UNIVERSITY OF OSLO
Department of Informatics

Comparative Analysis of Network Attacks Against FQDN Using Honeynet

Samuel Feshazion Afeworki

Network and System Administration

Oslo University College

January 20, 2014

UNIVERSITAS OSLOENSIS
MDCCCXI

Comparative Analysis of Network Attacks Against FQDN Using Honeynet

Samuel Feshazion Afeworki

Network and System Administration
Oslo University College

January 20, 2014

Abstract

Domain names were developed to ease the challenges that human beings face in remembering large sets of numbers such as IP addresses. It is therefore common practice to name servers based on the services they provide or the administrative group they belong to. There is however a disadvantage with this practice, as it gives the same level of information to cyber adversaries about the role of servers within an organization's network. This information might be a security threat by itself. This paper investigates the impact that attractive fully qualified domain names (FQDN) might have in making such servers more targeted than servers with non-attractive FQDN, and addresses the null hypothesis of the author, that the majority of the attacks, do not take into consideration the FQDN. Statistical analysis of the collected data from the virtual honeynet set up for this investigation, shows that the null hypothesis is true, and having attractive FQDN does not have a significant impact on the preferences of cyber adversaries to probe systems within a given organization's network. Furthermore, in this thesis, an attempt is being made to identify previously reported offensive source IP addresses and updated trends of the source country for the majority of the malicious instigators are presented.

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor Evangelos Tasoulas for his support throughout the project. His guidance has helped me from the start until the end of the thesis work.

I would also like to thank Dr. Haarek Haugerud, Associate Professor at HiOA, for his initial idea of this thesis and invaluable support throughout the project.

My sincere thanks also goes to my lecturers and staff in Oslo and Akershus University College of Applied Sciences (HiOA) and Oslo University who have all played an irreplaceable role my journey to be possible.

I would also like to thank for my family and friends who have been utmost supportive during all times.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advancement in Information and Communication Technology has come a long way to contribute in a dramatic change that a mankind has seen so far. This advancement made most of daily operations to be carried out via applications and services over computer networks from one end of the world to another in a short period of time. Computer networks or commonly known as networks refers to all computers and other hardware devices interconnected via communication channels for the purpose of sharing resources and information [1].

Today, our daily life is becoming more dependent on Information and Communication Technology than ever before. With the wide availability and accessibility of Internet in almost every corner of the world, businesses are steadily moving from traditional shops to e-commerce, institutions of higher education continue to make their services, training and teaching, available to a wide variety of audience via distance learning rather than only on-campus lecturing, governmental service are moving to e-government, timely information are being shared and accessed almost everywhere due to the incredible progress in the area of Information and Communication technology (ICT). The benefits ICT brought to human life in the digital age could be volumes of text. What is pointed out here is only the beginning. However, ICT has also brought a number of social challenges such as privacy concern, fraud, identity theft, cyber espionage, hacking via computer virus, worms, Trojans to mention a few [2, 3].

Internet is a multifaceted unit which is comprised of various networks, users, and resources. Though this technology was designed to improve the well-being of an individual's and an organization's daily operation by allowing only legitimate users to access systems and application when requested. However, network systems which are the backbone of the Internet are suffering from cyber adversaries who are permanently trying to exploit potential vulnerabilities within these systems. A compromised host on a network will surrender its resources and can be used as a tool to launch an attack to other legitimate hosts in the network. Hackers try their best to use all their tools and skills as possible to get detailed information about a network then launch attack against the network.

In order to counter these malicious users and their tools, getting enough information about them gives a security professional an upper hand in safeguarding the network systems. The lesson that could be learned from analysing attacks includes the types of attack such as Distributed Denial of Service (DDoS), password based attack etc, most probed services SSH, SMTP, HTTP, MySQL etc and nature of the attacks-is it a targeted or random, identifying malicious IP address, domains and their geographical location. This will enable system administrators and network security professionals to take necessary precautions before the attack materialize and also it will help to design a recovery plan in case the attackers succeed in compromising the network systems. The main intention behind designing honeypot was to learn about hackers [4]. The Cuckoo's Egg by Clifford Stoll and the whitepaper An Evening with Berferd in Which a Cracker Is Lured, Endured, and Studied by Bill Cheswick are the two early publications which can be referred as the core stone for the designing of honeypot [4].

Computer and network systems have made an incredible advancement in the last two decades. They became very sophisticated enough to perform high level algorithmic execution and disseminate data within a blink but at the same time challenges against their true intention have also sky-rocketed. Therefore, if these systems are not properly secured and are breached, the consequence could be very severe to an organization or an individual.

Cyber-attack has being on the rise for the last decades as the power and capability of computer systems advances. It has become often a news headline. On the last week of August 2013 two major incidents of cyber-attack reported by different news agency about the downtime of New York Times web site and the China Internet Network Information Center (CNNIC). " The New York Times blamed a prolonged website outage on Tuesday 27 August 2013 due to an attack on the domain name registrar of Melbourne IT. A story published by The Times on Tuesday 27 August afternoon quoted the company's CIO, Marc Frons, as saying the disruption was the result of an external attack on the registrar by the Syrian Electronic Army, or someone trying "very hard" to impersonate the hacking group. The paper's main site was intermittently unavailable for several hours that afternoon and remained unavailable until after 7.00 p.m. ET." [5].

The second successful attack which took place within the same week of the month against China Internet Network Information Center (CNNIC) was reported by Voice of America news agency. After a distribute denial of service

(DDOS) attack on The china Internet Network Information Center (CNNIC) on August 25, Matthew Prince, the chief executive of CloudFlare, a company that provides Web performance and security services for more than a million websites, said " China saw a 32 percent drop in Internet traffic for domains in the company's network during the two-hour attack" [6].

On the third of October 2013 Adobe Systems Inc., a software developer, confirmed that their systems had been compromised by a sophisticated cyber-attack. The hackers were able to retrieve information of 2.9 million Adobe customers, including customer names, encrypted credit or debit card numbers, expiration dates, and other information relating to customer orders [7]. Another attack against Australian websites was successfully carried out by anonymous that is allegedly originated from Indonesia. The attack affected over 100 sites. Small sized businesses were the victims of the attack and their website messages were replaced by "Stop spying on Indonesia", and an image of a Guy Fawkes silhouette on the Indonesian and Australian flags. This attack was as a response to the allegation that Australian embassy in Indonesia was participating in the United States of America (US) lead spy network. The claim was made based on the leaked information by former US National Security Agency employee, Edward Snowden [8].

The attacks towards computer networks are aimed in countering the pillars of Information security, which are Confidentiality, Integrity and Availability (CIA). Confidentiality refers to making sure that information is accessed by authorized parties only. Integrity refers to the authenticity and completeness of information. Availability refers to making sure the needed information is accessible by authorized parties when needed [9].

The internetworked world made achieving the goals of these three pillars of IT more challenging as data and information are being processed, stored and transferred via computer networks. Therefore, computer and network security has become a crucial part in the field of ICT and is considered as a battle field for the blackhat community (hackers) and system or network administrators[10, 11].

ICT has become a backbone of organizations and the smooth organizational operations is heavily dependent on network security. Network security is a process of securing network systems from been harmed by malicious activities. Network security is not available as a single product but it is a system that combines multiple layers of security[12]. The effectiveness of network security is measured by its ability in identifying threats, analyzing threats, and then selecting adequate security controls to combat or neutralize them[12]. Different techniques and tools have been developed to detect, prevent and respond against malicious activities.

One of the tools used in the area of network security is Intrusion Detection System (IDS). There are three types of IDS known as host-based, network based and hybrid IDS. Host-based intrusion detection system (IDS) is software deployed on the system itself and can scan all resources within it [13]. The activities are logged to a secure database and are compared against any known malicious events listed in the knowledge base[13]. OSSEC is an open source example of HIDS which is able to analyze logs, alert, detect, response and check file integrity [14]. The second type of IDS is network based intrusion detection system (NIDS) which is deployed on the network to examine network packets against attacks. NIDS receives all packets on a particular network segment, including switched networks via one of several methods, such as taps or port mirroring. Hybrid IDS combines HIDS and NIDS[13].

Intrusion Prevention System (IPS) also known as intrusion detection prevention systems (IDPS) is a system that monitors a network for malicious activities such as security threats or policy violations. IPS's key features include identifying suspicious activity, log information, attempt to block the activity, and report. It can be implemented as a hardware or software. IPS are basically extensions of intrusion detection systems with one major difference in their ability to actively blocking or preventing intrusions that are detected. For example, an IPS can drop malicious packets by blocking traffic from an offending IP address etc[15].

With the rapid change the way attacks are carried out in the digital world, timely information and knowledge about attackers and their motives and tools gives a privilege to network security professionals. IDS and IPS have some shortcomings, such as detecting new attacks due to the lack of signatures in their database, collecting more information about attacker's activities, methods and skills, generating high number of false positives. These tools are also mostly deployed to protect organization assets from cyber threats rather than learning about network adversaries and their tools [16]. Therefore, honeypot a tool that aimed in learning about these adversaries and their techniques was developed.

Honeypot is a decoy computer system that is dedicated to be attacked, probed and compromised in order to learn about cyber adversaries techniques, motives and tools [17]. The value of a honeypot depends on effectiveness and attractiveness of its setup [17]. Honeypot can be categorized based on interaction to the attackers, deployment type, its architecture or purpose of deployment [18].

Honeynet is a network of honeypots which are specifically designed for the purpose of being probed, attacked and compromised by the blackhat community [18]. It is a category of high level interaction honeypot that are mostly used in the field of research to study tactics, motives and tools of cyber adversaries. It can also be used to identify zero day attacks [19]. Honeynet collects more information about attacker's behaviour and tactics as it is a set of different computer systems with services that are installed and configured in the same way as those of production system, but data stored on these systems are not real. However, from the intruders point of view these data and the system should seem real in order for the intruders to think that they compromised a production systems [17].

The core technology of honeynet includes three aspects: Data control mechanism, Data capture mechanism and Data analysis mechanism. They are security measures designed with avoiding any compromised honeypot being used as a tool of attack by the hacker [20]. Honeypots have been actively used as a research tool within the network security research works for since late 90th [21].

1.1 Motivations

The internetworked systems are constantly suffering from malicious activities as they have been probed to find out how they can be exploited due to their vulnerabilities by cyber criminals for different reasons. Cyber crime are becoming more and more a concern to individuals and business in daily operation. Hardly a day goes by without an incident where a system connected to the Internet been scanned or probed by attackers. The advancement in the area of computer technology has been incredible, but at the same time the ill action towards these systems has also increased in magnitude and tactics. Today, network scanning tools are capable of retrieving more detailed information about network systems including IP address, OS, domain names etc [22].

TCP/IP has become the de facto network communication protocol[23]. It is a four layers suite with each layer is developed to perform a specific task individually and work well in coordination with the rest of the layers in order for the communication between Internet hosts to be successfully achievable [13]. Communication between two Internet hosts takes place due to the unique IP address of each Internet host is assigned to and the port numbers that are allocated to service. The combination of IP address and port number makes a network socket.

Computers only understand bits which are 0s and 1s. Therefore, they are assigned IP address that can identify each Internet host and made communication possible across networks. But the limitation of human mind to remember many numbers could hinder the advantages that can be harvested from this technology if a solution has not been provided. This would have been one of the main difficulties if resolving IP addresses to domain names that humans can remember more easily than a series of numbers and vise-versa. Based on

CIA World Factbook in 2012 there were around 903,909,315 Internet host in the world [24]. According to Internet Systems Consortium, Inc quarterly survey about each Internet host based on address space and domain names as of July 2013, there are 996,230,575 [25]. Taking this two source as credible within less than a year there were 92,321,260 new Internet hosts. This shows how difficult it is to remember the IP address one need to use.

It is a common practice to call things based on the service that they provide or give them descriptive names. This practice of giving domain names to internet hosts by the services they offer or by the administrative group that they belong to is also commonly used or practical. For example mail.example.com refers to the fully qualified domain name (FQDN) of a mail server within the domain example.com or accounting.example.com refer the fully qualified domain name (FQDN) of a server that hosts the accounting applications and related information of an organization. This helps legitimate users to remember it easily and which Internet host to access based on their particular need as the name is very descriptive.

However, the same level of information- about the domain name that was intended to minimize the tendency to memorizing, is also available to cyber adversaries. Every small information can give extra advantage to an attacker about network system as he or she might look for vulnerable system and systems that have more valuable and confidential information on the Internet.

Honeynet is a favourable tool to carry out such research project as there is no real production value on the systems deployed but they appear real production systems from the intruder's point of view.

1.2 Problem Statement

This paper will try to verify the claim saying attractive fully qualified domain names such as hr.example.com or accounting.exmple.com can attract cyber adversaries attention than non-attractive fully qualified domain names. This will be investigated by deploying virtual honeypots and make them intentionally vulnerable. The following research questions will be ad-dressed on this paper.

1. Are attractive fully qualified domain names more targeted than non-attractive fully qualified domain names(FQDN)?
2. Does a different platform have a role in the popularity of a host?
3. Which services are more probed? Does the probe has the same magnitude on both groups?

4. How many of the source IP addresses are previously identified for their malicious nature? Identify source IP address geographical location?

1.3 Thesis Structure

This paper is organized in seven chapters where the background and literature review that are relevant to this research are categorized under chapter 2. Chapter 3 of this paper is dedicated for the approach and methodology that the research will follow to reach its goal. This includes the hardware and software experimental set-up and how data is collected and analysed to answer the problem statement of this research. The result of the experiment will be presented on chapter 4 and the next section of the document will analyse the result found in this project under category chapter 5. Chapter 6 will consist the discussion section that elaborates on the general overview of the project and its findings as a whole and future work. Chapter 7 will be the conclusion section of the document. References and Appendix will be provided at the end of the paper.

Chapter 2

Background and literature

The use of computer technology in our individual or organizational day to day activities have become inevitable as we are becoming more and more dependent on the services provided in the digital world. This technology makes more sense and contribute a greater deal to our well-being because it enables us to connect with each other over networks. Many easy looking things can be an amazing example of the magic behind computer networks. For example cash withdrawing from ATM where an individual is identified via personal identification number against his or her stored information on a bank's database and the service is provided or denied based on user's legitimacy as well as account balance. Another sparkling example that computer networking brought to our daily life is that the ease and effectiveness of collaboration via different means such as video conference, IRC etc.

The success and sustainability of an organization is highly affected in the timeliness and accuracy of information. Information refers to a processed data. In today's world data is collected, processed, stored and transmitted through computer networks. Therefore, securing computer networks in the digital age becomes very crucial for the existence of an organization can depend on the safety and security of the data transmitted to and accessed by via computer networks.

2.1 Principles of Information Security

When we think of securing a computer network, basically our main objective is securing data and information. Therefore, it is necessary to briefly discuss Information security and its pillars. In a simple and universal definition, information security refers to making sure that information confidentiality has been maintained, its integrity is not compromised and it is available to legitimate user when needed. These pillars is abbreviated by CIA in the field of computer science. Elaborating these pillars might give a better understanding and how they are related to this project [13, 26].

2.1.1 Confidentiality

Information is power has been a long standing proverb in human history. Any information that has been accessed by unauthorized entity lacks confidentiality. Hence it has been a long norm that information has been categorized in different levels and only those who have permission are allowed to access them. In today's world this norm has been adopted and is practiced with a greater attention as most information are collected, processed, stored and transmitted via computer networks[26].

Information accessed by unauthorized entities might have an overwhelming consequence to individuals, organizations and nations. According to the study conduct by Symantec Corporation and Ponemon Institute in 2013, the cost of data breach for companies in US is estimated to be about $5.4 million and Germany $4.8 million. France and Australia are listed on top for loosing high number of customers due to breach of data by unauthorized entities [27] On the internetworked world information confidentiality can be threatened due to malware, intruders, improperly configured systems, social engineering, and lack of proper network security.

2.1.2 Integrity

In the field of information security the term integrity refers to trustworthiness, origin, completeness, and correctness of information as well as the prevention of improper or unauthorized modification of information [9]. Information integrity can be compromised intentionally or unintentionally whichever the case, the consequences of a compromised information can be very damaging. For example as patients medical record are stored on a computer systems any unauthorized alteration to this vital data might put the patient's life in danger. The threats that can materialize against information integrity on computer networks could be data deletion, injection or replacing [26].

2.1.3 Availability

The third pillar of information security is availability. Data or information might be collected, processed and stored as well its integrity might have been maintained but if it is not available to the legitimate users when it is needed, its values might decrease or create user dissatisfaction. However, availability doesn't only focus on the information itself but also on the applications and systems [26].

Any downtime of a system leads to unavailability of the information which could lead to a delay in critical decision making that might have a very significant impact depending on the urgency of the matter. Denial of service attack is the biggest threat to this pillar of information security in cybercrime. The outage of The New York times for 7 hours on the 28th of august 2013 is a good example of this. A security researcher at Tripwire Ken Westin has told BBC that if a denial-of-service attack is successful millions of users are at risk [28].

Though only CIA are highlighted above but terms identification, authentication and authorization are part and parcel of information security. The image below depicts the components of information security.

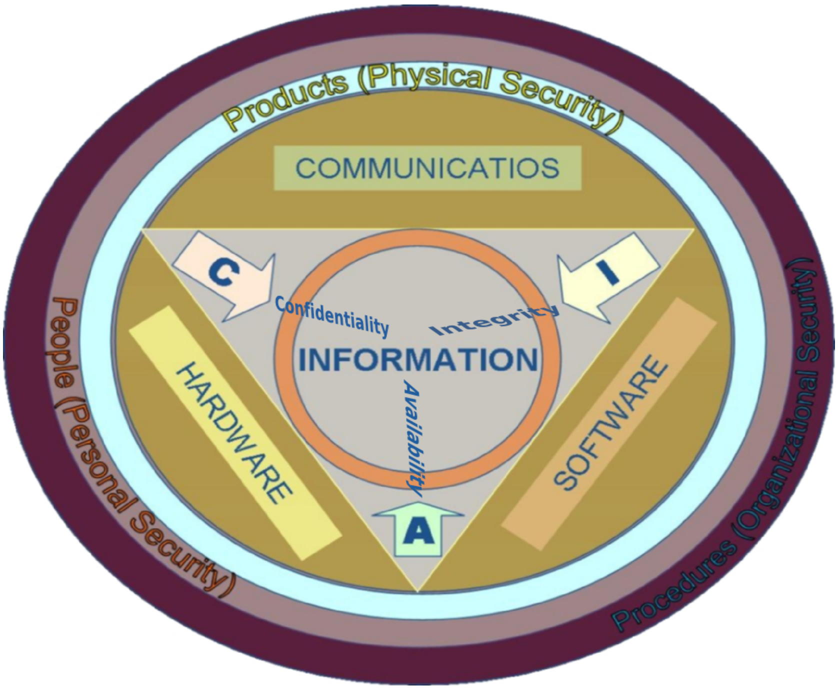

Products (Physical Security)
COMMUNICATIOS
C
I
Confidentiality
Integrity
INFORMATION
Availability
A
HARDWARE
SOFTWARE
People (Personal Security)
Procedures (Organizational Security)

Figure 2.1: Information security components[29].

2.2 TCP/IP Stack

TCP/IP is extensively deployed and has become a de facto protocol in the internetworking computing facilities. It is a four layer based network model where each layer carries out a specific task in the communication process [23]. Each layer is developed independently without impacting the others but it cooperates to provide full functionality. The four layers of TCP/IP are briefly summarized below [13].

Layer 4. Application Layer

The top most layer of TCP/IP is the application layer which defines application protocols and how host programs interface with the next layer. All higher-level protocols such as Domain Name System (DNS), Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), Remote Desktop Protocol (RDP), X Windows etc. operates at this layer [13, 30].

Layer 3. Transport Layer

The next layer to the application layer in TCP/IP network mode is the transport layer. This layer delineate the level of service and status of the connection during data transfer. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) main protocols belong to this layer [13, 30].

Layer 2. Internet Layer

This layer is responsible for packing data into IP datagrams that contains information of the source and the destination addresses (logical address or IP address) which is used to forward diagrams to hosts in networks. Internet Protocol (IP), Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), Reverse Address Resolution Protocol (RARP) and Internet Group Management Protocol (IGMP) are protocols that belong to this layer [13, 30].

Layer 1. Network Access Layer/Interface Layer

Network Access Layer refers to physical network and physical related apparatus that are in charge for data transmission. Routing and data synchronization, data format checking, signal conversion, error detection processes are handled by this layer. Some of the protocols that belongs to this layer includes Ethernet, FDDI, Frame Relay, Token Ring etc [13, 30].

2.3 Domain Names

As mentioned in the above section TCP/IP has become a dominant network protocol on the Internet. Data transmission between endpoints on the Internet is carried out by Internet Protocol (IP) which uses IP addresses that are allocated to both endpoints in computer networks. IP addresses are binary numbers meaning that 0s and 1s which can be understood by computers. IPv4 and IPv6 are the two versions that are being used by Internet Protocol (IP) today.

These two versions consists 32 and 128 bits fixed length of addresses respectively. These bits are not easily memorable by humans therefore the need to translate them to human readable numeric values is inescapable. For example 01001010.01111101.00101011.01101100 is an IPv4 address in a form of binaries to 74.125.43.108. Though the conversion of the computer understandable binary values to human readable numeric values helped humans to remember some limited IP addresses relatively easily, it does not eliminate the human memory challenges as the number of hosts increase within an organization or the general rapid increase of Internet hosts. The other problem with numeric IP addresses are that they are not descriptive. For example 74.125.43.108 does not give any information at all. Therefore, domain name becomes a necessity to solve the problem.

Based on CIA World Factbook for 2012 there were around 903,909,315 Internet host in the world [24]. Internet host refers to any computer system directly connected to the Internet. Internet Systems Consortium, Inc which conducts complete quarterly survey about each Internet host based on address space and domain names has also it quarterly survey data for Jul 2013 and the total count based on their survey was 996,230,575 [25].

The need for domain names was due to the growth in the Internet [31]. The following graph was retrieved from Internet Systems Consortium, Inc web page that shows the rapid increase in Internet hosts within the last two decades starting from Jan 1994 to Jan 2013 [25].

Domain name represents a given IP address in the Internet. Humans remember domain names easily than the numeric IP addresses and hosts on network can be identified to each other based on a trusted host name authentication mechanism. For example the www.hioa.no is a name that is assigned to the web server at Oslo and Akershus University College of Applied Sciences (HiOA) which has an IPv4 of 158.36.78.65. The domain name provided while trying to connect a destination point in the Internet needs to be converted to numeric IP addresses that a computer could understand. This is called domain name resolution. The mapping or resolving of the domain names to IP address and vice-versa is performed by Domain Name System (DNS). DNS is a hierarchical data base that consists of name space which is made up of domain names tree. Every node within this tree might or might not have resource records related to the domain name.

It is a usual naming practice to call Internet host based on the service they are offering or the administrative group they belong to. For example mail.example.com refers to the fully qualified domain name of the mail server of the domain example.com and accounting.example.com to refer to the fully qualified domain name of the server that hosts the accounting applications and related information of an organization. This helps users to easily remember domain names

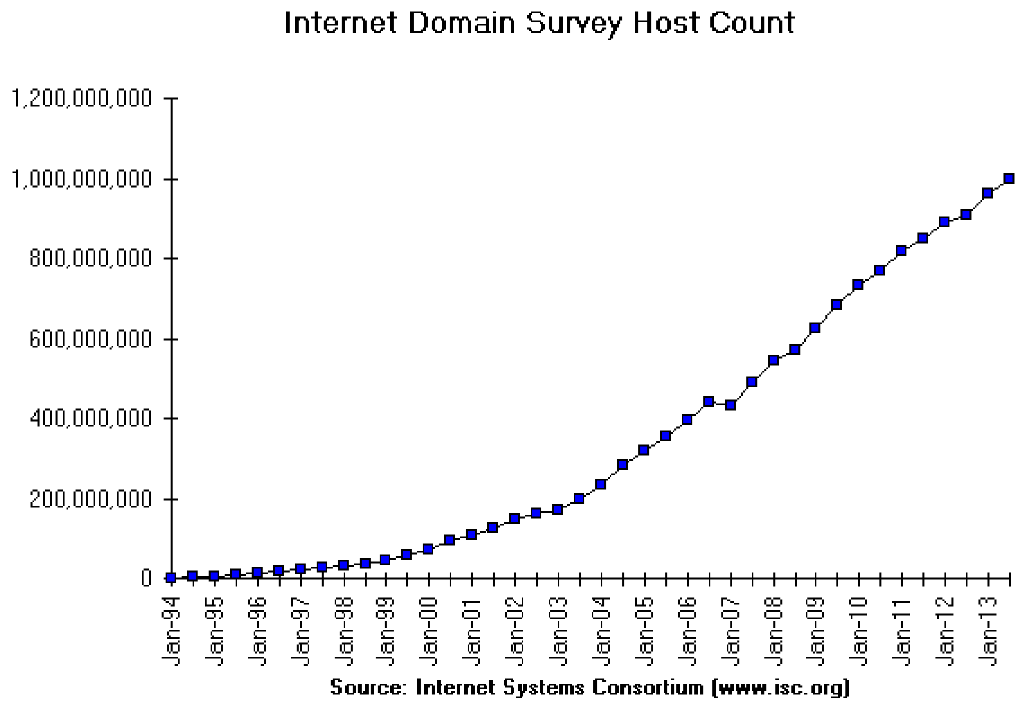


Figure 2.2: Internet Domain Survey Host Count[25].

and access a particular internet host by name as the name is very descriptive. However, since the same level of information is also available to cyber adversaries it can give an attacker an idea on a potential systems to target as he or she might look for vulnerable system or systems that have more valuable and confidential information on the Internet.

2.4 Network Scan and Attack

2.4.1 Network Scan

Computer networks are continuously probed. Network probing refers to trying to gain unauthorized access to a network system. Port scanning is one type of network probing. It can be performed by system or network administrators to verify system or network status and it can also be used by attackers to check host's availability and the range of services running on host that are active and exploitable due to known vulnerability on them [32].

Port scan can be carried out via several ways to probe network to get more information about network systems by cyber adversaries. TCP connect()scan, TCP SYN or half-open scan, TCP FIN,XMAS, and NULL scans, TCP idle scan, UDP scan, ICMP echo scanning are some of the common scanning mechanism that are integrated with scanning technologies like Nmap [33]. Tcp scan is considered as the main port scanning mechanism and the most stable of all port scans as it tries to manipulate the three-way handshake connection establishment process of TCP protocol [34].

A legitimate TCP connection is a three-way handshake meaning that the client sends SYN (synchronize) packet to the server, on receiving the packet the server replies with SYNchronize-ACKnowledgement (SYN+ACK) packet and the client sends an ACKnowledgement (ACK) packet to confirm and so that they establish TCP socket connection for data to be transferred between them. After the communication between the two end computers is over, the TCP socket connection will be torn down with another three-way communication [35]. The following figure depicts the three-way communication establishment.

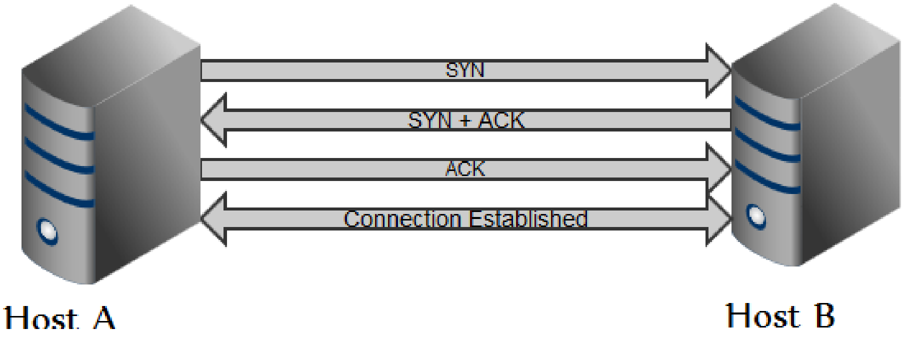


Figure 2.3: TCP Three-Way Handshake Connection Establishment Procedure.

TCP protocol acknowledges data been successfully received and reassembled in order. Thus unlike UDP, TCP qualifies as reliable protocol [35]. For example in the case of TCP SYN or half-open scan, the scanner host send SYN packet when receive SYN + ACK packet, it does not reply with ACK but the connections remains half open until the TCP connection is timed out, normally in 75 seconds. This might force the server to drop any connection if the backlog queue of the server reaches its limit [36].

2.4.2 Network Attack

An attack in computer and network security refers to an attempt to bypass a system's security due exploitable vulnerabilities that exist in operating systems (OS), service or applications [34]. Attacks generally can be classified as

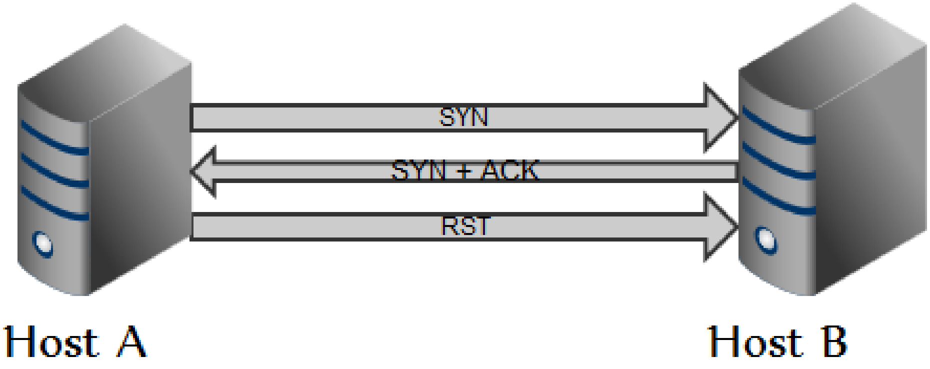


Figure 2.4: TCP SYN or half-open scan Procedure.

passive or active. Active attacks are those attacks launched to destroy or alter data while passive attacks are aimed at intercepting or reading data on the network without making any modification on them but try to extract valuable information like passwords, credit details etc [37].

Network attacks can also be targeted or opportunistic. Targeted network attacks are specifically aiming to compromise a particular network systems. These type of attacks are usually coordinated and well planned as they are unique by their nature [38]. The recent attack that targeted Australian Internet hosts between October and November of this year which was claimed by the Anonymous group originated in Indonesia is a good recent example of such an attack [8]. Opportunistic network attacks are attacks that are commonly referred as random attacks which look for network systems which can be compromised due to the vulnerability that exists with them. The vulnerability could be on OS, services, misconfiguration etc [39].

Attack against a network systems can be internal-attack which are launched from within the premise of a specific network or external-attack that is launched from outside [37]. Some of the widely used and known network attacks includes brute-force attack, dictionary attack, Distributed Denial of Service (DDOS), code injection, password attacks, Tcp SYN flooding attack, password sniffing etc [37, 23]. Attacks can exploit vulnerability at each layer of the TCP/IP suite. Previous researches have showed that more than 90% of Internet attacks use TCP [36, 40]. The next section will briefly highlight some of the commonly used and known cyber-attack methods against network systems.

- TCP SYN Flooding Attack is one of the widely used denial-of-attack tactics to compromise network systems. Though it has been known since mid-90s, but no full remedy has yet been successful against it[37, 23]. It manipulates the three-way handshake of TCP connection mechanisms, but does not send the ACK packet after receiving the SYN+ACK packet from the listener for its first packet which is SYN. The three-way handshake in TCP connection establishment has been explained on the section Network Scan and is also illustrated by figure on the same section. The attacker floods the target machine with SYN packets directing at the listing TCP port in order to cause denial-of-service for legitimate users [36]. The aim of this attack is not overloading network resources or memory of the host, but exhausting the backlog of the server which is associated with the port number due to full of bogus half-open connection which results the rejection of legitimate SYN segments. This attack is often carried out by spoofed IP addresses which will not verify the SYN+ACK from the listener for the initial SYN packet it has received. This type of attack is applicable to TCP based service such FTP, SMTP etc [41].

- Dictionary attack is another form of break-in to a network system based on systematically entering dictionary words as a username and password. The success of this attack relies on the legitimate users way of setting their password on systems. It is a common practice that user choose ordinary words. This can be easily prevented by set strong alphanumeric passwords and apply strong password enforcement mechanism. A form of dictionary attack is often used by spammers by sending messages that contain names and word to email address [23].
- Brute-force attack is another common attack method that tries to break into a network system by trial and error mechanism. Nowadays such attacks are performed by automating a consecutive guesses to find a pair of user name and password combination in order to get access to network systems. Such attacks can be prevented by limiting the number of failed attempts permitted at a given time and choosing strong password which consists of alpha-numeric with upper and lower case and special characters [37].
- Code Injection is an attack that is performed by injecting malicious codes to program or web applications. The injected malicious code will be executed with the application to fulfill its purpose. The attack takes advantage of inaccurate input/output validation vulnerabilities that exist in web or program application. Successful code injection attack could result in malfunctioning operation or asset destruction. Some of the widely used code injection attacks includes Sql Injection, HTML Injection, Cross Site Scripting, HTTP Response Splitting, HTTP Request Splitting and XML Poisoning Attack [20, 41].

2.5 Network Security

Network security refers to safeguarding computer network components and resources accessed via network from unauthorized users. It is a fundamental constituent of every network design. A good network security should safeguard all network layers [11]. The effectiveness of network security depends on the knowledge and understanding of the threats around the network, the weakness within the network that can be exploited and how can they be exploited. The online threats that can cause damage to a computer network have been outlined on the introduction part of this paper and under the section Principles of Information Security, they are categorized against which of the information security pillars that they can materialize. Bruce Schneier on his book titled "Secrets and Lies: Digital Security in a Networked World" classifies security goals into in preventing, detecting, and responding against malicious activity [12].

There are different tools and technology that are used to secure networks and learn about attacks against computer networks. These tools can be categorized based on which goal of network security they intend to achieve, some of them can be applicable to prevention, detection and responding.

2.5.1 Prevention

Prevention in network security refers to avert any unauthorized activities against computer networks and the resources using the network. This model of network security deals with:

- Authentication: any access to a network is authenticated via different mechanisms such as user id and passwords, PIN, digital certificate, Kerberos etc. If not authenticated then user will be prevented from accessing the system [13].
- Authorization: confirms what privileges the authenticated user has. For example which directories the user can access[13].
- Access Control: is used to assure integrity and confidentiality of data. This mechanism can be applied via security policy like Discretionary Access Control (DAC) or Mandatory Access Control (MAC) [13].

Firewall is one way of preventing network security by allowing only authorized connections to and from a network or computer based on predefined policy. Firewall can be deployed as a hardware or software or a combination of both [13, 14, 15, 18].

2.5.2 Detection

Detection from a network security perspective refers to identify any malicious activity in a network or a system. Detection of intrusion can be performed on host or network level or hybrid of both [13]. Host-based intrusion detection system (HIDS)is deployed on a host system and tries to detect any abnormal activities on the system. Captured activities are compared against signatures, rules and heuristics in knowledge base to identify unauthorized activities. Some example of HIDS includes OSSEC, Tripwire, AIDE [14, 13].

Network-based intrusion detection system (NIDS)is deployed on a network to detect any abnormal activities on network traffic. NIDS accesses the network traffic through either network tap, span port, or hub. Any abnormal activity will be flagged. Snort is an example of NIDS. The role of a IDS is passive, only gathering, identifying, logging, and alerting [13].

2.5.3 Response

Security is all about minimizing the magnitude of risks. Risks cannot be completely avoided, but they can be reduce to a limit. In the digital world a complete prevention against threats is almost unachievable due to the wide range of weakness that expands from the systems and network vulnerability to legitimate users ignorance or lack of awareness [12]. Detecting an attack on time is a marvelous achievement but if an appropriate response on due course is not given, the effectiveness of the security measure in place is not mature [42].

Response does not only mean getting rid of the problem but also been able to trace and find the attacker. This might not be an easy task on the Internet as attackers may impersonate their origin, but if successful it might be used as evidence against the intruder should prosecution take place. The incident response measure could include deny access to an intruder, report to Incident Response Team or responsible party, containing an intrusion and limiting the actions of an intruder , gather as much information as possible , clean affected system or network and restore system to operational status [12, 42].

2.6 Honeypots and Its Categories

According to the book called "Secrets and Lies: Digital Security in a Networked World" authored by Bruce Schneier security was labeled as process, but not a product[12]. Based on this quote securing a network means learning about the possible network assaults (attacks) that can take advantage over the weaknesses (vulnerabilities)of the network. Though there are a number of different security solution available today to hinder and learn about network attacks, but none of them can provide a full solution as standalone. Therefore, complementing of the security systems is inevitable. Honeypot is one of these tools that has been used in researching and learning about computer and network attacks in the last decades [20, 17].

Honeypot is a decoy information system aimed in deceiving intruders by convincing them that they are interacting with a real production system [17]. The information within this system is not of any use for an organization, but they are purposely designed to look like real in order to attract attackers. Therefore, any connection to and from the honeypot is considered malicious.

Honeypots has got several advantages over the other network security tools such as intrusion detection systems. Honeypot is a proactive detection technique while IDS is defensive. IDS generates high number of false positive while honeypot does not as there should not be any connection to and from the system. Honeypot generates more sufficient data for forensics analysis than that of IDS. It detects zero-day attacks, while IDS has shortage to detect them due to the lack of signature on the knowledge base it uses to compare captured activities. Honeypots can be categorized based on their interaction level, [21, 43]

2.6.1 Honeypot by Interaction Level

Honeypots can be classified based on the level of interaction a honeypot will have with an intruder depending on what kind of information are needed to be collected and to what level does a system or service been emulated to pretend to look like a real system or service. This also determines whether to setup low, high or medium interaction level honeypot [14, 4, 21].

Low-Interaction Honeypot

This type of honeypots does not provide to the intruder a real services and operating systems, but emulated once. Low interaction honeypots are easy to install due to their non complex design and basic functionality. The attackers have a very limited level of interaction as the services are predesignated. Setting up Kojoney SSH server and allowing an intruder to penetrate the system via brute force attack while collecting information about the intruder activities is an example of low level interaction honeypot [44].

Low interaction honeypots are more useful in detecting illicit scans and connection trials. The risk low interaction honeypot poses to other systems is low as the intruder is provided limited functionality and no real operating system. Though this can be seen as an advantage, the information gathered about the attacker is also limited. For most part the information collected is limited to the time and date attack as well IP and ports of source and destination of the attack. Low interactive honeypots are mostly limited for known attacks or behaviour [4, 44]. Some low level interaction honeypot are highlighted below:

- BackOfficer Friendly (BOF)runs on barely Unix but mainly on windows-based OS and emulates some basic services such http, ftp or mail. It fakes replies which keeps the attacker to interact therefore information can be gathered. BOF monitors only limited number of ports and its value is in detection [4, 44, 45].

- HoneyBOT is another windows-based low interaction honeypot and listens to wide range of sockets within the system by mimicking services with vulnerability. It fools an intruders by making them to think that they attempt is successful against a real system. The activity of the attacker will be logged for later analysis [46].
- PHP.HOP is used to trap attackers targeting web application vulnerabilities. It emulates various well known web application vulnerabilities [16].
- Glastopf is designed with the ability to emulate thousands of web applications vulnerabilities in order to collect information about web application attacks. It responds to the attackers as he or she expects from the vulnerability tried to be exploited [44, 47].
- Kojoney was designed to emulate SSH service. It was developed in Python programming language and has the capability to present informative statistical text report [48].
- Honeyd simulates TCP and UDP services and aims in detecting, capturing and alerting illicit activities. It can be implemented in Unix like and some version of Windows-based OS to listen to network traffic. It replies to network packets that are destined to one of IP addresses of the honeypots that it stimulates [49].

Medium Interaction Honeypots

Medium Interaction Honeypots merges the advantages of low and high level interaction honeypots especially in detecting botnet and collecting malware. This kind of honeypots does not simulate completely the OS or services, but they provide higher level of interaction with the intruder comparing to low level interaction honeypot and deceive the intruders to send their payload which can be analyzed later. Installing and configuring medium interaction honeypots are time consuming and relatively not easy compared to low interactions honeypots. However, more information will be collected than low interaction honeypot [1, 4, 50]. Some of examples of medium interaction honeypots are listed below:-

- Mwcollectd was the first Open Source Medium Interaction Honeypot developed by Georg Wicherski. Version 3.0 was rewrote in September 2005 with advancement to the older version. Mwcollectd was merged with the nepenthes project in February 2006 [46, 50]. This is mentioned for the sake of history.
- Nepenthes is designed to run on variety OS, including Windows via Cygwin, UNIX like systems and BSD. It emulates vulnerabilities widely

used by worms to spread malicious substance. It can bind itself with wide range of ports to receive connections from the internet [46, 50].

- Multipot is another graphical Medium Interaction Honeypot designed for Windows-based OS. It is not very scalable for distribution and deployment dedication. It is developed based on mwcollect concept [46, 50].
- Kippo is a medium interaction honeypot designed to log brute force attacks as well shell interaction that take place with the intruder [51].

High Interaction Honeypots

High Interaction Honeypots are real systems that are dedicated to be compromised by adversaries. They are deployed with real operating system and applications that are made intentionally vulnerable in order to be exploited. They have better capability of collecting information about attacks and attackers activity than low and medium interaction honeypots [50, 52].

Deploying high interaction honeypots gives the leverage of learning and detecting about new tools, vulnerabilities in operating systems or applications and communication mechanisms or method used among members of blackhat community. It can become accustomed to new command and control protocols [1, 44].

Though high interaction honeypots provides much information about the attacks and attackers activity is an advantage, but they are very complex to setup and pose a high risk as the attacker is provided with real OS and applications and this system can be used as a tool to attack other legitimate systems on the internet. To avoid harm to other legitimate systems on the internet, this kind of setup must be highly controlled and preferably deployed behind firewall [4, 11]. Some examples of such type of honeypots include:-

- Argos is based on open source emulator QEMU. It virtual high-interaction honeypots designed to identify zero-day attacks like new worms. It efficiently monitors and detects a compromised virtual machines in time by logging all traffic via tcpdump to a database [16, 53].
- High Interaction Honeypot Analysis Toolkit (HIHAT)is web-based high-interaction Honeypots with the ability to analyze and visualize data collected via graphical user interface. Some of the key feature of HIHAT include detecting Sql-Injection, (Source) File-Inlcusions, produces statistics about system traffic and provides geographical location and other details of the attacker [52].
- Honeynet is one type of high interaction honeypots. It is a network of several honeypots which can consists different plantforms and OS . Deploying honeynet could yield in gathering a large amount of data about

different types of attacks at the same time and make a qualitative analysis based on the information collected [16, 52]. Honeywall CDROM is the core element in the setup of honeynet which collects traffic information and activities performed on the honeypot.

- Symantec Decoy Server is developed by Symantec as a commercial high-interaction honeypot. It takes over the host system hardware to create four cage which will act as honeypot. Every cage has its own operating system and the attacker will interact with them by considering the interaction is taking place with a real system without realizing they are under trap [52, 53].

2.6.2 Honeypot by Purpose

Honeypot can also be categorized based on their purpose rather than only by their interaction level. This type of honeypot categorization was coined by Marty Roesch, developer of Snort [4].

Production honeypots

Production honeypots are deployed as part of security mechanisms for organizations network. They are designed as security enforcement and detect illicit activities but the attacker's information collected through these types of honeypots are limited comparing to research honeypots. They are fairly easy to setup and are often used by commercial institutions [4, 43, 54].

Research honeypots

Research honeypots are deployed with the intention to study about the black-hat community. Their main purpose is to learn in a detailed manner about the threats the organization's network face rather than security enforcement. By using this types of honeypot an organization can learn more about where the attacks are originated, the way they organize, tools and tactics that they used to attack etc [4]. These types of honeypots are highly used by research and universities institution to learn more about cyber threats. Examples of such setup is The Georgia Tech Honeynet at Georgia University which was established during the summer of 2002 and Brigham Young University security engineering lab for undergraduate and graduate students called ITSecLab [55, 56]. Honeynet is an example of this type of honeypots [4, 54]. Though honeypots are classified based on their purpose, but they can also serve as research and production honeypots as their distinction is not absolute.

2.6.3 Honeypot by Deployment

Physical honeypot usually refers to high-interaction honeypots where a real system with its own IP is intentionally made vulnerable to be compromised completely. Setting up physical honeypot is time consuming and expensive. Like their setting up maintaining them is not easy either [54]. Virtual honeypot refers to deploying a honeypot using a virtual machine with the help of virtualization technology. This approach gives more benefit due to the advantages provided by virtual technologies such as inexpensive to deploy, scalability and simplicity of maintenance [16, 54]. More than one honeypot can be deployed on a single physical machine which all can share the physical hardware the host and in case of public IP shortage in deploying, they can be assigned a private IP that can be accessed via NAT or bridge from the Internet.

2.6.4 Honeypot by Architecture

Traditionally the deployment of honeypots have been server-side based. However, in recent years client-side based side honeypots have evolved. Unlike the server-side based honeypots which wait to be attacked, client side honeypots crawl on the internet to find servers that have malicious contents [21, 57]. Capture-HPC is an example of High-interaction honeyclients that crawls to visit web pages [58]. An example of low-interaction client honeypot is HoneyC which is open source and works with various platforms. It was developed in Ruby programming language [59].

2.7 Literature Review

In the last decade honeypots have been actively used to perform network security related research works [1]. Under this section previous relevant research works to this project are summarized.

”Experiment using Distributed High-Interaction Honeynet (D2H)” [60]

The research was conducted by setting up honeypots on Amazon clouds and locally at Oslo And Akershus University College Of Applied Science. The research focused on identifying similarities and differences among regional and global attacks, which type of honeypot setup yields more information about attackers and their tactics, pros and cons of honeypot setup and data collection as well analysing in supporting for effective network security setup [60].

The honeypots deployed on Amazon clouds were made to send their logs to a source log server hosted on the premises of HiOA for further analysis. This was a precaution to save information deletion in case a honeypot was compromised by an adversary. Setting up a fully fledged honeypots on the Amazon cloud was not achieved due to privilege restriction in place from the service provider. However, useful data have been retrieved from this set up [60].

The research was focused on SSH and Web service attacks and it used Debian Squeeze operating system in all the honeypots deployed. The paper stated Sebek a key logging feature of Honeynet was not successfully installed though a reasonable amount of time was spent on it. The research concluded with some future works which could further be explored. It has stated that though installation of Sebek client was not successful but having key logger tools within the deployment of honeypots will result in learning more about attackers [60]. The experiment was only conducted on one operating system and with only two services, expanding it to learn more about different types of attack and attacker’s behaviour under different platforms and against various services is worth investigating as every OS flavour has its advantages over the other which makes it to be preferable to run some applications and services as well each service has its own value from adversary and organization point of interest [60].

”A comparative study of attacks against Corporate IIS and Apache Web Servers” [61]

The research work was designed to verify the suggestion that saying Windows operating systems based servers are particularly targeted by cyber adversaries. To verify the claim, the research studied a particular vector of attack. It is specifically designed to investigate the attractiveness of Web servers to hackers. The web servers software that have be deployed in this research was Apache web server which is a leading Linux web server and IIS Microsoft web server [61]. The project had no intention in testing the security level of the operating systems under use, but only to investigate if the IIS web server system attracts more attackers than that of Apache web server systems [61].

Both web server systems in this project were configured to simulate financial services websites. The two systems were web pages were identically configured and provide identical data. The only difference among the systems were the server headers. Network services that were allowed in the firewall in this project were only HTTP web service at port 80 and HTTPS at port 443. Any other network traffic to other services were blocked by the firewall [61].

Using this setup the research work answered which of the two systems got most vulnerability scan and on which system did attackers spend more time to compromise the system. There were two honeynets deployed in this experiment. Each honeynet had two fictitious financial web sites that were running under VMWare environment [61]. The data analysis on the research work was conducted on the average number of individual source IP addresses that attacked IIS and Apache. The research was conducted on two phases and in both phases Apache web servers were found to be less attractive to attackers than IIS web servers. The researcher suggested this could be due to the wide market penetration of W32 and from the perception of high security level in Linux systems than that of Windows. The researcher also highlighted that the research work does not demonstrate the security level of Linux or Microsoft Windows [61].

Some of the future works that the research highlighest are installing Apache on Windows server and run similar test to see if attackers avoid Apache installed on Windows than Apache on Linux servers. One another potential research area suggested by the author is to study why is Linux less attractive [61].

”Implementation of high interaction honeypot to analyze the network traffic and prevention of attacks on protocolo/port basis” [11]

This research work was first conducted at PUNJAB TECHNICAL UNIVERSITY JALANDHAR, PUNJAB in INDIA in the year 2012 and in January 2013 it was published on International Journal of Computer Applications [11].

The objective of the research experiment was to set up low and high interaction honeypots, analyze traffic captured and detect attacks on protocol basis. The research had deployed Windows XP SP2 and Ubuntu 11.04 for executing the experiment. The paper gave some informative guide on how to setup honeyd which is a low-interaction honeypot and Honeywall that is the gateway and the core deployment of setting up honeynet.The paper had claimed the same challenge about Sebek client installation as that of [60].

After the successful deployment of the honeynet, four flooding attacks were launched via hping3 which is a command-line oriented TCP/IP packet analyzer. The three attacks launched were aimed in observing and analyzing the consequences of the attacks on the walleye interface and the fourth attack was to learn how Sebek reacts and log the keystrokes [11].

Some of the papers reviewed throughout the time of this project claimed difficulties while setting up honeypot projects but they did not document how they solved the problem which could help other researches. However, [11] has precisely outlined the challenges faced during the setup process and how they were solved in a step by step instruction. The open source community is based up on the contribution of each individual or group that uses the open source products and a minor contribution has a tremendous impact on this community [11].

As the attacks were launched by the same researchers that have deployed the honeypot, it might not give much information about cyber criminals. However, the research was one of the good guides for learning about honeypot setup and testing for newbie in deploying honeypot before launching it online.

”Experiences with a Generation III Virtual Honeynet” [54]

The IT infrastructure is exposed to a daily assault due to the incredible malicious activities that exists in today’s internetworked world. Organization’s assets are continuously probed by cyber criminals for different motives via different exploit mechanisms the likes of worms, virus, DDOS attacks etc. There are different tools in place to counter for online malicious activities such as firewall and IDS. However, these tools have known shortages to deal with the dynamical change of tools, techniques and skill levels of the cyber criminals. These tools are more of defensive tools and they are passive in nature as their functionality depends on predefined rules and signatures which make them of no use when it comes to detecting or protecting for a new attacks[54].

Honeypot is not aimed at solving organization's network security dilemma. However, it is designed to learn about threats an organization network faces and how to improve network security by analyzing the data that is collected from the honeypot. The research utilized the benefits of virtualization technology by deploying multiple Ubuntu servers over a physical machine. The preferred virtualization technology on this research was VMware solution. A virtual high-interaction honeypot known virtual honeynet was deployed to study about attacks between September 2008 and November 2008 [54]. Like any other software honeynet is also in continuous development process. Therefore since its introduction by The Honeynet project in 1999 as Generation one, it has reached the third generations by improving the shortages identified on this generation and on generation two. Honeywall gateway in the first generation required one interface to connect to the Internet and one to the honeypot. The third generation of honeynet adopted the architectural design of second generation, but improvement was made in deployment and management of Honeywall which is the core element in Honeynet and unlike the previous version the installation has been made much easier by creating a Honeywall Roo that is a CD ROM that includes snort, snort_inline, Argus, tcpdump, hflow2, walleye interface and Sebek server [54].

This research used virtual honeynet by installing Honeywall Roo and on the specified research period of time was able to collect 30,000 attacks against the network and these attacks were categorized based on ports and services as well source IP alongside country of origin. According to this research the first successful attack was after four days and was via SSH brute force attack [54].

The research was conducted at the School of Engineering and Advanced Technology (SEAT) Massey University in New Zealand and used free or open source products only specifically Ubuntu server. According to the research paper installation of Sebek client was succeful unlike [11] and [60], but its installation step was not document. The paper conclude with a future work on comparing Honeynet at different virtualization environment and enhancing the data collection method via automation[54].

"Comparative Survey of Local Honeypot Sensors to Assist Network Forensics" [62]

The paper demonstrates the usefulness and impact of locally experimented researches in making global statistical decision and learning the difference and similarities of attack trends in network security by deploying low-interaction honeypots over several locations and comparing them based on some parameters to conclude about abnormal and particular activities. The research is conducted from volunteers as part of an academic project started by deploying high-interaction honeynet in each research site. However as the number of interested volunteers increased the deployment of high-interaction honeypot created difficulties. Unlike low-interaction honeypot the deployment of high-interaction honeypot is time consuming and has more risk not only that but the hard restriction imposed by their implementation and as each site differ its capacity to provide specific requirements might not be logical which will have an impact on statistical preciseness [62]. Another foreseeable challenge in deploying high-interaction honeypots is that their deployment involves real system and services configuration and on the process of deployment due to different people participated on the project with different competence misconfiguration is highly expected which could have its own impact on the credibility of the data collected. Therefore, the research team decided to use Honeyd a low-interaction honeypot as it emulates systems and services. Installation of sensor image and configuration files of Honeyd was automated from central station so each site will have the same configuration setup. The only requirement from an individual site with this approach is only one physical machine. Project participants have been given access to the central database where information is stored [62].

On this basis statistical analysis of attack origin, attackers OS, attack timing per day, attacker's domain, port probes were analyzed for two voluntarily participant sensors located in an academic network in France and academic network in Taiwan which had the same configuration and setup and run for the same period of time. As the result revealed by the research the Taiwanese sensor were targeted by several specific attacks which brought the attention of The Ministry of Education (MOE) of Taiwan to introduced Information Security Management System (ISMS) address the issue [62].

"Detection and Characterization of Port Scan Attacks" [63]

The complexity of Internet design had played a role in increasing the way weakness can be misused. Attacks to the network systems can be executed via vulnerabilities within services, applications, OS, misconfiguration etc. Port scanning is one the popular mechanism used to find a vulnerable host on the internet [63]. Communication between the attacking machine and the target machine starts by sending a message at a specific port from the attacker's machine and wait to get a feedback from the target machine. From the response the attacker can learn relevant host information which will help the intruder to identify what kind of attack to launch [63].

The research work was aimed in analyzing and characterizing port scanning traffic. The research developed a set of heuristics which was applied to trace network data. With the help of these data identifying and grouping suspicious packets were achieved and were used to retrieve relevant port scanning traffic statistics [63].

There are a range of tools that are used to scan port on the network. Nmap (Network Mapper) is one of the most known free and open source scanning tool that is able to retrieve a detailed information about hosts in a network with the services as well their version, types of firewalls in use, OS etc. Port scan can cause a various challenges to a network such as wasting resources, network obstruct, paving away for future attack [63].

Port scan was define for the paper of the research as all anomalous messages sent from a unique source IP at a time of trace with this port scan was categorized into vertical-targeting more than one port at a single host, horizontal-targeting identical ports on various hosts, block-a combination of vertical and horizontal. Based on their predefined criteria 9927 vertical scans, 5623 horizontal scans, and 2008 block scans observations were made. However, the research have seen some abnormality between vertical and horizontal scans that could be explored and more researches on Port scan traffic needed to be conduct were the paper's conclusion [63].

Similar statistical techniques can be applied to perform this project and as the research paper claimed there was little data set during experiment, so exploring this as extension of this project could be carried out.

"A Review of Port Scanning Techniques" [64]

Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) main protocols belong to transport layer of the TCP/IP suite [13, 30] The report paper was dedicated in analyzing port scanning techniques specifically on techniques used by TCP port scanners. Knowledge about the existing port scanning techniques will a network and system administrator an upper hand how to deal with them. The paper gave an explanatory about TCP segment with the six flag bits which are Synchronize (SYN) the first flag in initiating legitimate three-way handshake connection, FIN flag that tells the data sending is finished, RST a flag that tells the connection to reset, an urgent pointer which is represented by URG, ACK flag that confirms data was received and PSH flag that inform the receiver to send the data to the application level as fast as possible [64] .

The paper continues its elaboration on the most used TCP scanning techniques which are TCP connect() that is a scanning technique that is carried out by completing the three-way handshake connection establishment mechanism used by TCP to figure out the status of the service on network systems and the other most used techniques in TCP scanning is TCP SYN scanning which initiates the connection to a target system by sending SYN packet as if to execute a legitimate TCP connection, but based on the reply from the target the scanning host will be aware if the target is listening or not. If a target host is listening it will send a SYN/ACK packet to confirm that it real does listen but if RST packet is been sent, it means that the port is not in listening state. However, if the scanning host send RST if it receive SYN/ACK from the target host as from start the intention was not to establish a real connection but to just find out the status of the service. The paper also provide more information about indirect, stealth, fragmented, decoy and coordinated scanning techniques [64]. The paper has provided a good insight about the types of TCP scanning which are helpful in understanding the theoretical aspect of the scanning mechanism.

”Fast Portscan Detection Using Sequential Hypothesis Testing” [65]

The research was conducted in MIT Computer Science and Artificial Intelligence Laboratory. Network systems are being continuously probed daily by intruders to get illegitimate access to systems. The paper noted that devising an effective algorithm for port scanning is not yet been an easy task due various reasons as it is hard to easily identify what attempt is malicious or not. The other issue the paper raised as a reason for port scanning detection is that the granularity of identity meaning that how we categorize probes. The paper does not ignore the existence of some port scan detection mechanism, but admitted also the shortage in the existing solution in quality and quantity [65].

The research was aiming to address the problem of prompt detection. The algorithm was developed based on the assumption that if any non-useful connection from any host will be considered as a scan. Due to the sites volunteered to test the algorithm were using Bro NIDS, the researches algorithm was designed to fit on the Bro algorithm. The two sites involved in testing the algorithm were LBL and ICSI which are research laboratories with 6000 and 200 hosts respectively. The research had collected six dataset where each dataset covers 24 hour period. All the dataset used to test the algorithm was TCP connection summary found from the two sites [65].

The research developed an online detection algorithm Threshold Random Walk (TRW) which was then compare to Bro and Snort to measure it performance level. The paper claimed TRW has the advantage over Snort in its analysis as it is not dependent in window of time. The future work of this research includes how TRW can be able to respond, evasion and gaming, currently TRW can receive limited information which needs to improve, managing state meaning tracking every source host [65]. The paper has provide a guide to carry out hypothetical statistics while giving some clarification about port scan identification techniques.

”Learning More About Attack Patterns With Honeypots” [2]

Almost all human daily activities have become heavily dependent on computer systems. This is a proof by itself how far that this industry has come since its innovation. However, the threats against these systems have also increased almost at equal level. Network systems are flooded with bulk email (spam), virus, worms, malware, hacking to systems etc are also become part of the daily life of computer systems. Therefore, learning about network systems weakness and how they can be exploited by cyber adversaries is non stoppable journey [2].

Honeypot is a decoy computer that is intentionally made vulnerable to be compromised by cyber adversaries in order to learn about tactic, methods and motives of the intruders. Honeypot can be deployed on a local network to study about attacks against that specific network or environment, but this might not give security personnel wide image about the global attacking phenomena. Therefore, the research group decided to participate on the globally distributed honeypot project called lurre.com [2]. The main idea behind this project was to collect attack related data from different geographically located networks and store each data from a particular censor on database which can be accessible by any participant via access permission. The data collected at one center could be used by other member of the project which enable the participant to compare and contrast the attack behaviour against the locally observed behaviours [2].

The research carries out some statistical analysis based on the collected data from the experiment. Based on the statistical analysis the average number of attack sources per day was 184.94 and 2022 was the maximum attack source observed on this project. Attack source refers on this project as the source IP address that carries out attack on target system. Though, there is a clear variation on the source IP per day collected but the research group has stated that the reason is not certain but it could be due to receiving many packets from broken systems that uses NAT. The other statistical analysis that this research paper produced based on comparison on six censors was that Windows operating system was heavily used by source attacks comparing other operating system platforms. The research experiment was conduct for six months and the total attack source was 153,791 and this was further categorized based on IP address related to country using Maxmind GeoIP and the result shows the majority source attack were from USA at 24 % of all the attack source followed by china and Germany with 18% and 7% respectively. On port basis the analysis showed that TCP ports 135 and 445 had received the highest number of connection [2].

Chapter 3

Approach and Methodology

The initial step in caring out a research is to set up a well defined experimental laboratory which will play a great role in the success and fairness of the final outcome. To carry out this experiment virtual high-interaction honeypot will be deployed. Unlike low-interaction honeypot which limits attackers by emulating services, high-interaction honeypot provides the attacker with real service and OS [66]. Once an attacker compromised a system more detailed information will be collected through implementation of high-interaction honeypot which can be stored for further analysis. Though implementing high-interaction honeypot is time consuming, as it should look like a real production system in order to deceive the hacker, the time spent on implementing it is worth the information that can be collected about the hacker's activities [11].

On this project an automation installation script that was developed using kickstart as an assignment in system administration II course will be modified and used to ease the challenges of setting up high-interaction honeynet. The other challenge with such honeypots is that the risk it involves as the hacker is provided with real OS and services, a compromised system might be used to attack other legitimate systems in a network [60]. Therefore, attentive precaution needs to be taken. Honeynet, a high-interaction honeypot, was designed with this security concern in mind by the honeynet project. Therefore, outgoing network traffic from the honeypots will be controlled and set to a limit at the honeynet gateway. The implementation of high-interaction honeynet as a standalone research tool or in combination with low-interaction or with medium interactions honeypot had yielded a good result in learning about the ever growing cyber-attack [11, 54, 60, 62]. The subsections below will give detailed insight about the whole architectural setup, virtual environment choice, hardware and software specifications, security measures, data collection, hypothesis testing and analysis that will be carried out in this project.

3.1 Architectural Platform

The term platform is used in diverse ways which is tricky to reconcile but Oxford English Dictionary defines it as a raised level surface on which people or things can stand, usually a discrete structure intended for a particular activity or operation. As mentioned in 2.6.3 Honeynet can be deployed in a physical or virtual platforms. The challenges of deploying physical honeypot has also being outlined on the same section. Therefore, this project will be carried out on virtual environment.

3.1.1 Virtual High Interaction Honeypot

One of the greatest achievements in the computers industry is the golden invention of running multiple different operating systems within one single physical machine. Though the popularity of virtualization has created more attentions in the last few decades but its history goes way back to 1960s [67]. Setting up virtual honeypot has several advantages as stated on the book titled " Virtual Honeypots From Botnet Tracking to Intrusion Detection" by Niels Provos and THorsten Holz. Virtual honeypots are cost effective, scalable, easy to manage and simply portable, in case they get compromised, as they are files [67]. This is one of the many reasons for choosing the experiment be performed on virtual machines than physical ones. There are several free and open source virtualization solution to choose from like the free version of Vmware, Xen, User-Mode Linux, Kernel-based Virtual Machine (KVM), Proxmox etc. and there are commercial virtualization solutions. Kernel-based Virtual Machine (KVM) will be used on this project because the author is fairly familiar with the product.

3.2 Experiment Design

The network setup for this experiment will be designed as shown in figure 3.1. There will be sixteen honeypots grouped into two and they will be deployed on two physical machines. Both groups will be deployed on the same infrastructure in order to avoid any biased results.

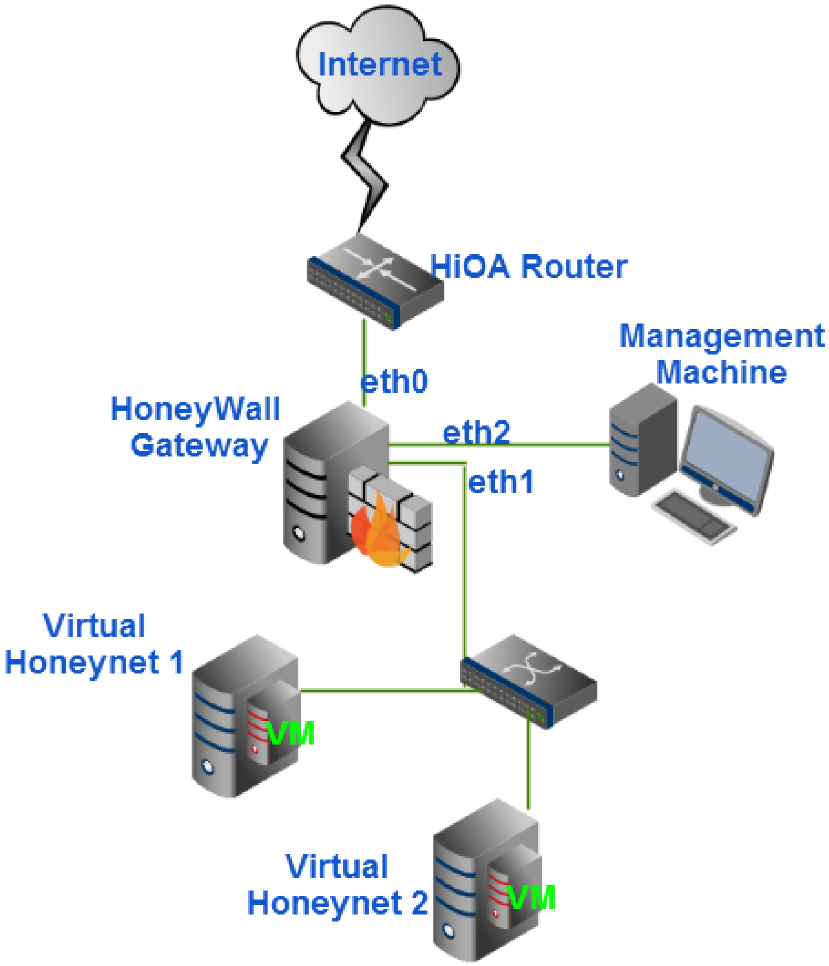


Figure 3.1: Network setup

3.2.1 Kernel-Based Virtual Machine (KVM)

As mentioned above, the preferred virtualization technology for this project will be KVM. KVM is a virtualization technology which allows different operating systems to run on a physical machine concurrently. These different unmodified images of Linux or Windows operating system images can run as individual physical machine with its own network card, disk, graphics adapter, etc. KVM virtualization supports Intel-VT and AMD-V processor technologies. For this project Intel-VT based processor will be used [68].

Virtual machines under KVM can be created via graphical user interface or command line. In this project after the basic installation of via kickstart the rest of the installation will be done via graphical user interface. Each physical machine or host will be configured to enable every virtual machine to access

Internet on its own public IP address so as to directly connect the honeynet with the Internet. This will be achieved by configuring bridge networking on the host and make the virtual machines use the bridge interface while they are been created.

3.3 Hardware and Software Setup

Section 3.2 described how the architectural design will look like. This section will describe first the hardware and then software that made up the design. The three physical server machines that will be used on this project are Dell PowerEdge 2950. Table 3.1 illustrates the physical server specification.

Physical Server	Model	Speed	Memory	OS	NIC
Dell PowerEdge 2950 (Honeywall Gateway)	Intel(R) Xeon(R) Quad Core E5335	2.00 GHz	4 GB	Centos Roo-1.4 Honeywall CDROM	3 Used
Dell PowerEdge 2950 (Virtual Honeynet 1)	Intel(R) Xeon(R) Quad Core E5335	2.00 GHz	32 GB	Ubuntu 12.04.1 LTS x86_64 (64 bit)	2 Used only 1
Dell PowerEdge 2950 (Virtual Honeynet 2)	Intel(R) Xeon(R) Quad Core E5335	2.00 GHz	32 GB	Ubuntu 12.04.1 LTS x86_64 (64 bit)	2 Used only 1

Table 3.1: Physical Machines Hardware Specifications

All virtual machines that will be deployed on this research experiment will have the same hardware and software setup but they will be using two different open source operating systems. Each virtual honeynet group will be have eight virtual machines where four virtual machines with Centos operating system installed and four Ubuntu operating system installed. Table 3.2 shows one from each virtual machine specification for demonstration purpose.

Virtual Environment	Model	Speed	Memory	OS	Virtual NIC
KVM	GenuineIntel	2.00 GHz	4 GB	Centos 5.9 i686	Virtio (Bridged by br0)
KVM	GenuineIntel	2.00 GHz	4 GB	Ubuntu 10.04.4 LTS i686	Virtio (Bridged by br0)

Table 3.2: Virtual Machines Hardware Specifications

The application software that will be installed on each virtual machine will be identical in order to avoid any biased attack attraction due to the vulnerabilities that exists within specific application on honeypot. A through investigation about known software vulnerabilities will be conducted and they will be installed on the honeypots. For example if version 2.11.10.1 of phpMyAdmin , an open source software which is developed via PHP programming language to administer MySQL, is installed on one virtual machine that is running Centos operating system in group Virtual Honeynet 1, the same application will be installed on another Centos honeypot that is categorized under Virtual Honeynet 2. Though this comparison is not paired comparison but any biased installation might have an impact on the outcome. Therefore, careful installation will be given high priority. The software that will be installed on the honeypots will be chosen based on known vulnerabilities that are listed under public knowledge by information security vulnerabilities databases such as Common Vulnerabilities and Exposures (CVE), Open Sourced Vulnerability Database (OSVDB), National Vulnerability Database and US-CERT.

3.4 Security and Honeywall Gateway

Compromised honeypots can be used to cause a severe damage to other legitimate systems on the network. Safeguarding other legitimate systems on the network should be given high priority [42]. While conducting this experiment, implementing a mechanism to reduce the damage that can result from a compromised system within this project is a priority. Therefore, the latest version of Honeywall CDROM also known as Roo-1.4 which is based on Centos operating system that uses a full hardware of a machine will be installed to act as defense mechanism against any serious harm that could take place on legitimate system as a result of any compromised honeypot from this project.

Based on the documentation of The Honeywall Project, the machine that is going to host the Honeywall gateway should have two atleast network interface cards. This means the possibility of managing the server from remote area is not possible [69]. One of the physical machines will be dedicated for Honeywall CDROM. The machine will have three network interfaces as shown on the figure Network Setup. One interface will be used to connect the Honeywall CDROM to the internet that will be the eth0 of the machine, the second interface that is the eth1 will be used to connect the honeypots to the Honeywall CDROM so that they will be able to connect to Internet while every activity from and to them is monitored and logged and the third interface named eth2 will be used for accessing the Honeywall web page and remote logging via SSH from allowed remote IP addresses. One of the aims of the Honeywall CDROM development by the Honeynet Project is to protect damages from compromised honeypots to legitimate systems. The machine that will host the CDROM roo image will be configured to act as a firewall gateway in order to monitor and control all network traffic to and from the honeypots. Outgoing network traffic will be limited per hour, but incoming network traffic to the services that will be monitored are going to be allowed. Any outgoing connection that exceed the limit will be dropped at the Honeywall gateway to avoid suspicion by intruders and dropped connections will not be reported to the sender. Successful attack will be reported to a given email address of the author. This e-mail is going to be checked constantly in order to follow the progress of successful attacks and take necessary action when all the necessary information is gathered. The Honeywall gateway will be invisible to the outside world.

The Honeywall CDROM is a bootable high-interaction honeynet which is comprised of variety of network security tools that can help in data capture, control and analysis developed by the Honeywall Project [69]. In this project these features will also be utilized to get data that can help to answer the problem statement. Honeywall CDROM is also capable of logging keystrokes by installing client side of Sebek on the honeypot and configure them to send their log files via a dedicated UDP port to Sebek server which is installed while installing the operating system. Data collected can also be analyzed at the graphical web interface or in depth analysis can be via other tools like Ethreal or Wireshark as the data are saved in pcap format. The following section briefly explains the component of Honeywall gateway and their applicability [52, 69].

- Sebek Server: the server side used to accept and log keystrokes sent from Sebek client (honeypots).
- Snort: logs packets at real-time and can be used to perform analysis based on content or probe.
- Snort_inline: a modified version of snort which accepts packets from iptables.
- Tcpdump: logs network traffic in .pcap format which can be read and analyzed with network analyzing tools such as Wireshark.
- Walleye Web Interface: will be used to remotely administer, configure and analyze data on daily bases.

3.5 Data Collection

The data that are needed to address the problem statement of this research will mainly be collected from the log files of the services that are deployed in the sixteen honeypots that are grouped into two. The data collection process will be done via Perl scripts that run every mid night of each day during the project life time. The retrieved data will be labelled based on the identification created for each honeypot and grouped to the group it belongs. All data will be isolated and secured to avoid any data corruption. The aim of this research is to find out if network attacks target attractive FQDN than non-attractive FQDN. To address the problem statement the data that will be collected from the log files are source IP address, source port, destination IP, destination port, initial and final timestamps related to source IP addresses and number of attempts each source IP address made at a particular service.

The number of connections attempts a unique source IP address made to each honeypot will be used to address which services are more probed and find out if there is a difference in magnitude in attempting to break-in systems with attractive domain names than non-descriptive domain name.

Honeywall gateway logs any network traffic from and to the honeypots from the launch date to the end of the project on hourly bases for each day [51]. These data provides information about the traffic that have been trespassing through it to and from honeypots, attackers OS, attackers origin etc. For this project the log files of server will be used to retrieve attackers information such as attackers country of origin, OS used by attackers. Attackers country of origin will then be compared to the country of origin of the attackers retrieved via perl script from `www.Dshield.org` and `http://www.geoiptool.com/` databases.

The data collected by the Honeywall gateway is stored in pcap format [51]. This needs to be converted to comma separated values (csv) format which will be able to be read and computed by script to retrieve the required information. This file can be easily imported to Microsoft Excel spreadsheet and RStudio for statistical calculation and graphing for the purpose of analysis.

The purpose of this research is to verify the claim that hosts with attractive fully qualified domain name are more attacked than hosts with non-attractive fully qualified domain name. To address the problem statement of this research, two groups of IP addresses where each group has eight public IPs that can be accessed directly from the Internet will be deployed. The information to be collected in this project from the Honeywall gateway specifically is the inbound connection in order to analyze network connection made to a specific destination IP from unique source IP on daily, weekly and throughout the project duration . These data will be categorized and labeled according to where they are being retrieve from and also grouped accordingly.

SANS Technology Institute had developed Internet Storm Center (ISC)that is used to collect and monitor malicious Internet activity. ISC uses the DShield distributed intrusion detection system for data collection and analysis. DShield collects data about malicious activity from across the Internet. This data is cataloged and summarized and can be used to discover trends in activity, confirm widespread attacks, or assist in preparing better firewall rules [70]. To address question four of the problem statement a perl script will be developed to check the malicious nature of the source IP addresses. Another script will also be developed to retrieve country of origin of the source IP addresses.

The data which will be collected will be stored on daily basis and labeled based on the identification created for each honeypot and grouped to where they belong accordingly. All data will be isolated and secured to avoid any data corruption.

Data collection for research purpose are often categorized into quantitative and qualitative. The aim of quantitative data collection is to test the hypothesis while qualitative is to understand social interactions [71]. The authors of the book titled Interactive Statistics Martha A. and Brenda G. has defined quantitative research as "Explaining phenomena by collecting numerical data that are analyzed using mathematically based methods (in particular statistics)" [71]. The definition makes it clear and understood that the research of this project falls under that category. The authors of [72] argued that quantitative research does not accept evolution methodology. That means data collection for such research should be identical and should be retrieved at all time of the collection period of time.

Inaccurate data collection will influence the final result of a research [71]. In this project inaccuracy of data collection can arise if there is a time delay in replacing any honeypot that is taken down after it has been compromised in order to prevent any harm that could emanate from it to other legitimate systems. To avoid this phenomenon the implementation of each machine that runs Centos or Ubuntu operating system will be the same throughout the project by taking clone from each honeypot's operating system after its initial installation is completed. This clone will immediately replace any similar honeypot in case there is a need to take it down after it has been compromised. One of the reasons of running virtual honeypot in this project is that there should not be any delay of replacing a compromised honeypot as it could lead to a biased data collection process which will make the final decision to be unfair and biased as well. This means that there will eight virtual machines for each group that will run 24/7 throughout the project.

Any data from compromised honeypot will be stored in a secure environment for later analysis to avoid any malware replication. Backup of all the daily data will also be taken and stored at another machine which will not be accessible by other users or over the Internet. All data from the Honeywall gateway and the honeypots will be labeled hourly and on a daily basis with an identification in order to segregate them according to where they are retrieved from and they will be used for further analysis to address the problem statement of this project.

3.6 Hypothesis Testing

The reason behind any data collection to investigate if there is evidence to support the research hypothesis [71]. In order to address the problem statement statistical approach is required after collecting the data. Describing all the statistical approaches is not the intention of this project however the approach that will be used to answer the problem statement of this research work is going to be highlighted.

One type of statistical inference is hypothesis testing. Hypothesis testing is a systematic method of testing a research hypothesis about population from a sample [73]. Final conclusion about the population's parameter is made based on test carried out on the sample. The test is carried out to verify the uncertain assumption made by a researcher about parameter or distribution [3, 74]. The assumption that is made by the researcher is known as null hypothesis and is denoted by H_0. The competing hypothesis to the null hypothesis is known as alternative hypothesis and is denoted by H_a [73].

Hypothesis testing involves stating the research question, specify the null and alternative hypothesis, calculating the test statistic, compute probability of test statistic and make conclusion whether there is an evidence to support the research hypothesis. The research hypothesis can be accepted or rejected depending on the result of the experiment. If the null hypothesis is rejected based on the experiment, then the alternative hypothesis is true [74, 73].

The final outcome of the test is based on the sample drawn from the entire population which makes the researcher to expect of a possibility of error in the outcome. Due to this consideration there is type I and type II errors in hypothesis testing. Rejecting a true null hypothesis is type I error and failing to reject a false null hypothesis is type II error. The decision relies on the p-value which is the probability value, from the sample data. The test will be conducted at some significance level or alpha. The common used significance levels 90%, 95% and 99% [74, 73]. Often hypothesis testing is done in the scientific world to compare two groups in order to determine if there is a significant difference between them based to a given comparison criteria or variable. Hypothesis testing can be one and two tailed alternative hypotheses [73].

Test statistic is a mathematical approach that helps to determine whether the null hypothesis is true or not [74]. As this thesis work is to compare data from two groups IP addresses and find out the impact of attractive fully qualified domain name in making a computer more targeted by attackers as compared to non-attractive fully qualified domain name. Therefore, the two-independent-samples t-test will be used to compare the results of the data collected.

The means and the standard deviations are very crucial while comparing two independent groups. Two comparable groups are called independent if there is no relation between the objects in the groups even if they are drawn from the same population [73]. For example taken random sample result of second year female and male students from a university and compare their grades. Though both groups are from the same population which is second year university students in this case, but their results are not dependent. In such scenario one-sample test is not applicable to test the null hypothesis as the two groups are not dependent. Therefore, two-independent-sample t test is used to compare the difference between the two independent means [75, 73].

In this project the test computation will be done via an open source statistical computing program RStudio and Microsoft Excel as the author is familiar with the software. However, to manually calculate t test for two independent sample test by hand is shown below.

$$t = \frac{M_x - M_y}{\sqrt{\dfrac{S_x^2}{n_x} + \dfrac{S_y^2}{n_y}}}$$

$$S^2 = \frac{\sum(x - M)^2}{n - 1}$$

- M_x refers to the first mean
- M_y refers to second mean
- S_x refers to sample standard deviation (variance) of the first group
- S_y refers to sample standard deviation (variance) of the second group
- n refers to sample size
- x and y refers to the individual scores

3.7 Data Analysis

The idea of setting up honeypot is to collect data which will be used to create knowledge so as to enhance the defense mechanism against abuse of computer network. If collected data cannot be analyzed effectively and correctly, the value and power of knowledge that would have been generated from it will diminish. The data collected in this project will be labeled in order to statistically analyzed them based on time stamp, source IP, source port, destination IP and destination port and group them according to which group they belong.

Based on the data collected from the log files and the Honeywall gateway the statistically computed result will be analyzed to interpret the findings of this research. Analysis will be done to accept or nullify the claim that says attractive fully qualified domain name are not targeted more than non-attractive fully qualified domain names". The other analysis will be done on the results on the information retrieved about the source IP addresses to answer the last question of the problem statement.

3.8 Wireshark

Wireshark is an open source network packet sniffer and analyzer tool which can be installed and run in various operating systems. Packet sniffer in computer networking refers to a tool that can help to capture network traffic. Packet analyzer is a tool with the capability to analyze the captured data. Wireshark has GUI and can be accessed via Tshark from a command line. It is capable of capturing live network packets and also can read previously captured files. Wireshark captures files in libpcap format the same as Tcpdump which is a command line network analyzer. Wireshark can be queried to filter from captured files such by protocol basis [76].

The log file stored in Honeywall gateway are stored in pcap format which can be read by Wireshark. This file can then be converted to other format via Wireshark [76]. Though with the huge amount of data captured by the Honeywall gateway about the network traffic to and from the honeynets this might not be a wise idea but still considering the challenges that arise in data collection and analyzing using the visual aspect will give a direction how to go about it. Wireshark will be used in this project to do some analysis as well.

Chapter 4

Results

This section is dedicated to the actual lab setup and the results obtained from the experiment. As mentioned in section 1.2 the main aim of this experiment is to verify if attractive FQDN will be more targeted than non-attractive FQDN by cyber adversaries. Therefore, setting up what is proposed in the above section enabled to execute the research.

4.1 Actual Test bed Setup

The integral part of any research that deploys high-interaction honeynet is the Honeywall CDROM roo. In this project version 1.4 is installed in a dedicated server with the specification that was shown in 3.1. Unlike version 1.3, this version is developed on Centos Operating system as its base and its installation processes have been simplified by making graphically installable [69]. The server has three network interface card where eth0 and eth1 are configured in a bridge networking mode. This hides the server from outside world while monitoring each activity that goes through it from and to the honeynets. Eth0 is connected to the Internet to allow the honeynets that are connected to the Honeywall gateway through eth1 to be reachable from outside world while they are being monitored. The third network interface card which is eth2 is configured for management purposes at port 22 for remote login via SSH and securely accessing the Walleye web interface via port 443 only from authorized IPs. The Honeywall gateway is configured to log every incoming and outgoing activity to and from the honeynets.

The Honeywall gateway encompasses Snort among the other tools. Snort is an Intrusion detection and prevention system which is able to log packet on real time and can be used for traffic analysis. It can detect vulnerability exploit attempts, port scans and other malicious network activities. Snort is configured to log network traffic to and from the honeynets. In this project the log file is used to retrieve additional information about the unique IP that are logged by each honeypot.

As compromised systems could be used by intruders to cause severe harm to legitimate systems in the Internet, the outbound connection from the honeynets was restricted by protocol bases. Outgoing TCP and UDP connection from the honeynet are limited to 20 per hour and ICMP connections were limited to 50 per hour by the firewall equipped within the Honeywall gateway. Any outgoing connection that exceeds the specified limit is subjected to be dropped without any alert to the intruder that his or her connection attempts were dropped by the firewall of the gateway.

The Honeywall gateway is also configured to send alert email to the administrator of the server when there is a suspected activity. The email was regularly checked to avoid any serious harm in case there is a compromised honeypot within this project. Data was collected on daily basis for this project. The web interface was also regularly checked for any abnormal behaviour of outgoing network traffic behaviour.

4.2 KVM and Virtual Honeypot Setup

The virtual environment used for this project is KVM. Both physical servers were equipped with Intel processor and KVM was installed as para-virtualization hypervisor Ubuntu being the base OS for both physical servers. All the sixteen virtual machines should be accessible directly from the Internet, both host servers are configured to act as bridge in order to allow the eight virtual honeypots deployed on each host server to share the physical NIC. Creating virtual machines in KVM can be done via virt-manager which is GUI or command line via virt-install. The packages that are needed and used to install KVM on Ubuntu are shown below.

```
apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder
bridge-utils kvm  virt-viewer virt-manager virt-top
```

After the basic installation via Kickstart, the process of creation virtual machines in this project is done by the GUI feature of KVM. as only two virtual honeypots need to be created initially, one with Centos and one with Ubuntu operating system. They are then cloned to create 8 identical virtual machines for each OS, which makes 16 virtual honeypots in total hosted by two physical servers. Each physical server hosted four Centos based and four Ubuntu based virtual honeypots with the hardware and software specified in table 3.1.

The public IPs addresses assigned to the honeypots are allocated for this project from the IP pool of Oslo and Akershus University College (HiAO) subnet 128.39.120.0/24. These IP addresses are grouped into two groups. Eight of them are assigned attractive FQDN while the other eight IP addresses are given non-attractive FQDN. All honeypots that have attractive FQDN are grouped as Virtual Honeynet 2 (VH2) and all honeypots with non-attractive FQDN are grouped as Virtual Honeynet 1 (VH1).

In order to avoid any biased attack attraction all virtual honeypots are configured with the same application software that have known vulnerabilities based on the information collected related to network security threats listed on databases such as Common Vulnerabilities and Exposures (CVE), Open Sourced Vulnerability, National Vulnerability Database and US-CERT. After going through the lists of these databases the following application are installed in the honeynets. PhpMyAdmin version 2.11.10.1, Roundcube Webmail version 0.1, Wordpress version 3.4.0, Joomla version 2.5.3 and version 1.8.3 of GnuCash which is financial accounting software and other application software that resemble to the honeypots with attractive FQDN. Creating the same environment of comparison was one task that was carefully carried out in this project as any difference in setup could lead to biased results. Therefore, any package that is installed in any machine will be installed in all. All installed software applications were properly configured and simple web pages were hosted.

Honeynets are deployed to deceive cyber adversaries by pretending as a real production system. Though announcing via web crawlers might increase the number of attacks count but for the purpose of the research the most known search engines were exclude from crawling the web pages hosted. This is to minimize the number of count that could result due to search engines. The excluded search engines are Google, Yahoo, Bing, Baidu Spider, MSN bot etc. Therefore robot.txt file was edited to exclude the most known web crawlers.

Based on previous research works that were reviewed prior to the installation of this setup, most commonly used and identified weak username and password combination are set in order to allow intruders to login via SSH remote login. Once the actual setup is done the honeynets were launched. The next section is dedicated to show the result of this project.

4.3 Data Collection Procedure

The core idea of this research is to find out if computer systems with attractive FQDN will be more attacked than computer systems with non-attractive FQDN. As mentioned in section 3.5 , information about unique source IP addresses that attacked the honeypots were collected. This information is retrieved from each honeypot log files every mid-night for 70 days. On this project several Perl scripts were developed to retrieve relevant information from log files. Information retrieved includes unique source IP, initial and final timestamps, source port, and destination IP address. All scripts that were used on this project with comments about their purpose and usuage are attached on the appendix section.

The data collection process started Saturday 21st September and ended on Friday 29th November of 2013. Any source IP address attacking a single destination IP address to one or all of the services that were deployed within it was counted as one for a particular destination honeypot. Each source IP address attacked any service within a particular honeypot is also recorded as unique source IP to that particular service and the number of attempts that particular source IP address made to each service is also recorded to compare the severity of attack attempts each honeynet gets per service and to determine which services are more probed during the 70 days experiment.

For example if source IP address 192.168.0.10 attempted to attack destination IP 192.168.0.135 on Oct 1 at 02:00:00 and then at the same day at 16:34:45 then on Oct 20 at 10:23:12 via any type of vulnerability, it was counted as one unique source IP to the destination IP. The same source IP address was registered per service level as well. However, the number of trials this unique source IP made at each service to get illegitimate access is recorded in order to conduct service level analysis that will help to find out if the severity of attempts will significantly differ due to the attractiveness of the FQDN and also to determine which services are more probed. The results are presented below.

The other data collection step was to retrieve information about the unique source IP addresses that have attacked the honeynets. The information was retrieved from the database mentioned in 3.5. The information retrieved from these database are previous reports about malicious activities of the source IP from Dshield database and country of origin. From the database of Geo Ip Tool database is also country of origin of attackers. Based on the information from these database and the Honeywall gateway information, the country of the attackers are reported and analysed.

4.4 Number of Attacks per Destination IP

Based on the counting mechanism for unique source IP address illustrated above, the total number of unique source IP addresses that are found on each honeypot throughout the project life period are shown on 4.1 and 4.2. Figures 4.1 and 4.2 represents for the honeypots assigned non-attractive FQDN and attractive FQDN respectively. The total unique source IP addresses that attacked VH1 are 2365 and a total of 2395 unique source IP addresses attacked VH2 during the project life time. The total results found on this experiment for each honeynet are depicted in 4.3.

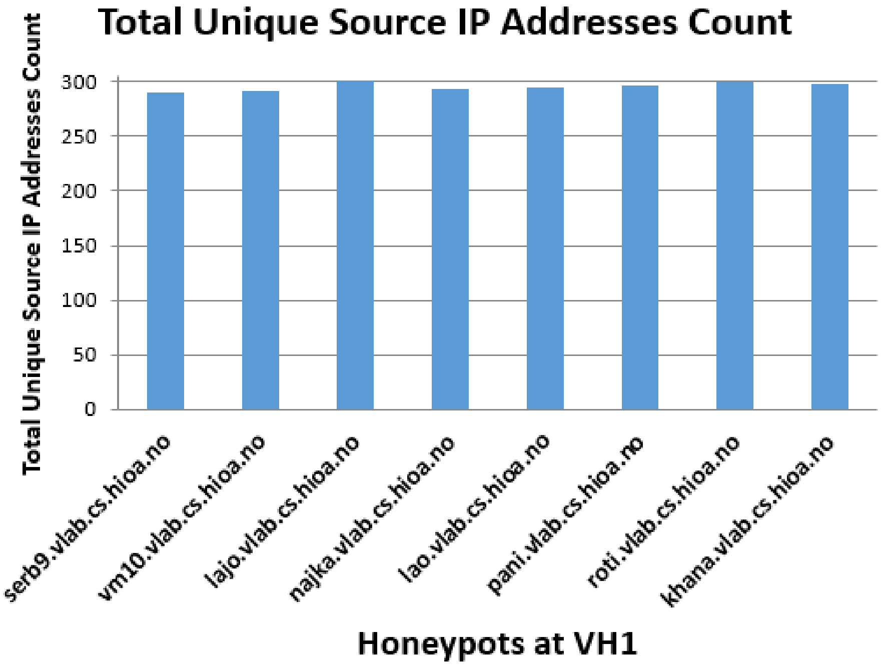


Figure 4.1: Total unique source IP at VH1

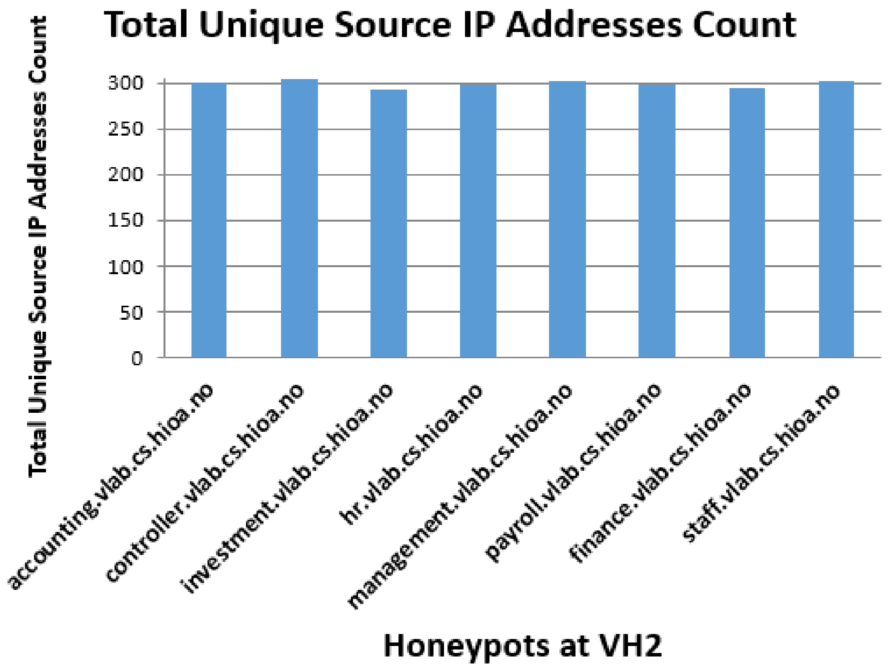


Figure 4.2: Total unique source IP at VH2

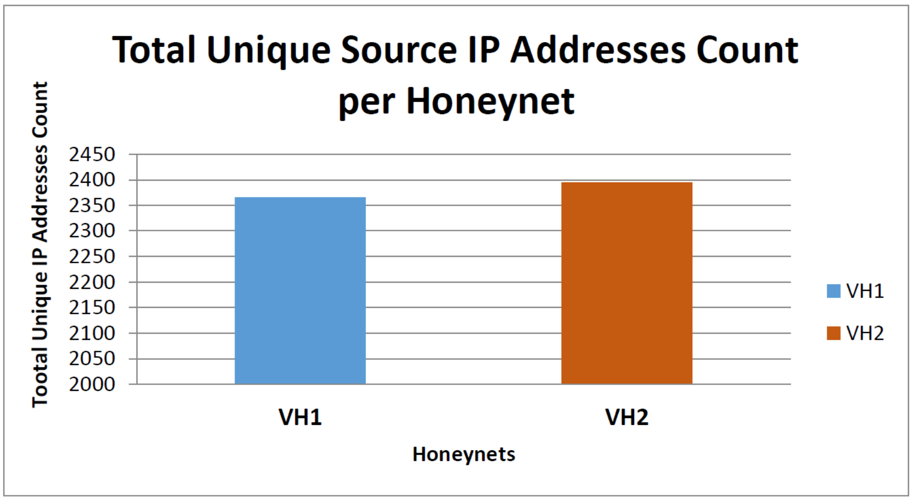


Figure 4.3: Total Unique Source IP addresses Attacked VH1 and VH2

During the observation period out of the total unique source IP addresses that attacked the honeynets, 3682 have attacked both honeynets. While the remaining were almost equally divided between the two honeynets, 524 unique source IP addresses have attacked only VH1 and 554 unique source IP addresses attacked only VH2. Figure 4.4 illustrates the percentages of attack.

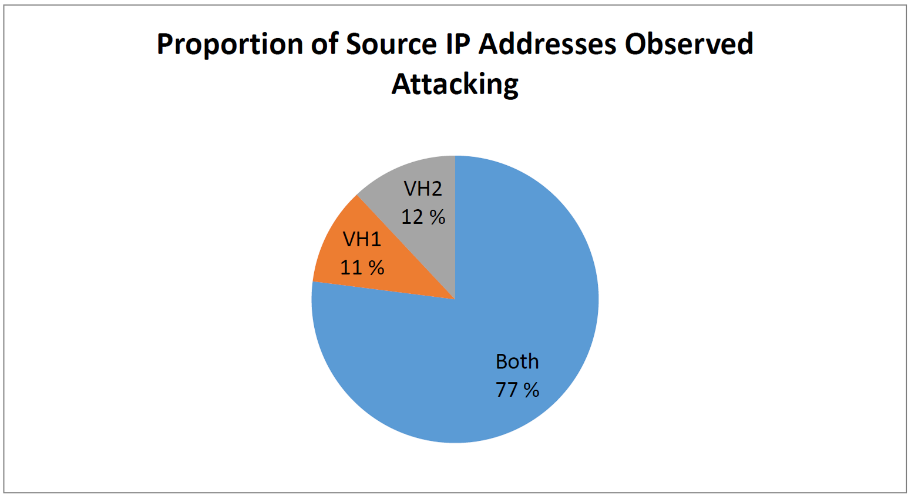


Figure 4.4: Proportion of Unique Source IP Addresses Attacked Honeynet

During the observation period of this project out of the total unique source IP addresses that attacked VH1, SSH log files have generated the most unique source IP addresses, 1114 of them. The log files that has contributed the second most are the apache log files with the count of 622 unique source IP addresses. The unique source IP addresses that tried to get illegitimate access only via HTTPS,FTP, MySQL, and SMTP log files were 56, 77, 60 and 42 respectively. Source IP addresses that attacked both SSH and HTTP were 14. Two source IP addresses were observed attacking SSH, HTTP and HTTPS during the period of observation. The rest of the total unique source IP addresses that attacked VH1 were seen in all log files which accounted 378.

From the total unique source IP addresses that attacked VH2, most of the attacks were SSH attacks. They generated 1125 of the total source IP addresses counts. Unique source IP addresses observed attacking only HTTP were 637 counts. While MySQL and SMTP have generated the same unique source IP addresses counts as that of their counterpart in VH1 that is 60 and 42 respectively. HTTPS and FTP log files have contributed 57 and 78 unique source IP addresses of the total count in VH2 respectively. Unique source IP addresses that were observed attacking both SSH and HTTP were 12 and 3 unique source IP addresses were observed attacking SSH, HTTP and HTTPS in VH2. The total unique source IP addresses observed attacking all the services with VH2 were the same as that of VH1 that is 378 unique source IP addresses. VH2 has seen 30 more unique source IP addresses than VH1 on the seventy days of observation.

4.5 Source IP per Destination Port

In computer networking communication between two nodes take place due to the existence of unique IP addresses and port number. Port numbers are designated for particular services that run within the computer. Cyber adversaries also use these ports to compromise computer systems. As mentioned in section 4.4 each service that were open and running in this project have made different contribution to the total count of source IP addresses to the honeynets. In this section the total number of source IP addresses that attacked each service during the seventy days observation is illustrated. In 4.4 any source IP address that attacked one or more service was counted as one. Therefore, it is obvious the total unique source IP addresses that is counted in this section will be higher compared to what is illustrated in that section as the count in this section is service based not destination based. 1508 source IP adddresses attacked VH1 via SSH attack while 1520 source IP addresses tried to compormise VH2 through SSH attack. Web application attacks over HTTP port 80 were the second most prevalent attack type during the observation period time. There were 1016 source IP addresses that attacked VH1 and 1030 source IP addresses tried to get illegitimate access at VH2 via HTTP attack. The source IP addresses that logged at HTTPS were 436 at VH1 and 438 at VH2. FTP log files have recorded 455 for VH1 and 456 for VH2. While MySql log files have logged 438 in both honeynets and SMTP log files logged 420 unique source IP addresses in both honeynets. Figure 4.5 depicts the total source IP addresses attacked each service.

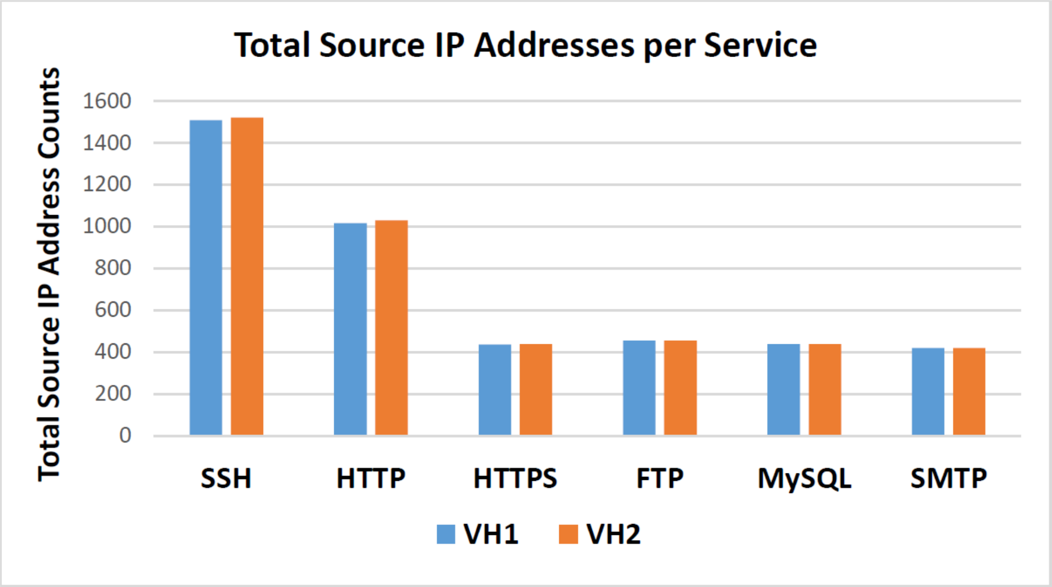


Figure 4.5: Total Unique Source IP Addresses Attacked Each Service

4.6 Connection Attempts

The total number of source IP addresses that attacked each service within both honeynets have been presented in section 5.2. However, throughout the project the number of connection attempts to break-in the honeypots were way bigger than the total unique source IP addresses presented in the above section. Because in section 5.2 unique source IP is counted only once no matter how many times it tried to break-in through a particular service.

The daily average of connection attempts to each service at each group during the experiment period may give a clear picture about the rigorous attempts that took place to break-in the honeynets. SSH login attempts were mostly used to break-in the honeynets during the seventy days observation. On average there were 606 daily login attempts on VH1 and 612 daily login attempts on VH2. The daily average attempt on HTTP was 57 to 62 for VH1 and VH2 respectively. The rest of the services on daily average had the same number of connection attempts for VH1 and VH2. Figure 4.6 depicts the daily average break-in attempts to the honeynets during the project period. This figure also depicts which services were most probed in order to get illegitimate access.

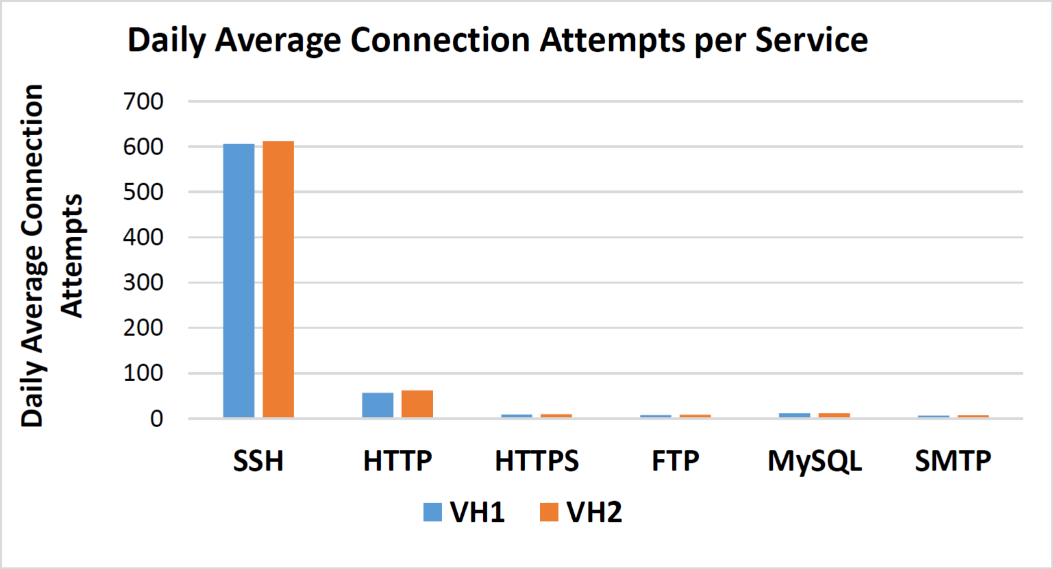


Figure 4.6: Daily Average Break-in Attempts per Service at VH1 and VH2

4.7 Unique Source IP per Operating System

As mentioned in section 3.3, in this project Centos version 5.4 and Ubuntu 10.04 were deployed to determine whether the type of platform will increase the chance of a host been a favored target. The intention of deploying the two operating systems was to see if a different OS makes a host more targeted than the other. The honeypots that had Ubuntu as their base OS in VH1 are named lajo, pani, roti and khana and in VH2 accounting, controller, management and staff had Ubuntu as their base OS. The rest of the honeypots were using Centos as their base OS. Based on the 70 days observation, systems that used Ubuntu as their base OS have seen slightly higher number of unique source IP addresses than their counterpart Centos based systems.

In VH1 the total unique source IP addresses that attacked honeypots that had Ubuntu as their base OS are 1196 while those attacked Centos based honeypots were 1169. In VH2 1210 unique source IP addresses were observed attacking honeypots that were using Ubuntu OS while 1185 unique source IP addresses were observed attacking honeypots Centos based honeypots. The total number of unique source IP addresses attacked each version of the OS in VH1 and VH2 during this project duration is presented in figure 4.7.

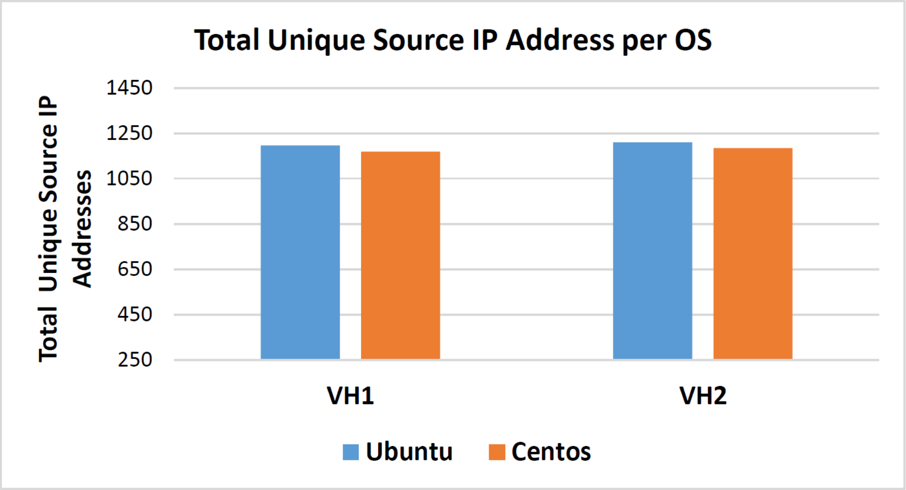


Figure 4.7: Unique Source IP addresses Attacked OS for VH1 and VH2

4.8 Previously Reported Malicious IP Addresses

As mentioned in section 3.5 a perl script was developed to retrieve information about the source IP addresses that attacked the honeynets from external sources. The two websites that are used for retrieving information about the geographical location and previous malicious nature are mentioned in 3.5. The information gathered about malicious IP addresses in ISC is collected from volunteers to throughout the world in order to fight back the most malicious abuser of Internet. Those IP addresses which are not found in the Dshield database do not mean that they are not malicious by nature, but as contributors to the database are volunteers the chance of all malicious IP address been reported is not gauranteed [70].

In this project a perl script was developed to check the source IP addresses that attacked the honeynets against Dshield database. As of writting, out of the total IP addresses that have attacked the honeynets deployed to address the problem statement of the research in the seventy days of observation, 33 % of IP addresses have been reported previously to Internet Storm center (ISC)[70] for their malicious activities.

4.9 Attackers Operating Systems

The Honeywall gateway uses passive fingerprinting techniques to find out the operating system of an attacking machine. This technique minimizes the risk of alerting the attackers compared to active fingerprinting. The tool that is incoparated in Honeywall gateway is an open-source fingerprinting tools known as p0f. During the seventy days of observation the most used operating system by the cyber adversaries attacked the honeynets was the family of Microsoft Windows which accounted to 58 % out of the total OS, followed by Linux family OS at 26 % and 16 % were not recognized. The lack of recognition of the 16 % attacking operating systems could be due to skilled hackers configured the systems OS to trick p0f. Figure 4.8 depicts the results in percentile.

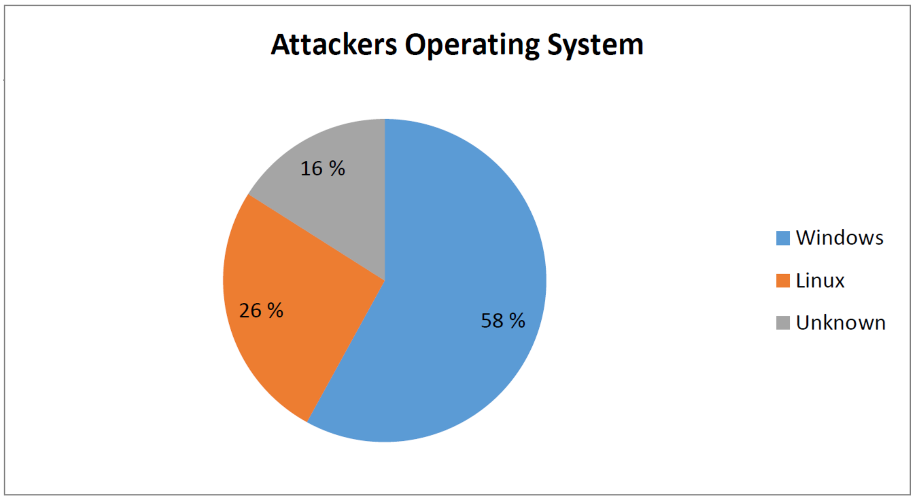


Figure 4.8: Attackers IP OS During Observation Period

4.10 Attackers Country of Origin

The attackers country of origin in this project were decided based on the country information about each source IP addresses collected from the Perl script that checks the status of source IP address against Dshield database, the country information revealed by the Honeywall gateway and a Perl script that was developed to check country of each source IP address observed attacking the honeynets in the seventy days against `http://www.geoiptool.com/`. All script used are attached as appendix.

Based on the results even in the total number of source IP addresses that attacked the Honeynets throughout the project period time, IP addresses originated from US has excelled on top by contributing 29.01% out of the total observed attacking the Honeynets. This was followed by source IP addresses from China with 19.14% and the third ranked country with 10.49% source IP addresses was the Netherlands.

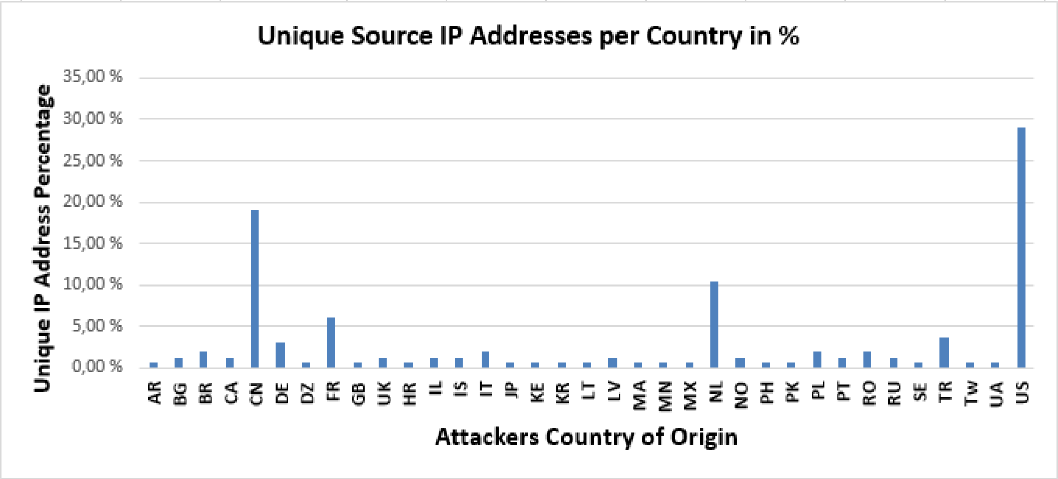


Figure 4.9: Source IP Adddresses Country of Origin in Percentage

depicts the percentile of source IP addresses per country of origin. The country of origin are represented in two letter country code as of ISO (International Organization for Standardization).

Chapter 5

Analysis

The previous section has shown the results that were found from the seventy days observation the number of attacks systems with attractive FQDN and non-attractive FQDN received throughout the project life time.

The results showed the unique source IP observed per destination and the total number of attacks each honeynet received on the due course of the observation time, the number of unique source IP addresses attacked each service in both honeynets, the number of connection attempts took place on the services, and the total of unique source IP addresses attacked Centos and Ubuntu based honeypots. The analysis of these results is presented in this section.

5.1 Attractive vs Non Attractive FQDN

As it has been illustrated in section 4.4 and depicted in figure 4.3,the total unique source IP addresses that were observed attacking VH1 during the time of this project were 2365 while those registered attacking VH2 were 2395. The difference of unique source IP addresses between VH1 and VH2 were only 30 unique source IP addresses that is VH2 got 30 more unique source IP addresses attackes. Taking into account the observation period length and the number of honeypots deployed in this project, this difference appears to be not significant.

Figure 5.1 depicts daily average of unique source IP addresses attacked VH1 and VH2 during the seventy days of observation in this project. On average there were 4.22 daily unique source IP attacked systems with non-attractive FQDN that are referred in this project as VH1 and there were 4.27 daily unique source IP attacked systems with attractive FQDN that are referred as VH2. On average there seems no significant difference between the unique source IP addresses that attacked VH2 and VH1.

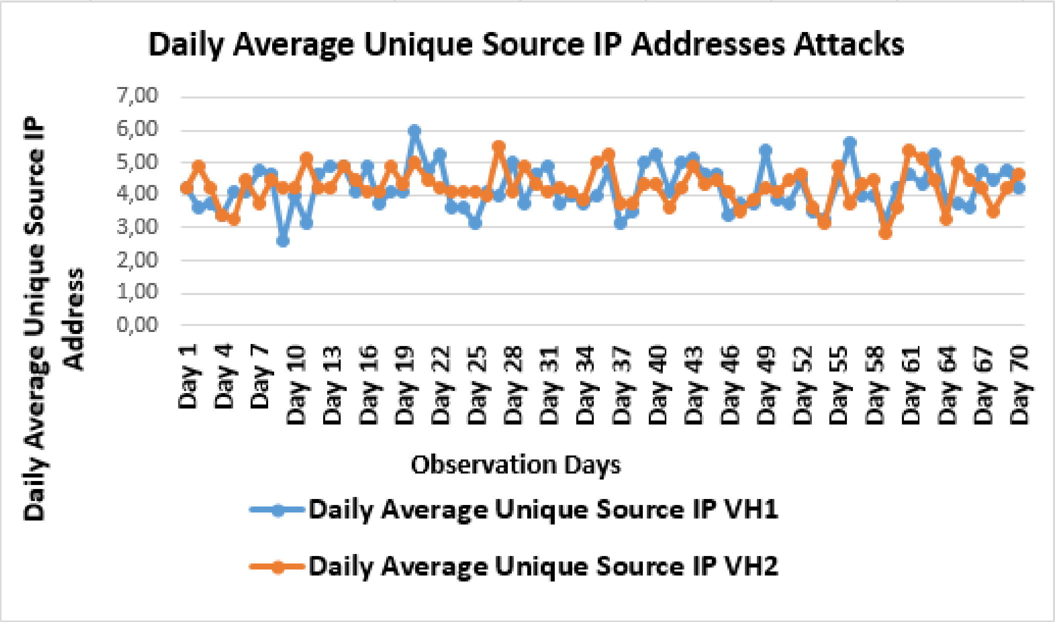


Figure 5.1: Daily Average of Unique Source IP Addresses Observed Attacking VH1 and VH2

5.2 Source IP per Destination Port

Section 5.2 has illustrated the total number of unique source IP addresses that attacked each service in VH1 and VH2. Section 5.2 also showed the count difference between VH1 and VH2 for each service. According to the results found in this project, the most used ports to get illegitimate access were SSH and HTTP. The other services that were monitored in this project had seen less than 500 unique source IP addresses throughout the observation time on both VH1 and VH2. VH2 has seen 2 more unique source IP addresses in HTTPS attacks than that of VH1 and 1 more unique source IP address in FTP attacks comparing to FTP attacks in VH1. This difference appears to be not a significant difference due to the time length and the number of honeypots involved in this project.

The number of unique source IP addresses that attacked MySQL and SMTP were the same for both VH1 and VH2. The number of connection attempts that took place on these services were also the same on average as shown in 4.6. Therefore, the analysis is going to emphasize on SSH and HTTP attacks only.

5.2.1 SSH ATTACKS

SSH attack was mostly used throughout the observation period. 1508 unique source IP addresses have attacked VH1 and 1521 unique source IP addresses have attacked VH2. VH2 has seen 13 more unique source IP addresses than VH1. This difference also appears to be not significant with the time length of the project.On average daily the unique source IP addresses that attacked VH1 on due course of this project were 2.69 while 2.71 unique source IP addresses attacked VH2, about 3 unique source IP addresses each day for each group. On average there was equal unique source IP addresses attacking VH1 and VH2 daily during this project period. This shows that regardless the FQDN a networked computer system has, the chance of been attacked on average is approximately the same.Figure 5.2 depicts the daily average of unique source IP addresses attacked through SSH attack to compromise systems in VH1 and VH2 during the seventy days of observation period.

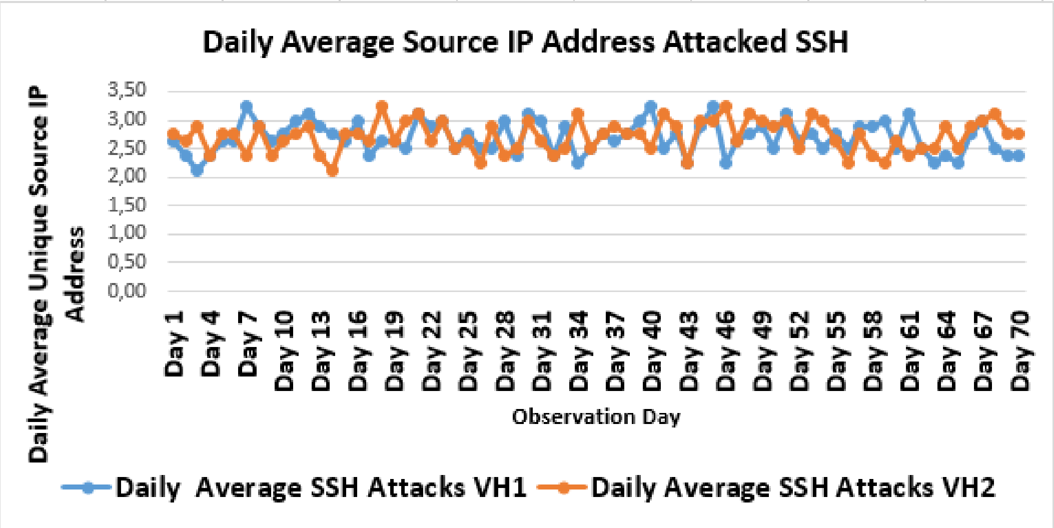


Figure 5.2: Daily Average of Unique Source IP Addresses Attacked SSH VH1 and VH2

5.2.2 HTTP ATTACKS

The second most attacked port during this project was HTTP port via web application vulnerability. As it was mentioned in section 5.2, the total number of unique source IP addresses that attacked VH1 through HTTP attacks during this observation period were 1016 and 1030 source IP addresses attacked VH2. VH2 has got 14 more unique source IP addresses that attacked HTTP than that of VH1. This makes on average daily there were 1.81 unique source IP addresses attacked VH1 and 1.84 unique souerce IP addresses attacked VH2

HTTP attacks. This difference appears to be not significant taking into consideration the period of the observation.

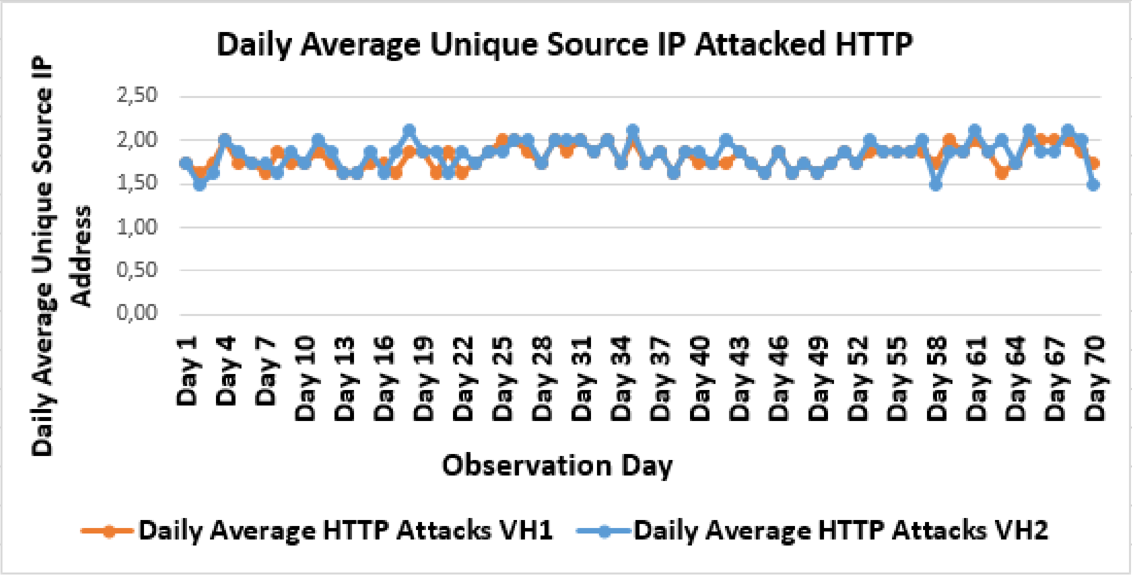


Figure 5.3: Daily Average of Unique Source IP Addresses Attacked HTTP VH1 and VH2

Figure 5.3 depicts the daily average of unique source IP addresses that attacked VH1 and VH2 during the seventy days of observation.

5.3 Number of Connection Attempts

The other comparison criteria to addresses the problem statement was to see if the number of connections attempts by the source source IP addresses took place to compromise VH1 and VH2 during the observation period. The average number of connection attempts against VH1 and VH2 during this project life time was illustrated in section 4.6. On average daily there were 6 more SSH login attempts to VH2 than VH1. This difference appears to be not sufficient enough to conclude that attractiveness of FQDN makes a computer system to be a preferred target by cyber intruders when it comes to SSH attack.

The daily average of connection attempts via web vulnerability on HTTP was 57 to 62 for VH1 and VH2 respectively during this project life time. On average daily VH2 has seen 3 more HTTP connection attempts VH1. This difference appears also to be not significant difference to conclude that attractiveness of FQDN makes a networked computer system more targeted by cyber adversaries for HTTP attacks.

5.4 Attacks per Operating System

The total unique source IP addresses that attacked honeynets based on OS was presented in section 4.7. Those unique source IP addresses attacked honeypots with Ubuntu as their base OS in VH1 were 1196 and those attacked Centos based honeypots were 1169. On average daily there were daily 4.27 unique source IP addresses attacked Ubuntu based honeypots and those unique source IP addresses that attacked honeypots which used Centos OS were 4.18. The difference between the the number of unique source IP addresses that attacks Centos and Ubuntu Oses based honeypots in VH1 on average daily appears to be not significant.

The total number of unique source IP addresses that attacked hosts with Ubuntu operating system were 1210 and 1185 unique source IP addresses attacked honeypots with Centos operating system in VH2 in the seventy days of observation. On average daily there were 4.32 unique source IP addresses that attacked Ubuntu based honeypots and 4.23 unique source IP addresses attacked Centos based honeypots in VH2 on due course of this project. The difference between the the number of unique source IP addresses that attacks Centos and Ubuntu Oses based honeypots in VH2 on average daily appears also to be not significant.

5.5 Statistical Analysis

The above sections of the analysis have shown descriptively that the during the seventy days of observation the attractiveness of FQDN of a networked computer systems does not seem to make a computer to be a more preferable target than computer systems with non-attractive FQDN significantly by hackers. The no significant difference between the number of unique source IP addresses attacked honeypots with attractive FQDN and the number of unique source IP addresses attacked honeypots with non-attractive FQDN will be analysed using statistical method. The statistical analysis also will be applied to verify the no significant difference between the average number of attacks agianst SSH and HTTP as well.

Based on the number of unique source IP addresses that attacked each group of honeynet for a continuous seventy days on daily bases, the average number of unique source IP addresses calculated for 24 hours for the two groups that consists eight honeypots each.

The standard deviation of the number of attacks for attractive FQDN and non-attractive FQDN are assumed not equal. The research hypothesis is tested using student t distribution. Since the sample size is large, n=70, the Central

Limit Theorem guarantees that the difference between the mean number of attacks that an attractive FQDN receive and the mean number of attacks that non-attractive FQDN receive has approximately a normal distribution regardless of the nature of their respective population distribution.

Hypothesis stating and Assumptions

- H_0: μ_1-μ_2 = 0
- H_1: μ_1-$\mu_2 \neq 0$
- The two groups are independent as the attacks in one group does not affect the other and elements in the groups are not paired.
- Each observation day is independent.
- Attackers are random as they find the honeynets in their own way.
- Variances are known to be equal
- Significance level (α) is 0.05 .

In a statistical test, the significance difference between two groups is determined by the calculated p-value. P-value is a number which lies is between 0 and 1. If the computed p-value is closer to 1, this indicates there are no significant difference between the two groups. If the p-value is closer to 0, this implies that the means of the two groups of data are significantly different [75].

A two sided independent t-test was conducted using R statistical software. $\mu_1$represents the mean for attractive FQDN and μ_2 represents the mean for non-attractive FQDN. The results and its interpretation is presented below.

5.5.1 Attractive vs Non Attractive FQDN

Based on the daily average of unique source IP addresses that attacked attractive FQDN and non-attractive FQDN via all the services that were open and running throughout the project period time. Table 5.1shows the sample statistics: Sample sizes for Attractive FQDN and for Non-Attractive FQDN, the daily mean of unique source IP addresses attacked VH2 and VH1, sample standard deviation and standard error mean for VH2 and VH1 respectively.

	Attractive FQDN	Non Attractive FQDN
Sample Size	n_1 = 70	n_2 = 70
Sample Mean	$\bar{x_1}$= 4.279286	$\bar{x_2}$ = 4.225571
Sample Standard deviation	0.5383343	0.671748
Sample Standard Error Mean	0.06434326	0.08028924

Table 5.1: Computed t-test Values for VH2 and VH1

The computed p-value is 0.6025 which is greater than the level of significance α= 0.05 that was chosen for this project. There is sufficient sample evidence to support the claim that, on average, there is no significant difference between the number of attacks that computers with attractive FQDN receive and the average number of attacks that computers with non-attractive FQDN receive.

5.5.2 SSH ATTACKS

The computed values of the number of unique source IP adddresses that attacked attractive FQDN and non-attractive FQDN via SSH attack in the seventy days of observation are presented in 5.2.

Table 5.2 shows daily mean of unique source IP addresses attacked SSH at VH2 and VH1 respectively, sample standard deviation and standard error mean for VH2 and VH1 respectively.

	Attractive FQDN	Non Attractive FQDN
Sample Size	n_1 = 70	n_2 = 70
Sample Mean	$\bar{x_1}$ = 2.716714	$\bar{x_2}$ = 2.695143
Sample Standard deviation	0.2742459	0.2828465
Sample Standard Error Mean	0.03277865	0. 0.03380663

Table 5.2: Computed T-Test Values for SSH Attacks at VH2 and VH1

The computed p-value for SSH attacks for attractive FQDN and on-attractive FQDN is 0.6476 which greater than the significance level choosen on this project. This confirms that the claim, on average, there is no significant difference between the number of SSH attacks that computers with attractive FQDN receive and the average number of attacks that computers with non-attractive FQDN receive.

5.5.3 HTTP ATTACKS

The second most used port to compromise the honeypots by the adversaries in this project was HTTP at port 80 via web application vulnerability. The computed values of the number of unique source IP addresses that attacked HTTP service at VH2 and VH1 are presented in 5.3.

	Attractive FQDN	Non Attractive FQDN
Sample Size	n_1 = 70	n_2 = 70
Sample Mean	$\bar{x_1}$ = 1.842143	$\bar{x_2}$ = 1.816714
Sample Standard deviation	0.156695	0.1230968
Sample Standard Error Mean	0.01872863	0.01471288

Table 5.3: Computed T-Test Values for HTTP Attacks at VH2 and VH1

The computed p-value for HTTP attacks for attractice FQDN and non-attractive FQDN is 0.2876 which is also greater than the significance level choosen for this project that is 0.05. There is sufficient sample evidence to support the claim that, on average, there is no significant difference between the number of HTTP attacks that computers with attractive FQDN receive and the average number of attacks that computers with non-attractive FQDN receive.

5.5.4 Comparing within Groups

To analyze the equality of the mean within each group, that is within the attractive FQDN and within non Attractive FQDN an analysis of variance, ANOVA specifically a single factor ANOVA was performed in Microsoft Excel. The analysis is conducted to verify if there is a significant difference between the number of attacks the received within the same group.

Anova: Single Factor						
SUMMARY						
Groups	Count	Sum	Average	Variance		
Accounting	70	301	4,30	4,65		
Controller	70	304	4,34	3,76		
Investment	70	292	4,17	2,84		
Hr	70	299	4,27	3,48		
Management	70	303	4,33	2,63		
Payroll	70	299	4,27	2,93		
Finance	70	295	4,21	3,36		
Staff	70	302	4,31	2,22		
ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	1,683929	7	0,240561	0,074416	0,999358	2,026155
Within Groups	1784,414	552	3,232635			
Total	1786,098	559				

Table 5.4: ANOVA Sigle Factor Test For VH2

The list of name under column Groups are hostnames for the attractive FQDN. As it can be seen from 5.4 comparison of mean number of attacks received by each honeypot with attractive FQDN via ANOVA test also demonstrated that there is no significant difference (F=0.0744, p=0.9993). On the summary section the averages and variances results have been rounded for clarity of purposes. The lists under column Groups on 5.5 are hostnames under the domain vlab.cs.hioa.no for the non-attractive FQDN. A comparison of mean number of attacks received by each honeypot within the non-attractive FQDN via ANOVA test also demonstrated that there is no significant difference (F=0.0655, p=0.9995). On the summary section the averages and variances results have been rounded for clarity of purposes.

Anova: Single Factor

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
serb9	70	290	4,14	3,20
vm10	70	291	4,16	3,24
lajo	70	301	4,30	3,20
najka	70	293	4,19	3,83
lao	70	295	4,21	4,37
pani	70	297	4,24	3,23
roti	70	300	4,29	3,69
khana	70	298	4,26	4,11

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	1,65535714	7	0,236479592	0,065548823	0,99957796	2,026154933
Within Groups	1991,44286	552	3,607686335			
Total	1993,09821	559				

Table 5.5: ANOVA Sigle Factor Test For VH1

Anova Test For SSH ATTACKS

The Anova test for the equality of mean number of SSH attacks each honeypot received within VH1 and VH2 has also demonstrated that there is no significant difference between the number of SSH attacks received by each honeypot at VH1 (F=0,161439; p=0,992367) and (F=0,045855; p=0,999872) for VH2. The calculated Anova test using Microsoft Excel for VH1 and VH2 for SSH attacks are attached on the appendix section of the document.

Anova Test For HTTP ATTACKS

The equality of the mean within each group using Anova single test for HTTP attacks has also demonstrated that there is no significant difference between the number of attacks received by each honeypot at VH1 (F=0,123284, p=0,996708) and (F=0, 104505, p=0,998058) at VH2. The calculated Anova test using Microsoft Excel for VH1 and VH2 are attached on the appendix section of the document.

5.6 Attackers Profile

5.6.1 Malicious IP Addresses

Section 4.8 has mentioned the result retrieved from www.dshield.org database about previous status of the IP addresses that have attacked the honeynets during the observation period. Out of the total IP addresses that have attacked the honeynets in the seventy observation days, 33 % of them have been reported as been malicious. Out of the 33 % previously reported, IP addresses that orginated from US are ranked first by contributing 10.71 % followed by IP addresses from China with 8,04 %. Figure 5.4 depicts the percentile each country contributed to the 33% that was explained in section 4.8.

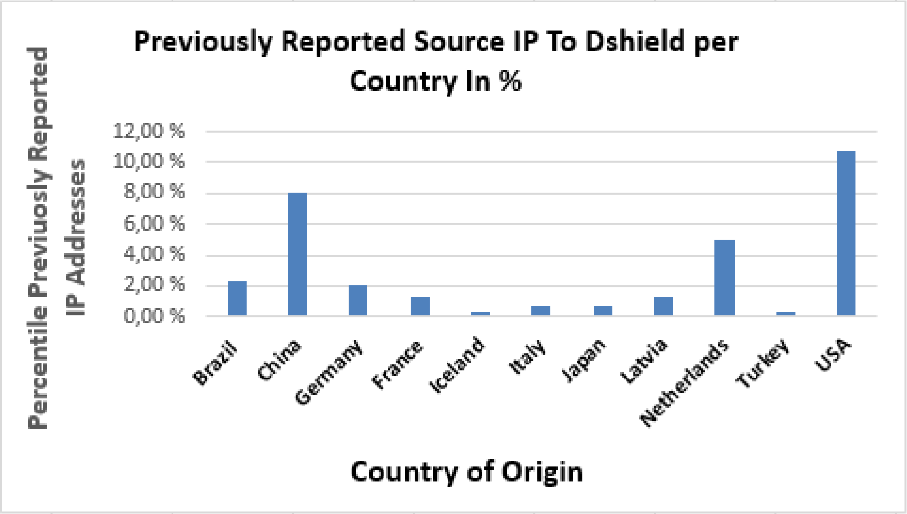


Figure 5.4: Reported Malicious IP Addresses per Country in Percentage

The top ten IP addresses that have made the highest number of attempts in the honeynets on SSH and HTTP and that have been reported to be malicious according to information retrieved on Dshield are presented in 5.6and 5.7 respectively. The source IP address with the highest number of connection attempts is listed on top of the tables for both services.

IP Address	Hostname	Country	Reports	Target Port
219.149.138.230	219.149.138.230	CN	3966	22
82.221.105.6	82.221.105.6	IS	151243	22
5.79.78.230	hosted-by-kingrdp.com	NL	16332	22
198.20.69.98	singlehop1.shodanhq.com	US	289090	22
209.126.230.71	internetsurvey-2.erratasec.com	US	473720	22
221.230.54.115	221.230.54.115	CN	42070	22
46.218.179.49	reverse.completel.fr	FR	117713	22
61.160.251.136	61.160.251.136	CN	30871	22
212.116.159.146	212.116.159.146	BG	31296	22
211.95.76.242	211.95.76.242	CN	21605	22

Table 5.6: Top 10 IP Address Attacked SSH During Project Period and Their Status in Dshield

IP Address	Hostname	Country	Reports	Target Port
189.38.56.67	renacor02.dominiotemporarioidc.com	BR	12410	80
113.57.188.106	arpa.hb.cnc.cn	CN	21042	80
183.60.48.25	183.60.48.25	CN	52822	80
92.240.68.152	92.240.68.152	LV	79577	80
92.240.68.153	92.240.68.153	LV	80799	80
198.20.69.74	singlehop2.shodanhq.com	US	639407	80
198.20.70.114	singlehop3.shodanhq.com	US	436358	80
94.102.49.211	94.102.49.211	NL	498662	80
192.151.144.234	192.151.144.234	US	346286	80
66.7.220.78	dimer.hostdimer.com	US	6184	80

Table 5.7: Top 10 IP Address Attacked HTTP During Project Period and Their Status in Dshield

5.6.2 Country of Origin

Section 4.10 has shown the country of origin of the source IP addresses that have attacked the honeynets and the percentage each country contributed to the total. The country of origin of a source IP address was issued in this document based on the information retrieved from the three source that were mentioned in section 4.10. In the seventy days of observation period, some of the IP addresses that attacked the honeynets belong to the same domain name or network address. The top five domains that have participated in attacking the honeynets with the highest number of IP addresses are presented 5.8 alongside the country of origin of the IP address. As some of the network addresses had no domain names while retrieving information about the attackers from the three databases on the due course of the project. For those network addresses with no domain names, a blank space is left under the Domain Name field of the table.

Domain names orginated from US has contributed the highest number of source IP addresses from the same network addresses (domain name). The US based amazon instances have made a total of ten IP addresses from the domain name compute-1.amazonaws.com. Based on the analysis in the Honeywall gateway

Network Address	Domain Name	Country	Source IP Addresses Count
200.165.0.0/17 (200.165.0.0-200.165.127.255)	medquimica.com.br	BR	3
113.56.0.0/15 (113.56.0.0-113.57.255.255)	arpa.hb.cnc.cn	CN	3
92.240.64.0/19 (92.240.64.0-92.240.95.255)		LV	2
198.20.64.0/18 (198.20.64.0-198.20.127.255)	shodanhq.com	US	6
54.221.0.0/16 (54.221.0.0-54.221.255.255)	compute-1.amazonaws.com	US	10

Table 5.8: Top 5 Networks with Highest Number of Hosts Attacked Honeynets

log files about network traffic to the honeynets, four of the IP addresses from shodanhq.com had attacked the sixteen honeypots via TCP SYN flooding attack. This domain is also USA based.

TCP SYN Flooding attack is one of the widely used denial-of-attack tactics to compromise network systems [37]. TCP SYN Flooding attack manipulates the three-way handshake of TCP connection mechanisms, but does not send the ACK packet after receiving the SYN+ACK packet from the listener for its first packet which is SYN. The three-way handshake in TCP connection establishment has been explained on the section (Network Scan) and is also illustrated by figure on the same section. The attackers flooded the target machine with SYN packets directing at HTTP port in order to cause denial-of-service [36].

The aim of this attack is not overloading network resources or memory of the host, but exhausting the backlog of the server which is associated with the port number due to full of bogus half-open connection in order to which results the rejection of legitimate SYN segments. This attack is often carried out by spoofed IP addresses which will not verify the SYN+ACK from the listener for the initial SYN packet it has received [41].

The remain two IP addresses from shodanhq.com domain have involved in brute force attack on SSH. One of the two which attacked the honeynets via brute force attack with the hostname singlehop2.shodanhq.com and IP address 198.20.69.98 is one of the most attempted IP addresses that tried to compromise the honeynets with username and password login attempts via port 22 (SSH service). Brute-force attack is another common attack method that tries to break a network system by trial and error mechanism. Nowadays such attacks are performed by automating a consecutive guesses to find a pair of user name and password combination in order to get access to network systems [37].

Chapter 6

Discussion

The previous section has analysed the results that were found from this setup to address the problem statement. This chapter will evaluate the project process in general and observations that during the process. It also highlights future work.

6.1 Setup Tradeoffs

The setup enabled to collect data that are essential to answer the problem statement of the research questions. The data collected included unique source IP, initial and final timestamp, destination port, and the number of connection attempts a particular source IP addresses made to a specific destination IP on the services that were open and running within it. This data collection was done every mid-night for a continuous seventy days on sixteen honeypots which made two honeynets. Each honeynet had eight honeypots that were hosted in one physical server. Eight of the honeypots were give an attractive FQDN while the other eight were given non-attractive FQDN. The honeynets were launched at the same time and where running for a continuous seventy days.

This research was performed in a local network at HiOA network in a controlled setup. The honeynets were controlled through the Honeywall gateway which was able to monitor every traffic movement from and to the honeynets that were assigned attractive FQDN and non-attractive FQDN. The IP addresses that were allocated to carry out the research were from the same subnet from the IP pools of HiOA network and they were a sequential IP numbers. This might have an impact that the number of unique source IP addresses attacked both honeynets indeed be largely the same. If it was possible to run some honeypots in a different subnet under the same domain, it would have enhanced the data collection and analysis to develop a wider knowledge out of the project while running it still under controlled environment. However, this was not done as there was only one subnet allowed to be used on this project by HiOA due to the risks that honeypot projects involve.

As depicted in 4.4 out of the total number of unique source IP addresses which attacked the services that were open and running on due course of this project, the 77% had attacked both honeynets. Though this might seem as a deficiency for not getting a wider view about global image by setting up a number honeynets in different geographical location via cloud computing, but it also gives the opportunity to have a full control over the observation environment as data collection needs to be precise. As stated in [71] inaccuracy in data collection will lead to unreliable and biased decision. The other reason for setting up a local honeynets were to avoid the number of attack counts that could generate due to targeted attacks that could be due to some political or other non-constant phenomena that could lead to cyber-attacks. The continuous cyber-attacks that targeted Australian web site in the months of October and November of 2013 by a hacking group called Anonymous Indonesia [8] is a good example of non-constant cyber-attack phenomena. Data collected in such incidence could not be reliable to make a general decision for the events that take place in the Internet. Therefore, to address the problem statement of this research setting up local honeynet with the same setup configuration was the preferred option.

The other option that might had increase the number of attacks the honeypots could receive was to deploy them under the second top level domain which is hioa.no. The subdomain the honeypots deployed was vlab.cs.hioa.no which is pretty obvious to cyber adversaries that the hosts are dedicated for educational domain for computer labs. This might also play a role on the decreasing the attractiveness of the host or if there really are hackers looking after an attractive FQDN, these hosts will not be much attractive that they could be if they were hosted under hioa.no The other option that might had increase the number of attacks the honeypots could receive was to deploy them under the second top level domain which is hioa.no. The subdomain the honeypots deployed was vlab.cs.hioa.no which is pretty obvious to cyber adversaries that the hosts are dedicated for educational domain for computer labs. This might also play a role on the decreasing the attractiveness of the host. If there really are hackers looking after an attractive FQDN, these hosts will not be much attractive that they could be if they were hosted under hioa.no. If it was possible to acquire more attractive domain names such as for financial institutions domain like bank.no, the attractiveness of the FQDN might had an impact. However, this was not possible on this project so the research was only able to be carried out in what was available.

6.2 Prior Experiment

Before the launch of the honeynets to address the problem statement, there was a trial experiment conducted using four public IP addresses. The two IP addresses were assigned FQDN and the two were bare IP addresses without domain names. The purpose of the trial was the same as the research questions, but the two IP addresses were not given any domain name and the observation were only done to SSH and HTTP attacks. The trial experiment was performed for seven days. Based on the seven days trial experiment results, a brief analysis on the number of attacks were performed. Though, this experiment cannot be used to make a conclusive decision but mentioning the findings is worth. The results and brief analysis of this trial experiment also showed that on average the number of attacks IP addresses with FQDN got was not significantly different than those IP without domain names got via SSH and HTTP attacks. The IP addresses that were used for the trial experiment were not used to perform the research work.

6.3 Sebek Client

One of the main objectives behind the design of honeypots was to learn about the hackers activities after they are successfully compromise the system [4]. This could have been achieved by installing a kernel level keylogger on the honeypot. The Honeywall project have developed a data capture tool known as Sebek. This data capture tool has a client and server side. The server side of Sebek is incorporated with the Honeywall CDROM Roo but the client side of it is installed in the honeypots. Sebek client is intended to capture all attackers activity such keystrokes, uploaded files and send them to Sebek server using a dedicated UDP port.

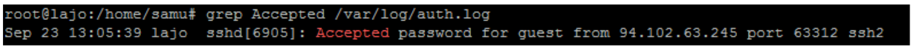
This type of keylogger could minimize the risk of alerting experienced hackers from realizing that they are being deceived. Though securely copying honeypot log files to a remote server could have been implemented to track the activity of the hackers after successful attack, but this is not a guaranteed solution specially when dealing with professional hackers as they can announce the existence of the honeynet which will reduce the value of the honeynet. If the honeynet is identified, attackers might ignore it or launch high-level attacks against the network that might interrupt the data collection process to address the problem statement or fill bogus data which could lead to a wrong conclusion [77].

In this project the installation of Sebek client was attempted in the current and older versions of Linux family as well in Windows OS. However, the attempts was not successful. On the due course of the project life time, the Sebek client version for both Linux and Windows OS is outdated and no further work has been done to make the client side of data capture tool made by the Honeywall project for over five years. Some of the related previous research papers [11, 60] that were reviewed on due course of the project had also stated that the attempt made to install Sebek client was not successful, though those papers stated that they only tried to install it on latest and one version of Linux OS.

6.4 Compromised Systems

During the project life time there were in a total eleven honeypots compromised through SSH brute force attacks. As a compromised systems could be used by cyber adversaries as attacking tool to legitimate systems on the Internet, any compromised honeypot in this project was closely followed but allowed to run for few days before taking it down. This was done to avoid any suspicion by successful attacker that there is a honeynet in the given network. The other reason for letting the compromised honeypots running for a limited period of time was to see if there is any trace of the attackers on the system that can be analysed to learn about the activities of the hackers. However, there were no much interesting traces that were retrieved after analysing the compromised honeypots to be reported except the most commonly used Linux commands like ”ls”, ”w” etc. that have been constantly reported on several related previous research papers that have been reviewed on due course of this project. Therefore, the author decided just for demonstration purpose only to present about two of the successful break-in in via SSH brute force .

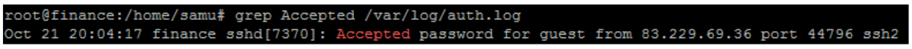
The first honeypot that was compromised on this project was after three days of the honeynets been launched. The successful attack was carried out via remote login through SSH on Sept 23 at 13:05:39 from a source IP address 94.102.63.245 and source port 63312. The source IP is originated from Netherlands. The username and password used to break-in the honeypot was ”guest” and guest. Most of the weak username and password combination that was used in this project were retrieved from previous honeypot projects [60, 78] that were conducted in the same premises as this project was performed.



```
root@lajo:/home/samu# grep Accepted /var/log/auth.log
Sep 23 13:05:39 lajo  sshd[6905]: Accepted password for guest from 94.102.63.245 port 63312 ssh2
```

Figure 6.1: Successful Login Attempt via SSH at lajo

The second honeypot that was compromised via SSH brute force attack was a honeypot with non-attractive FQDN. The IP address that compromised the honeypot was 83.229.69.36 and source port 44796 and the attacker had succeed after several attempts for a period of one month since the launch of the honeynets. The successful login was on 21st Oct 2013 at 20:04:17 via username ”guest” and password ”Guest1”. This addition of capital and a single digit took the hackers one month to successfully break-in to the system.

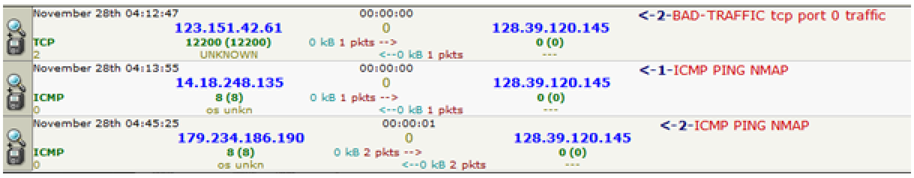


```
root@finance:/home/samu# grep Accepted /var/log/auth.log
Oct 21 20:04:17 finance sshd[7370]: Accepted password for guest from 83.229.69.36 port 44796 ssh2
```

Figure 6.2: Successful Login Attempt via SSH

6.5 Port Zero

One of the observation from the network traffic to and from the honeypots which was logged by the Honeywall gateway was the netwrok traffic to port zero. In the seventy days period of this project is the heavy incoming network traffic to port zero via TCP, UDP and ICMP protocols were observed. Though the three protocols have been used by the adversaries to connect to port zero but TCP connection attempts dominated most. According to Internet Assigned Numbers Authority (IANA), port 0 is reserved meaning that there should no any service running on it. Cyber adversaries can use such traffic to fingerprint victims OS as different Oses could reply to port zero traffic differently. However, the heavy incoming network traffic to port 0 can be sign of a possible reconnaissance attack. The network traffic to port zero during this project period have excelled all the services that were monitored except network traffic to SSH and HTTP services. Figure 6.3 illustrates the network traffic to port zero via TCP and ICMP protocols at a particular destination IP from three different source IP addresses that took place on November 28.



```
November 28th 04:12:47                 00:00:00                             <-2-BAD-TRAFFIC tcp port 0 traffic
                  123.151.42.61           0            128.39.120.145
TCP               12200 (12200)   0 kB 1 pkts -->          0 (0)
2                   UNKNOWN              <--0 kB 1 pkts     ---
November 28th 04:13:55                 00:00:00                             <-1-ICMP PING NMAP
                  14.18.248.135           0            128.39.120.145
ICMP                  8 (8)       0 kB 1 pkts -->          0 (0)
0                    os unkn             <--0 kB 1 pkts     ---
November 28th 04:45:25                    00:00:01                             <-2-ICMP PING NMAP
                   179.234.186.190           0            128.39.120.145
ICMP                    8 (8)        0 kB 2 pkts -->          0 (0)
0                      os unkn              <--0 kB 2 pkts     ---
```

Figure 6.3: Snapshot Network Traffic to Port Zero

6.6 Future Work

This research was carried out in a local network controlled setup. The domain name that the honeypots shared was vlab.cs.hioa.no which was an educational institute domain. The research has shown the number of attackers and severity of conncetion attempts that hosts with attractive FQDN gets are approximetly the same with host with non-attractive FQDN under the same domain. However, in order to verify whether an attractive FQDN will make a host more targeted by cyber adversaries or not needs to investigated. Therefore, some potential research areas emerged from this project are listed below.

- On this project most of the attackers were observed attacking the sixteen honeypots. Investigating if this scenario applies to other subnets as well could help in clarify if the attacks are the targeting a specific subnet under the same domain as the subnet could represent different administrative group within an organization.

- Set up the same testing environment with the same configuration under different domains. For example in financial institution domains.
- A trial experiment of the number of attacks IP addresses with no domain name and IP addresses with domain name received via SSH and HTTP attacks was conduct prior to the experiment, but the experiment duration was not reasonable enough to make a conclusion. Therefore, expanding the experiment period could enable to figure out if there is a difference in the number and severity of attacks due to domain name.
- The Honeywall project has made a remarkable work in building the Honeywall CDROM ROO which incorporates server side data capture tool or Sebek server. However, the client side of Sebek is not compatible with current OS versions. Making Sebek client working with current OS will give security professional and network adminstrator more information about hackers motivation, tactics and activities.
- During the period of this project a noticable heavy traffic to port zero has been observed via TCP, UDP and ICMP protocols. Recently the network traffic at port zero has created attention to network security professionals also. Therefore, conducting a research on this will enhance the security measure needed to be taken.
- Identify malicious IP addresses at local network and create automated reporting mechanism to ISC in order to create a public awareness about the IP addresses that are malicious in nature.

Chapter 7

Conclusion

The problem statement of this research work was to address if servers with attractive FQDN will be more targeted than servers with non-attractive FQDN. To address the problem statement, a controlled local virtual honeynet was set up. The two groups of honeynet had each eight honeypots with public IP addresses. The eight honeypots within group virtual honeynet 1 (VH1) were assigned non-attractive FQDN while the other eight honeypots under virtual honeynet 2 (VH2) were assigned attractive FQDN. These two groups of honeynet were online for consecutive seventy days. There were six services that were open and running on the due course of this project. The services that were monitored under this project were SSH, HTTP, HTTPS, FTP, MySQL and SMTP in both honeynets and the results and statistical analysis presented in 4 and 5 shows that the attack frequency is not affected by the attractive FQDNs. Thus, validating the null hypothesis there is no difference between the average number of attacks that servers with attractive domain names receive and the average number of attacks that servers with non-attractive domain names receive.

Following are the summarized answers to the questions defined in section 1.2.

1. The finding shows that the attractiveness of fully qualified domain name does not create a significant difference on average in the number of unique source IP addresses attacked the honeynet with attractive FQDN than that of the average number of unique source IP addresses attacked the honeynet with non-attractive FQDN. In this regard the attractiveness of the FQDN have not made the servers more targeted than attackers were not targeting the honeypots based on their names.

   Analysis on service level was also conducted on the two most attacked services in this project which were SSH and HTTP. The analysis conducted on attacks per service level have also not shown a significant difference between the average number of attacks received by the services at honeynet with attractive FQDN and the average number of attacks received by the services under honeynet with non-attractive FQDN. The other analysis conduct was to find out if there was a significant difference between the number of attacks received within the same group of honeynet that is the attractive and non-attractive FQDN through Anova single factor test. This result also shows that there is no significant difference between the number of attacks each honeypot received.

All the analysis conducted in this regard shows that the attacks were not targeting for the attractive FQDN rather randomly attacking for any vulnerable system.

2. In each honeynet there were four honeypots that had Centos as a base OS and the other four honeypots had Ubuntu as a base OS. On average, the number of attacks received by Centos based honeypots and Ubuntu based honeypots were not significantly different. This finding applies for honeynet with attractive and non-attractive FQDN.

3. In both groups SSH services were the most targeted service by the attackers followed by HTTP services. The third most probed service during the observation period was FTP then followed by HTTPS. Probes against MySQL and SMTP were ranked fifth and sixth respectively. This applies for both honeynets. On average the daily probe magnitude between the similar services, for example SSH servervice in non-attractive (VH1) and SSH in attractive (VH2) FQDN, between the two groups of honeynets were not significantly different.

   The high number of SSH brute force attack attempts by the adversaries through weak username and password combination indicates that though news about cyber-attacks have raised than they use to be, but still there is ignorance in choosing a strong username and password combination by network system users. Therefore, enforcing a strong username and password combination rule is inevitable.

4. Dshield.org database was used to address this question. Based on the information retrieved, some of source IP addresses that have attacked the honeynet have been observed attacking other networks. Out of all the source IP addresses attacked both honeynets, 33 % of them have been observed attacking other networks. However, the majority or 67 % of the source IP addresses attacked the honeynets in this project was not found in the Dshield database. Although these source IP addresses are not found in Dshield database, it does not mean they can be eliminated from a potential threats to the whole Internet as the information in Dshield database is collected from volunteers meaning that not all malicious IP addresses are reported. Therefore, enhancing the Dshield database or creating a general public awareness about malicious IP addresses can help in minimizing the threats that the Internet faces today.

Based on the information retrieved about attacker's country of origin most of the attacks source were originated from USA. In all the analysis that were conducted about the source IP address information related to the country of origin, USA based source IP addresses have dominated the numbers. However, the author is not blaming any country or the owners of domain names (IP addresses) that are mentioned in this document as been actively engaged in malicious activities because the source IP addresses might be victims themselves which are operating under command and control (C & C) or botnets.

Bibliography

[1] Gilmand Alimerkaj. The honeynet project: development of ait's honeynet. 2010.

[2] Thorsten Holz. Learning more about attack patterns with honeypots. In *Sicherheit*, pages 30–41, 2006.

[3] James Temple. Cyberattacks popular way to conduct social protest., 2011.

[4] Lance Spitzner. *Honeypots: tracking hackers*, volume 1. Addison-Wesley Reading, 2003.

[5] Jaikumar Vijayan. Ny times fingers melbourne it hack for site outage, 2013.

[6] Matthew Hilburn. China hit by 'largest ever' hack attack, 2013.

[7] ARKIN BRAD. Important customer security announcement, 2013.

[8] BBC News. Anonymous indonesia' attacks australian websites, 2013.

[9] Wolfgang Boehmer. Appraisal of the effectiveness and efficiency of an information security management system based on iso 27001. In *Emerging Security Information, Systems and Technologies, 2008. SECURWARE'08. Second International Conference on*, pages 224–231. IEEE, 2008.

[10] Christian Czosseck and Kenneth Geers. *The Virtual Battlefield: Perspectives on Cyber Warfare*, volume 3. Ios Press, 2009.

[11] Gurdip Kaur and Jatinder Singh Saini. Implementation of high interaction honeypot to analyze the network traffic and prevention of attacks on protocol/port basis. *International Journal of Computer Applications*, 62(16):22–29, 2013.

[12] Bruce Schneier. *Secrets and lies: digital security in a networked world*. Wiley. com, 2011.

[13] Carl Endorf, Jim Mellander, and Eugene Schultz. *Intrusion detection and prevention: the authoritative guide to detecting malicious activity*. McGraw-Hill/Osborne, 2003.

[14] OSSEC OSSEC. Ossec–an open source host-based intrusion detection system.

[15] Xinyou Zhang, Chengzhong Li, and Wenbin Zheng. Intrusion prevention system design. In *Computer and Information Technology, 2004. CIT'04. The Fourth International Conference on*, pages 386–390. IEEE, 2004.

[16] Niels Provos and Thorsten Holz. *Virtual honeypots: from botnet tracking to intrusion detection*. Pearson Education, 2007.

[17] Lance Spitzner. Honeypots: Definitions and value of honeypots. *Available from: www. tracking-hackers. com/papers/honeypots. html*, 2003.

[18] Lance Spitzner. Know your enemy: Honeynets. *Honeynet Project*, 2005.

[19] Slideshare. Honeynet handbook, 2010.

[20] Nathalie Weiler. Honeypots for distributed denial-of-service attacks. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2002. WET ICE 2002. Proceedings. Eleventh IEEE International Workshops on*, pages 109–114. IEEE, 2002.

[21] Mohammadzadeh Hamid, Mansoori Masood, and Honarbakhsh Roza. Taxonomy of hybrid honeypots. In *International Conference on Network and Electronics Engineering*, volume 11. IACSIT Press,Singapor, 2011.

[22] Gordon Fyodor Lyon. *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Insecure, 2009.

[23] ATTRIBUTED BY ROGER NEEDHAM and Butler Lampson. Network attack and defense. *white paper*, 2008.

[24] The World FACTBOOK. Internet hosts, 2012.

[25] ISC. Isc domain survey, 2013.

[26] John E Canavan. *The Fundamentals of Network Security*. Artech House, 2001.

[27] Institute Ponemon. 2013 cost of data breach study: Global analysis, 2013.

[28] Dave Lee. New york times and twitter struggle after syrian hack, 2013.

[29] Wikipedia. Information security components. retrieved, 2013.

[30] Theodore Socolofsky and Claudia Kale. Rfc 1180-tcp. *IP tutorial*, 1991.

[31] Paul V Mockapetris. Domain names-concepts and facilities. 1987.

[32] Robert Shirey. Rfc 2828: Internet security glossary. *The Internet Society*, 2000.

[33] Kris Katterjohn. Port scanning techniques, 2007.

[34] Patrick Engebretson. *The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy*. Access Online via Elsevier, 2011.

[35] Michael Rash. *Linux Firewalls: Attack Detection and Response with iptables, psad, and fwsnort*. No Starch Press, 2007.

[36] Haining Wang, Danlu Zhang, and Kang G Shin. Detecting syn flooding attacks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1530–1539. IEEE, 2002.

[37] Simon Hansman and Ray Hunt. A taxonomy of network and computer attacks. *Computers & Security*, 24(1):31–43, 2005.

[38] Cotton Michael. Targeted network attacks. *white paper*, 2011.

[39] Art Conklin, Gregory White, Chuck Cothren, Dwayne Williams, and Roger L Davis. *Principles of computer security: security+ and beyond*. McGraw-Hill, Inc., 2004.

[40] David Moore, Colleen Shannon, Douglas J Brown, Geoffrey M Voelker, and Stefan Savage. Inferring internet denial-of-service activity. *ACM Transactions on Computer Systems (TOCS)*, 24(2):115–139, 2006.

[41] Nikita Patel, Fahim Mohammed, and Santosh Soni. Sql injection attacks: Techniques and protection mechanisms. *International Journal on Computer Science and Engineering*, 3(1):199–203, 2011.

[42] Collie Byron. Intrusion investigation and post-intrusion computer forensic analysis.

[43] Iyatiti Mokube and Michele Adams. Honeypots: concepts, approaches, and challenges. In *Proceedings of the 45th annual southeast regional conference*, pages 321–326. ACM, 2007.

[44] Mohammadzadeh Hamid, Mansoori Masood, and Honarbakhsh Roza. A survey on dynamic honeypots. In *International Journal of Information and Electronics Engineering*, volume 2, 2012.

[45] Lance Spitzner and Marty Roesch. The value of honeypots, part two: Honeypot solutions and legal issues. *Security Focus. Retrieved November*, 6:2005, 2001.

[46] omnisecu. Leading honeypot products, 2010.

[47] Lukas Rist, Sven Vetsch, Marcel Kossin, and Michael Mauer. Know your tools: Glastopf-a dynamic, low-interaction web application honeypot. *The Honeynet Project*, 2010.

[48] Jose Antonio Coret. Kojoney-a honeypot for the ssh service, 2006.

[49] Niels Provos. Honeyd-a virtual honeypot daemon. In *10th DFN-CERT Workshop, Hamburg, Germany*, volume 2, 2003.

[50] Georg Wicherski. Medium interaction honeypots. *German Honeynet Project*, 2006.

[51] Craig Valli. Ssh–somewhat secure host. In *Cyberspace Safety and Security*, pages 227–235. Springer, 2012.

[52] Honeywell project. *Know Your Enemy: Learning about Security Threats*. Addison Wesley, 2004.

[53] Mark Meijerink and Jonel Spellen. Intrusion detection system honeypots. *Master Program System and Network Administration, University of Amsterdam, Amsterdam*, 2006.

[54] Fahim H Abbasi and RJ Harris. Experiences with a generation iii virtual honeynet. In *Telecommunication Networks and Applications Conference (ATNAC), 2009 Australasian*, pages 1–6. IEEE, 2009.

[55] John Levine, Richard LaBella, Henry Owen, Didier Contis, and Brian Culver. The use of honeynets to detect exploited systems across large enterprise networks. In *Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society*, pages 92–99. IEEE, 2003.

[56] Gordon W Romney, C Higby, BR Stevenson, and N Blackham. A teaching prototype for educating it security engineers in emerging environments. In *Information Technology Based Higher Education and Training, 2004. ITHET 2004. Proceedings of the FIfth International Conference on*, pages 662–667. IEEE, 2004.

[57] Christian Seifert, Ian Welch, Peter Komisarczuk, et al. Honeyc-the low-interaction client honeypot. *Proceedings of the 2007 NZCSRCS, Waikato University, Hamilton, New Zealand*, 2007.

[58] Radek Hes, Peter Komisarczuk, Ramon Steenson, and Christian Seifert. *The Capture-HPC client architecture*. School of Engineering and Computer Science, Victoria University of Wellington, 2009.

[59] Kaushik Gaurav and Tyagi Rashmi. Honeypot : Decoy server or system setup together information regarding an attacker. *International Journal of Computer Science and Information Technology*, 2012.

[60] Daniel Huluka. Experiment using distributed high-interaction honeynet (d2h). 2013.

[61] Wright Craig. A comparative study of attacks against corporate iis and apache web servers, 2007.

[62] Pei-Te Chen, Chi-Sung Laih, Fabien Pouget, and Marc Dacier. Comparative survey of local honeypot sensors to assist network forensics. In *Systematic Approaches to Digital Forensic Engineering, 2005. First International Workshop on*, pages 120–132. IEEE, 2005.

[63] Cynthia Bailey Lee, Chris Roedel, and Elena Silenok. Detection and characterization of port scan attacks. *Univeristy of California, Department of Computer Science and Engineering*, 2003.

[64] Marco De Vivo, Eddy Carrasco, Germinal Isern, and Gabriela O de Vivo. A review of port scanning techniques. *ACM SIGCOMM Computer Communication Review*, 29(2):41–48, 1999.

[65] Jaeyeon Jung, Vern Paxson, Arthur W Berger, and Hari Balakrishnan. Fast portscan detection using sequential hypothesis testing. In *Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on*, pages 211–225. IEEE, 2004.

[66] William W Martin. Honey pots and honey nets-security through deception. *SANS Institute Paper*, 2001.

[67] Fernando J Corbató, Marjorie Merwin-Daggett, and Robert C Daley. An experimental time-sharing system. In *Proceedings of the May 1-3, 1962, spring joint computer conference*, pages 335–344. ACM, 1962.

[68] KVM. Kernel based virtual machine., 2013.

[69] Lanz Spitzner. Know your enemy: Honeywall cdrom roo 3rd generation technology, 2005.

[70] ISC Home page. Isc history and overview., 2001.

[71] Martha Aliaga and Brenda Gunderson. *Interactive statistics*. Prentice Hall, 1999.

[72] Donald R Cooper, Pamela S Schindler, and Jianmin Sun. Business research methods. 2006.

[73] Rand R Wilcox. *Introduction to robust estimation and hypothesis testing*. Academic Press San Diego, 1997.

[74] Whitney K Newey and Daniel McFadden. Large sample estimation and hypothesis testing. *Handbook of econometrics*, 4:2111–2245, 1994.

[75] Conrad Carlberg. *Statistical Analysis: Microsoft Excel 2010*. Pearson Education, 2011.

[76] Ulf Lamping and Ed Warnicke. Wireshark user's guide. *Interface*, 4:6, 2004.

[77] Shyh-Sen Huang and Nathan McReynolds. An investigation of honeypots and honeynets. *Network Security*, 2004.

[78] Wondimagegn Endale Kibret. Analyzing network security from a defense in depth perspective. 2011.

.1 AppendixA

This perl script is designed to analyse SSH log files.

Listing 1: SSH Log Analyzer

```

#!/usr/bin/perl

# This perl script is designed to analyse SSH log
# files of failed attempts. It calculates statistic
# of event within the log file and converts time to local.
# This script was originally developed by  Evangelos Tasoulas
# and Modified by the author of the document to fit for the project
# needs.

# Needed packages
use Getopt::Std;
use strict "vars";
use Time::Local;
use Statistics::Descriptive;



# Global variables
my $VERBOSE = 0;
my $DEBUG = 0;
my $OVER2Lines = 0;

#####################
# handle flags and arguments
# Example: c == "-c", c: == "-c argument"
my $opt_string = 'hvdf:a';
getopts( "$opt_string", \my %opt ) or usage() and exit 1;

# print help message if -h is invoked
if ( $opt{'h'} ){
    usage();
    exit 0;
}

$VERBOSE = 1 if $opt{'v'};
$DEBUG = 1 if $opt{'d'};
$OVER2Lines = 1 if $opt{'a'};

# main program content
my $FILENAME = $opt{'f'};
# If no file has been supplied, use the
# one from the default path.
if ($FILENAME eq "") {
    $FILENAME = "falmawit.log";
}
my %SCANS;
my %connections_per_sec;


verbose("Filename is $FILENAME\n");

die "Error: No such file:$FILENAME\n" \
     unless -f $FILENAME;
#Open file to analyse
open(FILE,"$FILENAME") or die \
      "Unable to open a file:$!\n";

#Loop through file
while (my $line = <FILE>) {
    if ($line =~ /^(\S{3} \d\d \d\d:\d\d:\d\d) (\S{1,})\
    .*authentication failure.*rhost=(\S[^ ]*) / ) {

        my $remotehost = $3;
        my $target = $2;

        (getUnix($1) - getUnix($SCANS{$remotehost}
                            {"stop date"})) . "\n"
       if getUnix($SCANS{$remotehost} \
          {"stop date"});
        push(@connections_per_sec, \
          ((getUnix($1) - getUnix($SCANS{$remotehost} \
         {"stop date"}))) if getUnix($SCANS{$remotehost}\
         {"stop date"});
        debug("Printing Conns Per Sec \
        (" . @connections_per_sec . ") \n");
        push @{ $connections_per_sec{$remotehost} },\
          ((getUnix($1) - getUnix($SCANS{$remotehost}\
         {"stop date"})))
        if getUnix($SCANS{$remotehost}{"stop date"});
        $SCANS{$remotehost}{"connections_per_sec"} \
        {((getUnix($1) - getUnix($SCANS{$remotehost} \
        {"stop date"})))}++;
        $SCANS{$remotehost}{"count"}++;
        $SCANS{$remotehost}{"target"}{$target} = 1;
        $SCANS{$remotehost}{"start date"} = $1 \
        unless $SCANS{$remotehost}{"start date"};
        $SCANS{$remotehost}{"stop date"} = $1;
    }
```

```
}


if($OVER2Lines) {
    print "Only remote hosts with more than 2 \
    connection attempts will be printed out!\n\n";
    print_results(2);
}
else {
    print_results(0);
}


#####################
# Helper routines

sub usage {
     # prints the correct use of this script
     print "Usage:\n";
     print "-h    Usage\n";
     print "-v    Verbose\n";
     print "-d    Debug\n";
     print "-f    Choose log file to read\n";
     print "-a    Print only hosts with more than 2 \
      connection attempts\n";
}

sub verbose {
     print $_[0] if ( $VERBOSE );
}

sub debug {
     print $_[0] if ( $DEBUG );
}

my %MONTHS = (
    "Jan" => 0,
    "Feb" => 1,
    "Mar" => 2,
    "Apr" => 3,
    "May" => 4,
    "Jun" => 5,
    "Jul" => 6,
    "Aug" => 7,
    "Sep" => 8,
    "Oct" => 9,
    "Nov" => 10,
    "Dec" => 11
);

sub getUnix {
    my $date = $_[0];

    if ( $date =~ /^(\w{3}) (\d\d) (\d\d):(\d\d):(\d\d)/) {
        my $month = $1;
        my $day = $2;
        my $hour = $3;
        my $minute = $4;
        my $second = $5;
        my $year;

        # Avoid hard coding of the year. Determine it automatically.
        (my $CurrentSecond,
        my $CurrentMinute,
        my $CurrentHour,
        my $CurrentDay,
        my $CurrentMonth,
        my $CurrentYear,
        my $CurrentWeekDay,
        my $CurrentDayOfYear,
        my $CurrentIsDST) = localtime(time);

        $CurrentYear += 1900;
        $CurrentMonth++;
        #my $CurrentDate = "$CurrentDay/$CurrentMonth/$CurrentYear";
        #print $CurrentDate;
        if($MONTHS{$month} = 11 && $CurrentMonth = 0) {
            $year = $CurrentYear - 1;
        }
        else {
            $year = $CurrentYear;
        }
        return timelocal($second, $minute, $hour, \
         $day, $MONTHS{$month}, $year);
    }
}

sub print_results {

  my $attempts = 0;
  my $start_date = 0;
  my $stop_date = 0;
  my  $scan_length = 0 ;
  my $tBTNattempts = 0;
  my $DstIP ;
   my $number_of_minumum_attempts = $_[0];
   my $total_attempts = 0;

    print "SourceIP \n";

print "SourceIP \t DestIP \t NumAttempts \t \
     StartDate \t EndDate \t ScanLength/sec \t \
     TimeBTN_Attempts \n";
   foreach my $remotehost (keys %SCANS) \
   {
      if($SCANS{$remotehost}{"count"} > \
      $number_of_minumum_attempts) {

         my @data_difference_between_connections_in_seconds = \
         @{ $connections_per_sec{$remotehost} };
         # Some extreme values between the scans can
         # change the mean value dramatically,
         # so use the statistics library to improve our results.
         my $numberOfDifferenceBetweenConnections =
         scalar \
         (@data_difference_between_connections_in_seconds);

         # if we had only one connection then there is
         # no difference between  the attempts as there
         # is only one attempt.
         # That means that $numberOfDifferenceBetweenConnections
         # equals to 0;
         # if we had only two connections, then then just
         # print the time difference between them.
         # $numberOfDifferenceBetweenConnections equals to
         # 1 in this case.
         # In any other case check the median and the mean
         # value of the total values.
         # If the difference is less than 2, print the mean
         # value otherwise print the median value.
         if ($numberOfDifferenceBetweenConnections == 0) {
           ## print "Time between connection attempts: 0\n";
         }
         elsif($numberOfDifferenceBetweenConnections == 1) {
            ## print "Time between connection attempts: \
            " . @data_difference_between_connections_in_seconds[0] \
             . "\n";
         }
         else {
            my $stat = Statistics::Descriptive::Full->new();

            $stat->add_data \
             (@data_difference_between_connections_in_seconds);

            my $median = $stat->median;
            my $mean = $stat->mean;
            debug("Median: " . $median . "\n");
            debug("Mean: " . $mean . "\n\n");
            if(abs($mean-$median) > 2) {
 print "Time between connection attempts: \
         " . $median . "\n";
            }
            else {
 print "Time between connection attempts: " \
        . $mean . "\n";
                 $tBTNattempts =  $mean ;
            }
         }

print "Remote scan from: `$remotehost`\n";
print " $remotehost \n";
print "Remote scan from: " . $remotehost . "\n";
          foreach my $target ( keys %{ $SCANS{$remotehost}
print "Targeted: `" . $target . "`\n";
         $DstIP = $target ;

          }

print " " . $SCANS{$remotehost}{"count"} . "\n";
print " " . $SCANS{$remotehost}{"start date"} . "\n";
          # my $attempts, $start_date, $stop_date, $scan_length;

          # print "Number of attempts:" . $SCANS{$remotehost} \
                  {"count"} . "\n";
          $attempts =   $SCANS{$remotehost}{"count"} ;


          # print "Start date: " . $SCANS{$remotehost}\
                  {"start date"} . "\n";
          $start_date =  $SCANS{$remotehost}{"start date"} ;

          my $unixstart = getUnix($SCANS{$remotehost} \
          {"start date"});
          # print "Stop date: " . $SCANS{$remotehost} \
            {"stop date"} . "\n";
          $stop_date =  $SCANS{$remotehost}{"stop date"} ;

          # print  . $SCANS{$remotehost}{"stop date"} . "\n";
```

```
            my $unixstop = getUnix($SCANS{$remotehost} \
                  {"stop date"});
            # print "Scan length in seconds: " . \
                    ($unixstop - $unixstart) . "\n";
            $scan_length =  ($unixstop - $unixstart) ;

        }
        $total_attempts += $SCANS{$remotehost}{"count"};

###########################################################

 print " $remotehost \t  $DstIP \t $attempts \t  \
            $start_date \t $stop_date \t $scan_length \t \
             $tBTNattempts \n";
             #print " $remotehost \n";
    }
    print "Total number of connection attempts: " . \
            $total_attempts . "\n";

}
```

.2 AppendixB

This program prints IP addresses that are found in a give file with their number of occurrence.

Listing 2: Retrieve IP and number of occurrence of each.

```

#!/usr/bin/perl
# This program will print IP addresses that are found in the
# given file along side the number of occurrence of each IP

use warnings;
use strict;

# Declaring variables and assign relevant values
# the first variable should be the file path
# the second variable holds hash value
my $log = "$filename";
my %seen = ();

# Open the file and read it line by line
# for each observed IP address increment number of
# observation by one
open (my $fh, "<", $log) or die "unable to open $log: $!";

while( my $line = <$fh> ) {
    chomp $line;

     if( $line =~ /(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})/ ){
         $seen{$1}++;
     }
}
close $fh;

# Print results
 print "Remote IP \t \t\t\t  Observed \n";
for my $key ( keys %seen ) {
#    print "$key \t \t $seen{$key}\n";
print "$key  \n";

}
```

.3 AppendixC

This perl script finds the country of origin of IP addresses that are stored in a file by retrieving information from www.geoiptool.com database.

Listing 3: IP address's Country Locator

```

#!/usr/bin/perl
# This perl script finds the country of origin of IP addresses
# stored in a file by looking information about an IP addresses on
# www.geoiptool.com .
# Run the program: ./geoIP.pl -i  $filename

# Needed packages
use Getopt::Std; # packages user opt arguments ...
use strict "vars"; # strict on variables

# Global variables

my $VERBOSE = 0; # our is also another option
my $DEBUG = 0;

################################

# handle flags and arguments
# Example: c == "-c", c: == "-c argument"

my $opt_string = 'hvdi:cCHn'; # h-help, v-verbose, d-debug
getopts( "$opt_string", \my %opt ) or usage() and exit 1;

# print help message if -h is invoked
if ( $opt{'h'} ) {
    usage();
    exit 0;
    }


my $VERBOSE = 1 if $opt{'v'};
my $DEBUG = 1 if $opt{'d'};


# main program content


my $file = $opt{'i'};

my @ip;
```

```

open(FILE,”\$filename”);
while(my $line=<FILE>){
        push(@ip,$line);
}
close(FILE);

for(my $i=0; $i<=$#ip; $i++){
        print ”$ip[$i]\n”;
        get_location($ip[$i]);
}


######################################
# Helper routines

sub usage {
    # prints the correct use of this script
    print ”Usage:\n”;
    print ”−h Usage\n”;
    print ”−v Verbose\n”;
    print ”−d Debug\n”;
    }

 sub verbose {
  print $_[0] if ($VERBOSE or $DEBUG);
 }

 sub debug {
  print $_[0] if ($DEBUG);
  }

sub get_location{

#print ”IP = $_[0]\n”;

  # Retrieve IP information from website database
open(GEO, ”wget −q −O − \
      http://www.geoiptool.com/en/?IP=$_[0]  |”);
        while(my $line = <GEO>){


# Check matching criteria and prints country and city.

  if ($line =˜ /City<\/strong>:\s(.*)<br>. \
       *Country<\/strong>:\s(.*)<br><str/i){
         print ”Country: $2\n”;
        print "City: $1\n";

        }
   elsif ($line =~ /Host\sName:<\/span><\/td>/){
         $line = <GEO>;
        if ($line =~ /width="198".+class="arial_bold">
        \(.+)<\/td>/){
         print "Hostname: $1\n";

                                last;
                        }
                }
        }
        close(GEO);
}
```

.4 AppendixE

This perl script checks if attackers IP address has been reported previously to Dshield database.

Listing 4: Check Attacker Status against Dshield

```

#!/usr/bin/perl
# This perl script retrieves IP addresses status if it has
# been previously reported to Dshield as  been a malcious IP.
# The script expects a filename with the IP
# address to be provided. Look below how to run it.
# Run the program: ./dshield.pl -i  $filename

# Needed packages
use Getopt::Std; # packages user opt arguments ...
use strict "vars"; # strict on variables
use HTML::TableExtract;

# Global variables

my $VERBOSE = 0; # our is also another option
my $DEBUG = 0;

###############################

# handle flags and arguments
# Example: c == "-c", c: == "-c argument"

my $opt_string = 'hvdi:cCHn'; # h-help, v-verbose, d-debug
getopts( "$opt_string", \my %opt ) or usage()\
          and exit 1; # exit 1 is for error,

# print help message if -h is invoked
if ( $opt{'h'} ) {
    usage();
 # proper exit. exit without number takes 0 as a default.
    exit 0;
    }


my $VERBOSE = 1 if $opt{'v'};
my $DEBUG = 1 if $opt{'d'};

# main program content
```

```
my $file = $opt{'i'};

my @ip;

open(FILE,"$file");
while(my $line=<FILE>){
        push(@ip,$line);
}
close(FILE);

for(my $i=0; $i<=$#ip; $i++){

        get_location($ip[$i]);
}


#####################################
# Helper routines

sub usage {
    # prints the correct use of this script
    print "Usage:\n";
    print "-h Usage\n";
    print "-v Verbose\n";
    print "-d Debug\n";
    }

 sub verbose {
  print $_[0] if ($VERBOSE or $DEBUG);
 }

 sub debug {
  print $_[0] if ($DEBUG);
  }

sub get_location {
my $te = HTML::TableExtract->new();
        my ($ts,$row)='';
 # Retrieve IP information from Dshield database

   open(GEO, "wget -q -O - \
    http://www.dshield.org/ipinfo.html?ip=$_[0]  |");
     while(my $line = <GEO>){

if ( $line =~ /IP Address \(click for more detail\)\:/ )

    {
```

```

 $line = "<table>"."$line"."</table>";

# Eextract the individual fields out of the entire line.

$te->parse($line);

# Examine all matching tables
 foreach $ts ($te->tables) {

 foreach $row ($ts->rows) {
     my @values = grep {defined} @$row;
     my $record = join(' ',@values);
                      $record =~ s/^\s+//g;
                      print "$record\n";
                      }
             last;
                      }

                      print "\n";

             }

              }
             close(GEO);
}
```

.5 AppendixF

This perl script analysed apache log file.

Listing 5: Apache Log Analyser

```

#!/usr/bin/perl -w
# This perl script analysed apache
# log file. It prints the total number of
# attempts for each method and as well prints
# the attempt count with type of vulnerability
# tried

use strict;
use warnings;

open (LOG, "$filename");

my $options = {};
my $methods = {};
my $urls    = {};
my $host;
my $date;
my $url_with_method;
my $status;
my $size;
my $referrer;
my $agent;

while (my $line=<LOG>) {
 ($host,$date,$url_with_method,$status,$size,$referrer,$agent) \
    = $line =~ m/^(\S+) - - \[(\S+ [\-|\+]\d{4})\] \
       "(\S+ \S+ [^"]+)" (\d{3})
       (\d+|-) "(.*?)" "([^"]+)"$/;

# Uncommenting next line will make filtering for
# the specified month only in the log

 my ($method, $url, $http) = split /\s+/, $url_with_method;

  $url =~ s/\?(.*)//;
  $referrer=~ s/\?(.*)//;

  push @{$methods->{$method}}, $url;
  $urls->{$url} -> {host     } -> {$host}      ++;
  $urls->{$url} -> {count    }                 ++;
  $urls->{$url} -> {referrer} -> {$referrer} ++;
```

```
}


# Prints the type of methods with the total
# number for each method

print "Connection Methods \t Number Of Counts \n";
foreach my $m (keys %{$methods}) {
  print "$m :  \t\t" . @{$methods->{$m}} ." \n";

}
print "Number of Attempts  Attempts Mechanism \n";
  foreach my $url (sort {$urls->{$b}->{count} <=> $urls->
  {$a}->{count} } keys %{$urls})
      {
  printf ("%5d %s\n\n", $urls->{$url}->{count},$url, "\n");

my @linesOut;

if ($options->{f}) {
  my $currentLine=0;
 foreach my $host (sort {$urls->{$url}->{host}->{$b} <=>
 $urls->{$url}->{host}->{$a} } keys %{$urls-> \
    {$url}->{host}}) {

   last if $currentLine > $options->{t};
   $linesOut[$currentLine] .= sprintf "  %5d %-35.35s" ,
   $urls->{$url}->{host}->{$host}, $host;
   $currentLine++;

    }

  }

  if ($options->{r}) {
    my $currentLine=0;
    foreach my $referrer (sort {$urls-> \
    {$url}->{referrer}->{$b} <=>
    $urls->{$url}->{referrer}->{$a} } \
    keys %{$urls->{$url}->{referrer}}) {
      last if $currentLine > $options->{t};
      $linesOut[$currentLine] .= \
      sprintf "  %5d %-55.55s" ,$urls->
      {$url}->{referrer}->{$referrer}, $referrer;
      $currentLine++;
    }
  }
```

```

}
```