dodana mogućnost spremanja te strukture na disk i kasnijeg ponovnog korištenja. Osnovna namjena je spremanje informacija skupljenih tokom skeniranja, kako bi se njima mogli poslužiti drugi testovi, odnosno (otkada je uvedeno spremanje na disk) kako bi se njima moglo služiti neko kasnije skeniranje u budućnosti. Podaci su grupirani i organizirani u stabla, gdje svako stablo u korijenu ima naziv grupe i grana se u pojedine podgrupe koje u listovima drže vrijednosti. Podacima se lako pristupa preko ugrađenih NASL funkcija. Primjerice, ovo je segment iz skripte koja provjerava može li se Nessus spojiti na odredišnu stanicu (skripta 21745, "Local Checks Failed"):

```
if ( get_kb_item("HostLevelChecks/" + svc + "/failed") )
{...
```

Ovdje skripta dohvaća podatak iz baze znanja, ako takvog podatka ima i u ovisnosti o rezultatu se daljnje izvođenje grana. Ako rezultata nema, primjerice ako nema zapisa, funkcija se evaluira kao neistina. Suprotno, ako se u bazi znanja nađe ovakav zapis:

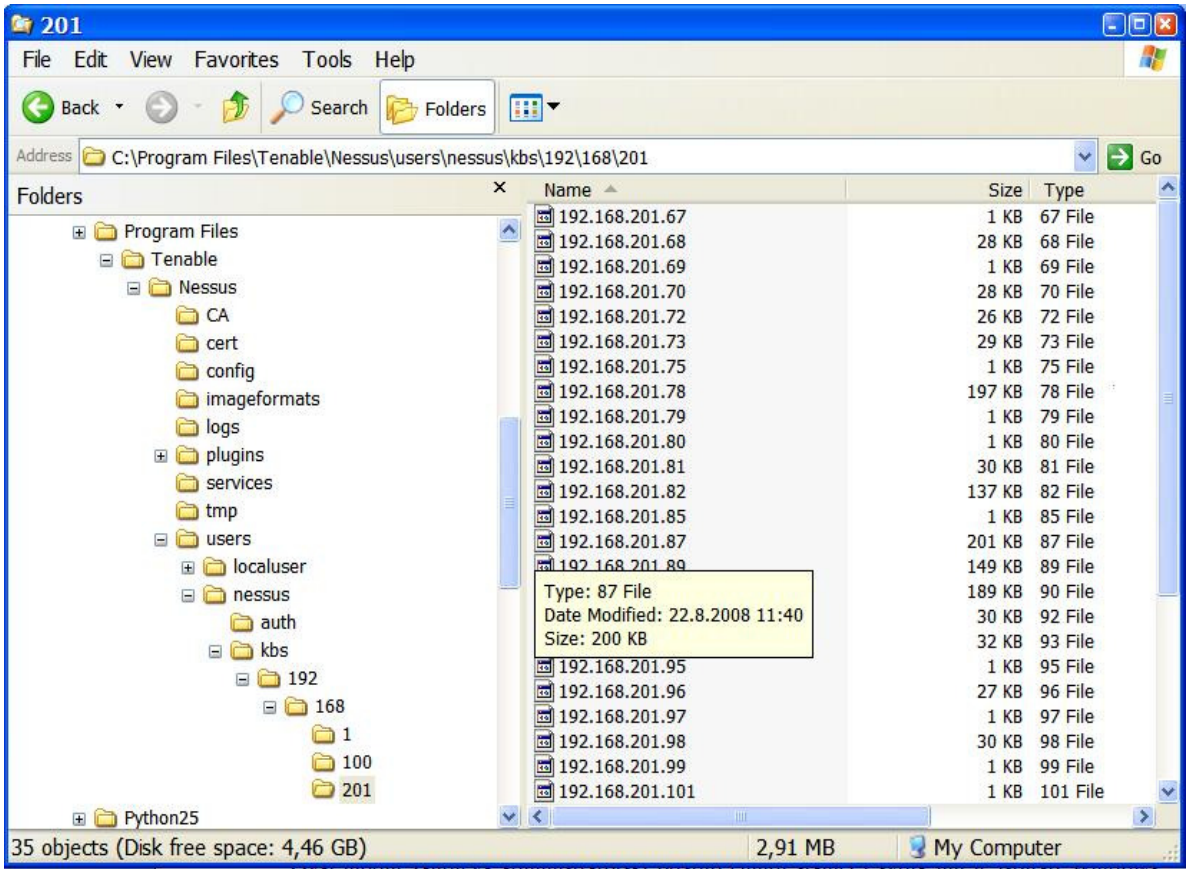
```
1219399239 3 HostLevelChecks/smb/failed=1
```

... tada se funkcija evaluira kao istinita, varijabla "`svc`" postavlja se na tekstualnu vrijednost "smb" i dalje u izvještaju se formatira u informaciju kako se Nessus nije uspio spojiti na stanicu koristeći predane Samba ovlasti (također, ovdje može pisati i "ssh", ako se za udaljeno spajanje koristi SSH protokol). Ostale brojke u zapisu iz primjera su dio internog formata zapisa u bazi znanja.

Primjer: modul 13855 – *Installed Windows Hotfixes*

Ovaj modul zahtijeva administratorski pristup ciljnoj stanici i svrha mu je ispitati Windows Registry o instaliranim Windows zakrpama. U realizaciji koja je omogućena pomoću *Knowledge Basea*, to se svodi na samo jedan upit i spremanje dobivenih podataka koji tako postaju dostupni svim zainteresiranim daljnjim upitima. Također, tada se zapiše i da se ovaj test odvio. Uz mehanizam međuovisnosti koji testovi imaju, svaka daljnja potreba za tim informacijama dobiva se prvo utvrđivanjem da su ti podaci dohvaćeni ranije, a potom čitanjem iz baze znanja. Ovo rješenje općenito omogućava pisanje testova koji troše manje resursa (u vidu procesorskog vremena i mrežnih kapaciteta).

Baza znanja, ako se piše na disk, zapravo se sastoji od tekstualnih datoteka koje se (u slučaju Windows instalacije) nalaze u zasebnom direktoriju korisnika koji je pokrenuo skeniranje, unutar glavnog direktorija instalacije. Primjer je (za korisničko ime "nessus") vidljiv na slici:



Slika 4.2. Smještaj baze znanja

Smještaj po direktorijima, kao što je vidljivo, organiziran je kao i segmenti IP adrese, a u zadnjem direktoriju se nalaze tekstualne datoteke imenovane prema punoj adresi stanice. Ove

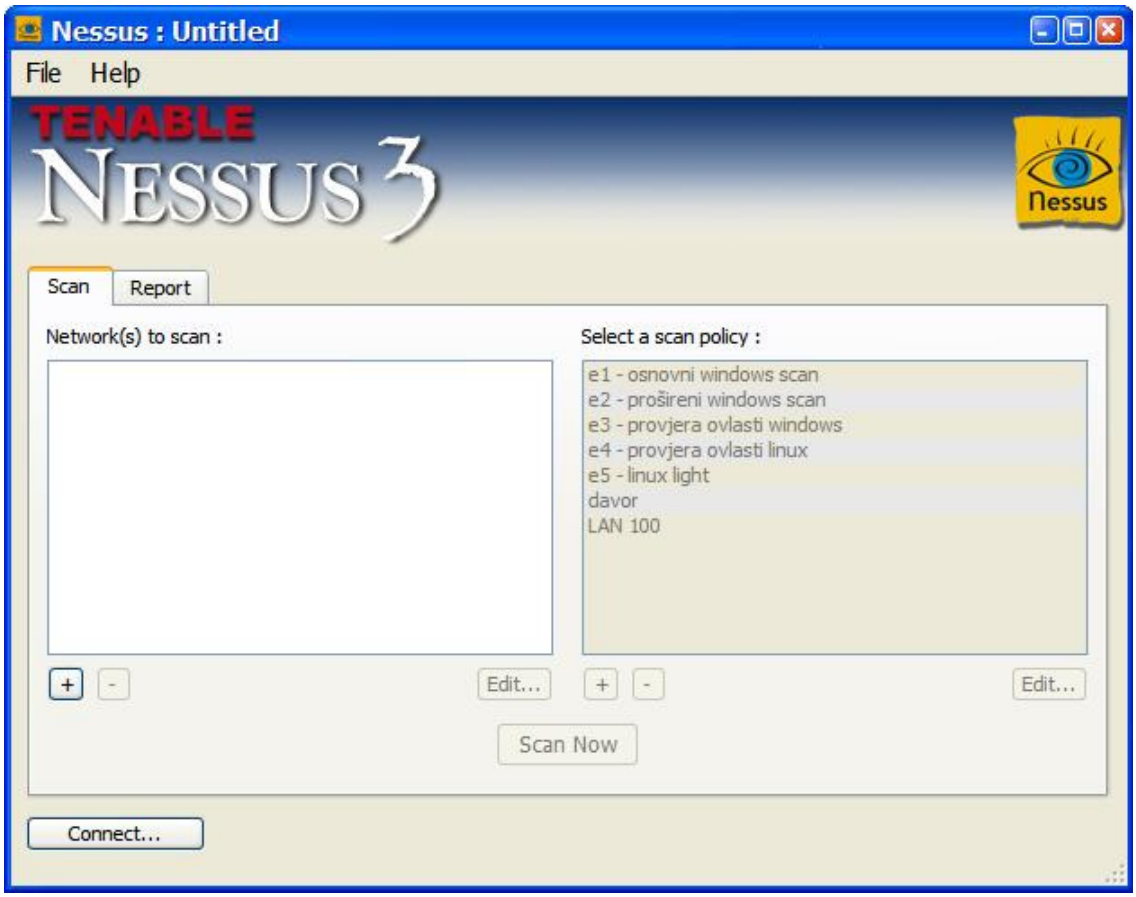
datoteke su od velike pomoći za analizu rada Nessusa kada se posumnja da isti ne radi dobro, ili kada nije jasno zašto neke stvari "ne napravi".

4.2.4 Klijent

Postavljanje i pokretanje politike skeniranja

Nessus klijent služi za definiciju sjednice skeniranja (eng. *scan session*) – izbor provjera pojedinih ranjivosti i određivanje parametara skeniranja, poput vrste skenera pristupa, broja paralelnih provjera i svih ostalih konfiguracijskih mogućnosti – sve se postavlja kroz klijent aplikaciju.

Inicijalno sučelje je dosta intuitivno, nudeći stranicu "Scan" za postavljanje i pokretanje skeniranja i stranicu "Report" za pregled izvještaja.



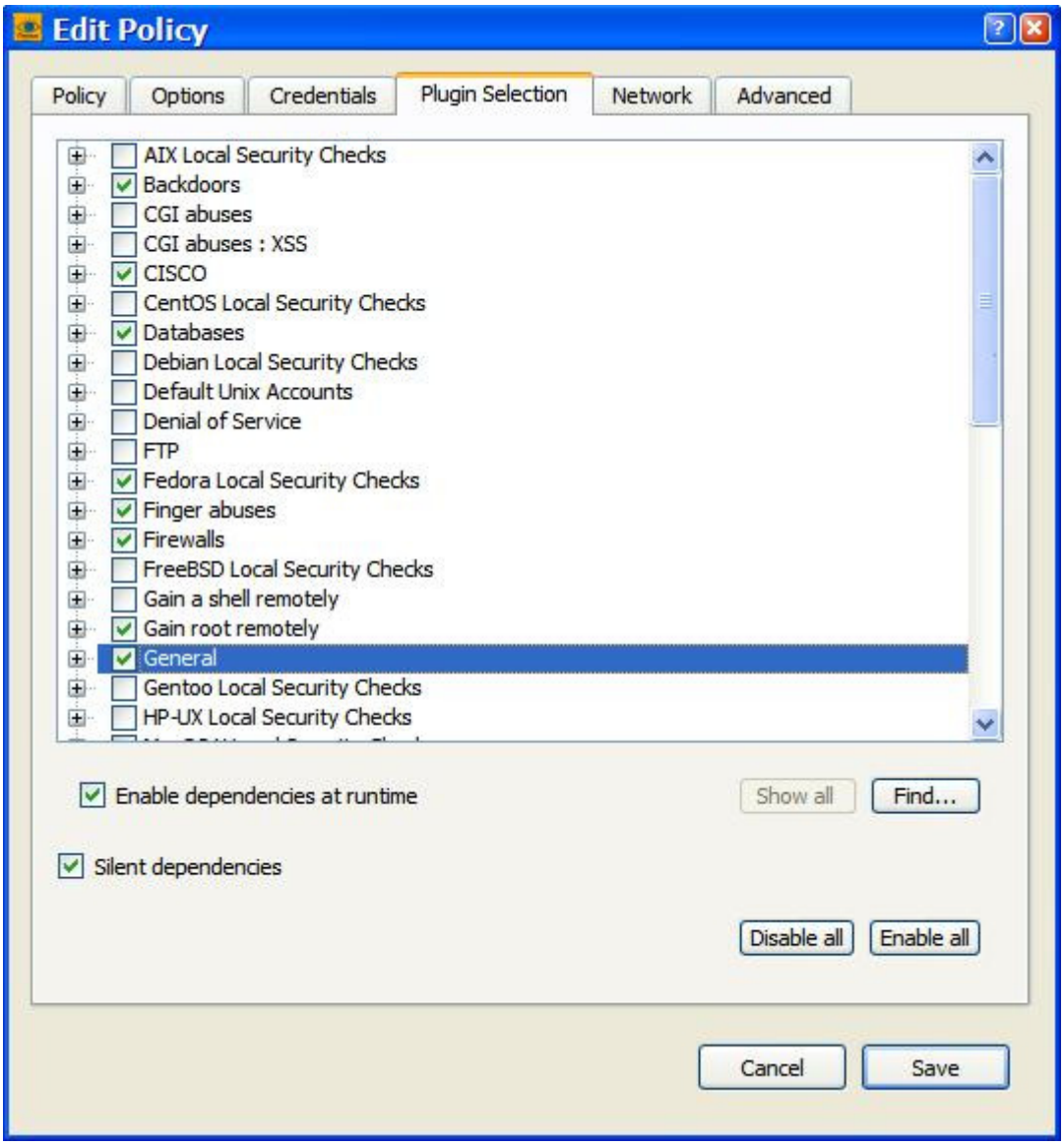
Slika 4.3. Nessusovo inicijalno sučelje

S lijeve strane se unose mreže ili zasebne stanice koje treba skenirati. Unos se započinje klikom na gumb sa znakom "+" u donjoj lijevoj strani i moguć je u više formata:

- Single host – unos jedne IP adrese ili DNS (ili NetBIOS) imena;
- IP Range – unosi se raspon kao početna i završna IP adresa ;
- Subnet – unosi se mreža i mrežna maska;
- Hosts in file – odabere se datoteka u kojoj su popisane IP adrese ili imena stanica.

Gumb "Connect" otvara novi ekran u kojem se odabire Nessus skener koji će se koristiti. Taj ekran, osim popisa definiranih poslužitelja, nudi mogućnosti za dodavanje, brisanje i izmjenu liste poslužitelja, kao i postavljanje SSL sloja u komunikaciji između poslužitelja i klijenta. Nakon spajanja na jedan od poslužitelja, omogući se izbor s popisa politika skeniranja na desnoj polovici prozora. Opet, moguće je modificirati i brisati postojeće te dodati nove politike. Nakon izbora politike, klikom na gumb "Scan Now" počinje skeniranje odabranih stanica. Dakako, najzanimljiviji dio priče je uređivanje politike skeniranja.

Ekran za uređivanje politike skeniranja sadrži više stranica, kako je vidljivo na slici 4.4.:



Slika 4.4. Uređivanje politike skeniranja - izbor testova

Stranice koje se nude su:

- **Policy** – ovdje se postavlja ime politike, njezina trajnost (jednokratna ili je treba sačuvati na skeneru), način spremanja lozinki za ovlasti i slobodni komentar.

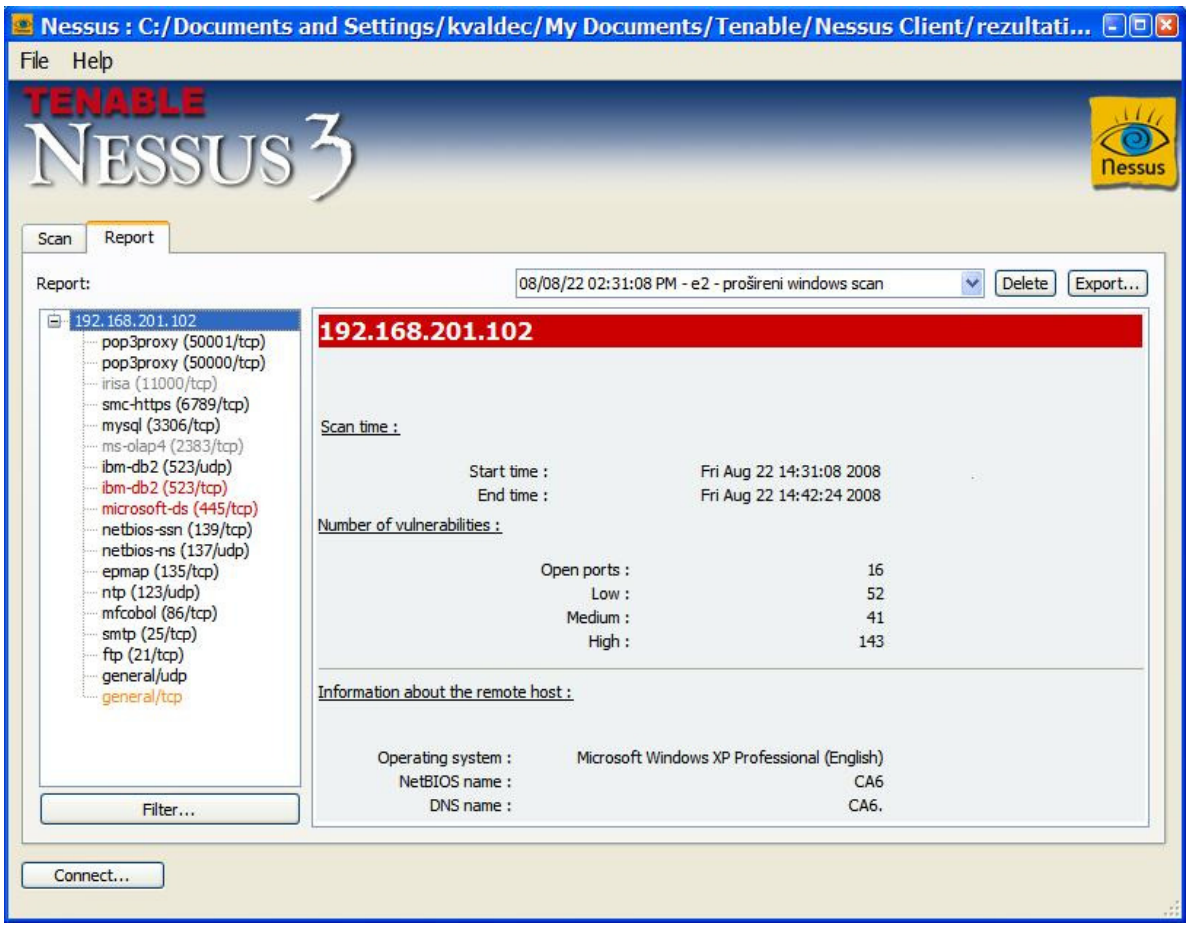
- **Options** – ovdje se postavljaju neki osnovni parametri skeniranja poput broja paralelnih testova, broj stanica koje će se skenirati paralelno, spremanje baze znanja (i drugih podataka o skeniranju) na disk, koje pristupe i koje skenere pristupa koristiti itd.
- **Credentials** – na ovoj stranici se upisuju korisnička imena i lozinke s kojima se skener može služiti kod skeniranja. Pritom, nude se sljedeći mehanizmi autentifikacije:
    - Windows credentials – za Windows mreže, npr. domenski korisnički račun;
    - SSH settings – postavke za prijavu preko SSH;
    - Oracle settings – specifično za korisnike Oracle proizvoda;
    - Kerberos configuration – podaci poput lokacije centra za distribuciju ključeva kod Kerberos autentifikacije;
    - Cleartext protocol settings – podaci za prijavu preko nesigurnih protokola poput telneta te rexec i rsh naredbi;
- **Plugin selection** – prikazano na slici 4.4. Na ovoj stranici se biraju testovi koji će se izvoditi. Oni su organizirani u grupe i mogu se odabrati svi testovi iz grupe, ili se ona može raširiti pa se izabiru samo pojedini. Kao što je vidljivo na slici, samih grupa ima dosta, a one mogu sadržavati dosta testova. Spomenutih skoro 23 500 testova su svi na ovom ekranu, unutar svojih grupa.
- **Network** – sadrži opcije za kontrolu potrošnje mrežnih resursa, poput broja paralelnih TCP sjednica i vremena koliko se čeka na odgovore na mrežnu aktivnost.
- **Advanced** – ova stranica se gradi dinamički, ovisno o odabranim testovima i sadrži konfiguracijske parametre za pojedine testove, ako su oni prisutni. Osim toga, sadrži i neke konstantne konfiguracijske sekcije, poput izbora treba li skenirati osjetljive naprave (npr. pisače). Osim toga, ovdje se nalaze korisnička imena i lozinke za pojedine servise, kao što su web, FTP i slični. Ovi se mogu razlikovati od korisničkih imena za pristup stanici (a ta imena i ne moraju biti dozvoljena) i oni će se koristiti kako bi se sustav probao prijaviti na odgovarajući servis (HTTP, NNTP, POP2, POP3, IMAP i FTP)

Pregled rezultata

Kada je sve postavljeno i kada su odabrani svi testovi, politika se sačuva klikom na gumb "Save" i pokrene. Po završetku skeniranja, na stranici "Report" se pojavi izvještaj, grupiran prema IP adresama skeniranih resursa. Izvještaj nudi relativno slabe mogućnosti filtriranja prema tekstu unutar rezultata ili naziva, prema identifikacijskom broju skripte koja je možda javila kakav rezultat te prema razini kritičnosti. Postoje tri razine kritičnosti:

- High – ovo se smatra sigurnosnom rupom i sustav se smatra izuzetno ranjivim ako se među rezultatima nađu ovakve ranjivosti.
- Medium – većinom se radi o upozorenjima i ranjivostima koje teško mogu eskalirati u velik problem.
- Low – ovdje se zapravo ne radi o ranjivostima, već o raznim informacijama koje mogu i ne moraju biti važne, a koje je skener našao.

Primjer rezultata vidljiv je na slici 4.5.:



Slika 4.5. Rezultati skeniranja, osnovni prikaz

Ovaj primjer (preuzet iz praktičnog rada opisanog kasnije) prikazuje, primjerice, izuzetno ranjivo računalo, sa čak 143 uočene ranjivosti visokog prioriteta! Lijeva strana prikazuje sve otvorene pristupe koje je skener pronašao, odnosno servise koji se nude na određenim vratima. Njihova boja govori kakve su ranjivosti pronađene u pojedinoj grupi pa crvena predstavlja visoko rizične, žuta srednje rizične, a crna označava samo informativne zapise. Siva predstavlja otvorene pristupe za koje nije uočena baš nikakva dodatna informacija. Klikom na pojedinu grupu, u desnoj strani prikaže se niz informacija vezan uz odabrani pristup, a klik na IP adresu prikazuje sažetak za odabranu adresu (tako je na slici 4.5.).

Svi izvještaji iz jedne sjednice dostupni su iz menija lijevo od gumba "Delete" – koji odabrani izvještaj briše, dok ga gumb "Export" izvozi u html formatu, odnosno u formatima "NBE" i (napuštenom) "NSR", koji su specifični za Nessus [8]. Primjer i objašnjenje NBE formata dostupan je na kraju ovog rada, u dodatku C.

Html format izvještaja je lijepo formatirana web stranica koja daje kvalitetan pregled pronađenih ranjivosti i za određenu klasu stanica je posve dovoljan. Za kompleksnije potrebe je pogodan format "nbe", za kojeg je lako napraviti vlastiti parser i generator izvještaja.

4.2.5 Zaključno o Nessusu

Svakako se radi o vrlo moćnom alatu, koji se odlikuje fleksibilnošću i prilagodljiv je mnogim primjenama. Osim vrlo pogodne poslužitelj-klijent arhitekture i općenito visoke kvalitete izvedbe, njegovo današnje stanje rezultat je dvaju čimbenika:

1. nekada je bio otvorenog izvornog teksta,
2. danas je dio palete proizvoda tvrtke Tenable,

Prvi faktor uzrokovao je napredak Nessusa od dobre ideje do najkorištenijeg sigurnosnog analizatora, koji na tom području dominira već godinama i smatra se *de facto* standardom. Nessus je tako imao široku bazu korisnika koji su doista doprinijeli njegovom razvoju kroz sugestije i dojave grešaka. NASL, razvijen isključivo za Nessus, postao je dosta bitan jezik i poznavanje rada s njim postalo je nešto čime se drugi komercijalni alati hvale u vlastitom marketingu, ako ga podržavaju – što je često slučaj. NASL otvara mogućnost neograničenog prilagođavanja rada Nessusa vlastitim potrebama, kroz pisanje vlastitih skripti, ili naručivanje skripti od specijaliziranih tvrtki.

Drugi faktor donekle je usmjerio razvoj Nessus poslužitelja u modul koji se uklapa u rad drugih programa iz ponude tvrtke Tenable. Primjerice, "Security Center" je Tenableov komercijalni proizvod koji služi za kvalitetno centralizirano upravljanje većim brojem Nessus skenera i agregiranje njihovih izvještaja u kvalitetne višerazinske dokumente. Takvo nastojanje utjecalo je na razvoj, stabilizaciju i standardizaciju izlaznih formata dokumenata i općenito standardiziralo ulaznu i izlaznu stranu poslužiteljske komponente.

S druge strane, Nessusu se može prigovoriti na Windows klijentsku aplikaciju, koja ima neke nedostatke. Osnovni nedostatak je rad s izvještajima, kojima ne bi škodile veće mogućnosti filtriranja, grupiranja i sortiranja. No, to je politička odluka i te su funkcionalnosti smještene (i naplaćene) u Security Center. Osim toga, Windows klijent i poslužitelj ne nude mogućnost odvojenog skeniranja, u kojem bi se klijent nakon pokretanja mogao odspojiti i pokupiti gotove rezultate kasnije. Skeniranje je ovako neodvojivo od klijenta, odnosno klijent mora biti spojen na server cijelo vrijeme. Za pokretanje druge sjednice potrebno je pokrenuti novi klijent, jer za vrijeme skeniranja nije moguće podešavati i pokretati druge zadatke.

4.3 Retina

Retina je sigurnosni analizator tvrtke eEye, poznate po istraživanjima na području računalne i mrežne sigurnosti [14]. Za razliku od Nessusa, nije nikada bila otvorenog izvornog teksta (i to je općenito vidljivo u radu s programom). Pisana je za Windows operacijski sustav pa poslužiteljski dio više nalikuje Windows servisu, kako je vidljivo na slici 4.6, a klijentski dio aplikacije više nalikuje standardnom MS Windows programu, kako je vidljivo na slici 4.7.

4.3.1 Arhitektura Retine

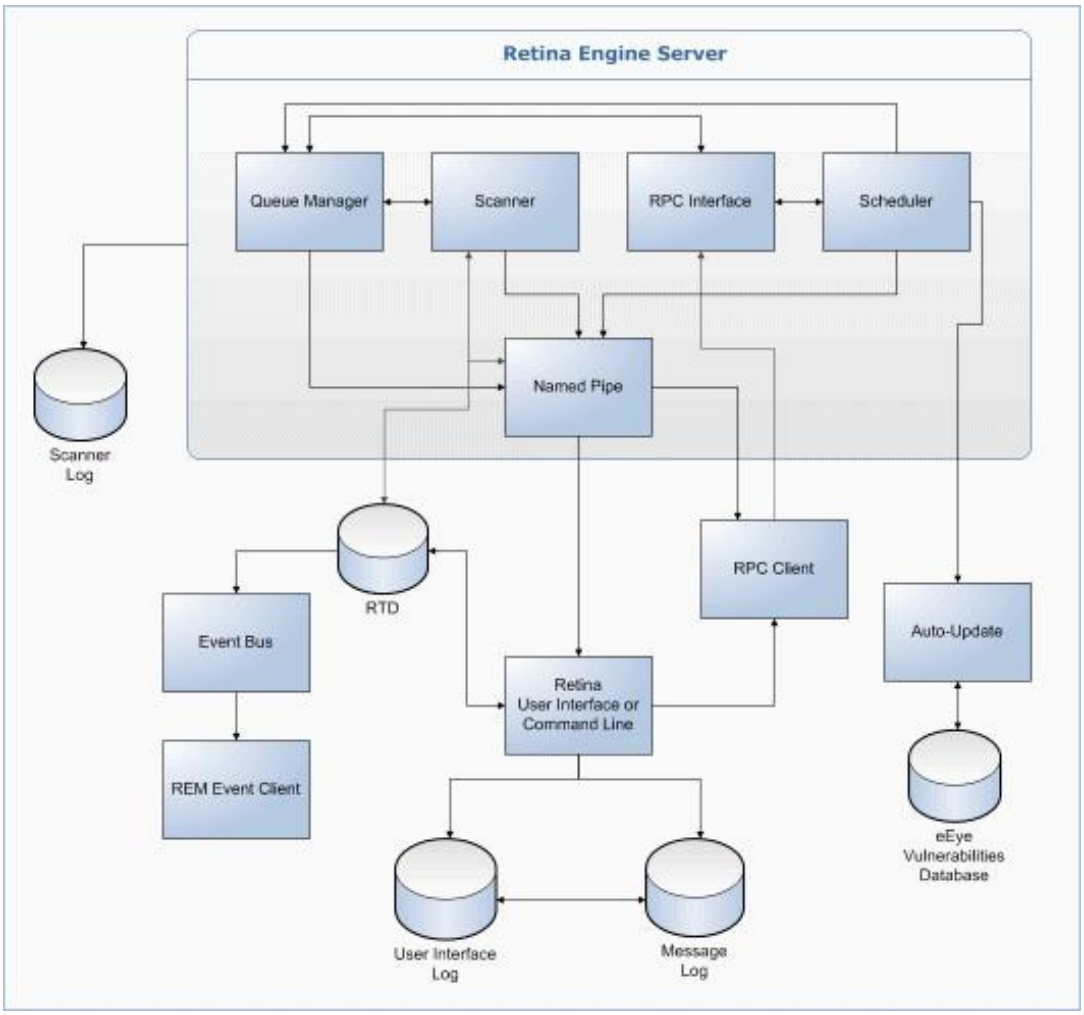
Retina, slično kao i Nessus, se u ulozi skenera uklapa u veće "rješenje". Instalacijski paket sadrži sljedeće dijelove:

- **Retina servis ("retinaengine.exe")** – Windows servis koji se interno sastoji od odvojenih podsustava:
    - **skener** – obavlja poslove skeniranja; čita/piše podatke u podsustav za upravljanje redom provjera i bazu rezultata testova (funkcionalno nalik bazi znanja iz Nessusa).
    - **upravljač reda provjera** – kontrolira redoslijed provjera.
    - **raspoređivač poslova** – kontrolira redoslijed poslova skeniranja, tj. upravlja radom jedne sjednice.

- **RPC sučelje** – komunikacijsko-kontrolno sučelje prema drugim programima.
- **imenovani cjevovod** – za potrebe međusobne komunikacije klijentskog dijela, skenera i baze rezultata.

- **Korisničko sučelje ("retina.exe")** – Windows klijentska aplikacija. Namjena joj je lokalni pristup skeneru i bazi ranjivosti te generiranje izvještaja.
- **RPC klijent ("RetRPC_Client.exe") –** obavlja poslove komuniciranja podataka između RPC sučelja i korisničkog grafičkog sučelja ili drugih aplikacija.
- **Aplikacijska sabirnica –** podatkovni kanal za informacije i kontrolne podatke prema i iz skenera. REM Event Client odavdje čita podatke iz baze rezultata testova.
- **REM Event Client –** Ako je Retina integrirana u REM liniju rješenja, onda ovaj klijent proslijeđuje podatke centraliziranom poslužitelju koji prati sve događaje.
- **Program za nadogradnju programske podrške –** služi za automatizaciju nadogradnje.

Povezanost tih dijelova vidljiva je na slici 4.6.:

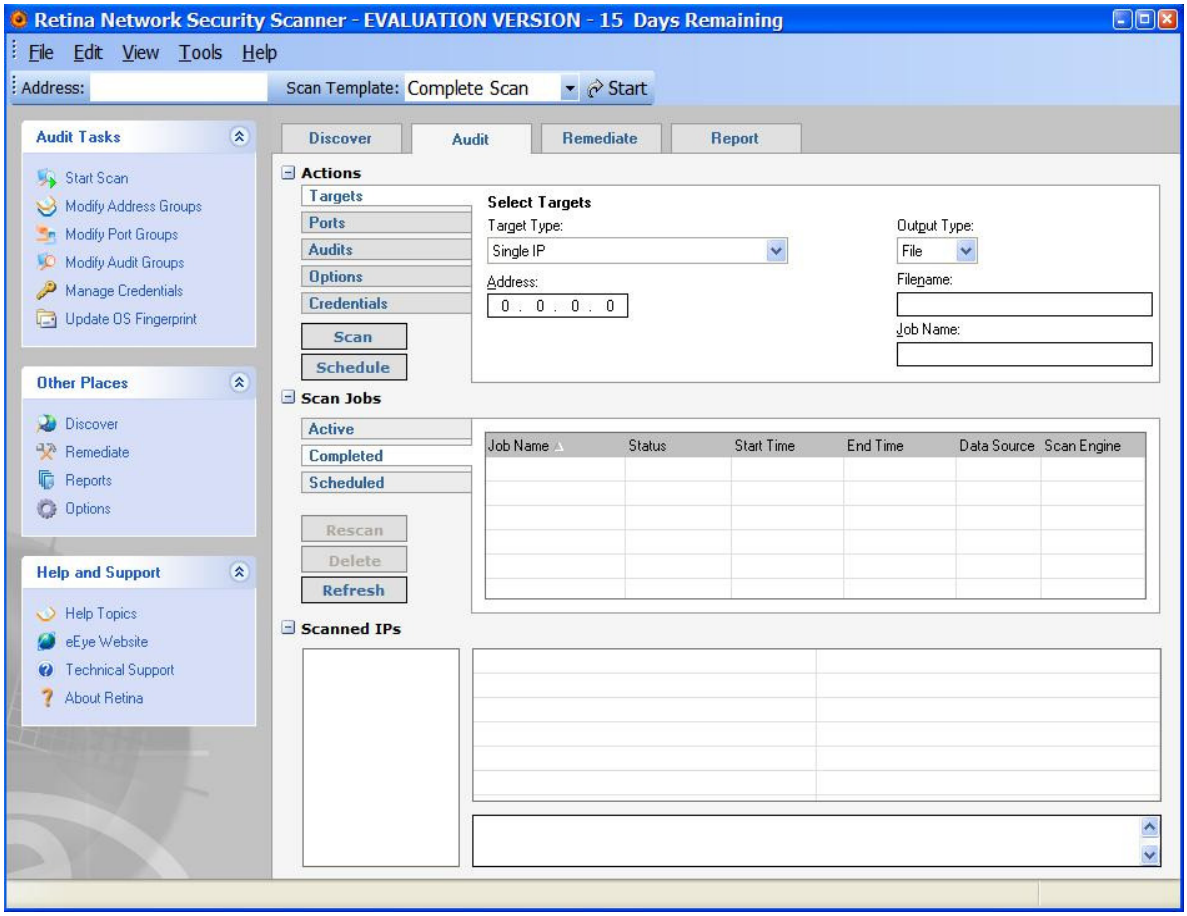


Slika 4.6. Interna organizacija Retina skenera

4.3.2 Sučelje

U osnovi, Retina nudi vrlo slične funkcionalnosti kao i Nessus, ali nešto ljepše organizirane. Proces procjene ranjivosti podijeljen je na logičke cjeline koje su prikazane na pojedinim stranicama unutar ekrana:

- **Discover** – ovdje su opcije za otkrivanje aktivnih stanica. Imenovanje i deklaracija stanica ili mreža koje treba skenirati moguća je istim zapisima kao u Nessusu, uz dodatak CIDR notacije, što je pohvalno.
- **Audit** – ovdje se postavljaju specifičnosti procjene ranjivosti i ova stranica odgovara izboru testova u Nessusu. Stanice se opet unose, a mogu se koristiti one pronađene u prethodnom koraku. Tu se nude opcije izbora pristupa na kojima će se tražiti servisi. Testovi se biraju na podstranici "Audits" i to u obliku grupa, a s instalacijom dolazi par predinstaliranih grupa, poput testova na glavnih 20 ranjivosti koje je prijavio SANS i slično. Dakako, mogu se definirati vlastite grupe. Sve navedeno je moguće i u Nessusu, koji dolazi s manje predefiniranih grupa. Također, na ovoj stranici je pregled poslova (sekcija "Scan Jobs", vidljivo na slici 4.6.), gdje se tokom skeniranja ili otkrivanja stanica može vidjeti napredak.
- **Remediate** – ova stranica nudi izradu izvještaja pogodnih za daljnju reakciju na rezultate.
- **Report** – na ovoj stranici se upravlja procesom izrade izvještaja.



Slika 4.7. Retina - inicijalni ekran

S lijeve strane su grupe prečaca, čiji sadržaj ovisi o stranici na kojoj se nalazimo i pomoću kojih se Retinom može upravljati nešto ergonomičnije i brže.

Retina ima nešto kvalitetnije sučelje za izbor testova. Za svaki test koji se odabire se može prilikom odabira pročitati neke detalje, dok je kod Nessusa takvo nešto nevidljivo, ako se koristi Tenableov Windows klijent (spomenuti Nessconnect također prikazuje informacije, i to one koje pišu u opisnim dijelovima skripti, što je najbolje rješenje). Ovo, mora se napomenuti, i nije veliki plus, odnosno nije veliki nedostatak Nessusa. Pri učestalom radu je doista dovoljan naziv testa i njegov smještaj unutar neke grupe, a ovakav opis je pogodan u periodu učenja pa samo na prvi pogled izgleda kao velika prednost Retine. Stranice vezane uz izradu izvještaja također idu u prilog Retini, u odnosu na Nessus.

Izvještaji koje Retina nudi u demo verziji nude više mogućnosti grupiranja i sortiranja te izvoz u Microsoft Word, pri čemu je čak vođeno računa o prijelomu stranica! Dodatno, svaka odvojena sekcija može imati predviđeni prostor za komentare, što je u ovoliko raznolikom procesu kao što je procjena ranjivosti izuzetno korisno. Također, među izlaznim formatima nudi se i XML, koji je puno raširenji standard za opis podataka od Nessusovog "nbe" formata.

4.4 Usporedba Nessusa i Retine

Na području sučelja i izvještaja, Retina odnosi pobjedu. No, to je rezultat politike i taktike natjecanja između dvije kompanije. Dio funkcionalnosti koje donose prevagu na stranu Retine su u rješenju koje nudi Tenable naprosto postavljene drugdje, u druge aplikacije koje Tenable prodaje. Tako je Retina naoko primamljivija, no treba imati na umu da oba rješenja, kada se plate, nude vrlo sličan skup funkcionalnosti. Nessus, s druge strane, nudi puno više u besplatnoj verziji – koju Retina zapravo i nema, već ima demo verziju s rokom trajanja od 15 dana.

Na području pouzdanosti i preciznosti, oba alata su višestruko nagrađivana i spadaju u vrh ponude. Neki testovi sugeriraju da je Retina nešto preciznija, drugi testovi sugeriraju isto za Nessus, a istina je, po običaju, u sredini. Veću razliku pokazuju u brzini rada s podrazumijevanim postavkama, gdje se Nessus pokazao nešto brži, za približno isti izbor testova. Izbor testova je dosta teško uskladiti jer su grupirani na različit način pa preciznost i brzina teško dolaze do kvalitetne usporedbe. No, kao veći faktor za obavljanje kvalitetne procjene ranjivosti, pokazali su se propusnost mreže i strogost sigurnosnih zaštitnih stijena preko kojih se prelazi pa i na ovom području zapravo nema prevage.

Na području potrošnje resursa, oba skenera su dosta skloni potrošnji. Razlike postoje i, prema viđenom, u korist Nessusa su (procesorsko vrijeme i memorija), odnosno približno su jednaki za potrošnju mrežne propusnosti. No, ovo ne bi smio biti faktor koji utječe na odluku. Nabava ili odabir snažnijeg poslužitelja je bolje rješenje. Slično razmatranje vrijedi i za cijenu, gdje Tenable trenutno ima pogodnije modele od eEye, nudeći relativno jeftine licence po skeneru. Za 1200 američkih dolara godišnje se dobije kompletna licenca za Nessus skener, koji se onda može koristiti u potpunosti, odnosno programska podrška i dostupne ranjivosti su bez ograničenja. Cijene Retine kreću od 575 američkih dolara godišnje, no za tu cijenu skener može skenirati samo 32 IP adrese. Maksimalna podmreža koju može skenirati je 256 IP adresa i takav paket košta 1650 američkih dolara, no za takve veličine podmreža Retina se zna ponašati nepouzdano pa je i povrat uloženog novca ponekad nepouzdan.

Smatram da pravu prevagu, i to u korist Nessusa, donosi jezik NASL. Opredjeljenje za Tenableovo rješenje bazirano na Nessusu otvara mogućnost pisanja vlastitih testova u obliku dobro dokumentiranih skripti. Edukacija za pisanje skripti je donekle investicija u obliku vremena i novca, ali (kao i svako znanje) vrlo je brzo isplativa. Ekvivalent koji Retina nudi s

Enterprise licencom je nešto lakši za korištenje (izgradnja provjere pomoću čarobnjaka), ali je u odnosu na skriptni jezik znatno ograničen. Dodatno, postojanje ostalih provjera, eksplicitno navođenje međuovisnosti i baze znanja kod Nessusa omogućava kontrolirano povezivanje i korištenje rezultata drugih testova.

4.5 Zaključno o alatima

Opisani alati općenito su vrlo snažni, kvalitetni i fleksibilni. Jako dobro su prilagodljivi opisanoj metodologiji i oba imaju predviđenu tehničku mogućnost uklapanja u šire rješenje. Vrijedi naglasiti važnost upravo te točke. Naime, opisana metodologija je doista vrlo općenita i dobro pokriva potrebe u većini slučajeva na tržištu. No, kod najvećih organizacija, alate najčešće treba doraditi ili uklopiti u neki postojeći sustav. Taj posao mogu odraditi zaposlenici same organizacije, u dogovoru s tvrtkama Tenable ili eEye, ili se taj posao može od njih (ili trećih strana) dodatno naručiti. U svakom slučaju, mogućnost uklapanja napisanih skenera u željene nadogradnje predstavlja velik plus u ocjeni i izboru alata.

Za veće kompanije navedeni alati nisu dostatni. Procjena ranjivosti u takvim organizacijama dio je veće sigurnosne politike i dolazi u obzir isključivo u kombinaciji sa centraliziranim upravljanjem podacima. Osim toga, generiranje izvještaja mora biti kvalitetnije i raznovrsnije, uz puno veće mogućnosti prilagodbe organizaciji, poput umetanja vlastitog "*brandinga*" i ostalih detalja.

Primjer: ***Tenable Log Correlation Engine***

Vezano uz uklapanje procjene ranjivosti u širu sigurnosnu politiku vrijedi spomenuti i Tenableov najskuplji proizvod – Log Correlation Engine (LCE), odnosno alat za korelaciju zapisa. Dostupan je samo kao nadogradnja na već kupljeni Tenable Security Center, program za centraliziranu procjenu ranjivosti, i jedna licenca trenutno košta čak 50 000 američkih dolara.

Osnovna namjena alata je sakupljanje, normalizacija i povezivanje kontrolnih zapisa (eng. *log entries*) sa svih zamislivih uređaja na mreži, od sustava za otkrivanje napada, preko lažnih sustava za privlačenje napadača (eng. *honeypot*) i sigurnosnih zaštitnih stijena, pa do zapisa sa svih Tenableovih programa. Ključna riječ je povezivanje, odnosno korelacija zapisa, čime procjena ranjivosti zapravo prelazi svoje granice. Osim prepoznavanja prometa koji bi mogao biti sumnjiv, LCE može reagirati na statističke promjene u uobičajenom prometu. Također, LCE se može skriptirati u jeziku TASL – Tenable Application Scripting Language, čime se postiže velika kontrola nad izuzetno velikom količinom istovremenih događaja.

Dobar primjer je korelacija rezultata skeniranja i rezultata koje proizvede IDS. Ako se sustav nađe pod napadom i IDS to prepozna, može doći do reakcije, ovisno o konfiguraciji IDS-a. No, ako je sustav zaštićen od prepoznate vrste napada, reakcija IDS-a je suvišna jer sustav zapravo nije ugrožen. LCE to može prepoznati i naglasiti, kako se ne bi gubilo vrijeme na istraživanje opasnosti od tekućeg napada.

Sljedeće poglavlje daje prikaz izvedbe procjene ranjivosti na aktivnoj, produkcijskoj, heterogenoj mreži.

5. Izvedba procjene ranjivosti mrežnog sustava

U ovom poglavlju bit će prikazan primjer praktične izvedbe procjene ranjivosti, prema naputcima i specifičnostima navedenih alata iz ranijih poglavlja. Vrijedi, pritom, napomenuti par ograničenja uz koja se praktični dio ovog rada izvršavao.

Kao prvo, procjena ranjivosti izvršavala se uz određena "akademska" ograničenja i nije bilo potpisivanja nikakvog ugovora. Procjena ranjivosti vršena je na konkretnoj produkcijskoj mreži postojeće firme i uz svu odgovarajuću pažnju koju takva mreža zahtijeva. Pritom, faza ugovaranja procjene dobro je aproksimirana fazom dogovaranja, odnosno svi važni koraci su doista obavljeni, osim specificiranja aktivnosti ugovorima i potpisivanja dokumenata o neotkrivanju poslovnih tajni, što je podrazumijevano i dogovoreno usmeno.

Drugi čimbenik je ograničavanje skeniranja i o tome će biti riječi kasnije tokom poglavlja. Mreže koje su bile predmet skeniranja su, radi konkretne organizacije tvrtke, dio dijeljene infrastrukture pa je ograničavanje u određenoj mjeri ostalo na logičnoj razini, u smislu da su neki testovi možda opteretili mrežnu opremu dijela organizacije koji nije bio unutar plana skeniranja, i to isključivo iz tehničkih razloga poput dijeljenja adresnog prostora razašiljanja i slično.

Organizacija u kojoj je rađena procjena ranjivosti zapravo je grupa koja se sastoji od više firmi koje samostalno posluju unutar nje. Iz toga proizlaze neke činjenice koje su utjecale na cijeli proces, od kojih treba izdvojiti sljedeće:

- mrežnom infrastrukturom se upravlja dijelom na razini grupe i dijelom na razini pojedine tvrtke; ona je dijeljena među tvrtkama i dijelovi izvan ingerencije jedne od tvrtki nisu smjeli biti svjesni skeniranja;
- potrebno je paziti na tajnost pronađenih podataka;
- većinu testova treba izvršiti na način i u vrijeme kada nikome ne smetaju.

Sve kritične odluke morale su biti u skladu s tim naputcima.

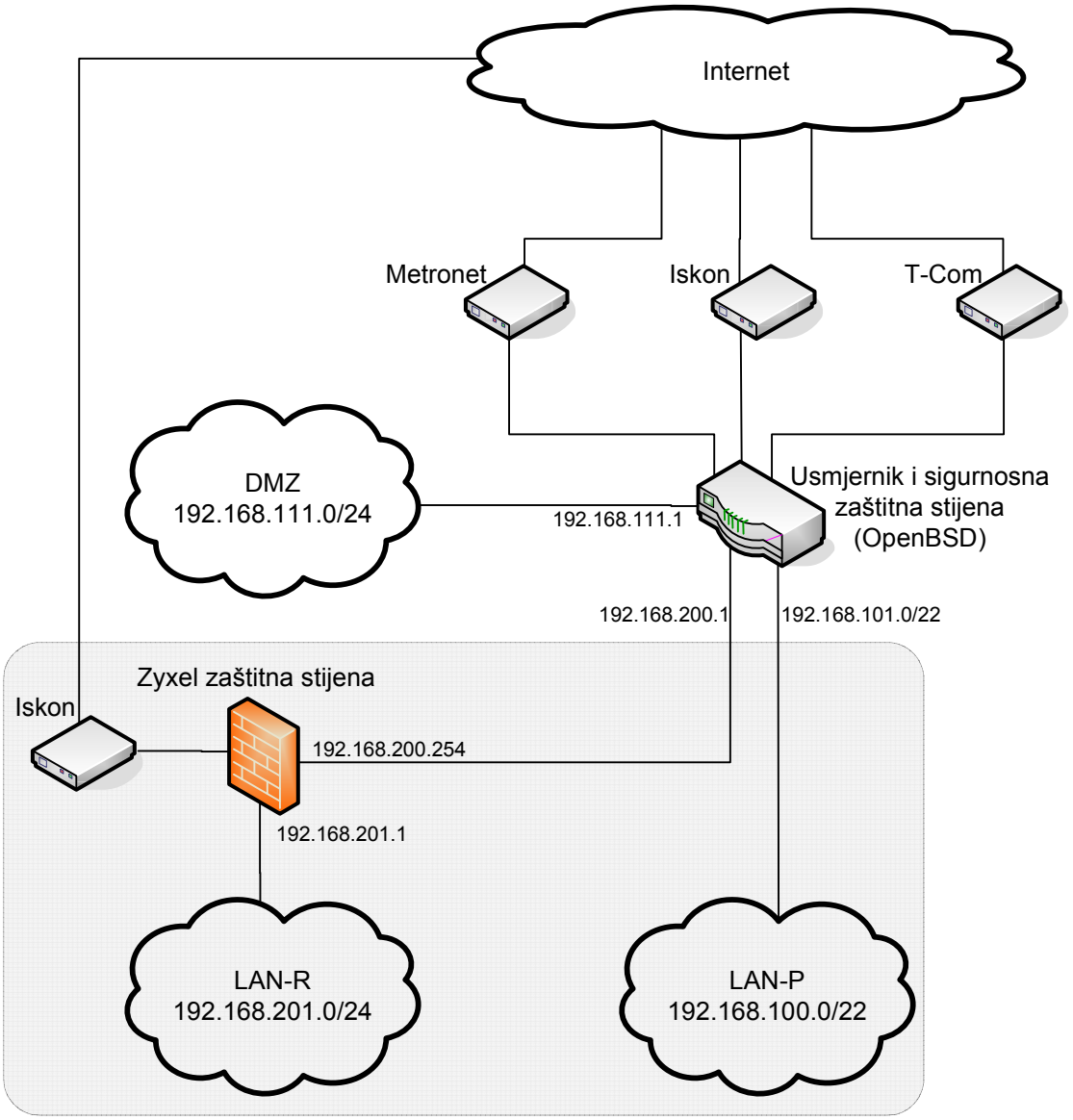
5.1 Mapiranje mreže

Firma o kojoj je riječ je, prema vrsti usluga koju nude, programerska i bavi se izradom poslovnih, pretežito bankarskih aplikacija. Iz ovoga proizlazi da ima nekakvu razvojnu grupu (ili odjel), i to je na kraju predstavljalo posebnu grupu računala, a unutar nje se na stanicama zato mogla očekivati određena vrsta programske opreme, poput baza podataka ili razvojnih okolina za korištene jezike. Osim toga, pod ingerencijom te firme nalazio se i određen broj poslužitelja na izdvojenoj lokalnoj mreži koja je pripadala samo njoj te određen broj poslužitelja na izdvojenoj lokalnoj mreži čije stanice su vidljive s Interneta (tj. poslužitelji u "demilitariziranoj zoni", odnosno DMZ-u).

U inicijalnom intervjuu s administratorima, dogovoreno je prosljeđivanje konkretne sheme trećeg sloja mreže i njezin prikaz u obliku dijagrama, vidljiv je na slici 5.1.

Nakon dobivanja osnovne sheme mreže, dogovoreno je da će se skenirati sve označene podmreže. Dogovoreno je smještanje Nessus skenera i to jednog na računalo smješteno na razvojni LAN-R i drugog na poslužiteljski LAN-P. Za DMZ inicijalno nije donesena odluka, ali je naznačeno da će se vjerojatno koristiti jedan od prva dva skenera i da će promet ipak ići preko usmjernika. Time se pristalo na rizik ispada usmjernika i na planiranje testova kako bi

se to izbjeglo. Pogodnost je manjak sigurnosnih zaštitnih stijena između pojedinih podmreža, odnosno točnije, vrlo labava pravila filtriranja unutar organizacijske mreže.



Slika 5.1. Shema 3. sloja ispitne mreže

Slijedeći korak je bio upoznati pojedine podmreže. Administratorima je postavljen upit za detaljniju specifikaciju svake od mreža, odnosno postavljena su pitanja:

- Koliko otprilike ima aktivnih stanica na svakoj podmreži?
- Koje su namjene pojedinih stanica na podmrežama i koji operacijski sustavi se mogu nalaze na njima?
- Gdje se nalaze IDS i slični programi?
- Gdje se nalaze uska grla i koji poslužitelji su od posebne važnosti?
- Koristi li se DHCP i na kojim podmrežama?

Paralelno s postavljanjem upita, korišten je Nmap za inicijalno istraživanje podmreža. Kako se rad ne bi poremetio, Nmap je korišten s konzervativnim postavkama s obzirom na brzinu izvođenja i općenitu invazivnost na ciljne mreže. Također, posjećena je i fizička lokacija mreže i tada su na nekim od podmreža uočeni mrežni pisači, bežične pristupne točke i IP telefoni te preklopnici (eng. *switch*) i usmjernici. Vrijedi napomenuti da su se kratko nakon početka skeniranja javili neki oprezniji korisnici s upitima što se događa, jer su uz pomoć osobnih sigurnosnih zaštitnih stijena odmah uočili skeniranje pristupa. Nakon objašnjenja skeniranje je nastavljeno i na kraju su rezultati pokazali da su upravo takvi korisnici imali daleko najbolje osigurana računala.

Nakon cjelokupnog istraživanja mreže, sakupljene su sljedeće informacije o pojedinim podmrežama. Osnovna obilježja razvojne podmreže su prikazana u tablici 5.1.

Tablica 5.1. Osnovna obilježja: LAN-R

LAN-R - 192.168.201.0/24	
Vrste stanica	• Windows stolna računala i pretežito Windows prijenosna računala. • Mrežni pisači (Kyocera) i IP telefoni (Broadcom). • Bežične pristupne točke. • Preklopnici (s web sučeljem za konfiguraciju).
Korisnici	• Programeri. • Voditelji projekata, šefovi odjela i njihovi zamjenici. • Administrativni djelatnici. • Svi imaju administrativne ovlasti na računalima.
Granica mreže	• Zyxel usmjernik i sigurnosna zaštitna stijena prema internoj mreži na 192.168.201.1. • Vlastiti izlaz na Internet preko Zyxel uređaja. • Redovita VPN spajanja prema poslovnom partneru. • Potencijalno probijena bežična pristupna točka. • Potencijalno nekontrolirane instalacije na prijenosnim računalima.
Posebnosti	• Programeri većinom koriste prijenosna računala i često su na lokacijama poslovnih partnera. • Prijenosna i stolna računala ponekad se "nasljeđuju" bez reinstalacije. • Većina računala koristi specifičnu VPN programsku podršku konfiguriranu prema sigurnosnim zahtjevima poslovnog partnera, na čiju se mrežu spajaju, a koja uključuje strogi paketni filter. • Prosječno oko 50 - 65 aktivnih stanica. Mnoge skrivene iza zaštitnih stijena. • Imat će skener unutar granica (jedno prijenosno računalo).

Obilježja poslužiteljske podmreže su prikazana u tablici 5.2.:

Tablica 5.2. Osnovna obilježja: LAN-P

LAN-P - 192.168.100.0/22	
Vrste stanica	• Windows (većinom verzija 2003.) i Linux (većinom Debian) poslužitelji (telefonija, podatkovni poslužitelji, poslužitelji za videonadzor i slično). • Mrežni pisač (HP). • Preklopnici (s web sučeljem za konfiguraciju).
Korisnici	• Grupa u cjelini. • Firma unutar grupe.
Granica mreže	• OpenBSD usmjernik i zaštitna stijena na 192.168.101.0.
Posebnosti	• Miješana mreža s poslužiteljima na kojima firma ima i na kojima nema ovlasti. • Potreban povećan oprez kod skeniranja. • Prosječno oko 30 aktivnih stanica. Filtrirane one izvan granica skeniranja. • Imat će skener unutar granica (poslužitelj za telefoniju).

I, na kraju, obilježja demilitarizirane zone su prikazana u tablici 5.3.:

Tablica 5.3. Osnovna obilježja: DMZ

DMZ - 192.168.111.0/24	
Vrste stanica	• Linux poslužitelji (web, mail).
Korisnici	• Grupa u cjelini – poslužitelji izvan plana skeniranja. • Firma unutar grupe – poslužitelji u i izvan plana skeniranja.
Granica mreže	• OpenBSD usmjernik i zaštitna stijena na 192.168.111.1.
Posebnosti	• Miješana mreža s poslužiteljima na kojima firma ima i na kojima nema ovlasti. • Potreban povećan oprez kod skeniranja. • Skener možda neće biti unutar granica, no ako hoće, bit će korišteno prijenosno računalo s LAN-R.

Na temelju ovih podataka, slijedeći sastanak s administratorima sustava poslužio je za detaljnije specificiranje korištenih testova. Vrijedi primijetiti da je u ovoj fazi određen i izbor uloge, i to smještajem skenera na domenu razašiljanja te dodijeljenim ovlastima.

5.2 Diferencijacija testova i vrijeme ispitivanja

Tablice 5.1.-5.3. su ispisane i predočene administratorima sustava uz zahtjev za komentarima. Slijedeći korak bio je izbor testova koji će se obavljati na pojedinim podmrežama i u tom izboru su se uvažavali traženi komentari. Testovi su izabrani i prikazani administratorima koji su ih odobrili, čime su ostvareni svi netehnički i skoro svi tehnički preduvjeti za skeniranje. Tek nakon analize testova određene su ovlasti na pojedinim mrežama koje će imati skener. Preostalo je još aktivirati korisnički račun skenera i odrediti vrijeme skeniranja za pojedine mreže.

U nastavku je prikazano kako su specifičnosti pojedine podmreže utjecale na odabir testova i vrijeme ispitivanja.

5.2.1 Razvojni LAN-R – priprema i planiranje

Činjenica da su na ovoj mreži pretežito prijenosna računala, značila je da sigurno nisu sva korištena računala skenirana, odnosno da neka možda nisu na lokaciji u vrijeme skeniranja. Osim toga, i među prisutnim računalima bilo je onih koji su, radi prirode posla, morali biti VPN-om spojeni na udaljenu lokaciju, čime je njihovo skeniranje onemogućeno. One stanice koje i nisu bile spojene na udaljenu lokaciju, i dalje su imale spomenuti paketni filter, kojeg je bilo potrebno zaobići. Konfiguracija paketnog filtera nije dolazila u obzir jer je ona centralizirana na lokaciji poslovnog partnera. Drugim riječima, bez intervencije ne bi bilo moguće skenirati gotovo niti jedno računalo. Još jedan problem je predstavljala dodjela adresa pomoću protokola DHCP. Trebalo je osigurati način da se IP adrese povežu s ljudima koji koriste računala.

S obzirom na broj stanica i administrativne ovlasti korisnika, u plan reakcije na rezultate procjene ulazili su sami korisnici skeniranih računala. Nakon pregleda rezultata, u planu je bilo prosljeđivanje tih rezultata korisnicima koji bi onda bili zaduženi za popravljanje uočenih problema. Na ovakvu odluku utjecao je broj stanica koje će biti skenirane i priroda tih stanica. Naime, namjena tih stanica je bila razvojna, no politika poduzeća je bila dosta labava oko korisničkih ovlasti pa su ih korisnici rabili kao osobna računala i na njih instalirali koješta. Automatizirani skener je u stanju pronaći raznovrsne ranjivosti u mnogo vrsta programa i tada

je odlučeno kako je najbolje da za popravljanje pojedinog problema bude odgovoran (i doslovce zadužen) korisnik računala.

Iz navedenih razloga, skeniranje se odvijalo u više navrata, koristeći pauze u radu preko VPN-a (i pauze u radu općenito) te, dijelom, izvan radnog vremena, ako bi korisnici ostavljali računala u prostorima firme. Dio računala nije skeniran i odlučeno je da se rezultati dobiveni skeniranjem prisutnih smatraju reprezentativnima i da se ne-skenirana računala smatra otprilike jednako ranjivima kao najgori slučajevi među skeniranima, dok se ne pokaže suprotno.

Od testova, korišteni su svi dostupni iz grupa testova koje se odnose na Windows operacijski sustav (u Nessusu to su grupe "**Windows**", "**Windows: Microsoft Bulletins**" i "**Windows: User management**"). Time je pokriven izuzetno velik raspon ranjivosti, od onih specifičnih za uobičajene programa koji se nalaze na tipičnim Windows instalacijama, preko konfiguracije korisničkih računa, pa sve do Microsoftovih objava pojedinih ranjivosti i zakrpa za njih. Osim toga, ispitivana računala su provjerena na sve poznate ranjivosti iz područja web poslužitelja jer mnoge aplikacije ponekad nude funkcionalnost kroz nekakve emulacije rada web poslužitelja, odnosno ponekad doista poslužuju određene podatke koristeći iste protokole (istovremeno nudeći i iste ranjivosti kao stvarni web poslužitelji). Kompletnu listu, kao i rezultate, može se pronaći u popratnim materijalima, a vrijedi još navesti da su pokretane provjere iz grupe "**Peer-to Peer File Sharing**", s namjerom otkrivanja programa za raspodijeljeno dijeljenje datoteka. Takvi programi su često protivni sigurnosnoj politici poduzeća, a i sami su često ranjivi na preljeve spremnika i slične ranjivosti. Što se tiče zaštite autorskih prava i intelektualnog vlasništva, neke skripte iz ove skupine pokušavaju na dosta pojednostavljeni način pokriti i ovu tematiku. Neke od njih, tako, samo postojanje MS Word dokumenta ili MP3 datoteke prijavljuju kao potencijalno kršenje zakona, što, dakako, uopće ne mora biti slučaj.

5.2.2 Poslužiteljski LAN-P i DMZ – priprema i planiranje

Računala na ovim podmrežama dijele infrastrukturu sa sustavima koji su izvan granica skeniranja. Iako su poduzeti koraci koji bi trebali svima osigurati neometan rad, odlučeno je da se ove podmreže skeniraju isključivo izvan radnog vremena. U izboru računala vodilo se naputkom da treba skenirati samo računala koja imaju DNS ili NetBIOS ime, čime su zapravo uključena samo predprodukcijska i produkcijska računala, a isključeni su usmjernici, sigurnosne zaštitne stijene i slični strojevi, koji igrom slučaja nisu imali takva imena (odnosno samo su imali IP adrese). Na DMZ-u je, doduše, bilo par iznimaka i te su bile izvan granica skeniranja.

Za njih su uglavnom odabrani testovi specifični za njihov operacijski sustav (najčešće Debian, Windows Server 2003 i jedan CentOS na DMZ-u) i za servise koje nude. Mada je isto nastojanje bilo i na razvojnoj mreži, ovdje je višestruko naglašeno kako testovi ne bi smjeli biti opasni, odnosno otkrivanje ranjivosti ne bi smjeli potkrjepljivati pokušajem da se ranjivost iskoristi. Osim toga, kako bi skeniranje što manje ometalo normalan rad, ono je izvršeno uz postavke koje manje opterećuju mrežu. Skeniranje na LAN-P izvršeno je odjednom za sve vrste operacijskih sustava, odnosno kako bi se u jednom navratu sakupilo što više informacija, bez podjele na skeniranja ovisna o OS-u.

S ovime je zaključena prva faza procjene ranjivosti:

- mreža je mapirana i dobro upoznata;
- uzete su u obzir sve specifičnosti svake od podmreža;
- osjetljive stanice su zaštićene izborom testova i smještajem skenera;

- dodijeljene su ovlasti skeneru, skener je instaliran, obavljena inicijalna snimanja;
- odabrani su pojedini testovi.

Uslijedio je niz skeniranja, odnosno sljedeća faza.

5.3 Skeniranje ispitne mreže

Dio pokretanja skenera obavljen je u istraživačke svrhe još u fazi mapiranja mreža. Za svaku od podmreža iz tih razloga pokretan je Nmap, sa svrhom otkrivanja aktivnih stanica, ili prosječnog broja aktivnih stanica. Pritom je korišten njegov izlaz kao dio ulaza u Nessus, mada i sam Nessus nudi opcije za otkrivanje aktivnih računala. Dodatno, i Nessus je pokretan u sličnu svrhu, ali nakon dodavanja domenskog ili ssh korisnika na ciljne mreže. Tada bi se pokrenuo Nessus s posebnim politikama koje ne sadrže puno testova, odnosno imaju isključivo dijagnostičku namjenu otkrivanja računala na kojima skener nema ovlasti. Ovako proizvedeni rezultati bi služili kao povratna veza prema administratorima sa svrhom potvrde radi li se o računalu koje s pravom ne nudi pristup ili postoje problemi s pravima korisnika.

Ova "pomoćna" skeniranja imaju svrhu pripreme za prava skeniranja i potragu za pravim ranjivostima, a bila su vrlo korisna i zapravo su se na kraju pokazala kao neophodna. Naime, pokazalo se kako uzastopno skeniranje s istom politikom može ponekad vratiti različite rezultate! Do određene diskrepancije dolazilo je najčešće zbog faktora koje je nemoguće kontrolirati, poput latencije i opterećenja mreže, odnosno opterećenja skenera ili ciljnog stroja. Neki rezultati su ponekad naprosto kasnili radi povećanog vremena potrebnog ciljnoj stanici da odgovori na neku provjeru. Na ovakve faktore može se utjecati postavkama skenera koje određuju razinu paralelizacije skeniranja i broja paralelnih testova. No, kako pronaći pravu mjeru, nemoguće je općenito reći. Kako je pouzdanost ispitivanja, na kraju, od najveće važnosti, ovome je posvećeno dosta analize i kroz praktičan rad, osmišljene su sljedeće preporuke, koje su zapravo korištene kroz praktičan rad i koje su se pokazale primjerenima:

- Treba usporediti rezultate aktivnosti koje vrati Nmap s rezultatima koje ponudi Nessus.
    - Ako su različiti, više povjerenja dati Nmapu, povećati vremena čekanja i ponovo pokrenuti pretragu s Nessusom; kada se rezultati poklope, povećati vremena čekanja za otprilike sekundu.
- Planirane testove postepeno uključivati, odnosno povećavati broj planiranih testova postepeno kroz par inicijalnih skeniranja.
    - Provjeriti rezultate dok je njih malo – ako dođe do isteka vremena, to će biti vidljivo i tada treba još povećati vremena čekanja.
- Kada postepeno povećanje uključuje oko četvrtine planiranih testova, uključiti ostale testove.
- Planirati izvođenje u vremenu kada je mreža manje opterećena.

Dakako, ovakvo postupanje vodi prema usporavanju testova i zato treba postupno prilagođavati podskup testova, kako se ne bi pretjeralo.

Osim mrežnih utjecaja, postoji i par provjera koje mogu utjecati na ponovljivost testa, a one su većinom vezane uz provjere korisničkih računa. Primjerice, neki testovi iskušavaju uobičajene lozinke za podrazumijevane (predinstalirane) korisničke račune. U kombinaciji sa

čestom konfiguracijom koja dopušta samo određen broj pokušaja prijave za korisnički račun, može se dogoditi da se takav račun zablokira, odnosno privremeno onemogući. Ako drugi testovi ovise o tom korisničkom računu, oni se neće moći izvesti. Rješenje je u izdvojenom skeniranju na tipične lozinke. Gore navedene preporuke često otkriju i takve međuovisnosti testova koje se lakše dijagnosticiraju kada se testovi uvode postupno.

Na kraju, postepeni pristup doveo je do toga da su dogovoreni testovi, kada su jednom pokrenuti, izvedeni u razumnom vremenu i bez greške! Uz to, nije uzrokovano niti jedno ispadanje nekog od ciljnih sustava, mada je bilo lažnih prijava. Naime, skeniranje računala na LAN-R dijelom je izvedeno izvan radnog vremena, kako je navedeno ranije. Sutradan ujutro primijećeni su problemi u radu s elektroničkom poštom i krivnja je, očekivano, prvo pala na skener i procjenu ranjivosti. No, kako su bila pogođena i računala koja nisu bila skenirana, a poslužitelj nije uopće bio skeniran taj put, pokazalo se da se skeniranje poklopilo s djelomičnom nadogradnjom poslužitelja elektroničke pošte. Ovakvi događaji zapravo naglašavaju važnost postojanja ugovora koji predviđa stvarnu opasnost štete nastale sigurnosnim skeniranjem i detaljnu raspodjelu odgovornosti u svakom predvidivom slučaju.

5.3.1 Razvojni LAN-R – skeniranje

Prije pokretanja skeniranja bilo je potrebno fizički posjetiti lokaciju i zatražiti (ili ručno izvršiti) gašenje spomenutog paketnog filtera. Također, isti se morao ponovo uključiti nakon izvedbe skeniranja. Taj dio posla su odradili zaposlenici ako su bili nazočni, odnosno izvršitelj procjene (koristeći ovlasti korisnika koji je postavljen na domeni za potrebe skeniranja) za računala čijih korisnika nije bilo na lokaciji u tom trenutku. Osim toga, iz popisa aktivnih stanica uklonjeni su pisači, telefoni i preklopnici, prema dogovoru s administratorima mreže.

Konačno skeniranje je trajalo nešto preko 66 minuta i pronašlo je 22 aktivne stanice, s različitim rezultatima. Vrijedi napomenuti par zanimljivosti iz rezultata. Prvo, računalo s kojeg je skeniranje započelo imalo je manjkave rezultate jer skener nije uspio pristupiti podsustavu Registry Windows operacijskog sustava. Tehnički, za taj neuspjeh mora biti kriv skener i to se može smatrati njegovim nedostatkom, jer su svi preduvjeti za pristup Registryju ispunjeni. Preciznije, kriv je priključak koji nije uspio iskoristiti dovoljne ovlasti skenera na fizičkom stroju domaćinu.

U klasifikaciji ranjivosti koja ih dijeli na visoko i srednje rizične te na informacije, njih čak 11 imalo je visoko kritičnih ranjivosti. Među preostalih 11, još 4 su imali srednje rizične ranjivosti, a preostalih 7 su većinom bile specifične stanice poput bežične pristupne točke, adaptera za IP telefoniju i jednog svježe instaliranog računala bez ikakve dodatne programske podrške i s najsvježijim zakrpama.

Nadalje, najsigurnije računalo nije imalo čak niti jednu visoko rizičnu ranjivost, a svega još dva su imali sličnu razinu sigurnosti, od kojih je jedno koristio isti korisnik koji je koristio ovo najsigurnije. Drugi, koji je imao samo jednu kritičnu ranjivost i svega par srednje rizičnih je čak na računalu imao instaliran Secunia PSI – besplatan program koji služi upravo skeniranju vlastitog računala protiv poznatih sigurnosnih rupa [15].

Najranjivije računalo imalo je 189 visoko rizičnih ranjivosti i 40 srednje rizičnih – daleko najviše u obje kategorije.

5.3.2 Poslužiteljski LAN-P – skeniranje

Skeniranje ove mreže potrajalo je oko sat vremena, a skenirana računala imala su uglavnom odlične rezultate. Poslužitelji su mahom dobro održavani, uz par primjetnih iznimki, koje su opet u dijeljenoj ingerenciji, odnosno o njima se u pravilu brine druga ekipa administratora.

Jedna od iznimaka je Windows poslužitelj koji ima instalirane neke programe tipične za osobnu ili radnu stanicu, poput programa za razne multimedijalne sadržaje, mrežno druženje i slično.

Posebno zanimljiv je Windows poslužitelj koji upravlja sustavom za videonadzor. Također rekorder u svojoj kategoriji, ovaj poslužitelj ima čak 126 visoko rizičnih ranjivosti, mnoge vezane uz poznate ranjivosti za koje je Microsoft objavio zakrpe. No, kako pokazuje jedan od informativnih zapisa, na ovom računalu je isključeno automatsko nadograđivanje, što je vjerojatno potaknulo takvo stanje. Dodatno, na ovom računalu primijećena je mogućnost prijave za rad kao administrator, ali bez lozinke. Ova kombinacija ranjivosti može dovesti do narušavanja privatnosti zaposlenika.

5.3.3 DMZ – skeniranje

Demilitarizirana zona je na kraju skenirana skenerom lociranim na razvojnoj mreži LAN-R. Radilo se o još 4 poslužitelja, s iznimno malim brojem ranjivosti, većinom iskoristivih jedino uz određene konfiguracije (odnosno, gledano s druge strane, ove se ranjivosti mogu otkloniti pažljivom konfiguracijom). Skeniranje je trajalo nešto preko 16 minuta.

Skeniranje je obavljeno s druge podmreže jer se radilo o malom broju ciljnih poslužitelja, pa nije očekivan velik promet, a na ciljnoj mreži nije bilo pogodno spajati novi poslužitelj za potrebe skeniranja.

5.4 Faza analize rezultata

Analiza rezultata (i procjena ranjivosti općenito) na "izlazu" iz procesa zapravo ima dvije dimenzije – pregled utvrđenog stanja i naputke za popravak. Naputci za popravak i osiguravanje su važan dio procjene. Kod manjih do srednjih organizacija možda i najvažniji, jer dobro izvedena procjena ranjivosti može biti sasvim dovoljan oblik zaštite, uz korištenje osnovnih tehničkih postupaka. No, kod većih firmi (i opreznijih srednje velikih), opis trenutnog stanja jednako je važan, ako ne i važniji. Dobro poznavanje stanja mreže je osnova za daljnji rad na osiguravanju, bilo to penetracijsko ispitivanje ili cjelokupna provjera stanja informacijskog sektora. Iz tog razloga je dosta važan dio procjene ranjivosti upravo izvještaj (odnosno niz izvještaja). I ovdje zapravo nema jasno definiranih pravila, a i potrebe pojedinih osoba u firmama se dosta razlikuju – viši menadžment slabo mari za jedan tehnički izvještaj, a i prilagođeni izvještaj je manje privlačan u odnosu na niz prilagođenih izvještaja koji zapravo nude ono najvažnije, a to su trendovi (kretanje broja sigurnosnih rupa, broja zatvorenih propusta te stanje omjera uloženog novca i ostvarene sigurnosti s potencijalnim gubitkom u slučaju proboja). Ovaj rad se koncentrirao na potrebe administratora i njihove želje. Na kraju, administratori sustava nisu neupoznati s informacijskom sigurnošću i na njima ostaje osiguranje sustava, pogotovo u mreži koja je bila predmet proučavanja.

U ovom konkretnom slučaju, administratori su zahtijevali integralne izvještaje koje generira Nessus za potrebe arhiviranja i reference te modificirane izvještaje za svako zasebno računalo. Modificirani izvještaji trebali bi, osim izdvajanja rezultata za pojedino računalo, sadržavati samo visoko i srednje rizične ranjivosti. Također, izvještaji koji izlaze iz područja firme moraju imati skrivene osjetljive podatke, poput korisničkih imena, mrežnih domena itd. Osnovni plan reakcije bio je poslati korisnicima mreže LAN-R zasebne izvještaje (koji bi

prije bili pregledani) uz zahtjev za potrebnim popravcima. Pritom, pri preuzimanju izvještaja, korisnik bi dobio usmene upute za sve važnije stavke koje bi administrator uočio, ili bi mu bile naglašene od strane izvršitelja procjene. Za poslužiteljske podmreže, administratori su tražili sugestiju oko rangiranja ranjivosti i sažeti prikaz potrebnih radnji za osiguravanje svakog ranjivog poslužitelja. Ti izvještaji dostupni su u pratećim materijalima.

Pri analizi i daljnjoj distribuciji rezultata određena količina pažnje posvećena je tajnosti, kako korporativnih, tako i privatnih podataka korisnika. Skener, između ostalog, među rezultate često uvrsti i ispis pronađenih datoteka, ako je prikladno. Primjerice, ako za ciljno računalo skener uoči dostupnost dijeljenih mrežnih medija, on će ispisati što je na tim medijima (npr. mrežnim direktorijima) dostupno i ako u to spadaju neki povjerljivi podaci, treba pripaziti što se dalje događa s njima. U nastavku slijede zaključci doneseni za pojedine podmreže. Oni su raspravljeni s administratorima kao dio napora za poboljšanje stanja i na temelju tih zaključaka su planirani daljnji koraci.

5.4.1 Razvojni LAN-R - rezultati

Rezultati u potpunosti odgovaraju onima na tipičnim radnim stanicama. Riječ je većinom o poznatim propustima u programima za reprodukciju multimedijalnih sadržaja i raznih vrsta dokumenata, poput Winampa, RealPlayera, FlashPlayera, Adobe Acrobat Readera, njihovih komponenti i slično. Osim toga, često su prijavljeni nezakrpani programi poput Internet Explorera ili čak kritičnih komponenti Windows OS-a, poput podsustava za upravljanje i bilježenje događaja (eng. *Microsoft Event System*) te RPC podsustava.

Dobar primjer vrlo opasne ranjivosti pronađene na ovaj način je vidljiv u pratećim materijalima i izdvojen ovdje. Riječ je o relativno svježoj ranjivosti, za koju je Microsoft objavio zakrpu 12.8.2008. Prema službenom opisu, napadač bi mogao izvršiti proizvoljni izvršni tekst s ovlastima prijavljenog korisnika, a kako se ovdje radi o potpunom skupu ovlasti, tako je napadač u prilici preuzeti kontrolu nad cijelim računalom. Za realizaciju napada dovoljno je konstruirati posebnu grafičku datoteku! Nessusova prijava vidljiva je na slici 5.2.

Vulnerability in Microsoft Windows Image Color Management System Could Allow Remote Code Execution (952954)

Synopsis :

Arbitrary code can be executed on the remote host through the Microsoft
Color Management System (MSCMS) module of the Microsoft ICM componenents.

Description :

The remote host contains a version of the Color Management Module which
is vulnerable to a security flaw which may allow an attacker to execute
arbitrary code on the remote host by crafting a malformed image file and
entice a victim to open it.

Solution :

Microsoft has released a set of patches for Windows 2000, XP and 2003 :

http://www.microsoft.com/technet/security/bulletin/ms08-046.mspx

Risk factor :

High / CVSS Base Score : 9.3
(CVSS2#AV:N/AC:M/Au:N/C:C/I:C/A:C)
CVE : CVE-2008-2245
BID : 30594

Nessus ID : 33875

Slika 5.2. Nessus prijava vrlo ozbiljne ranjivosti

Realizacija ovakvog napada zapravo je izuzetno lagana. Iako korisnici već godinama paze da ne pokrenu izvršni kôd, i dalje nema nepovjerenja prema slikama i ostalim multimedijalnim dokumentima. Drugim riječima, dojam je da nam ovakvi dokumenti ne mogu naštetiti i upravo ovo povjerenje može uzrokovati probleme.

Slika 5.2. također prikazuje tipičnu Nessus prijavu, kakva je sastavni dio Nessusovog izvještaja. Može se primijetiti kako se radi o vrlo dobro formatiranom i sažetom izvještaju koji nudi sve potrebne informacije, od opisa do reference na rješenje. Kada nije riječ o programskom rješenju, u "**Solution**" sekciji pišu upute kako promijeniti konfiguraciju ili koje već korake treba poduzeti kako bi se problem riješio. Ovaj način je vrlo pogodan za početnika, jer se relativno lako poslužiti gotovim rješenjem i jasnim uputama. No, i administratori imaju puno koristi od toga, jer im poveznica nudi izravan put prema potrebnim zakrpama. Također, vrijedi naglasiti "**Nessus ID**" unos, na kraju izvještaja. Riječ je o šifri

provjere koja je rezultirala dojavom i klikom na poveznicu se prikazuje njezina dokumentacija i izvorni tekst na službenim Tenableovim stranicama.

Rezultati cijele mreže ipak sugeriraju sljedeće: zapažen dio ranjivosti uzrokuju programi izvan Microsoft Windows Update dosega. Na većini računala su komponente koje Microsoft obnavlja (kroz standardnu ponudu zakrpa) relativno ažurne i njihova sigurnost je prihvatljive razine. No, velik dio korištenih programa nije iz te domene, a o njihovom ažuriranju nije vođeno toliko računa. Tipičan primjer, uz spomenute programe, su i razni poslužitelji baza podataka te dosta rasprostranjena Java.

Java primjerom ukazuje na ranije spomenutu kontrolu rizika. Naime, ponekad se moramo odlučiti na određenu razinu rizika jer znamo da nam iz nekog razloga implementacija zakrpe ne dolazi u obzir – a Java programi često uzrokuju ovakvo stanje jer su ponekad ovisni o verziji Jave za koju su pisani. Uklanjanje verzije Jave za koju je poznato da je ranjiva može nam onemogućiti korištenje programa i na to ponekad ne možemo pristati. Također, na računalu može biti instalirano više verzija Jave i programi mogu općenito biti konfigurirani da koriste jednu od njih – onu najnoviju, osim iznimno. Time zapravo u potpunosti kontroliramo ovu ranjivost i prijava je tada zapravo lažna uzbuna.

Općeniti zaključak za ovu mrežu je da je dosta nesigurna. Računala imaju dosta poznatih ranjivosti i sreća je da koriste rečeni paketni filter, u protivnom bi statistički vrlo vjerojatno bila zaražena nekom vrstom neželjenog softvera kroz automatizirane pokušaje napada.

Rješenje je u ažuriranju svih programa koji se koriste i deinstalaciji nepotrebnih. To je proslijeđeno korisnicima na realizaciju. Također, ona računala koja ne koriste Windows Update bit će rekonfigurirana da to čine.

5.4.2 Poslužiteljski LAN-P i DMZ – rezultati

Ovdje su rezultati puno bolji i ova računala su, kako se i očekuje, puno pažljivije ažurirana. Rezultati onih lošijih među njima, a koji nisu direktno odgovornost dodijeljenih administratora, proslijeđeni su odgovornima u integralnom obliku.

Za Linux poslužitelje se većinom radi o blago zastarjelim verzijama jezgre, najviše za jednu verziju. Osim toga, većina se ranjivosti odnosi na zastarjelu verziju Samba poslužitelja. Oba problema riješena su unutar distribucije Linuxa koja je u igri pa se jednostavnom nadogradnjom rješavaju ovi problemi.

Na LAN-P se može pronaći poslužitelja s instaliranim neodgovarajućim programima i ponegdje su prijavljene neostvarive ranjivosti, primjerice ranjivosti u Internet Exploreru ili mogućnost nasilnog preuzimanja X11 sjednice (eng. *session hijacking*). No, nije za očekivati da će se poslužitelj koristiti kao stanica za "surfanje" Internetom ili da će se uopće podizati grafičko sučelje pa su ove ranjivosti manje opasne.

Općeniti zaključak za obje mreže je da imaju svojih problema i da su računala nedovoljno osigurana, ali da su, uzevši u obzir sve okolnosti poput smještaja i organizacijski uspostavljenog prava pristupa njima, relativno pouzdana. Linux poslužitelje treba nešto češće ažurirati, a isto vrijedi i za par Windows 2003 poslužitelja.

Neka računala, proglašena dovoljno važnima da radi neometanog rada budu izostavljena iz skeniranja, ostaju nepoznanice (poput sigurnosnih zaštitnih stijena i određenih poslužitelja) i preporučena je posebna pažnja u radu s njima. Upravo zbog manjka informacija, takva računala ne spadaju u kontrolirani rizik, već su – upravo suprotno – nekontrolirana opasnost.

5.4.3 Općenita slika sigurnosti mrežnog sustava

Kroz provođenje procesa procjene ranjivosti su administratori sustava dobili bolju kontrolu i detaljniji uvid u mnoge aspekte mreže. Osim za izravnu procjenu ranjivosti, profitom u cijelom postupku smatraju upravo kvalitetan pregled aktivnih resursa i mogućnost automatizirane provjere proizvoljnih parametara kroz pisanje vlastitih priključaka, čime je omogućena emulacija cjelokupnog pregleda stanja sustava, na jednoj nižoj razini.

S obzirom na viđeno, zaključeno je kako je mreža relativno sigurna izvana. Prijavljeni nedostaci na DMZ-u bit će riješeni redovitim ažuriranjem u dosadašnjem ritmu i sigurnost iz ove perspektive izravno ovisi o duljini između dva ažuriranja sustava, padajući od dana ažuriranja prema sljedećem. Iznutra, mreža je poprilično ranjiva. Uspije li napadač dobiti korisničke ovlasti na mreži, istog trenutka dobiva pristup velikoj količini podataka na mreži organizacije i potencijal da iskorištavanjem pronađenih propusta podigne te ovlasti na višu razinu, čime opasnost postaje još veća. Na ovom području će se morati poraditi, pogotovo na poslužiteljima koji su na mreži LAN-P.

Razvojna mreža doživjet će organizacijske promjene jer stanje na njoj nije bilo zadovoljavajuće. To se posebno odnosi na računala koja imaju isključen Windows Update. Što se ostalih programa tiče, njihovo ažuriranje će se poticati uvođenjem politike reinstalacije sustava s svakim incidentom – čime će se gubiti svi nestandardni programi i podaci nađeni na računalu (koje, ipak, nije privatno vlasništvo korisnika).

Kroz ovakav pristup može se uvidjeti sva pogodnost skeniranja s ovlastima. Naime, da se sustav skenirao bez ovlasti, rezultati ne bi nosili niti desetinu informacije koju nose ovako. Takva situacija je dosta realističnija, u smislu da nalikuje onoj koju zapravo ima napadač, no treba imati na umu da napadačima i ne treba toliko informacija. Napadaču je dovoljno pratiti vijesti iz svijeta sigurnosti i *pretpostaviti* da će negdje naći neki propust. Iskusnom napadaču je dovoljno svega par takvih pretpostavki. Zato je puno bolje iskoristiti priliku za informirano skeniranje s ovlastima, jer se potencijalno velik broj poznatih ranjivosti može pronaći automatski na mnogo stanica.

5.5 Zaključno o izvedbi procjene ranjivosti

Kao što je u teorijskoj razradi i pretpostavljeno, glavnina posla je u pripremi. Ta faza je u izvedbi značila brojne sastanke s rukovodećim ljudima ispitivane tvrtke i nebrojene kontakte s administratorima. Kroz upoznavanje mreže i kroz izvedbu procjene primijećeno je da može doći do nepredvidivih problema i vrlo specifičnih posebnosti mreže, odnosno rješavani su problemi koji na nekoj drugoj lokaciji vjerojatno ne bi postojali. I obratno, na drugoj lokaciji susrest će se posve novi problemi, specifični za tu lokaciju.

Nakon brojnih dogovorenih detalja uslijedio je donekle automatizirani dio posla, za kojeg se mora napomenuti da je manje automatiziran nego što marketing navedenih alata sugerira. Skeniranje često treba ponavljati, modificirati i prilagođavati faktorima koji se mijenjaju na dnevnoj bazi. Korišteni alati su u velikoj mjeri doista izvrsni, nudeći prilagodljivost svakoj vrsti mreže. No, oni ostaju samo alati i njihovo korištenje omogućava masovnu i brzu pretragu za velikim brojem ranjivosti, ali ne omogućava selekciju problema koje prve treba riješiti, niti su u stanju povezati više ranjivosti u doista ozbiljniju prijetnju.

Krajnji rezultat je ponekad, mora se reći, nepouzdan. Među rezultatima mogu se naći problemi koji to nisu, odnosno neka prijava može biti lažno pozitivna, uzrokujući gubitak vremena i fokusa na problem. No, još gore, u rezultatima je moguće nepostojanje prijave za postojeći problem i takav onda prolazi nezapaženo. U tom slučaju naprosto nema pomoći jer

se radi o granicama mogućnosti alata – i njih je pisao čovjek pa nisu posve pouzdani, baš kao ni ostali programi koje provjeravaju.

Proces ispitivanja ranjivosti je dosta složen, s puno prilika da krene u lošem smjeru. No, rezultat je višestruko koristan i iskoristiv na puno načina, od pregleda sigurnosti sustava do kvalitetnog pregleda postojećih resursa za, primjerice, potrebe planiranja. Također, dobre popratne pojave uključuju i bolji nadzor nad svakodnevnom uporabom službenih uređaja i imaju općenito dobar utjecaj na ponašanje cijele organizacije po pitanju osobnih i drugih tajnih podataka.

Na kraju, vrijedi još jednom pogledati primjer dijela Nessus izvještaja na slici 5.2. i uočiti "CVE" zapis u dnu prikazane stavke: CVE : CVE-2008-2245. Riječ je o takozvanom CVE identifikatoru pronađene ranjivosti (eng. *Common Vulnerabilities and Exposures*), a to je poseban identifikator koji (kroz poseban proces) pojedinim ranjivostima dodjeljuje korporacija MITRE. Riječ je o prihvaćenom, centraliziranom sustavu imenovanja poznatih ranjivosti sa svrhom lakšeg dijeljenja podataka o pojedinim problemima između različitih sigurnosnih analizatora, baza podataka i metodologija. Ovo je izuzetno korisno, jer se radi o jedinstvenom naporu da se brojne ranjivosti jedinstveno imenuju, odnosno da se uspostavi nekakav standard koji će prekinuti primjetan nered u referenciranju ranjivosti. To je problem koji je dobro poznat u svijetu virusa i crva, a kod ranjivosti je također u porastu – za svaku ranjivost bi svaki zasebni proizvođač naveo svoje "ime", odnosno identifikator koji će smatrati referentnim. Na ovaj način je omogućena unifikacija tih identifikatora, a Nessus i Retina uz svaku ranjivost za koju mogu, navode i CVE identifikator (i još neke, manje važne), a čime dodatno naglašavaju svoje prihvaćanje aktualnih standarda i omogućavaju lakše korištenje rezultata skeniranja.

6. Zaključak

U svijetu gdje je mrežno računarstvo već postalo nezaobilazan dio poslovanja i svakodnevnog života, sigurnost više nije zanemarena djelatnost. Dapače, riječ je o rastućoj, zasebnoj industriji unutar informacijske tehnologije, koja nudi nove tehničke mogućnosti na području prevencije i reakcije na sigurnosne incidente.

Automatizirana procjena ranjivosti mrežnih sustava prvenstveno spada u preventivne djelatnosti, otvarajući pritom dodatne mogućnosti na području upravljanja rizikom. Radi se o postupku koji ima svoje specifičnosti i zahtijeva znatan napor kako bi se prilagodio posebnostima pojedine organizacije, no, izvede li se pravilno, procjena ranjivosti vraća puno više. Dobiveni rezultati otkrivaju stvarno stanje sustava koji može biti vrlo složen i, zapravo, nesagledliv "ručnim" metodama. Ovisno o potrebama, na osnovu tih podataka može se sustav osigurati, dalje ispitivati ili nadograđivati.

Izložena metoda i korišteni alati odlikuju se fleksibilnošću neophodnom za korisno ostvarenje procjene ranjivosti. Njihova je najveća snaga upravo u toj prilagodljivosti svakoj vrsti potreba kroz detaljno razlaganje u faze, odnosno konfiguraciju i uporabu alata.

Negativnom stranom mogla bi se nazvati cijena postupka koja, ovisno o razini složenosti ciljne mreže, može biti dosta velika. Naime, osim cijene samih alata, veći je trošak opisane prve faze, odnosno prilagodbe ciljanom sustavu. U izvedbi, ovaj dio procesa može potrajati dosta dugo i ne može se automatizirati.

Ipak, sigurnost računalnog sustava je stanje kojem svaka korporacijska mreža mora težiti, neovisno o cijeni i komplikacijama u izvedbi. Premda se uvijek mora računati na određenu količinu kompromisa i manjih nedostataka, visoka razina informacijske sigurnosti je postala obaveza svake organizacije, a redovita automatizirana procjena ranjivosti uz uporabu alata poput Nessusa i Retine je najkvalitetniji, nezaobilazan prvi korak u ispunjenju te obaveze.

7. Literatura

[1] Brewer, Eric; Gauthier, Paul; Goldberg, Ian; Wagner, David: Basic Flaws in Internet Security and Commerce, http://www.cs.berkeley.edu/~daw/papers/endpoint-security.html, 2. rujan 2008.

[2] TCP Idle Scan, http://nmap.org/book/idlescan.html, 2. rujan 2008.

[3] Microsoft Security Bulletin MS04-028, http://www.microsoft.com/technet/security/bulletin/MS04-028.mspx, 2. rujan 2008.

[4] Ogorkiewicz , Maciej; Frej, Piotr: Analysis of Buffer Overflow Attacks, http://www.windowsecurity.com/articles/Analysis_of_Buffer_Overflow_Attacks.html, 2. rujan 2008.

[5] Rajesh, Jose: Preventing Buffer Overflows, http://palisade.plynt.com/issues/2004Sep/buffer-overflows/, 2. rujan 2008.

[6] Friedl, Steve: SQL Injection Attacks by Example, http://www.unixwiz.net/techtips/sql-injection.html, 2. rujan 2008.

[7] Gee, Garret: Nessus Tools: HTML reports, http://garrettgee.com/2007/10/21/nessus-tools-html-reports/, 2. rujan 2008.

[8] Deraison, Renaud: NBE export format, http://list.nessus.org/pipermail/nessus/2002-June/002430.html, 2. rujan 2008.

[9] Microsoft, Windows XP Security Guide, http://www.nsa.gov/snac/downloads_all.cfm, 2. rujan 2008.

[10] Brenner, Bill: Misconfigured networks create huge security risks, http://searchsecurity.techtarget.com/news/article/0,289142,sid14_gci1303728,00.html, 2. rujan 2008

[11] http://www.penetrationtests.com/, 2. rujan 2008.

[12] http://en.wikipedia.org/wiki/Risk_management, 2. rujan 2008.

[13] Tenable Security Nessus, http://www.tenablesecurity.com/nessus/, 2.rujan 2008.

[14] eEye Retina, http://www.eeye.com/html/products/retina/index.html, 2. rujan 2008.

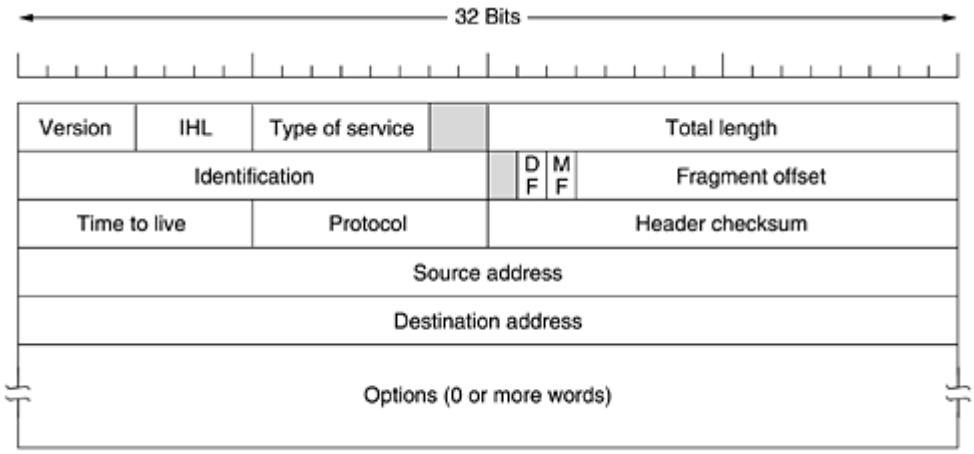
[15] Secunia PSI, https://psi.secunia.com/, 2. rujan 2008.

[16] Deraison, Renaud; Meer, Haroon; Temmingh, Roelof; van der Walt, Charl; Alder, Raven; Alderson, Jimmy; Johnston, Andy; Theal, George A.: Nessus Network Auditing, Syngress Publishing, 2004.

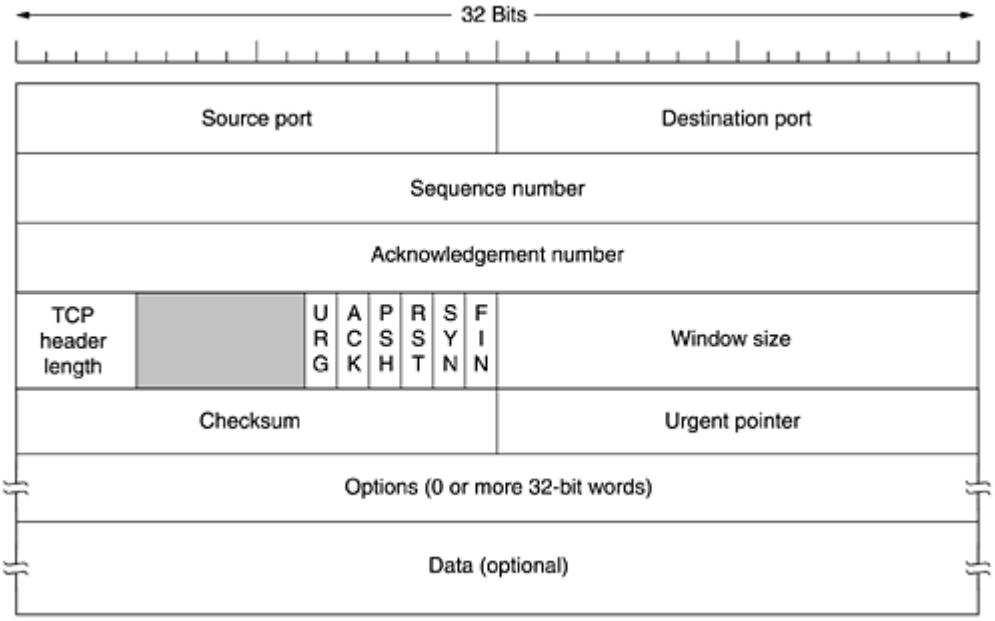
[17] Sigurnost računalnih mreža, http://os2.zemris.fer.hr/index.php?kat=98, 2. rujan 2008.

[18] R. Shirley: RFC4949 - Internet Security Glossary, Version 2, http://www.faqs.org/rfcs/rfc4949.html, 2. rujan 2008.

Dodatak A – IP paket i TCP segment



Dodatak - Slika 1: IP paket



Dodatak - Slika 2: TCP segment

Dodatak B – NASL primjer

Primjer je NASL skripta "apache_ssl_overflow.nasl", ID = 10918

```
# This script was written by Renaud Deraison <deraison@cvs.nessus.org>,
# with the impulsion of H D Moore on the Nessus Plugins-Writers list
#
# See the Nessus Scripts License for details
#

if(description)
{
 script_id(10918);
 script_bugtraq_id(4189);
 script_cve_id("CVE-2002-0082");
 script_version("$Revision: 1.11 $");

 name["english"] = "Apache-SSL overflow";

 script_name(english:name["english"]);

 desc["english"] = "
The remote host is using a version of Apache-SSL which is
older than 1.47

This version is vulnerable to a buffer overflow which,
albeit difficult to exploit, may allow an attacker
to obtain a shell on this host.


Solution : Upgrade to version 1.47 or newer
Risk factor : High";

 script_description(english:desc["english"], francais:desc["francais"]);

 summary["english"] = "Checks for version of Apache-SSL";
 summary["francais"] = "Vérifie la version de Apache-SSL";

 script_summary(english:summary["english"], francais:summary["francais"]);

 script_category(ACT_GATHER_INFO);


 script_copyright(english:"This script is Copyright (C) 2002 Renaud Deraison",
            francais:"Ce script est Copyright (C) 2002 Renaud Deraison");
 family["english"] = "Gain a shell remotely";
 family["francais"] = "Obtenir un shell ŕ distance";
 script_family(english:family["english"], francais:family["francais"]);
 script_dependencie("find_service.nes", "no404.nasl", "http_version.nasl");
 script_require_keys("www/apache");
 script_require_ports("Services/www", 80);
 exit(0);
}

#
# The script code starts here
#
include ("http_func.inc");
include ("backport.inc");

port = get_http_port(default:80);

if(get_port_state(port))
{
 banner = get_backport_banner(banner:get_http_banner(port: port));

 serv = strstr(banner, "Server");
 if(ereg(pattern:".*Apache(-AdvancedExtranetServer)?/.* Ben-SSL/1\.([0-9][^0-9]|[0-3][0-
9]|4[0-6])[^0-9]", string:serv))
 {
   security_warning(port);
 }
}
```

Dodatak C – NBE format

Nessus nudi izvoz u vlastitom NBE formatu (od Nessus Back End), nasljedniku također vlastitog NSR formata. Linije (stavke) u NBE formatu imaju sljedeći oblik [8]:

```
<kategorija>|<podmreža>|<adresa>|[informacije]
```

Pojedine stavke mogu izgledati ovako:

- `<kategorija>` može biti:
  - `"timestamps"` – linija označava vremenski zapis
  - `"results"` – linija sadrži neki od rezultata skeniranja
- `<podmreža>` označava nadgrupu adrese. Ako je adresa zadana kao ime u obliku "`računalo.domena.nad-domena`", to će biti "`domena.nad-domena`", a u slučaju IP adresa, to će biti adresa mreže (bez adrese računala, odnosno dio pokriven mrežnom maskom)
- `<adresa>` označava adresu stanice čiji je rezultat u pitanju
- `[informacije]` ovisi o kategoriji, odnosno radi li se o vremenskom zapisu ili rezultatu skeniranja; može biti:
  - `<akcija>|<vrijeme>` ako je kategorija jednaka "timestamps" i tada je akcija jedno od sljedećeg:
    - "scan_start" – početak cjelokupnog skeniranja
    - "scan_end" – kraj cjelokupnog skeniranja
    - "host_start" – početak skeniranja pojedine stanice
    - "host_end" – kraj skeniranja pojedine stranice
  - `<pristup>|<id-provjere>|<razina>|<izvještaj>` ako je kategorija "results" – ovo je stari NSR zapis i taj zapravo nosi podatke koji se vide u Nessus izvještajima

Primjer:

```
timestamps|||scan_start|Fri Aug 22 11:18:47 2008|
timestamps|||scan_end|Fri Aug 22 12:25:32 2008|
timestamps||192.168.201.251|host_start|Fri Aug 22 11:18:47 2008|
timestamps||192.168.201.251|host_end|Fri Aug 22 12:25:32 2008|
results|192.168.201|192.168.201.251|general/tcp|10180|Security Note|The
remote host is considered as dead - not scanning\n
```

Prva dva retka kažu kada je skeniranje započelo i kada je završilo – to je vidljivo iz kategorije i odgovarajuće akcije ("timestamps" i "scan_start" te "scan_end"). Sljedeća dva se odnose na pojedinu stanicu i peti je prikaz rezultata. Za ovo računalo neće biti rezultata jer je Nessus zaključio da na toj adresi nema ničega. Ovo je primjer prijave postojeće rupe:

```
results|192.168.201|192.168.201.103|microsoft-ds (445/tcp)|31794|Security
Hole|\nSynopsis :\n\nArbitrary code can be executed on the remote host by
sending a malformed file\nto a victim.\n\nDescription :\n\n [... skraćeno]
```