Projektiranje sigurnosti informacijsko-komunikacijskog sustava primjenom mehanizma Honeypot

Nagy, Tomislav

Master's thesis / Diplomski rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:119:815348

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**



Repository / Repozitorij:

Faculty of Transport and Traffic Sciences - Institutional Repository







SVEUČILIŠTE U ZAGREBU

FAKULTET PROMETNIH ZNANOSTI

Tomislav Nagy

PROJEKTIRANJE SIGURNOSTI INFORMACIJSKO-KOMUNIKACIJSKOG SUSTAVA PRIMJENOM MEHANIZMA HONEYPOT

DIPLOMSKI RAD

Zagreb, 2016.

Sveučilište u Zagrebu

Fakultet prometnih znanosti

DIPLOMSKI RAD

PROJEKTIRANJE SIGURNOSTI INFORMACIJSKO-KOMUNIKACIJSKOG SUSTAVA PRIMJENOM MEHANIZMA HONEYPOT

DESIGN OF SECURITY OF INFORMATION AND COMMUNICATION SYSTEM USING THE HONEYPOT METHODOLOGY

Mentor: izv. prof. dr. sc. Dragan Peraković

Student: Tomislav Nagy

JMBAG: 0246001960

Zagreb, rujan 2016.

PROJEKTIRANJE SIGURNOSTI INFORMACIJSKO-KOMUNIKACIJSKOG SUSTAVA PRIMJENOM MEHANIZMA HONEYPOT

SAŽETAK

Honeypot sustav je sigurnosni resurs čija vrijednost leži u mogućnosti da bude ispitan, napadnut ili kompromitiran. Administrator sustava je u mogućnosti prikupiti informacije o napadaču, vektorima napada i alatima koje napadač koristi kako bi zadobio pristup napadnutom sustavu. U svrhu ovog istraživanja korišteni su *Valhala* i *Bifrozt* alati. *Valhala* je nisko/srednje interaktivni *Honeypot* sustav što znači da napadač u interakciji sa sustavom može koristiti samo ograničen popis naredbi. *Valhala Honeypot* sustav uspješan je u detekciji raznih automatiziranih skripti, no vješti napadači su u mogućnosti vrlo brzo otkriti prisutnost takvog sustava. *Bifrozt* je visoko interaktivni *Honeypot* sustav. Napadaču je dana mogućnost koristiti cijeli sustav i svaka je od usluga istovjetna produkcijskoj usluzi. Administrator je u mogućnosti prikupiti veliku količinu informacija, a da pri tome napadač toga nije ni svjestan. S obzirom na to da je cilj istraživanja bio otkriti mogu li *Honeypot* sustavi uspješno zabilježiti upad u sustav, otkriti informacije o napadaču, vektorima napada i alatima koje napadači koriste, oba su *Honeypot* sustava zadovoljila.

KLJUČNE RIJEČI: Botnet mreže; Honeypot; Honeynet; Valhala Honeypot; Bifrozt Honeypot

SUMMARY

Honeypot system is a security resource whose value lies in the possibility of being tested, attacked or compromised. The system administrator is able to collect information about the attacker, attack vectors and tools that a hacker uses to gain access to the attacked system. For the purposes of this study were used Valhalla and Bifrozt tools. Valhalla is low / medium interactive Honeypot system which means that an attacker can only use a limited list of commands to interact with the system. Valhala Honeypot system was successful in detecting a variety of automated scripts, but skilled attackers are still able to quickly detect the presence of such a system. Bifrozt is highly interactive Honeypot system. The possibility that is given to the attacker is to interact with the whole system and each of the services is identical to the production system. The administrator is able to collect a large amount of information, while at the same time attacker was not even aware of it. Considering that the aim of the research was to discover whether Honeypot systems can successfully record an intrusion in the system, reveal information about the attacker, attack vectors and tools used by the attackers, both Honeypot systems satisfied requirements.

KEYWORDS: Botnet networks; Honeypot; Honeynet; Valhala Honeypot; Bifrozt Honeypot

SADRŽAJ

1. UVOD 1
2. DEFINICIJA NAPADAČA 2
2.1. Opis napadača 2
2.2. Povijest i motivi napadača 3
2.3. Alati kojima se koriste napadači 4
2.4. Etička načela korištenja računala 5
3. *BOTNET* MREŽE 7
3.1. Ciljevi *Botnet* mreže 7
3.2. Vrste napada *Botnet* mreža 7
3.3. Arhitektura *Botnet* mreže 9
3.4. Primjer *Botnet* napada 10
3.5. Zaštita od *Botnet* napada 12
4. *HONEYPOT* SUSTAVI 14
4.1. Povijest *Honeypot* sustava 14
4.2. Vrste *Honeypot* sustava 15
4.2.1. Nisko interaktivni *Honeypot* sustavi 15
4.2.2. Srednje interaktivni *Honeypot* sustavi 16
4.2.3. Visoko interaktivni *Honeypot* sustavi 16
4.2.4. Klijentski i poslužiteljski *Honeypot* sustavi 17
4.3. Strategija implementacije *Honeypot* sustava 18
4.3.1. Klasična implementacija orijentirana prema Internetu 19
4.3.2. Unutarnja implementacija 19
4.4. Prednosti i mane korištenja *Honeypot* sustava 19
5. *HONEYNET* SUSTAVI 21
5.1. Arhitektura *honeynet* sustava 22
5.1.1. Nadzor podataka 22
5.1.2. Prikupljanje podataka 23
5.1.3. Analiza podataka 23
5.1.4. Upravljanje podacima 23
5.2. Prednosti korištenja *honeynet* sutava 24
5.3. Prva, druga i treća generacija *honeynet* sustava 25

6. SIMULACIJA NAPADA 30
6.1. *Valhala Honeypot* sustav 30
6.1.1. Topologija mreže *Valhala Honeypot* sustava 30
6.1.2. Općenito o *Valhala honeypot* sustavu 32
6.1.3. Pristup FTP poslužitelju 34
6.1.4. Pristup *Telnet* poslužitelju 37
6.1.5. Pristup SMTP poslužitelju 39
6.1.6. Pristup *Finger* poslužitelju 40
6.1.7. Pristup *Web* poslužitelju 41
6.1.8. Pristup *POP3* poslužitelju 42
6.1.9. Rezultati simulacije *Valhala Honeypot* sustava 43
6.2. *Bifrozt Honeypot* sustav 44
6.2.1. Topologija mreže 45
6.2.2. Administracija *Bifrozt* proxya 46
6.2.3. Simulacija napada na *Bifrozt* sustav 48
6.2.4. *Spoofing* korisničkog imena i zaporke 49
6.2.5. Prijenos datoteka SCP protokolom 51
6.2.6. Rezultati simulacije *Bifrozt Honeypot* sustava 52
7. ZAKLJUČAK 54
8. POPIS LITERATURE 55
9. POPIS ILUSTRACIJA 57

1. UVOD

Računalne mreže sve su izloženije napadima koji podrazumijevaju krađu povjerljivih informacija te uništenju podataka na računalu. Napadači, kako bi zadobili pristup sustavu, koriste propuste u operativnim sustavima i programima. Postojeće zaštite kao što su vatrozidi i antivirusni programi često nisu dovoljni kako bi se spriječili takvi napadi. U svrhu toga dizajnirani su računalni mamci odnosno *Honeypot* sustavi. To je tehnologija koja se razvija već desetak godina. U zaštiti računalnih mreža njihova popularnost sve više raste. Primjenom *Honeypot* sustava, reakcija se na postojeće prijetnje smanjuje.

Svrha je istraživanja *Honeypot* sustava dobiti odgovor na pitanje kako pravovremeno i u cijelosti zaštititi informacije koje se nalaze na računalnoj mreži korisnika. Primjenom *Honeypot* sustava korisnik je u mogućnosti prikupiti informacije o napadaču, tehnikama napada i alatima koje napadač koristi. Nakon što korisnik dobije uvid u stanje sigurnosti sustava u mogućnosti je povećati razinu sigurnosne zaštite. Cilj je istraživanja ispitati je li način na koji *Honeypot* sustav radi ispravan i nudi li korisniku bolju zaštitu. Rad je podijeljen u sedam cjelina.

1. Uvod
2. Definicija napadača
3. *Botnet* mreže
4. *Honeypot* sustavi
5. *Honeynet* sustavi
6. Simulacija napada
7. Zaključak

U drugom poglavlju opisani su napadači, njihova povijest, motivi napada i alati koje napadači koriste. U trećem poglavlju opisane su tehnike napada, način stvaranja distribuirane mreže napadača i zaštite od istih. U četvrtom poglavlju opisani su računalni mamci, njihova povijest, klasifikacija prema namjeni, strategija implementacije, prednosti i mane. U petom poglavlju opisani su visoko interaktivni računalni mamci, način priupljanja i analize dobivenih podataka. U šestom poglavlju opisani su *Valhala* i *Bifrozt Honeypot* alati, podešavanje parametara i simulirani su napadi na *Honeypot* sustave.

2. DEFINICIJA NAPADAČA

2.1. Opis napadača

Klasični se pristup analizi mrežne sigurnosti temelji na korištenju raznih tehnologija poput sustava za otkrivanje upada (IDS – *Intrusion Detection System*), sustava za prevenciju upada (IPS – *Intrusion Prevention System*), vatrozida te naprednijih vrsta vatrozida (UTM – *Unified Threat Management*) koje obuhvaćaju funkcije IPSa, antivirusa, antispama, filtriranja na bazi sadržaja, balansiranja opterećenja linka, zaštitu od gubitaka podataka i VPN – *Virtual Private Network*. Potrebno je pokriti navedena tehnološka područja s namjerom da se postigne organizacijska sigurnost s osnovama mrežne sigurnosne arhitekture, dizajna, politike i procedure. Zajednički je cilj, osobama kojima prijeti napad hakera, osigurati "prihvatljivi sigurnosni rizik" i ciljati na mrežni sigurnosni model upotrebljavajući druge sigurnosne sustave i sustave zaštite.

Ukoliko se u obzir ne uzme ljudski faktor, nije moguće efikasno zaštititi računalne mreže i informacijske sustave te je stoga potrebno definirati vrste napadača. Napadačem se smatra osoba koja svoje računalno znanje koristi kako bi ugrozila sigurnost računala ili podataka pohranjenih na računalu, a često ih se naziva hakerima. Treba naglasiti da se napadači koji provode zlonamjerne ili kriminalne aktivnosti nazivaju *crackerima* kako bi se napravila jasna razlika između njih i hakera koji djeluju prema etičkim načelima. Haker nema namjeru nanošenja zla i štete ljudima i računalnim sustavima. Hakera se može definirati kao [1]:

1. osobu koja uživa istraživati detalje programskih sustava te kako povećati njihove kapacitete i poboljšati učinkovitost
2. nekoga tko programira s puno entuzijazma (ponekad fanatizma) te mu je sam posao programiranja ispred bilo koje druge aktivnosti
3. osobu koja uživa u intelektualnom izazovu rješavajući ga koristeći inteligenciju i kreativnost.

Crackerom se smatra osoba koja pokušava provaliti u računalo drugog korisnika pri tome koristeći sve dostupne metode i alate (crve, trojanske konje, *rootkitove*, sigurnosne propuste, društveni inženjering, i sl.) kako bi ga oštetila ili ukrala informacije. Postoje dvije vrste *crackera* [2]:

1. oni koji posjeduju računalna znanja koja su potrebna da bi sami napisali zlonamjerne programe i
2. oni koji se samo znaju koristiti tim alatima.

2.2. Povijest i motivi napadača

Prvi poznati incident vezan za hakiranje datira iz 1878. godine. U tvrtki *Bell Telephone Company*, skupina dječaka zaposlenih na poziciji telefonskih operatera prespajali su, prekidali i prisluškivali pozive korisnika. Oko 1960-tih riječ *hack* postala je sinonimom za zaobilaženje standardnog načina funkcioniranja nekog sustava. Izraz je došao od strane entuzijasta na MIT-u (*Massachusetts Institute of Technology*) koji su izmijenili karakteristike maketa vlakova, što su kasnije primijenili na novim računalnim sustavima. Rani hakeri bili su vrhunski programeri posvećeni modifikaciji programa u svrhu poboljšanja njihovih karakteristika. U mnogim slučajevima kvaliteta koda je bila elegantnije napisana naspram inačica profesionalnih programa. Možda je najbolji primjer entuzijazma iz tog vremena razvoj UNIX operacijskog sustava od strane Dennisa Ritchieja i Keitha Thomsona. Desetak godina kasnije pojavljuje se nova vrsta hakera koji svoj fokus stavljaju na hakiranje telefonskih sustava. Poznati su pod imenom *phreakers.* Njihov je cilj bio upadanje u mrežu telefonskog operatera u svrhu besplatnog telefoniranja [3].

Zlatno doba hakiranja kreće od 1980-ih godina. Računala nisu više predmetom istraživanja zagriženih entuzijasta i poslovnih korisnika, već se područje interesa seli na „običnog“ korisnika. Uz pomoć modema, uređaja koji omogućuje komunikaciju između dva računala putem telefonske linije, doseg hakiranja značajno se proširio. Kasnih 1980-ih dolazi do podjele među hakerima. Velik broj njih više nije bio zadovoljan proučavanjem sustava u svrhu učenja, već su fokus stavili na iskorištavanje sustava u svrhu osobne koristi. Doneseni su zakoni koji takve radnje kriminaliziraju u vidu zatvora i većih novčanih kazni.

Oko 1990-ih broj je kriminalnih radnji koje vlade i poslovni sustavi trpe sve veći i veći. Jedan od najpoznatiji hakera tog doba, Kevin Mitnick, uhićen je čak dva puta te je odslužio nekoliko godina zatvorske kazne i dobio zabranu pristupa računalu na nekoliko godina. Najnoviji oblik hakiranja uključuje pronalaženje i spajanje na nezaštićene bežične mreže, a poznato je po imenu *whacking* [2].

Motivi hakera mogu se podijeliti u tri kategorije [1]:

1. Iz rekreacije – oni koji upadaju u računalne mreže i sustave iz „zabave“ ili samo kako bi dokazali da imaju potrebne vještine i znanja da to naprave.
2. Radi novčane naknade – oni koji hakiraju računalne sustave iz osobne koristi te zarađuju na ucjenama ili su plaćeni od strane treće osobe kako bi upadali u računalne sustave
3. Iz osvete – tu spadaju nezadovoljni korisnici ili bivši zaposlenici te ljutiti konkurenti.

2.3. Alati kojima se koriste napadači

Postoji nekoliko učestalih alata koje koriste računalni kriminalci u svrhu penetracije kako bi zadobili pristup resursima neke mreže, a to su trojanski konji, virusi, crvi, skeneri ranjivosti i razni exploiti [1].

Vrsta programa „Trojanski konj“ dobila je naziv po poznatom grčkom osvajanju grada Troje. To su programi za koje se čini da obavljaju poželjne funkcije na računalu, a ustvari obavljaju korisniku neobjavljene zlonamjerne radnje. Obično se sastoje od dva dijela, klijenta i poslužitelja. Poslužitelj se pokreće na računalu žrtve i otvara napadaču mogućnost da načini štetu ili ukrade podatke. Trojanski se konji šire kada ljudi zagrizu tzv. mamac i otvore program, misleći da on dolazi iz povjerljivog izvora.

Virusi su računalni programi koji mogu sami sebe kopirati i zaraziti korisnikovo računalo bez njegova znanja. Širenje virusa moguće je putem Interneta, mrežne strukture ili preko prijenosnih medija. Jednom kada računalo biva zaraženo virusom, isti se može kopirati i izmijeniti samog sebe kako bi ga se teže otkrilo. Najčešće se ubacuju u pokretačke datoteke programa (*executables*) te se pri pokretanju zaražene datoteke šire na druge. Da bi se virus umnožio, mora moći pokrenuti svoj kod i pisati u radnu memoriju. Iz ovog se razloga virusi ubacuju u pokretačke datoteke. Ukoliko korisnik pokrene zaraženi program, virus se učitava u memoriju.

Crv je računalni program koji također ima sposobnost umnožavanja samog sebe. Crvi koriste mrežne sustave kako bi se umnožili i zarazili računala. Za razliku od virusa, crvi ne ubacuju svoj programski kod u neki od programa na računalu. Poznato je iz prijašnjih iskustava da mogu izbrisati datoteke sa zaraženog računala, zapakirati podatke napadima ili čak poslati podatke putem elektroničke pošte. Obično sadržavaju dio koda „stražnja vrata“ koji omogućuje napadačima da korisnikovu IP adresu koriste za *phishing* ili kako bi prikrili *web* adresu svoje stranice. Otvaranjem „stražnjih vrata“ također se omogućuje lakša zaraza drugim vrstama zlonamjernih programa.

Skenere ranjivosti koriste hakeri kako bi u što kraćem roku provjerili računala na mreži s ciljem da pronađu neku vrstu ranjivosti. Kako bi zadobili pristup, često koriste skenere portova da ustanove koji su portovi „otvoreni“ na određenom računalu [2].

Jedan je od takvih alata i *nmap*. *Nmap* je besplatan program otvorenog koda koji služi za istraživanje mreže i provjeru ranjivosti. Mnogi sustavi i mrežni administratori pronalaze ga korisnim za zadatke kao što su provjera mrežnog inventara, vremensko upravljanje nadogradnje servisa i nadziranje vremena neprekidnog rada servisa ili računala. *Nmap* na temelju IP paketa analizira koja su računala dostupna, koje servise (naziv aplikacije i verzija) koriste, koji operacijski sustavi ih pokreću (operativni sustv i verzija) i koji tip vatrozida odnosno paket filtera koriste, ali i mnoge druge karakteristike. Skalabilan je alat koji je dizajniran za brzo skeniranje velikih mreža, ali također radi vrlo dobro i ako se koristi za analizu jednog računala. Radi na svim važnijim računalnim operacijskim sustavima te su službeni programski paketi dostupni na *Linux*, *Windows*

i *Mac OS X* platformi. Osim sučelja u komandnoj liniji – CLI, *Nmap* uključuje i napredno grafičko sučelje koje se naziva *Zenmap*, zatim prilagodljivi alat za otkrivanje grešaka, preusmjeravanje i prijenos podataka – *Ncat*, alat za usporedbu skeniranih rezultata – *Ndiff* i generator *IP* paketa i analizu – *Nping* [4].

Exploit se može definirati kao aplikacija koja iskorištava poznate ranjivosti nekog sustava. To može biti neki softver, dio nekog koda ili sekvenca računalnih naredbi koje iskorištavaju poznatu grešku ili ranjivost u svrhu proizvodnje neželjenih ili neočekivanih ponašanja unutar računalnog softvera, hardvera ili neke računalne elektronike. Takva neželjena ponašanja softvera odnosno hardvera često prouzrokuju zadobivanje kontrole nad sustavom koji napadač cilja, pa čak i DoS napad [5].

Izraz DoS označava napad uskraćivanja usluga. Takav napad karakterizira namjerno generiranje velike količine mrežnog prometa da bi se zasitili mrežni resursi i poslužitelji. Zbog prevelikog opterećenja oni više nisu u stanju pružati namijenjene usluge. Posljedica je toga nemogućnost legitimnih korisnika da koriste mrežne usluge poput elektroničke pošte, Interneta i sl.

Mnoge su metode klasifikacije aplikacija za iskorištavanje ranjivosti sustava, ali najčešće ih možemo podijeliti u dvije kategorije prema načinu na koji *exploit* stupa u kontakt s ranjivim softverom. Udaljeni *exploit* koji radi putem računalne mreže iskorištava sigurnosne propuste bez prethodnog pristupa tom sustavu, dok lokalni *exploit* zahtijeva prethodni pristup napadnutnog sustava na način da se povećaju prava pristupa osobe koja vrši napad na sustav [5].

2.4. Etička načela korištenja računala

Pri korištenju računalne opreme potrebno je pridržavati se nekih etičkih načela. Kako bi zaštitili samoga sebe, a i pružili sigurnost drugima, američki *Computer Ethics Institute* predstavio je tzv. Deset zapovijedi računalne etike. Ova pravila predstavljaju vrlo važan dio računalne kulture pa ih se navodi u nastavku kako bi ih svaki korisnik pri korištenju računalne opreme imao na umu.

Deset zapovijedi računalne etike [1], [6]:

1. Računalo se ne smije koristiti da bi se naudilo drugima.
2. Ne smije se uplitati u računalne poslove drugih ljudi.
3. Ne smije se neovlašteno pristupati dokumentima na računalima drugih ljudi.
4. Računalo se ne smije koristiti za krađu.
5. Računalo se ne smije koristiti za lažno predstavljanje.

6. Na računalu se ne smije umnožavati ili koristiti zakonom zaštićene programske pakete koje korisnik nije platio.

7. Nije dozvoljeno upotrebljavati računalne resurse drugih ljudi bez dozvole ili odgovarajuće naknade.

8. Nije dozvoljeno prisvajanje intelektualnog vlasništva drugih ljudi.

9. Potrebno je misliti o društvenim posljedicama programa koji pišemo ili sustava koji razvijamo.

10. Računalo se uvijek mora koristiti na načine koji osiguravaju uvažavanje i poštovanje drugih.

Pri svakoj upotrebi računala potrebno je savjesno i odgovorno ponašanje upravo iz razloga kako ne bi napravili štetu drugima, kompromitirali sigurnost njihove računalne opreme i podataka na istoj. Isto tako, potrebno je voditi računa o sigurnosti vlastitih podataka i pravovremeno se zaštititi od moguće krađe ili gubitka podataka. Često se događa u praksi da se ljudi ne pridržavaju ovih etičkih načela, što je upravo i razlog koji dovodi u pitanje računalnu sigurnost. Kako bi se korisnici ipak uspjeli obraniti, potrebno je razvijati sustave poput računala mamaca, antivirusnih programa, vatrozida i njima sličnih.

3. *BOTNET* MREŽE

Procesorska snaga i računalni resursi općenito mogu biti povećani umrežavanjem računala. Iako svako pojedino računalo nema impresivnu snagu, čak je možda i pomalo zastarjelo, svejedno doprinosi ukupnoj snazi tako dobivenog grozda računala. Samo višak procesorskog vremena na svim tim računalima mogao bi dostići snagu mjerljivu u teraflopima (jedan je teraflop trilijun operacija s pomičnom točkom u sekundi), što naravno ovisi o količini računala u *Botnet* mreži.

Važno je napomenuti da postoje legalne i legitimne mreže sa sličnim svojstvima, a glavna je razlika pristanak vlasnika računala da sudjeluje u toj mreži te preda višak svojih računalnih resursa na korištenje „vlasniku" mreže. Takve mreže računala, odnosno njihova procesorska snaga, većinom se koriste u znanosti prilikom obrade podataka nekih istraživanja. I to je neka vrsta *Botnet* mreže, odnosno samo računalo te osobe je *bot* (skraćenica od robot) odakle dolazi i naziv *Botnet* za tako umrežena računala [7].

3.1. Ciljevi *Botnet* mreže

Danas je puno veća vjerojatnost da iza neke *Botnet* mreže stoji kriminalna organizacija, a ne skupina računalnih entuzijasta. Glavni je razlog tomu razlika u motivaciji. Dok su prije 2000. godine *Botnet* mrežama upravljali entuzijasti željni slave te su ih koristili za međusobno „ratovanje" i nadmetanje, danas je glavna motivacija profit. *Botnet* mreže danas se najčešće koriste za četiri vrste napada, a to su distribuirani napad uskraćivanjem usluga, slanje velikih količina neželjene elektroničke pošte, pokušaj krađe identiteta i privatnih korisničkih podataka i prevare klikom [8].

3.2. Vrste napada *Botnet* mreža

Svaki poslužitelj ima neku granicu propusnosti podataka, odnosno kapacitet koliko najviše radnji može napraviti u određenoj jedinici vremena. Ako se ta granica prijeđe, poslužitelj se može srušiti, odnosno postati nedostupan ili barem prespor za normalno korištenje. U današnje vrijeme većina tvrtki ovisi o svom *on-line* identitetu, odnosno dostupnosti barem osnovnih informacija o tvrtki na Internetu. Štoviše, velik broj tvrtki obavlja barem dio svojih djelatnosti preko Interneta. U današnje se vrijeme obavlja i većina financijskih transakcija preko Interneta. Iz svega navedenog očito je da je danas svakoj tvrtki važna dostupnost, odnosno ispravan rad njihovog poslužitelja ili poslužitelja tvrtke koja za njih drži *web* stranice ili obavlja financijske usluge. Prilikom distribuiranog napada uskraćivanjem usluga, upravitelj *Botnet* mreže naredi svim računalima koje nadzire, odnosno svim računalima u mreži da pristupe određenoj Internet stranici, usluzi, odnosno

poslužitelju u isto vrijeme. Na taj način, ovisno o broju računala koja nadzire, ozbiljno ugrozi rad tog poslužitelja. Financijska motivacija iza ovakve vrste napada najčešće je iznuda novca od napadnute tvrtke za prestanak takvih napada. Međutim, moguć je i scenarij u kojem konkurentska tvrtka plati napadačima oštećivanje konkurencije zbog povećanja svog tržišta ili neke vrste odmazde [9].

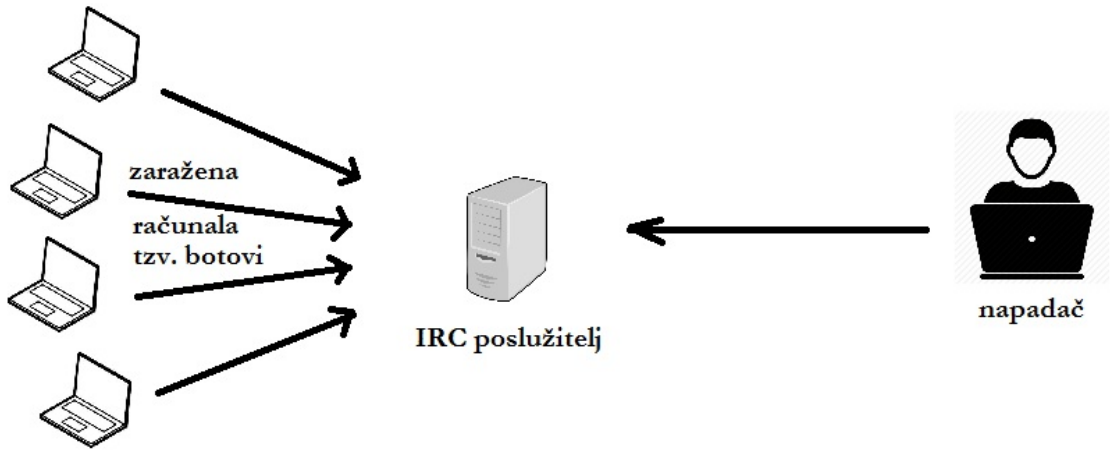
Zaražena računala pod nadzorom kriminalaca često se koriste i za slanje neželjene elektroničke pošte. Financijska korist ovakve vrste napada krije se u sadržaju te neželjene pošte. Naime, ona se najčešće koristi kao sredstvo marketinga određenog proizvoda, a prema istraživanju Sveučilišta u Kaliforniji, provedenom 2008. godine, uspješnost reklamiranja preko neželjene elektroničke pošte iznosi 0,00001%. Koliko se god ta brojka čini malena, kada se uzme u obzir količina neželjene pošte koju najveće *Botnet* mreže, zvane i *spambots*, mogu poslati u jednom danu te cijenu reklamiranog proizvoda, dolazimo do dnevne zarade od preko 400 000 američkih dolara. Štoviše, prema riječima istraživača većina spambotova *Botnet* mreža korištenih za slanje neželjene elektroničke pošte ima uspješnost od oko 2%, a najveći *Botnet* koji se koristi za slanje neželjene elektroničke pošte, *Cutwail*, može poslati 74 milijarde neželjenih poruka na dan. Svakako treba napomenuti da proizvod naveden u poruci ne postoji te da se njegovom „kupnjom“ ustvari financira daljnje slanje neželjene pošte, odnosno kriminal [7].

Većina je računala u *Botnet* mreži zaražena i nekom vrstom zlonamjernog programa koji prati aktivnost korisnika. Takvi su programi poznati i pod imenom *spyware*. Ti su programi napisani na način da prate unos korisničkih podataka s tipkovnice, a napadačima su prvenstveno zanimljive financijske transakcije, odnosno financijski podaci korisnika. Nakon krađe korisnikovih pristupnih podataka određenim financijskim uslugama, kao što je internet bankarstvo, *PayPal* ili broj kreditnih kartica, taj program ih šalje upravitelju *Botnet* mreže te on ima pristup financijama zaraženog korisnika.

Popularnost određenih Internet stranica i usluga se, između ostalog, mjeri i brojem posjeta tim stranicama. Prilikom prevare klikom, upravitelj *Botnet* mreže pokušava umjetno podići popularnost određene stranice tako da naredi velikom broju računala da posjete tu stranicu, stvarajući prividno veliki interes za njenim sadržajem. Također, vlasnik te stranice može zarađivati oglašavanjem drugih tvrtki i plaćen je po posjetu stranici, odnosno broju prikazivanja reklama. S druge strane, ovakve prevare mogu biti korištene na financijsku štetu tvrtkama koje su žrtve napada. Kako većina oglašavanja na Internetu funkcionira na principu naplate po broju prikaza ili posjeta, tvrtka plaća oglašivaču samo broj prikazivanja svog oglasa, a ne neku fiksnu cijenu. *Botnet* mreže mogu biti iskorištene za prikaz i posjet takvih oglasa, koštajući tvrtku koja se oglašava na taj način.

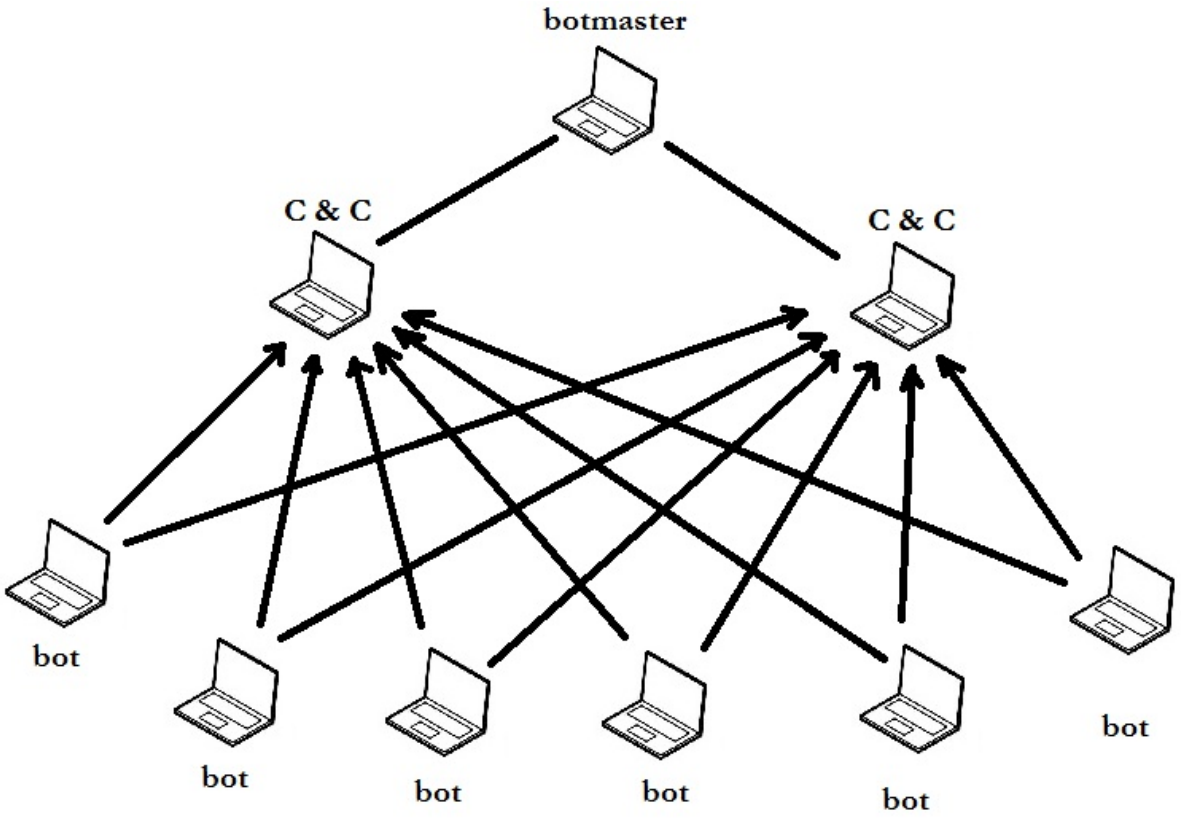
3.3. Arhitektura *Botnet* mreže

Od samih začetaka, odnosno genijalne ideje da se za komunikaciju i upravljanje drugim računalima iskoristi protokol za razgovor putem Interneta, napadači se prvenstveno oslanjaju na tu vrstu komunikacije u *Botnet* mreži. Protokol IRC razvijen je za komunikaciju korisnika na forumima, masovnim porukama ili privatnim porukama, u stvarnom vremenu odnosno bez kašnjenja, stoga je posebno pogodan za izdavanje naredbi većem broju računala u istom trenutku. Napadač koji želi preuzeti nadzor nad drugim računalima postavi jedno računalo kao IRC poslužitelja te ga koristi za komunikaciju s drugim zaraženim odnosno ranjivim računalima. Preko tog poslužitelja napadač ustvari upravlja *Botnet* mrežom, kako je prikazano na slici 3.1. [11].



sl. 3.1. Arhitektura *Botnet* mreže [11]

U današnje vrijeme upravitelji *Botnet* mreža, poznati i pod engleskim nazivima *botherder* ili *botmaster*, još uvijek primarno koriste neku vrstu protokola IM (*instant messaging*) za upravljanje *Botnet* mrežom, odnosno protokola za komunikaciju u stvarnom vremenu. Kako se ti protokoli koriste na većini socijalnih mreža, onemogućavanje tih protokola za zaštitu računala nije primjenjivo u većini slučajeva. Također se koristi i „klasičan“ komunikacijski protokol HTTP (*Hypertext Transfer Protocol*) preko kojeg se odvija većina www (*world wide web*) prometa na Internetu [10]. Pritom sama arhitektura *Botnet* mreže nije bitno promijenjena, no ono po čemu se današnje *Botnet* mreže razlikuju od onih prije nekoliko godina jest redundantnost. Naime, botmasteri koriste dva, tri ili više poslužitelja za komunikaciju s cijelom mrežom zaraženih računala. Arhitektura *Botnet* mreže koja koristi dva komunikacijska poslužitelja prikazana je na slici 3.2, a lako se može zamisliti mreža s više od dva takva poslužitelja.



sl. 3.2. Arhitektura *Botnet* mreže s dva upravljačka poslužitelja [10]

Ti poslužitelji poznati su još pod kraticom C&C koja dolazi od engleskih riječi za naredbe i kontrolu (*command* and *control*). Više C&C poslužitelja koristi se za slučaj da neki budu otkriveni, odnosno iz bilo kojeg drugog razloga nedostupni. U tom bi slučaju, bez dodatnih poslužitelja, *Botnet* mreža i dalje postojala, međutim nitko ne bi imao nadzor nad njom te ne bi mogao njome upravljati zbog nemogućnosti podjele zadataka zaraženim računalima [12].

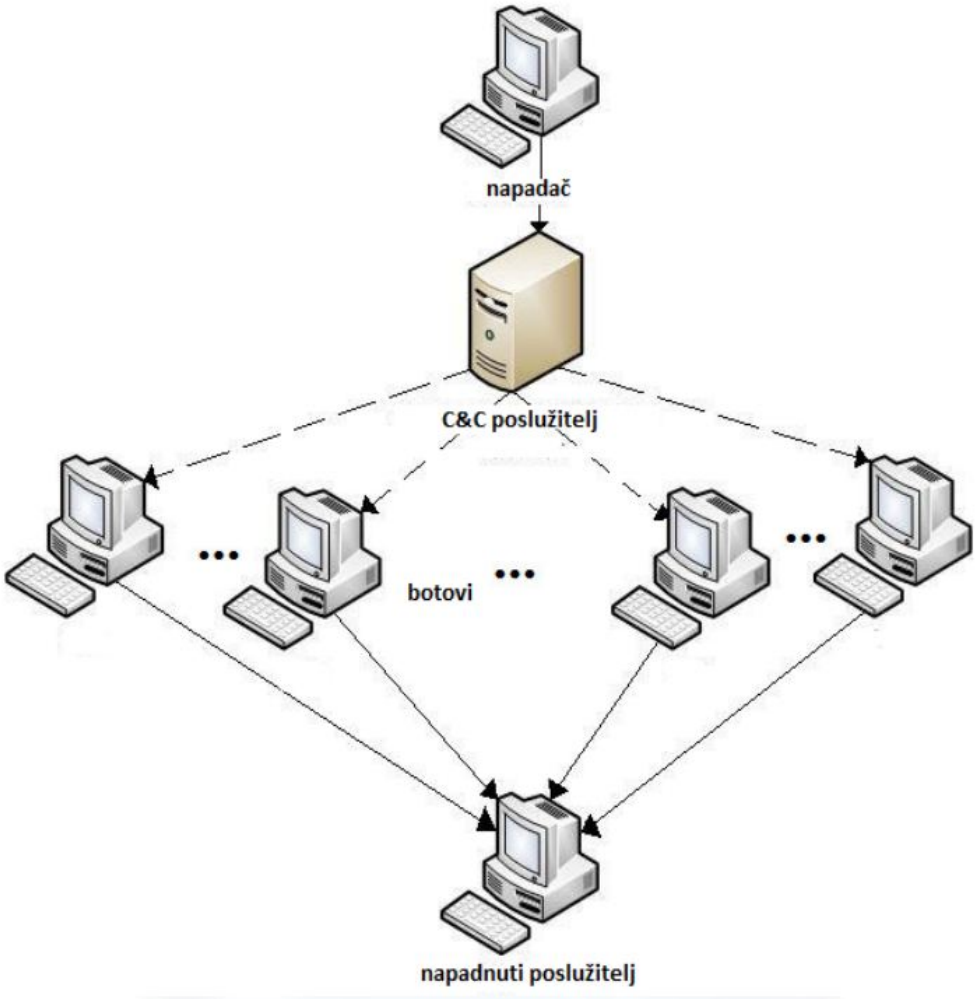
3.4. Primjer *Botnet* napada

Važno je napomenuti da se upravitelji *Botnet* mreža najčešće ne bave drugim vrstama kriminala, već samo iznajmljuju računalne resurse zaraženih računala drugim kriminalnim organizacijama. Na taj način izbjegavaju kriminalnu odgovornost za ilegalne radnje za koje su *Botnet* mreže iskorištene.

Nakon postavljanja poslužitelja za upravljanje *Botnet* mrežom, napadač kreće u „regrutaciju“ računala za *Botnet* mrežu. Računala korisnika koja postanu dio *Botnet* mreže najčešće

ne sadrže posljednje sigurnosne zakrpe za operacijski sustav ili odgovarajući program za zaštitu računala. Postoje dva načina na koja se zlonamjerni program, odnosno klijent za *Botnet* mrežu (*bot*) učitava na takvo računalo. Prvi od njih zahtijeva korisničku interakciju, odnosno napad računalnim virusom. Drugi je način napad računalnim crvom koji ne zahtijeva nikakvu korisničku interakciju, a sposoban je sam dalje se širiti. Napadač uspijeva proširiti zlonamjerni program (virus ili crv) koji će instalirati *bot* klijent na zaražena računala. Nakon uspješne instalacije *bot* klijenta, zaražena se računala uspješno spajaju na upravljačkog poslužitelja te čekaju daljnje naredbe upravitelja *Botnet* mreže [12].

Nakon uspješnog uspostavljanja *Botnet* mreže, tek sada slijede kriminalne radnje za koje je ona uspostavljena. Osim već opisane krađe podataka od vlasnika *Botnet* mreže zaraženih računala, te radnje najčešće uključuju iskorištavanje računalne moći mreže zaraženih računala, botova. Upravitelj *Botnet* mreže najčešće iznajmljuje računalnu snagu zaraženih računala ili prodaje ukradene financijske podatke korisnika drugim kriminalnim organizacijama sa specifičnim ciljevima. U ovom primjeru, nakon uspješnog uspostavljanja *Botnet* mreže, upravitelj mreže će iznajmiti svoje usluge trećoj strani, zainteresiranoj za reklamiranje svog lažnog i nepostojećeg proizvoda neželjenom elektroničkom poštom. Upravitelj *Botnet* mreže prima narudžbu za posao koji zahtijeva upravljanje *Botnet* mrežom. Nakon što naručitelj dostavi upravitelju mreže sadržaj poruke koju želi poslati u velikim količinama, upravitelj *Botnet* mreže šalje neželjenu elektroničku poštu sa svih računala u svojoj *Botnet* mreži [10].



sl. 3.3. Izvođenje DdoS napada pomoću *Botnet* mreže [7]

Nakon što je upravitelj *Botnet* mreže uspostavio nadzor nad većim brojem računala, on njihovu računalnu snagu može iznajmiti i za izvođenje distribuiranog napada uskraćivanjem usluga. Sve ovisi o željama i potrebama kriminalne organizacije koja unajmljuje njegove usluge [7].

3.5. Zaštita od *Botnet* napada

Što se tiče zaštite od *Botnet* mreža, razlikuju se aktivna i pasivna zaštita. Pasivna zaštita podrazumijeva zaštitu poslužitelja i osobnih računala, dok aktivna podrazumijeva korištenje *Honeypot* sustava. Zbog geografske raspršenosti računala u *Botnet* mreži, velike količine zaraženih računala i činjenice da zaražena računala najčešće imaju promjenjivu, odnosno dinamičku IP (*Internet protocol*) adresu, zaštita poslužitelja filtriranjem IP adresa nije primjenjiva.

Ograničen uspjeh može biti postignut analizom specifičnosti operacijskih sustava računala koje izvode DDoS napad te njihovim daljnjim onemogućavanjem pristupa poslužitelju (*Passive OS fingerprinting*). Skuplja je metoda zaštite korištenje specijaliziranog NIDS (*Network based intrusion detection system*) sklopovlja. Većina *Botnet* mreža, odnosno *Botnet* klijenata prima naredbe s upravljačkog poslužitelja koji se nalazi iza stalne adrese. To najčešće nije statička IP adresa, već je poslužitelj registriran na nekoj od besplatnih usluga koje omogućuju tu funkcionalnost, kao što su *DynDns.org*, *NoIP.com* i *Afraid.org*. Te se adrese mogu lako iščitati iz *bot* klijenta na zaraženom računalu te se mogu poduzeti odgovarajuće mjere da se taj *Botnet* poslužitelj onemogući u daljnjoj komunikaciji sa zaraženim računalima [10].

Iako odgovarajuće institucije onemoguće *Botnet* upravljački poslužitelj u daljnjoj komunikaciji sa zaraženim računalima u *Botnet* mreži, to ne znači da ta računala više nisu zaražena. Lako je moguće da *Botnet* upravitelj na neki način uspije osviježiti *bot* klijente s novom adresom *Botnet* upravljačkog poslužitelja ili da neki drugi napadač preuzme nadzor nad tim zaraženim računalima, ubacivši u bot klijente adresu svog *Botnet* upravljačkog poslužitelja.

Aktivna zaštita od napada *Botnet* mreže uključuje postavljanje posebnog *Honeypot* sustava za praćenje i proučavanje aktivnosti napadača. Praćenje *Botnet*a sastoji se od dva koraka, a to su sakupljanje informacija o postojećim *Botnet* mrežama (analizom već prikupljenog malicioznog koda) i priključenje *Honeypot* sustava na Internet radi daljnjeg prikupljanja informacija [9].

4. *HONEYPOT* SUSTAVI

Honeypot sustave teško je definirati iz razloga što je to tehnologija koja je u stalnom procesu promjene i koristi se u različitim aspektima sigurnosti kao što su prevencija, detekcija i sakupljanje informacija. Jedinstveni su po tome što ne rješavaju specifične sigurnosne probleme već su tehnologija općenite namjene. *Honeypot* je visoko prilagodljiv alat za primjene u područjima mrežne forenzike i detekcije upada u sustav. Zbog navedenih razloga, na Internetu je dostupan velik broj definicija sustava, no valja izdvojiti jednu koja najbolje opisuje sustav:

***«Honeypot* sustav je sigurnosni resurs čija vrijednost leži u mogućnosti da bude ispitan, napadnut ili kompromitiran» [13].**

Honeypot sustavi pomno su promatrani mrežni mamci koji imaju sljedeću svrhu [14]:

1. ometanje napadača u traženju vrijednih resursa na mreži,
2. davanje ranog upozorenja o novim napadima i vrstama napada te
3. omogućavaju detaljnu analizu tijekom i nakon napada na *Honeypot* sustave.

Svi *Honeypot* sustavi rade na istom principu: nitko ih ne bi trebao koristiti ili biti s njima u interakciji, dakle sve transakcije ili interakcije su po definiciji neovlaštene.

4.1. Povijest *Honeypot* sustava

Honeypot sustavi imaju jedinstvenu povijest. Iako su koncepti poznati već nekoliko desetaka godina, tek su nedavno razvijene komercijalne verzije softvera ili su pak objavljeni radovi. Prvi dokumentirani javni radovi o *Honeypot* sustavima su „The Cuckoo's Egg“ autora Clifforda Stolla i „An Evening With Berferd“ autora Billa Cheswicka napisani između 1990. i 1991. godine. Godine 1997. pojavljuje se jedan od prvih *honeypot* sustava, a to je *Deception Toolkit* autora Freda Cohena. Prva komercijalna verzija *honeypot* sustava je *CyberCop Sting* koja je razvijena 1998. godine. Iste godine pojavljuje se i besplatna inačica temeljena na *Windows* platformi nazvana *Back Officer Friendly*. Godinu nakon, u sklopu *Honeypot* projekta, krenula je serija publiciranih dokumenata pod imenom „Upoznaj svojeg neprijatelja“ koja je pridonijela širenju svjesnosti i potvrdila vrijednost *honeypot* tehnologije. Od 2000. nadalje sve više i više organizacija prihvaća tehnologiju i koristi je u detekciji napada i istraživanju novih prijetnji [14].

4.2. Vrste *Honeypot* sustava

Vrsta *Honeypot* sustava odabire se ovisno o cilju koji se želi postići. Ukoliko se želi naučiti što je više moguće o alatima i taktikama napada, koristit će se vrlo složen *Honeypot* sustav koji je pogonjen kompletnim operacijskim sustavom.

Zanima li nas detekcija neovlaštenog pristupa sustavu, dovoljno će biti koristiti vrlo jednostavan *Honeypot* sustav koji može emulirati određeni raspon servisa. Ukoliko netko pristupi tom sustavu, to vrlo vjerojatno znači neovlašteni pristup. Ako je pak cilj uhvatiti najnovijeg računalnog crva, tada je potreban posebno prilagođen vrlo inteligentan *Honeypot* sustav koji će biti u interakciji s crvom i istovremeno analizirati njegovu aktivnost [14].

4.2.1. Nisko interaktivni *Honeypot* sustavi

U osnovi se sve usluge (servisi) i operacijski sustav *Honeypot* sustava emuliraju, odnosno ne postoji pravi operativni sustav već se oponaša rad nekog stvarnog sustava. Interakcija napadača s pravim sustavom predstavlja neprihvatljiv rizik za organizaciju. Napadač bi mogao iskoristiti kompromitirani *Honeypot* sustav kako bi proširio napad na druga računala. Takav je rizik smanjen korištenjem nisko interaktivnih *Honeypot* sustava i svaki je maliciozni napad na njega ograničen na točno određene oponašane usluge. Na primjer, u slučaju SSH servisa (*Secure Shell* – kriptirani pristup udaljenom mrežnom uređaju) oponašan je port 22 koji uključuje SSH *login* i još neke dodatne naredbe te ukoliko iskusni napadač pokuša kompromitirati takav sustav, vrlo će brzo otkriti ograničenja takve usluge i tada će mu to dati do znanja da se radi o *Honeypot* sustavu [15].

Primarna vrijednost nisko interaktivnih *Honeypot* sustava njihova je sposobnost da odrede neovlašteno skeniranje (portova) mreže i pokušaj spajanja na istu. *Honeypot* sustav bilježi sve podatke vezano za svaki pokušaj skeniranja mreže. Osim toga, bilježi i sve podatke vezane za interakciju napadača s oponašanim uslugama kao što su npr. izvorišna i odredišna IP adresa, izvorišni i odredišni broj portova i još mnoge druge parametre. Prikupljanjem takvih podataka stvara se okvir za mogući odgovor *Honeypot* sustava prema napadaču. Na primjer, napadač pošalje zahtjev prema simuliranom FTP poslužitelju kako bi uspostavio vezu s njim te tada taj poslužitelj odgovara na zahtjev napadača. U nisko interaktivnim sustavima, odgovori na zahtjev napadača ograničeni su na određeni broj poznatih ponašanja. Postoje točno određeni koraci prema kojima *Honeypot* sustavi reagiraju prema specifičnim tipovima napada. Ukoliko *Honeypot* sustav nema informaciju o vrsti napada, tada sustav ne ulazi u interakciju s napadačem.

Što je veći stupanj interakcije napadača s *Honeypot* sustavom, to je veća količina prikupljenih podataka o samom napadaču. Mogućnost učenja o napadima na sustav povećava se ukoliko se broj naredbi, koje napadač može iskoristiti pristupanjem simuliranih usluga, poveća. Na

primjer, ukoliko napadač unese određene naredbe i dobije nepredviđene rezultate, odmah će biti svjestan da je u interakciji s *Honeypot* sustavom. U tom trenutku bilo kakva interakcija s napadačem završava i moguće je da napadač iskoristi kompromitirani *Honeypot* sustav za napad na ostala računala u mreži.

Instalacija, konfiguracija i implementacija ove je vrste *Honeypot* sustava jednostavna. U osnovi, to je računalni program instaliran na računalu korisnika i podešen da simulira željene servise. Mogućnost pogreške u instalaciji i implementaciji takvog softvera relativno je niska.

4.2.2. Srednje interaktivni *Honeypot* sustavi

S obzirom na razinu interakcije s napadačem, ova se vrsta sustava nalazi između nisko interaktivnih i visoko interaktivnih *Honeypot* sustava. Srednje interaktivni *Honeypot* sustavi, za razliku od nisko interaktivnih sustava, prema poznatim vrstama napada sudjeluju više u interakciji s napadačem. Na primjer, u uniksoidnom operativnom sustavu kao što je *Linux*, sistem administrator može implementirati virtualni operacijski sustav i koristiti *Chroot Jail* naredbu. *Chroot Jail* svojstvo je *Unix* operacijskih sustava koje se koristi kako bi ograničili određenog korisnika da pristupi određenim datotekama. To svojstvo ograničavanja određenih datoteka, koje zahtijevaju veće ovlasti pristupa, osigurava veću razinu sigurnosti [14].

Napadač će pokušati zadobiti *Chroot* privilegije kako bi imao slobodan pristup sustavu. Svaka se njegova interakcija sa sustavom bilježi i nadzirana je od strane administratora stvarnog (fizičkog) operativnog sustava. Velik se broj kritično važnih informacija o ponašanju napadača bilježi, kao što su naredbe koje napadač unosi. Međutim, ova situacija nosi sa sobom određenu dozu opasnosti u otkrivanju stvarnog operacijskog sustava i kompromitiranju istog od strane napadača.

Instalacija i implementacija srednje interaktivnih *Honeypot* sustava relativno je teška i zahtjevna. Za razliku od nisko interaktivnih *Honeypot* sustava, puno je teže postaviti ograničenja i visoku razinu sigurnosti sustava kako bi se spriječili napadi na druge sustave u mreži. Međutim, srednje interaktivni *Honeypot* sustavi uspijevaju prikupiti mnogo više informacija o napadaču nego što to mogu nisko interaktivni, a mogu sadržavati maliciozne programske kodove ili neke druge zanimljive i korisne informacije [16].

4.2.3. Visoko interaktivni *Honeypot* sustavi

Visoko interaktivni *Honeypot* sustavi, za razliku od nisko i srednje interaktivnih, daju napadaču mogućnost interakcije s pravim operacijskim sustavom i nemaju nikakva ograničenja u

interakciji koja prethodno navedeni *Honeypot* sustavi imaju. Svrha podizanja razine interakcije s napadačem povećanje je prikupljenih informacija o napadu te se iz tog razloga koristi pravi operacijski sustav za razliku od samo simuliranih usluga. Mogućnost učenja o novim slabostima sustava i vrstama napada na sustav mnogo su veća od bilo kojih drugih *Honeypot* sustava [14].

Projektiranje, konfiguracija i održavanje visoko interaktivnih *Honeypot* sustava predstavlja izuzetno težak zadatak i zahtijeva mnogo truda i vremena. Nadalje, ukoliko napadač kompromitira takav sustav i zadobije potpunu kontrolu nad njim, može ga iskoristiti za napade na druga računala. Iz tog razloga potrebno je poduzeti ozbiljne mjere predostrožnosti u eliminaciji takvog rizika u ovoj vrsti *Honeypot* sustava.

Implementacija sustava za detekciju upada (IDS – *Intrusion Detection System*) u praćenju malicioznih aktivnosti može biti od pomoći u ograničavanju rizika kompromitacije sustava. Pristup visoko interaktivnim *Honeypot* sustavima trebao bi biti ograničen uređajem za kontrolu pristupa. Na primjer, ukoliko se visoko interaktivni *Honeypot* sustav nalazi iza vatrozida i napadač zadobije puno kontrolu nad *Honeypot*om, vrlo će teško moći kompromitirati ostala računala u mreži. Vatrozid blokira pristup i ograničava aktivnosti napadača u određenom dijelu sustava. Međutim, ovakav bi dizajn mogao učiniti napadača sumnjičavim i tada bi mogao shvatiti da je nadgledan. Osim toga, uvođenjem vatrozida i postavljanjem ograničenja na aktivnost napadača u točno određenom dijelu mreže zahtijeva više truda i vremena u održavanju [16].

Primjer visoko interaktivnog *Honeypot* sustava je *honeynet*. *Honeynet* se sastoji od mreže računala dizajniranih da privuku i prate aktivnosti napadača. Kada napadač napadne žrtvu, sve će aktivnosti vezane za pravi operacijski sustav biti zabilježene. Ovakav tip arhitekture omogućuje više kontrole administratoru i smanjuje sigurnosni rizik upada u sustav. Također se koristi *honeywall* pristupnik (*gateway*) kako bi se omogućio dolazni promet prema *Honeypot* sustavu i kontrolirao odlazni promet prema Internetu.

4.2.4. Klijentski i poslužiteljski *Honeypot* sustavi

Većina je *Honeypot* sustava instalirana, podešena i fizički smještena u mreži na način da se ponašaju poput poslužitelja. Zauzimaju određeno mjesto u mreži i čekaju napadača. Izlažu određene ranjivosti i na taj način mame napadače da iniciraju maliciozne radnje. Napadač pokušava iskoristiti ranjivosti sustava i kompromitirati sustav. Ovu vrstu *Honeypot* sustava nazivamo serverski ili poslužiteljski *Honeypot* sustav.

Druga se vrsta *Honeypot* sustava koja radi na drugačijem prinicipu od poslužiteljskog zove klijentski *Honeypot* sustav. Prvi takav sustav razvio je Lance Spitzer 2004. godine u svrhu prikupljanja informacija o malicioznim poslužiteljima koji pokušavaju iskoristiti ranjivosti računala krajnjih korisnika i preuzeti kontrolu nad njima. Klijentski *Honeypot* sustavi simuliraju aktivnosti klijenta u interakciji između klijenta i poslužitelja [17].

Kada se klijent spaja na maliciozni poslužitelj, poslužitelj započinje napad i iskorištava ranjivosti klijentskog operacijskog sustava kako bi zadobio kontrolu nad istim. Ova vrsta napada, na strani klijentskog računala, postala je vrlo učestala u zadnje vrijeme. Na primjer, *Adobe Flash* platforma za pregledavanje multimedijalnih sadržaja ima određene ranjivosti koje napadači iskorištavaju ubacivanjem malicioznog koda koji se izvršava na klijentskim računalima tako što korisnik posjeti neku malicioznu stranicu ili "skine" određenu multimedijalnu SWF datoteku [18].

Klijentski je *Honeypot* dizajniran za traženje malicioznih poslužitelja i prikupljanje informacija o napadima na klijente. Klijent šalje zahtjev poslužitelju nakon čega čeka odgovor. Odgovor je detaljno analiziran te otkriva namjere i ponašanje napadača. Prilikom projektiranja klijentskog *Honeypot* sustava, važno je uzeti u obzir kompatibilnost s poslužiteljskim protokolima. Da bi se postigla povezanost i interakcija s poslužiteljem, protokoli koje koristi klijent trebaju biti usklađeni s poslužiteljskim [19].

Klijentski *Honeypot* sustavi otkrivaju bilo koje poznate i nepoznate napade na sustav u realnom vremenu koji su usmjereni na aplikacije klijenta. Klijentski *Honeypot* sustavi imaju generalno sličan način funkcioniranja. Indeksiraju mrežni segment u potrazi za malicioznim poslužiteljima i traže točno određene web stranice i pokreću testiranje kako bi ustanovili postoji li na stranicama maliciozni sadržaj. Brzina indeksiranja ovisi o mrežnoj propusnosti, ali i o samom algoritmu indeksiranja. Povećanjem tih parametara povećava se efikasnost klijentskog *Honeypot* sustava. Indeksiranje na istoj lokaciji više puta trebalo bi izbjegavati zbog mogućnosti detekcije.

4.3. Strategija implementacije *Honeypot* sustava

Postoji nekoliko načina implementacije *Honeypot* sustava. O ciljevima koje želimo postići ovisi tip i količina *Honeypot*a koje ćemo koristiti. Bez dovoljne količine znanja o vrsti *Honeypot* sustava i informacijama koje tražimo, bilo bi bezvrijedno i neučinkovito postaviti *Honeypot* bilo gdje u mreži. Jedan je od osnovnih koraka imati temeljito znanje o vrsti *Honeypot*a prije same implementacije sustava.

Osim toga, ljudski i tehnički resursi strahovito utječu na implementaciju *Honeypot* sustava. Na primjer, visoko interaktivni *Honeypot* sustavi zahtijevaju napredna sigurnosna i računalna znanja i vještine kako bi ih implementirali. U nekim slučajevima, potrebno je implementirati grupu *Honeypot* uređaja u jedan honeynet, za razliku od samo jednog *Honeypot* sustava, u svrhu povećanja prikupljanja informacija [20]. U tom slučaju, potreban nam je i *honeywall* pristupnik.

Osim toga, dok je god glavna svrha *Honeypot* sustava zavaravanje napadača i prikupljanje informacija o istome, vrlo je važno zadržati napadača nesvjesnog o postojanju *Honeypot* sustava. Ukoliko napadač postane svjestan da je u interakciji s *Honeypot* sustavom, može namjerno unositi krive podatke u sustav kako bi zavarao promatrače koji su taj *Honeypot* sustav postavili [14].

4.3.1. Klasična implementacija orijentirana prema Internetu

Ova vrsta implementacije uključuje izlaganje *Honeypot* sustava javnoj mreži odnosno Internetu. Sustav prati ponašanje napadača i prikuplja informacije o crvima i ostalim malicioznim aktivnostima. Najčešće se nalazi u DMZ-u (Demilitariziranoj zoni), tako da je dostupan na Internetu [20]. Demilitarizirana je zona fizička ili logička podmreža koja sadrži i otkriva određene servise odnosno usluge organizacije nekoj vanjskoj mreži smanjenog povjerenja kao što je Internet [21].

4.3.2. Unutarnja implementacija

Ova vrsta implementacije služi za dobivanje informacija o prijetnjama i zarazama unutar mreže neke organizacije. Na primjer, u slučaju da neki od zaposlenika odluči kompromitirati neko od računala unutar lokalne mreže. U interakciji s napadačem *Honeypot* sustav ga prevari i zbog toga identitet, ponašanje i taktike napadača postaju poznati. Postavlja se na razna mjesta unutar lokalne mreže te ne bi trebao sadržavati vrijedne podatke [20].

4.4. Prednosti i mane korištenja *Honeypot* sustava

Honeypot sustavi vrlo su moćni alati koji nam daju mogućnost stjecanja velike količine informacija o napadima na mrežne računalne sustave. Mogu prikupiti razne informacije o napadaču, na primjer gdje se nalazi, kako se ponaša i koje alate koristi. Također imaju mogućnost pohraniti kopiju malicioznih softvera poput crva koje napadači koriste kako bi kompromitirali sustav. Te se informacije zatim koriste za pronalaženje sigurnosnih rješenja pri mrežnim napadima i sprečavanju istih u budućnosti. *Honeypot* sustavi često prikupljaju samo maliciozne datoteke, no ne i sav promet. Na ovaj način, potrebno je manje podatkovnog prostora te je lakše analizirati informacije i pronaći rješenja.

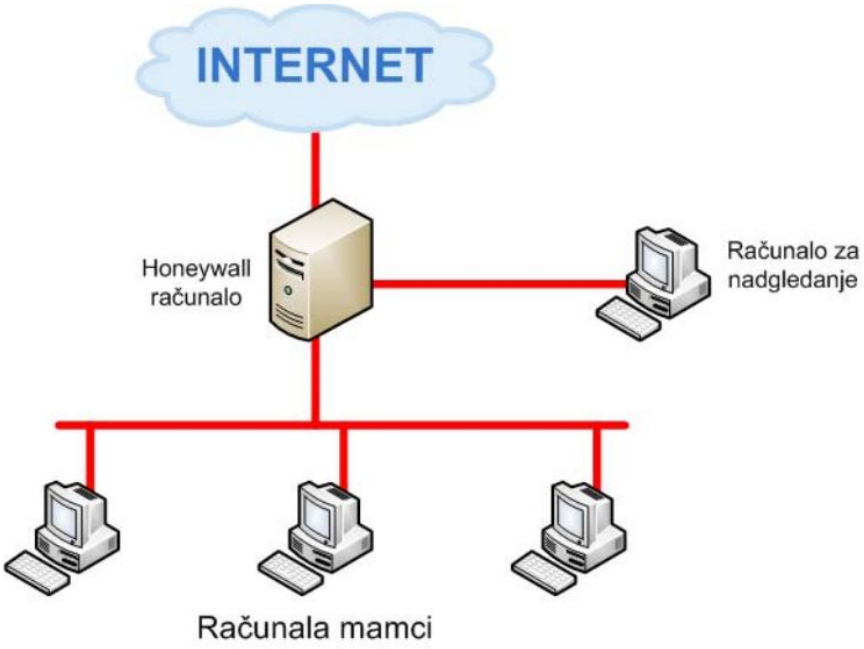
Većinu je *Honeypot* sustava jednostavno instalirati, podešavati i održavati. Za održavanje *Honeypot* sustava nisu potrebne baze podataka s potpisima (*signatures*) jer rade na prinicpu da sav dolazni promet smatraju sumnjivim. Potrebno je samo nekoliko koraka kako bi se *Honeypot* sustav implementirao i započeo proces prikupljanja podataka. Informacije dobivene iz tih podataka mogu predstavljati nove alate i tehnike koje napadači koriste. Dakle, postoji mogućnost da nastanu nova rješenja u vezi s mrežnom sigurnosti temeljena na prikupljenim podacima [22].

Za razliku od vatrozida i sustava za otkrivanje upada (IDS), *Honeypot* sustavi imaju dovoljno resursa za normalan rad. Na primjer, vatrozidi ponekad mogu zablokirati sve veze, a ne samo one bez prava pristupa, jer im je tablica zapunjena i ne prepoznaju nove veze. Ukoliko kroz sustav za otkrivanje upada prolazi velika količina prometa i podatkovni se spremnik senzora zapuni, tada sustav počinje odbacivati IP pakete. *Honeypot* sustavi nemaju takvu vrstu problema jer je za njihov rad potrebno manje resursa nego za druge sigurnosne alate [16].

Honeypot sustavi mogu prikupiti informacije samo ukoliko je napadač u aktivnoj i neposrednoj interakciji sa sustavom. Ukoliko se dogodio napad na nekom drugom sustavu u mreži koji može uzrokovati veliku štetu, *Honeypot* neće prikupiti podatke o napadaču. Štoviše, kamufliranje *Honeypot* sustava i ograničavanje napadača da postanu svjesni istog može biti teška zadaća. Veliki bi sigurnosni problem mogao nastati ukoliko napadač shvati da nije u interakciji s pravim sustavom već s *Honeypot*om. U tom slučaju, mogao bi iskoristiti kompromitirani *Honeypot* za napad na druga računala ili za lažiranje podataka o napadu.

5. *HONEYNET* SUSTAVI

Honeynet je *Honeypot* sustav visoke interakcije. Nastao je u okviru *The Honeynet Project* organizacije. Primarni je cilj te organizacije podizanje razine svijesti svih korisnika Interneta o postojanju prijetnji i ranjivosti. Sekundarni je cilj pružanje dodatnih informacija o motivima napadača, njihovoj međusobnoj komunikaciji, koracima nakon uspješnog napada, organizaciji napadača itd. *Honeynet* je trenutno vrhunac razvoja *Honeypot* tehnologije i obuhvaća sve prije navedene vrijednosti za takve sustave [18]. Kao *Honeypot* sustav visoke interakcije, *honeynet* nudi napadaču stvarni operacijski sustav, stvarne programe na njemu i stvarne usluge. *Honeynet* se razlikuje od drugih sustava (za privlačenje i detekciju napadača) i po tome što omogućava korisniku izgradnju čitave mreže (*net*) *Honeypot* sustava. *Honeynet* je čitava arhitektura i ne radi se samo o jednom proizvodu koji je dovoljno instalirati na računalo i pokrenuti.



sl. 5.1. Prikaz *honeynet* sustava [18]

Arhitektura se sastoji od strogo kontrolirane mreže računala u kojoj se može nadzirati i upravljati svim aktivnostima. U toj se mreži nalaze objekti kojima je jedina svrha da budu ispitani, napadnuti i na kraju kompromitirani.

5.1. Arhitektura *honeynet* sustava

Kako bi neki sustav bio *honeynet* mora imati četiri osnovna elementa: nadzor podataka (*data control*), prikupljanje podataka (*data capture*), analizu podataka (*data analysis*) i upravljanje podacima (*data collection*). Svaki se *Honeypot* sustav koji posjeduje navedene elemente smatra *honeynet* sustavom [14].

5.1.1. Nadzor podataka

Nadzor podataka je element koji omogućava smanjenje rizika za organizaciju. Svrha je nadzora nad podacima ograničiti aktivnost napadača prema i od samog *Honeypot* sustava. Cilj je napadaču onemogućiti daljnje aktivnosti usmjerene prema vanjskim sustavima, onima koji se ne nalaze unutar *honeynet* sustava, kako bi se smanjio ili u potpunosti uklonio rizik za iste.

Postoji nekoliko zahtjeva za implementaciju *honeynet* sustava [18]:

1. automatsko i ručno nadziranje podataka što znači da se napadača mora moći nadzirati automatski, ali i ručno, ukoliko se ukaže potreba za time,

2. najmanje dva sloja nadzora nad podacima kako bi se smanjila vjerojatnost pogreške,

3. mogućnost nadzora nad svim dolaznim i odlaznim vezama,

4. mogućnost kontrole nad svim neovlaštenim aktivnostima (misleći na aktivnosti koje definira administrator *honeynet*-a) te je ovdje uključena i kontrola napadača kako ne bi naštetio vanjskim sustavima,

5. element nadzora nad podacima mora biti podesiv od strane administratora u svako doba,

6. način upravljanja *honeynet*-om mora biti što bolje zaštićen od napadača (najbolje bi bilo da napadač nema načina za otkrivanje načina upravljanja *honeynet* sustavom),

7. moraju postojati barem dva načina uzbunjivanja ukoliko napadač kompromitira neki od *Honeypot* sustava i

8. daljinsko upravljanje procesom nadzora nad podacima (administrator mora moći upravljati ovim procesom s bilo koje lokacije).

5.1.2. Prikupljanje podataka

Prikupljanje podataka svodi se na prikupljanje informacija o svim aktivnostima napadača bez njegova znanja. Komponenta za prikupljanje podataka mora zadovoljiti određene zahtjeve. Ponajprije, podaci o aktivnosti napadača ni u kom slučaju ne smiju biti pohranjeni lokalno na *Honeypot* sustavu. Ukoliko bi to bio slučaj, napadač bi mogao kompromitirati podatke što je nedopustivo. Zatim, podaci ni u kom slučaju ne smiju biti „zagađeni“ od strane onih koji nisu napadači. Primjer bi bio ispitivanje određenih alata na *Honeypot* sustavu od strane legalnih korisnika (administratora).

Nadalje, aktivnost na mreži, aktivnost sustava, aktivnost pripadajućih programa te aktivnost korisnika *Honeypot* sustava mora biti arhivirana na razdoblje od barem jedne godine. Također, administrator mora biti u mogućnosti nadgledati prikupljene podatke u realnom vremenu s udaljene lokacije, a standardizirani dnevnik svakog *Honeypot* sustava mora postojati. Isto tako, administrator sustava mora voditi standardizirani dnevnik za svaki *Honeypot* sustav koji je kompromitiran. Ono što senzori unutar *honeynet* sustava zabilježe mora biti u GMT vremenskoj zoni. Pojedini *Honeypot* sustavi mogu koristiti lokalnu vremensku zonu, no svi podaci koji se šalju na analizu moraju biti konvertirani u GMT vremensku zonu. Naposljetku, resursi koji se koriste za skladištenje podataka moraju biti propisno osigurani kako podaci ne bi bili kompromitirani [18].

5.1.3. Analiza podataka

Primarna je svrha *honeynet* sustava analiza podataka koje sustav prikupi u interakciji s napadačem. Iz tog razloga potreban je mehanizam pretvaranja prikupljenih podataka u korisne informacije. Ukoliko sustav analize nije implementiran, *honeynet* sustav je beskoristan. Svaka organizacija ima specifičnu potrebu pa iz tog razloga postoje različiti načini analize podataka.

5.1.4. Upravljanje podacima

Upravljanje je podacima zahtjev koji se jedino odnosi na organizacije koje imaju nekoliko *honeynet* sustava u svojem vlasništvu. Smisao ovog elementa u centraliziranom je prikupljanju podataka koje sakupi više sustava. Centraliziranim prikupljanjem podataka i njihovom kasnijom analizom organizacija može povećati vrijednost implementiranih *honeynet* sustava.

Kako bi ovaj element bio uspješno ostvaren, postavlja se nekoliko zahtjeva [14]:

1. Imenovanje i mapiranje *Honeypot* sustava mora postojati za svaki implementirani *honeynet* sustav. To uglavnom znači da mora postojati baza podataka s IP adresama i imenima svih *Honeypot* sustava koji se nalaze unutar implementiranih *honeynet* sustava.

2. Prijenos podataka od svakog pojedinog *honeynet* sustava prema centralnom poslužitelju mora biti siguran. Kriteriji povjerljivosti podataka, integriteta i neporecivosti moraju biti zadovoljeni.

3. Organizaciji se mora ostaviti mogućnost da prikrije osjetljive podatke, ukoliko ih smatra povjerljivima.

4. Svi udaljeni *honeynet* sustavi moraju biti sinkronizirani. To znači da svi moraju preko NTP (*Network Time Protocol*) protokola biti sinkronizirani s istim NTP poslužiteljem.

5.2. Prednosti korištenja *honeynet* sutava

Prednost ove tehnologije svakako je mogućnost otkrivanja više informacija o napadačima. Kako se radi o strogo kontroliranoj mreži računala, korisnik je u mogućnosti motriti svaki pokret napadača – svaki unos s tipkovnice nakon što je sustav kompromitiran, oblik komunikacije s drugim napadačima ili sustavima, korištene alate itd. Posebno je značajno što su napadači stvarni poput napadnutih sustava pa postoji mogućnost otkrivanja novih alata i tehnika koje napadači koriste.

Također postoji mogućnost analize stvarnih podataka s ciljem predviđanja mogućih napada. Kako *honeynet* sustav radi u stvarnom okruženju, njegovim je korištenjem moguće dobiti bolji uvid u trenutno stanje računalne sigurnosti na Internetu. Podaci dobiveni na ovaj način mogu poslužiti za predviđanje mogućeg napada i pomoći u njegovoj prevenciji. Novi alati i tehnike napadača jedino se na taj način mogu otkriti, s obzirom na to da su drugi alati (antivirusni program, sustav za otkrivanje napadača - IDS) nemoćni pred tim izazovom jer uglavnom koriste bazu „potpisa“ poznatih napada i zloćudnih programa [18].

Honeynet sustavi značajno pomažu pri odgovoru na sigurnosni incident. Istraživačke organizacije poput *The Honeynet Project* sve dobivene informacije javno objavljuju i na taj način omogućavaju zainteresiranim stranama da iz tuđih iskustava steknu novo znanje i na taj način brže reagiraju na incident ili ga u potpunosti izbjegnu. Koristeći znanje dobiveno na ovaj način, moguće je izgraditi nove alate i otkriti nove tehnike čijim bi se korištenjem smanjio rizik.

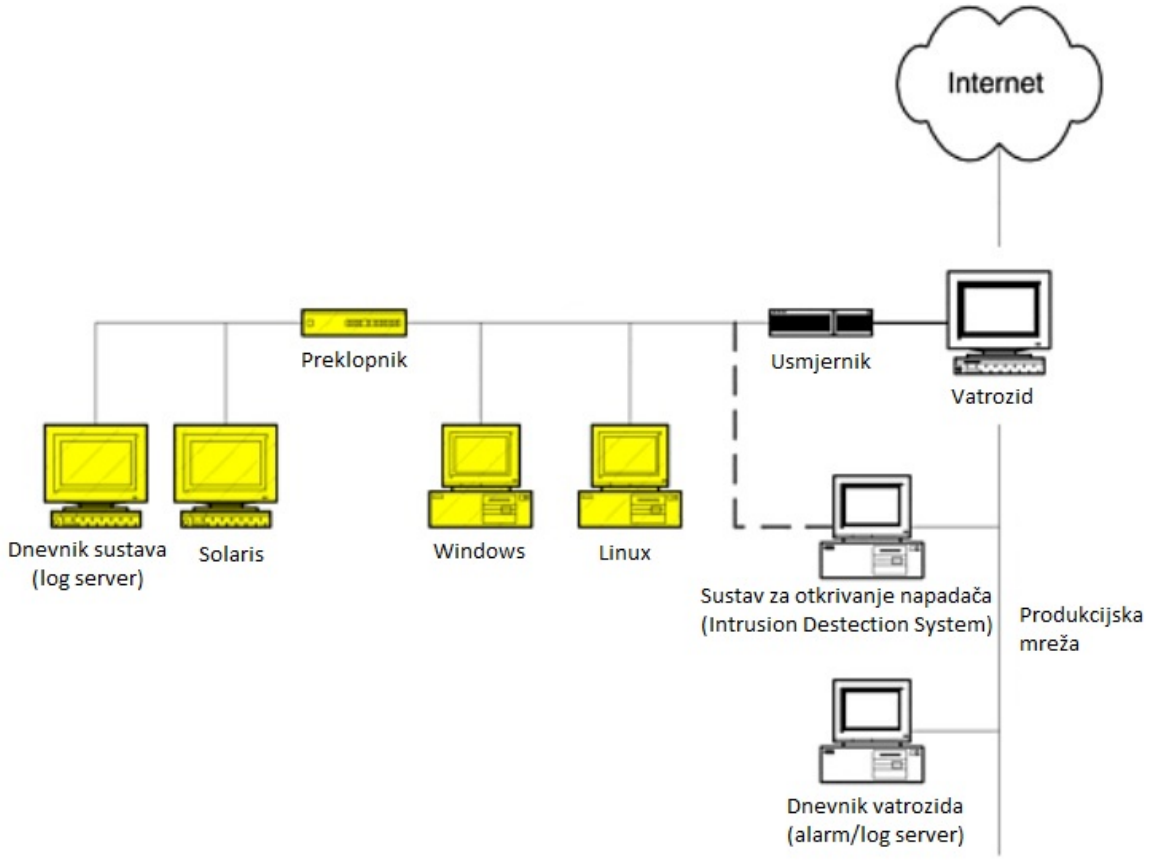
Bitna je prednost *honeynet* sustava korištenje tehnologije kao ispitnog poligona za nove tehnologije. Koliko god da je uloženo truda u novu tehnologiju, moguće je postojanje vrlo kritičnih propusta koji bi mogli ugroziti sigurnost onoga koji tu tehnologiju koristi [14]. Iz tog je razloga zgodno takvu tehnologiju pokrenuti u strogo kontroliranom okruženju kakvo pruža *honeynet* sustav

i vidjeti što će se dogoditi. Na taj je način moguće otkriti sigurnosne propuste unutar nove tehnologije bez stvarnog rizika.

5.3. Prva, druga i treća generacija *honeynet* sustava

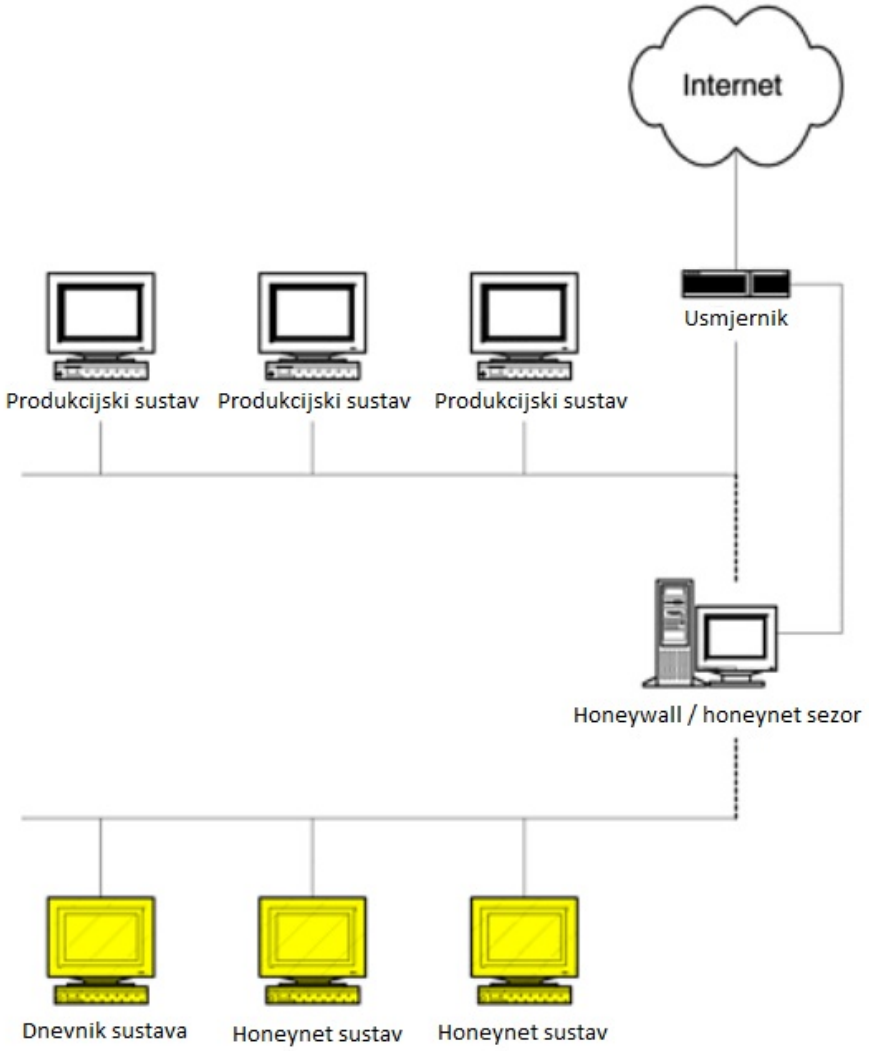
Honeynet sustav prvi je put predstavljen 1999. godine. Iako je postojalo već nekoliko rješenja s ovog područja, *honeynet* sustav donio je dvije potpuno nove stvari: zbog svoje prirode sustav je u mogućnosti priskrbiti mnogo zanimljivije podatke o napadačima i može uhvatiti nove tehnike napadača i njihove alate. *Honeynet* prve generacije bio je prvi pravi *Honeypot* sustav visoke interakcije [14].

Honeynet sustav prve generacije bio je ograničen u svojim mogućnostima kontrole nad napadačem. Zbog svoje implementacije uglavnom je služio kao alat za otkrivanje automatiziranih napada, kao i napada koji su djelo početnika. Napredniji je napadač mogao na jednostavan način otkriti pravu svrhu ovog sustava jer je na lagan način bilo moguće otkriti alate korištene za nadzor podataka. Tipičan primjer *honeynet*-a prve generacije prikazan je na slici 5.2.



sl. 5.2. *Honeynet* prve generacije [18]

Honeynet druge generacije pojavio se 2002. godine kao poboljšani nasljednik *honeynet* sustava prve generacije. Autori su nastojali olakšati samu izgradnju *honeynet*-a i otežati otkrivanje njegove prave svrhe napadačima. Uklanjanje određenih nedostataka iziskivalo je izmjenu same arhitekture. Izgled nove arhitekture prikazan je na slici 5.3 [14].



sl. 5.3. *Honeynet* druge generacije [18]

Honeynet treće generacije razvijen je od strane *Honeynet* projekta 2005. godine, a korišten sve do 2015. Arhitektura treće generacije *honeyneta,* za razliku od druge generacije, ne razlikuje se značajno već donosi poboljšanja u nadzoru samog sustava. Alati za udaljeni nadzor su implementirani te je također optimizirano prikupljanje i centralizacija informacija. Na primjer, *web*

sučelje za upravljanje *honeywallom* u stanju je povezati događaje, prikazati prikupljene informacije i promijeniti *honeynet* konfiguraciju s jednog mjesta [23].

Honeynet prve generacije vrlo je jednostavan sustav. Izolirana mreža *Honeypot* sustava nalazi se iza vatrozida koji onemogućava napad na sustave izvan *Honeynet* sustava. Vatrozid i usmjernik brinu se o nadzoru podataka i omogućavaju sve dolazne veze, ali ograničavaju odlazne, posebice one usmjerene prema produkcijskim sustavima. Vatrozid ograničava broj dozvoljenih odlaznih veza prema želji administratora. Što je veći broj dozvoljenih veza, veća je i količina informacija koju je moguće dobiti, no povećava se i rizik te je stoga potrebno pažljivo odvagnuti rizik i želju za prikupljanjem informacija.

Kako je jedan od zahtjeva pri nadzoru podataka dva sloja nadzora, ovdje je implementiran drugi sloj koristeći usmjernik koji se nalazi između prespojnika i vatrozida. Uloga je usmjernika dvostruka – skrivanje vatrozida od napadača i drugi kontrolni mehanizam u nadzoru podataka. Kada napadač kompromitira jedan od *Honeypot* sustava, on će na izlazu vidjeti usmjernik, a ne vatrozid koji ga kontrolira što će smanjiti njegovu sumnju u napadnuti sustav. Usmjernik također služi za sprječavanje napada na određene pristupe (zatvaranjem istih) i općenito kako bi umanjio opasnost od zloćudnih napada. Uzbunjivanje administratora obavlja vatrozid koji šalje administratoru poruku svaki put kada se nešto dogodi s *Honeypot* sustavima. Za prikupljanje podataka namijenjena su tri različita uređaja – dnevnik vatrozida, sustav za otkrivanje napadača (IDS) i dnevnik sustava (*log server*). Kako sav promet prema i od *Honeypot* sustava mora proći kroz vatrozid, on je idealan za prikupljanje podataka o sustavima [24].

Na slici 5.2. vidljivo je da sustav za otkrivanje napadača ima dva različita mrežna sučelja – jedno (puna crta) je spojeno na produkcijski dio mreže i preko njega se administrira sustav, dok je drugo (isprekidana crta) spojeno na *honeynet* sustav te ne posjeduje IP adresu, nego samo pasivno sluša sav promet u *honeynet* sustavu. Prva je funkcija ovog sustava bilježenje cjelokupnog prometa na *honeynet* sustavu. Obično se sva aktivnost bilježi na samom sustavu za otkrivanje napadača u obliku prikladnom za kasniju analizu. Druga je funkcija alarmiranje administratora, ukoliko se pojavi bilo kakav sumnjiv promet.

Honeynet sustav koristi bazu potpisa za otkrivanje sumnjivog prometa što ukazuje na veliki nedostatak – sustav nije u mogućnosti otkriti nove alate i tehnike napadača. Dnevnik sustava bilježi sve što se događa na sustavima za privlačenje napadača i to se čini lokalno, unutar *honeynet* sustava, ali i na udaljeni server. Bilježe se akcije napadača i sve što on napiše na tipkovnici sustava šalje se na udaljenog poslužitelja.

Honeynet sustav prve generacije imao je i određene nedostatke. Prvi je problem u nadzoru podataka – ukoliko je napadaču dozvoljen određeni broj odlaznih veza, on može lansirati određeni broj napada na vanjske sustave. Drugi je problem postojanje mogućnosti da napadač otkrije pravu prirodu *honeynet*-a te nakon toga lažira podatke kako bi administratora naveo na krivi put. Ovaj je problem postao posebno izražen kada je tehnologija *honeynet* sustava postala popularna. Treći je problem u načinu prikupljanja podataka, s obzirom na to da svi sustavi koji se nalaze u *honeynet*-

u prve generacije rade na mrežnom sloju pa ih je moguće prevariti korištenjem nekog oblika enkripcije. Svi su navedeni problemi otklonjeni u *honeynet* sustavu druge generacije.

Osnovna je razlika u tomu što se za nadzor podataka ne koristi vatrozid, već je dodan novi uređaj koji se naziva *honeywall*. Uloga je *honeywall*-a dvostruka – on ima funkciju sustava za otkrivanje napadača i vatrozida koji su postojali u *honeynet* sustavu prve generacije. Inače *honeywall* radi na drugom, podatkovnom, sloju OSI modela. Dakle, radi kao premosnik (*bridge*) između *honeynet*-a i ostatka svijeta. To donosi određene prednosti u odnosu na prijašnju implementaciju *honeynet* sustava - omogućava implementaciju *honeynet* sustava kao dijela produkcijske mreže koji je odvojen od nje samo *honeywall*-om, dok se s druge strane napadaču značajno otežava otkrivanje samog *honeywall*-a [24].

Kako je *honeywall* jedini ulaz i izlaz iz honeynet-a, na njemu se vrši nadzor podataka korištenjem dviju metoda – ograničavanjem odlaznih veza i korištenjem mrežnog sustava za prevenciju napadača (*Network Intrusion Prevention System*) koji ima ulogu sprječavanja poznatih napada. Nadzor podataka još je veći izazov kada se ima na umu da to treba obaviti na način da napadač ne bude svjestan da je nadziran.

Za ograničavanje odlaznih veza u *honeynet* sustav druge generacije koristi se alat pod imenom *IPtables*. *IPtables* je alat koji služi za upravljanje netfilter komponentom operacijskog sustava. On dolazi standardno uz sve modernije *Linux* jezgre. Kod *honeywall*-a, *IPtables* služi za ograničavanje prometa u četiri različite kategorije: TCP, UDP, ICMP i *OTHER*. Za svaki od navedenih protokola može se podesiti željeni broj veza u željenom vremenu. Naravno, broj odlaznih veza ovisi o tome koliki se rizik može preuzeti. Samo ograničavanje veza provodi se zbog raznih automatiziranih alata koje bi napadač mogao pokrenuti jednom kada kompromitira jedan ili više *Honeypot* sustava unutar *honeynet*-a. Kada napadač iscrpi sve raspoložive veze, *IPtables* zaustavlja svaki daljnji pokušaj stvaranja veze dok ne istekne vremenski limit broja veza. Zanimljiva je mogućnost *IPtables* alata blokiranje veza za svaki pojedini protokol. Primjerice, ako se iscrpi broj TCP veza, moguće je raditi nove veze drugim protokolima [15].

Druga komponenta namijenjena nadzoru podataka je mrežni sustav za sprječavanje neovlaštenih aktivnosti – NIPS. Namjena je ovog sustava otkrivanje poznatih napada i, ukoliko je na taj način postavljen sustav, blokiranje istih. Sustav za otkrivanje napadača pregledava svaki odlazni paket i ukoliko se radi o poznatom napadu moguće ga je blokirati. Na ovaj se način značajno smanjuje mogućnost uspješnog napada na sustave koji su izvan *honeynet* sustava. No, postoji i mana ovog sustava – on je u mogućnosti otkriti samo poznate napade. Mrežni sustav za sprječavanje neovlaštenih aktivnosti u *honeynet*-u druge generacije je *snort_inline*, posebna inačica poznatog alata za otkrivanje napadača *Snort* [15].

Prikupljanje podataka jedna je od najvažnijih aktivnosti *honeynet*-a te bez tog elementa *honeynet* nema smisla. Ključ dobrog prikupljanja podataka je prikupljanje svih dostupnih podataka sa što više različitih slojeva. Primjerice, ukoliko se podaci prikupljaju sa samo jednog sloja (npr. samo ono što napadač piše na tipkovnici), moguće je izgubiti cjelokupnu informaciju (npr. što se

dogodilo kada je pokrenuo određenu naredbu). Iz tog se razloga podaci u *honeynet* sustavu druge generacije prikupljaju na tri različite razine: dnevnik vatrozida, mrežni promet i dnevnik sustava.

Dnevnik vatrozida već postoji, s obzirom na to da je za ograničavanje broja veza odabran dio jezgre operacijskog sustava – *netfilter*. Dnevnik vatrozida može se pronaći u */var/log/messages* datoteci. Ovo je kritična informacija jer govori o aktivnostima napadača. Osim toga, koristeći dnevnik vatrozida, moguće je otkriti nove, do tada nepoznate, načine napada.

Hvatanje cjelokupnog mrežnog prometa također ima veliku ulogu pri prikupljanju podataka. Na osnovu uhvaćenog prometa, kasnije se mogu konstruirati akcije napadača i potanko istražiti njegovi motivi. Kod *honeynet* sustava druge generacije u tu se svrhu koristi program imena *Snort*. Iako bi se isti učinak dobio korištenjem *snort_inline* komponente, to nije preporučljivo zbog samih performansi sustava.

Treći se element, hvatanje napadačevih akcija na samom sustavu, pokazao kao najteži za ostvariti. Napadači danas često koriste enkripciju kako bi zaštitili komunikaciju s napadnutim računalom te stoga prikupljanje podataka direktno „sa žice“ nema smisla. Iz tog je razloga bilo potrebno razviti sustav koji prikuplja podatke o napadaču sa samog napadnutog računala, ali, da izazov bude veći, bez znanja napadača. Kako se informacije prikupljaju sa samog napadnutog računala, uspješno je riješen problem enkripcije. U tu se svrhu u *honeynet* sustavima druge generacije koristi alat imena *Sebek*. *Sebek* je dodatak jezgri operacijskog sustava i omogućava praćenje i bilježenje svih aktivnosti napadača [15].

Naravno, potrebno je i implementirati neki oblik obavještavanja administratora sustava o aktivnostima na *Honeypot* sustavu. Idealna je situacija ona u kojoj postoji osoba koja je zadužena za nadzor nad cjelokupnim *honeynet* sustavom. No kako je to vrlo zahtjevno i često nije praksa, potrebno je implementirati i sustav za automatizirano uzbunjivanje. U sustavima druge generacije koristi se automatizirani alat za uzbunjivanje pod imenom *Swatch*. *Swatch* je program za automatizirano nadziranje raznih dnevnika. On se može postaviti na način da uzbunjuje administratora u određenim situacijama koje se mogu detaljno navesti u programu.

Postojeći *Honeypot* sustavi pate od raznih „otisaka prstiju“ (*fingerprinting*) tehnika, a arhitektura ne može u potpunosti iskoristiti sve mogućnosti analize podataka zbog nedovoljno preciznog sustava kontrole podataka. Za rješavanje tog problema ubuduće se predlaže korištenje SDN (*Software Defined Network*) *Honeynet* sustava [25], [26].

6. SIMULACIJA NAPADA

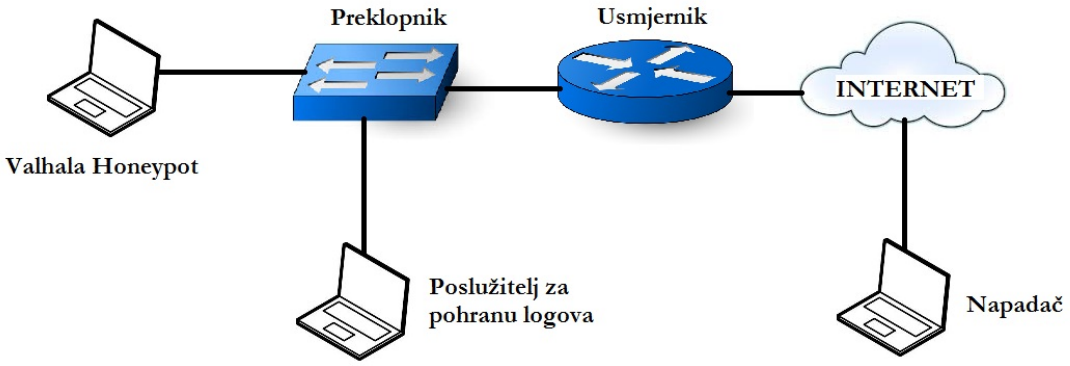
U ovom će poglavlju biti opisana simulacija napada na nisko/srednje interaktivni *Honeypot* sustav *Valhala* i na visoko interaktivni *Honeypot* sustav *Bifrozt*. Svrha simulacije napada je utvrditi mogućnosti prikupljanja informacija o napadaču, tehnikama napada i alatima koje napadač koristi kako bi zaštitili informacijski sustav. S druge strane, cilj istraživanja je prikazati radi li *Honeypot* sustav ispravno i može li ponuditi potpuniju zaštitu informacijskog sustava uz već postojeće konvencionalne zaštite navedene u drugom poglavlju.

6.1. *Valhala Honeypot* sustav

Valhala Honeypot vrlo je jednostavan program za korištenje. Može se svrstati u kategoriju nisko do srednje interaktivnih *Honeypot* sustava. Softver daje usluge *Web*, FTP, TFTP, *finger*, pop3, SMTP, *echo*, *daytime*, *telnet* poslužitelja i prosljeđivanje portova. Projekt *Valhala Honeypot* postao je javno dostupan 2004. godine i od tada se stalno razvija. Verzija korištena u ovom radu dostupna je od 2012. godine. Autor programa je Marcos Flávio Araújo Assunção, također i autor 9 knjiga o digitalnoj sigurnosti, profesor, poduzetnik i internacionalni predavač [27].

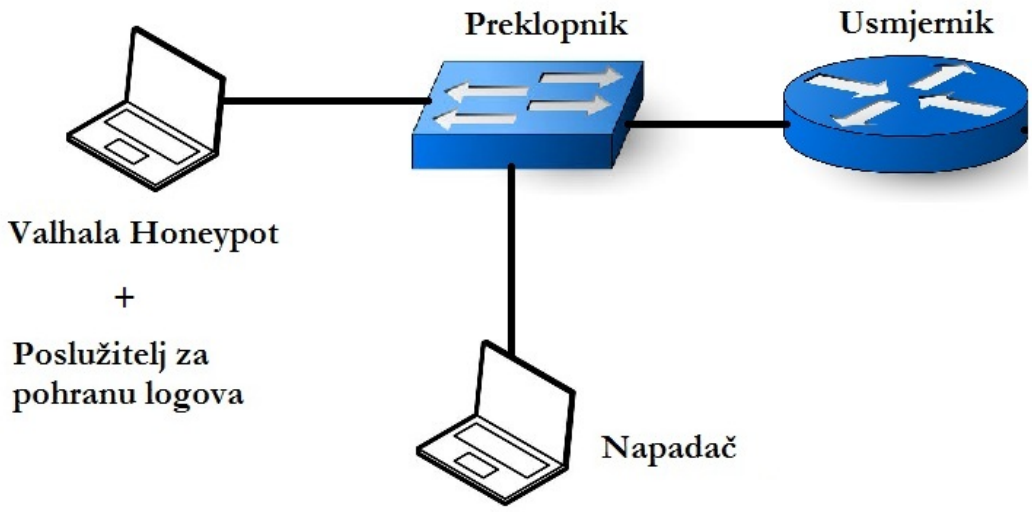
6.1.1. Topologija mreže *Valhala Honeypot* sustava

Predviđena mrežna topologija sastoji se od mrežnih usmjernika, preklopnika, zatim *Valhala Honeypot*a, poslužitelja za spremanje logova i napadača koji pristupa lokalnoj mreži putem Interneta. Stvarna je shema nešto drugačija, iako je princip isti.



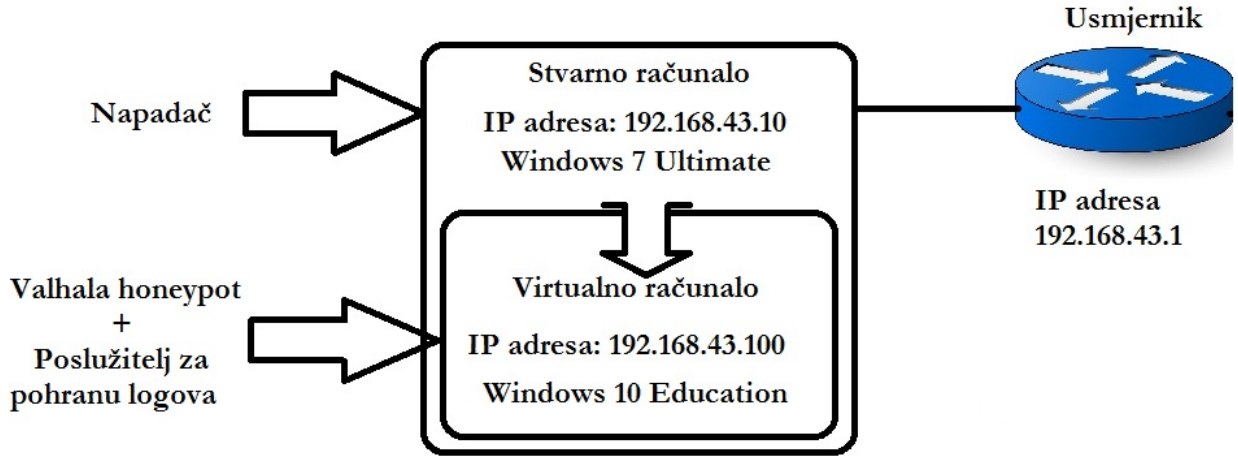
sl. 6.1. Predviđena mreža topologija

Zbog nemogućnosti reproduciranja slanja logova na udaljeni poslužitelj, računalo zaduženo za *Valhala Honeypot* ujedno je i poslužitelj za pohranu podataka. Nadalje, radi jednostavnosti prikaza napada, napadač se nalazi u lokalnoj mreži.



sl. 6.2. Stvarna logička topologija

Na slici 6.2. je logički prikaz stvarne mrežne topologije, a na slici 6.3. prikazana je stvarna fizička topologija *Valhala Honeypot* sustava.

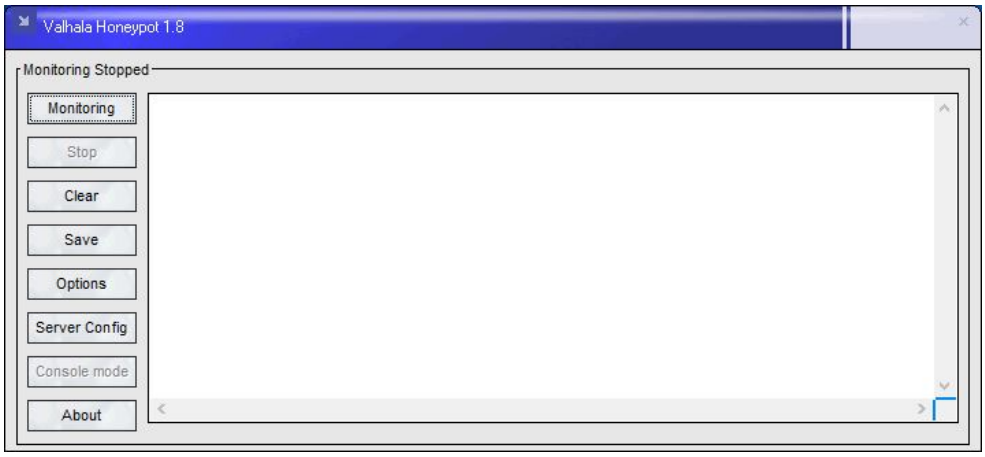


sl. 6.3. Stvarna fizička topologija

Za provođenje simulacije *Honeypot* sustava korišteno je fizičko računalo na kojem je instaliran *Windows 7 Ultimate* operacijski sustav i na njemu je pokrenuto virtualno računalo s *Windows 10 Education* operacijskim sustavom. Na virualnom računalu instaliran je *Valhala honeypot* softver. Logovi se pohranjuju u direktorij gdje je smještena izvršna datoteka *Honeypot*a. Napadač prije samog napada skenira portove. Ukoliko je neki od portova otvoren, napadaču se otvara mogućnost interakcije sa sustavom.

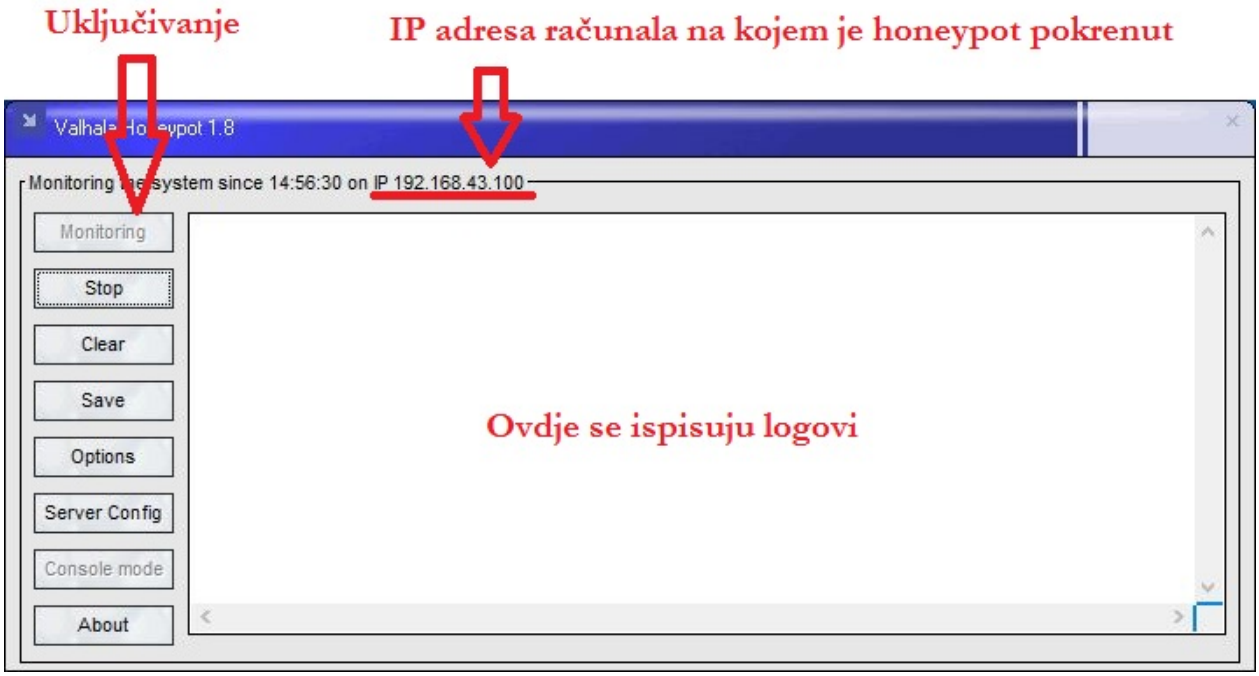
6.1.2. Općenito o *Valhala honeypot* sustavu

Program se sastoji od jedne izvršne datoteke, a proces instalacije softvera vrlo je jednostavan. *Valhala Honeypot* moguće je instalirati samo na *Windows* operacijskim sustavima. Testirane verzije sustava na kojima radi *Honeypot* su *Windows 7 Ultimate* i *Windows 10 Education*. Program je moguće isprobati potpuno besplatno, a poveznica na program nalazi se u prilogu. Korištena verzija *Honeypot* sustava je 1.8. Iako postoji i verzija 1.9, prethodna verzija prevedena je na engleski jezik. Nakon pokretanja izvršne datoteke otvara se sučelje programa koje je prikazano na slici 6.4.



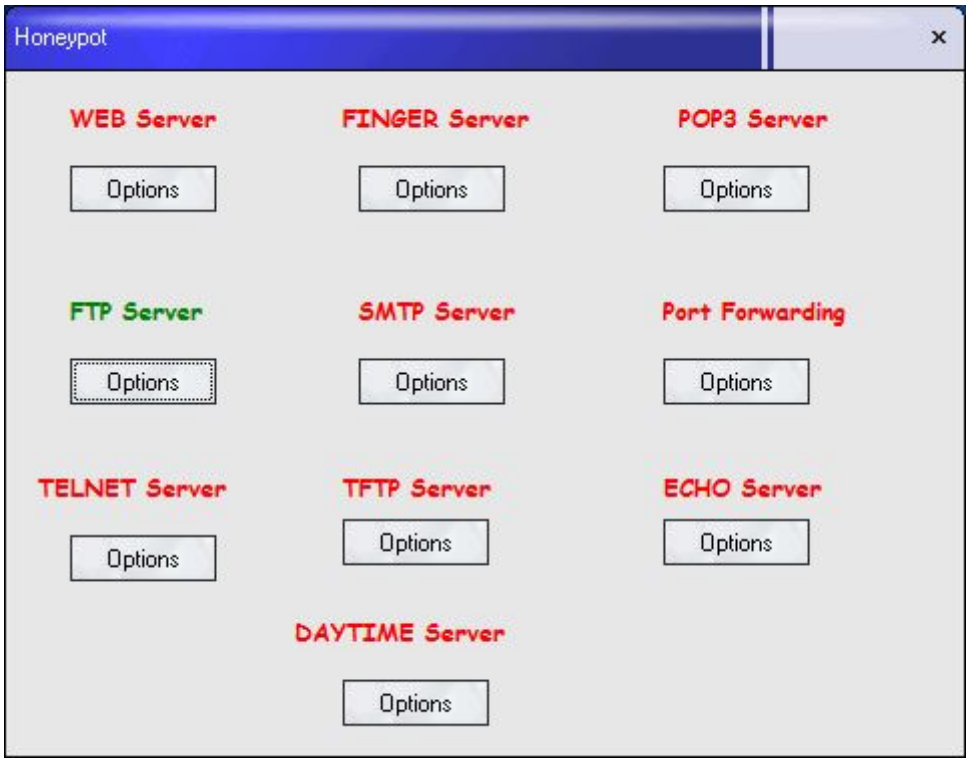
sl. 6.4. Sučelje *Valhala Honeypot* sustava

Uključenjem Monitoring sučelja moguće je pokrenuti Valahala honeypot. Honeypot počinje "slušati" promet koji dolazi na točno određena vrata odnosno portove. Također, program se smanjuje te odlazi u Taskbar.



sl. 6.5. Puštanje u rad *Honeypot* sustava

Uključenjem *Options* sučelja moguće je podesiti slanje logova sustava na neki udaljeni poslužitelj ili putem elektroničke pošte. Također je moguće podesiti parametre *Honeypot* sustava s nekog drugog poslužitelja, a tu su još i mnoge druge opcije kao na primjer propuštanje dodatnih vrata (portova).

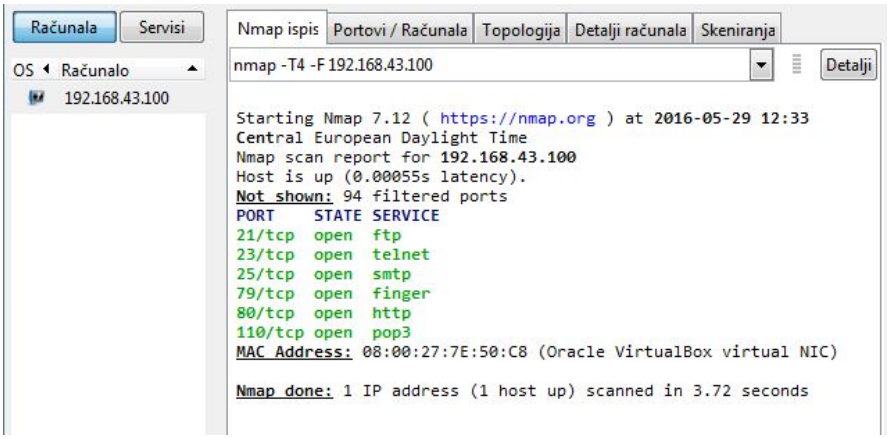


sl. 6.6. Glavni izbornik za podešavanje poslužitelja

Na slici 6.6. prikazani su poslužitelji koje je moguće podesiti. Bitno je napomenuti kako su usluge *WEB*, TFTP i FTP poslužitelja jedine „prave“ usluge za razliku od ostalih koje su samo simulirane. „Uključene“ usluge označene su zelenom, dok su ostale crvenom bojom.

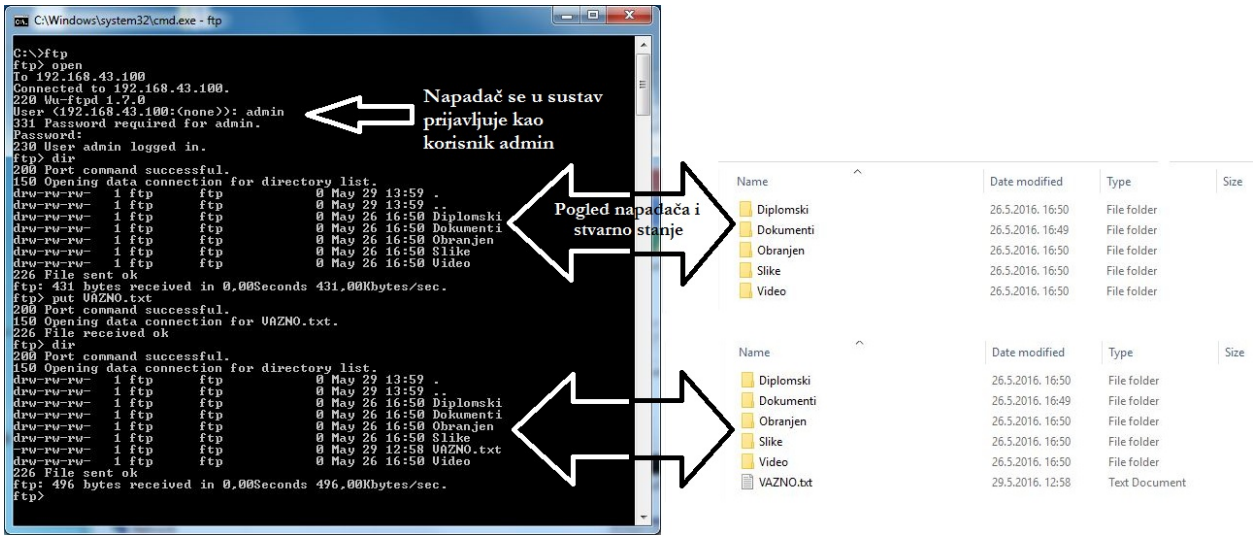
6.1.3. Pristup FTP poslužitelju

Napadač koristi *nmap* skener ranjivosti odnosno inačicu s grafičkim sučeljem *zenmap*. Na sljedećoj slici 6.7. prikazani su rezultati skeniranja. Vidljivi su trenutno otvoreni portovi na virtualnom računalu, odnosno na samom *Honeypot*u. Otvoreni portovi su 21 (*File Transfer Protocol*), 23 (*Telnet*), 25 (*Simple Mail Transfer Protocol*), 79 (*Finger protocol*), 80 (*HyperText Transfer Protocol*), 110 (*Post Office Protocol verzija 3*).



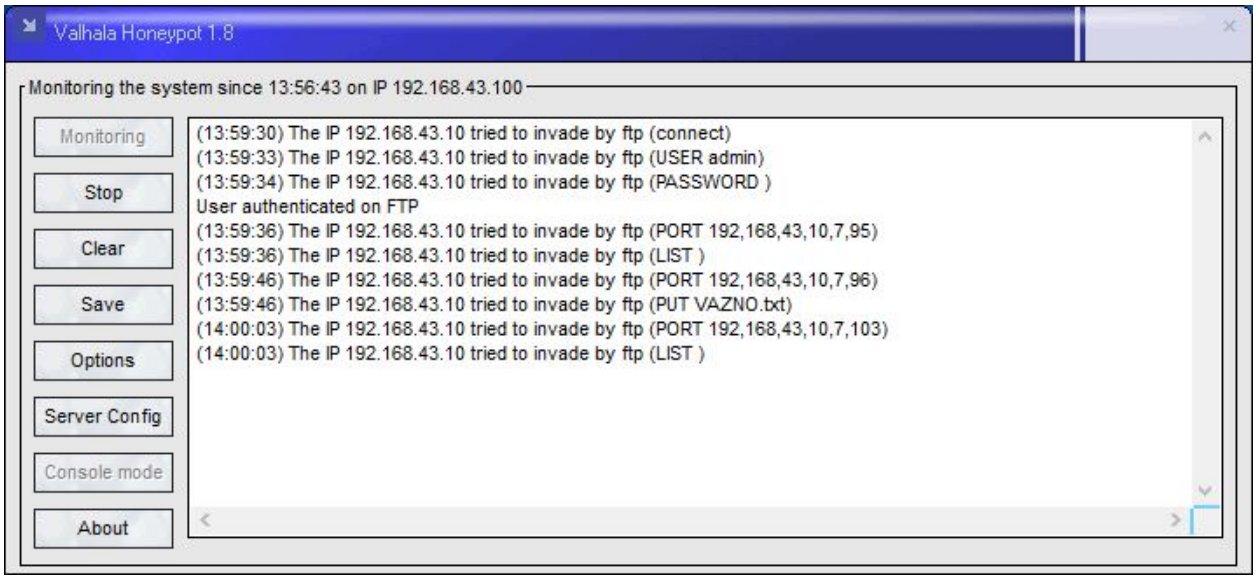
sl. 6.7. *Zenmap* skeniranje portova

U prvom primjeru napadač pristupa FTP poslužitelju. *Honeypot* sustav svaki pokušaj prijave u sustav bilježi u logovima iz kojih možemo iščitati koje parametre je koristio.



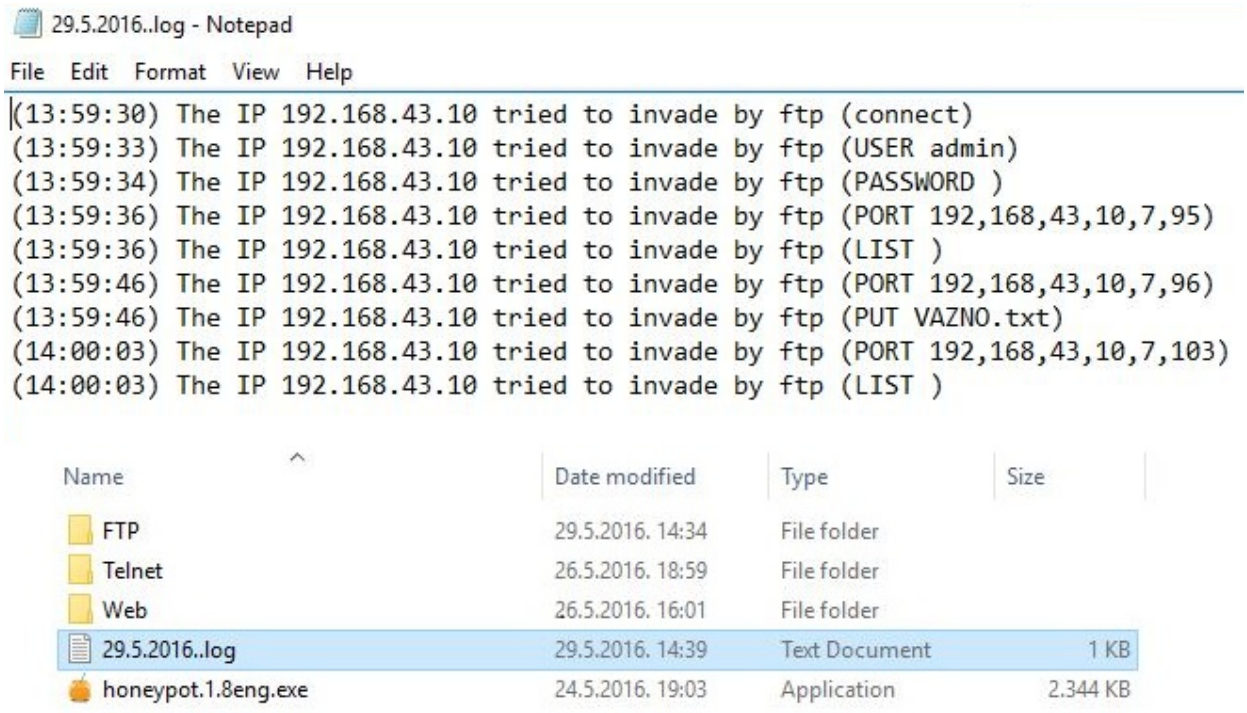
sl. 6.8. Napadač u interakciji s *FTP* poslužiteljem

Na slici 6.8. je vidljivo spajanje napadača na *Honeypot* FTP poslužitelj, a na slici 6.9. su prikazani logovi spajanja.



sl. 6.9. Logovi spajanja na *Valhala Honeypot* FTP poslužitelj

Ukoliko se uključi lokalna pohrana log datoteka, *Honeypot* ih sprema u zadani direktorij *C:\Valhala* u obliku *[datum].log*. Na slici 6.9. prikazana je log datoteka iz koje je vidljivo da napadač pristupa s IP adrese 192.168.43.100 te koristi korisničko ime admin i lozinku „prazno polje“. Iz linije loga u 13:59:46 sati korištena je naredba *PUT* kojom napadač stavlja datoteku VAZNO u tekstualnom formatu na *honeypot* sustav.

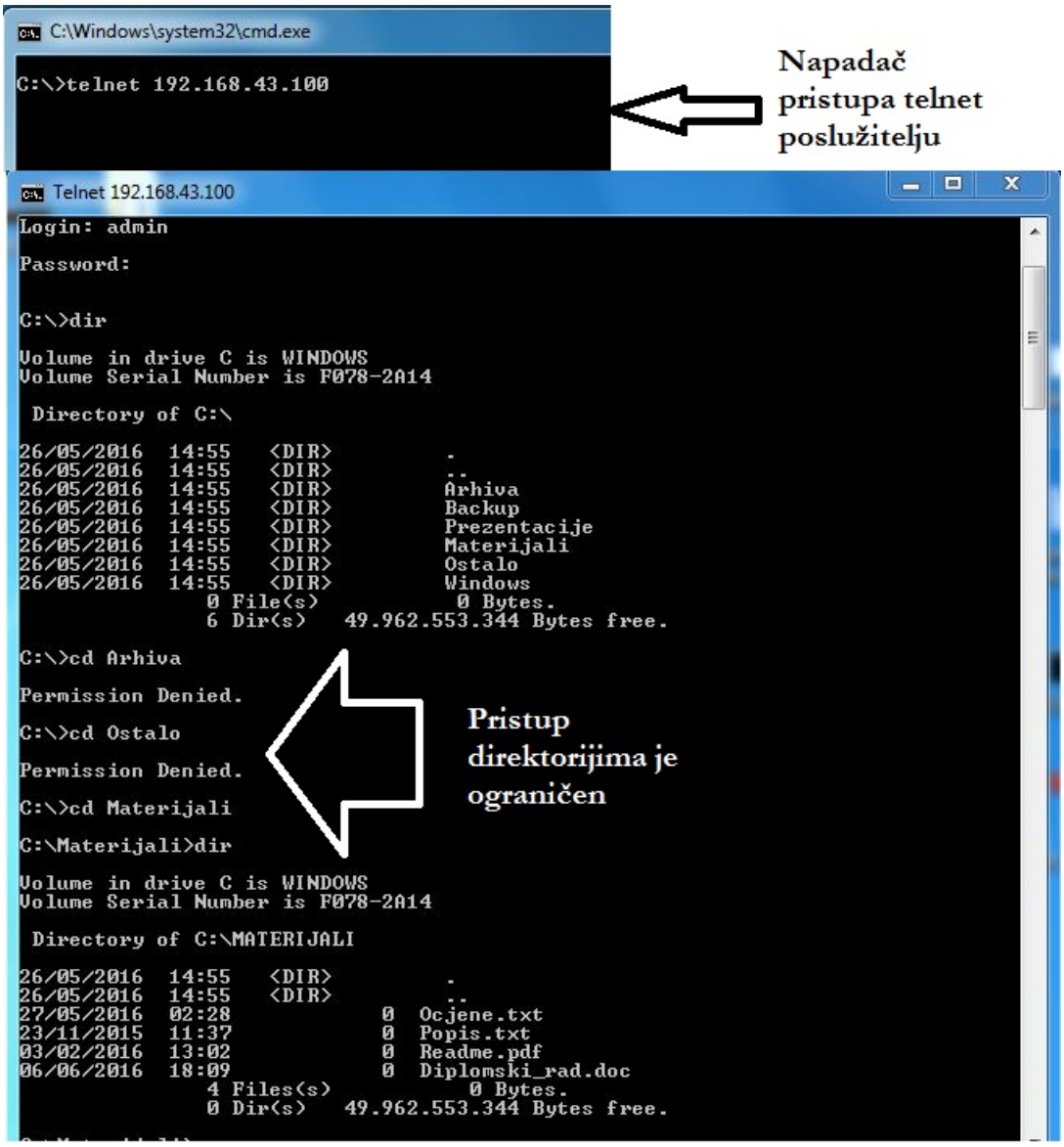


sl. 6.10. *Log* datoteka u direktoriju izvršne datoteke *Honeypot*a

Na slici 6.10. prikazan je zapis *Valhala* sustava dobiven u interakciji s napadačem kojeg je moguće otvoriti u bilo kojem tekstualnom procesoru.

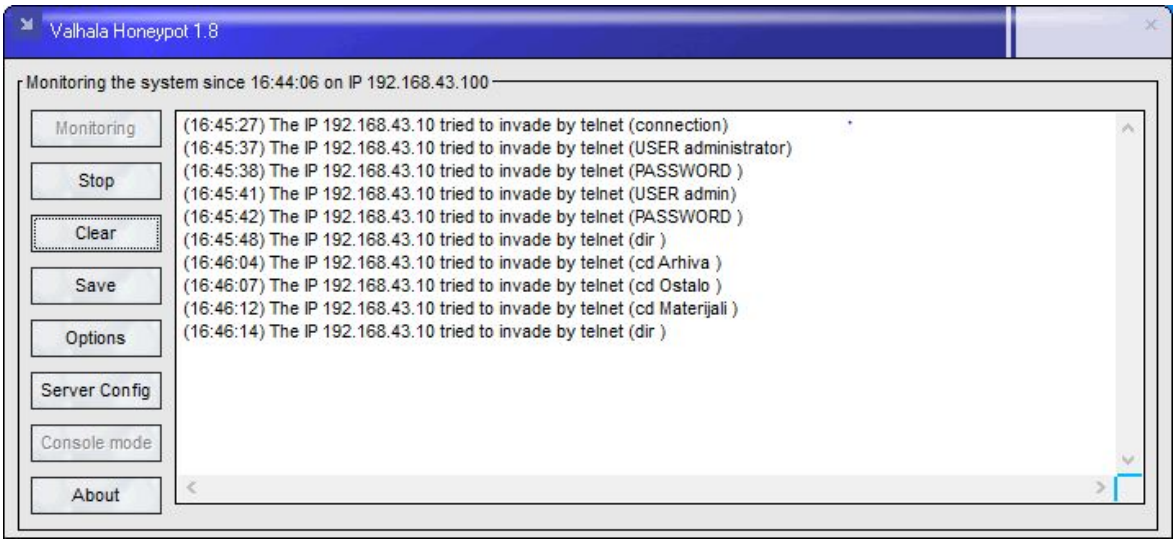
6.1.4. Pristup *Telnet* poslužitelju

Kako bi se lakše zabilježio upad napadača u sustav, isti se podešava na način da se uključe zadane postavke (korisničko ime admin i lozinka „prazno polje“). Iz slike 6.11. vidljivo je da je pristup direktorijima „Arhiva“ i „Ostalo" zabranjen, a direktoriju „Materijali“ dopušten. Također, vidljivi su i podaci o particiji kao što su naziv i serijski broj.



sl. 6.11. Ograničen pristup *Telnet* poslužitelju

Na slici 6.12. vidljiv je zapis *Honeypot* sustava koji je zabilježio interakciju napadača s *telnet* poslužiteljem. Bitno je primijetiti da se svaki pokušaj unošenja korisničkog imena i lozinke bilježi, ali i točno što je napadač unio u naredbeni redak.

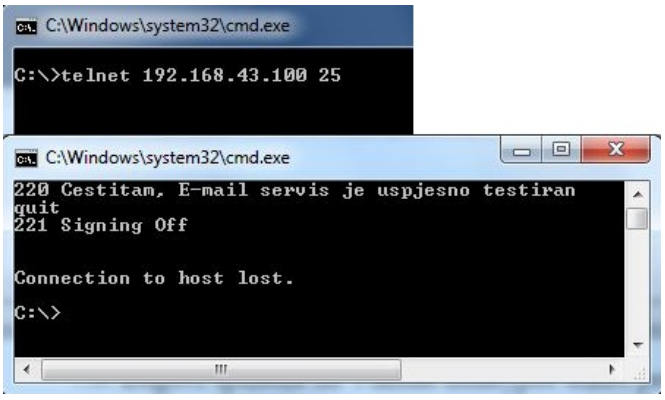


sl. 6.12. Logovi spajanja na *Valhala Honeypot* telnet poslužitelj

Vidljiv je unos podataka kao što su korisničko ime administrator i lozinka „prazno polje“, te cd i dir naredbe koja se koristi za otvaranje i pregled direktorija koji se nalaze u sustavu.

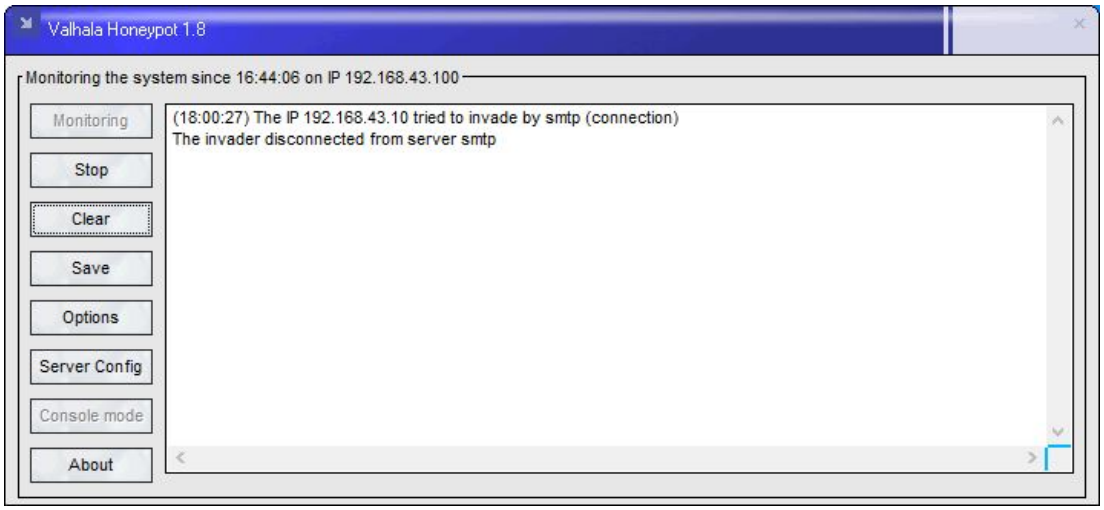
6.1.5. Pristup SMTP poslužitelju

Napadač testira SMTP poslužitelj spajanjem putem *telnet* usluge na port 25. Sustav šalje odgovor i bilježi svaku interakciju.



sl. 6.13. Pokušaj spajanja na *SMTP* poslužitelj

Na slici 6.14. zabilježen je pokušaj testiranja rada SMTP poslužitelja od strane napadača.

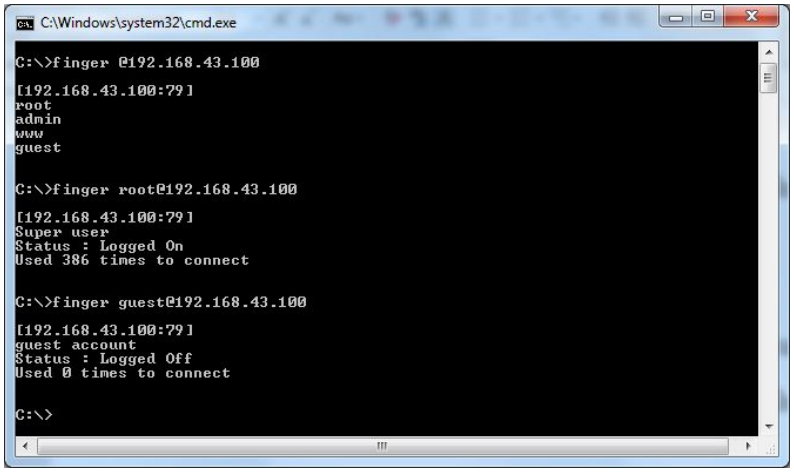


sl. 6.14. Logovi spajanja na *Valhala Honeypot SMTP* poslužitelj

U logovima vidljiv je zapis IP adrese s koje napadač pristupa (192.168.43.10) i protokol (SMTP) s kojim je uspostavio vezu.

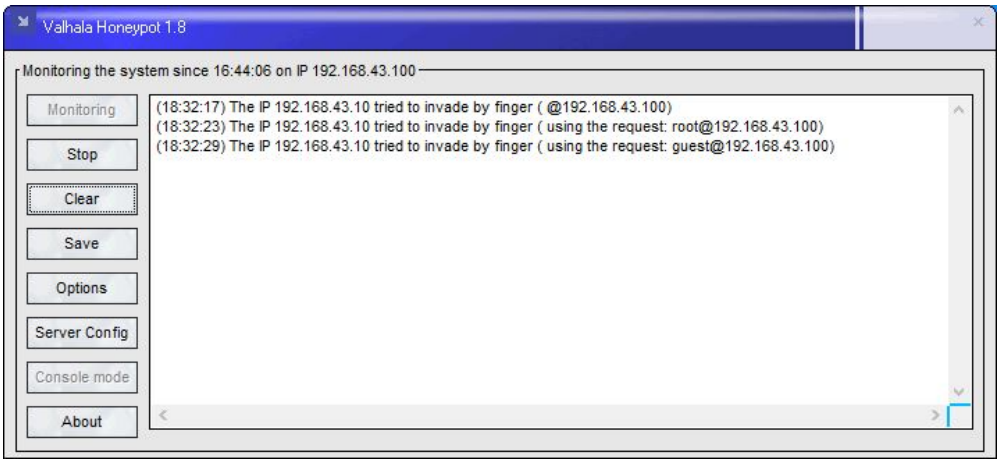
6.1.6. Pristup *Finger* poslužitelju

Napadač može provjeriti koji su korisnici na sustavu te za svaki pojedinačni račun može dobiti informacije o vrsti računa, statusu i koliko je puta korisnik koristio svoj račun, odnosno koliko se puta prijavio u sustav.



sl. 6.15. Pokušaj spajanja na *finger* poslužitelj

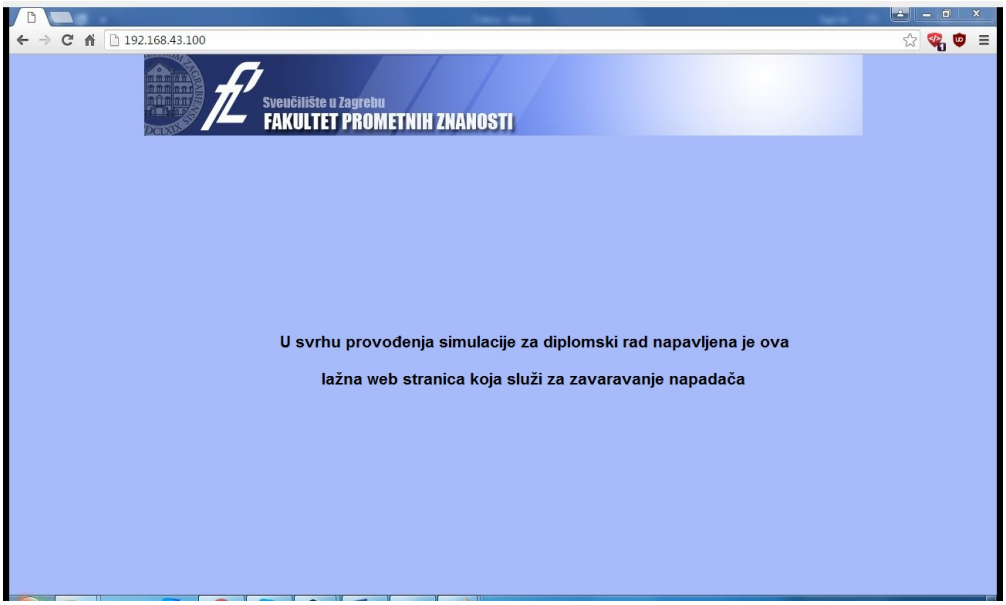
Svaku od tih stavki moguće je podesiti u *Honeypot*u tako da napadač dobije lažnu sliku sustava. Na slici 6.16. je zabilježen pristup *Honeypot*u. U logovima su vidljivi IP adresa s koje napadač pristupa sustavu (192.168.43.10) i zahtjev za spajanje koji mu omogućavaju da se autentificira kao korisnik root i guest.



sl. 6.16. Logovi spajanja na *Valhala Honeypot finger* poslužitelj

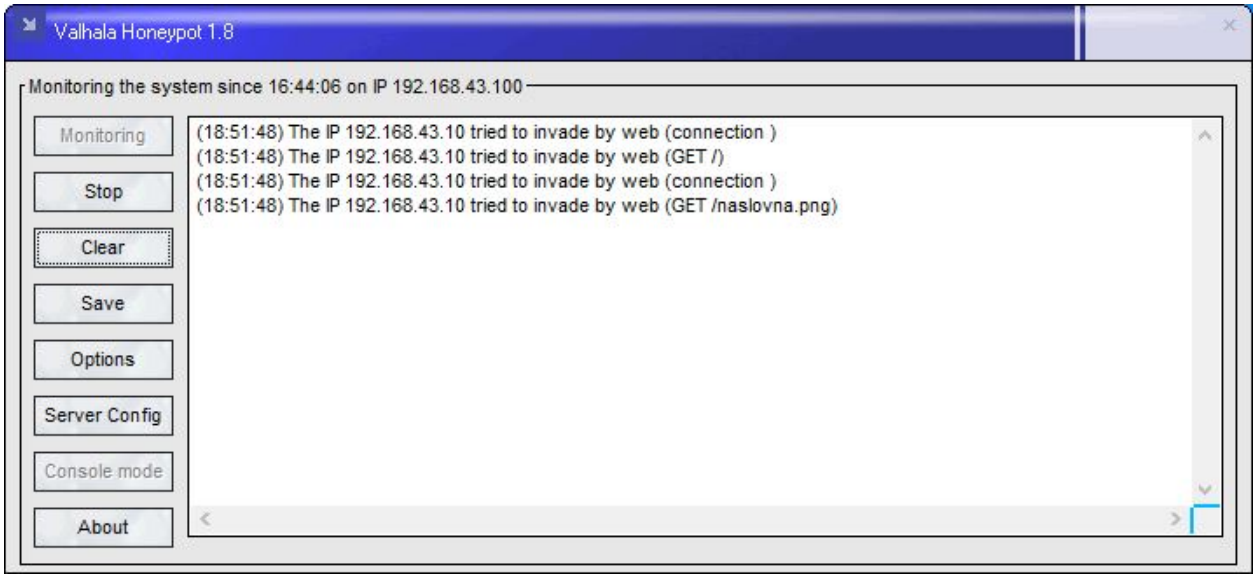
6.1.7. Pristup *Web* poslužitelju

Na sljedećoj slici 6.17. prikazan je primjer lažne *web* stranice.



sl. 6.17. Spajanje na *Web* poslužitelj

U opcijama *Honeypot*a definirana je putanja, u ovom slučaju *C:\Valhala\Web*, na *index.html* datoteku. U toj datoteci napisan je kod za testni primjerak web stranice.

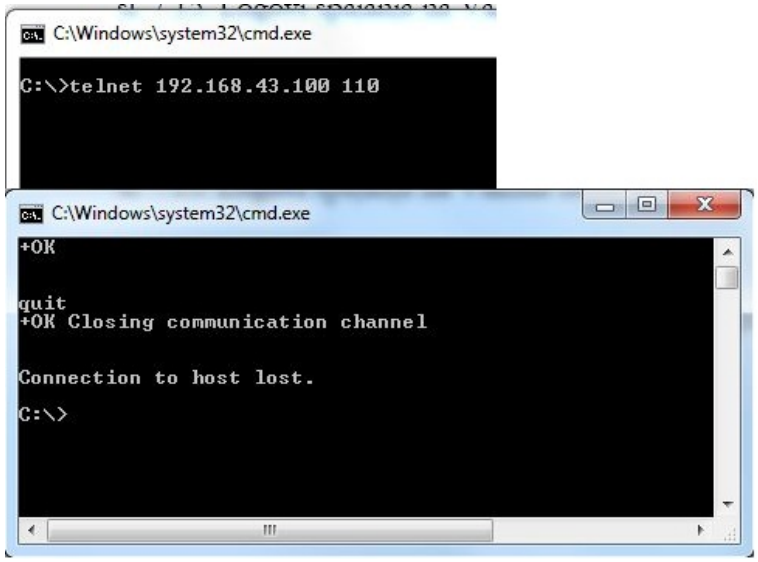


sl. 6.18. Logovi spajanja na *Valhala Honeypot Web* poslužitelj

Na slici 6.18. *Honeypot* sustav je zabilježio pristup *web* stranici. U logovima je moguće vidjeti uspostavljanje veze (*connection*), zatim dohvaćanje početne stranice (root /) i slike početne stranice (naslovna.png).

6.1.8. Pristup *POP3* poslužitelju

Napadač se spaja na POP3 poslužitelj na port 110. Od *Honeypot*a dobiva odgovor da je usluga aktivna. Istovremeno *Honeypot* sustav pohranjuje logove koji bilježe interakciju napadača sa sustavom.



sl. 6.19. Testiranje POP3 poslužitelja

Na slici 6.19. vidljiv je pristup napadača POP3 poslužitelju. Telenet programom testira se njegova funkcionalnost.

6.1.9. Rezultati simulacije *Valhala Honeypot* sustava

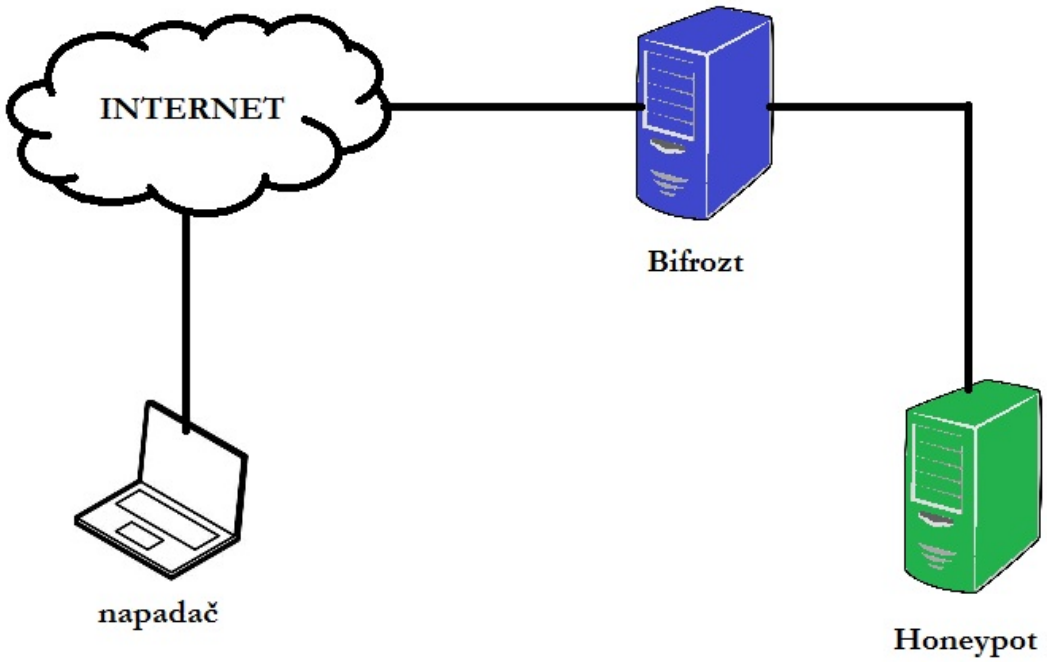
Na temelju prethodnih testiranja *Valhala Honeypot* sustava može se utvrditi da je sustav zadovoljio očekivanja. U traženju vrijednih resursa sustava napadač je uspješno ometan i prevaren, kao što možemo vidjeti u primjerima FTP poslužitelja gdje napadač vidi listu direktorija u putanji koja ustvari ne postoji te je na taj način izoliran od stvarnog sustava. Nadalje, usluga *telnet* poslužitelja je simulirana te je svaka interakcija s tim poslužiteljem gubitak vremena za napadača, kao što su i usluge SMTP, *finger* i POP3 poslužitelja. Napadač se spaja na *Web* poslužitelj putem Internet preglednika gdje *Honeypot* odgovara na *http* zahtjev i na taj način zadobije njegovo poverenje da je sustav legitiman. Na temelju dobivenih logova sustava možemo vidjeti tehnike napada, odnosno koje naredbe napadači unose i kojim se alatima koriste, kao što su *nmap* skener portova, što može pridonijeti podizanju sigurnosti produkcijskih sustava u mreži.

Kao što je navedeno u poglavlju 6.1.2. (informacije dobivene od proizvođača softvera), *Valhala Honeypot* sustav koristi usluge simuliranih poslužitelja koji predstavljaju nizak sigurnosni rizik za sustav odnosno organizaciju, ali i „prave“ usluge kao što su *Web*, FTP i TFTP poslužitelj. U određenim uvjetima, napadač bi mogao iskoristiti propuste „pravih“ usluga kako bi zadobio pristup sustavu. Također zbog niže razine interaktivnosti s napadačem, *Valhala Honeypot* sustav

može biti relativno brzo otkriven nekom iskusnijem napadaču. No, uspješan je u zavaravanju automatiziranih skenera odnosno skripti, ali i napadača početnika.

6.2. *Bifrozt Honeypot* sustav

U prethodnom je poglavlju opisan *honeynet* sustav koji je nastao od strane *Honeynet Project* organizacije. Podrška za programsko rješenje visoko interaktivnog *Honeypot* sustava *Honeywall* CDROM prestaje 2015. godine. Budući da *Honeywall* CDROM nije aktualan program, u potrazi za visoko interaktivnim *Honeypot* sustavom odabran je *Bifrozt Honeypot* sustav.



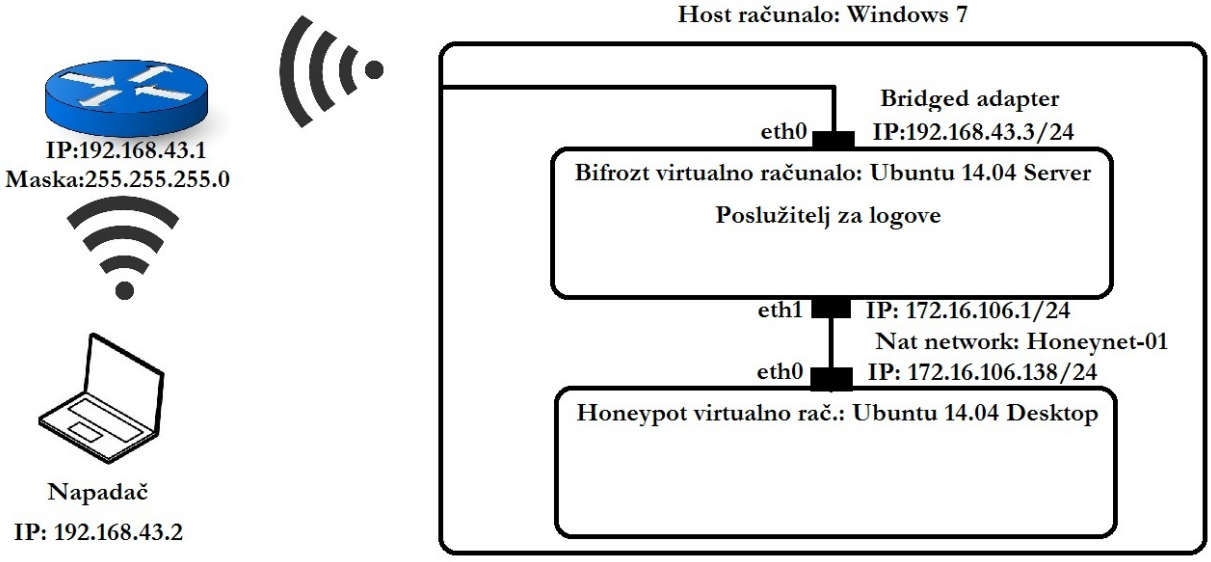
sl. 6.20. Općeniti prikaz *Bifrozt Honeypot* sustava [28]

Općenito govoreći, *Bifrozt* je NAT uređaj s DHCP poslužiteljem koji je implementiran s jednim mrežnim adapterom spojenim na vanjsku mrežu (*Internet*) i drugim adapterom na internu mrežu (*Honeypot*). Za razliku od drugih NAT uređaja, *Bifrozt* radi kao SSH *proxy* između napadača i *Honeypot* sustava. Ukoliko se instalira SSH poslužitelj na internoj mreži odnosno *Honeypot*u, *Bifrozt* će zabilježiti u teksutalnu datoteku svaku interakciju napadača s *Honeypot*om i spremiti kopije svih preuzetih dokumenata. Kako bi SSH poslužitelj u internoj mreži radio ispravno nije

potrebno instalirati nikakav dodatni softver, kompajlirati jezgru sustava ili koristiti posebnu vrstu operacijskog sustava. *Bifrozt* postavlja ograničenje na izlazni promet na određeni broj portova. Ukoliko se prekorači količina prometa, broj izlaznih paketa se smanjuje [28].

6.2.1. Topologija mreže

Za potrebe ovog diplomskog rada napravljena je topologija mreže koja se sastoji od bežičnog rutera, jednog računala koji preuzima ulogu napadača i drugog računala unutar kojeg se virtualizira *Bifrozt proxy* i *Honeypot* sustav. Na sljedećoj je slici 6.21. prikazana shema korištene mreže.

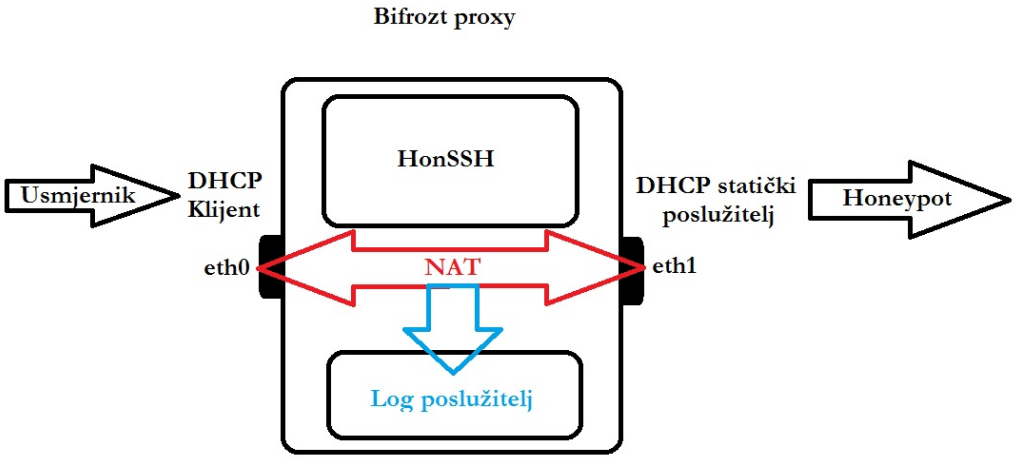


sl. 6.21. Shema *Bifrozt Honeypot* mreže

Na prethodnoj slici 6.20. prikazana je shema na kojoj je vidljivo da napadač na *Bifrozt* sustav pristupa putem Interneta. U ovom radu testirat će se koncept rada sustava te iz tog razloga napadač neće biti anoniman već lokalno računalo u mreži.

Kako bi se detaljnije prikazalo što točno je *Bifrozt Honeypot* sustav, mora se pogledati svaku pojedinačnu komponentu sustava. Iz perspektive napadača računalo nazvano *Honeypot* predstavlja legitiman resurs dok je iz perspektive administratora *Honeypot* računalo tu sa svrhom da bude ispitano, napadnuto i testirano. Na fizičko računalo instaliran je operacijski sustav

Windows 7, a na njemu su instalirana dva virtualna stroja bazirana na *Ubuntu 14.04 Server* i *Desktop* verziji operacijskog sustava. Na slici 6.22. prikazana je shema *Bifrozt* proxy sustava

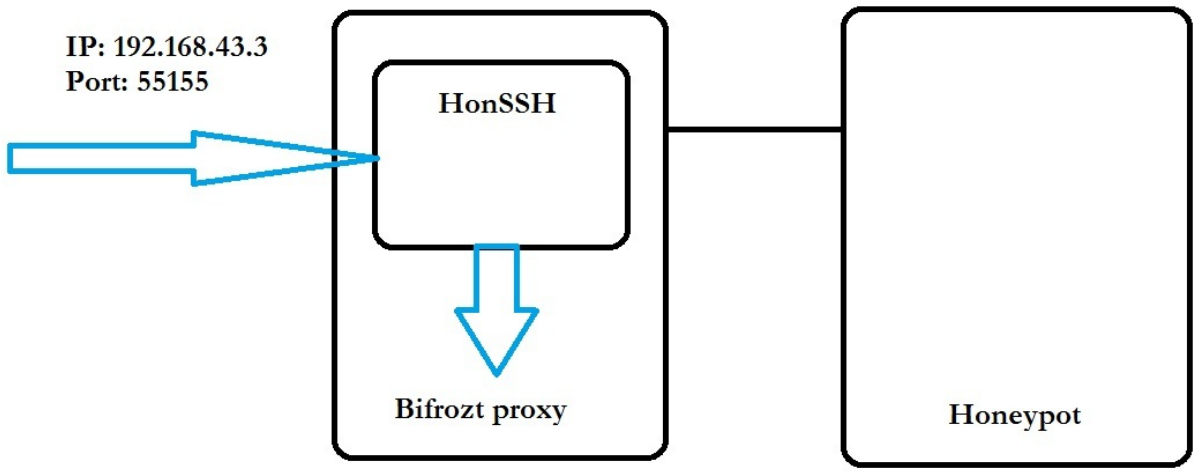


sl. 6.22. Shema *Bifrozt proxy* sustava

Honeypot računalo sastoji se od dvije komponente, a to su operacijski sustav *Ubuntu* 14.04 *Desktop* verzija i SSH poslužitelj. *Bifrozt* se sastoji od operacijskog sustava *Ubuntu 14.04 Server* i *Bifrozt* softvera. *Bifrozt* računalo je *proxy* odnosno premosnik između vanjske mreže (na primjer lokalna mreža 192.168.43.0/24) i unutarnje mreže (u ovom slučaju *Honeynet-01*). U svrhu premošćivanja, korištena je NAT (*Network Address Translation*) tehnologija.

6.2.2. Administracija *Bifrozt* proxya

U svrhu provođenja diplomskog rada, sustavu je moguće pristupiti lokalno s *host* računala unutar samog virtualnog stroja ili s vanjske mreže pomoću programa za udaljeni pristup kao što je *Putty*. Bitno je za naglasiti da je pristup administraciji moguć jedino ako se sustavu pristupa na administrativni port 55155. Na sljedećoj slici 6.23. prikazana je shema pristupa *Bifrozt* sustavu od strane administratora sustava.



sl. 6.23. Administrator pristupa *Bifrozt* proxyju

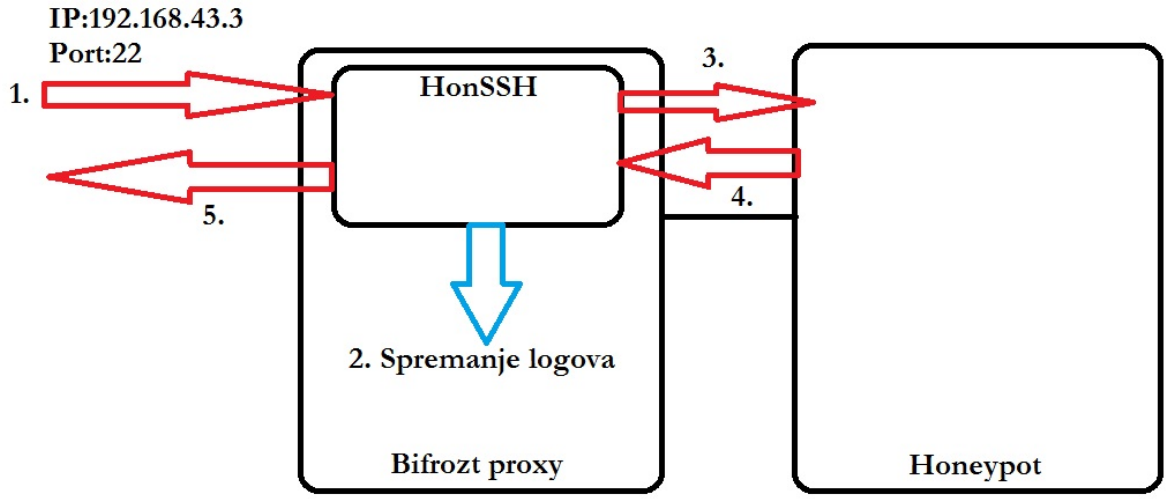
Ukoliko se *putty* pokrene *Bifrozt* šalje, radi autentifikacije, otisak sigurnosnih ključeva računalu s kojeg pristupamo. Ukoliko se na upit odgovori potvrdno uspostavlja se trajna veza sa SSH poslužiteljem.

Nakon autorizacije u *Bifrozt* sustav, mogu se podesiti sljedeći parametri. U direktoriju */opt/HonSSH* nalazi se datoteka *HonSSH.cfg* gdje je zapisana konfiguracija *HonSSH*a. Svaki pokušaj interakcije napadača s *Honeypot* sustavom bilježi se u logovima u direktoriju */opt/HonSSH/sessions/<ime_računala>/<*IP*_adresa_napadača>*. Instalacija *Bifrozta* na računalo sprema se u datoteku *Bifrozt_setup.log* u direktorij */var/log/*. Mrežne postavke nalaze se u direktoriju */etc/network/* u kojem se nalaze datoteke *interfaces* i *ipv4hater*. U datoteci *interfaces* moguće je podesiti postavke mrežnih adaptera, a u *ipv4hater* postavke vatrozida. Promjena postavki DHCP poslužitelja koji daje statičku IP adresu *Honeypot*u nalazi se u direktoriju */etc/dhcp/*, odnosno u datoteci *dhcpd.conf*. Zadnja je bitna datoteka *sshd.conf* te se ona nalazi u direktoriju */etc/ssh/*. U njoj su zapisane postavke SSH poslužitelja.

Kao što je već navedeno u prethodnom poglavlju, *Bifrozt* uz pomoć vatrozida (*/etc/netwrok/ipv4hater*) postavlja određena ograničenja na ulazni i izlazni promet. Na ulazno sučelje (*eth0*) otvorena je veza prema portovima 22 i administrativnom portu 55155. Protokoli koji su propušteni u odlaznom prometu iz interne mreže gdje je *Honeypot* smješten prema vanjskoj mreži obuhvaćaju ICMP (*ping*), FTP, *Telnet*, DNS (*Domain Name System*), HTTP, NTP (*Network Time Protocol*), SNMP (*Simple Network Management Protocol*) i IRC (*Internet Relay Chat*). Svaki od tih protokola ima ograničenje odlaznog prometa s *Honeypot*a prema vanjskoj mreži. Ukoliko broj paketa prelazi definiranu granicu tada se odlazni paketi režu, odnosno onaj dio paketa koji prelazi tu granicu se ne šalju.

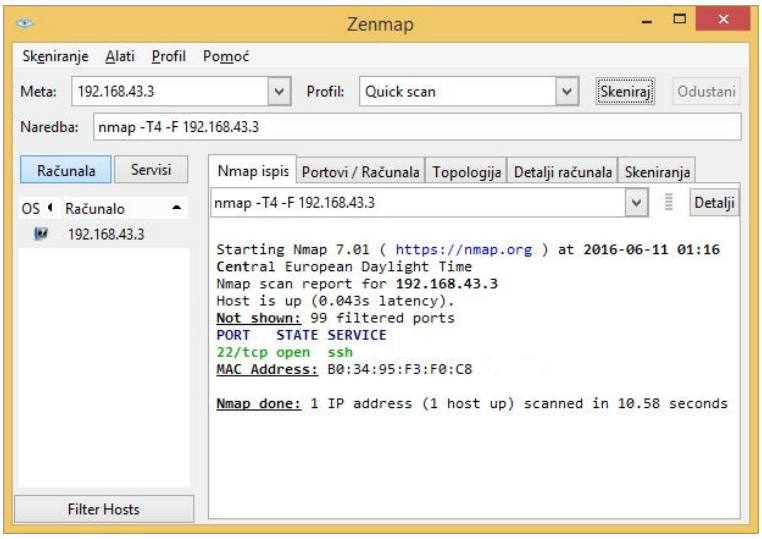
6.2.3. Simulacija napada na *Bifrozt* sustav

Napadač koji se nalazi u lokalnoj mreži pristupa *Honeypotu* na port 22. Prvo dolazi u interakciju s *Bifroztom*.



sl. 6.24. Napadač pristupa *Honeypot* računalu

Bifrozt je za napadača transparentan te ga uopće „ne vidi“. On misli da se na adresi 192.168.43.3 nalazi *Honeypot* (odnosno legitimno računalo), iako se nalazi *Bifrozt* koji dolaznu SSH vezu prosljeđuje *Honeypot* računalu i svaki pokušaj interakcije bilježi u logovima. Zatim *Bifrozt* filtrira odlazni promet s *Honeypot*a, koji ide prema vanjskoj mreži, po definiranim pravilima vatrozida. Na sljedećoj slici 6.25. vidljiv je pregled skeniranja mreže *nmap* programom.



sl. 6.25. Napadač skenira računalo

Napadač vidi da je otvoren *port* 22, nakon čega se pokušava spojiti na *Honeypot* računalo. Svaki pokušaj spajanja *Bifrozt* bilježi u log datoteci /opt/*HonSSH*/log/*HonSSH*.log.

```
2016-06-12 10:08:32+0200 [HonsshClientTransport,client] [SSH] - Detected Public Key Auth - Disabling!
2016-06-12 10:08:32+0200 [HonsshClientTransport,client] [PLUGIN][OUTPUT-TXTLOG] - LOGIN_FAILED
2016-06-12 10:09:34+0200 [HonsshClientTransport,client] connection lost
2016-06-12 10:09:34+0200 [HonsshClientTransport,client] [CLIENT] - Lost connection with the Honeypot: Bifrozt (172.16.106.138:22)
2016-06-12 10:09:34+0200 [HonsshClientTransport,client] Stopping factory <honssh.client.HonsshClientFactory instance at 0xb69dd9ec>
                         20160612_100734_904107,192.168.43.2,admin,admin,False
                         20160612_100734_904107,192.168.43.2,admin,adadministrator,False
```

sl. 6.26. Zabilježen pokušaj autorizacije od strane napadača

Također u direktoriju */opt/HonSSH/log* postoji i datoteka u zapisu [godinamjesecdan_vrijeme_broj] u kojoj su zapisani korisničko ime i zaporka koje je napadač unosio prilikom autorizacije u sustav.

6.2.4. *Spoofing* korisničkog imena i zaporke

HonSSH se nalazi između napadača i *Honeypot*a i to mu daje mogućnost presretanja i izmjene IP paketa. *Spoofing* (*Password Spoofing*) je tehnika zavaravanja gdje napadač unese proizvoljno korisničko ime i zaporku, a *HonSSH* zamjeni s pravim i tako dozvoli napadaču da se autorizira u sustav.

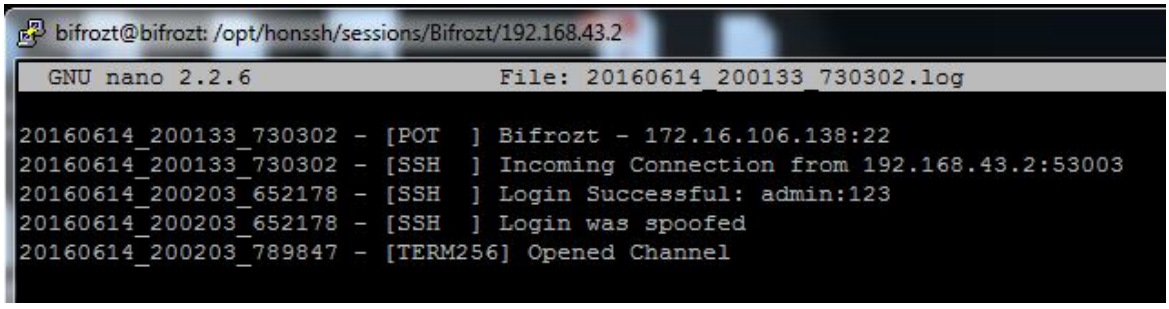
HonSSH spoofing podržava autorizaciju više korisnika i dva mehanizma izvedbe – fiksni i dinamički. Fiksni radi na način da administrator sustava napravi listu s korisničkim imenima i zaporkama na koju će se sustav referencirati prilikom pokušaja autorizacije. Dinamički radi na principu da korisnik unese postotak uspješnosti nasumičnog autoriziranja u sustav. Što je veći postotak, veća je vjerojatnost da autorizacija bude uspješna.

Na primjer, kako bi ispravno podesili fiksni *spoofing* potrebno je napraviti sljedeće izmjene. U datoteci */opt/HonSSH/HonSSH.cfg* pod opcijom *Password Spoofing* potrebno je staviti „*enabled = true*“, zatim u tekućem direktoriju napraviti datoteku *users.cfg* (postoji datoteka *users.cfg.default* koja služi kao referenca).



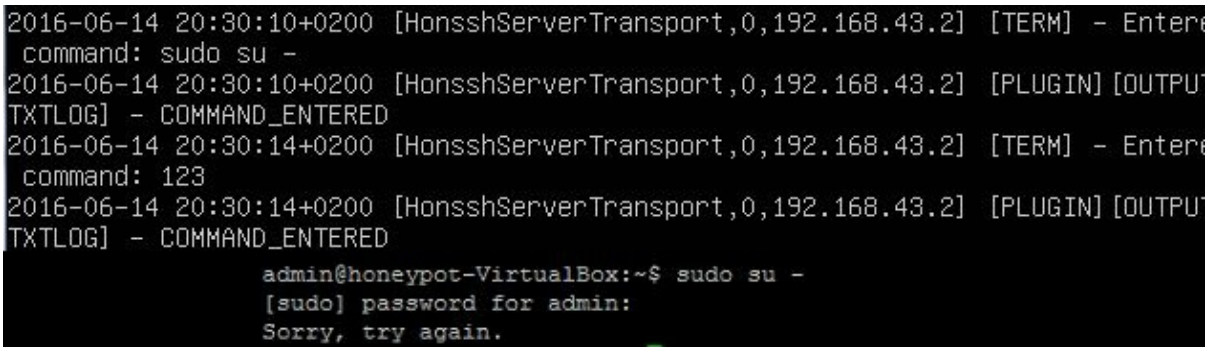
sl. 6.27. *Users.cfg* datoteka u kojoj su smješteni korisničko ime i *Spoofing* zaporke

U svrhu provođenja diplomskog rada napravljen je novi korisnik na *Honeypot* računalu. Korisničko ime je *admin* i zaporka *admin*.



sl. 6.28. Napadač dobiva ograničen pristup sustavu

HonSSH presreće pogrešnu zaporku napadača (123) i mijenja je ispravnom (*admin*). Na taj način napadač zadobiva pristup sustavu, ali bez administratorskih prava.

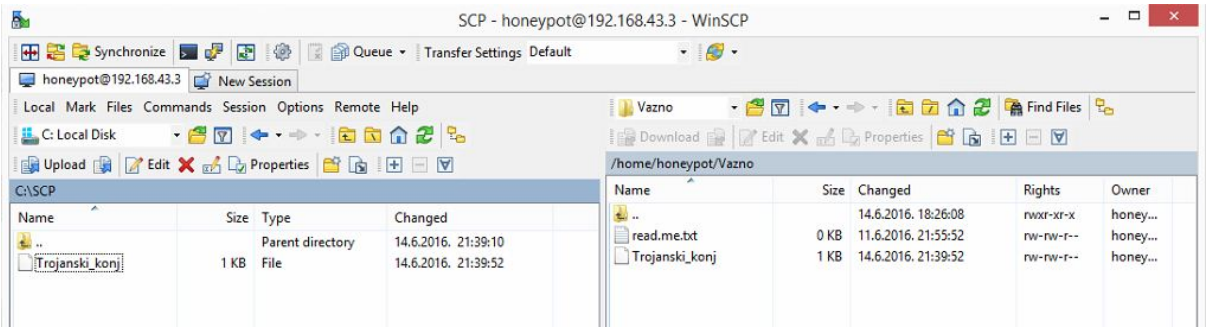


sl. 6.29. Pokušaj dobivanja *root*/administratorskih privilegija

Ukoliko pokuša zatražiti administratorska prava zaporkom kojom je i ušao u sustav, dobit će odgovor da je zaporka pogrešna.

6.2.5. Prijenos datoteka SCP protokolom

Napadač se spaja na port 22 putem SCP protokola. U ovom primjeru računalo napadača koristi *WinSCP* program za interakciju sa SSH poslužiteljem.



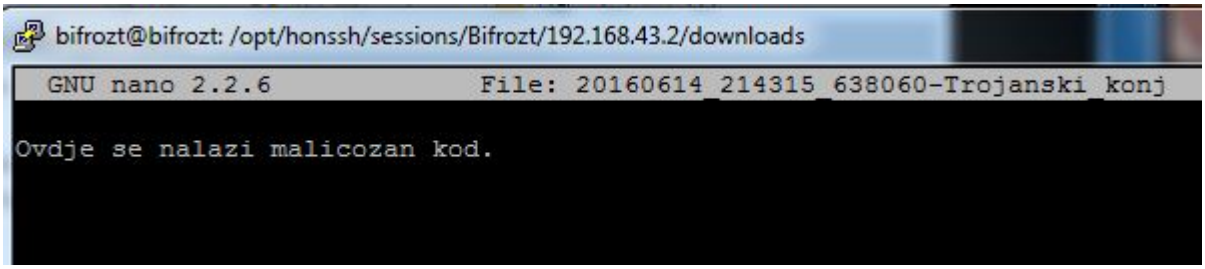
sl. 6.30. Napadač stavlja maliciozni sadržaj na *Honeypot*

Napadač ostavlja maliciozni kod na *Honeypot*u što je moguće vidjeti iz logova sustava.



sl. 6.31. Zabilježeni logovi prijenosa datoteke na *Honeypot*

Iako je u logovima zabilježen prijenos datoteke, teško je iz njih zaključiti o kakvoj se datoteci radi.



sl. 6.32. Analiza malicioznog koda

Iz tog razloga *Bifrozt* sprema kopiju datoteke u direktorij /opt/*HonSSH*/sessions/*Bifrozt*/ <IP_adresa_napadača>/downloads/<godina_mjesec_dan_vrijeme_broj-ime_datoteke>

6.2.6. Rezultati simulacije *Bifrozt Honeypot* sustava

Glavna je svrha visoko interaktivnog *Honeypot* sustava *Bifrozt* zavarati napadača i pri tome naučiti što je više moguće o vektorima napadima i alatima koje napadači koriste. Ovom su simulacijom prikupljene informacije kao što su IP adresa napadača, korisničko ime kojim se pokušao autorizirati u sustav te zaporka koju je unosio (informacije zabilježene u logovima sustava). Uz tehniku password *spoofing* napadač je zavaran na način da misli kako je pogodio korisničku zaporku i da ima puni pristup sustavu. Također prilikom slanja datoteke *Honeypot* uspješno bilježi vrijeme slanja i naziv datoteke, ali i sprema kopiju kako bi omogućio kasniju analizu programskog koda. Iako je *Bifrozt* u razvojnoj fazi, pokazao se kao dobar alat za istraživanje napada jer je u mogućnosti prikupiti dovoljno informacija o napadaču i pružiti potpuniju zaštitu informacijskog sustava što je vidljivo u poglavljima 6.2.3, 6.2.4 i 6.2.5.

Honeypot na koji se napadač spaja koristi „prave“ usluge. Nijedna usluga nije simulirana, a svaki program i usluga ima punu funkcionalnost kao i na produkcijskim sustavima. Za razliku od

nisko interaktivnih sustava, visoko interaktivni predstavlja veću opasnost od zloupotrebljavanja sustava. Ukoliko napadač zadobije administratorska prava, može otkriti da je u interakciji s *Honeypotom*. Tada je u mogućnosti prekinuti svaku interakciju ili izbrisati iz logova svaki pokušaj interakcije i proširiti napad na druga računala unutar lokalne mreže.

7. ZAKLJUČAK

Kako bi ugrozili računalne sustave napadači primjenjuju razne tehnike napada. Za računalne mreže korisnika, *Botnet* predstavlja sve veću opasnost. Nadogradnjom *Botnet* mreža novim programskim alatima otežano je otkrivanje i zaustavljanje napadača. Pojavom *Honeypot* sustava došlo je do povećanja sigurnosti računalnih sustava. *Honeypot* sustavi nemaju nikakvu produkcijsku svrhu što donosi određene prednosti kao što su sigurnost da je svaka interakcija sa sustavom maliciozna. Takvi sustavi imaju i svoje mane jer je napadaču stavljeno na raspolaganje korištenje cijelog sustava ili samo dijela sustava. Napadač je u mogućnosti, u slučaju zadobivanja kontrole nad *Honeypot* sustavom, širiti maliciozne aktivnosti na produkcijska računala u mreži. Stoga je vrlo bitno ispravno postaviti *Honeypot* sustav kako bi zaštitili računalnu mrežu jer u protivnom to može još i više narušiti sigurnost sustava.

U ovom radu detaljno je opisano podešavanje parametara i simulacija napada na *Honeypot* sustave. Korišteni su *Valhala Honeypot* i *Bifrozt Honeypot* alati. *Valhala Honeypot* predstavlja nisko interaktivni sustav za privlačenje i detekciju napadača. *Valhala* osim simuliranih usluga koristi i nekoliko „pravih“ odnosno produkcijskih usluga, stoga ga možemo smjestiti i u srednje interaktivne *Honeypot* sustave. *Valhala honeypot* sustav uspješan je u prikupljanju informacija o samom napadu na sustav kao što su IP adresa napadača i port odnosno usluga kojoj pristupa napadač što je vidljivo iz logova sustava. Iz poglavlja 'Pristup *web* poslužitelju' vidljivo je da je sustav u mogućnosti zavarati napadača tako što lažnu *web* stranicu prikaže kao legitimnu. Za razliku od *Valhala Honeypota*, *Bifrozt* je visoko interaktivni *Honeypot* što znači da napadač ima više mogućnosti u istraživanju sustava. Administrator sustava je u mogućnosti prikupiti veliku količinu podataka o samom napadaču, vektorima napada i alatima koje koristi kako bi zadobio pristup sustavu što je vidljivo u logovima sustava. Iako je još u razvoju, *Bifrozt* sustav se pokazao kao odličan primjer u zavaravanju napadača koristeći *spoofing* tehniku.

Valaha Honeypot i *Bifrozt* sustavi služe svrsi jer su zadovoljili prethodno postavljene ciljeve. Iako trenutna primjena *Honeypot* sustava leži u tome da budu napadnuti i istraženi kako bi administrator napadnute mreže prikupio što je više moguće informacija o napadu, u budućim istraživanjima bilo bi dobro istražiti mogu li *Honeypot* sustavi igrati aktivniju ulogu u obrani računalnih mreža. Osim što pasivno čekaju i prikupljaju informacije, *Honeypot* sustavi bi mogli zadobiti pravo pristupa na računalu samog napadača. Takav pristup otvara i neka moralna i etička pitanja. Sigurno je da bi takav pristup omogućio administratorima sustava prikupljanje još i više informacija o samom napadaču.

8. POPIS LITERATURE

[1] Hrvatska akademska i istraživačka mreža: *Računala mamci i ponašanje napadača*, str 1-10, 2008.

[2] Levy, S.: *Hackers: Heroes of the Computer Revolution - 25th Anniversary Edition*, O'Reilly Media, 2010.

[3] Clarks, Z., Clawson, J., Cordell, M.: *A brief history of hacking*, Historical Approaches to Digital Media, LCC 6316, str. 1 - 3, 2003.

[4] URL: https://nmap.org/ (pristupljeno: lipanj 2016.)

[5] Erickson, J.: *Hacking: The art of Exploatation, 2nd Edition*, No Starch Press, 2008.

[6] URL: http://computerethicsinstitute.org/publications/tencommandments.html (pristupljeno: travanj 2016.)

[7] Centar Informacijske Sigurnosti: *Botnet mreže*, broj 26, str 5 - 14, lipanj 2011.

[8] Bu, Z., Bueno, P., Kashyap, R., Wosotowsky, A.: *The New Era of Botnets*, McAfee, str 1 – 16, 2010.

[9] The *Honeynet* Project: *Know your Enemy: Tracking Botnets*, The *Honeynet* Project, 2005.

[10] Ramneek, P.: *Bots & Botnet: An Overview*, SANS Institute, str 1 – 18, 2003.

[11] Schiller, C. A., Binkley, J., Harley, D., Evron, G., Bradley, T., Willems, C., Cross, M.: *Botnets: The Killer Web*, Elsevier, str 77 – 95, 2007.

[12] Ollmann, G.: *Botnet Communication Topologies: Understanding the intricacies of botnet command-and-control*, Damballa Inc., str 1 – 4, 2009.

[13] Mokube, I., Adams, M.: *Honeypots: Concepts, Approaches, and Challenges*, str 321-322, 2007.

[14] Spitzner, L.: *Honeypots: Tracking Hackers*, Addison Wesley, 2002.

[15] URL: http://www.infosec.gov.hk/english/technical/files/*Honeypot*s.pdf (pristupljeno: svibanj 2016.)

[16] McGrew, R., Vaughn, R. B.: *Experiences With Honeypot Systems: Development, Deployment, and Analysis*, 39th Hawaii International Conference on System Sciences, str 1-3, 2006.

[17] Qassrawi, M. T., Zhang, H.: *Client Honeypots: Approaches and challenges, New Trends in Information Science and Service Science*, str 19 - 25, 2010.

[18] The Honeynet Project: *Know Your Enemy: Honeynets*, The Honeynet Project, str 1, 2006.

[19] Nazario, J.: *PhoneyC: A Virtual Client Honeypot*, Nazario, str 1 – 8, 2009.

[20] Gorzelak, K.: *Proactive detection of network security incidents*, European Uinion Agency for Network and Information Security (ENISA), str 17 – 19, 2011.

[21] Webb. J., *Network Demilitarized Zone*, ICTN 6870, str 2 – 3, 2015.

[22] Akkaya, D., Thalgott, F.: *Honeypots in Network Security*, Linnaeus University, str. 5, 2010.

[23] Information and Communications Security: 9th International Conference, Lecture Notes in Computer Science, Springer, str 462 – 463, 2008.

[24] Kobus, J.: *A 2nd Generation Honeynet – Introduction, Ingredients, Setup, Deployment and Brief Results*, str 1-9, 2003.

[25] Wonkyu, H., Ziming, Z., Adam, D., Gail-Joon, A.: *HoneyMix: Toward SDN-based Intelligent Honeynet*, Arizona State University, str 1, 2016.

[26] Grbavac, M.: *Sofverski dizajnirane mreže*, časopis Mreža, str. 65, listopad 2014.

[27] URL: http://www.downloadplex.com/Windows/System-Utilities/Operating-Systems/Specifications-*Valhala-Honeypot*_494927.html (pristupljeno: lipanj 2016.)

[28] URL: https://www.honeynet.org/node/1191 (pristupljeno: srpanj 2016.)

9. POPIS ILUSTRACIJA

sl. 3.1. Arhitektura Botnet mreže [7]

sl. 3.2. Arhitektura Botnet mreže s dva upravljačka poslužitelja [7]

sl. 3.3. Izvođenje DdoS napada pomoću Botnet mreže [7]

sl. 5.1. Prikaz honeynet sustava [18]

sl. 5.2. Honeynet prve generacije [14]

sl. 5.3. Honeynet druge generacije [14]

sl. 6.1. Predviđena mreža topologija

sl. 6.2. Stvarna logička topologija

sl. 6.3. Stvarna fizička topologija

sl. 6.4. Sučelje Valhala Honeypot sustava

sl. 6.5. Puštanje u rad Honeypot sustava

sl. 6.6. Glavni izbornik za podešavanje poslužitelja

sl. 6.7. Zenmap skeniranje portova

sl. 6.8. Napadač u interakciji s FTP poslužiteljem

sl. 6.9. Logovi spajanja na Valhala Honeypot FTP poslužitelj

sl. 6.10. Log datoteka u direktoriju izvršne datoteke Honeypota

sl. 6.11. Ograničen pristup Telnet poslužitelju

sl. 6.12. Logovi spajanja na Valhala Honeypot telnet poslužitelj

sl. 6.13. Pokušaj spajanja na SMTP poslužitelj

sl. 6.14. Logovi spajanja na Valhala Honeypot SMTP poslužitelj

sl. 6.15. Pokušaj spajanja na finger poslužitelj

sl. 6.16. Logovi spajanja na Valhala Honeypot finger poslužitelj

sl. 6.17. Spajanje na Web poslužitelj

sl. 6.18. Logovi spajanja na Valhala Honeypot Web poslužitelj

sl. 6.19. Testiranje POP3 poslužitelja

sl. 6.20. Općeniti prikaz Bifrozt Honeypot sustava [28]

sl. 6.21. Shema Bifrozt Honeypot mreže

sl. 6.22. Shema Bifrozt proxy sustava

sl. 6.23. Administrator pristupa Bifrozt proxyu

sl. 6.24. Napadač pristupa Honeypot računalu

sl. 6.25. Napadač skenira računalo

sl. 6.26. Zabilježen pokušaj autorizacije od strane napadača

sl. 6.27. Users.cfg datoteka u kojoj su smješteni korisničko ime i Spoofing zaporke

sl. 6.28. Napadač dobiva ograničen pristup sustavu

sl. 6.29. Pokušaj dobivanja root/administratorskih privilegija

sl. 6.30. Napadač stavlja maliciozni sadržaj na Honeypot

sl. 6.31. Zabilježeni logovi prijenosa datoteke na Honeypot

sl. 6.32. Analiza malicioznog koda