Botnet

Olika former av fenomenet och att skydda sig mot det

ErikÅhlgren

Erik Åhlgren

Examensarbete

Informationsteknik 2016

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Informationsteknik
Identifikationsnummer:	13396
Författare:	Erik Åhlgren
Arbetets namn:	Botnet olika former av fenomenet och att skydda sig mot det
Handledare (Arcada):	Jonny Karlsson
Uppdragsgivare:	

Sammandrag:
Arbetet är en litteraturstudie som handlar om hur man kan skydda sig mot botnetattacker. En bot är ett skadligt program som installerar sig själv på en dåligt skyddad dator genom att utnyttja de sårbarheter som finns. Olika skyddsmetoder beskrivs och deras förmåga att skydda mot olika slags attacker utvärderas. De viktigaste metoderna som gås igenom är: Honeynet/honeypot, Snort IDS, Oövervakad IDS för Zero-day attacker, datautvinningbaserad detektering av botnettrafik i nätverksflöden och Collaborative pattern baserade filtrerande algoritmer för att upptäcka botnet. Resultatet av examensarbetet är en kritisk jämförelse mellan skyddsmetoder samt en översikt över vilka metoder som rekommenderas för att skydda sej mot olika slags botnetattacker. Avslutningsvis ges förslag på vidareutveckling och framtida forskning inom området.

Nyckelord:	**Bot, Botnet, IDS, Skyddsmetoder, Botnet-attacker, Datasäkerhet.**
Sidantal:	43
Språk:	Svenska
Datum för godkännande:	27.12.2016

DEGREE THESIS	
Arcada	
Degree Programme:	Information technology
Identification number:	13396
Author:	Erik Åhlgren
Title:	Botnet: Different types of the phenomenon and protection methods
Supervisor (Arcada):	Jonny Karlsson
Commissioned by:	

Abstract:
This thesis is a litterature review on botnet attack detection. A bot is a malicious program that installs itself on a poorly protected computer by exploiting the existing vulnerabilities. Various protection methods are described and their ability to protect against different types of attacks are evaluated. The methods which are reviewed are: Honeynet/honeypot, Snort ids, Unsupervised network IDS for zero-day attacks, Mining-based detection of botnet traffic in network flow and Collaborative pattern-based filtering algoritm to detect botnets. The result of the thesis is a critical comparison between the protection methods, as well as an overview of the methods recommended to protect against various botnet attacks. Finally suggestions are made for further development and future research.

Keywords:	Bot, Botnet, IDS, Protection methods, Botnet-attacks, Data security.
Number of pages:	43
Language:	Swedish
Date of acceptance:	27.12.2016

INNEHÅLL / CONTENTS

Figurer

Tabeller

Förkortningar

C & C - Command and control
DDoS - Distributed deniel of service
DDNS - Dynamic domain name system
IP - Internet protocol
HTTP - Hyper text transfer protocol

FTP - File transfer protocol

P2P - peer to peer

IRC - internet relay chat

DNS - Domain name system

IDS - Intrusion detection system

ML - Machine learning

CGI - Common gateway interface

TCP - Transmission control protocol

UDP - User datagram protocol

ICMP - Internet control message protocol

SMTP - Simple mail transfer protocol

UNIDS – Unsupervised network intrusion detection system

CPBFA - Collaborative pattern-based filtering algoritm

1 INLEDNING

En bot är ett skadligt program som tar en dåligt skyddad dator under kontroll och får datorn att göra det boten vill. Efter att datorn är ombildad, tillägger programmet datorn till ett nätverk av datorer. (Bhatia et al. 2011 s. 178; Nagendra et al. 2014 s. 553)

Även i Finland har attacker utförsts nyligen. Riksdagens och försvarsministerits Internetsidor fälldes för några timmar i mars 2016 (Salokorpi 2016, Nieminen 2016). Också finska banker såsom Nordea och OP har blivit drabbade. (Yle 2015)

I examensarbetet, som baserar sig på kvalitativ forskning, görs en genomgång av botnetattacker och hur man skyddar sig mot dem. Tanken med arbetet är att kartlägga vilka olika skyddsmetoder det finns och att göra en kritisk jämförelse av vilken metod som fungerar bäst. Syftet med arbetet är att kartlägga olika former av botnetattacker och vidare utreda olika metoder som skyddar mot dem. Utvärderingen baserar sig enbart på litteraturstudier och inga egna test har utförts.

Temat är aktuellt eftersom botnet-attacker är vanliga i dagens samhälle. Jag vill redan i inledningen beskriva en rapport av Kaspersky om hur botnet-attacker har utvecklats år 2016 kvartal 1. Statistiken är tagen av Kasperskys egna ”DDoS Intelligence”:

Dataresurser i 74 olika länder har blivit attackerade av DDoS. Enligt (Kaspersky 2016) har botnet-attacker under första kvartalet 2016 utvecklats märkbart. Största delen av de attackerade resurserna ligger i 10 olika länder. Kina, Syd Korea och Amerika var de mest attackerade länderna. Över 70% av DDoS attackerna var 4 timmar långa och den längsta attacken var 8 dagar lång (år 2015 var den längsta attacken 2 veckor). Data som Kaspersky har samlat visar att trenden i attackerna är att attacktiden har förkortats och att frekvensen av attacker har ökat. Själva attackerna är mera komplexa.

Examensarbetet är strukturerat enligt följande:

Kapitel 2 handlar om metodik och kvalitativ forskning. Kapitel 3 handlar om vad botnet är och vilka typer av attacker det finns. I kapitel 4 kartlägger jag alla de skyddsmetoder som jag har valt och som nämns i många olika artiklar eller som är nya och fungerande. I kapitel 5 går jag igenom resultaten för de skyddsmetoder som listats i kapitel 4. I kapitel 6 görs slutsatser.

2 METODIK

Arbetet är en litteratursudie om fenomenet botnet och olika metoder att skydda sig mot botnetattacker. De skyddsmetoderna som beskrivs i arbetet är indelade i fem kategorier. En metod i varje kategori har valts för närmare granskning. De är valda p.g.a att de är nämda i många artiklar och att de har konstaterats vara framgångsrika. Dessutom har skyddsmetoder nämnts, ur olika artiklar, som faller inom de olika kategorierna.

2.1.1 Allmänt om kvalitativ forskning

Jouni Tuomi (2009 s. 94) nämner att Kvalitativa forskningar har ett brett spektrum av beskrivningar av hur forskningsanalys har genomförts. I olika metodhandböcker finns det också många olika beskrivningar av analyser. Forskaren Timo Laine (Jyväskylä universitet, Institutionen för filosofi) presenterade en kvalitativ ram för att illustrera utvecklingen av analysen:

1. Man skall göra ett beslut om vad som är intresserande av material som du läser och det är viktigt att göra ett starkt beslut.
2. a. Gå igenom materialet, separera och markera de saker som ingår i ditt intresse.
   b. Allt annat lämnas bort av denna forskning.
   c. Samla ihop markerade material du behöver och separera det från resten av materialet.
3. Kategorisera och gör ett tema.
4. Skriv en sammanfattning.

2.1.2 Innehållsanalys

I en innehållsanalys försöker man ur materialet skapa en teoretisk helhet. Materialet väljs utgående från forskarens uppgift och själva forskningens mening. Grunden ligger i att analysenheterna inte är förutbestämda utan skapas från materialet. (Tuomi 2009, 97)

Innehållsanalys som metod kan användas för att analysera dokument på ett systematiskt och objektivt sätt. Dokument kan vara bl.a. böcker, artiklar, intervjuer eller rapporter. Man kan säga att all skriven dokumentering kan analyseras med innehållsanalys. Ana-

lysmetoden fungerar också på ostrukturerat material, för meningen med metoden är att beskriva kort och på ett allmänt sett olika forskade fenomen. (Tuomi 2009 s. 195)

3 BOTNET ÖVERSIKT

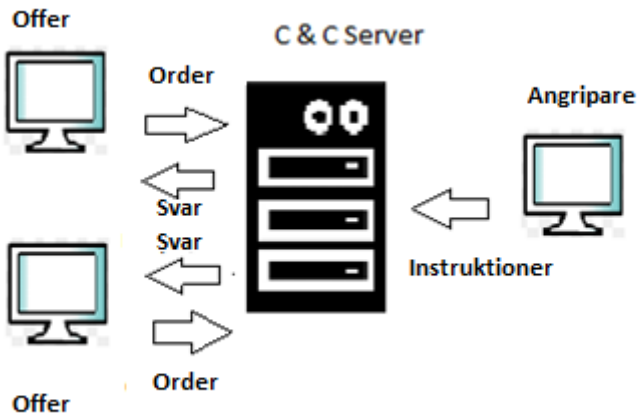
3.1.1 Bot

En bot är ett skadligt program som tar en dåligt skyddad dator under kontroll och får datorn att göra det boten vill. Efter att datorn är ombildad, tillägger programmet datorn till ett nätverk av datorer. (Bhatia et al. 2011 s. 178; Nagendra et al. 2014 s. 553)

3.1.2 Botnet

Ett botnet är ett nätverk av datorer som övervakas av en botmaster. En botmaster använder botar för att övervaka infekterade datorer och öka antalet datorer i nätverket. Botmastern styr botnetet genom kommandon och en kontrollmekanism (C & C), mekanismen kallas för C & C-server. (Bhatia et al. 2011 s. 178; Nagendra et al. 2014 s. 553).

Nedan illustreras kommunikationsflödet i ett botnet. (Se figur 1)



Figur 1. Kommunikationsflödet i botnet. (Bhatia et al. 2011)

3.1.3 **Botnet-fenomenet**

Botnet andvänds för att sprida attacker via Internet, såsom:

- DdoS (Distributed denial of Service)
- Spam
- Spionprogram
- Trojansk häst
- Andra olika skadeprogram

Botnet är ett nätverk av infekterade maskiner, bots. Genom botnetattacker övertas datorer, utan att användaren själv vet om saken. Själva processen utförs av en central enhet "C&C" eller botmaster. (Karim et al. 2014 s. 946)

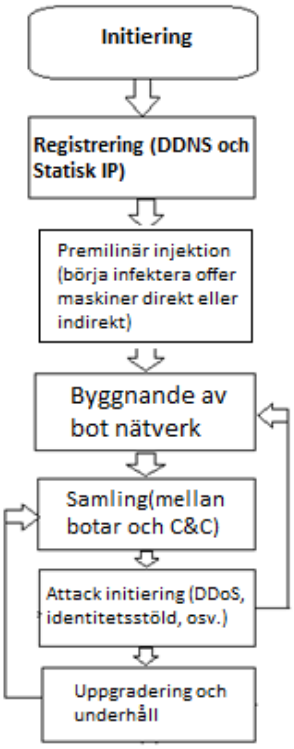
Botmasterns vill öka antalet av botar i nätverket. Med ett stort antal av botar är det möjligt att göra attackerna stora och förstöra mycket. Skillnaden mellan ett botnet och en normal attack är att "C&C" skickar instruktioner till maskinerna, d.v.s botarna får instruktioner av botmastern om vad de skall göra. T.ex. att sätta igång en mask eller att skicka spam. (Karim et al. 2014 s. 946)

Botmasterns uppgift är att meddela botarna när attackerna skall göras. (Karim et al. 2014 s. 946)

3.1.4 **Livscykel för botnet**

Initiering är det första steget när ett botnet uppbyggs. I initieringsstadiet sätter botmastern parametrar för att kunna kommunicera med botar. Efter initiering kommer registreringsprocessen. Registreringsprocessen sker mellan botmastern och DDNS (Dynamic Domain Name System). En statisk IP address för botmastern grundas. Därefter sker insprutningen, som kan utföras på många olika sätt, såsom via virus som nedladdar oönskade laddningar eller via epost. (Karim et al. 2014 s. 947)

Botarna bygger sina nätverk genom att installera en illvillig kod. Den infekterade maskinen utför sökningar och skadlig programvara installeras. Efter nedladdningen har maskinen blivit en bot. Nedladdningen sker vanligtvis via HTTP (Hyper Text Transfer Protocol), FTP (File Transfer Protocol) eller P2P – Protokoll. Figur 2 nedan beskriver Botnets liscykel. (Karim et al. 2014 s. 947)



Figur 2. Botnet flödes diagram. (Karim et al. 2014)

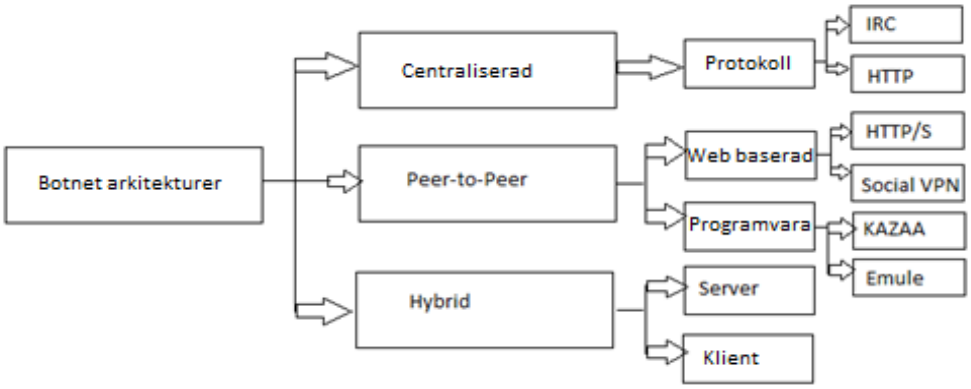
Nästa steg heter gruppering. I detta steg bygger man en förbindelse mellan botarna och "C&C". Steget håller också upp förbindelsen även om boten startar om d.v.s förbindelsen tappas inte fast boten är avkopplad. Grupperingssteget är därför en kontinuerlig process. (Karim et al. 2014 s. 948)

När förbindelsen mellan boten och C&C är gjord, kommer attack-steget. I attack-steget väntar botarna på kommandon från C&C. Botmasterns syfte med kommandon är att göra aktiviteter såsom DDoS attacker, analysering av nätverkstrafik, identitetsstöld, sökning av sårbarheter på datorn o.s.v. Det finns många olika saker som en botnetattack kan göra. (Karim et al. 2014 s. 948; Massi et al. 2010 s. 5)

Sista steget heter uppgradering och underhåll. Botmastern skickar en ny kod till botarna så att deras beteende förändras. Skyddsmekanismerna kan inte då spåra botarna. Det här är det mest riskfyllda steget för botnetet. (Karim et al. 2014 s. 948)

3.1.5 **Botnetarkitektur**

Flexibilitet är nätverkets styrka. I nätverket kan det finnas tusentals datorer som är fjärrstyrda. En fördel i nätverket är också det att man kan lösa kommunikationsproblem mellan enheterna på många olika sätt. Ett botnet kan uppbyggas på olika sätt. (Se figur 3) (Karim et al. 2014 s. 948)



Figur 3. Taxonomi av botnet arkitekturer. (Karim et al. 2014)

Central C&C arkitektur: Centraliserad C&C liknar den vanliga klient/server arkitekturen. IRC protokollet är ett exempel på en centraliserad arkitektur där botarna grundar en stark förbindelse mellan många olika enheter. IRC och HTTP protokollen är de mest använda protokollen i den centrilaserade arkitekturen. Fördelar med den centrilaserade arkitekturen:

1. Distribuering
2. Behöver ingen speciell hårdvara
3. Responstiden mellan botarna och servern
4. Botmätaren är i direkt “kontakt’ med botarna.
5. Botarna får snabbt updatering av botmastern

Nackdelen med en centraliserad “C&C” arkitektur är att det är lätt att stänga upptäckta botnet. (Karim et al. 2014 s. 948-949; Ghafir et al. 2015 s.76)

Peer-to-peer (P2P) arkitektur: I en decentraliserad arkitekur kan botneten lättare skaffa stora mängder av botar. Det är svårt att undvika decentraliserade botnet för att:

1. Stänga en P2P botnet skall man först hitta ett flertal botar. Det är mycket svårt att fånga flera botar i en P2P arkitektur.
2. P2P botnet har inget centraliserat nätverk. Det är mycket svårt att bestämma området som är infekterat av botnetet.
3. Det är svårt att abryta en P2P förbindelse för att botarna är inte beroende av varandra.

Förbindelsen, i ett P2P botnet, är mycket enkel. Botmastern skickar ett kommando till en "peer" bot och boten skickar kommandot till andra "peer" botar. Hanteringen av ett P2P botnet är svårare än hanteringen av en centraliserad arkitektur. (Karim et al. 2014 s. 949; Ghafir et al. 2015 s.76)

3.1.6 Hybrid C&C arkitektur

En hybrid arkitektur är en blandning av Centraliserad och P2P arkitektur.

Hybridmodellen delas i två grupper: tjänare bot och klient bot. Tjänare botens roll är att vara samtidigt en kund och en server. Tjänare boten har statiska IP addresser och därför lyssnar klient boten inte på inkommande anslutningar/trafik som är kopplad ihop med en dynamisk IP. En tjänare bot skickar IP addresser/information till en lista (peer list) och lyssnar på inkommande trafik. (Karim et al. 2014 s. 949; Ghafir et al. 2015 s.76)

3.1.7 Typisk klassificering

IRC Botnet: Botmastern fungerar som IRC-server och använder IRC-kanaler för att skicka kommandon till botnetet. Alla i botnetet är anslutna till kanalen. Kommandon skickas till deltagarna med hjälp av ett allmänt IRC-protokoll. (Bhatia et al. 2011 s. 178)

HTTP Botnet: Botmastern fungerar som en webbserver och botarna är anslutna till webbservern. Kommandon är inkapslade i HTTP-meddelanden. (Bhatia et al. 2011 s. 178)

P2P Botnet: Nyare typer av botnet som använder befintliga P2P protokoll för att sprida kommandon. Denna typ av botnet är svårare att upptäcka än de andra botneten. (Bhatia et al. 2011 s. 178; Nagendra et al. 2014 s. 554)

4 SKYDDSMETODER MOT BOTNET

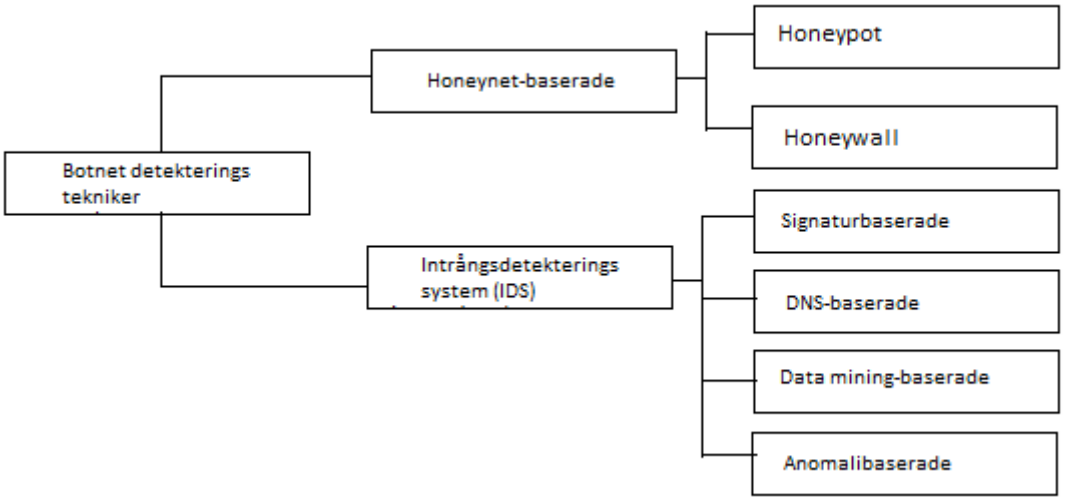
4.1.1 Allmänt om botnet-detekteringstekniker

Nya typer av botnet använder tekniker för att inte bli upptäckta av detekteringssystem. Detta är en utmaning för många säkerhetsexperter och därför har botnet- detektionstekniker varit mycket under forskning. (Limarunothai et al. 2015 s. 54)

Botnetdetektionsteknikerna är uppdelade i två huvudsakliga tekniker: den Honeynet baserade tekniken och intrångshanteringssystem (IDS). Honeynet används för att förstå botnets aktiviteter och karaktär. IDS tekniken används för att upptäcka botnet. IDS tekniken kan delas in i fyra kategorier, som beskrivs i figur 4:

- Den signaturbaserade
- Den anomalibaserade
- Den DNS-baserade
- Datautvinningsbaserade (Data mining-baserade)

(Limarunothai et al. 2015 s. 54; Amini et al. 2014 s. 140)



Figur 4. Botnet upptäknings tekniker. (Limarunothai et al. 2015)

4.1.2 Den signaturbaserade tekniken

Den signaturbaserade tekniken upptäcker botnet genom att iaktaga nätverksflöden och genom att hitta mönster som liknar redan existerande. Fördelen med denna teknik är att den upptäcker kända botnet och kräver låg resurs för att behandla. Den signaturbaserade tekniken kan bara upptäcka kända botnet. Informationen om kända botnet är lagrad i en signaturdatabas. (Limarunothai et al. 2015 s. 54-55; Vania et al. 2013 s 27)

I arbetet beskriver jag Snort IDS-tekniken i detalj i kap. 4.3. Snort nämns i många olika articklar och många metoder använder Snort som en del av en större skyddshelhet.

4.1.3 Den anomalibaserade tekniken

Den anomali-baserade tekniken fungerar genom att övervaka nätverkstrafik och genom att klassificera trafiken som normal eller onormal. Denna teknik kan iaktta ovanliga nätverkstrafiksaktiviteter såsom:

- Höga trafikvolymer
- Hög nätverksfördröjning
- Trafik på ovanliga portar
- Ovanligt systembeteende

Den anomali-baserade tekniken består av två steg. Första steget kallas för träningsfas, då skapar tekniken en normal trafikprofil. Andra steget kallas för anomalidetektering. Här jämförs den normala trafikprofilen med den nuvarande trafiken. Meningen är att hitta avvikelser i trafiken. Nyttan av metod är att den kan känna igen okända nya botnet. (Limarunothai et al. 2015 s. 55; Vania et al. 2013 s 27)

I arbetet beskriver jag UNIDS-tekniken i detalj i kap. 4.4. Metoden är ny och har visat sig vara framgångsrik metod. Andra anomali-baserade tekniker är t.ex. Anomaly based IDS using Backpropagation Neural Network. (Mane et al. 2016 s. 29) och FCAAIS: Anomaly based network intrusion detection through feature correlation analysis and association impact scale. (Jyothsna et al. 2016 s. 103)

4.1.4 Den DNS-baserade tekniken

Den DNS-baserade detektionstekniken liknar den anomali-baserade tekniken. Tekniken baserar sig på DNS-information som frambrings av botnetet. Botnetet måste utföra en DNS-förfrågan för att lokalisera en viss C&C server som är värd för en DDNS (Dynamic domain name sytem) leverantör. Man kan upptäcka botnet genom att övervaka avvikelser i DNS trafiken. När en förfrågan blir upptäckt, kan botnetet upptäckas. (Limarunothai et al. 2015 s. 55; Vania et al. 2013 s 27)

Med hjälp av den DNS-baserade tekniken kan man upptäcka botnet, men det är också möjligt att hitta botmasterns adress. Om botmastern använder få DNS-frågor, då är det svårt att upptäcka adressen. (Limarunothai et al. 2015 s. 55)

I arbetet beskriver jag närmare CPBFA-tekniken i kap. 4.6. Metoden är ny och framgångsrik. I artikeln Botnet Detection through DNS based approach beskrivs en annan teknik av DNS-baserad detektion. (Dange et al. 2013 s 497)

4.1.5 Den datautvinningsbaserade tekniken

Det är svårt att skilja laglig och godartad trafik från skadlig trafik. Botneten har utvecklats för att gömma sig från detektionssystem. Botnetet använder normala protokoll för C&C kommunikation och trafiken är mycket nära godartad trafik. Det är svårt att upptäcka godartad trafik med en anomali-baserad detektionsteknik och därför behöver vi flera Data datautvinningbaserade detektionstekniker såsom ML. (machine learning). (Limarunothai et al. 2015 s. 55; Vania et al. 2013 s 27)

I arbetet beskriver jag i detalj datautvinningbaserad detektion av botnettrafik i nätverksflöden i kap. 4.5. Metoden är ny och förutses bli framgångsrik. En anna datautvinningbaserad teknik är ”Soft-Man” and Data Mining based Distributed Intrusion Detection System. (Zheng 2016 s. 145)

4.2 Honeypot

Honeypot är en datorresurs vars uppgift är att ta emot attacker i stället för det nätverk som man vill skydda. För attakeraren förefaller Honeypot att vara den dator man vill attackera. Honeypot kan köra olika operativsystem och ett antal olika tjänster, därför fungerar Honeypot som en bra fälla. (Bhatia et al. 2011 s. 180; Ghafir et al. 2015 s.77)

4.2.1 Research honeypots

En research honeypot andvänds för att få data om hackern. Informationen som experterna får via research honeypoten, andvänds för att förutse attacker. Med informationen kan experten förbättra intrångsdetektionen. Informationen är också viktig för att utveckla bättre säkerhetssystem. (Sharma 2013 s. 7)

4.2.2 Production honeypots

Production honeypots är en viktig del av företags- och industrinätverkssäkerhet. Honeypots fungerar som ett tidigt varningssystem, så att information om hot blir snabbt upptäckta. Production honeypots ger information om hotet till administratorn före attacken. (Sharma 2013 s. 7)

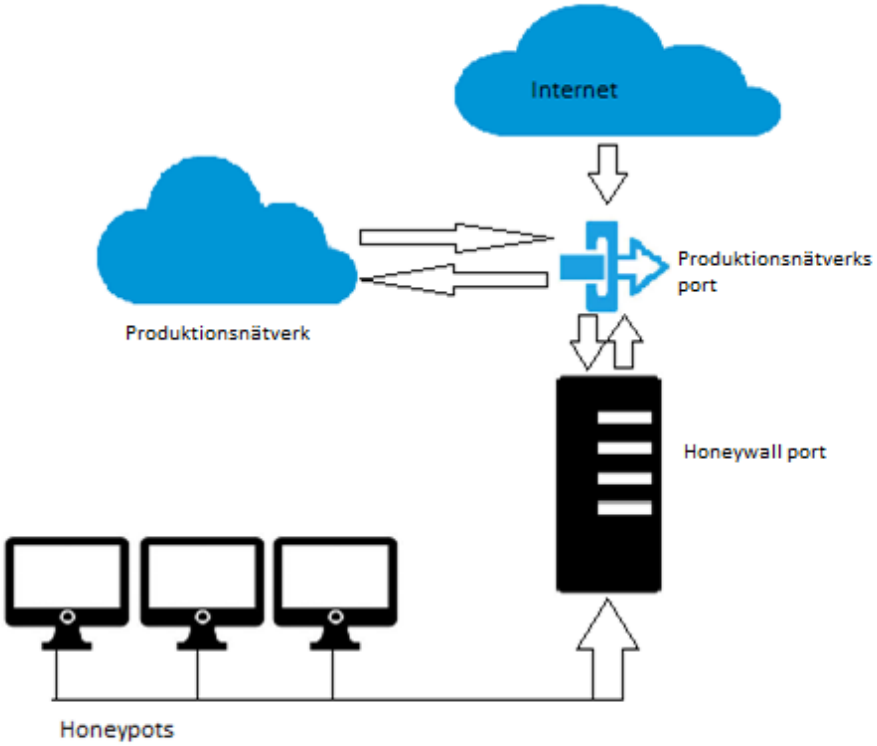
Det finns två olika typer av honeypots: låg nivå interaktion och hög nivå interaktion.

Honeypots med låg nivå interaktion efterliknar tjänster och en fullständig tillgång till honeypot får attakeraren inte. Dessa honeypots är lätta att identifiera av attakeraren genom ett simpelt kommando. (Sharma 2013 s. 7; Bhatia et al. 2011 s. 180; Nikkhahan et al. 2009 s. 78)

Hög nivå interaktion ger anfallaren ett system att kommunicera med. Anfallaren får fullständig tillgång till systemet och därefter kan denna avfyra attacker till nätverket. Hög nivå interaktion ger information om anfallaren och det uppgörs register om vad anfallaren har gjort (aktiviteter och aktioner). (Sharma 2013 s. 7; Bhatia et al. 2011 s. 180; Nikkhahan et al. 2009 s. 78)

4.2.3 **Honeynet**

För att bilda ett honeynet behövs det två eller fler honeypots (Se figur 5). Honeynet används för att övervaka ett större eller stora nätverk. Man behöver mera än en honeypot för att övervaka ett stort nätverk. Honeypot och honeynet är vanligtvis en del av större intrångsdetekteringssystem. (Sharma 2013 s. 8)



Figur 5. Honeynet arkitektur. (Karim et al. 2014)

4.2.4 **Honeypots målsättnig**

Huvudsyftet med honeypots är att spåra attackeraren genom ”connection tracking” eller “pattern flow detection” teknik. Idén är att få attakeraren att tro att informationen som man försöker stjäla är från den riktiga användaren. Honeypot fungerar som en kopia av den riktiga datorn och själva informationen är felaktig, d.v.s. idén med honeypot är att störa anfallaren. (Sharma 2013 s. 8; Nikkhahan et al. 2009 s. 77)

Produktionshoneypots används för att hjälpa minska risker och för att störa attakeraren från att anfalla själva produktionssystemet. (Sharma 2013 s. 8)

Forskningshoneypot används för att samla information och bevis om anfallaren. Forskningshoneypot skyddar inte nätverket, men ger tillgång till mycket information om anfallarens verktyg och vilka hackningstekniker anfallaren använder. (Sharma 2013 s. 8; Ghafir et al. 2015 s.77)

4.2.5 Nätverks lockbete

Honeypots är nyttiga för att övervaka nätverk. Honeypots är utplacerade så att de inte är i själva produktionsnätverket. När attacker händer, kommer en del av trafiken att hamna ut för honeypots. Systemet är uppbyggt så att den normala trafiken inte kommer att stöta på honeypots och därför är varningarna som kommer från honeypots riktiga och pålitliga. Man kan också använda honeypots som lockbeten, anfallarna blir störda och de vet inte om nätverket är värdefullt. (Sharma 2013 s. 9)

4.2.6 Insamling av skadeprogram

Honeypot kan automatiskt samla in information om skadeprogram som sprider sig självständigt. Det är bra att ha data om aktiva skadliga koder. Med informationen som man fått från skadeprogrammet kan man förbättra intrångsdetektionen eller antivirusprogrammet. Uppsamlingshoneypot laddar ner det aktuella skadeprogrammet och skriva ner händelsen om tillräckligt information fås. Låg nivå interaction honeypot kan fånga en skadlig kod som utnyttjar kända sårbarheter, eftersom de är beroende av simulering. Mer omfattande kapningar kräver en hög nivå interaction honeypot, med ett riktigt operativsystem. (Sharma 2013 s. 9)

4.2.7 Nackdelar med honeypot

Det finns flera viktiga fördelar med att använda honeypots, men det finns också nackdelar:

- Om hackern inte attackerar honeypot, då är det inte möjligt att fånga information.

- Det finns en nackdel med fingeravtryck. Fingeravtryck är att en attackerare kan identifiera en honeypot, eftersom de har vissa förväntade egenskaper eller beteenden. Det är lätt för en erfaren hacker att förstå att han attackerar en honeypot.
- Honeypot kan användas som en zombie och då är systemet under hot.

Honeypot används främst för att få information om skadliga program inte för att vara ett skydd mot dem. Information behövs för att man kan vara förberedd mot attacker. (Sharma 2013 s. 11)

4.3 Snort IDS

Snort är ett system med en öppen källkod IDS (Intrusion detection system) som fungerar som en paketsniffare som övervakar nätverkstrafik i realtid. Snort granskar paketen för att upptäcka farliga och misstänksamma avvikelser. Snort baserar sig på libcap (library packet capture) som är ett verktyg som man använder för att analysera TCP/IP trafik. (Chanthakoummane et al. 2015 s. 88; Csubák et al. 2016 S.54)

Snort-IDS kan upptäcka följande attacker:

- DOS attacker (deniel of service)
- Buffertöverskridning
- CGI attacker
- Portskanning
- SMB attacker (Server message block)

När snort upptäcker ett misstänksamt beteende, skickar den ett meddelande till syslog (system log) som en separat fil. (Chanthakoummane et al. 2015 s. 88; Roesch 1999 S. 229)

Snort har testats av en grupp som heter NSS group (en organization som testar nätverkssäkerhet). Även om snort har en öppen källkods IDS, var den enligt NSS klart en av de bästa systemen för intrångsdetektering. Snort har även testats av stora företag (Cisco och Symantec o.s.v.) (Chanthakoummane et al. 2015 s. 88)

Snort är uppdelad i flera olika komponenter. Komponenterna arbetar tillsammans för att upptäcka särskilda attacker och för att föra fram data i sådan form som användaren vill ha. (Chanthakoummane et al. 2015 s. 88)

4.3.1 Paketavkodare

Paketavkodaren tar paket från olika typer av nätverksgränssnitt och förbereder paketen för förbehandling. Om paketen inte förbehandlas skickar avkodaren paketen direkt till detekteringsmotorn. (Chanthakoummane et al. 2015 s. 88)

4.3.2 För-processorer

För-processorer är komponenter som kan användas med Snort för att arrangera datapaket före detekteringsmotorn upptäcker att paketen är infekterade eller används av intrångaren. (Chanthakoummane et al. 2015 s. 88)

4.3.3 Detekteringmotor

Detekteringsmotorn är den viktigaste delen av Snort. Motorns uppgift är att upptäcka intrångsaktiviteter som finns i paketen. (Chanthakoummane et al. 2015 s. 88)

4.3.4 Logging och alarmsystem

Om detekteringsmotorn hittar misstänksam data i ett paket, kan informationen loggas eller användas för att frambringa en varning (syslog eller winalert). (Chanthakoummane et al. 2015 s. 88)

4.3.5 Utmatningsmoduler

Utmatningsmoduler kan göra olika operationer beroende på hur man vill spara utmatningarna som frambings av snorts logging och alarmsystemet. Regelrubriken och regelalternativen fungerar tillsammans, detta illustreras i figur 6. (Chanthakoummane et al. 2015 s. 89)

Figur 6. Struktur av Snort Regler. (Chanthakoummane et al. 2015)

4.3.6 Utvärdering av Snort-IDS regler för botnet detektering

Regelrubriken beskriver egenskaper som ett paket har och informerar vad snort skall göra om paketet liknar regeln. Regelalternativen följer den information som regelrubriken ger, t.ex. ett varningsmeddelande. (Chanthakoummane et al. 2015 s. 89; Csubák et al. 2016 S.54)

4.3.7 Bothunter och Botsniffer

Bothunter är ett passivt övervakningsnätverk som drivs av snort. Den sätter i samband inkommande inbrottsalarm med utbundna kommunikationsmodeller som ger tecken på att den lokala värden är infekterad. (Behal et al. 2010 s. 2)

Botsniffer grundar sig på två huvundkomponter, monitor "motorn" och korrelations "motorn". Monitormotorns uppgift är att undersöka nätverkstrafik, och att frambringa anslutningsregister över misttänkta C&C protokoll. Korrelationsmotorn analyserar händelserna som monitormotorn har upptäkt. Den utför gruppanalyser om tid och rum samt om samband och likheter mellan aktivitetsbeteendet hos kunder som ansluter till samma IRC eller HTTP server. (Gu et al. 2008 s. 3-4)

Snort använder både bothunter och botsniffer. Snort jämför det framkomna värdnamnet med en svartlista. (Chanthakoummane et al. 2015 s. 89)

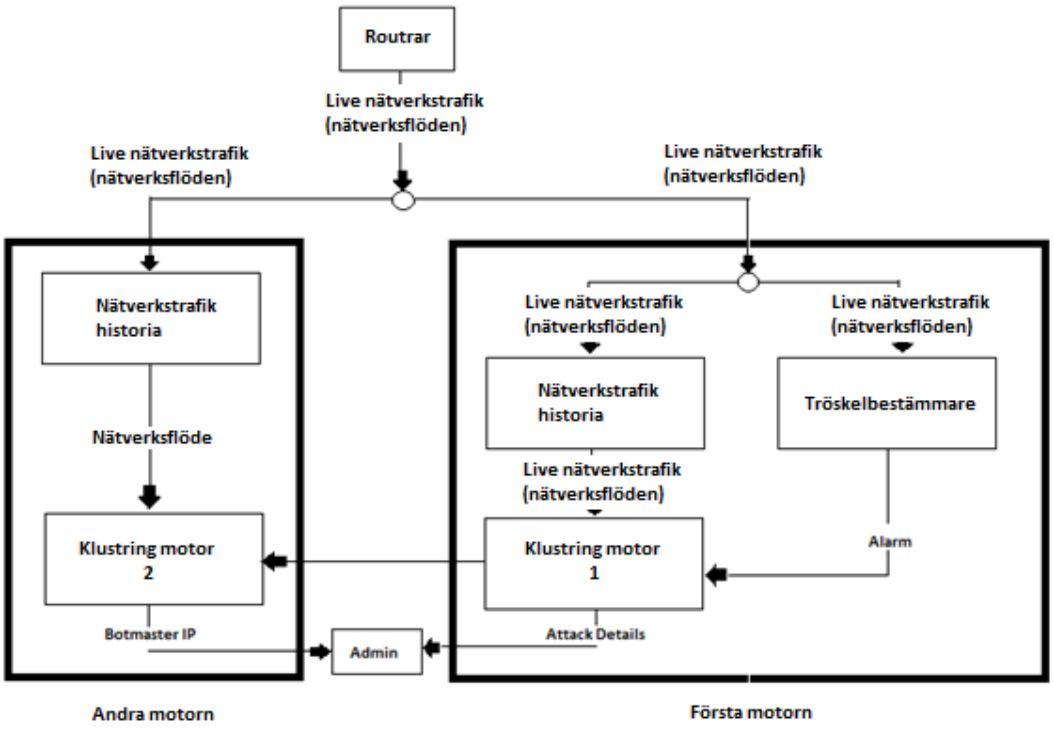
4.4 Oövervakad IDS för Zero-day attacker(UNIDS)

Tekniken är delad i två separata motorer. Den första motorn kontrollerar nätverksbeteende i realtid, för att upptäcka snabbt spridande attacker såsom:

- DoS
- DDoS
- Skanning och maskutbredning

Den andra motorn (botnet detekteringsmotorn) behöver mera tid för att få tag på tillräckligt med information för att hitta den slutliga botmastern. (Amoli et al. 2016 s. 3)

Denna model använder NIDS (Network Intrusion Detection Systems) live nätverksflöden som en ingång (från router). Överföring av nätverksströmmar till NIDS är en lösning av nätverksövervakning. Paket-överföring i port-spegling producerar en stor mängd av nätverkstrafik och orsakar en flaskhals under nätverskattacker. Arkitekturen av NIDS illustreras i figur 7. (Amoli et al. 2016 s. 3)



Figur 7. Arkitekturen i NIDS. (Amoli et al. 2016)

4.4.1 **Första motorn (Dynamisk självanpassningbar tröskelvärde)**

Den första motorn övervakar beteendet i nätverket genom en automatiserad självanpassande tröskel för att upptäcka förändringar som kan orsakas av intrång.
För att förhindra en hög mängd av falska alarm i en obalanserad nätverkstrafik, anpassar sig nätverkströskeln till det aktuella statuset av nätverket för att bestämma den normala volymen av nätverkstrafik i framtiden (nästa sekund). (Amoli et al. 2016 s. 4)

Ett högt nätverksflöde kan vara förväntat då antalet online-användare är stort, men det är onormalt om antalet maskiner i nätverket är litet. Eftersom antalet nätverksanvändare är vågliknande, kan förhållandet mellan utgående och inkommande nätverkflöden (per maskin) ger mera exakta uppgifter om den verkliga statusen för nätverket. (Amoli et al. 2016 s. 4)

4.4.2 **Första motorn (Klustring motor)**

När volymen av nätverksflöden passerar tröskeln, använder NIDS DBSCAN (Density-based spatial clustering of applications with noise) för att föra samman antalet inbundna och utgående nätverkflöden för varje maskin så att attackeraren/anfallaren kan spåras. För att öka noggrannheten, är nätverkstrafiken delad i två faser: träning (självlärande) och detektering. (Amoli et al. 2016 s. 5)

Under träningsfasen, städar NIDS-klustret nätverkstarfiken som överförs före alarmtröskeln så att det exakta avtåndet under fasdetekteringen kan fastslås. (Amoli et al. 2016 s. 5)

Nätverksintrång såsom DoS, DDoS, Skanning, Spam och snabbt spridande maskar för med sig ett stort antal av nätverksströmmar inom en kort tidsperiod, det är därför möjligt att upptäcka deras beteende som ljud/oljud (extremvärden). Första motorn skickar uppgifter till administratorn vid intrång. Uppgifterna visar antalet maskiner, portar och annan nyttig data för att klassificera attacken. (Amoli et al. 2016 s. 5)

4.4.3 Andra motorn (botnet detektions modul)

Andra motorns uppgift är att upptäcka interna botnet (botar eller botmaster). DDoS attacker är komponerade av botnet och andra motorn analyserar och för samman nätverkstrafiken före DDoS attacken kan finna den interna IP-adressen. Genom klustring är det möjligt att upptäcka ursprunglig botnetkommunikation som oljud när man jämför det med den övriga nätverkstrafiken. (Amoli et al. 2016 s. 6)

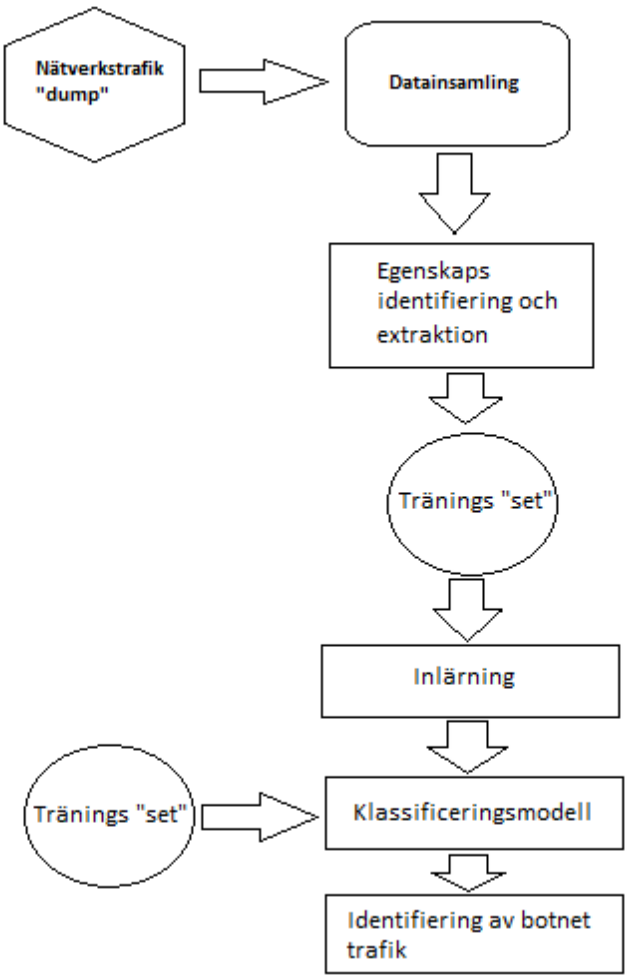
När den första motorn klassificerar en attack som DDoS, skickar den anfallarens/attackerarens IP-adress till andra motorn. Andra motorn analayserar möjligheten att finna botmastern. C&C kommunicerar med botar flera gånger under en bots livstid och därför uppgör den andra motorn en lista över IP-adresser. IP-adresserna har kontakt med attackeraren före en DDoS attack, darför anses denna vara en eventuell botmaster. (Amoli et al. 2016 s. 6)

4.5 Datautvinningsbaserad detektion av botnettrafik i nätverksflöden

Huvudsyftet med denna teknik är att förutsäga botnettrafik i nätverket med hjälp av maskininlärning (ML). Problemet med att upptäcka botnettrafik är att den utformas som en binär klassificieringsuppgift och löses med kraftfulla övervakade mönsterinlärningsalgoritmer (supervised pattern learning algorithms). ML teknikerna är effektivare än statistiska metoder, för ML teknikerna lär sig träningsdata (training data) genom att ta intelligenta tips från data och genom att förutse vad som kommer att hända. (Kalaivani et al. 2016 s. 536)

I denna teknik använder man CTU13 dataset (ett dataset som av botnettrafik som är fångats av CTU universitet i republiken Checkien). (Garcia et al. 2014 s.100) Klassificeringsmodellerna är byggda med hjälp av Support Vector machine (SVM), neurala nätverk, naïve bayes och decision tree. De här är alla ML klassificeringsmodeller. Nätverksadministratörer och kommunikationstjänsteleverantörer kan använda denna identifieringmodell för att skydda sina nätverk och sina användare från avancerada skadliga koder och bontnets hot. (Kalaivani et al. 2016 s. 536)

Denna modell består av fyra faser: Datainsamling, träning och testning, funktions-extraktion och klassificering. Arkitekturen (Figur 8) ser ut så här:



Figur 8. Blockschema för botnet trafikklassificering. (Kalaivani et al. 2016)

4.5.1 **Egenskapsextraktion**

Egenskapsextraktion spelar en viktig roll i byggandet av klassificerare. Tydliga egenskaper som hjälper att förutse trafikflödet i botnetet dras ut från CTU databasen. Datasset består av funktioner så som:

- Starttid
- Varaktigheten av bestämt flöde

- Käll- och denstinationsport som används för att ange tjänster som erbjuds av lokala- eller fjärrvärdar.
- Käll- och destinations IP-adresser som används av paket för att röra sig i nätverk
- Protokoll som specificerar samverkan mellan kommunicerande enheter
- ToS-fältet (type of service) används för att tilldela prioriteter för IP-paket och totala byte. (Kalaivani et al. 2016 s. 537)

Med endast dessa funktioner kan man inte skilja botnettrafik från normal trafik. Men med funtioner som:

- Genomsnittlig bytehastighet
- Genomsnittlig pakethastighet
- Ping byte
- Tidsjämförelse
- Skadliga portar som hjälper till att avgöra botnettrafik som identifieras och extraheras.

Kan man få hjälp i att identifiera botnettrafik. (Kalaivani et al. 2016 s. 537)

4.5.2 Flerlagrig perceptron

Flerlagrig perceptron-nätverk (MLP) är den mest utnyttjade nätverksklassificeraren. MLP är flexibel och består av icke-linjära modeller av ett antal enheter organiserade i flera lager. Det invecklade MLP nätverket kan ändras genom att variera antalet lager och antalet enheter i varje lager. (Kalaivani et al. 2016 s. 538)

MLP är ett värdefullt verktyg för att lösa problem, då man har litet eller ingen kunskap om förhållandet mellan ingångsvektorer och deras utmaningar. (Kalaivani et al. 2016 s. 538)

4.5.3 Beslutsträdsinduktion (decision tree induction)

Beslutsträdsklassificering framkallar data som liknar ett binärt träd och kallas därför för ett beslutsträd. Varje grennod representerar ett val mellan alternativ och varje lövnod

representerar en indelning eller ett beslut. Beslutsträdsmodellen innehåller regler för att förutspå målvariabler. Algoritmen klassiciferar bra, även om det finns olika mängder träningsexempel och många egenskaper i stora databaser. (Kalaivani et al. 2016 s. 538)

J48 algoritmen är en form av C4.5 beslutsträdslärande. Denna implementering producerar beslutsträdsmodellen. Algoritmen använder girig teknik för att införa ett beslutträd. En beslutsträdsmodell är byggd för att analysera träningsdata och modellen används för att gruppera osynlig data. (Kalaivani et al. 2016 s. 538)

4.5.4 **Naïve Bayes klassificiering**

Naïve Bayes klassificeraren (NB) är en enkel men effektiv klassificierare som används av många informationsbehandlingprogram. NB tekniken bygger sig på Bayes teori och är lämplig för att användas när data mängden är hög. NB klassificerare antar att effekten av ett variabelvärde på en given klass är oberoende av värdet av en annan variabel. Tilläggs sannoliketer matas in och metoden väljer den med det högsta värdet. (Kalaivani et al. 2016 s. 538-539)

4.5.5 **Stödvektormaskin (Support vector machine)**

SVM är en ny metod för övervakad mönsterklassificering. SVM har framgångrikt tillämpats på flere erkänningsproblem.
SVM är en utbildningsalgoritm för att få fram grupperings- och regressionsregler från data. SVM är mest lämpande för att arbeta effektivt i utrymmen med högintensiva egenskaper. Den bygger på starka matematiska grunder och resulterar i kraftfulla algoritmer. (Kalaivani et al. 2016 s. 539)

Standard SVM-algoritmen bygger på en binärklassificerare. Ett sätt att bygga en binär klassificerare är att konstruera ett hyperplan och avskilja klassmedlemar från icke- medlemmar i inmatningsutrymmet. (Kalaivani et al. 2016 s. 539)

4.6 Collaborative pattern baserade filtrerande algoritmer för att upptäcka botnet

Collaborative pattern-based filtrering (CPF) algoritmen upptäcker skadliga domäner och IP-adresser som används av botnet. Algoritmen är uppbyggd av en kombination av Case based reasoning (CBR) och Fuzzy mönsteridentifiering så att botnet upptäcks. (Panimalar et al. 2016 s. 155)

Fuzzy mönsteridentifiering filtreringsalgoritmen (FPRF) upptäcer bra realtidsbotnet. FPRF algoritmen försöker identifiera domämnamn och IP-adresser som används av bot C&C-servrar. FPRF är delad i tre olika faser som kallas: trafiksreduktion, funktionsextrahering och Fuzzy mönsteridentifiering. (Panimalar et al. 2016 s. 156)

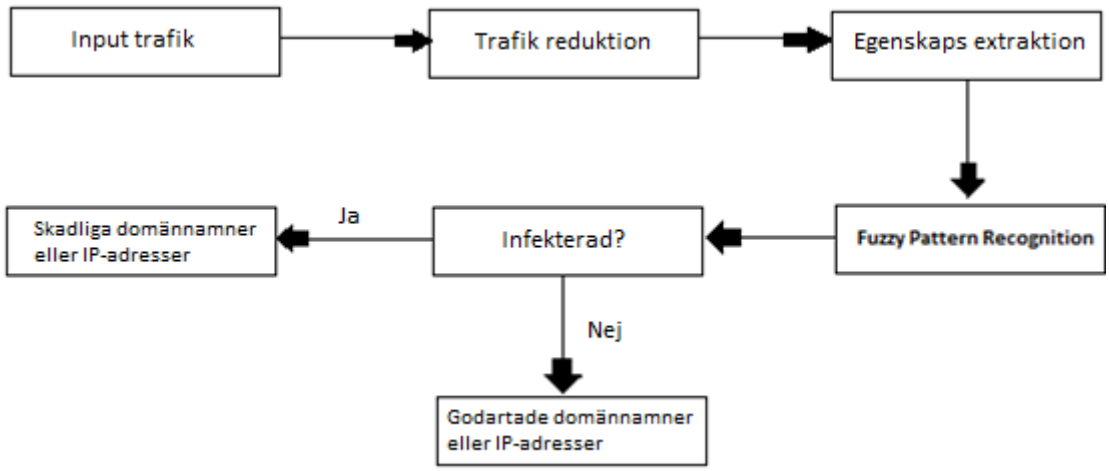
Trafikreduktionen använder en inre filtreringsmetod för att eliminera paket som är viktiga för att upptäcka botnet.

I unktionsextraherings fasen drar ut funktioner som kan igenkännas från inputspår så som; frågeintervalltider och laststorlekar.

Fuzzy mönsteridentifieringstekniken används för att på förhand känna igen givna domännamn eller skadliga IP-adresser. (Panimalar et al. 2016 s. 156)

Varje fråga ”query” avslutas med fuzzy fasen där funktionen bedöms som skadlig eller icke skadlig. Funktionsextraktionsfasen använder mycket tid för att observera funktioner från inputspåren. Prognostiderna kan variera mycket när DNS- och IP frågor görs. Tekniken försöker hitta lösningar för att förkorta processtider och höja på kvaliteten av prognosnogrannheten. (Panimalar et al. 2016 s. 156)

FPRF Algoritmen (Figur 9):



Figur 9. Fuzzy mönsteridentifiering baserad filtrerings algoritm. (Panimalar et al. 2016)

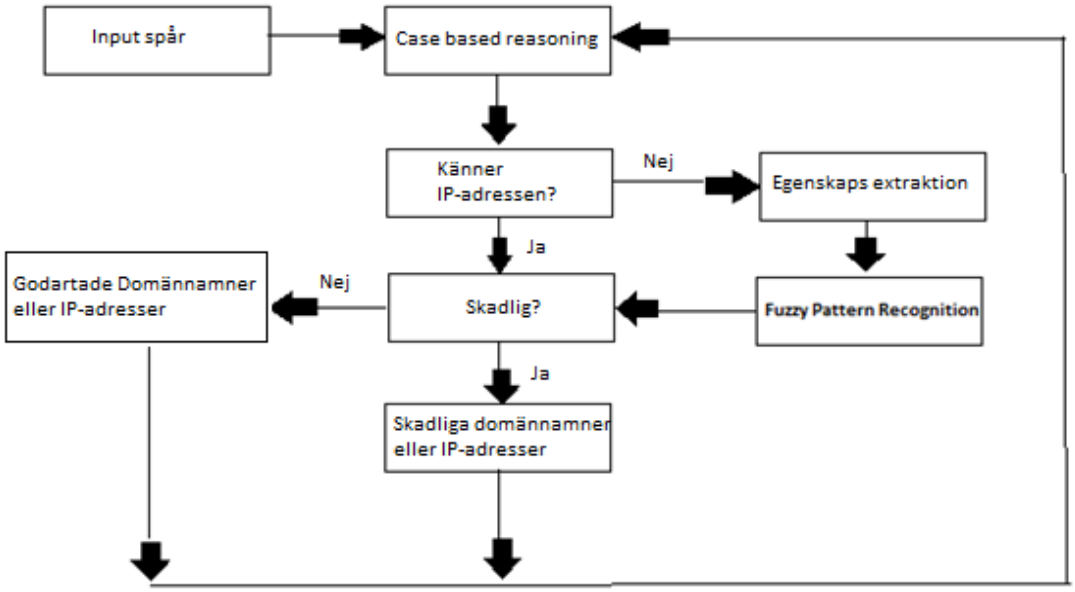
4.6.1 **Collaborative Pattern-Based Filtering (CPF) Algoritm**

Tekniken optimerar behandlingstiden utan att kvaliteten på resultaten lider. Tekniken har tre faser: kunskapsdatabas (case based reasoning), funktions extraktion och Fuzzy mönsteridentifiering. (Panimalar et al. 2016 s. 157)

Inputspåren fastställer riktigheten och tillgängligheten av domännamnet eller IP-adressen i kunskapsdatabasen; om de finns, då klassifieras den. När okända domännamn eller IP-adresser påträffas, då skickas en förfrågan till funktionsextraktionsfasen. Funktionsextraktionsfasen väljer beteenden som grundar sig på defitioner som medlemsfunktionen har definierat. (Panimalar et al. 2016 s. 157)

Slutligen skickas okända frågor till fuzzyfasen för att klassificeras som skadliga eller icke skadliga.

CPF algoritmen (Figur 10):



Figur 10. Collaborative Pattern baserade filtrerande algoritm för att uptäcka botnet. (Panimalar et al. 2016)

Tekniken utvärderar fyra olika beteenden:

1. Frambringar frågor om misslyckade domännamn (DNS): En bot har en innbyggd domännamnslista för alla möjliga C&C- servrar. För att inte bli upptäckt, ändrar C&C servrar ofta sitt läge till offline eller så stängs de av. Då kan boten inte svara på DNS-frågan som C&C servern ställer. Resultatet är ett misslyckat DNS-svar. (Panimalar et al. 2016 s. 157)

2. Fråge intervaller: Om svaret till en DNS-frågan misslyckas, kan en bot ge samma domännamn igen eller nästa domännamn från den inbyggda domän-namnslistan. En bot försöker kontakta C&C-servern på ett ofta använt intervall och intervallet hjälper till att identifiera boten. (Panimalar et al. 2016 s. 158)

3. Att frambringa misslyckade nätverksströmmar: botar använder också egna IP-adresslistor. Adresslistan används då boten försöker kontakta en oåtkomlig C&C-server. Annars, kan inte IP-adressen som erhållits från DNS servern bli

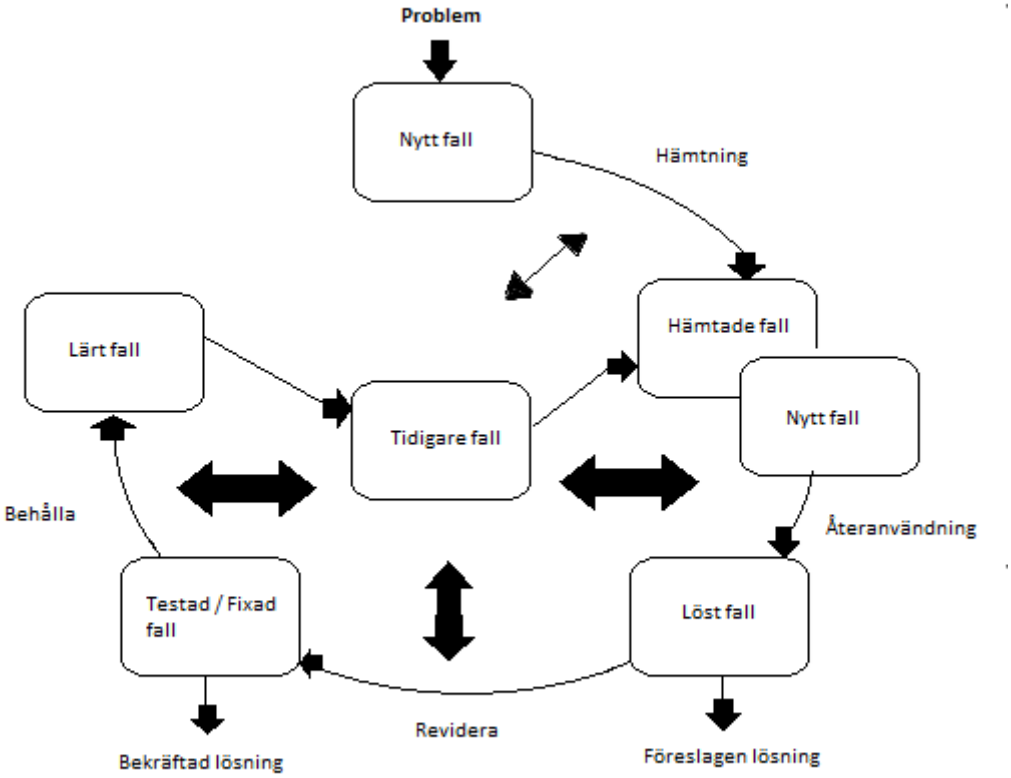
kontaktad. I båda fallen frambringas misslyckade netvärdsflöden. (Panimalar et al. 2016 s. 158)

4. Samma laststorlek för olika nätverksflöden: I allmänhet har botar flera lass som inkluderar SYN, UDP, GET och DNS. En bot försöker ladda ner kommandon när den når en C&C server. Kommandona ändras inte genast storleken av botlasten är okänd. TCP och UDP nätverksflöden motverkas på olika sätt. (Panimalar et al. 2016 s. 158)

4.6.2 Case based reasoning (CBR) fasen:

CBR som illustreras i figur 11, utgår från att ett nytt problem ofta liknar tidigare problem och därför kan tidigare lösningar vara användbara. CBR-systemets styrka är i att använda fall från fall förvaret effektivt. CBR-systemet beskrivs ofta som en CBR cycla. CBR cyclan innehåller fyra faser: hämtning, återandvändning, revidering och behålling.

- Hämtningsfasen: frågar om det finns tidigare fall som liknar de nya fallen. Denna fas tar fram liknande fall från fallförvaret.
- Återanvändning: fasen föreslår en lösning till de nya problemen. Lösningarna tas ur fallförvaret.
- Revidera: fasen används när en tydlig egenskap framstår så att den kan användas i framtiden.
- Behålling: fasen försöker lagra det nya fallet för framtida användning. (Panimalar et al. 2016 s. 158)



Figur 11. CBR cycla. (Aamodt, 1994)

Tekniken använder case baserad reasoning och Fuzzy mönsteridentifiering. CBR har en viktig roll i upptäckning av botar med hjälp av gamla problem och när den försöker förutse okända hot, använder den Fuzzy mönsteridentifiering. CBR upprätthåller fallhistoria om tidigare problem i två listor: en huvudlista och en sannolikhetslista. (Panimalar et al. 2016 s. 158)

Hämtningen är en viktigt del av CBR. I hämtning tas det fram liknande fallbeskrivningar från kunskapsarkiven och de som passar bäst framställs i återanvändningsfasen.

4.6.3 Funktions extraktions fasen

Fasen tar fram funktioner från DNS-frågor. Funktionerna mäts och bedöms som skadliga eller normala. Två inputspår samlas in för utvärdering: DNS paket-spår och nätverksflödesspår. Paketspår hjälper till att identifiera intervallmönster av DNS-förfrågningar. Nätverksströmspår hjälper till att identifiera förhållandet mellan DNS-frågor och nätverksflöden. (Panimalar et al. 2016 s. 159)

Funktionsexktraktionsfasen frambringar funktioner från paket som tas emot och den tar fram mönster från två olika spår som använts. (Panimalar et al. 2016 s. 159)

4.6.4 Fuzzy mönsteridentifierings fasen

Botar arbetar på ett dynamiskt sätt för att öka svårigheten att bli upptäckta. Observering av beteende ger mera information om botar och därmed kan boten upplösas från datorn. Fuzzy mönsteridentifieringstekniken används för att upptäcka botar på paketnivå, spår av DNS-frågor och nätverksflöden. (Panimalar et al. 2016 s. 159)

4.6.5 DNS fasen

I denna fas är det tre funktioner som definieras för att känna igen DNS-förfrågan:

- Inaktiv skadlig DNS-fråga
- Skadlig DNS-fråga
- Normal DNS fråga (Panimalar et al. 2016 s. 159 -160)

4.6.6 Nätverksflödesfasen

Följande tre funktioner används i denna fas:

- Inaktiv skadlig IP-adress
- Skadlig IP-adress
- Normal IP-adress (Panimalar et al. 2016 s. 160)

4.7 Allmänna motåtgärdar mot DDoS attacker

Fyra typer av försvarsmetoder mot DDoS attacker. Två av metoderna är proaktiva och två är reaktiva:

- Ingress/ Egress Filtering
- D-Ward
- Hop Count Filtering (HCF)
- Syn Cookies

4.7.1 Ingress/ Egress Filtering

I ingress/egress filtrering är ”edge” routern programmerad så att inkommande paket (ingress filtering) och utgående paket (egress filtering) filtreras. Paketfiltrering grundar sig på käll-IP-adresser utanför adressutrymmet. Paketen mottas vid routerns gränsnitt. En källadressen ufanför den tilldelade rymden bedöms vara falsk. Filtreringen kan också använda andra kännetecken så som portnummer eller protokolltyp. Metoden är proaktiv och den kan skydda nätet mot både direkta och tänkande DDoS attacker.(Aamir et al. 2014 s. 11; Gupta et al. 2010 s. 272)

Ingress/egress filtrering är lätt för Internetleverantören (ISP) eller nätverksadministratören att använda. Man måste veta, om IP addressen är inom rymden som nätverket använder. Filtrering förhindrar IP “Spoofing”. (Aamir et al. 2014 s.11)

4.7.2 Nackdeler med ingress/egress filtrering

- en erfaren anfallare kan förfalska IP-adresser.
- applikationsskiktet anfalls. Då används inte spoofing.
- att förverkliga filtering och andra regelverk ökar kostnaderna. (Aamir et al. 2014 s. 11)

4.7.3 D-WARD

D-ward är en brandmur som är installerad vid “source-end” nätverk. D-ward upptäcker DDoS attacker genom att samla trafikstatistik av utgående paket. Statistiken samlas i gränsrouterna och D-ward jämför statistiken med de givna modellerna av nätverkstrafik som baserar sig på transport- och applikationsprotokoll. (Aamir et al. 2014 s. 11; Mirkovic et al. 2005 S. 11 – 12)

D-ward försvarstekniken upptäcker snabbt attacker som är baserade på trafikavvikelser. Den kan identifiera kraftiga översvämningar och begränsa trafiken för att förhindra attacken från att göra mer skada. (Aamir et al 2014, 11; Mirkovic et al. 2005 S. 11 – 12)

D-ward är en “source-end” försvarsmekanism och därför är DDoS attacker begränsade, men D-ward har några nackdelar så som:

- nätverskprestandan är försämrad på grund av beräkning av trafikavvikelser i “edge” routern.
- stor ”overhead” tas ut på routern och darför har routern ett högt krav på processorkraft.
- det är svårt att upptäcka laglig och olaglig trafik, darför kan resultaten vara både falska positiva och falska negativa. (Aamir et al. 2014 s. 11)

4.7.4 Hop Count Filtering

Hop count filtering är en paketfilteringsteknik. HCF observerar TTL (time-to live) värden på inkommande trafik. TTL-värdet av ett paket värderas, det görs ett antagande över hur mycket paketet i fråga motsvarar. Skillnaden mellan det ursprungliga och det observerade värdet ger hoppantalet. Användarens server upprätthåller en lista över vanliga legitima IP addresser och motsvarande hopp räknas. (Aamir et al. 2014 s. 11; Yi et al. 2008 s. 10)

I en DDoS attack förstörs paket med falska avsändaradresser för att deras adress inte finns på listan eller så har deras källadress inte matchat räkningen av de relevanta hoppen. Tyvärr så har HCF nackdelar:

- tekniken är giltig endast för statiska IP-adresser

- tekniken ger inte tillgången till lagliga användare av NAT (Network Address Translation). NAT användaren kommunicerar via en allmän IP-adress
- lagliga användare kan inte kommunicera, för att de inte är med på listan och blir då avvisade. (Aamir et al. 2014 s. 11)

4.7.5 Syn Cookies

Syn cookies tekniken är den mest lovande försvarstekniken mot SYN (Synkronisering) flood attacker. I stället för att lagra sekvensnumret (ISN) av SYN paketet, lagrar servern identifieringsuppgifter av SYN/ACK-paketet. Denna identifieringsgskod är också ett sekvensnummer (autentiseringscookie) som tas fram och lagras av servern. Servern svarar vidare med ett SYN/ACK paket till begäraren. För att beräkna sekvenskoden (cookie värdet), använder servern en hashfunktion för att beräkna värdet på vissa paketparametrar d.v.s källadress, källport, destinationsadress, destinationsport och största segmentstorleken (MSS, maximum segment size). (Aamir et al. 2014 s. 11-12; Wesley 2006 S. 10)

Syn Cookies byter värde varje minut. Ett hemligt värde används också. Det hemliga värdet byts ut vid varje start av servern. När servern har mottagit ett paket med ACK flaggning d.v.s den sista singnalen från TCP trevägs-handskakningen, verifieras cookien. Om alla värden är korrekta, kopplas anlsutningen. (Aamir et al. 2014 s. 12)

Nackdelar med Syn cookies:

- en server som använder SYN cookies har inte tillräcligt styrka mot kraftiga SYN flood attacker.
- servern kan inte skicka ett förlorat SYN/ACK-paket, den nödvändiga informationen finns inte längre.
- beräkningskraft och resurser som en server använder mot en stor SYN flood attack kan göra servern långsam/stressad. (Aamir et al. 2014 s. 12)

5 RESULTAT

I tabell 1 summeras och jämförs de olika skyddsmekanismerna som behandlades I kapitel 5.

Tabell 1. Övergripande sammanfattning av de presenterande skyddsmekanismerna

	Äldre teknik. Över 5 år	Bara detektering	Detektering + förhindra	Låg budget/billig	>90% framgång *	uppåtgående	"färdig" produkt
Snort	x		x	x	x		x
HP/HN	x	x		x			x
UNIDS			x	x	x	x	
CPBFA			x	x	x	x	
Mining			x	x	x	x	

*Över 90% framgång i sitt egen fält. Alla dessa metoder fungerar på olika sätt och förhindrar olika attacktyper.

Tabell 2. En sammanfattning av de presenterade skyddsmekanismerna samt vilka typer av hot de skyddar mot

	DOS/ DDOS	Spam	Mask och skanneri	bara upptäck-ing av skade-program	Skadlig IP	Skadlig Domän	Skadig kod
Snort	x	x	x				
HP/HN				x			
UNIDS	x	x	x				
CPBFA	x	x	x		x	x	
Mining	x	x	x				x

När man tittar på tabell 2 så ser man att alla metoderna upptäcker olika attacker med hjälp av olika sätt:

Snort uppätcker rädan upptäckta botnet som finns i en databas. Nackdelen är att den inte skyddar mot nya attacker (som inte har upptäckts förut).

Honeypot/honeynet är endast en informationsförsamlare. Nackdelen är att metoden inte förhindrar attacker.

UNIDS baserar sig på två olika motorer, första motorn upptäcker DoS/DDoS, spam, maskar och port-skanneri. Andra motorns jobb är att upptäcka interna botnet (botar eller botmaster). Huvudsakliga nackdelen på NIDS är komplexiteten och en lång beräknings-tid.

CPBFA metoden upptäcker skadliga domäner och IP-adresser som används av botnet. Metoden använder case based reasoning (CBR) och Fuzzy mönsteridentifiering (FPRF). CBR upptäcker botar med hjälp av förkunskaper av gamla problem. FPRF algoritmen försöker identifiera domännamn och IP-adresser som används av bot C&C-servrar. Metoden baserar sig huvudsakligen på DNS-frågor och nätverksflöden. Nackdelen är att

den är begränsad till att förutsäga botar baserade i ett lokalt ”arkiv”. CPF algoritmen använder endast bergränsade resurser och den kan förutse inaktiva nya botar.

Datautvinningsbaserade metodens huvudsyfte är att förutse botnettrafik i nätverket med hjälp av maskininlärning (ML). Metoden upptäcker attacker som är HTTP, Spam eller DNS baserade (nätverksflöden). Nackdelar som ML har: stora data krav och limitering, det är inte en garanti att ML kommer att alltid fungera.

När man jämför metoder med varandra så är det CPBFA och Datautvinningsbaserade metoden som liknar mest varandra. Metoderna använder olika algoritmer och metoder, men själva resultaten ser ganska lika ut. Metoden som är mest intressant är UNIDS för att tanken om en icke-övervakad IDS är perfekt. Om metoden vidare utvecklas och beräkningstidrna förkortas anser jag att UNIDS har en mycked bra framtid.

Trots att snort ids och honeynet/honeypot är gamla tekniker (Snort utväcklades 1998 och Honeynet/Honey pot utväcklades 1999), så används de i många olika nya tekniker som en del av en större helhet. Tanken är att använda Snort och honeynet som en tilläggsteknik för att hjälpa huvudtekniken och för att få en maximal framgång i att upptäcka och förhindra botnet. Snort fungerar som ett virusprogram d.v.s. den känner till gamla botnet och Honeynet ger bra information om botnet och det är administratorns jobb att använda informationen som han har fått.

Utvärdering av nyare metoder:

UNIDS:

Tabell 3. UNIDS resultat. (Amoli et al. 2016)

	UNIDS
Falska positiva	3.61%
Äkta negativa	96.39%
Noggrannhet	98.39%
Återkallelse	100 %
Precision	98.12%

Datautvinningsbaserad detektion av botnettrafik i nätverksflöden:

Tabell 4. Resultat av datautvinningsbaserade tekniken. (Kalaivani et al. 2016)

Mining baserade tekniken					
Evaluerings	Klassificeraren				
kriterier	DT	NB	NN	SVM	
Precision	0.973	0.984	0.336	1.000	NB = Naive Bayes
Återkallelse	0.929	0.991	0.533	0.998	NN = Neurala nätverk
F-mätning	0.95	0.987	0.412	0.998	SVM = Stödvektor maskin
Prediktionsnoggrannhet	95.7%	98.4%	54.5%	99.8%	DT = Decision tree

CPBFA:

Tabell 5. Resultat av CPBFA. (Panimalar et al. 2016)

CPBFA	
Prediktionsnoggrannhet	
Algoritmer	Medel
FPRF	95.84
CPF	96.21

När man jämför resultat av nyare metoder så varierar prediktionsnoggranheten mycket lite. Alla metoderna fungerar på olika sätt och upptäcker olika attacker. Det är svårt att säga vilken av metoderna som är bäst, men om man funderar på temat ur en administrativ synvinkel så verkar icke-övervakad IDS som en mycket bra teknik för många olika användare för att metoden upptäcker och förhindrar attacker utan att den behöver ett människas ingripande. Icke-övervakad IDS skickar också information åt administratorn.

De nya teknikerna som jag går igenom i arbetet har en låvande framtid. All teknikerna hade en bra framgång, men de bör testas mera och större samspelstorlek krävs för att 100 procentig säkerhet uppnås, så att de fungerar så som testerna har visat. Som alla tekniker så behöver de också utveckling.

Utgående från materialet verkar en kombination av tekniker vara det bästa sättet att upptäcka och förhindra botnet. Tanken är att få så mycket information som möjligt, det är mycket viktigt att få information före attackerna.

6 SLUTSATSER

Arbetets målsättning/syfte var att beskriva botnetfenomenet och vilka metoder det finns för att skydda sig mot dem. De beskrivna metoderna valdes p.g.a att det fanns flere artiklar var de var nämda eller vilka hade låvande resultat.

Resultatet av arbetet var att UNIDS, CPBFA och datautvinningsbaserade IDS var de mest fungerande metoderna mot attacker. De bästa resultaten uppnås genom att använda olika metoder som en del av större skyddssystem d.v.s att använda t.ex. Snort och Honeynet/honeypot som delar av skyddssystemet, tillsammas med ovannämda metoderna

I arbetet framgår vilka metoder som fungerar som skydd mot olika slags attacker. Enligt den tillgängliga littaraturen och forskningen är de metoder som beskrivits i arbetet de bästa mot attacker.

Arbetets målsättning uppnåddes, för att de bästa skyddsmetoderna beskrivs (enligt litteraturen). Nackdelen med arbetet är att man inte kan bevisa att skyddsmetoderna verkligen fungerar såsom de framstår i artiklarna d.v.s arbetet innehåller inga test eller empirisk forskning.

Den tekniska utveckligen framskrider med väldig fart, det betyder att forskning och nya metoder behövs för att säkerställa det behövliga skyddet för datasystem mot mera avancedare attacker. De nya metoderna behöver användas mera för att man skulle få en bättre noggrannhet av resultaten.

.

KÄLLOR

Aamodt, Agnar & Plaza, Enric. 1994, Case-Based Reasoning: Foundational Issues, *Methodological Variations and System Approaches, AI communications*, 7(1), s.39-59.

Aamir, Mustafa & Zaidi, Mustafa Ali. 2014, DDoS Attack and Defense: Review of Some Traditional and Current Techniques.
Tillgänglig: http://cryptome.org/2014/01/ddos-defense.pdf
Hämtad: 29.8.2016

Amini, Pedram & Asmi, Reza & Araghizadeh, Muhammad Amin. 2014. Botnet Detection using NetFlow and Clustering, *ACSIJ Advances in Computer Science: an International Journal.* 3(2), S. 139-149.

Amoli, Payam Vahdi & Hamalainen, Timo & David, Gil & Zolotukhin, Mikhail & Mirzamohammad, Mahsa. 2016, Unsupervised Network Intrusion Detection Systems for Zero-Day Fast-Spreading Attacks and Botnets. *International Journal of Digital Content Technology and its Applications*. 10(2), s. 1-13.

Behal, Sunny & Brar, Amanpreet Singh & Kumar, Krishan (2010) Signature-based Botnet Detection and Prevention.

Tillgänglig:https://www.researchgate.net/profile/Sunny_Behal/publication/267846973_Signature-based_Botnet_Detection_and_Prevention/links/565be66d08aefe619b2488f4.pdf

Hämtad: 27.8.2016

Bhatia, J.S & Sehgal, R.K & Kumar, Sanjeev. 2011, Botnet Command Detection using Virtual Honeynet. *International Journal of Network Security & Its Applications (IJNSA).* 3(5), s. 177-189.

Chanthakoummane, Youksamay & Saiyod, Saiyan & Benjamas, Nunnapus & Khamphakdee, Nattawat. 2015, Konferens papper. *Evaluation Snort-IDS Rules for Botnets Detection*. National Conference on Information Technology: *NCIT*. s. 87-82.

Csubák, Dániel & Szücs, Katalin & Vörös, Péter & Kiss, Attila. 2016. Big Data Testbed for Network Attack Detection, *Acta Polytechnica Hungarica.*13(2), s. 47- 57.

Dange, Amit & Gosavi, Prashant. 2013. Botnet Detection through DNS based approach. *International Journal of Application or Innovation in Engineering & Management.* 2(6), s. 497 – 501.

Garcia, Sebastian & Grill, Martin & Stiborek, Honza & Zunino, Alejando. 2014. An empirical comparison of botnet detection methods. *Computers and Security Journal.* 45, s. 100 – 123.

Ghafir, Ibrahim & Svoboda, Jakub & Prenosil, Vaclav. 2015. A Survey on Botnet Command and Control Traffic Detection. *International Journal of Advances in Computer Networks and Its Security– IJCNS.* 5(2), s. 75-80.

Gu, Guofei & Zhang, Junjie & Lee, Wenke. 2008, Konferens papper. *BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic.* Computer Science and Engineering Faculty Publications Computer Science and Engineering. s. 1-19.

Gupta, B. B. & Joshi, R. C. & Misra, Manoj. 2010. Distributed Denial of Service Prevention Techniques. *International Journal of Computer and Electrical Engineering*. 2(2), s. 268 – 276.

Jyothsna, V & Prasad, Rama. 2016. FCAAIS: Anomaly based network intrusion detection through feature correlation analysis and association impact scale. *ICT Express 2.* s. 103 – 116.

Kalaivani, P & Vijaya, M.S. 2016, Mining Based Detection of botnet traffic in Network Flow. *IRACST - International Journal of Computer Science and Information Technology & Security.* 6(1), s. 535-541.

Karim, Ahmad & Salleh, Rosli Bin & Shiraz, Muhammad & Shah, Syed Adeel Ali & Awan, Ifran & Anuar, Nor Badrul. 2014. Botnet detection techniques: review, future trends, and issues. *Journal of Zhejiang University-SCIENCE C (Computers & Electronics*). 15(11), s. 943-983.

Kaspersky (2016) Botnet DDoS Attacks in Q1: Decrease in Length, Increase in Number.
Tillgänglig: http://www.kaspersky.com/about/news/product/2016/Botnet-DDoS-Attacks-in-Q1-Decrease-in-Length-Increase-in-Number.
Hämtad: 7.6.2016
Limarunothai, Ritthichai & Munlin, Mohd Amin. 2015, Trends and Challenges of Botnet Architectures and Detection Techniques. *Journal of information science and technology*. 5(1), s. 51 – 57.

Mane, Vrushali D & Pawar S.N. 2016. Anomaly based IDS using Backpropagation Neural Network. *International Journal of Computer Applications*. 136(10), s. 29 – 34.

Massi, Joseph & Panda, Sudhir & Rajappa, Girisha & Selvaraj, Senthil & Revanklar, Swapana. 2010. Botnet Detection and Mitigation. Konferens papper. Seidenberg School of CSIS, Pace University. s.1-8.

Mirkovic, Jelena & Robinson, Max & Reiher, Peter & Oikonomou George (2005) Distributed Defense Against DDoS Attacks.

Tillgänglig: http://www.isi.edu/~mirkovic/publications/udel_tech_report_2005-02.pdf

Hämtad: 19.12.2016

Nagendra Prabhu. S & Shanthi, D. 2014, A Survey on Anomaly Detection of Botnet in Network, *International Journal of Advance Research in Computer Science and Management Studies*. 2(1), s. 552-556.
Nikkhahan, Bahman & Aghdam, Akbar Jangi & Sohrabi, Sahar. 2009. E-government security: A honeynet approach. *International Journal of Advanced Science and Technology*. 5, s. 75 – 84.

Nieminen, Iiro-Matti (2016) Puolustusministeriön verkkosivuille tehtiin palvelunestohyökkäys – sivut toimivat taas.

Tillgänglig:
http://yle.fi/uutiset/puolustusministerion_verkkosivuille_tehtiin_palvelunestohyokkays__sivut_toimivat_taas/8761012.

Hämtad: 8.6.2016

Panimalar, P & Rameshkumar, K. 2016, Collaborative Pattern-Based Filtering Algorithm for Botnet Detection. *World Engineering & Applied Sciences Journal*. 7(3), s. 155-162.

Roesch, Martin. 1999. Snort – Lightweight Intrusion Detection for Networks. Konferens papper. *Systems Administration Conference.* s. 229 – 238.

Salokorpi, Jussi (2016) Palvelunestohyökkäys sulki eduskunnan nettisivut – kolmas hyökkäys viranomaissivuille muutamassa päivässä.

Tillgänglig: http://yle.fi/uutiset/palvelunestohyokkays_sulki_eduskunnan_nettisivut__kolmas_hyokkays_viranomaissivuille_muutamassa_paivassa/8763481.

Hämtad: 8.6.2016

Sharma, Abhishek. 2013, Honeypots in network security. *International Journal of Technical Research and Applications.* 1(5), s. 7-12.
Yle (2015) Nordea: Verkkopankki palvelunestohyökkäyksen kohteena.

Tillgänglig: http://yle.fi/uutiset/nordea_verkkopankki_palvelunestohyokkayksen_kohteena/7718202.
Hämtad: 8.6.2016

Tuomi, Jouni (2009) Laadullinen tutkimus ja sisällönanalyysi. Helsinki: Tammi s. 94,97,195

Vania, Jignesh & Meniya, Arvind & Jethva, H. B. 2013. A Review on Botnet and Detection Technique, *International Journal of Computer Trends and Technology*. 4(1), s. 23-29.

Wesley, M. Eddy. 2006. Defenses against TCP SYN flooding attacks. *The Internet Protocol journal.* 9(4), s. 1 – 43.

Yi, Fasheng & Yu, Shui & Zhou, Wanlei & Hai, Jing & Bonti, Alessio. 2008. Source-based filtering scheme against DDOS attacks. *International journal of database theory and application*. 1 (1), s-9-20.

Zheng, Jun. 2016. ”Soft-Man” and Data Mining based Distributed Intrusion Detection System. *International Journal of Security and its Applications.* 10(8), s. 145 – 150.