SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1750

Ispitivanje ranjivosti računalnog sustava

Luka Drvoderić

Zagreb, rujan 2008.

Zahvaljujem se mojoj obitelji (mami Mariji, tati Antunu, sestri Dini) koja mi je pružila veliku podršku tijekom života, a posebice mojeg školovanja.

Hvala i mentoru doc.dr.sc. Marinu Golubu na zanimljivim predavanjima, te stručnom vodstvu kroz ovaj rad.

Posebno sam zahvalan Bogu za život, za sve prilike koje mi pruža kao i odgovornosti na koje me poziva.

Sve mogu u onome koji mi daje snagu. (Biblija, Filipljanima 4:13)

Sažetak

U ovom radu obrađeni su sigurnosni zahtjevi i uobičajene prijetnje s kojima se susreće računalni sustav. Opisane su preventivne mjere zaštite, najčešće korišteni mehanizmi, kroz pojedine razine OSI referentnog modela (*engl. open systems interconnection reference model*). Posebna pažnja posvećena je penetracijskom testu kao važnom postupku otkrivanja ranjivosti i vrednovanja sigurnosti sustava. U praktičnom djelu ovog rada izvršeno je ispitivanje ranjivosti računalnog sustava na više propusta raznih programa. Ispitivanju je podvrgnuto računalo bez zaštite, te računalo koje je štićeno sigurnosnom stijenom i mehanizmom za otkrivanje napada prelijevanja spremnika. U svrhe ispitivanja razvijena je aplikacija koja ima mogućnost automatiziranog ispitivanja ranjivosti računalnog sustava i jednostavno dodavanje novih testova.

Abstract

The security requirements and the common threats that computer system meets are handled in this work. The preventive security measures, most commonly used mechanisms, are described through individual levels of open systems interconnection reference model. Special attention is dedicated to the penetration test as an important procedure in vulnerability detection and system security evaluation. Testing of the computer system vulnerabilities on more failures of various programs is done in the practical part of this work. Computer without protection, and a computer that is protected with firewall, and the mechanism to detect buffer overflow attacks were tested. For the purpose of testing, application that has the automated possibility of testing the vulnerability of computer system and simply adding new tests is developed.

Sadržaj

1. Uvod

Zahvaljujući stalnom razvoju, informacijske tehnologije danas rješavaju najrazličitije zadatke, od automatizacije složenih procesa do svakodnevne uporabe u najširem krugu korisnika. Svaki je računalni sustav podložan programerskim pogreškama, a većina takvih grešaka ima kritične posljedice. Spomenute pogreške su veoma rasprostranjene na računalima u Internet prostoru. Motivirani napadači veoma lako pronalaze takve propuste i jednako lako ih iskorištavaju.

Ne treba zanemariti da složenost sustava zahtjeva pažljivu konfiguraciju, a pogreške u jednom sustavu mogu se odraziti u drugim sustavima. Propusti li se pravovremeno prepoznavanje i otklanjanje računalne ranjivosti na svim sustavima, a naročito onim koji se nalaze na Internetu, cjelokupni sustav se izlaže velikim rizicima.

Informacijska sigurnost podrazumijeva zaštitu informacija i informacijskih sustava od neovlaštenog pristupa, korištenja, objavljivanja, poremećaja, preinake ili uništenja. Vlada, vojska, financijske institucije, bolnice, privatna poduzeća i slične organizacije posjeduju velike količine tajnih informacija o njihovim zaposlenicima, klijentima, proizvodima, istraživanjima i financijskim stanjem. Danas se mnoge od tih informacija prikupljaju, obrađuju i pohranjuju u elektroničkom obliku na računalima i odašilju se putem mreže na druga računala. Dopadnu li informacije o poslovnim klijentima, financijama, novoj proizvodnoj liniji u ruke konkurenta, takvo narušavanje sigurnosti može prouzročiti gubitak poslovanja, pravne tužbe ili čak stečaj tvrtke. Zaštita tajnih podataka je poslovni zahtjev, a također i etički zahtjev.

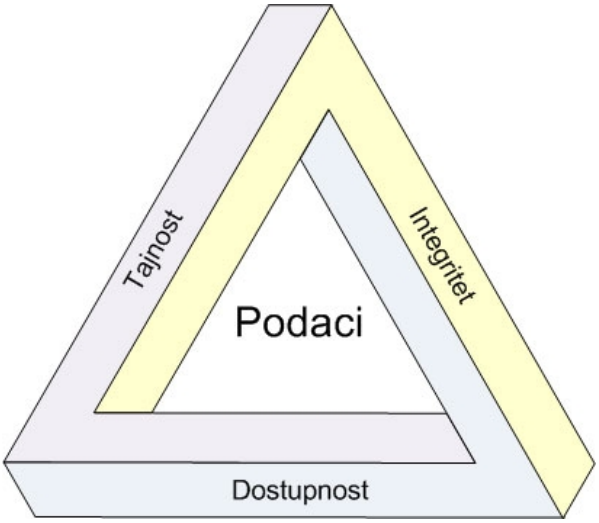
Važno je spomenuti da se godišnje zabilježi nekoliko tisuća programskih ranjivosti na komercijalnim ili javno dostupnim programskim paketima. Ako se tome pribroje mnoge konfiguracijske pogreške koje se nalaze na istim resursima, očito je da će održavanje sigurnosti računalnih sustava biti zahtjevniji proces što je više ovakvih propusta prisutno na sustavu.

Otkrivanje ranjivosti je prvi korak u procesu ispitivanja sigurnosti sustava. Svi sigurnosni standardi i brojni dokumenti o dobroj brizi o računalnim resursima najozbiljnije preporučuju redovitu provjeru sigurnosti računalnih resursa kako bi se na vrijeme prepoznale i otklonile ranjivosti. No, prije nego se krene u proces otkrivanja ranjivosti i zaštite sustava važno je vrlo dobro upoznati se sa sigurnosnom problematikom, prijetnjama i zahtjevima.

U drugom poglavlju pobliže su opisana tri sigurnosna zahtjeva: tajnost, integritet i dostupnost podataka. Sigurnosne prijetnje opisane su u trećem poglavlju kroz pobliže upoznavanje tipova napadača, ciljeva i metoda napada kao i nekoliko najčešćih napada na mreži. Četvrto poglavlje usredotočeno je na preventivne mjere zaštite, najčešće korištene mehanizme zaštite kroz različite razine OSI referentnog modela. Peto poglavlje posvećeno je penetracijskim testovima. Izloženi su osnovni razlozi zašto je preporučljivo sigurnosni sustav podvrgavati ovakvim testovima. Također je opisana taksonomija kao i proces izvođenja penetracijskog testa, dok se na kraju poglavlja navode i određeni zahtjevi penetracijskog testa sa stajališta organizacije, ispitivača, tehnike i etike. Šesto poglavlje opisuje pobliže izrađenu aplikaciju, te postupak testiranja aplikacije na dva tipa računala; bez zaštite i sa primijenjenom zaštitom.

2. Sigurnosni zahtjevi

Svaka organizacija posjeduje povjerljive podatke koje želi zadržati sigurnima. Kako bi podatak bio siguran, a ujedno i koristan ovlaštenim osobama, potrebno je u svakom trenutku zadovoljiti tri sigurnosna zahtjeva. Slika 2.1. prikazuje sigurnosne zahtjeve kao i njihovu međusobnu povezanost. To su zahtjev za tajnošću podataka (*engl. data confidentiality*), zahtjev za integritetom podataka *(engl. data integrity)* i dostupnost podataka *(engl. data availability)*.



Slika 2.1. Sigurnosni zahtjevi

Ovisno o primjeni i kontekstu korištenja jedan od ovih zahtjeva može biti važniji nego preostala dva. Na primjer, državna uprava će zaštititi elektronički poslane dokumente kriptiranjem kako bi spriječila neovlaštenoj osobi čitanje sadržaja tih dokumenata. Stoga je tajnost informacije u ovom slučaju najvažnija. Ukoliko pojedinac uspije otkriti sadržaj kriptiranog teksta te ponovo pošalje izmijenjenu kriptiranu verziju dokumenta narušuje integritet poslane poruke. S druge pak strane velike organizacije koje posluju putem Interneta biti će značajno oštećene ukoliko je njihova mreža izvan funkcionalnosti određeni vremenski period. Stoga je dostupnost ključni zahtjev takvih organizacija.

Dok rizik koji se javlja u ovim kategorijama ovisi o pojedinom kontekstu, općenito pravilo je da su ljudi najslabija sigurnosna veza. To je upravo razlog zašto su sposobnost i volja svakog pojedinog korisnika da koristi informacijski sustav na siguran način kritični argumenti.

2.1. Tajnost podataka

Tajnost se odnosi na ograničenje pristupa i otkrivanja informacijama samo ovlaštenim korisnicima. Drugim riječima sprečavanje pristupa i otkrivanja bilo kakvih informacija neovlaštenim pojedincima ili sustavima. Kako bi zahtjev tajnosti bio ostvaren, koriste se ponajprije različite metode autorizacije korisnika. Primjer te metode je korištenje korisničkog imena i lozinke koja jedinstveno identificira svakog korisnika. Također se koriste i kontrolne metode koje ograničuju svakom pojedinom korisniku pristup samo dozvoljenim podacima

sustava, kao i metode kriptiranja kako bi se zaštitila direktna informacija podataka koji putuju mrežom.

Na primjer, transakcija kreditnom karticom putem Interneta zahtijeva da se broj kreditne kartice pošalje od kupca prema trgovcu i od trgovca prema transakcijsko procesnoj mreži. Sustav nastoji očuvati tajnost podataka kriptiranjem broja kreditne kartice tijekom transmisije. Također je ograničen broj lokacija gdje se taj broj može pohraniti (u bazi podataka, dnevničkoj datoteci, sigurnosnoj kopiji, ispisanom računu, itd.), te se ograničava pristupa takvim mjestima neovlaštenoj osobi. Ukoliko neovlaštena strana zadobije broj kreditne kartice na bilo koji način, dogodilo se narušavanje tajnosti.

Narušavanje tajnosti moguće je izvršiti na različite načine. Jedan od oblika koji narušuje tajnost je nepažnja zaposlenika organizacije. Zaposlenik dozvoljava da netko drugi gleda na zaslon njegovog računala na kojem se trenutno nalaze povjerljivi podaci. Krađa ili prodaja prijenosnog računalo koje sadrži osjetljive podatke o organizaciji, zaposlenicima, također može završiti kao posljedica narušavanja tajnosti. Dijeljenje tajnih informacija preko telefona je također čin narušavanja tajnosti ukoliko osoba koja prima informacije nije za njih ovlaštena.

Tajnost je nužna (ali ne i dovoljna) kako bi privatnosti podataka koje sustav pohranjuje ostala očuvana.

2.2. Integritet podataka

Značenje integriteta podataka odnosi se na potpunost i nepromjenjivost, odnosno podrazumijeva da se podaci ne mogu izmijeniti bez prethodnog ovlaštenja. Integritet je narušen kada zaposlenik (bilo slučajno ili zlonamjerno) obriše važne podatke, kada računalni virus zarazi i uništi podatke, kada je zaposlenik u mogućnosti izmijeniti iznos vlastite plaće u bazi, kada neovlašteni korisnik uništava neku Internet stranicu, kada je netko u mogućnosti pridijeliti iznimno velik broj glasova na Internet anketi, itd.

2.3. Dostupnost podataka

Dostupnost se odnosi na dostupnost informacijskih resursa. Informacijski sustav koji nije dostupan upravo kad je informacija potrebna je loš u istoj mjeri kao da ga i nema. Može biti i mnogo gore, ovisno o tome koliko je organizacija vezana za tu računalnu i komunikacijsku infrastrukturu. Danas su gotovo sve moderne organizacije visoko ovisne o funkcioniranju nekog informacijskog sustava. Mnoge doslovce ne mogu poslovati bez njih.

Dostupnost, kao i drugi sigurnosni aspekti, može biti zahvaćena mnogim tehničkim problemima (neispravnost dijelova računala ili komunikacijskih uređaja), prirodnim pojavama (poplave, nevrijeme) ili pak ljudskim pogreškama (slučajnim ili namjernim).

3. Sigurnosne prijetnje

3.1. Profil napadača

U širem smislu, termin napadač se koristi za bilo koju osobu koja se upliće u neki sigurnosni sustav bez prethodne autorizacije. No, unatoč tome, postoji jasna razlika između dobronamjernog napadača (*engl. hacker*), zlonamjernog napadača (*engl. crackera*) i djetinjastog napadača *(engl. script kiddie)*.

Dok se dobronamjernim napadačima smatraju eksperimentalno usmjereni programeri koji napadaju sigurnosne propuste informacijskog sustava iz tehničkih razloga, zlonamjerni napadači su ljudi kriminalne tenzije koji napadaju slabe točke informacijskog sustava kako bi zadobili ilegalnu premoć, društvenu pozornost ili poštovanje. Djetinjasti napadači su najčešće napadači ograničenog znanja i vođeni su radoznalošću napada na proizvoljne ili istaknute sustave. Pri tom koriste napadačke alate koji se mogu pronaći na Internetu.

Zlonamjerni napadači koji posjeduju povlašteno znanje o organizaciji koju napadaju nazivaju se unutarnji napadači (*engl. insiders*). To su najčešće nezadovoljni zaposlenici organizacije koji koriste njihovo znanje u unutarnjim aferama kako bi naudili organizaciji. Šteta zadana sa strane unutarnjeg napadača je osobito velika jer su oni najčešće upoznati s tehničkom i organizacijskom infrastrukturom i moguće da već unaprijed poznaju postojeće ranjivosti sustava.

Uz ove kategorije napadača, industrijska špijunaža također posjeduje ozbiljne prijetnje. Cilj industrijske špijunaže jest zadobiti znanje poslovnih tajni kao što su inovativna tehnička rješenja, strategije i ideje koje pomažu u postizanju konkurentne prednosti koristeći takve informacije u osobnu korist.

3.2. Vrste napada prema njihovom cilju

Napad je radnja u kojoj pojedinac ili grupa pojedinaca pokušava neovlašteno pristupiti, modificirati ili oštetiti neki sustav ili okolinu. Napadi mogu biti bilo jednostavni ili složeni, bilo usmjereni ili neusmjereni. Bitno je naglasiti da se mogu pojaviti potpuno neočekivano izrazito snažnim intenzitetom.

Napadi na mrežne resurse postali su uobičajeni u današnjem svijetu koji je ovisan o Internetu. Napadi se pokreću iz različitih razloga; uključujući novčanu dobit, zlonamjernost kao izazov, prijevaru, rat, izazivanje terorističkih sukoba ili pak zadobivanje ekonomske prednosti. Napadi su usmjereni na kompromitiranje tajnosti, integriteta i dostupnosti mreža kao i njihovih resursa.

Općenito se svaki napad prema cilju koji nastoji ostvariti može svrstati u jednu od sljedeće tri kategorije:

- **Napad pristupa** (*engl. access attack*) je napad u kojem napadač želi zadobiti pristup povjerljivim resursima za koje nije ovlašten.
- **Napad modifikacije** (*engl. modification attack*) je napad u kojem napadač želi prisilno izmijeniti informacije u sustavu za koje nije ovlašten.

- **Napad uskraćivanjem usluge** (*engl. denial of service attack*) je napad u kojem napadač želi poremetiti i onemogućiti rad mreže, usluga, sustava u cjelini.

3.2.1. Napad pristupa

Napadom pristupa pokušava se zadobiti pristup informacijama za koje napadač nije autoriziran da ih posjeduje. Ovakav tip napada usmjeren je na dobivanje tajnih informacija, a ostvaruje se bilo kroz unutarnji ili vanjski pristup. Ovaj napad se također pojavljuje i u situacijama kada je dostupan fizički pristup do informacija.

Pretraživanje odbačenog materijala

Pretraživanje odbačenog materijala (*engl. dumpster diving*) je vrlo česta fizička metoda napada pristupa. Organizacije generiraju velike količine dokumenata u normalnom slijedu događaja. Većina informacija naposljetku završi u otpadu ili pak reciklaži. Taj materijal može sadržavati informacije koje su po prirodi vrlo osjetljive. U mnogim slučajevima, informacije pronađene među odbačenim materijalom mogu biti vrlo vrijedne za napadača. Odbačene informacije mogu uključivati tehnička uputstva, liste lozinka, telefonske brojeve kao i razne organizacijske dijagrame. Važno je napomenuti da informacija sa samo jednim zahtjevom tretiranja dotične kao trgovačke tajne zahtjeva da bude zaštićena i nedostupna neautoriziranim pojedincima. Ukoliko dokument posjeduje organizacijske trgovačke tajne informacija te je iz nepažnje odbačen može biti pronađen od druge osobe. Ta druga osoba može iskoristiti nađenu informaciju zbog njezine neadekvatne zaštite. Stoga se u okruženjima visoke sigurnosti, npr. okruženju državne vlade, osjetljivi dokumenti bilo režu ili čak i spaljuju. No, ipak većina poduzeća to ne radi i izlaže se riziku.

Prisluškivanje

Druga uobičajena metoda koja se koristi za napad pristupa je metoda prisluškivanja (*engl. eavesdropping*). Prisluškivanjem se hvataju informacije na putu između dva sustava. To je proces bilo trenutnog prisluškivanja ili naknadnog preslušavanja dijelova razgovora. Prisluškivanje također uključuje i napadače koji prisluškuju mrežni promet. Ovaj tip napada je općenito pasivan. Moguće je da netko čuje plan neke osobe jer je primjerice zvuk telefonske slušalice prilikom razgovora preglasan. Prisluškivanje je pasivan proces u kojem prilika da se prisluškuje konverzacija ovisi o nepažnji strana koje komuniciraju.

Snooping

Snooping je metoda u kojoj napadač pregledava zabranjene podatke u nadi da će među njima naći neku važnu informaciju. Ovi podaci mogu biti pohranjeni bilo elektronski, bilo na papiru. U slučaju fizičkog njuškanja, moguće da će napadač pregledavati vaš koš za smeće ili pak vašu arhivu podataka. Računalno njuškanje pak s druge strane podrazumijeva nekoga tko pretražuje elektronske podatke pokušavajući naći među njima nešto interesantnog da iskoristi.

Presretanje

Presretanje (*engl. interception*) može biti bilo aktivan bilo pasivan proces. U mrežnim okruženjima, pasivno presretanje uključuje nekoga tko rutinski nadgleda mrežni promet. Aktivno presretanje pak podrazumijeva postavljanje računala između pošiljatelja i primatelja kako bi uhvatili i izmijenili informacije prije nego što stignu s izvorišta na odredište. S perspektive presretanja, taj proces je tajan. Posljednja stvar koju napadač u činu presretanja

želi je biti otkriven. Presretanje se stoga može događati godinama, bez da o tome išta znaju privatne strane koje komuniciraju tim kanalom.

Državne agencije rutinski koriste presretanje kako bi kontrolirali inteligenciju o mogućnostima i lokacijama neprijatelja

3.2.2. Napad modifikacije

Cilj napada modifikacije je izmijeniti sadržaj informacije na neovlašten način. Ovaj napad sličan je napadu pristupa iz razloga što prvenstveno zahtijeva pristup do informacija. Motivacija za ovakav tip napada može biti podmetanje informacija, kao primjerice promjena ocjena iz predmeta, nedozvoljena izmjena stanja kreditne kartice, ili bilo koji drugi razlog. Varijacija napada modifikacije je **napad odbacivanja autorstva** (*engl. repudiation attack*).

Napadi modifikacije uključuju brisanje, umetanje ili modificiranje informacije s namjerom da se ona krajnjem korisniku učini valjanom. Ovakvi napadi se vrlo teško otkrivaju. Općenito, napadač najprije ulazi u sustav primjenom napada pristupa, a zatim djeluje modifikacijom. Narušavanje i izmjena sadržaja Internet stranica najčešći je oblik napada izmjene.

Napad odbacivanja autorstva je djelo odbijanja sudjelovanja u nečem što se dogodilo. Napad odbijanja autorstva najčešće se događa kad aplikacija ili sustav ne kontrolira i ne bilježi aktivnosti korisnika, te na taj način omogućuje napadaču zlonamjernu manipulaciju ili krivotvorenje identifikacije novih akcija. Ovaj napad napadač također može iskoristiti da izmijeni autorske informacije radnji koje je izvodio s namjerom da zabilježi lažne podatke u dnevničke datoteke. Primjer ovog napada obično uključuje klijenta koji tvrdi da nikad nije dobio uslugu koja mu je bila naplaćena. U ovakvoj situaciji, teret dokaza je na trgovcu da pokaže da je informacija o nenaplaćenom računu točna. Ukoliko je podatak modificiran od napadača, verifikacija točnosti informacije može biti otežana.

Ovaj tip napada također obuhvaća radnje kojima napadač predstavlja informaciju zbunjujućom. Primjer takvog čina može biti akcija napadača koji zabranjeno pristupa poslužitelju elektronske pošte i izazivački šalje informacije drugima. Ovakva informacija može učiniti štetu i sramotu osobi koja je prima ili pak i čitavom poduzeću. Takvi napadi se poprilično jednostavno izvode jer većina poslužitelja elektronske pošte ne provjerava valjanost izlaznih pošiljki. Napad povrede obično započinje kao napad pristupa.

3.2.3. Napad uskraćivanjem usluge

Cilj napada uskraćivanjem usluge je preplavljivanje sistemskih resursa kako sustav ne bi mogao odgovoriti na zahtjeve koji su mu upućeni. Napad uskraćivanjem usluge može biti izveden preplavljivanjem poslužitelja vrlo velikim brojem simultanih veza tako da on više ne uspijeva odgovarati. Drugi način je da se računalnom sustavu zapuni čitav prostor za pohranu podataka prijenosom velikih datoteka na sustav.

Napad uskraćivanjem usluge uključuje sljedeće:

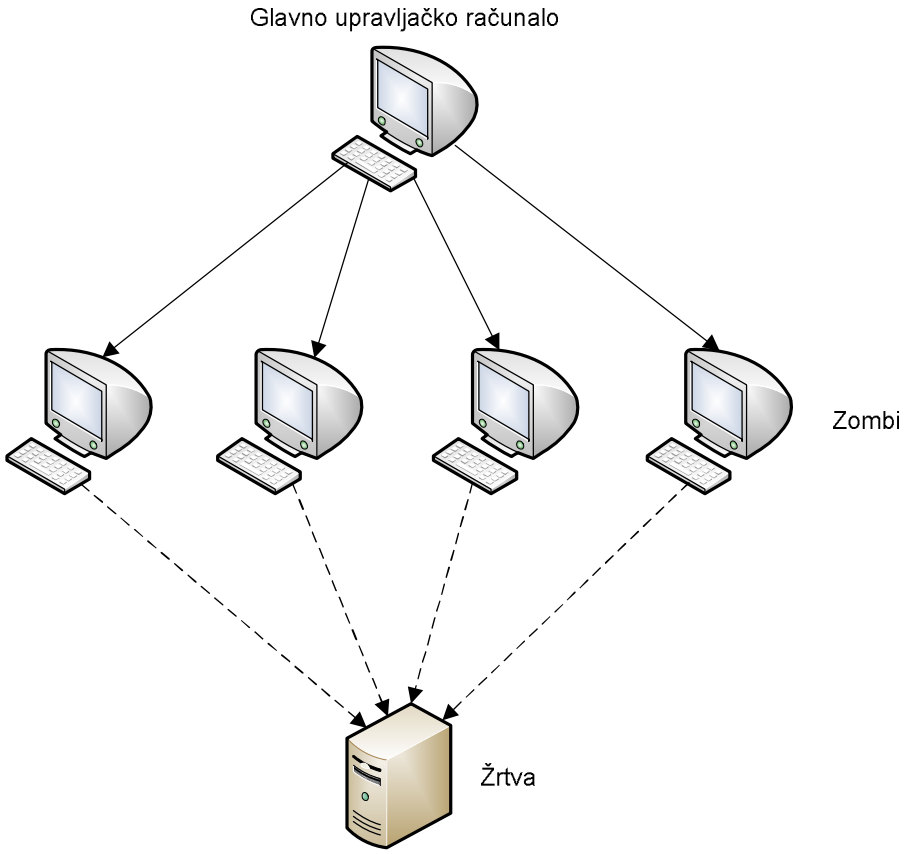
- **Prelijevanje spremnika** (*engl. buffer overflow*).
  Proces prima mnogo više podataka nego što ih očekuje. Ako proces nema proceduru da se nosi s tom prekomjernom količinom podataka, djelovat će na neočekivan način

koji napadač može iskoristiti. Na primjer, napad izobličenog signala (*engl. ping of death*) iskorištava Internet kontrolni protokol (*engl. Internet control message protocol*) tako da šalje ilegalan signalni paket veličine veće od 65K okteta podataka. Na taj način može uzrokovati prelijevanje sistemskih varijabli i prouzročiti nefunkcionalnost sustava.

- **SYN napad** (*engl. SYN attack*).
  U ovom napadu, napadač iskorištava korištenje spremničkog prostora (*engl. buffer*) za vrijeme inicijalizacije transmisijske sjednice (*engl. transmission control protocol session*). Napadač preplavljuje sistemski red sa mnogo zahtjeva za povezivanjem, ali ne odgovara na zahtjeve koje mu sustav šalje. Takav postupak uzrokovat će na napadnutom sustavu prekoračenje vremena čekanja na prikladan odgovor, te će sustav postati nefunkcionalan ili nedostupan.
- **Napad fragmentacijskog polja** (*engl. teardrop attack*).
  Napadač vrši izmjenu fragmentacijskog pomaka u sekvencijskim IP paketima. Ciljani sustav tada postaje zbunjen i nefunkcionalan u trenutku kad zaprimi kontradiktornu instrukciju kako su fragmenti pomaknuti.
- **Napad lažnim emisijskim signalom** (*engl. smurf attack*).
  Ovaj napad zahtjeva korištenje krivotvorenje IP adresa i Internet kontrolnog protokola kako bi preplavio mrežu prometom. Napadač na izvornoj lokacije šalje podvaljeni signalni paket na emisijsku adresu (*engl. broadcast address*) velike mreže na odredišnoj lokaciji. Taj modificirani paket sadrži krivotvorenu izvorišnu adresu koja je zamijenjena adresom ciljanog računala koje se napada. To će prouzročiti da odskočna lokacija emitira dezinformaciju prema svim uređajima na njenoj lokalnoj mreži, a zatim će svi uređaji odgovoriti ciljanom sustavu, koji tada postaje preplavljen odgovorima.

Srodan napad napadu uskraćivanjem usluge je distribuirani napad uskraćivanjem usluge, koji je također napad na mrežne resurse, s razlikom da se pokreće s velikog broja računala. Napadački program se instalira na veći broj računala i nepoznat je njihovim vlasnicima. Napadački program će na tim računalima ostati neaktivan sve dok ne dobije pokretački signal od glavnog računala. Taj signal okida takva računala koja zatim pokreću simultani napad na ciljanu mrežu ili sustav s tolikim intenzitetom da ih preplave. Slika 3.1. prikazuje događaj napada kao i glavno računalo koje upravlja tim napadom. Glavno upravljačko računalo može biti neko drugo nesumnjivo računalo. Računala koja primaju upute od glavnog upravljačkog računala nazivaju se zombiji (*engl. zombies*). Takav sustav obavlja jedino instrukcije koje su im dane od glavnog upravljačkog računala.

Najgori dio ovog tipa napada je da računala koja izvršavaju napad zapravo pripadaju običnim računalnim korisnicima. Napad ne daje nikakva posebna upozorenja tim korisnicima. Kada napad završi, napadački program može se automatizirano ukloniti sa sustava ili pak inficirati računalo virusom te uništiti tvrdi disk kako bi se uništili dokazi napada.



Slika 3.1. Distribuirani napad uskraćivanjem usluge

Općenito se vrlo malo toga može učiniti kako bi spriječili napad uskraćivanjem usluge ili distribuirani napad uskraćivanjem usluge. Najbolja metoda suočavanja s ovim vrstama napada uključuje prevenciju protumjerama. Mnogi operacijski sustavi su barem djelomično podložni ovim vrstama napada.

3.3. Metode napada

Postoji nekoliko načina manipuliranja ili oštećivanja informacijskog sustava kao i pripreme napada na informacijski sustav. To su sljedeće metode:

- Mrežno orijentirani napad
- Socijalno inženjerstvo
- Zaobilaženje fizičkih sigurnosnih mjera

3.3.1. Mrežno orijentirani napadi

Mrežno orijentirani napadi su napadi na mrežne komponente, računalne sustave i aplikacije koji koriste funkcionalnosti mrežnog protokola. Ovakav tip napada iskorištava ranjivosti ili neadekvatnost u sklopovlju i programskoj podršci obavljajući pripremu ili izvršavajući napad.

Mrežno orijentirani napadi podrazumijevaju skeniranje vrata (*engl. port scanning*), IP podvalu (*engl. IP spoofing*), njuškanje (*engl. sniffing*), otimanje sjednica (*engl. session*

hijacking), napade uskraćivanjem usluge (*engl. DoS attacks*), prelijevanje spremnika (*engl. buffer overflow*), napad oblikovanim nizom (*engl. format string attack*), kao i svako drugo iskorištavanje ranjivosti u operacijskom sustavu, aplikaciji i mrežnom protokolu koje se odvija putem mreže.

3.3.2. Socijalno inženjerstvo

U napadu socijalnog inženjerstva napadač pokušava manipulirati ljudima sa privilegiranim znanjem kako bi iskoristio ljudsku slabost i zadobio sigurnosne informacije kao što su lozinke. Na primjer, napadač se može pretvarati kao da je zaposlenik organizacije i zavarati neočekivanog korisnika da mu otkrije svoju lozinku. Raspon mogućih scenarija ovakvih napada je poprilično širok. U širem smislu, socijalno inženjerstvo pokriva situacije u kojima su sigurnosne informacije stečene protuzakonitim utjerivanjem.

3.3.3. Zaobilaženje fizičkih sigurnosnih mjera

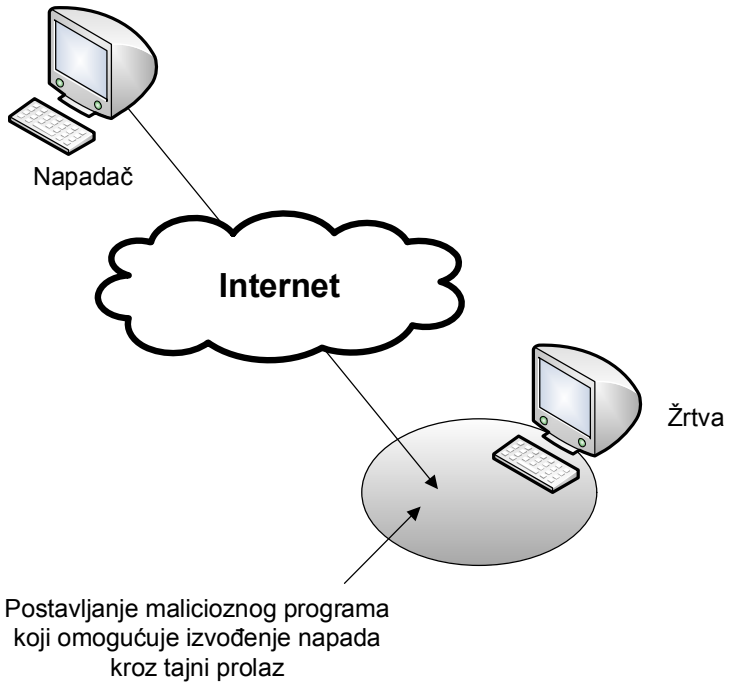
Ne smije se dogoditi da informacijska sigurnost tehničke infrastrukture ostane bez fizičke sigurnosti. Ukoliko se fizičke sigurnosne mjere mogu nadjačati i ako je zadobiven fizički pristup sigurnosnom sustavu, najčešće je samo stvar vremena kad će se napad ili manipulacija pohranjenih aplikacija i podataka uspješno dogoditi. Primjer ovog napada je neautorizirani ulaz u računalni centar organizacije i otimanje tvrdog diska na kojem su pohranjeni povjerljivi podaci. Ova kategorija također uključuje slučajno pretraživanje dokumenata sa sigurnosnim informacijama.

3.4. Uobičajeni napadi

Uobičajeni napadi opisani u daljnjem tekstu iskorišćuju potencijalne slabosti u implementaciji aplikacija kao i slabosti mrežnih protokola. Mnogi od ovih napada visokog su stupnja složenosti i stoga su poprilično rijetki.

3.4.1. Napad kroz tajni prolaz

Napad kroz tajni prolaz (*engl. backdoor attack*) koristi maliciozni računalni program koji napadač postavlja na ciljano računalo. Kada je maliciozni program uspješno postavljen i pokrenut napadač ga komunicirajući s njim iskorištava za zaobilaženje normalnog postupka autorizacije. Maliciozan računalni program koji omogućuje tajni prolaz izvršava se u pozadini i skriven je od korisnika. Vrlo je sličan računalnom virusu i stoga ga je poprilično teško otkriti i onemogućiti u potpunosti. Ovi maliciozni programi jedni su od najopasnijih računalnih parazita jer napadaču omogućuju da izvede gotovo bilo kakvu akciju na kompromitiranom računalu. Ovaj napad obično je napad pristupa ili napad modifikacije. Napadač može iskoristiti ovakav prolaz u računalo za špijunažu korisnika, kontrolu datotečnog sustava, instaliranje dodatnih malicioznih programa, kontrolu čitavog sustava uključujući svu programsku podršku i sklopovlje koje računalo posjeduje. Također se ovakvo računalo može iskoristiti za izvođenje napada na druge sustave. Slika 3.2. pokazuje kako se napad kroz tajni prolaz može iskoristiti u zaobilaženju mrežne sigurnosti.



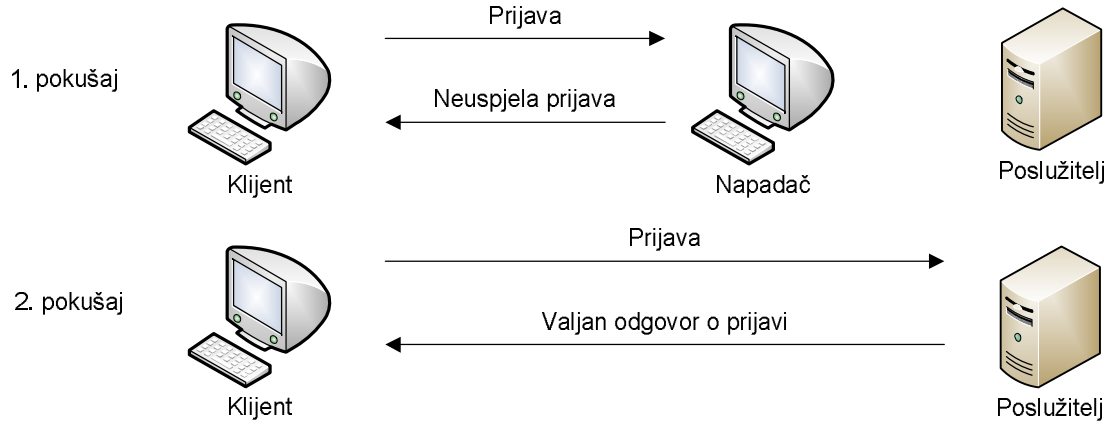
Slika 3.2. Postupak napada kroz tajni prolaz

Na kraju maliciozne programe koji otvaraju tajni prolaz napadač može iskoristiti za sakrivanje svojih tragova. Ukoliko napadač ne uspije dobiti neku vrijednu, korisnu informaciju s inficiranog računala on može uništiti funkcionalnost čitavog sustava upravo iz razloga da prikrije svoje radnje. To prvenstveno podrazumijeva nepovratno brisanje podataka pohranjenih na tvrdom disku.

Postoji velik broj alata za iskorištavanje propusta koji omogućuju napad kroz tajni prolaz. Jedni od najpoznatijih alata koji se koriste u ove svrhe su *Back Orifice* i *NetBus*.

3.4.2. Napad krivotvorenjem

Napad krivotvorenjem (*engl. spoofing attack*) je jednostavan pokušaj nekog ili nečeg da se lažno predstavi. Ovaj tip napada uobičajeno podrazumijeva napad pristupa. Vrlo uobičajen napad krivotvorenjem koji je bio popularan mnogo godina svodio se na pisanje lažnih sučelja za prijavu. Taj program bi postavljao korisniku upit u kojem bi on unosio svoje korisničko ime i lozinku. Bez obzira što bi korisnik upisao program bi dojavio nevaljan pokušaj prijave, a zatim bi preusmjerio kontrolu na stvarano sučelje za prijavu. Program bi prilikom unosa pohranio korisničko ime i lozinku koju bi napadač kasnije iskoristio. Ovaj napad bio je vrlo popularan na ranim višekorisničkim sustavima. Slika 3.3. pokazuje napad krivotvorenjem koji se pojavljuje kao dio procesa za prijavu na računalnu mrežu. Napadač u ovoj situaciji ima ulogu poslužitelja za klijenta koji se pokušava prijaviti. Neovisno o tome što klijent pokuša učiniti, lažan sustav neće uspješno provesti postupak autorizacije korisnika na mrežu. Kada taj proces završi lažan sustav će raskinuti vezu s klijentom, a klijent se nakon toga prijavljuje na valjani poslužitelj. Napadač na ovaj način dobiva valjano korisničko ime i lozinku korisnika.



Slika 3.3. Napad krivotvorenjem prijave

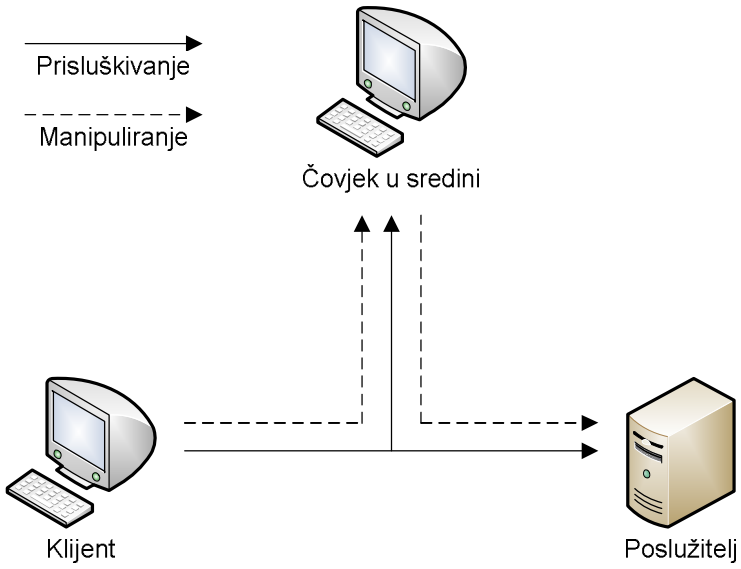
Napad krivotvorenjem najčešće završava prijevarom nekoga ili nečeg u mišljenju da se dogodilo nešto sasvim legitimnog.

3.4.3. Napad s čovjekom u sredini

Pretpostavimo da klijent želi komunicirati s poslužiteljem. U međuvremenu, napadač (čovjek u sredini) želi prisluškivati komunikaciju i po mogućnosti dostaviti lažnu poruku poslužitelju. Kako bi započeli, klijent mora upitati poslužitelja za njegov javni ključ. Ukoliko poslužitelj pošalje svoj javni ključ klijentu, a napadač ga je pritom presreo započeo je napad s čovjekom u sredini (*engl. man in the middle attack*). Napadač zatim šalje klijentu krivotvorenu poruku sa svojim javnim ključem, te tvrdi da je poruka pristigla od poslužitelja. Klijent vjerujući da taj javni ključ pripada poslužitelju kriptira svoju poruku javnim ključem napadača. Tako kriptiranu poruku klijent šalje natrag poslužitelju. Napadač ponovo presreće i dekriptira poruku, zadržava kopiju, a poruku ponovo kriptira ispravnim poslužiteljevim javnim ključem. Kad poslužitelj primi ovu poruku bit će uvjeren da je ona poslana od klijenta.

Napad s čovjekom u sredini je oblik aktivnog prisluškivanja u kojem napadač stvara neovisnu vezu sa žrtvama i prenosi poruke između njih. Žrtve su uvjerene da komuniciraju direktno jedna s drugom putem privatne veze kad zapravo cijelu komunikaciju kontrolira napadač. Napadač mora biti u mogućnosti presresti sve poruke između dviju žrtava i ubaciti novu poruku.

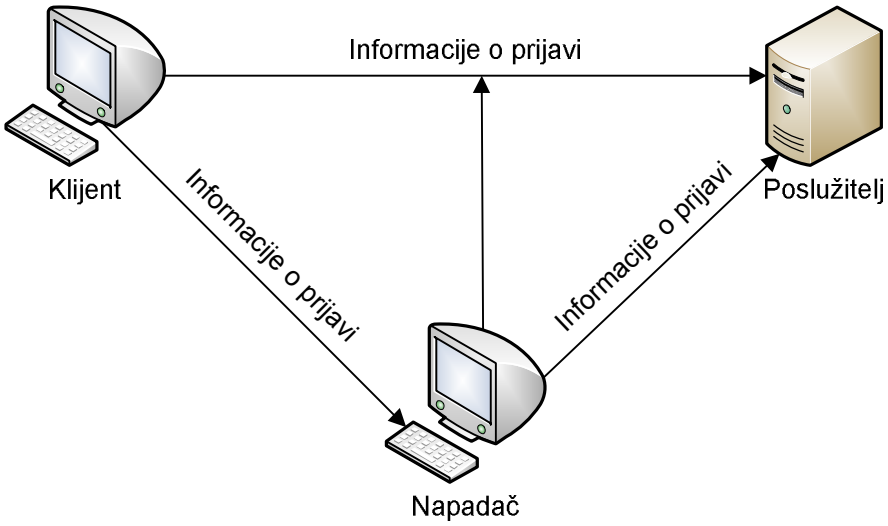
Napad s čovjekom u sredini pojavljuje se u dva oblika. Jedan oblik je prisluškivanje dok je drugi manipuliranje podacima. Prisluškivanje (*engl. eavesdropping*) je radnja u kojoj napadač prima niz komunikacijskih podataka. To nije toliko direktni napad već više curenje informacija. Prisluškivač može pohraniti i analizirati podatke koje prisluškuje. Manipuliranje (*engl. manipulation*) naspram prisluškivanja zahtjeva da napadač ne prima samo podatke već da ih je sposoban i izmijeniti i proslijediti dalje. Slika 3.4. prikazuje ova dva oblika napada s čovjekom u sredini.



Slika 3.4. Napad s čovjekom u sredini - prisluškivanje i manipuliranje

3.4.4. Napad ponovnim slanjem paketa

Napad ponovnim slanjem paketa (*engl. replay attack*) je napad u kojem napadač presreće i pohranjuje poruke te ih zatim pokušava poslati kasnije, oponašajući jednog od sudionika. Napad ponovnim slanjem paketa najčešće se koristi kao napad pristupa. U raspodijeljenoj okolini informacije o korisničkim imenima i lozinkama često se šalju između klijenata i autorizacijskog sustava. Napadač može uhvatiti takve informacije te ih kasnije ponovo poslati kako bi zadobio pristup sustavu. Ovo se također može dogoditi i sa sigurnosnim certifikatima sustava kao što je *Kerberos*. Slika 3.5. prikazuje napadača koji lažno predstavlja prethodno uhvaćen certifikat autorizacijskom poslužitelju. U ovom slučaju napadač prethodno presreće legitimne informacije, certifikat klijenta, te ih pohranjuje. Nakon toga napadač pokušava koristiti takve informacije kako bi ušao u sustav.



Slika 3.5. Napad ponovnim slanjem paketa

Ukoliko ovakav napad uspije, napadač će zadobiti sva prava i privilegije koje posjeduje vlasnik originalnog certifikata. Ovo je osnovni razlog zašto većina certifikata sadrži jedinstveni identifikator sjednice kao i vremenski žig. Ukoliko je valjanost certifikata zastarjela, takav certifikat će se odbaciti.

3.4.5. Napad pogađanja lozinke

Iz razloga što je korištenje lozinke najčešći mehanizam prijave korisnika na informacijski sustav napad pogađanja lozinke (*engl. password guessing attack*) je učestali napadački pristup. Napad pogađanja lozinke primjenjuje metodu opetovanog slanja lozinki prema autorizacijskom sustavu koji se napada. To su lozinke koje napadač generira bilo korištenjem grube sile (*engl. brute force attack*) ili pak korištenjem rječnika (*engl. dictionary attack*).

Napad grubom silom je pokušaj sistematičnog pogađanja lozinke sve dok se ne dogodi uspješan ishod. Ovakav napad obično se izvodi vrlo dugi period. Stoga bi lozinke trebale biti dovoljno dugačke kako bi ih bilo što teže pogoditi i kako bi spriječili ovakve napade.

Napad rječnikom je napad koji koristi rječnik uobičajenih riječi kako bi pokušao pronaći lozinku nekog korisnika.

Ovakvi napadi mogu biti automatizirani, te postoji velik broj alata za njihovo provođenje.

Važno je napomenuti da će neki sustavi dojaviti korisniku situacije u kojima je korisničko ime valjano, a lozinka neispravna. Na taj način mogu dati napadaču do znanja koristi li ispravno korisničko ime prilikom pogađanja lozinke. Stoga je bolje da sustav prihvaća samo ispravan par korisničkog imena i lozinke te u slučaju neispravnog unosa zahtijeva ponovo cijeli proces prijave.

4. Preventivne mjere zaštite

4.1. Razine preventivnih mjera zaštite

Svaka organizacija koja želi efikasno štititi privatnost podataka, zaštiti mrežu od virusa, crva i drugih složenih napada mora koristiti veći broj sigurnosnih tehnologija.

Mrežna sigurnosna tehnologija može se općenito podijeliti u četiri kategorije:

- Zaštita na razini paketa
- Zaštita na razini sjednice
- Zaštita na razini aplikacije
- Zaštita na razini datoteke

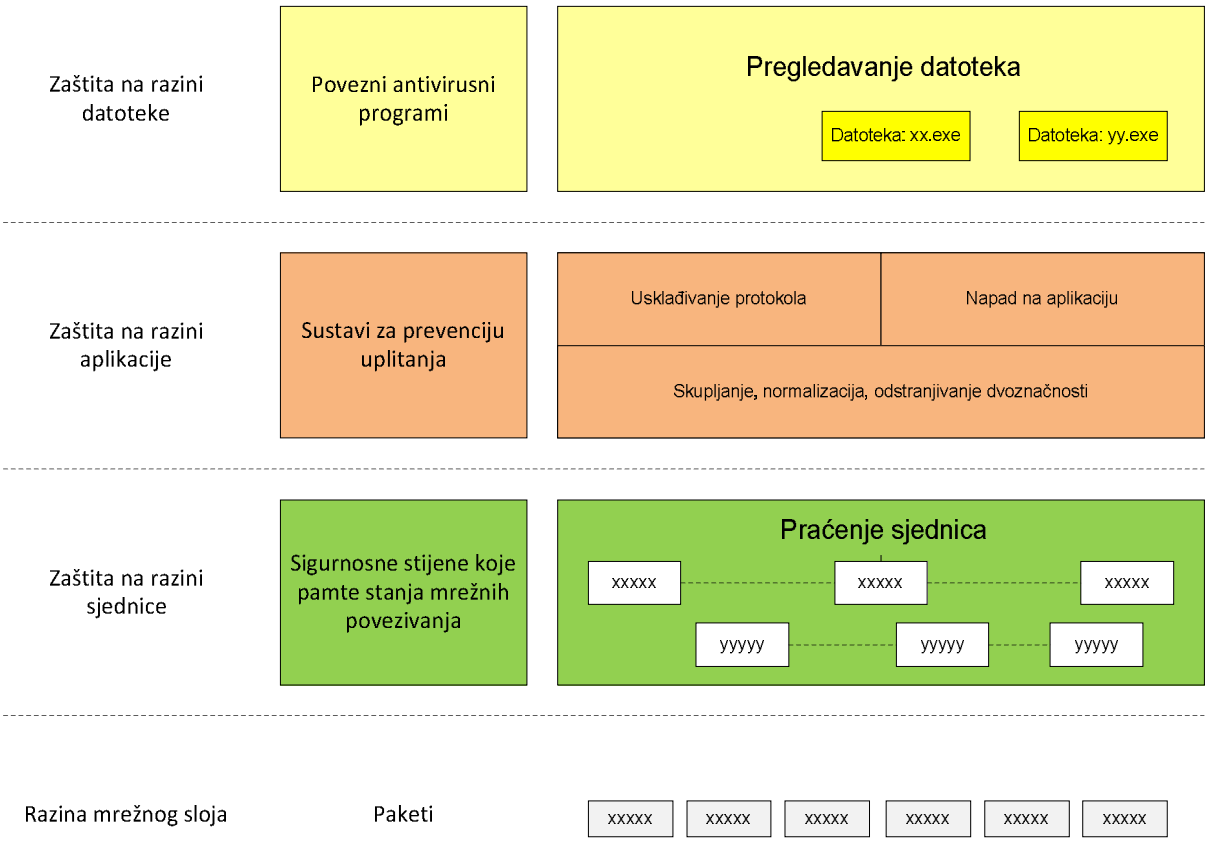
Mrežna sigurnosna tehnologija zaštite na razini paketa, na razini sjednice, razini aplikacije i razini datoteke daje niz preventivnih mjera koje podupiru sigurnost mrežnog sustava. Ulaganje u sve ove razine zaštite je nužno za većinu organizacija kako bi njihova računalna mreža bila efikasno zaštićena.

Tablica 4.1. uspoređuje četiri kategorije mrežne sigurnosne tehnologije. Vrednovanje svake kategorije na temelju protokola i aplikacija koje obuhvaća, obliku zaštite te relativnim performansama omogućuje organizaciji da odabere odgovarajuću mrežnu sigurnosnu tehnologiju kako bi zaštitila svoju mrežu.

Tablica 4.1. Usporedba kategorija zaštite

	Zaštita na razini paketa	**Zaštita na razini sjednice**	**Zaštita na razini aplikacije**	**Zaštita na razini datoteke**
Primjer	Filtriranje paketa (ACL liste na usmjerivačima, sigurnosne stijene bez pamćenja stanja paketa)	Sigurnosne stijene koje pamte stanja paketa	Sustavi za prevenciju uplitanja i posredničke sigurnosne stijene	Povezni antivirusni programi
Mehanizam	Pregledava se zaglavlje paketa	Pregledava se zaglavlje paketa i kontrolna polja	Pregledavaju se aplikacijska polja	Pregledava se datoteka unutar aplikacijskog prometa
Obuhvaćeni protokoli i aplikacije	Nikakvi (razina paketa)	Mnogo njih	Srednji broj	Malen (elektronička pošta, web i prijenos datoteka)
Zaštita	Klijent prema klijentu i poslužitelj prema klijentu	Klijent prema poslužitelju i poslužitelj prema klijentu	Većinom klijent prema poslužitelju	Većinom poslužitelj prema klijentu
Relativne performanse	Visoke	Visoke	Srednje	Niske

Slika 4.1. ilustrira nadzorne funkcije koje se izvršavaju prilikom analize paketa. Sigurnosne stijene koje pamte zapise stanja mrežnih povezivanja analiziraju pakete na sesijskoj razini, sustav za prevenciju uplitanja na aplikacijskoj razini, a povezni antivirusni programi na razini datoteke.



Slika 4.1. Ilustracija nadzornih funkcija koje se izvršavaju prilikom analize paketa

4.1.1. Zaštita na razini paketa

Zaštita na razini paketa, također poznata i kao filtriranje paketa jedna je od najčešće korištenih mjera kontrole i zaštite pristupa na mrežu. Koncept je poprilično jednostavan. Na temelju informacija zaglavlja paketa određuje se da li se pristigli paket smije ili ne smije propustiti na mrežu. *Cisco IOS Access Control List* je jedan od najviše korištenih sustava za filtriranje paketa. *IPChains* je također popularan program za filtriranje paketa koji dolazi kako sastavni dio gotovo svih verzija Linuxa.

Dvosmjerna mrežna komunikacija stvara mnoge izazove kao i probleme za mrežnu sigurnost baziranu na filtriranju paketa. Ukoliko se primjerice blokira sav dolazni promet, time su posljedično spriječeni i odgovori vanjskih sustava na upite postavljene od unutarnjih izvora. Također je posljedica ovog pristupa i pojavljivanje dvostrukog propusta, bilo za odlazni bilo za dolazni promet. Sigurnosne stijene koje ne pamte stanja paketa analiziraju svaki paket zasebno. Sve se to događa na mrežnom sloju OSI referentnog modela. No, većina veza koristi TCP protokol, koji koristi sjednice, s namjerom kako bi se razmjena podataka odvijala

pouzdanije. Stoga se samo filtriranjem paketa može dogoditi propust prolaska malicioznih paketa koji su dio neke sjednice te time uzrokovati štetu štićenih resursa.

Sustavi za filtriranje paketa koji ne pamte stanja paketa također ne prate dinamičke protokole gdje se poslužitelj i klijent upravo dogovaraju o uspostavi sjednice. Mnoge usluge iniciraju vezu na statičkim vratima, ali dinamički otvaraju vrata kako bi ostvarili sjednicu. Neki protokoli koji koriste dinamička vrata jesu FTP (*engl. file transfer protocol*), RPC (*engl. remote procedure call*) i H.323 (protokol za audio i vizualnu komunikaciju). Kako bi omogućili ovim aplikacijama da mogu normalno obavljati komunikaciju kroz sustav za zaštitu paketa bez pamćenja njihovog stanja, uzrokovat će se veliki propust koji vrlo drastično smanjuje sigurnosnu zaštitu sustava za filtriranje paketa. Kako bi primjerice omogućili standardni FTP protokol, potrebno je dozvoliti sav promet na odredišnim vratima većim od 1023 i izvorišnim vratima 20. Na taj način otvara se veliki sigurnosni propust na mreži.

4.1.2. Zaštita na razini sjednice

Zaštitom na razini sjednice kontrolira se tok podatak između dviju ili više mreža. Praćenjem stanja sjednice odbacuju se oni paketi koji nisu dio sjednice i ne zadovoljavaju kriterije prethodno definirane sigurnosne politike. Sigurnosne stijene koje implementiraju zaštitu na razini sjednice čuvaju informacije o stanju svake mrežne sjednice te donose odluku o dozvoli ili zabrani temeljenu na toj tablici stanja. Najčešći sustavi za zaštitu na razini sjednice su sigurnosne stijene koje pamte zapise stanja mrežnih povezivanja (*engl. stateful firewall*).

Zaštita na razini sjednice je tehnologija bazirana na sjednici što znači da sigurnosne stijene na ovom sloju nadilaze pojedine pakete te nadgledaju uspostavljanje i zatvaranje komunikacijske sjednice između mrežnih računala. Takve sigurnosne stijene pružaju uslugu dinamičkih protokola te prepoznaju instrukcije promjene vrata u komunikaciji klijenta i poslužitelja. Takvim pristupom potrebno je dozvoliti samo nove veze dok se uspostavljene veze automatski prihvaćaju.

Iz razloga što zaštita na razini sjednice pruža gotovo sve vrijednosti kao i zaštita na razini paketa, često puta je zaštita na razini paketa nepotrebna za mnoge mreže.

4.1.3. Zaštita na razini aplikacije

Tehnikom zaštite na razini aplikacije nagleda se mrežni promet i dinamički se analizira prepoznajući znakove mogućeg napada i uplitanja u sustav. U sklopu sigurnosne mrežne infrastrukture postoje dvije tehnologije zaštite na razini aplikacije. To su posrednička sigurnosna stijena (*engl. proxy firewall*) i sustav za prevenciju uplitanja (*engl. intrusion prevention system*).

Posrednička sigurnosna stijena je mrežni sustav koji djeluje u korist klijenata koji pristupaju mrežnim uslugama tako da štiti klijenta i poslužitelja od izravne veze s kraja na kraj. Klijent uspostavlja najprije vezu s posredničkim poslužiteljem, a posrednički poslužitelj uspostavlja zatim vezu s odredišnim poslužiteljem. Posrednički poslužitelj zatim usmjerava podatke između sudionika.

Sustavi za prevenciju uplitanja su mrežni uređaji koji mogu prihvaćati ili odbacivati promet na temelju internet adrese, protokola ili usluga kao i analize i ovjere aplikacijske razine.

Sustav za prevenciju uplitanja prima promet s mreže, sakuplja tok prometa tražeći aplikacijske osnovne oblike i naredbe, kako bi otkrio sumnjiva područja koja upućuju na neku unaprijed definiranu malicioznu akciju. Akcije koje sustav poduzima variraju od toga da se samo zapiše dnevnička stavka sumnjivog događaja ili pak se veza u potpunosti prekida.

Posredničke sigurnosne stijene i sustavi za prevenciju uplitanja provjeravaju kontrolna i podatkovna područja aplikacijskog toka kako bi potvrdili da su dotične akcije dozvoljene sigurnosnom politikom i da ne predstavljaju prijetnju za odredišni sustav. Poznajući naredbe i osnovne oblike aplikacijske razine ovi sustavi mogu prepoznati sadržaj koji je izvan zadanih norma i sadržaj koji predstavlja poznati napad. Posredničke sigurnosne stijene i sustavi za prevenciju uplitanja izvode ponovo sastavljanje TCP (*engl. transmission control protocol*) toka podataka kako bi eliminirali dvoznačan promet koji može koristiti napadač koji pokušava prikriti svoje radnje.

Posredničke sigurnosne stijene uobičajeno podupiru uobičajene Internet aplikacije, uključujući HTTP (*engl. hypertext transfer protocol*), FTP (*engl. file transfer protocol*), telnet (*engl. telecommunication network*), rlogin (*engl. remote login*), elektronsku poštu i Internet novosti. Unatoč tome za svaku novu aplikaciju ili protokol treba se razviti posrednička sigurnosna stijena kao i prilagođena programska podrška i potrebne korisničke procedure.

Sustavi za prevenciju uplitanja općenito podupiru veći raspon aplikacija i protokola. Nove aplikacije se mogu dozvoliti kroz sustav za prevenciju uplitanja bez potrebnih promjena korisničkih radnih stanica. Stoga su sustavi za prevenciju uplitanja transparentniji za mrežu nego posredničke sigurnosne stijene.

Posredničke sigurnosne stijene i sustavi za prevenciju uplitanja također mogu otkriti i određene viruse ili trojanske konje. Na primjer, sustav za prevenciju uplitanja gledajući na naslovno područje, ime privitka ili tipa privitka unutar elektronske pošte može otkriti karakteristike poznatih virusa. Međutim, zaštita aplikacijske razine ne može činiti detaljnu analizu razine datoteka, koja je također posebice važna za otkrivanje velikog broja virusa.

4.1.4. Zaštita na razini datoteke

Zaštita na razini datoteke pruža mogućnost odvajanja datoteka u prometu kako bi se takve pregledale i kako bi se otkrili eventualni virusi, crvi i trojanski konji. Uobičajena tehnologija za zaštitu na razini datoteke u nekoj mreži je povezni antivirusni program *(eng. gateway antivirus)*.

Antivirusni sustav traži oznake virusa (jedinstvene nizove okteta koji identificiraju virus) te uklanja viruse iz takvih datoteka. Većina antivirusnih sustava otkrit će ne samo inicijalne viruse veći i mnoge njihove varijante.

Povezni antivirusni sustav pretražuje datoteke koje prolaze mrežom, uključujući HTTP promet i datoteke koje pristižu prometom elektroničke pošte. Ukoliko je otkrivena inficirana datoteka povezni antivirusni sustav uklonit će je iz prometa tako da neće imati utjecaja na ostale korisnike. Kako bi sustav uspješno pretraživao datoteke mora dobro poznavati široki skup protokola za kodiranje i algoritama za kompresiju podataka.

Od kad se pojavio i svakodnevno pojavljuje vrlo veliki broj virusa povezni antivirusni sustavi moraju biti u mogućnosti kvalitetno provoditi ovakva pretraživanja. Kako bi održali efektivnu

zaštitu od novih virusa zahtjeva se konstantno nadograđivanje baze definicija virusa. Pošto aplikacijski tok podataka u kojem se pretražuje virus mora biti potpuno cjelovito prikupljen kako bi ga sustav mogao analizirati, korisnici sustava moraju se suočiti s određenim kašnjenjima u toku podataka.

Antivirusni program tipično pretražuje datoteke u elektroničkoj pošti i Internet prometu promatrajući većinom komunikaciju od poslužitelja prema klijentu. Cilj virusa je oštetiti krajnji korisnički sustav koristeći elektroničku poštu i Internet poslužitelje za njihovo širenje mrežom. Posljedično je važno otkriti pojavu virusa za vrijeme njegovog slanja ili prilikom dohvaćanja s poslužitelja.

4.2. Mehanizmi za prevenciju napada

4.2.1. Sigurnosna stijena za filtriranje paketa

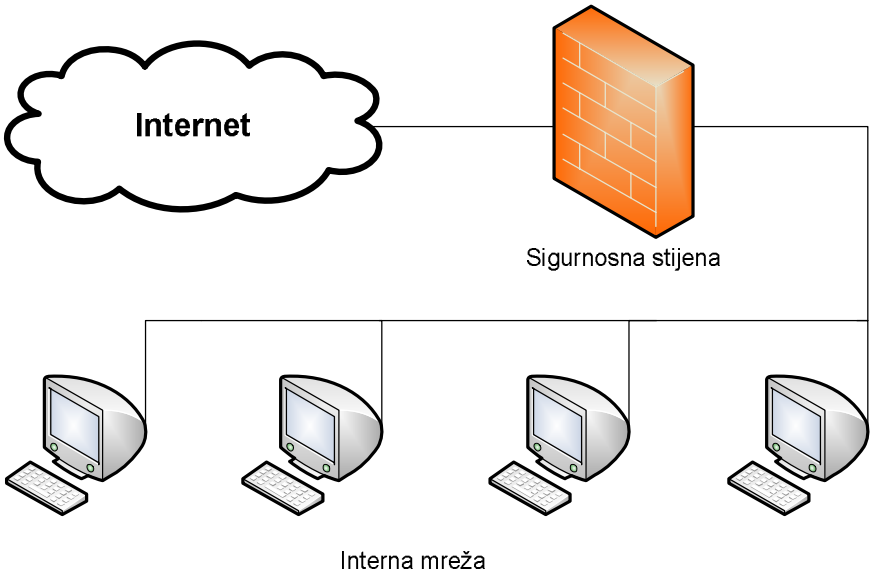
Filtriranje paketa (*engl. packet filtering firewall*) jedno je od najjednostavnijih i osnovnih poimanja mrežne sigurnosne stijene. Filtri su specijalizirane komponente sigurnosne stijene koji ispituju podatke koji izlaze ili ulaze u sigurnosnu stijenu. Dolazni i odlazni paketi uspoređuju se prema standardnim pravilima koja određuju da li će se paket propustiti ili odbaciti. U većini slučajeva skup pravila je predefiniran i bazira se na mnoštvu parametara. Pravila tako mogu uključivati izvorišnu i odredišnu IP adresu, izvorišni i odredišni broj vrata i korišteni protokol. Filtriranje paketa se općenito koristi na mrežnom sloju OSI referentnog modela i koristi neke od sljedećih parametara kako bi se paket koji dolazi na sigurnosnu stijenu propustio ili odbacio:

- **Izvorišna IP adresa paketa.**
  Svaki IP paket posjeduje izvorišnu IP adresu koja govori odakle paket dolazi. Dozvola odnosno zabrana prolaska paketa može se temeljiti upravo na njegovoj izvorišnoj IP adresi. Mnoga neautorizirana računala mogu se na ovaj način blokirati prema njihovoj IP adresi. Na taj način priječi se da neželjeni i nebitni paketi dolaze na računala unutar štićene mreže. Primjerice, značajna količina neželjene pošte (*engl. spam*) ili neželjenih reklama uzrokuje gubitak propusnosti i računalnih resursa. Filtriranjem paketa na temelju izvorišne IP adrese može biti poprilično korisno u eliminiranju većine ovakvog neželjenog sadržaja.
- **Odredišna IP adresa paketa.**
  Odredišna IP adresa je predviđena lokacija na koju paket treba stići. To je kraj transmisije. Jednosmjerni (*engl. unicast*) paketi imaju jednu odredišnu IP adresu i obično su namijenjeni jednom računalu. Višesmjerni (*engl. multicast*) ili emisijski (*engl. broadcast*) paketi imaju skup odredišnih IP adresa i obično su predviđeni za veći broj računala na mreži. Pravila se mogu osmisliti tako da blokiraju promet prema određenoj IP adresi na mreži kako bi smanjili opterećenje prema tom računalu. Ovakva pravila također se mogu koristiti za prevenciju neovlaštenog pristupa vrlo povjerljivim računalima na internoj mreži.

- **Tip Internet protokola koji paket može koristiti.**
  Paketi podatkovnog i mrežnog sloja OSI modela u svojem zaglavlju nose informaciju o tipu protokola koji se uporabljuje kako bi se ispravno postupilo prema paketu na odredišnoj adresi. Ti paketi mogu biti jedan od sljedećih tipova:
  - Normalni IP paket koji prenosi podatke
  - Obavještajni kontrolni paket kao ICMP (*engl. Internet control message protocol*)
  - Protokol za razlučivanje adresa kao ARP (*engl. address resolution protocol*) i RARP (*engl. reverse address resolution protocol*)
  - Protokol za pokretanje kao BOOTP (*engl. bootstrap protocol*)
  - Protokol za dinamičko konfiguriranje kao DHCP (*engl. dynamic host configuration protocol*)

  Filtriranje se može temeljiti na informacijama o protokolu koje paket nosi. Iako filtriranje paketa završava na mrežnom sloju, atributi transportnog sloja poput TCP zahtjeva, poruka potvrde, sekvencijskog broja i odredišnih vrata mogu se uključiti unutar pravila filtriranja.

Glavna prednost sigurnosnih stijena za filtriranje paketa je u brzini kojom se izvode operacije provjere paketa. Iz razloga što se većina posla na mreži obavlja na mrežnom sloju i niže, nije potrebno složeno znanje o aplikacijskom sloju prilikom procesiranja paketa. Vrlo često, sigurnosne stijene za filtriranje paketa postavljaju se na periferiju unutarnje mreže organizacije što je prikazano na slici 4.2. Razlog tome je što mogu biti vrlo praktična preventivna mjera koja pruža prvu liniju zaštite. Primjerice, korištenje sigurnosne stijene za filtriranje paketa može biti vrlo korisno u zaštiti od napada uskraćivanjem usluge kojem je cilj onemogućiti neki osjetljivi sustav u internoj mreži.



Slika 4.2. Uobičajeni primjer postavljanja sigurnosne stijene između Interneta i interne mreže

Iako su efektivnost, brzina i jednostavnost korištenja vrlo vrijedne značajke tehnike filtriranja paketa, također postoje i neke značajne slabosti. Iz razloga što se tehnike filtriranja paketa koriste od mrežnog sloja pa na niže nemoguće je za njih vršiti direktnu kontrolu aplikacijskog sloja. Dakle, specifični aplikacijski napadi mogu jednostavno ući u unutarnju mrežu. Kada napadač podvali lažnu IP adresu, filtriranje paketa postaje neefikasno kada filtrira te informacije na mrežnom sloju. Podvala lažne IP adrese glavna je metoda koju koristi napadač na osjetljivoj mreži. Mnoge sigurnosne stijene ne mogu otkriti lažne IP ili ARP adrese. U suštini, glavni razlog za korištenje sigurnosne stijene za filtriranje paketa je radi najopćenitijih napada uskraćivanjem usluge. Sigurnosne stijene za filtriranje paketa također ne mogu provoditi sigurnosnu kontrolom (poput kriptografije i autorizacije) jer ona funkcionira na višim mrežnim slojevima.

4.2.2. Sigurnosna stijena koja pamti stanja paketa

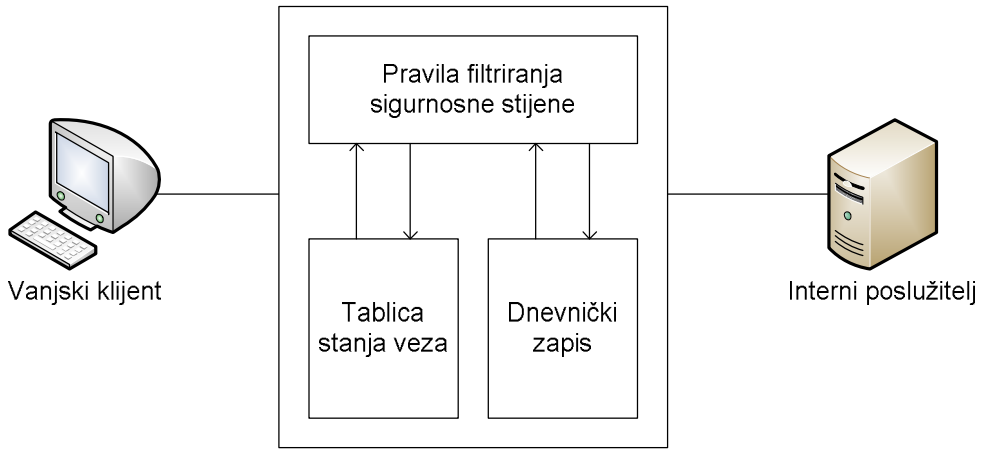
Tehnike filtriranja paketa uz pamćenje stanja (*eng. stateful firewall*) koriste složeniji pristup iako još uvijek koriste i osnovne metode sigurnosnih stijena za klasično filtriranja paketa. U mrežnoj komunikaciji, transportni sloj funkcionira na principu veza. Veza se definira kao legitimno komuniciranje jednog izvora i jednog odredišta koji međusobno odašilju i primaju informacije. Veza se može opisati sa četiri parametara:

- Izvorišna adresa
- Izvorišna vrata
- Odredišna adresa
- Odredišna vrata

Uobičajeno transmisijski kontrolni protokol na transportnom sloju OSI referentnog modela koristi ovakav mehanizam spojne veze i razlikuje se od nespojnog (*engl. connectionless*) Internet protokola prisutnog na mrežnom sloju.

Tehnike filtriranja paketa uz pamćenje stanja koriste dinamičku memoriju koja u tablici stanja pamti dolazne i uspostavljene veze. Svaki put kad vanjski entitet zahtjeva vezu prema nekom računalu na mreži, parametri veze bilježe se u tablici stanja. Kako bi legitimna konverzacija mogla započeti, slično kao i u tehnikama filtriranja paketa, moraju biti zadovoljena određena pravila. Iz razloga što tehnike filtriranja paketa uz pamćenje stanja uključuju i informacije viših mrežnih slojeva njihovom konfiguriranju se mora pristupiti vrlo pažljivo. Kada se postavi previše ograničenja na podatke koji dolaze na sigurnosnu stijenu, klijenti kao i legitimni udaljeni korisnici mogu osjetiti poteškoće performansi rada. To može rezultirati gubitkom poslovanja ili lošom produktivnošću komercijalne organizacije.

Tehnike filtriranja paketa uz pamćenje stanja koriste za filtriranje podatke transmisijskog kontrolnog protokola kao i kontrolne podatke viših mrežnih slojeva. Informacije o vezi pohranjene su u tablici stanja koja se uobičajeno kontrolira dinamički. Svaka veza je zabilježena u tablici, i nakon što je veza odobrena paketi se preusmjeravaju na temelju pravila koja su definirana za pojedinu vezu. Slika 4.3. pokazuje arhitekturu filtriranja paketa uz pamćenje njihovog stanja.



Slika 4.3. Arhitektura sigurnosne stijene koja pamti stanje paketa

Iako sigurnosne stijene koje pamte stanja paketa čine dobar posao u povećanju sigurnosnih mogućnosti (koje općenito nisu prisutne na sigurnosnim stijenama za klasično filtriranje paketa) nisu toliko fleksibilne i robusne. Uključivanje dinamičke tablice stanja i drugih mogućnosti u sigurnosnu stijenu čini njezinu arhitekturu kompleksnijom. To je direktno vezano sa brzinom izvođenja operacija na takvim sigurnosnim stijenama. Kako se broj veza povećava, sadržaj tablice stanja također se širi i može se povećati na veličinu koja uzrokuje zagušenje protoka. To će se manifestirati korisnicima mreže kao smanjenje brzine performansi. Još jedan problem s kojim se susreću sigurnosne stijene koje pamte stanja paketa je taj što ne mogu potpuno pristupiti kontroli protokola viših slojeva kao i aplikacijskim uslugama. Što je više sigurnosna stijena orijentirana na aplikacijsku razinu, sužava se njezin broj operacija a povećava kompleksnost arhitekture.

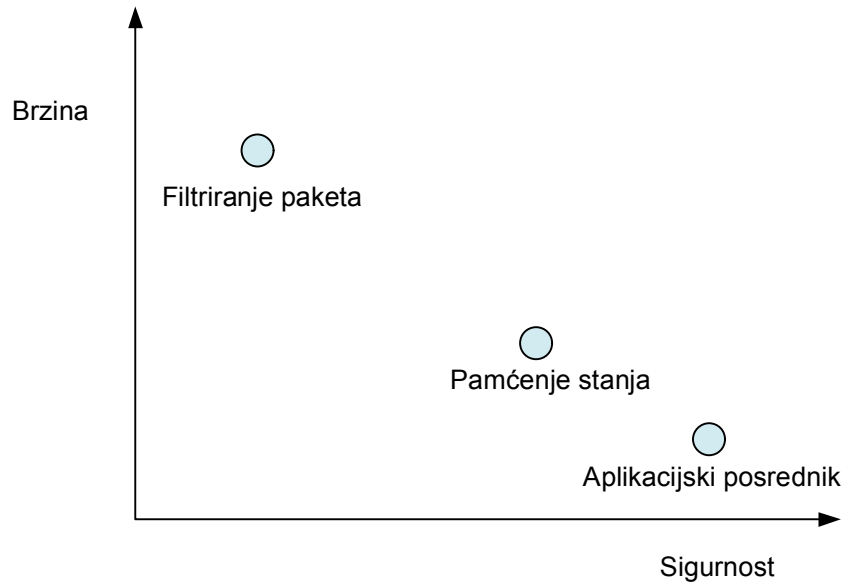
4.2.3. Posrednička sigurnosna stijena

Osnovna namjena posredničkih sigurnosnih stijena (*engl. proxy firewall*) je obavljanje operacija zaštite na najvišem sloju OSI referentnog modela (aplikacijskom sloju). Posrednik je nadomjesni sustav koji se koristi kod uspostavljanja spojno orijentiranih veza. Posrednik se primjerice može postaviti između udaljenog korisnika (koji može biti na javnoj mreži poput Interneta) i nekog poslužitelja na Internetu. Sve što udaljeni korisnik vidi je upravo posrednik te on ne poznaje identitet poslužitelja s kojim zapravo komunicira. Slično tome poslužitelj vidi također posrednika i ne poznaje pravog korisnika. Posrednička sigurnosna stijena stoga može biti dobar mehanizam zaštite i dobar sustav za filtriranje paketa između javne mreže i zaštićene privatne mreže. Iz razloga što posrednička sigurnosna stijena potpuno zaštićuje aplikacije i što se aktivnosti događaju upravo na aplikacijskom sloju, ove sigurnosne stijene vrlo su korisne za osjetljive aplikacije. Kako bi osnažili sigurnosnu implementaciju pristupanja posredniku mogu se koristiti razni autorizacijski mehanizmi, kao što su lozinke, biometrija i slično.

U većini slučajeva, posredničke sigurnosne stijene postavljaju se kao dopuna rada glavne sigurnosne stijene. Posrednički agenti su aplikacije i specifične implementacije protokola koje djeluju u korist predviđenih aplikacijskih protokola. Primjeri protokola za koje se mogu postaviti posrednički agenti su sljedeći:

- HTTP (*engl. hypertext transfer protocol*)
- FTP (*engl. file transfer protocol*)
- RTP (*engl. real-time transport protocol*)
- SMTP (*engl. simple mail transfer protocol*)

Glavni nedostatak korištenja posredničke sigurnosne stijene je nedostatak brzine. Iz razloga što se aktivnosti događaju na aplikacijskom sloju te se procesira velika količina podataka posredničke sigurnosne stijene ograničene su brzinom i troškovima. Unatoč nedostacima ipak pružaju bolju sigurnost bilo od sigurnosnih stijena za filtriranje paketa te sigurnosnih stijena koje pamte stanja paketa. Slika 4.4. pokazuje usporedbu tehnologija sigurnosnih stijena.



Slika 4.4. Usporedba tehnologija sigurnosnih stijena

4.2.4. Sustav za prevenciju uplitanja

Sustav za prevenciju uplitanja (*engl. intrusion prevention system*) je računalni sigurnosni uređaj koji nadgleda aktivnosti sustava kako bi otkrio maliciozno i nepoželjno ponašanje, reagirao u stvarnom vremenu, te spriječio takve aktivnosti.

Sustavi za prevenciju uplitanja nastali su s razlogom da riješe problem dvoznačnosti pasivnog mrežnog nadgledanja. Sustav za prevenciju uplitanja značajno je poboljšanje sigurnosne tehnologije te može donijeti odluke o pristupu na temelju sadržaja aplikacije, umjesto da koristi IP adrese i vrata kako to čine tradicionalne sigurnosne stijene. Osim mrežnih aktivnosti sustavi za prevenciju uplitanja mogu također služiti i na korisničkoj razini kako bi spriječili njihove potencijalne maliciozne aktivnosti. Sustav za prevenciju uplitanja mora biti vrlo dobar sustav za otkrivanje uplitanja, tj. mora imati nisku stopu netočnih akcija.

Sustavi za prevenciju uplitanja posjeduju neke prednosti nad sustavima za otkrivanje uplitanja. Jedna od prednosti je što izravno prate promet i uskraćuju napade u stvarnom vremenu. Uz to, sustav za prevenciju uplitanja nadgleda aplikacijski sloja OSI referentnog modela, odnosno protokole koji zahtijevaju posebnu pažnju (kao što su HTTP, FTP i SMTP).

Međutim, koristi li se mrežno orijentirani sustav za prevenciju uplitanja, potrebno je razmotriti da li se promet mrežnog segmenta kriptira. Razlog zašto tome valja obratiti pažnju je što ne postoji mnogo proizvoda koji su u mogućnosti nadgledati takav promet.

Korisnički orijentiran sustav za prevenciju uplitanja

Korisnički orijentiran sustav za prevenciju uplitanja prebiva na specifičnoj IP adresi, uobičajeno na jednom računalu. Korisnički orijentiran sustav za prevenciju uplitanja je potencijalni nasljednik metoda temeljenih na otkrivanju otisaka kao i heurističkih antivirusnih metoda. Razlog tome je što ne zahtjeva kontinuiranu nadogradnju kako bi ostao u toku s novim opasnostima. Kako korisnik pokušava ostvariti cilj tako da malicioznim kodom izmijeni sustav ili bilo koju programsku podršku koja se nalazi na računalu, korisnički orijentiran sustav za prevenciju uplitanja zapazit će neke od rezultirajućih promjena i uskratiti takvu aktivnost.

Mrežno orijentiran sustav za prevenciju uplitanja

Mrežno orijentirani sustav za prevenciju uplitanja je sustav kod kojeg se svaka akcija kojom se pokušava spriječiti uplitanje prema nekom domaćinu na mreži odvija s drugog računala, druge IP adrese na mreži.

Mrežno orijentirani sustav za prevenciju uplitanja čini sklopovlje i programska podrška koja je osmišljena da analizira, otkriva i pruža izvještaj o sigurnosno vezanim događajima. Takvi sustavi osmišljeni su da nadziru mrežni promet te na temelju njihove konfiguracije ili sigurnosne politike uskraćuju rad malicioznih paketa na mreži.

4.2.5. Antivirusni programi

Antivirusni programi su računalni programi koji pokušavaju identificirati, spriječiti ili eliminirati maliciozne programe. Termin antivirus koristi se iz razloga što su prvobitni primjeri ovakvih programa bili usmjereni isključivo na računalne viruse. Danas pak većina modernih antivirusnih programa rješava širok niz problema, uključujući prevenciju crva, lažnog odašiljanja podataka, programa za neovlaštenu kontrolu sustava, trojanskih konja i drugih malicioznih prijetnji. Antivirusni programi uobičajeno koriste dva pristupa:

- Pregledavanje datoteka kako bi pronašli poznate viruse koji zadovoljavaju definicije u rječniku virusa.
- Otkrivanje sumnjivog ponašanja nekog od računalnih programa koje može ukazivati na infekciju.

Pristup otkrivanja sumnjivog ponašanja naziva se još i heuristička analiza. Takva analiza uključuje prikupljanje podataka, nadgledanje vrata i druge metode.

Mnogi komercijalni antivirusni programi koriste oba pristupa s naglaskom na pristupu rječnika virusa. Neki ljudi smatraju sigurnosnu stijenu kao tip antivirusnog programa, što nije korektno.

Rječnik virusa

Kada antivirusni programi pregledavaju datoteke u pristupu rječnikom virusa oni se referenciraju na rječnik poznatih virusa kojeg je isporučio proizvođač antivirusnog programa.

Ukoliko dio koda u datoteci odgovara zapisu nekog od virusa prepoznatog u rječniku, tada antivirusni program poduzima jednu od sljedećih akcija:

- Pokušava ispraviti datoteku uklanjanjem virusa iz nje.
- Stavlja datoteku u karantenu, tj. čini je nedostupnom drugim programima.
- Briše inficiranu datoteku.

Kako bi antivirusni programi ispravno otkrio maliciozne radnje, rječnik definicija virusa mora se redovito nadograđivati.

Antivirusni programi koji koriste rječnik tipično pregledavaju datoteke kada ih operacijski sustav računala stvara, otvara, zatvara ili šalje. Na taj način može otkriti poznate viruse izravno prije nego počnu maliciozno djelovati.

Iako pristup rječnikom može vješto uskratiti djelovanje virusa u odgovarajućim okolnostima, autori virusa pokušali su otići korak dalje kako bi zaobišli ovaj mehanizam. Razvili su mnoge polimorfne viruse, koji se prikrivaju kriptiranjem svojih dijelova. Na taj način virus modificira svoj oblik kako ne bi zadovoljio oznaku virusa u rječniku antivirusnog programa.

U posljednje vrijeme pojavljuje se sve češće pristup uskraćivanja rada malicioznih programa koji koristi listu pouzdane programske podrške *(eng. whitelisting)*. Umjesto da se prate poznati loši programi, ova tehnika uskraćuje izvođenje svakog računalnog koda izuzev onog koji je prethodno bio identificiran kao valjan od strane računalnog administratora. Na ovaj način izbjegnuta je potreba konstantnog održavanja rječnika s oznakama virusa. Također je spriječeno pokretanje neželjenih računalnih aplikacija sve dok se one ne nađu na listi pouzdane programske podrške.

Otkad moderne organizacije koriste velik broj valjanih programa, uspješnost funkcioniranja ove tehnike ovisi uvelike o računalnom administratoru i njegovim sposobnostima da ispravno popisuje i nadograđuje listu pouzdanih programa. Uz sve to postoje također i neke implementacije koje omogućuju automatiziranje ovog procesa.

Otkrivane sumnjivog ponašanja

Pristup otkrivanja sumnjivog ponašanja, kao kontrast korištenju rječnika ne pokušava identificirati poznate viruse, već umjesto toga nadgleda ponašanje određenih programa. Ukoliko neki program pokuša zapisati podatke u izvršne datoteke antivirusni program će otkriti takvo sumnjivo ponašanje, obavijestiti korisnika i poduzeti odgovarajuće mjere uskraćivanja ove aktivnosti.

Za razliku od pristupa korištenjem rječnika, pristup otkrivanja sumnjivog ponašanja pruža zaštitu od sasvim novih virusa čije definicije još ne postoje ni u jednom rječniku. U svemu tome moguće je da se pojavi mnogo lažnih pozitivnosti, te je moguće da korisnik s vremenom postane neosjetljiv na sva upozorenja. Ukoliko korisnik ignorira odbija sva upozorenja antivirusni program očito ne pruža neku korist takvom korisniku.

Ostali pristupi

Neki antivirusni programi također koriste druge oblike heurističke analize. Primjerice, mogu pokušati simulirati početak koda svakog izvršnog programa kojeg sustav pokreće prije nego pruži pravu kontrolu takvom programu. Ukoliko izgleda da program koristi kod kojim

modificira samog sebe ili pak izgleda poput virusa (npr. neposredno pokušava pronaći ostale izvršne datoteke) može se pretpostaviti da je virus inficirao tu izvršnu datoteku. Unatoč tome, ovakve metode mogu rezultirati velikim brojem lažnih pozitivnosti.

Također postoje i druge metode otkrivanja virusa koje koriste model. Model simulira operacijski sustav i pokreće izvršnu datoteku u toj simulaciji. Nakon što program završi, antivirusni program analizira model te traži promjene koje se mogu manifestirati kao pojava virusa. Zbog sporijih performansi izvođenja, ovaj način otkrivanja se obično koristi jedino na zahtjev. Ova metoda također može biti nepouzdana ukoliko je virus nedeterministički i rezultira različitim aktivnostima ili pak završi bez ikakvih aktivnosti te ga je stoga vrlo teško otkriti iz samo jednog pokretanja simulacije.

5. Penetracijski test

Penetracijski test je jedan od načina identificiranja ranjivosti i vrednovanja sigurnosti nekog sustava ili mreže u kojima postoje određene sigurnosne mjere. Penetracijski test obično uključuje korištenje metoda napada upravljanih od povjerljivog pojedinca, penetracijskog ispitivača (*engl. penetration tester*), na način sličan napadu zlonamjernog napadača. Ovisno o tipu testiranja koje se obavlja, to može uključivati jednostavno pretraživanje IP adresa (*engl. Internet protocol address*) kako bi se identificirala računala koja posjeduju ranjive usluge ili pa čak iskorištavanje poznatih ranjivosti na sustave koji nisu zaštićeni najnovijim sigurnosnim zakrpama. Rezultati takvih testiranja i napada se zatim dokumentiraju i prezentiraju u obliku izvještaja vlasniku sustava kako bi se sustav zaštitio od identificiranih propusta.

Treba voditi računa da penetracijski test ne traje za uvijek. Vrijeme provođenja pojedinog testa varira. Penetracijski test je u osnovi pokušaj narušavanja sigurnosti sustava ili mreže i nije potpuna sigurnosna procjena. To znači da je penetracijski test samo pogled na stanje sigurnosti sustava u određenom trenutku vremena. Trenutno poznati sigurnosni propusti, ranjivosti ili neadekvatna konfiguracija prisutna u sustava neće biti uklonjeni sve do vremenskog trenutka provođenja penetracijskog testa.

Penetracijsko testiranje se obično provodi da poveća osviještenost menadžmenta o sigurnosti ili radi testiranja otpornosti sustava na napade te njihove sposobnosti i odziv. Također pomaže menadžmentu u procesu donošenja odluka. Moguće je da menadžment neće prihvatiti proces otklanjanja svih ranjivosti otkrivenih kompletnom procjenom ranjivosti sustava već samo važnih sistemskih slabosti otkrivenih kroz penetracijske testove. Razlog tome je što uklanjanje svih ranjivosti može biti skupo pa većina organizacija nije u mogućnosti izdvojiti toliki budžet.

Penetracijski testovi mogu imati ozbiljnih posljedica za mrežu na kojoj se provode. Ukoliko se loše provode mogu uzrokovati zagušenje mreže ili pad sustava. U najgorem slučaju mogu završiti upravo u scenariju za koji su namijenjeni da ga spriječe. To je kompromis sustava i penetracijskog ispitivača. Stoga je ključno dobiti odobrenje od strane menadžmenta organizacije prije provođenja penetracijskog testa na njihov sustav ili mrežu.

5.1. Razlozi za provođenje penetracijskog testa

Postoji više razloga zašto se preporuča provoditi penetracijske testove. Jedan od glavnih razloga je zasigurno pronaći i ispraviti sigurnosne propuste prije nego ih pronađe i iskoristi napadač. Iako je informatičko osoblje (*engl. department of information technology*) često svjesno da postoje ranjivosti obično je potreban i vanjski ekspert koji daje izvještaje o sigurnosti kako bi menadžment mogao pravovremeno reagirati i odobriti resurse potrebne za zaštitu sustava. Pokazalo se da je učinkovito imati takvog vanjskog eksperta koji će nadzirati sigurnost sustava. Testiranje novog sustava prije nego što se priključi na javnu mrežu također je dobar razlog za provođenje penetracijskog testa. Obično se penetracijskim testovima testiraju i sposobnosti informatičkog osoblja ciljane organizacije da odgovori na napad. Standardi za zaštitu podataka industrije kartičnog plaćanja i ostale sigurnosne regulacije također zahtijevaju posebno sigurnosno testiranje.

5.1.1. Otkrivanje propusta prije napadača

U svako doba postoje aktivni napadači koji koriste veliki broj automatiziranih alata tražeći propuste u sustavu ili mreži. Namjera takvih napadača je iskoristiti pronađene propuste i provaliti u sustav. Jedino nekolicina od njih koristi naprednije tehnike, još javno nepoznate ranjivosti (*engl. zero day exploit*), dok većina od njih koristiti dobro poznate i predvidive načine napada. Penetracijski testovi daju informatičkom osoblju jasnu sliku ranjivosti u njihovoj mreži. Cilj ovakvog penetracijskog testa je pronaći ranjivosti sustava kako bi se takvi mogli ispraviti prije nego ih otkrije napadač. U usporedbi, ovakav penetracijski test sličan je redovitom liječničkom pregledu čovjeka. Čak iako se vjeruje u trenutno zdravlje, kao ljudi prolazimo različita testiranja (neka stara, neka nova) kako bi se dijagnosticirale eventualne bolesti čiji simptomi nisu još razvijeni.

5.1.2. Izvještavanje menadžmenta o problemima

Ukoliko je informatičko osoblje već istaknulo menadžmentu nedostatak sigurnosti u sustavu, rezultati penetracijskog testa pomoći će u opravdanju izdvajanja sredstava potrebnih za njihovo rješavanje. Obično je informatičko osoblje organizacije svjesno slabosti u sigurnosti njihovog sustava, ali je s druge strane problem zadobiti menadžment da podupru obavljanje potrebnih promjena. Pokazalo se da menadžment ima više respekta prema vanjskoj ekspertnoj grupi za sigurnost (*engl. outside security expert*) te je stoga korisno posjedovati takvu u organizaciji. Unutar korporacije obično postoje političke nesuglasice i ograničenje resursa pa je vanjski ispitivač također pogodniji zbog neutralne strane. Informatičko osoblje najčešće zahtijeva porast budžeta za nove tehnologije. Koristeći neutralnu stranu da potvrdi nužne potrebe, menadžmentu je dano dodatno opravdanje za odobrenje ili zabranu novčanih prihoda za sigurnosnu tehnologiju. Slično tome informatičko osoblje upoznao je sa složenošću njihove okoline te načinima kako se kompromitirati sa sustavom. Menadžmentu nije neobično pretpostaviti kako bi bez takvog znanja napadač bio u nemogućnosti pristupiti neovlaštenim sadržajima. Koristeći također neutralnu stranu koja ne posjeduje unutarnje znanje, penetracijski ispitivački tim je u mogućnosti identificirati iste ranjivosti uočene od unutrašnjeg tima i pomoći uvjeriti menadžment kako bi ranjivosti trebale biti uklonjene.

Krajnja odgovornost za sigurnost informacijske tehnologije dakako stoji na menadžmentu. Ta odgovornost stoji prvenstveno na njima jer oni odlučuju koja je prihvatljiva razina rizika za njihovu organizaciju.

5.1.3. Potvrđivanje sigurnosti sustava

Ukoliko je informatičko osoblje organizacije pouzdano u njihovim postupcima i konačnim rezultatima, penetracijski test će to potvrditi. Postojanje vanjskog entiteta za potvrdu sigurnosti također daje dobar uvid u sigurnost neovisno o unutarnjim čimbenicima organizacije. Vanjski entitet može također dati i dobru procjenu efektivnosti unutarnjeg tima kao sigurnosnog osoblja. Penetracijski test neće učiniti mrežu sigurnijom, ali će ukazati na propuste između znanja i implementacije.

5.1.4. Sigurnosna obuka za informatičko osoblje

Penetracijski testovi daju informatičkom osoblju priliku da prepozna i odgovori na mrežni napad. Primjerice, ukoliko penetracijski ispitivač uspješno kompromitira sustav bez ičijeg

znanja, to je samo indikator neadekvatnog pristupa obuci osoblja zaduženog za sigurnost. Osoblje za testiranje, praćenje i rješavanje incidenata može na ovaj način pokazati svoje sposobnosti razlučivanja situacije stanja sustava i pokazati efektivnost njihovog odgovora na otkriveni napad. Kada osoblje za sigurnost ne uspije identificirati neprijateljsku aktivnost, rezultati penetracijskog testiranja mogu biti korisni u usavršavanju njihovih vještina na odaziv i uskraćivanje neprimjerenih aktivnosti u sustavu.

5.1.5. Otkrivanje propusta neusklađenosti

Korištenje penetracijskih testova u terminu otkrivanja propusta u neusklađenosti bliže je tehnikama ispitivanja (*engl. auditing*) nego pravom sigurnosnom inženjeringu (*engl. security engineering*). No, iskusniji penetracijski ispitivači često prelaze te granice. Jedan od razloga je što nisu sva računala nadograđena najnovijim zakrpama ili pak je neusklađeno računalo privremeno priključeno u sustav te je kao takvo postalo kritičnim resursom. U današnja vremena, okolina sustava se teško regulira, te stoga mnoge organizacije traže bolje načine za kontinuiranu procjenu njihove usklađenosti. Takve regulacije posjeduju višestruke komponente specifično povezane sa sistemskim ispitivanjem i sigurnošću.

5.1.6. Testiranje novih tehnologija

Idealno vrijeme za testiranje nove tehnologije je upravo vrijeme prije njenog puštanja u pogon. Izvođenjem penetracijskog testa na nove tehnologije, aplikacije i njihovu okolinu prije njihove javne aktivnosti često znači uštedu vremena i novca. Razlog tome je što je jednostavnije testirati i modificirati novu tehnologiju o kojoj nitko nije ovisan. Primjerice, testiranje novog web poslužitelja, nove bežične infrastrukture itd.

5.2. Klasifikacija penetracijskih testova

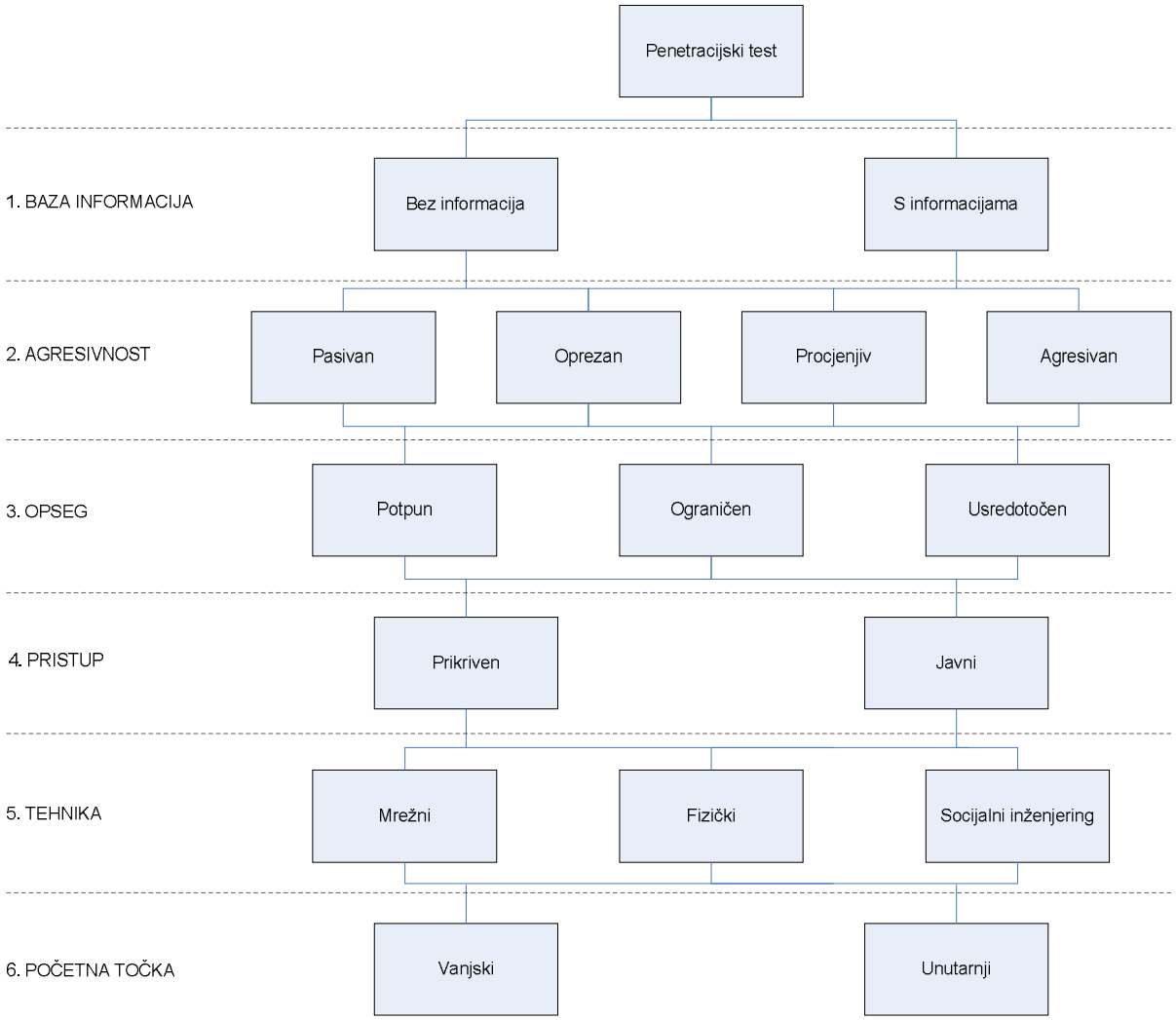
Penetracijske testove možemo klasificirati u nekoliko kategorija prikazanih na slici 5.1. Testovi se klasificiraju s obzirom na kriterije koji se koriste za njihov opis, odnosno s obzirom na obilježja koja ih međusobno razlikuju. Postoji šest glavnih kategorija:

- Penetracijski testovi prema bazi informacija
- Penetracijski testovi prema agresivnosti
- Penetracijski testovi prema opsegu
- Penetracijski testovi prema pristupu
- Penetracijski testovi prema primijenjenoj tehnici
- Penetracijski testovi prema početnoj točki napada.

Svaka od tih kategorija sadrži daljnje grupacije koje detaljnije opisuju obilježja i namjenu dotičnog testa.

Različita obilježja koja karakteriziraju neki penetracijski test moraju se prilagoditi na način da zadovolje cilj te osiguraju efikasan ishod testa s minimalnim rizikom. Kako bi se smanjio rizik, preporuča se primjena kombinacije različitih penetracijskih testova danih ovom klasifikacijom. Primjerice, u prvom koraku može se provesti oprezan, prikriven test bez informacija izvana, a nakon toga agresivan, javan test s informacijama iznutra. Takav pristup

kombinira prednosti testa bez informacija, realne simulacije pravog napada, s pogodnostima testa s informacijama kako bi se postigla maksimalna učinkovitosti i limitirala moguća šteta.



Slika 5.1. Klasifikacija penetracijskih testova

Važno je također imati na umu da je potrebno pažljivo birati koja kombinacija testa će se provoditi jer u protivnom test može biti beskoristan. Primjerice, agresivan test je obično otkriven vrlo brzo pa ga stoga nije prikladno koristiti u kombinaciji s prikrivenim tehnikama. Slično tome, javan penetracijski test nije prikladan za dobivanje povjerljivih informacija od zaposlenika koji su unaprijed bili upozoreni tehnikama socijalnog inženjeringa.

5.2.1. Penetracijski testovi prema bazi informacija

Kategorizacija penetracijskih testova prema bazi informacija (*engl. information base*) razlikuje testove s obzirom na početni nivo znanja i informacija o ciljanom računalnom sustavu ili mreži koja se testira. Postoje dva tipa ovakvih testova:

- Penetracijski test bez informacija o sustavu
- Penetracijski test s informacijama o sustavu

Test bez informacija o sustavu (*engl. black box penetration test*) sličan je napadu stvarnog vanjskog napadača gdje penetracijski ispitivač ne posjeduje gotovo nikakvo znanje o testiranom sustavu, osim eventualno internet adrese ili domenskog imena. Penetracijski ispitivač treba sam prikupiti dodatne informacije o ciljanom sustavu ili mreži koje su mu potrebne za provođenje testa. Nasuprot tome u testu s informacijama (*engl. white box penetration test*) penetracijski ispitivač obično posjeduje kompletno znanje o računalnom sustavu ili mreži koja se ispituje. To uključuje topologiju mreže, internet adrese, izvorni kod, detalje operacijskog sustava, itd. Ovaj način testiranja također je vanjski kao i test bez informacija, ali je precizniji i efikasniji iz razloga što predstavlja gori slučaj scenarija u kojem napadač ima prethodno znanje o sustavu.

5.2.2. Penetracijski testovi prema agresivnosti

Penetracijski testovi s obzirom na kriterij agresivnosti (*engl. aggressiveness*) odnose se na stupanj agresivnosti ispitivača prilikom izvođenja testa. Kako bi se postigla dovoljna razlika između pojedinih stupnjeva ova klasifikacija svrstava penetracijske testove u četiri kategorije:

- Pasivan penetracijski test
- Oprezan penetracijski test
- Procjenjiv penetracijski test
- Agresivan penetracijski test

Stupanj najniže razine agresivnosti je stupanj pasivnosti (*engl. passive penetration test*). Testirani sustav ispituje se samo pasivno što znači da se otkrivene ranjivosti sustava ne iskorišćuju. Drugi stupanj je stupanj opreznosti (*engl. cautious penetration test*). U ovom slučaju otkrivene ranjivosti se iskorišćuju jedino pod kontrolom i punim znanjem ispitivača, na način da sustav ostane neoštećen. To su primjerice radnje korištenja dobro poznatih lozinki ili pokušavanje pristupa kazalima web poslužitelja. Stupanj procjene (*engl. calculated penetration test*) je treći nivo u kategoriji agresivnosti. Ispitivač također iskorišćuje ranjivosti, ali također koristi i one koje mogu uzrokovati poremećaje sustava. To uključuje primjerice automatizirano pogađanje lozinka te iskorištavanje poznatih preljeva spremnika u precizno identificiranom ciljanom sustavu. Prije poduzimanja ovog koraka, ispitivač mora pomno procijeniti koliko je ovakva radnja poželjna za uspješnost testa, te koliko će biti ozbiljne njene posljedice. Najviši stupanj je stupanj agresivnosti (*engl. agressive test*). U ovom slučaju ispitivač pokušava iskoristiti sve potencijalne ranjivosti sustava. Primjerice, pokušaj preljeva spremnika koristi se čak i na ciljanom sustavu koji nije jasno identificiran ili pak je sigurnosni sustav deaktiviran namjernim preopterećenjima (napadom uskraćivanjem usluge). Ispitivač mora biti svjestan da ovakvo testiranjem može negativno utjecati čak i na okolinu testiranog sustava, bilo susjednih računalnih sustava ili mrežnih komponenta.

5.2.3. Penetracijski testovi prema opsegu

Opseg (*engl. scope*) određuje koji će sve dio sustava biti podvrgnut penetracijskom testiranju. Kada se penetracijski test provodi po prvi puta, preporuča se obavljanje potpunog testa kako bi izbjegli previđene propuste u neispitanim sustavima. Vrijeme potrebno za obavljanje

penetracijskog testa obično je izravno vezano za opseg sustava koji se testira. Identični i skoro identični sustavi obično se mogu ispitati u jednom testu, no čim postoje različite konfiguracije svaki sustav će se trebati podvrgnuti testu odvojeno. Tako u ovoj kategoriji postoje sljedeći testovi:

- Usredotočeni penetracijski test
- Ograničeni penetracijski test
- Potpuni penetracijski test

Ako se ispituje samo izdvojeni dio mreže ili samo dio sustava kažemo da je penetracijski test usredotočen (*engl. focused penetration test*). Ovakav test prikladan je obično nakon modifikacije ili proširenja okoline sustava. Slično tome, u ograničenom (*engl. limited penetration test*) penetracijskom testu ispituje se također ograničen broj sustava ili usluga, ali onih koji tvore jednu funkcijsku jedinicu. Primjerice, ispituju se svi sustavi u demilitariziranoj zoni. Također se može provodi i potpuni (*engl. full penetration test*) penetracijski test kojim je pokriveno ispitivanje cijelog sustava.

5.2.4. Penetracijski testovi prema pristupu

Klasifikacija prema pristupu (*engl. approach*) svrstava testove u dvije kategorije s obzirom na uočljivost prisutnosti ispitivača prilikom provođenja testa. To su kategorije:

- Prikriveni penetracijski test
- Javni penetracijski test

Penetracijski test koji se provodi trebao bi barem u početku biti prikriven (*engl. covert penetration test*). Takav test obično se koristi u inicijalnoj fazi te iskorišćuje samo metode koje se direktno ne identificiraju kao pokušaji napada na sustav. Tek ukoliko prikriveni pristup ne uspije dati reakciju preporuča se koristiti javne (*engl. overt penetration test*) metode, kao što je primjerice iscrpno pretraživanje vrata (*engl. port*).

5.2.5. Penetracijski testovi prema primijenjenoj tehnici

Penetracijski testovi s obzirom na primijenjenu tehniku (*engl. tehnique*) dijele se na:

- Mrežne penetracijske testove
- Fizičke penetracijske testove
- Socijalni inženjering

U konvencionalnim penetracijskim testovima sustav se napada isključivo preko mreže. Mrežni (*engl. network based penetration test*) penetracijski test je postupak koji simulira tipičnog napadača. Većina današnjih mreža koristi TCP/IP protokol, zbog čega se ovakvi testovi još nazivaju i penetracijski testovi bazirani na IP-u. Osim uobičajenih TCP/IP mreža postoje i ostale komunikacijske mreže koje se također mogu iskoristiti za izvođenje napada. Ova grupacija uključuje telefonske mreže, bežične mreže za mobilnu komunikaciju, bluetooth tehnologiju.

Danas veliku važnost u sigurnosti imaju sigurnosne stijene (*engl. firewall*). Konfiguracija tih sustava pruža vrlo visok stupanj sigurnosti, što povlači i izrazito tešku, ako ne i nemoguću "pobjedu" nad napadnutim sustavom. Često je jednostavnije i brže u takvim situacijama doći do potrebnih podataka svladavanjem sustava direktnim fizičkim napadom (*engl. physical attack*). Fizički napad može primjerice biti direktni pristup podacima na lozinkom nezaštićenoj radnoj stanici nakon dobivanja neautoriziranog pristupa u zgradu organizacije, odnosno ulaska u poslužiteljsku sobu. Ljudi su najčešće najslabija veza u sigurnosnom lancu. Tehnike socijalnog inženjeringa (*engl. social engineering*) obično uspijevaju zbog neadekvatne i nedovoljne upoznatosti ljudskog kadra sa sigurnošću. Ispitivač bi trebao uvijek obavijestiti klijenta o mogućim posljedicama socijalnog inženjeringa i stanju da će ta tehnika u većini slučajeva uspjeti, ukoliko korisnici nisu prethodno informirani.

5.2.6. Penetracijski testovi prema početnoj točki

Klasifikacija prema početnoj točki razlikuje penetracijske testove ovisno o mjestu s kojeg se test odvija. Početna točka penetracijskog testa (*engl. starting point*) je u praksi točka na kojoj ispitivač priključuje svoje računalo na mrežu. Ovisno o tome da li se napadi odvijaju unutar ili izvan ciljane mreže ili organizacije razlikujemo dva tipa penetracijskih testova:

- Vanjski penetracijski testovi
- Unutarnji penetracijski testovi

Većina napada odvija se preko Interneta. Vanjski (*engl. outside penetration test*) penetracijski test omogućuje otkrivanje i evaluaciju potencijalnih rizika ovakvog napada. Tipično su ovakvim testovima podvrgnute sigurnosne stijene i veze udaljenog pristupa. U unutarnjim (*engl. inside penetration test*) penetracijskim testovima ispitivač uobičajeno ne treba zaobilaziti sigurnosnu stijenu i druge mehanizme zaštite pristupa internoj mreži. Stoga unutarnji penetracijski test može procijeniti efekte pogrešaka u konfiguraciji sigurnosne stijene, uspješan napad na sigurnosnu stijenu ili napada osobe koja ima pristup internoj mreži.

5.3. Provođenje penetracijskog testa

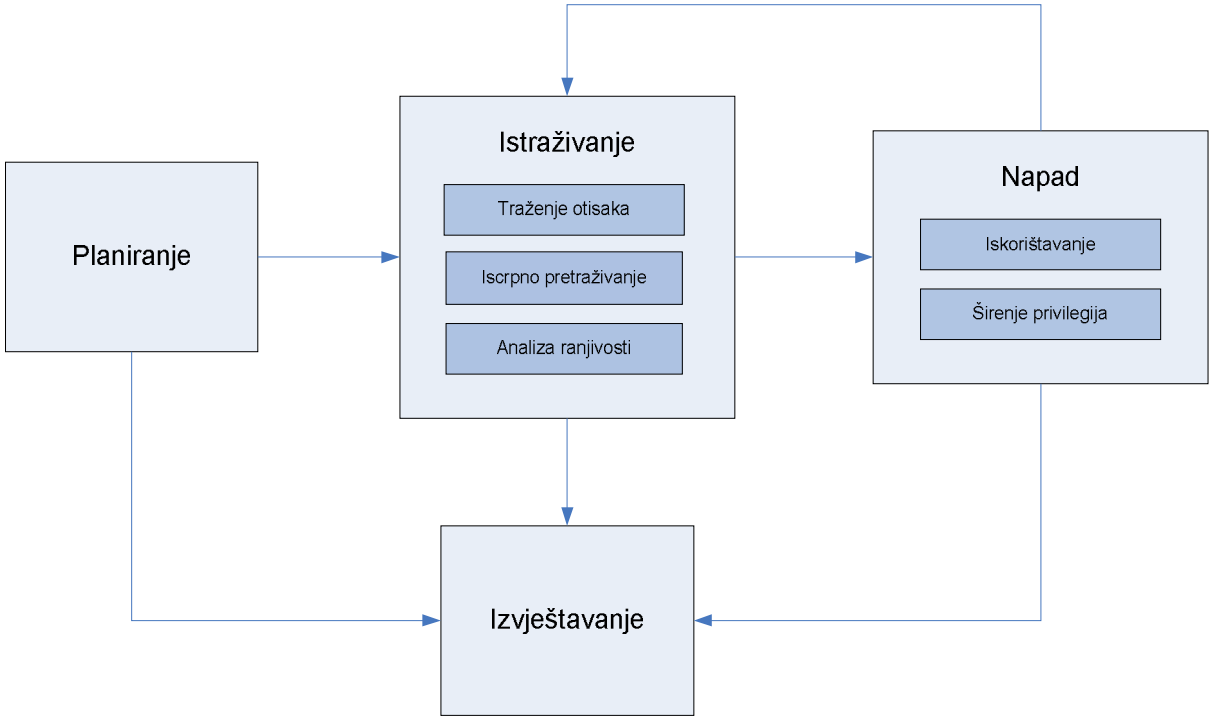
Faze provođenja penetracijskog testa prikazane su na slici 5.2. Općenito se svaki penetracijski test može provesti u četiri faze:

- Planiranje
- Istraživanje
- Napad
- Izvještavanje

5.3.1. Planiranje

U fazi planiranja (*engl. planning phase*) definira se doseg zadatka te dokumentacija i sporazum s menadžmentom koji treba odobriti provođenje penetracijskog testa. Tim penetracijskih ispitivača definira jednoznačnu strategiju zadatka. Faktori kao što su postojeća sigurnosna politika, industrijski standardi itd., obično se koriste kao ulazi za definiranje

opsega testa. Ova faza obično obuhvaća sve aktivnosti koje treba provesti prije započinjanja penetracijskog testa.



Slika 5.2 Četiri faze provođenja penetracijskog testa

Postoje različiti faktori koje je potrebno razmotriti kako bi se izvršio ispravno planirani kontrolirani napad. Za razliku od klasičnog napadača, penetracijski ispitivač ima mnogo ograničenja prilikom izvođenja testa. Stoga je potrebno ispravno planiranje za uspješno provođenje penetracijskog testa. Neka od tih ograničenja su vrijeme i legalne restrikcije. U realnoj situaciji napadač ima obično mnogo više vremena za pažljivo osmišljanje napada. Za penetracijskog ispitivača to je vremenski ograničena aktivnost. On se mora pridržavati striktnog vremena koje je prethodno dogovoreno. Također se mora imati u uvidu i faktor poput aktivnih poslovnih sati organizacije. Penetracijski ispitivač je također ograničen legalnim sporazumom koji određuje prihvatljive i neprihvatljive korake koje penetracijski ispitivač mora strogo poštivati. Također postoje i druga ograničenja koja organizacija može nametnuti penetracijskom ispitivaču. To su uglavnom ograničenja koja mogu utjecati direktno na poslovne učinke kao što je primjerice vrijeme neaktivnosti zbog pada sustava, curenje informacija (*engl. information leakage*) itd. Svi ti faktori moraju se razmotriti tijekom faze planiranja.

5.3.2. Istraživanje

Faza istraživanja (*engl. discovery phase*) je točka gdje zapravo započinje penetracijski test. Može se razmatrati kao faza prikupljanja i analize informacija. Tu fazu možemo podijeliti u tri kategorije

- Traženje otisaka
- Iscrpno pretraživanje
- Analiza ranjivosti

Traženje otisaka

Proces traženja otisaka (*engl. footprinting phase*) je nenametljiva aktivnost koja se izvodi s namjerom prikupljanja što većeg broja informacija o ciljanoj organizaciji i njezinom sustavu, bilo to informacije tehničke ili druge prirode. To uključuje pretraživanje Interneta, postavljanje upita različitim javnim repozitorijima (domenskim poslužiteljima, pretplatničkim listama, različitim grupama itd.)

Mnogi penetracijski ispitivači imaju tendenciju preskočiti ovu fazu, no ova faza daje značajnu količinu korisnih informacija. Ove informacije mogu se prikupiti bez direktnog zabadanja u ciljani sustav te stoga penetracijski ispitivač ostaje nezapažen. To su obično informacije detalja postavki informacijske tehnologije, elektronička adresa organizacije, konfiguracije uređaja, pa čak ponekad i korisnička imena i lozinke. Ove informacije obično su korisne za daljnji napad, primjerice tehnikama socijalnog inženjeringa. Penetracijski ispitivač mora ovu fazu iskoristiti što više i biti dovoljno kreativan u identificiranju različitih propusta. Na taj način ispitivač u najkraćem mogućem vremenu istražuje svaki aspekt koji bi mogao dovesti do relevantnih curenja informacija o traženoj organizaciji. Mnogi od tih postupaka mogu biti automatizirani različitim skriptama i programima.

Iscrpno pretraživanje

Faza iscrpnog pretraživanja (*engl. scanning and enumeration phase*) obično obuhvaća identificiranje aktivnih sustava, otvorenih vrata, usluga koji djeluju iza tih vrata, otkrivanje pravila sigurnosne stijene, identificiranje operacijskog sustava i njegovih detalja, otkrivanje mrežne staze, itd. Ta faza sadrži mnogo aktivnog testiranja na ciljanom sustavu. Stoga penetracijski ispitivač mora oprezno koristiti alate za te aktivnosti imajući na umu da ne preoptereti ciljani sustav pretjeranim prometom. Svi alati koji se koriste u ovoj fazi moraju se prethodno isprobati u testnom okruženju kako ne bi doveli do neočekivanih situacija. Različiti alati za otkrivanje otvorenih vrata dostupni su besplatno na Internetu. Neki od popularniji su *Nmap*, *SuperScan*, *Hping*. Nakon uspješne identifikacije otvorenih vrata, potrebno je identificirati i usluge koji su u njihovoj pozadini, bilo to korištenjem dostupnih alata, bilo manualno.

Preporuča se da penetracijski ispitivač točno potvrdi operacijski sustav, te imena i verzije usluga koji su aktivni na ciljanom sustavu prije nego ih navede u konačnom izvještaju. To će pomoći u identificiranju i eliminiranju različitih lažnih pozitivnosti pronađenih u kasnijim razmatranjima. Također postoji veći broj alata za prepoznavanje usluga i operacijskog sustava, dostupnih besplatno putem Interneta. To su primjerice *Xprobe2*, *Queso*, *Nmap*, *p0f*, *Httprint*, *Amap*, *Winfingerprint*.

Analiza ranjivosti

Nakon uspješnog identificiranja ciljnog sustava i prikupljanja potrebnih detalja iz prethodnih faza, penetracijski ispitivač trebao bi pokušati naći moguće ranjivosti koje postoje u ciljanom sustavu. Tokom ove faze penetracijski ispitivač također može koristiti automatizirane alate za

otkrivanje poznatih ranjivosti. Ti alati obično posjeduju vlastitu bazu koja sadrži informacije o najnovijim ranjivostima i njihovim detaljima.

Važno je za svakog penetracijskog ispitivača da bude u toku sa najsvježijim sigurnosno vezanim aktivnostima. Često puta ova faza ovisi o vlastitom iskustvu penetracijskog ispitivača. Stoga će uspješan penetracijski ispitivač uvijek biti opskrbljen informacijama o najnovijim ranjivostima bilo praćenjem foruma, grupa, dnevnika ili savjeta vezanih uz sigurnost.

U ovoj fazi penetracijski ispitivač također može ispitivati sustav primjenom neispravnih ulaza, nasumičnih nizova znakova, itd. ispitujući time greške i neočekivana ponašanja u izlazima sustava. Primjenjujući ove mogućnosti penetracijski ispitivač može naići na ranjivosti koje nisu mogli identificirati automatizirani alati. Stoga je vrlo važno za penetracijskog ispitivača da se ne oslanja isključivo na automatizirane alate, nego da primjenjuje i manualne tehnike testiranja. Mnogi dobri alati za pronalaženje ranjivosti dostupni su bilo kao komercijalni bilo kao alati otvorenog koda. Neki od njih su *Nessus*, *Shadow Security Scanner*, *Retina*, *ISS Scanner*, *SARA*, *GFI LANguard*. Bitno je ponovo naglasiti da penetracijski test nije samo puka aktivnost korištenja automatiziranih alata. Penetracijski ispitivač mora koristiti svoju stručnost i prosuđivanje u svakoj situaciji.

5.3.3. Napad

Ova faza je srž penetracijskog testa. Najinteresantnija je i najizazovnija. Faza napada može se općenito podijeliti na dvije faze:

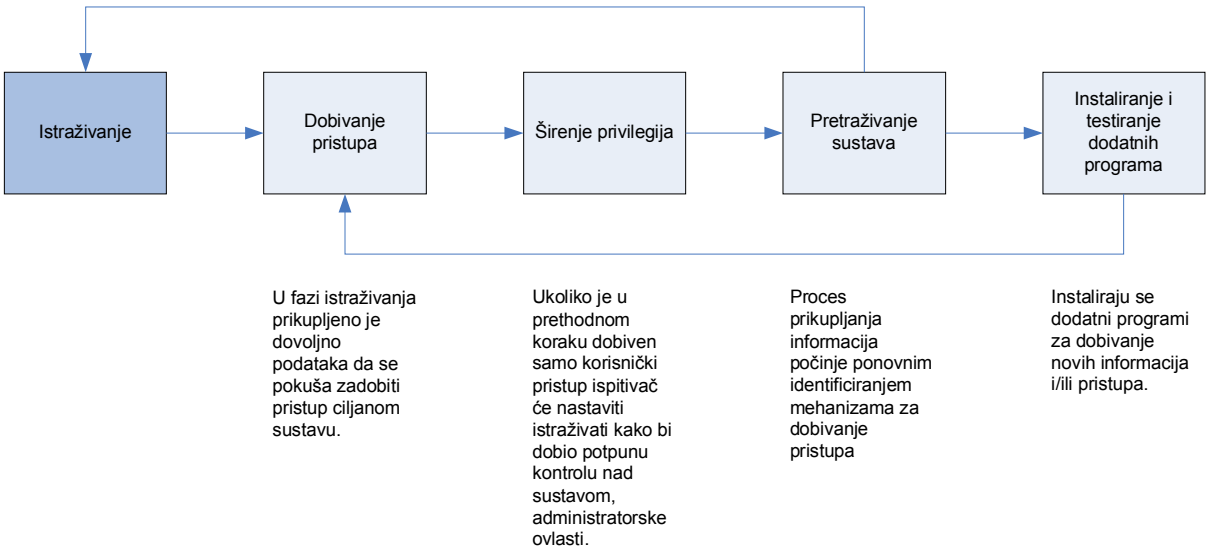
- Fazu iskorištavanja (*engl. exploitation phase*)
- Fazu širenja privilegija (*engl. privilege escalation phase*)

Iskorištavanje

Tokom ove faze penetracijski ispitivač pokušava iskoristiti upade za različite ranjivosti otkrivene u prethodnoj fazi analize ranjivosti. Na Internetu postoji mnogo repozitorija koji nude primjere upada za velik broj ranjivosti. Penetracijski ispitivač trebao bi poznavati osnovne vještine programiranja u *C* jeziku, posebice programiranje utičnica (*engl. socket*), te vještine skriptnih jezika poput *Perl-a*, *Python-a* ili *Ruby-a*. Takve vještine pomažu u razumijevanju i pisanju različitih upada, alata i automatiziranih skripti. Ova faza može biti opasna ukoliko joj se ne pristupa ispravno. Postoje vjerojatnosti da pokretanje upadnih programa onesposobi i sruši sustav. Stoga se svi upadni programi prethodno trebaju isprobati u testnom okruženju. Neke organizacije mogu zahtijevati zabranu primjene upadnih programa za određene ranjivosti. U takvom slučaju penetracijski ispitivač mora dati dostatne dokaze, tj. dobro dokumentirane koncepte koji jasno opisuju ranjivosti organizacije.

Također je važno spomenuti da postoje i dobra razvojna okruženja koja pomažu penetracijskom ispitivaču prilikom razvoja vlastitih upadnih programa i njihovog pokretanja na sistematski način. Nekoliko dobrih komercijalnih kao i onih otvorenog koda čine *The Metasploit Project*, *Core Security Technology's Impact*, *Immunity's CANVAS*. Razborito je da penetracijski ispitivač potpuno iskoristi potencijal takvih razvojnih okolina. Ta okruženja mogu uštedjeti dovoljno vremena za pisanje vlastitih upadnih programa.

Često puta uspješan upad neće dovesti direktno do administratorskih privilegija. U takvom slučaju mora se poduzeti sljedeći korak. Zahtjeva se daljnja analiza, istraživanje i traženje drugih propusta koji mogu dovesti do administratorskih privilegija. Stoga se može reći da postoji petlja između faze istraživanja i faze napada što je prikazano na slici 5.3.



Slika 5.3 Faza napada sa petljom na fazu istraživanja

Širenje privilegija

Kao što je prethodno spomenuto, postoje slučajevi u kojima uspješno upadanje neće dovesti do administratorskih privilegija. Tako će primjerice za određene ranjivosti penetracijski ispitivač dobiti tek korisničke privilegije. U takvom času potrebno je uložiti dodatni napor, nastaviti daljnju analizu na ciljanom sustavu kako bi se prikupile nove informacije koje mogu dovesti do administratorskih privilegija. Taj proces se naziva širenje privilegija. Primjerice kada penetracijski ispitivač zadobije korisničke privilegije može nastaviti pretraživati ranjivosti na lokalnom računalu. Ispitivač će možda trebati instalirati dodatne programe koji će mu pomoći u dobivanju viših privilegija. Penetracijski ispitivač također treba razmotriti mogućnost pivotiranja (*engl. pivoting*) nakon uspješnog provaljivanja u neki sustav. Pivotiranje je proces u kojem ispitivač koristi kompromitirani sustav kako bi napao ostale sustave u ciljanoj mreži. To će također dati jasan uvid utjecaja propusta u sigurnost organizacije. No, penetracijski ispitivač mora biti oprezan i prije nego što nastavi prodirati dalje treba dobiti dozvole od strane organizacije za koju provodi test. Dobar penetracijski ispitivač također će uvijek čuvati zapise svih izvedenih aktivnosti. Oni mogu posebno koristi u fazi izvještavanja i također su dokaz svih izvedenih akcija.

5.3.4. Izvještavanje

Posljednja faza u procesu provođenja penetracijskog testa je faza izvještavanja (*engl. reporting phase*). Ta faza može se izvoditi paralelno sa ostale tri faze ili na kraju faze napada. Mnogi penetracijski ispitivači nisu usredotočeni na tu fazu i slijede samo brzinski pristup u davanju izvještaja. No, ova faza najvažnija je od svih jer prije svega ciljana organizacija plaća upravo za taj konačni dokument.

Konačni izvještaj mora sadržavati oba aspekta, menadžerski i tehnički aspekt. Potrebno je detaljno prikazati sve pronalaske odgovarajućim grafovima, skicama, slikama, itd. kako bi dali valjanu prezentaciju ranjivosti i njihovog učinka na biznis ciljane organizacije. Završni rezime mora jasno opisivati obavljene radnje, otkrivene propuste te važne preporuke za rješavanje sigurnosnih pitanja. Temeljem tih pronalazaka definira se cijena za implementaciju danih preporuka.

Također je potrebno dati detaljan tehnički opis pronađenih ranjivosti i preporuku za njihovo ublažavanje. Svi pronađeni sigurnosni propusti moraju biti popraćeni odgovarajućim dokazima kao što je slika stanja sustava nakon uspješne primjene upadnog programa ili bilo kakva druga slična metoda.

Izvještaj mora biti precizan. Ništa nejasnog se ne smije ostaviti klijentu. Takva precizna dokumentacija uvijek pokazuje sposobnost uspješnog penetracijskog ispitivača.

Nužne stvari koje izvještaj treba sadržavati jesu:

- Kratak rezime
- Detaljni opis pronađenih propusta
- Stupanj rizika
- Utjecaj na biznis
- Preporuke za poboljšanje sigurnosti
- Zaključak

5.4. Zahtjevi penetracijskog testa

5.4.1. Organizacijski zahtjevi

Slijedeći organizacijski zahtjevi trebali bi se razjasniti između klijenta i penetracijskog ispitivača prilikom planiranja penetracijskog testa:

Tko i što će sve biti zahvaćeno penetracijskim testom?
Pored sustava klijenta, penetracijskom testu često su podvrgnuti i sustavi davatelja usluga (*engl. provider*) koji mogu fizički biti locirani unutar poslovnog prostora klijenta. Neki koraci testiranja, kao npr. napad uskraćivanjem usluge, može ovisno o zahtjevima propusnosti dovesti do poremećaja sustava davatelja usluga. Kako bi se zaobišli nesporazumi potrebno je o tome unaprijed obavijestiti i davatelja usluga.

Ukoliko se neke funkcije sustava koriste kao vanjske (npr. poslužitelj pružanja web usluga), takav sustav trebao bi se isključiti iz penetracijskog testiranja. Ukoliko pak se takvi sustavi uključe u penetracijsko testiranje trebalo bi se zahtijevati odobrenje sistemskog operatera ili vanjskog operatera.

Ispitivač mora napomenuti da je odgovoran za sigurnost informacijskog sustava, uključujući i vanjske sustave. Također je odgovoran i primjerice za integritet knjigovodstvenih podataka i mora jamčiti da ovi podaci neće procuriti vanjskim pružateljima usluga.

Da li je pojava mogućeg rizika razmotrena na odgovarajući način?

Penetracijski ispitivač mora također imati odgovorno osiguranje da se zaštiti od mogućih tužba i šteta treće strane. Iako provođenje penetracijskog testa mora biti odgovorno i pažljivo kako bi se smanjili mogući rizici mogućnost poremećaja sustava od treće strane ne može biti potpuno isključena.

Koje vrijeme je prikladno odabrati za provođenje testa?

Penetracijski test može smanjiti funkcionalnost produktivnog sustava. Kako je cilj testa otkriti ranjivosti, ali bez ugrožavanja aktivnosti sustava, stvarno testiranje trebalo bi se provesti u vremenu koje je potrebno prethodno definirati s obje strane. To bi se trebalo razmotriti u fazi planiranja penetracijskog testa. Penetracijski test često se provodi u vremenskom razdoblju od nekoliko dana. Vrijeme bi se trebalo odabrati prema tome kada su kritični procesi najmanje opterećeni i korišteni. Penetracijski test na ove osjetljive sustave obično se izvodi u sklopu penetracijskog testa u kojem ispitivač posjeduje potpune informacije o sustavu. Penetracijski test bez informacija posjeduje upitan nivo informacija kao i upitno znanje o iskoristivosti sustava te se stoga u ovakvim situacijama ne preporuča.

Što treba poduzeti ukoliko sustav postane nefunkcionalan ili se dogodi neki drugi hitni slučaj?

Prilikom testiranja moguće je da sustav postane nefunkcionalan unatoč pažljivosti koja se poštivala. U takvom ili u svakom drugom hitnom slučaju potrebno je definirati mjere koje će se poduzeti kod ovakvih neočekivanih situacija. Mora se barem specificirati koga i kada će se obavijestiti u slučaju otkrivene greške ili poremećaja. Uz to, potrebno je definirati i tipove grešaka o kojima će se dati izvještaj. Primjeri najčešćih poremećaja jesu:

- Potpuna nefunkcionalnost sustava
- Djelomični kvar određenog podsustava
- Neobično ponašanje sustava
- Znatno povećanje vremena odaziva sustava
- Poduzimanje protumjera kao reakcije na prikriveni penetracijski test
- Napad na sustav od treće strane

Koji zaposlenici će biti izloženi penetracijskom ispitivanju?

Broj zaposlenika koji će biti podvrgnuti testiranju ovisit će o opsegu i svrsi testa. Penetracijski testovi koji su ograničeni samo na ispitivanje sustava obično će u testiranju uključiti administratora te korisnike tog testiranog sustava. U testu koji razmatra produktivni sustav, osim korisnika sustava moguće je da će se u nekim slučajevima testiranju podvrgnuti i zaposlenici koji su na neki način vezani za rezultate testiranja sustava, te će ih se primjerice uskratiti ili omesti u njihovom radu. Ukoliko se primjenjuju tehnike socijalnog inženjeringa, stranke se moraju složiti oko zaposlenika koji mogu biti uključeni u tom testu kao i definirati djelokrug na kojem je to dozvoljeno.

Koliko će potrošiti vremena i koliki utrošak će penetracijski test prouzročiti za klijenta?

Kao rezultat penetracijskog testa klijent mora biti spreman očekivati moguće oštećenje informacijskog sustava ukoliko dođe do nepravilnosti izvođenja testa. Stoga je nužno poduzeti odgovarajuće mjere prije penetracijskog testiranja kako bi se potencijalni poremećaji održali na minimumu. To može primjerice uključivati postavljanje nekog zaposlenika da

nadgleda penetracijski test s klijentske perspektive te da u potrebi može zaustaviti njegovo izvođenje. Klijent bi također trebao uzeti u obzir potrebu za kreiranjem sigurnosnih kopija prije nego se počinje izvoditi penetracijski test. Također je nužno da se postavi plan nepredviđenih ishoda kao i eskalacijskih postupaka koji potpomažu kako u urednom izvođenju akcija tako i u primjeni odgovarajućih protumjera. Ukoliko se odabere penetracijski test s informacijama o sustavu, penetracijskom ispitivaču mora se omogućiti dostupnost važnih informacija i profesionalno partnerstvo s klijentom.

Koliko vremena i koliki napor će penetracijski test zahtijevati od ispitivača?

Kako bi mogli procijeniti ukoliko pružatelj usluga može uopće adekvatno izvesti penetracijski test potrebno je najprije definirati približne troškove, potrebno vrijeme i napor ispitivača. Potrebno je razmotriti sljedeće aspekte:

- **Svrhu i opseg penetracijskog testa.**
  Ispitivač i klijent zajedno definiraju model penetracijskog testa i procedura koje će se izvoditi na temelju svrhe penetracijskog testa. Na temelju modela i opsega penetracijskog testa moguće je odrediti resurse koje će ispitivač trebati (sklopovlje, programsku podršku, prikladne zaposlenike) prije nego što stvarni test započne.
- **Veličina infrastrukture koja će se ispitivati.**
  Veličina infrastrukture se najčešće izražava brojem IP adresa koje je potrebno testirati. Općenito, nije moguće odrediti točno vrijeme koje će ispitivač provesti na penetracijskom testiranju pojedinog sustava jer to ovisi o modelu i konfiguraciji sustava, iskustvu i posvećenosti ispitivača kao i o mnogim drugim čimbenicima. Također se kao jedan od čimbenika javlja pitanje da li je sustav koji se testira razmješten kao logički segment čiji je prolaz na javnu mrežu zaštićen centralnom sigurnosnom stijenom ili pak se radi o podijeljenoj infrastrukturi sa nekoliko različitih prolaza na javnu mrežu. Kako je ove faktore teško kvantificirati, moguće je samo izvesti vrlo općenitu tvrdnju da je potrebno to više vremena i napora za ispitivača što je veći broj sustava, tj. veća infrastruktura koja se ispituje.
- **Složenost infrastrukture koja se testira.**
  Složenost infrastrukture koja se testira je sljedeći važan faktor koji utječe na vrijeme i napor koji će ispitivač morati poduzeti. Tipične usluge koji se nude na Internetu jesu pretraživanje web stranica, dohvaćanje podataka i komunikacija elektronskom poštom. Ranjivosti aplikacija koje pružaju ove usluge vrlo često su poznate kako su te usluge vrlo često u uporabi. One su objavljene na mnogim mjestima na Internetu. Ukoliko se organizacija ograničava na ovakve rasprostranjene usluge, jasno je da je to infrastruktura niskog nivoa složenosti. Stoga će količina vremena i količina ljudskog napora u izvođenju ovakvog penetracijskog testa biti poprilično mala. Ukoliko pak se koriste složena rješenja i nestandardni i interaktivni programi to će zahtijevati više vremena kao i veći stupanj stručnosti kako bi se otkrile ranjivosti. To znači da će penetracijskom ispitivaču trebati dodijeliti više vremena i više iskusnog osoblja za izvođenje penetracijskog testa.

5.4.2. Zahtjevi ispitivača

Penetracijski test se treba prilagoditi prema situaciji klijenta i stoga ne postoji jednostavna standardizacija. Nakon što je penetracijski test jasno definiran potrebno je strogo slijediti definirani model unutar određenih okvira. Izvodi ga osoba koja ima višegodišnje iskustvo na području informacijske sigurnosti.

Potrebne su sljedeće vještine za stručno izvođenje penetracijskog testa:

- **Poznavanje sistemske administracije kao i operacijskog sustava.**
  Ovo znanje je potrebno radi vrednovanja slabosti u operacijskom sustavu ciljanog sustava te kako bi se olakšao pristup sustavima koji se podvrgavaju penetracijskom testu.
- **Znanje o TCP/IP protokolu i drugim mrežnim protokolima.**
  Od kad se rad Interneta bazira na korištenju TCP/IP protokola, koji je također postao i standard žičnih mreža, nužno je njegovo poznavanje. Znanje o TCP/IP protokolu je blisko povezano sa znanjem o drugim mrežama kao i OSI referentnim modelom.
- **Poznavanje programskih jezika.**
  Kako bi ispitivač mogao uspješno iskoristiti ranjivosti u aplikacijama i sustavu, poznavanje programskih jezika je vrlo korisno. Iako postoji velik broj automatiziranih alata s grafičkim sučeljem, iskorištavanje sigurnosnih propusta kao što je prelijevanje spremnika može se efektivno iskoristiti samo kada ispitivač ima potrebno programersko znanje.
- **Poznavanje informacijsko sigurnosnih proizvoda kao sigurnosnih stijena te sustava za prevenciju uplitanja.**
  Od kada su sigurnosna rješenja poput sigurnosnih stijena i sustava za prevenciju uplitanja postala široko uporabljiva, penetracijski ispitivač trebao bi biti upoznat kako ona funkcioniraju. Također bi trebao pratiti posljednje sigurnosne propuste s kojima se suočavaju takvi alati. Važno je imati pregled najčešćih proizvoda na tržištu na području informacijske sigurnosti.
- **Znanje o korištenju pomoćnih alata za izvođenje penetracijskog testa.**
  Uz bazično znanje, iskustvo korištenja napadačkih alata i pretraživača ranjivosti je nužno za izvođenje penetracijskog testa. Vještine rukovanja tih alata trebale bi se steći kroz praktično iskustvo. S vremenom, među mnoštvom dostupnih alata, određeni proizvodi stekli su široku primjenu (npr. *Nmap* za pretraživanje vrata). Postoje mnogi komercijalni kao i besplatni programi za izvođenje penetracijskog testa. Efikasnost uvelike ovisi o iskustvu i sposobnostima ispitivača u korištenju ovih alata.
- **Poznavanje aplikacija sustava.**
  Mnoge ranjivosti češće se nalaze u aplikacijama nego u operacijskom sustavu. One obuhvaćaju cijeli opseg aplikacija sustava, počevši od nedovoljno sigurnih makro funkcija u programima koji procesiraju tekst, preko ranjivosti Internet preglednika putem raznih skripta, pa do prelijevanja spremnika u velikim sustavima s bazama podataka. Ispitivač bi stoga trebao biti upoznat sa što većim brojem aplikacija.

Detaljno poznavanje uobičajeno korištenih aplikacija je osobito važno iz razloga što je i rizik od zlonamjernih napadača vrlo visok na tom području.

- **Kreativnost.**
  Osim visokih profesionalnih zahtjeva, kreativnost je važna činjenica kvalitete penetracijskog ispitivača. Kako kvalificirani penetracijski test može samo slijediti strogi model u ograničenom okruženju, bez sumnje će se javiti pitanje kako izvesti određenu stavku testa kada na prvi pogled izgleda nemoguće nastaviti dalje i pokušati kompromitirati sustav. Ovom problemu može se pristupiti pažljivim kombiniranjem informacija koje je ispitivač stekao, uz ranjivosti koje je identificirao te alate i tehnike kojima raspolaže. Vježbajući svoju inteligentnost, kreativan penetracijski ispitivač nalazi se u boljoj poziciji da izvede uspješan penetracijski test nego penetracijski ispitivač koji se samo oslanja na rezultate alata. Unatoč tome, kreativnost ne bi trebala nikad voditi ka nesistematskim ili kaotičnim testovima.

5.4.3. Tehnički zahtjevi

Sljedeći tehnički zahtjevi moraju se udovoljiti prije nego penetracijski ispitivač počinje izvoditi testiranje:

- **Pristup javnoj mreži.**
  Pristup Internetu ili javnoj telefonskoj mreži važan je preduvjet za izvođenje penetracijskog testa otkad se većina napada pokreće kroz te komunikacijske kanale. Kako većina pretraživača ranjivosti zahtjeva brojne mrežne resurse potrebna je dovoljno velika propusnost Internet veze.
- **Dostupnost prikladnih alata za penetracijsko testiranje.**
  Penetracijski ispitivač mora imati prikladne alate (prema vlastitoj potrebi) kako bi izvodio penetracijski test. Mnogi od tih alata dostupni su besplatno na Internetu. Uspješan test zahtjeva ispravne alata, a ne veliki broj alata. Važno je da ispitivač poznaje učinke kao i posljedice tih alata i da je u mogućnosti brzo doći do velikog broja rezultata kao i razlikovati lažne tvrdnje od istinitih.
- **Ispitivanje lokalne mreže.**
  Različiti alati moraju se najprije testirati lokalno, a tek se onda upotrijebiti u stvarnom penetracijskom testu. Ovaj tip testiranja omogućuje ispitivaču da se upozna s određenima alatima i rezultatima koje oni daju. Ukoliko je sustav testirane mreže prikladno konfiguriran, ovim alatima ispitivač se također može iskušati u otkrivanju ranjivosti.

5.4.4. Etički zahtjevi

Osim organizacijskih zahtjeva, zahtjeva ispitivača te tehničkih zahtjeva postoji i niz etičkih problema koji se moraju razmotriti prije penetracijskog testa. Stranke bi trebale razjasniti da li, i u kojoj mjeri je primjena tehnike socijalnog inženjeringa opravdana. Također bi trebali raspraviti o tome da li se smiju iskorištavati ranjivosti otkrivene prilikom izvođenja penetracijskog testa. Prije svega stranke bi trebale razjasniti da je penetracijski test uvijek

samo naručena aktivnost. Svako proaktivno ponašanje, kao primjerice pokretanje napada bez dozvole smatra se napadom i treba se odbaciti.

Korištenje tehnika socijalnog inženjeringa

Tehnike socijalnog inženjeringa vrlo su uspješne i stoga treba biti vrlo mudar da li će biti dozvoljene i u kojoj mjeri. Ova tehnika uspijeva iz razloga što svako ljudsko biće posjeduje određene karakteristike slabosti koje se mogu lako iskoristiti. To uključuje pozitivne karakteristike kao što su težnja da budemo ljubazni, da imamo osjećaj moralne dužnosti i da pomažemo, ali isto tako i manje pozitivne kvalitete poput oportunizma i neodgovornosti.

Gotovo će svaki zaposlenik dati povjerljive informacije ako napadač pristupa samopouzdano i daje čvrsti dojam. Ljudi to čine izvan vlastite volje, da pomognu s jedne strane, ali također s druge strane čine to i kao oportunističku nakanu. Ovaj tip slabosti se jedino može spriječiti pružanjem stručne obuke svim zaposlenicima. Također se može tvrditi da su tehnike socijalnog inženjeringa uspješne iz razloga nedovoljnih ili neadekvatnih sigurnosnih mjera. Ukoliko su primjerice lozinke generirane automatizirano i ukoliko su toliko komplicirane da ih je poprilično teško upamtiti, mnogi korisnici zapisat će ih kao bilješku na "sigurno" mjesto.

Kako korištenje tehnika socijalnog inženjeringa ima direktni utjecaj na zaposlenike klijenta te oni odražavaju vrijednost pouzdanosti ili sigurnosne svijesti, ove tehnike one koji su uključeni mogu učiniti bistrijima i osjetljivijima. To se događa više kada se tehnike socijalnog inženjeringa primjenjuju bez prethodnog upozorenja i kada se protumače zaposlenicima kasnije.

No mnogi sigurnosni stručnjaci odbacuju dozvolu korištenja tehnika socijalnog inženjeringa u sigurnosnim testovima, ili pak ih smatraju odgovarajućim tek kada su sigurnosni zahtjevi vrlo visoki. Korištenje tehnika socijalnog inženjeringa mora se stoga vrlo pažljivo razmotriti. Ispitivač mora uvijek obavijestiti klijenta o mogućim posljedicama socijalnog inženjeringa. Također treba navesti da će ta tehnika vrlo vjerojatno uspjeti ukoliko zaposlenici nisu prethodno obučavani.

Iskorištavanje ranjivosti

Ranjivost u aplikaciji ili operacijskom sustavu koja se može iskoristiti za preuzimanje kontrole sustava identificirat će se obično prije nego je sustav stvarno kompromitiran. Ovdje ispitivač treba promisliti da li je potrebno izvršiti taj posljednji korak iskorištavanja ranjivosti kako bi je potvrdio, ili pak je dovoljno samo istaknuti njeno postojanje. Ovo pitanje može se razriješiti tako da se razmotri definirana svrha testa kao i izvedeni uvjeti ponašanja. Ukoliko penetracijski test treba biti što realniji i informativniji bit će prikladno da se ne ograničavaju agresivnosti testirajućih postupaka. S druge strane, ukoliko je namjera testa izbjeći potencijalne poremećaje, ranjivosti se ne bi trebale izravno iskorištavati. U tom slučaju rezultat penetracijskog testa će biti samo identifikacija postojećih ranjivosti bez potvrde uspješnosti njihove iskoristivosti.

6. Opis praktičnog rada i aplikacije

Kao pomoć pri izvođenju penetracijskog testa, konkretnije u fazi istraživanja i fazi aktivnog pokušaja napada, izrađena je aplikacija za automatizirano ispitivanje ranjivosti sustava.

Aplikacija je razvijena u objektno orijentiranom programskom jeziku *C#* u razvojnom alatu *Microsoft Visual Studio 2008*. Cilj ove aplikacije je da korisniku omogući jednostavno ispitivanje ranjivosti sustava koristeći testove koji su prethodno pohranjeni u bazi. Također je jedan od ciljeva bio razviti što jednostavniji način dodavanja novih testova u aplikaciju.

Za izvor testova korištena je baza javno objavljenih propusta dostupna na Internetu *(milw0rm)*, a propusti su zatim prevedeni u testove za aplikaciju izrađenu u ovom radu. Kao baza aplikacije u koju se pohranjuju testovi koristi se poseban direktorij naziva *"exploits"*. Ovakav pristup omogućava brzo i jednostavno dodavanje novih testova što je vrlo važno kako bi aplikaciju lako osvježili najnovijim propustima, a ispitivanjem dobili sliku stanja sustava u aktualnom trenutku. Testovi su kao i aplikacija pisani u programskom jeziku *C#* (kao skripte), a njihov kod mora zadovoljiti određene minimalne uvijete.

6.1. Pravila definiranja novog testa

Svaki test mora imati definirano zaglavlje koje sadrži određene oznake. Oznakama se vrijednosti pridružuju znakom *":"* (*dvotočka*), a potrebno je minimalno definirati sljedeće oznake:

- **name** – ime testa
- **parameters** – ulazni parametri testa; nužno prvi parametar *"ip address"* bez vrijednosti

```
// name : exploit name
// parameters : ip address
```

Vrijednost parametru "*ip address"* pridružuje aplikacija nakon što korisnik putem grafičkog sučelja odabere računalo na kojem želi provesti test. Ostali parametri zadaju se na način da se međusobno odvajaju separatorom *";"* (*točka zarez*), a vrijednost im se pridružuje znakom *"="* (*jednako*).

```
// parameters : ip address; port = 8080; file to fetch = boot.ini
```

Opcionalan parametar zaglavlja je opis testa.

```
// description : exploit description
```

Nadalje svaki test mora imati definiran javni razred *"Exploit"* te javnu statičku metodu *"VulnerabilityTest"* unutar razreda.

```
public class Exploit
{
        public static int VulnerabilityTest(params string[] par)
        {
                // …
                // exploit source code
                // …
        }
}
```

Metoda *"VulnerabilityTest"* nužno prima parametre testa, a vraća aplikaciji *integer* vrijednost koja označava ishod testa:

- 0 – sustav nije ranjiv
- 1 – sustav je ranjiv

Svaki test također može izbaciti iznimku u svakom drugom slučaju. Povratnu vrijednost ili iznimku obradit će aplikacija te na taj način obavijestiti korisnika o ranjivosti sustava ili eventualnoj greški. Primjer jednog jednostavnog opisa datoteke testa prikazan je na slici 6.1 i slici 6.2.

```
// name : MiniWebSvr 0.0.9a
// description : Directory Transversal Vulnerability
// parameters : ip address; web port = 8080; file to fetch = boot.ini

using System;
using System.Text;
using System.Net;
using System.Net.Sockets;

public class Exploit
{
    public static int VulnerabilityTest(params string[] par)
    {
        IPAddress hostIp;
        int port;
        string fileToFetch;

        try
        {
            if (par.Length != 3) throw new Exception();
            hostIp = IPAddress.Parse(par[0]);
            port = Convert.ToInt32(par[1]);
            fileToFetch = par[2];
        }
        catch
        {
            throw new Exception("wrong parameters");
        }

        byte[] dataByte = new byte[1024];
        string dataSend = string.Empty;
        StringBuilder dataRecv = new StringBuilder();

        IPEndPoint ipep;
        Socket tcpSocket;
```

Zaglavlje s oznakama name, description i parameters

Javni razred «Exploit»

Statička metoda «VulnerabilityTest»

Bacanje iznimke u slučaju pogrešnih ulaznih parametara

Slika 6.1. Primjer opisa testa

```
        try
        {
            ipep = new IPEndPoint(hostIp, port);
            tcpSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
            tcpSocket.ReceiveTimeout = 3000;
            tcpSocket.Connect(ipep);

            dataSend = "GET /%../../../../../../../../../../../" + fileToFetch + " HTTP/1.0\r\n\r\n";
            dataByte = Encoding.ASCII.GetBytes(dataSend);
            tcpSocket.Send(dataByte, dataByte.Length, SocketFlags.None);

            while (tcpSocket.Receive(dataByte) != 0)
            {
                dataRecv.Append(Encoding.ASCII.GetString(dataByte));
            }

                    System.Threading.Thread.Sleep(1500);
                    tcpSocket.Close();

            if (dataRecv.ToString().Contains("404 Not Found"))
            {
                return 0;
            }
            return 1;
        }
        catch
        {
            throw new Exception("socket error");
        }
    }
}
```

Povratna vrijednost izvođenja napada

Bacanje iznimke u slučaju neispravne utičnice

Slika 6.2. Primjer opisa testa

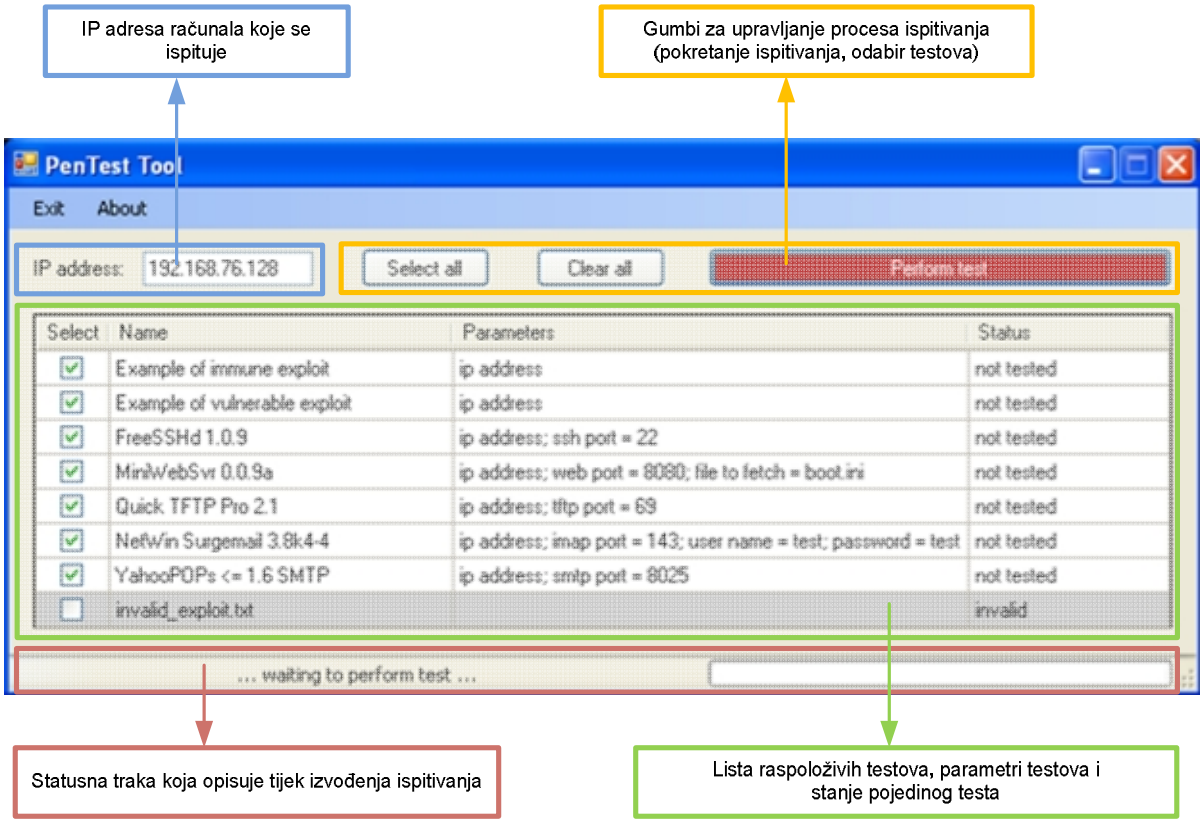
6.2. Vizualni pregled aplikacije

Osnovne cjeline aplikacije, također prikazane i na slici 6.3 su:

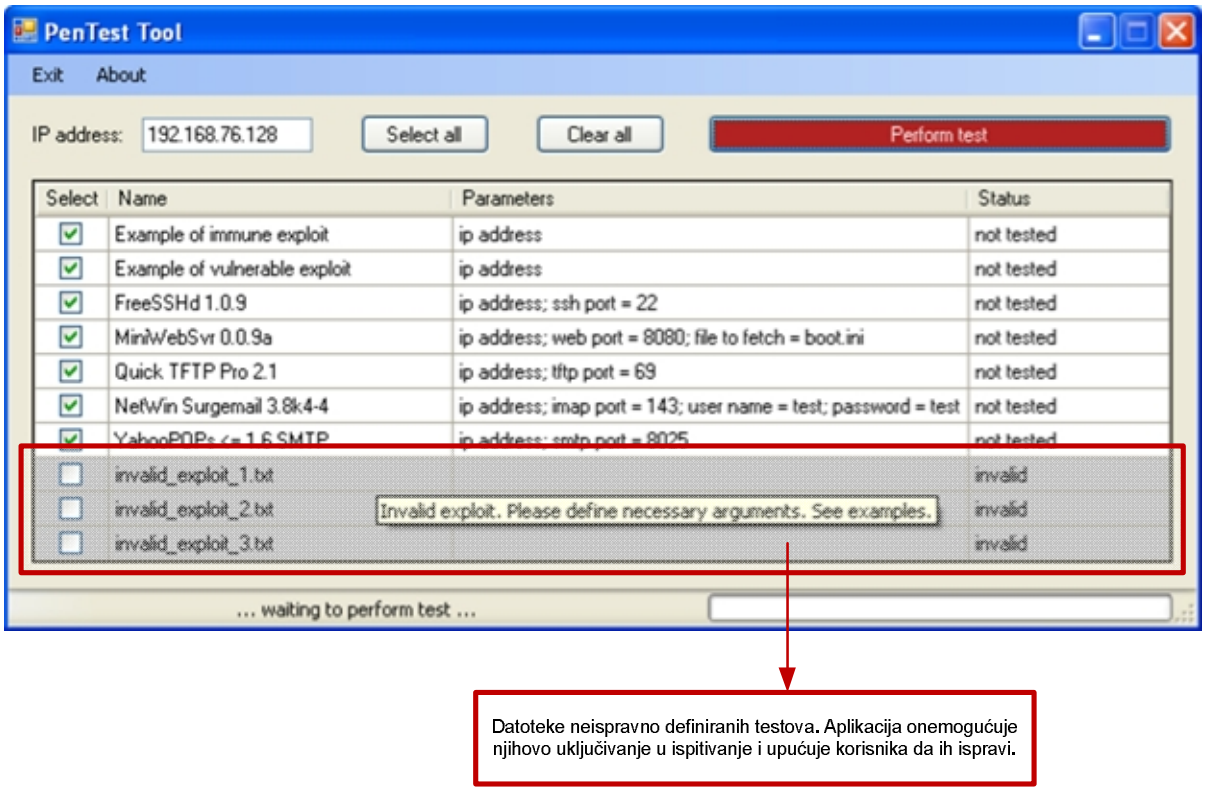
- IP adresa računala koje se ispituje
- Gumbi za upravljanje procesa ispitivanja
- Lista testova, opcija i informacija o njima
- Statusna traka za opis tijeka izvođenja ispitivanja

Aplikacija prilikom pokretanja učitava testove iz baze, a važne informacije napada vizualno predočuje korisniku. Korisniku je omogućeno da odabere sve ili samo neke od testova s kojima želi provesti ispitivanje ranjivosti sustava. Prije pokretanja ispitivanja potrebno je zadati IP adresu računala koja se ispituje, dok se ispitivanje provjere ranjivosti sustava pokreće pritiskom gumba *"Perform test"*.

Uz ispravno definirane testove, na slici 6.4. prikazano je da aplikacija također obavještava korisnika i o neispravno definiranim testovima koji nisu zadovoljili minimalne uvijete. Na taj način obavještava korisnika da poduzme odgovarajuće radnje da ih ispravi ili obavijestiti nadležnu osobu o mogućim greškama.

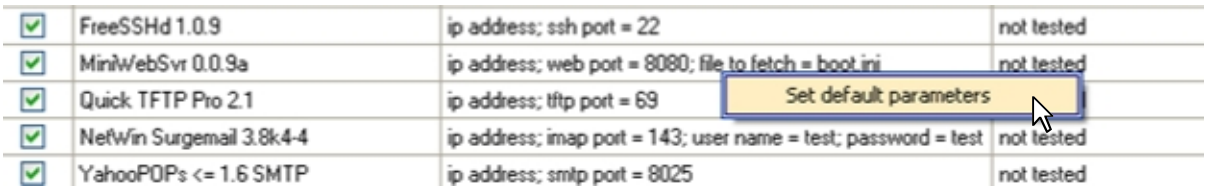


Slika 6.3. Pregled osnovnih cjelina aplikacije



Slika 6.4. Obavijest o neispravno definiranim datotekama testova

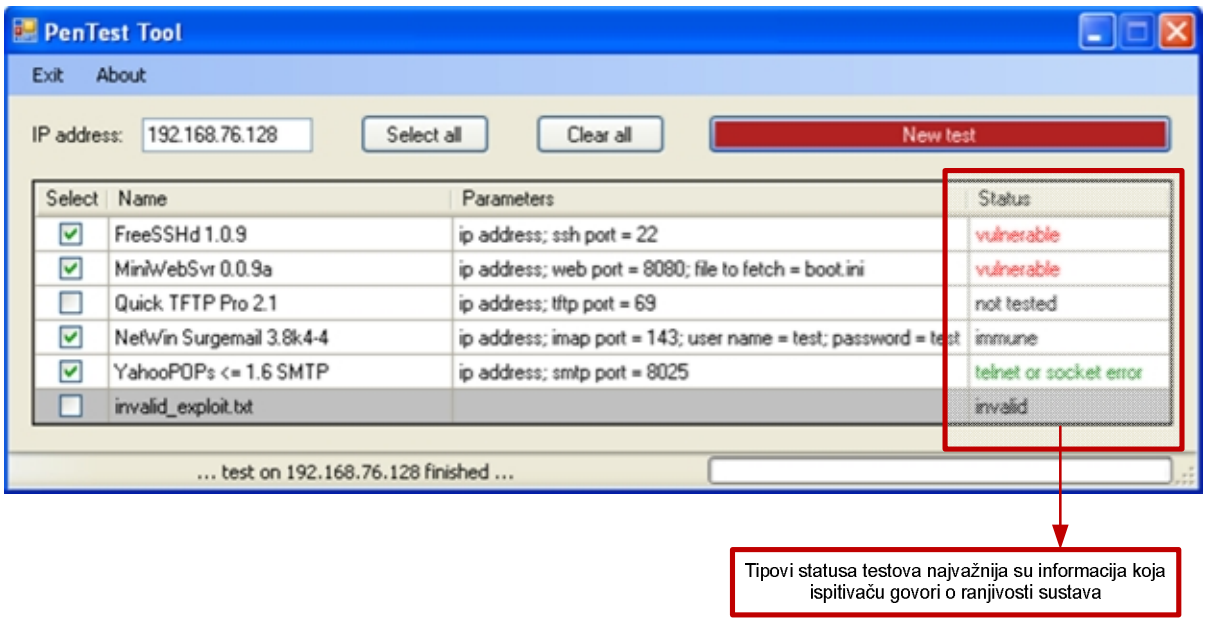
Za svaki test aplikacija prikazuje njegovo ime. Također prikazuje i ulazne parametre koje korisnik može proizvoljno mijenjati te ih prilagoditi vlastitim potrebama. U svakom trenutku korisnik također može vratiti predložene parametre za svaki pojedini test. Slika 6.5. prikazuje postavljanje predloženih parametara. Korisnik desnim klikom na polje parametara svakog pojedinog testa treba odabrati opciju *"Set default parameters"*.



Slika 6.5. Postavljanje predloženih parametara testa

Za svaki test prati se i njegov status kojim se korisnika obavještava o uspjehu, neuspjehu ili greški prilikom izvođenja. Slika 6.6. prikazuje neke od mogućih statusa testa nakon obavljenog ispitivanja na računalu. To su sljedeće mogućnosti:

- **not tested** – računalo nije bilo podvrgnuto ispitivanju dotičnog testa
- **immune** – sustav je otporan na dotični test
- **vulnerable** – sustav ima propust i ranjiv je na dotični test
- **invalid** – neispravno definirana datoteka testa
- ostale vrijednosti rezultat su iznimke, greške koju može izbaciti dotični test – primjerice **socket error**



Slika 6.6. Mogući tipovi statusa testa

Kako proces ispitivanja traje duži vremenski period, već ovisno o broju testova koji se ispituje, aplikacija također javlja stanje toka ispitivanja u statusnoj traci.

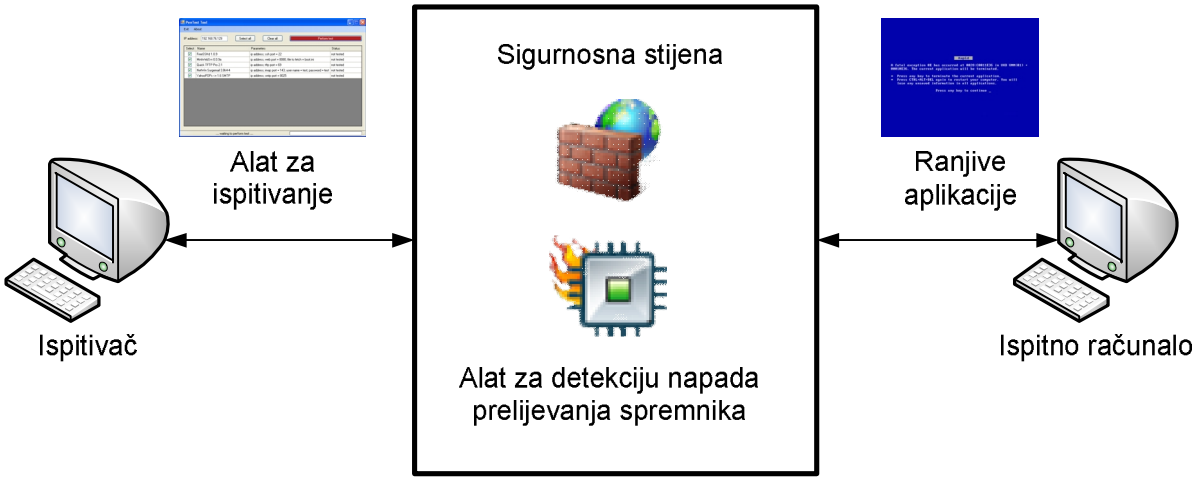
6.3. Ispitivanje rada aplikacije

Ispitivanju ranjivost sustava podvrgnuto je virtualno računalo na koje je instaliran Windows XP operacijski sustav uz paket zakrpa druge verzije. Na ispitno računalo također je instalirano i nekoliko aplikacija u kojima su pronađeni propusti. Računalo se ispitivalo u dva slučaja.

U prvom slučaju, slika 6.7., pristup ispitivača testiranom računalu bio je neometan, tj. računalo nije bilo štićeno nikakvim tehnologijama zaštite.

Slika 6.7. Ispitna okolina bez zaštite

U drugom pak je slučaju, na slici 6.8., na ispitivanom računalu bila postavljena klasična sigurnosna stijena koja dolazi kao sastavni dio Windows XP operacijskog sustava, te besplatan alat za otkrivanje i prevenciju napada prelijevanja spremnika (*Comodo Memory Firewall*).

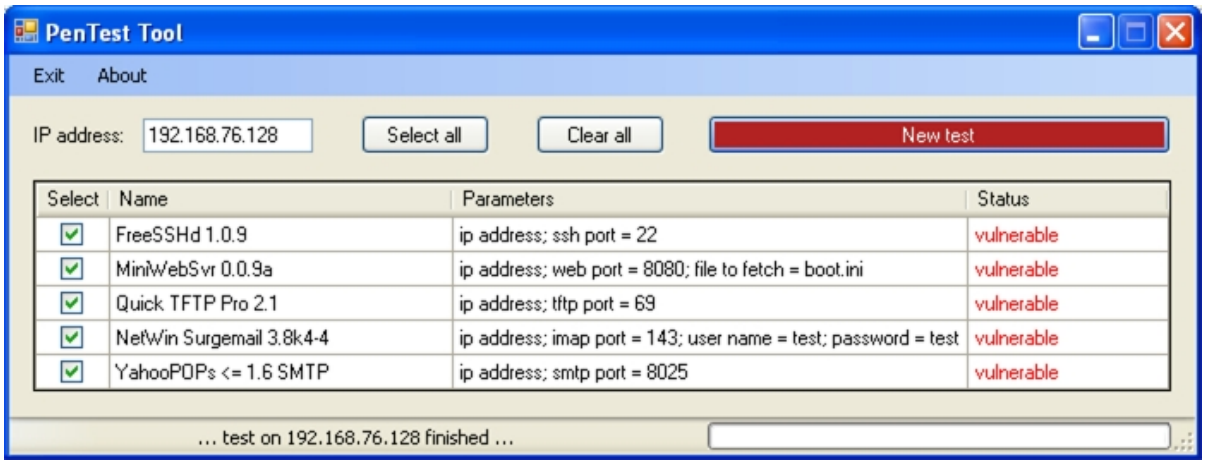


Slika 6.8. Ispitna okolina uz zaštitu

Većina testova koja je pisana i prilagođena za ovu aplikaciju, prema njenim standardima, koristila je metodu napada prelijevanja spremnika za dobivanje neovlaštenog pristupa računalu. To su bili testovi na aplikacije *FreeSSHd 1.0.9*, *Quicq TFTP Pro 2.1*, *NetWin Surgemail 3.8k4-4* te *YahooPOPs 0.6*. Također je vršeno ispitivanje testa na aplikaciju

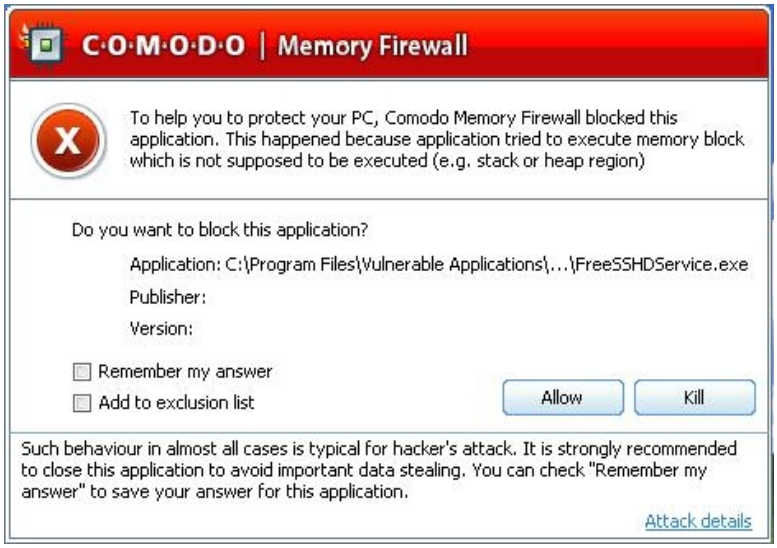
MiniWebSvr 0.0.9a u kojoj je otkriven propust mogućnosti pristupanja bilo kojoj datoteci na disku.

Nakon izvršenog ispitivanja računala bez zaštite rezultati su bili očekivani. Svaki test ukazao je na ranjivost sustava, zbog nedostatka zaštite na primijenjene napade. Slika 6.9. prikazuje ishod aplikacije za ispitivanje koja ispitivaču daje sliku poražavajućeg sigurnosnog stanja ispitivanog sustava.



Slika 6.9. Rezultati testiranja računala bez zaštite

U drugom slučaju, kada je računalo bilo štićeno sigurnosnom stijenom te alatom za otkrivanje i prevenciju prelijevanja spremnika pokazalo se da je zaštita na računalu bila korisna u određenoj mjeri. Neki od testova uspješno su uskraćeni i zabilježeni na ispitivanom sustavu. *Comodo Memory Firewall* prepoznao je neke od pokušaja napada kao napade prelijevanja spremnika. Slika 6.10. prikazuje da je za svaki otkriveni pokušaj napada korisniku javljeno da se radi o neovlaštenoj aktivnosti na računalu.

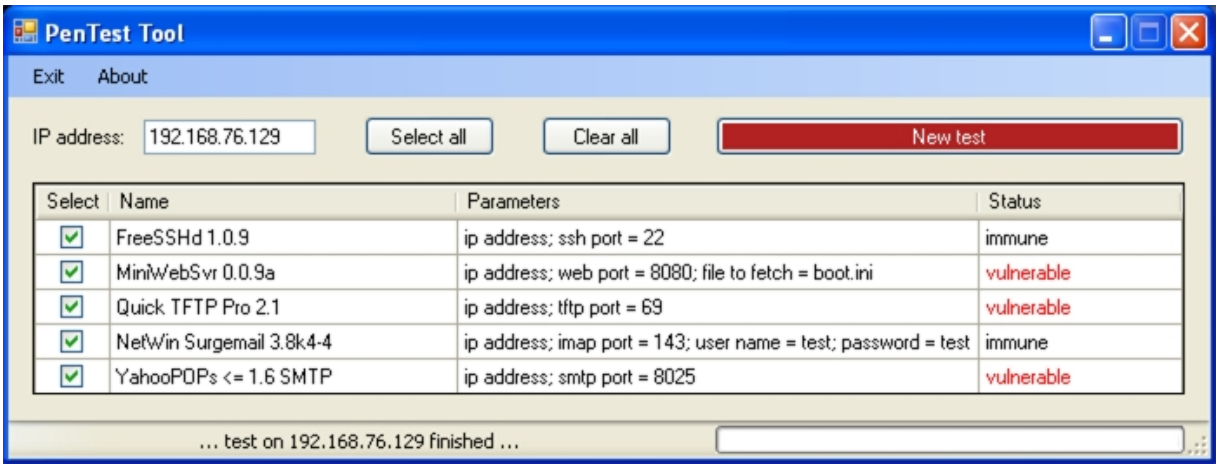


Slika 6.10. Dojava otkrivenog pokušaja napada na aplikaciju FreeSSHd

Svaki pokušaj napada također je zabilježen u dnevnički zapis aplikacije koja je štitila ispitivano računalo, što je prikazano na slici 6.11. Otkriveni su i zabilježeni pokušaji prelijevanja spremnika aplikacije *FreeSSHd* i *NetWin Surgemail*.

Slika 6.11. Dnevnički zapis alata za prevenciju prelijevanja spremnika

Sa strane ispitivač, iz rezultata aplikacije izrađene u ovom radu, prikazanih na slici 6.12., također se jasno može vidjeti da su otkriveni pokušaji napada neiskoristivi. Ispitivanje je pokazalo da je sustav imun na dotične napade upravo zbog primijenjene zaštite na sustav. Ali, također je potrebno uočiti da su neki od pokušaja napada prelijevanja spremnika uspješno provedeni te ih ispitivano računalo nije otkrilo. Razlog tome je nesavršenost alata za zaštitu. Aplikacija za ispitivanje te testove predočuje ispitivaču statusom ranjivosti.



Slika 6.12. Rezultati testiranja računala sa zaštitom

7. Zaključak

Svaki sustav podložan je programerskim pogreškama, a većina takvih grešaka ima kritične posljedice. Motivirani napadači veoma lako pronalaze takve propuste i jednako lako ih iskorištavaju. Složenost sustava zahtjeva pažljivu konfiguraciju, a pogreške u jednom sustavu mogu se odraziti u drugim sustavima. Propusti li se pravovremeno prepoznavanje i otklanjanje računalne ranjivosti na svim sustavima, a naročito onim koji se nalaze na Internetu, cjelokupni sustav se izlaže velikim rizicima.

Rezultati ispitivanja provedenih u praktičnom djelu ovog rada pokazuju da je zaštita računala automatiziranim alatima potrebna, ali nije i dovoljna. Ne postoji savršeni mehanizam zaštite koji može potpuno garantirati sigurnost nekog sustava. Ispitivanje je također s druge strane pokazalo da je mehanizmima zaštite moguće otkriti mnoge od zlonamjernih radnji napadača i spriječiti ga u njegovim postupcima.

Kako bi se stoga sustav i informacije održale sigurnima, svakako ga je potrebno zaštititi odgovarajućim automatiziranim mehanizmima. No, druga vrlo važna praksa, podrazumijeva redovito ispitivanje i vrednovanje sigurnosne politike koju sustav posjeduje.

Jedan od najpouzdanijih načina koji nam daje uvid u sigurnost sustava je provođenje penetracijskog testa. Provođenjem penetracijskog testa repliciraju se tipovi akcija koje bi poduzeo zlonamjerni napadač dajući nam time precizniju reprezentaciju u sigurnosno stanje sustava. Treba voditi računa da penetracijski test ne traje za uvijek, već je to samo pogled na stanje sigurnosti sustava u određenom trenutku vremena. Trenutno poznati sigurnosni propusti, ranjivosti ili neadekvatna konfiguracija prisutna u sustava neće biti uklonjeni sve do vremenskog trenutka provođenja penetracijskog testa. Stoga je potrebno redovito provođenje penetracijskog testa.

Aplikacija izrađena u ovom radu je pomoćni automatizirani alat za ispitivanje ranjivosti sustava. Prednost ove aplikacije je što omogućuje lako dodavanje novih testova, te se stoga jednostavno nadograđuje najsvježijim propustima. Ovu aplikaciju penetracijski ispitivač može koristiti prilikom ispitivanja sigurnosti, te na poprilično brz način, bez trošenja vremena na rutinske poslove, dobiti uvid o ranjivostima sustava. Prilikom ispitivanja sigurnosti, osim pukog korištenja aplikacije vrlo je važno da ispitivač koji je koristi bude stručnjak na području sigurnosti te da pažljivo promišlja o svojim postupcima prije izravnog ispitivanja. Ljudska inteligencija, snalažljivost, iskustvo i spretnost još uvijek su najvažniji vodiči kako u izvođenju procesa napada tako i u procesu zaštite sustava.

8. Literatura

1. Cole, E.; Krutz, R.; Conley, J.W.: **Network Security Bible**, Wiley Publishing, Inc., 2005.
2. Cross, M.; Johnson, N. L.; Piltzecker, T.: **Security+**, Syngress Publishing, Inc., 2003.
3. Pastore, M.: **Security+**, SYBEX, Inc., Alameda, 2003.
4. Tiller, S. J.: **The Ethical Hack**, Auerbach Publications, 2003.
5. Northcutt, S.; Novak, J.: **Network Intrusion Detection**, New Riders Publishing, 2002.
6. Whitaker, A.; Newman, D. P.: **Penetration Testing and Network Defense**, Cisco Press, 2006.
7. Bradley, T.: **Essential Computer Security**, Syngress Publishing, Inc., 2006.
8. Pereiera, J.P.: **Comparison of Firewall, Intrusion Prevention and Antivirus Technologies**, Juniper Networks, Inc., Sunnyvale, 2004.
9. Saindane, M. S.: **Penetration testing – A Systematic Approach**. Infosec Writers, dostupno na Internet adresi: http://www.infosecwriters.com/text_resources/pdf/PenTest_MSaindane.pdf. (09.07.2008).
10. Northcutt, S.; Shenk, J.; Shackleford, D.; Rosenberg, T.; Siles, R.; Manchini, S. **Penetration Testing : Assessing Your Overall Security Before Attackers Do**. SANS Institute, dostupno na Internet adresi: http://www.sans.org/reading_room/analysts_program/PenetrationTesting_June06.pdf. (25.7.2008).
11. Tuck Wai, C.: **Conducting a Penetration Test on an Organization**. SANS Institute, dostupno na Internet adresi: http://www.sans.org/reading_room/whitepapers/auditing/67.php. (20.07.2008).
12. **Information security**, dostupno na Internet adresi: http://en.wikipedia.org/wiki/Information_security. (12.6.2008.).
13. **OSI model**, dostupno na Internet adresi: http://en.wikipedia.org/wiki/OSI_model. (15.6.2008.).