DETERRING HACKING STRATEGIES VIA TARGETING SCANNING PROPERTIES

Saad Alsunbul[1,2], Phu Dung Le[1] and Jefferson Tan[1]

[1]Caulfield School of Information Technology, Monash University, Melbourne,Australia
[2]Computer Research Institute, King Abdullaziz for Science and Technology, Riyadh,Saudi Arabia

ABSTRACT

Network infrastructures have played important part in most daily communications for business industries, social networking, government sectors and etc. Despites the advantages that came from such functionalities, security threats have become a daily struggle. One major security threat is hacking. Consequently, security experts and researchers have suggested possible security solutions such as Firewalls, Intrusion Detection Systems (IDS), Intrusion Detection and Prevention Systems (IDP) and Honeynet. Yet, none of these solutions have proven their ability to completely address hacking. The reason behind that, there is a few researches that examine the behavior of hackers. This paper formally and practically examines in details the behavior of hackers and their targeted environments. Moreover, this paper formally examines the properties of one essential pre-hacking step called scanning and highlights its importance in developing hacking strategies. Also, it illustrates the properties of hacking that is common in most hacking strategies to assist security experts and researchers towards minimizing the risk of hack.

KEYWORDS

Hacking, network security, security properties, pre-hacking, scanning, necessary information

1. INTRODUCTION

Currently, network infrastructures have played important part in most daily communications for business industries, social networking, government sectors and etc. It has become a necessity for most occurring communications in which it incredibly eases exchanging information, storing and retrieving data. However, such a considerable advantage comes with many security threats. One major security threat to computer networks is hacking.

Hacking is a descriptive term used to describe the attitude and behaviour of group of people who are greatly involved in technical activities which, more commonly today than in previous years, result in gaining unauthorized access on their victims' infrastructures. Generally, hackers aim to study all technologies aspects in most infrastructures and explore their vulnerabilities.

Vulnerabilities within infrastructures are mostly derived from vulnerabilities within operating systems, network technologies, communication protocols, security postures, software errors or even from the integration between these technologies.

DOI : 10.5121/ijnsa.2015.7401

Currently, successful hacking attempts still present. For instance, in 2014, Sony Pictures Entertainment was struck by hacking attack after releasing a movie called "The interview". The attack was intended to literally move data from Sony network and leave no possibility to recover such lost data. [1]. Also, in 2012, the Saudi Arabia Oil Company (Saudi ARAMCO) was hit by an external virus named Shamoon, which spread and affected around 30,000 workstations. That malicious software aims to wipe computers hard drive indiscriminately. Fortunately, The company stated that the virus did not reach to the production line and they have not clearly identify the losses [2].

Such incidents have drawn security industry, experts and researchers towards addressing hacking. There are many security solutions suggested and practically deployed in most network infrastructures to address hacking in higher scale such as Firewalls, Intrusion Detection Systems (IDS), Intrusion Detection and Prevention System (IDP), Honeypot and Honeynet. However, there are still drawbacks associated with these security solutions that have proven their inability to address such attacks.

There is a noticeable difference between security experts and hackers in the scene of exploring vulnerabilities in computer systems [3]. For an effective secure defence system, a system designer must fully understand hackers' behaviours and propose a defence system that addresses the common ground between most of hacking techniques. Nevertheless, there is unnoticeable effort made by security experts and researchers with the aim to study the behavior of hackers.

The difficulty in pursuing this research field is caused by countless factors such as continuous emerging of new technologies, introducing new vulnerabilities and etc. However, there is indispensability to examine the behaviour of hackers in a great depth and find the common starting ground for most hacking strategies. The purpose of this article is to illustrate our finding of hacking root and common properties that must be satisfied by most hacking strategies that is to be assisting future security researches towards addressing hacking.

This paper is organized as follows: Section 2 discuses the related work made towards addressing hacking in large scale and their limitations. Moreover, section 2 highlights the importance of studying hacking behaviour in order to suggest future security solutions based on our finding in this article. Section 3 discuses the communication environment in general. Section 4 illustrates hacking strategies and the necessary information for most hacking strategies. The three essential pre-hacking steps are examined in section 5. Section 6 discusses the importance of scanning for obtaining necessary information for hacking strategies. Then, the importance of scanning in developing hacking strategies is illustrated in section 7. Examining the possibility of deterring hacking strategies via targeting scanning properties is illustrated in section 7. Section 8 concludes this paper.

2. RELATED WORK

Most of the current security solutions tend to embrace one option from the general classification of the current solutions; it is either a passive defence or active defence approach. Passive defence systems such as Firewalls, Intrusion Detection Systems (IDS) are security approaches that preclude or minimize all defined or common cyber attacks in the first place when there are hacking attempts. Despite the fact that most of these systems are essential for many network

infrastructures, there are still limitations present in each of these systems, which will be explained in detail in the following subsections. However, Active defence systems such as Intrusion Detection and Prevention (IDP), and Honeypots, are considered more advanced security approaches that detect common and some new intrusions and actively respond to these attacks.

2.1. Firewalls

The conceptual model of a firewall provides the ability to manage every sub-network separately and gives every department the capability to manage their own sub-network according to their policies and requirements [4]. This management feature escalates the importance of firewalls as a building block of any network design [5]. To make a decision about a packet; the packet must be examined under a sequence of rules. Then, the firewall generates the decision, which is applied to the packet [6] [7]. Firewalls as security technology fall into four types based on the filtering algorithm and the operation layer (IP, Transport or Application layers) and they are: Packet filtering, Circuit gateway, Application gateway and hybrid firewalls [8].

Despite the fact that firewall is one of the building blocks in any network design, there are limitations on using them as the one and only line of defence. Depending on the technical capability of firewall designers, errors might be introduced if a firewall designer is not highly trained and experienced [5]. The succession of malicious viruses and worms such as Blaster [9] and Sapphire [10] implies that most of firewall breaches are caused by configuration errors.

Even the modern design of firewalls, which demands the distribution of firewalls within the organization's network, it fails because of the complex protection requirements. What makes the situation even worse is that every host has limited users and all of them are treated as trustworthy users, the possibility of getting inside attack such as IP network spoofing, packet sniffing and denial-of-service is still possible [4]. Since those attacking strategies are mostly connected with human behaviour, the importance of having dynamic security posture have become more practically than the static security implementation.

2.2. Intrusion Detection Systems (IDS)

The conceptual model of IDS was introduced to be a real-time defence system to detect intruders inside networks, unauthorized use, abuse and misuse of computer systems [11][12]. Basically, it has been designed and proposed under the assumption that a normal user's behaviour is completely different than an intruder [12]. So, the gap between their behaviours is the key for spotting an intrusion. IDS has the ability to analyse, detect intrusion, recognize the source of attack and alleviate the effect of most of unexplored attacks [11].

IDS obtains data from different sources, which constructs a network infrastructure (networks and hosts), and that differentiation creates a classification for IDS. The classification consists of: Network-based IDS, Host-based IDS and Hybrid IDS [12].

Nevertheless, Detection technique requires a model of intrusion, which are: Anomaly or misuse detection techniques. Anomaly detection technique applies the concept of collecting user's profiles and defines them as normal behaviours or normal patterns [13]. Then, in real time, IDS analyses current users' sessions and maps them with defined normal behaviours to recognize

abnormal activities. The normal behaviour varies depending on the workload and number of operations and activities operated by users [13][14]. However, the second type is referred as misuse detection. Basically, it defines the basic techniques used by attackers and models them into the system under the term "signatures" [12][13]. In that case, the system processes all streamed audit files searching for these signatures [12][13][14].

Despite the fact that IDS systems are one important security component for most systems and commonly deployed, there are some drawbacks related to that technology. For example, Anomaly Intrusion Detection system has large number of false positive alerts. Furthermore, the dynamic feature which is supposed to detect new forms of attacks, is very difficult in reality. In other words, it detects only the modelled attacks and disregards any new invented attacks [15][16].

Even that the conceptual model of IDS in detecting modelled attacks seems to be realistic, it is far from being achieved due to attackers' abilities to change some information to deceive IDS such as port number, sequence number or protocol indicator [15]. That trick can pass IDS and provide the same result without being noticed. Furthermore, false positive alerts can cause overload in the victim's network. In reality, a hacker injects the target's network with common modelled attacks causing an IDS to generate detection alerts which regularly leads to overload the network. This attacking strategy is utilized to hide the real breaching path to the systems by shuffling it with the fake false positive alerts simply to avoid seizing the attention for the real attack [16].

2.3. Intrusion Detection and Prevention System (IDP)

One of the biggest drawbacks of IDS is the non-defendable mechanism in which it recognizes an attack without any single action in return. IDP shares the same characteristics with IDS but instead of generating alerts, IDP performs actions against the intrusion. As IDP inherits most of the design specifications and concepts; consequently, it inherits the drawbacks of the IDS as well. Moreover, there is an issue arises with IDP especially when it is standard with out-of-thebox configuration. Precisely, this configuration causes to recognize a large scale of normal activities as suspicious activities [17]. This issue is defined as false negative rate. Furthermore, it has been reported that %99 of the reported alerts by IDS/IDP are not related to the security aspects [17][18].

These limitations within the previous security systems have not reached to the satisfaction level for the security industry and the need to further researches has become vital. As a result of that, a Honeypot was introduced.

2.4. Honeypot

The founder of Honeypot, Lance Spitzner [19] defines it as "security resources whose value lies in being probed, attacked, or compromised". The fundamental concept of designing Honeypot is to study hackers' behaviours and assist the efforts made against hackers besides firewalls, IDS and IDPs. The basic concept is very simply, a collection of resources that have no reliance on the main operations of the original network and have no authorization mechanism and these resources are made to be breached by hackers; thus, their behaviours and techniques are recorded to gear up against them with proper security postures [20]. If that network does not get hacked, it has no value for security researchers [19][20].

Primary designed to study hackers' behaviours and eventually has become a tool for deception or an active supplement for Intrusion Detection Systems. One use of honeypot system is for deception; the deception is mainly targeted hackers, luring them to valueless network for their activities to be observed and recorded and these resources must appear realistic [21][22]. That valueless information might be a simulation for computers with known vulnerabilities or operating systems combined with network services with deliberate security holes [21][22]. Then, it is a Honeypot system responsibility to record a hacker's behaviour through background applications, capturing the transferred packets between the hacker and the Honeypot system; besides analysing and interpretation of the collected data [23]. The other use of honeypot system and the most common is by engaging it with other security postures. In this situation, Honeypot acts as a security supplement for IDS contributes to solve the major drawbacks in IDS which are the false positive and false negative rates. IDS combined with Honeypot may cut the number of false positive and false negative rates to %10 of the same production of IDS alone [20][23].

Usually setting up a honeypot forces the designer of the system to chose between two different methodologies, Low interaction and high interaction. Low interaction means that the trap (the deception) has limited resources due to the simulation process. Specifically, the simulated trap may have the ability to interact with few numbers of connections at one time compared with the actual capability of the real system [23]. This issue is one fine indicator for hackers that they have been deceived. Moreover, honeypot logs are stored locally; thus, escalates the risk of tampering logs file to serve hackers' deeds [23]. Likewise for high interaction, a high interaction honeypot has high cost and complicated configuration process in which if it has not configured probably, it eventually causes a low efficiency and utilization for the trap [23].

Additionally, most of honeypot is set to work in one segment making it very easy for hackers to detect them. Actually, the benefit gained from these traps is none. Hackers notice these traps and avoid presenting their tools and methodologies for breaching which leaves honeypot useless [23]. These traps are given for hackers to play with and show their evil tools and methods for hacking. In fact, hackers can detect these traps and continue to explore the vulnerabilities for one reason, launching an attack from honeypot. This reflection attack works for hackers as a barrier saves them from law enforcements and the third party may take the liability unless certain laws are presented [22].

Furthermore, these traps have a huge disadvantage, which is the main indicator for the trap itself. If it allows outbound connections to be made, they face the risk of being used as third party for another attack. In contrast, blocking outbound connections is very fine indicator for the trap, which makes it useless in many cases [20]. In addition, it is hard of honeypot to differentiate between legitimate users and hackers. Actually, it misses any attacks that does not enter the trap and goes directly to the victim's system.

2.5. Related work main limitations

Despite the fact that the security solutions discussed above provide some level of security, there are still limitation in addressing hacking. We Also found that these security solutions tend to neglect the behaviour of hackers and find similar hacking methodology. We found that hacking strategies come in different techniques and counting all the techniques is nearly impossible.We have examined the hacker behaviour based on [24] and found that hacker actually executes some

programs or tools that are classified under three categories. These categories are the prehacking steps (Footprinting, Scanning and Enumeration). The main purpose of this article is to formally and practically examine deeply the environment in which hacking takes place and the behaviour of hackers and find the properties that are shared between most of hacking strategies.

Studying the behavior of hackers provide us with complete understanding of the way they launch their attacks. "Think like a hacker" is the best way for security experts and researchers in future to suggested new security solutions and avoid vulnerabilities or at least discover them in the small time frame before hacker discovers it. Currently, the common security practice is based on suggesting a new security solution for a specific hacking strategy after vulnerability is being exploited; then, patched. See next figure for vulnerability lifecycle.

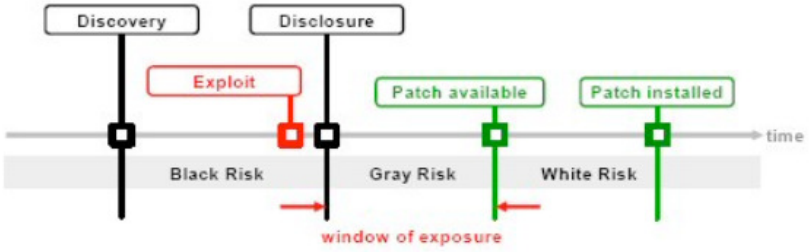

Figure 1: Vulnerability lifecycle for applications [25].

Moreover, hackers' attitudes, ethical beliefs and cultures towards hacking are important factors which widen a security experts' knowledge and create better chances for security experts and developers to be effective in the defensive line [26]. Furthermore, hackers have proven their abilities with impressive outcomes via their hacking techniques acquired mostly from their cultures. In fact, they intensively enrich their culture by exchanging the wealth of practice and methods, which clearly have demonstrated their effectiveness [27].

Furthermore, There is considerable gap between the way the hackers and computer scientists gain their skills and techniques creating better chances for hackers to continue and enrich their cultures. As a result of that, studying hackers' behaviors is essential to lead to future security solutions with high resistance and actively effective against most hacking attempts, which is the main aim for this article.

3. Communication In Network Infrastructures

Before we start examining the behaviour of hackers, the environment where hacking taking place is important to be inspected.

The design of network infrastructures comes in different topologies and network technologies that suit needs and requirements of organizations, which might forms complex design of network infrastructures. Most of network infrastructures contain and not limited to common hardware and software components that ensure the functionality. However, there are additional components for

functionality and security purposes. The common network infrastructure components are: Hosts, routers or switches, servers, cables, wireless routers, communication protocols, and network services. We define the common infrastructure components (INF) as:

$$INF = \{H, RT \parallel ST, S, C \parallel WL, P, SE\} \quad (1)$$

Setting up computer network with all services requires some H which stands for Host, where $H > 0$. Also, directing and managing the communicating with that network require RT or ST, where $RT > 0$ and $ST > 0$; Note that RT stands for routers and ST stands for switches. Nevertheless, S is a main element in term of functionality in any network which stands for servers; where $S > 0$. Also, C is the actually physical connection between all hardware within most computer networks, which stands for Cables. Moreover, P stands for intercommunication protocols, where $P > 0$. At last, SE stands for services provided to users, where $SE > 0$.

The form of communication that occurs within INF is based on packets. The formation of transmitted packets may differ based on deployed network protocols. Yet, the structure and formation of these packets must be predefined. As a consequence, communication protocols were defined. The main purpose of communication protocols is to facilitate the communication between participants in which it defines the structure and formation of packet, such as TCP/IP protocol for Internet.

Usually, packets are sent from source to destination hosts to satisfy an objective of the communication. The objectives of transmitted packets might vary from a normal request of webpage, to security breach. Nevertheless, counting all packets' objectives is close to impossible. From a security prospective, we categorize packets' objectives into two main categories, which are:

- A threat objective.
- None threat objective.

A threat objective appears when a request of the communication from a sender is intended to breach into computer system of a receiver. Thus, all generated and delivered packets for hacking purpose fall into the threat objective category. However, transmitted packets that generated for functionality purposes such as requesting webpage fall into the none threat objective category.

So, the communication between two participants is initiated to satisfy an objective, which is the objective of the transmitted packets. The communication between two participants appears when there are packets delivered from the sender to receiver and vice versa. In another words, if one participant does not send packets to the other participant, the communication in this case does not appear. Such predefined communication procedure provides tremendous functionalities. Yet, with great functionalities come communication threats.

3.1. Communication risk

Hacking threats are illustrated in the basic form of the communication, which are in the transmitted packets. Hacking happens in countless strategies based on deployed technologies in nominated victims' infrastructures. Nonetheless, most of hacking strategies heavily rely on necessary information in which hacking is nearly impossible to be performed without it (necessary information will be discussed in details in the following section).

The necessary information is obtained from the communication between hackers and their victims via two main techniques which are: engagement and analysis.

3.1.1. Engagement

The engagement is a sequence of communication between two participants where one participant sends packets to another participant, and receives responses; see the following figure.

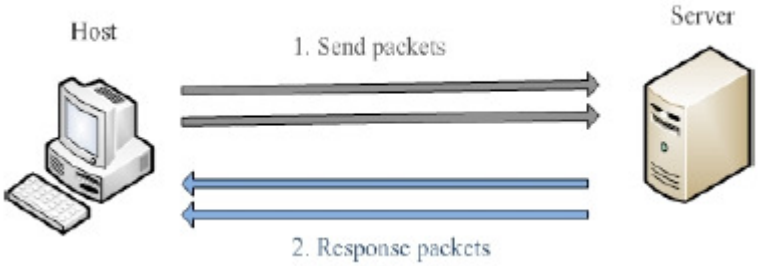

Figure 1: Engagement

It is limited to; a sender sends packets to the receiver, and the receiver responses with packets to the sender. Thus, when the sender is sending packets to the receiver and the receiver does not respond, it is not considered engagement. So, the engagement is defined as follows:

$$ENG = \langle Send(packet)^{1}, receive(packet)'^{1} \ldots\ldots Send(packet)^{n}, receive(packet)'^{n} \rangle \quad (2)$$

Where n and n', are the maximum number of sent and received packets required for the communication objective. packet'^{1}, is the packet generated by the receiver, which is a response to packet^{1}.

ENG is main requirement for current deployed technologies, which may become a highly risky functionality especially for hacking strategies. Eliminating ENG is impossible; however, controlling ENG between computers systems would eventually limit hacking risk that is discussed is section 7.

3.1.2. Analysis

Analysis is a process involves extracting information from transmitted packets. The analysis may include extracting information transmitted through ENG between the hacker and their victim or between legitimate users. The main purpose of analysis is to obtain the necessary information, which is discussed in details in section 8.2. Generally, analysis process may vary from a simple to complex processes depending on the required information for hacking strategy. Due to the variation of hacking strategies and continuous emerging of new hacking strategies, counting all analysis processes is impractical. However, this thesis focuses on analysis, as process that assists hackers to obtain the necessary information. Hence, we define analysis as:

$$ANL = EXTRACT\_NINF(packet^{1}, packet^{'1} .... packet^{n}, packet^{'n}) \quad (3)$$

Where *EXTRACT_NINF*, stands for extracting necessary information from transmitted packets between a hacker (which is the one who is performing the analysis process), and the victim.

The main requirement for analysis process is to know the structure and format of the transmitted packets that is defined in section 8.2. There are many protocols that wildly used such as TCP/IP protocol. Understanding how bits are organized to form packets is essential for any analysis process that eases extracting necessary information from transmitted packets.

4. Hacking Strategies

Hacking objectives vary because of countless factors such as stealing money, creating political issues, destroying repetition or just for fun. However, not all hacking attempts are successful in reaching their objectives and counting all hacking objectives is impractical.

This paper focuses on successful hacking attempts. Therefore, we define a successful hacking strategy (SHS) as a hacking strategy attempt where it satisfies its objective. SHS is number of executed operations by hackers where it designed to satisfy its objectives.

$$SHS = \{ SHSo_i : i \in I \} \quad (4)$$

SHSo, stands for number of executed successful hacking operations Where I = {1,2,3,4,5...n}. The following subsection illustrates a case study of common SHS that is remote password guessing attack on Windows systems. That case study highlights the necessary information of most remote password guessing attach as for most SHSs.

4.1. Case Study: remote password guessing attack

Windows systems come with built-in security features; however, the main guard of accessing Windows systems is still passwords. Thus, remote password guessing attack is one of the common threats to Windows systems [24]. Remote password guessing is wildly used by hackers

due to the simplicity of guessing and overturning authentication credential with high chance of gaining unauthorized access [24]. Remote password guessing attacks come in many strategies based on the type and version of the deployed windows systems.

The popular strategy is to target the Window file and print sharing services using Server Message Block (SMB) protocol [24]. SMB is accessible via ports TCP 445 and 139. Another remote password guessing attack path is by Microsoft Remote Procedure Call (MSPRC) on port TCP 135. For such attack, the hacker might use automate password guessing command called enum with a specified target IP. The following is an example for enum command [24]:

```
C:\enum –D –u administrator –f Dictionary.txt \\192.168.202.44
username: administrator
dictfile: dictionary.txt
server: 192.168.202.44
(1)  administrator |
return 1326, Logon failure: unknown user name or bad password.
(2)  administrator | password
[etc.]
(10) administrator | nobody
return 1326, Logon failure: unknown user name or bad password.
(11) administrator | space
return 1326, Logon failure: unknown user name or bad password.
(12) administrator | opensesame
password found: opensesame
```

Most of SHSs require necessary information in which it is impossible to choose or design suitable hacking strategies. As the case of password guessing attack the hacker has chosen this SHS based on the necessary information illustrated in the following section.

4.2. Necessary information

Necessary information is the mean need for any SHSs to take place. This information is essential for SHSs. The necessary information is listed as: IP address of a victim's system, operating system that runs on the victim's system, opening ports on the victim's system and running services on victim's system. This subsection discusses the importance of necessary information for SHSs with the use of the case study showed previously.

4.2.1. IP address

Generally, through normal ENG, packets are delivered from one host to another in shortest path possible to assure packets delivery speed in most communication technologies. With current technologies development the reliance on the IP address (IP) is important to assure that packets are delivered to intended recipients.

Consequently, IP address is necessary information for any ENG and SHSs. In the case study of remote password guessing attack illustrated previously, the IP address (192.168.202.44) was specified by pre-hacking step called scanning (scanning will be discussed in detail in section 5).

4.2.2. Operating system

The second step after specifying the IP address of the target system is identifying which operating system (OS) that is deployed in that target system. Operating system holds responsibility of managing most functionally. Security vulnerabilities vary based on the technical specification of the deployed operating system. In another words, every operating system has different vulnerabilities that are associated with it.

Thus, identifying which operating system that manages the target IP is necessary information for any SHSs. Specifying the type and version of deployed operating system in the victim's INF is heavily relied on ANL the ENG between the hacker and victim.

Going back to the case study of remote password guessing attack, we notice that the hacker uses (enum) command line that is suitable to target Windows system. Specifying the victim's operating system is done through scanning as well.

4.2.3. Running services and opening ports

Identifying what services (RS) are and in which ports (OP) they run on the victim's system is also necessary information for any SHSs. Services and opening ports from hackers' prospective are doors for their delivered packets in which it is impossible for them to enter such system without identifying possible entrances to their nominated victims' INFs.

Services are deployed to perform specific tasks in most operating system; thus, these services may have vulnerabilities that associated with them, or might just give assistance to hackers to enter the victim's system. Based on the hacker technical skills and knowledge about deployed technologies and software, they utilize these services and opening ports to their advantage and in worst case accomplishing SHSs.

Going back to the case study of the remote password guessing attack, the hacker has effectively identified the opening ports that are ports 445 and 139, and the running service on theses ports is SMB through executing scanning operations.

The necessary information might appear basic for some security experts in which it is the necessary information for most ENG. However, it is impossible for any ENG and SHSs to be performed without such information. Hence, we define the necessary information (NI) as:

$$NI = IP \wedge OS \wedge RS \wedge OP \tag{5}$$

Most of sophisticated hackers perform three pre-hacking steps in order to obtain the NI and additional information before choose or design hacking strategies. Usually, hacking strategies are designed to suit victims' system and finish up with SHSs.

The three pre-hacking steps are: foot printing, scanning and enumeration. Such great deal of NI and additional information is to be obtained from performing these three steps. However, NI is obtained from performing scanning alone. This paper focuses on scanning as root of hacking in which it provides NI for most SHSs. However, following subsections discuss foot printing and enumeration as well to highlight the importance of scanning between foot printing and enumeration.

5. Pre-Hacking Steps

Before hackers actually break in, most skilled hackers follow the same methodology. The methodology consists of three crucial steps which are: footprinting, scanning and enumeration [24]. These three steps must be performed by most experienced hackers, which give it the phrase "The root of hacking"; see next figure. The following subsections formally examine every pre hacking step.

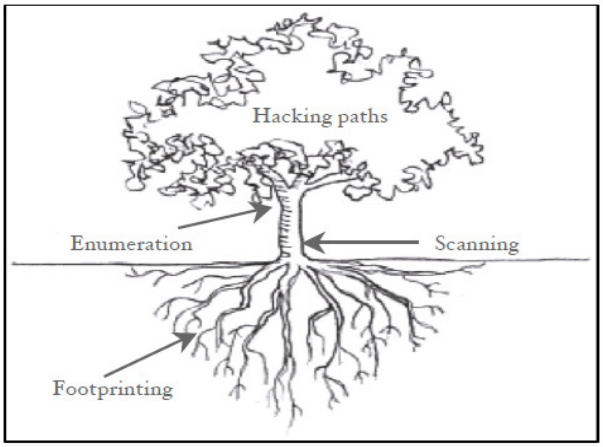

Figure 2: The behaviour of hackers.

5.1. Footprinting

Footprinting is a crafted technique in gathering information [24]. Basically, it is related to narrowing down the target of interest, investigating every entity related to the victim's INF. The hacker at this stage is trying to understand how the victim operates. They investigate the interrelation between the victim and everything around it without any ENG between the victim and hacker; In other words, no single packet is sent to the victim. For a successful focussed and surgical hack, the hacker must harvest the wealth of information about every feature of the organization's security postures [24].

Footprinting is a process, which involves execution of software operations specifically design to satisfy footprinting main objectives. Thus, we define footprinting (F) is defined as:

$$F = \{ Fo_i : i \in I \} \tag{6}$$

Fo, stands for footprinting operations, where I = {1,2,3,4,5,6…n}. The main aim of F is to end up with a unique detailed profile (UP) of the target's Internet, intranet [Info(in)], extranet [Info(ex)], remote access [Info(ra)], business partners [Info(bp)], deployed protocols [Info(pt)] and general information about security postures [Info(sp)] [24]. Thus, UP is defined as:

$$UP = Info(in) \wedge Info(ex) \wedge Info(ra) \wedge Info(bp) \wedge Info(pt) \wedge Info(sp) \quad (7)$$

A skilled hacker can narrow their target to specific domain names, IP addresses, routers, subnets and network blocks; starting from a selected victim and without a single packet being sent [24]. In F step, hackers investigate the interrelation between the victim's INF and every entity that is connected to it without sending a single packet to the victim with considerable time and effort made by hackers. So, we define F property as:

$$Fp = \forall x \in F: \neg ENG(x, INF) \quad (8)$$

After that, the hacker will proceed to the next step, which is scanning.

5.2. Scanning

By the previous step, the hacker obtains UP about their target. From that point, the hacker starts sending packets to their victim's system. Actually, the intention of developing the attack strategy (scripts and tools) is yet to be determined. In fact, they look for the point of entry to the victim's system searching for proper paths to get inside the target's system [24].

Scanning is the most critical pre-hacking step due the intrusiveness nature and critical information gained from it. It is a set of executed operations developed specifically to satisfy the scanning's main objectives. So, we define scanning as:

$$S = \{ So_i : i \in I \} \quad (9)$$

So, stands for scanning operations, Where I = {1,2,3,4,5…n}. S rely on the output of F which is UP, under the assumption that F are executed successfully and data contained in UP is sufficient enough to start S.

5.2.1. Scanning objectives

S is the most crucial process compared with F and enumeration. At this stage, the hacker searches for running services and in which ports they run and what the host that runs these services. S objectives are determining target, operating system, running services and opening ports in the intended INF. Hence, a successful execution of S operations means that the hacker has obtained NI. The following subsections examine every objective and use S tools for remote password guessing attack.

- **Determining the target:**

Determining the target (DTT) is a number of executed operations that fall within the range of So. Sophisticated hackers execute these operations to specific target within the intended INF. Basically, having predefined target eases the enumeration step and developing SHSs. It is one of the most critical steps since hackers cannot start probing without identifying which system they target. We define DTT objective as:

$$\exists\, y \in DTT \rightarrow IDENT\_targtet(\, y,\ NETWORK\_BLOCKS,\ INF) \tag{10}$$

Where IDENT_target stands for identification of the target system in the intended INF. Hackers obtain the IP address of the victims' INF by ANL the ENG.

Going back to the case study of remote password guessing attack, there are many tools that identify which system is listening for incoming traffic (which system hacker should target); however, the most common tool is network ping sweep using nmap command [24]. Nmap command operates by sending specific type of traffic such as ICMP (internet Control Message Protocol) to a list of IP addresses or network blocks of the specified victim and analyzing victim's replies. In another words, nmap sends ICMP packets to the list of IP addresses or network blocks and waits for responses. The following example illustrates how nmap identifies the target system [24].

```
[root] nmap –sp 192.168.1.0/24
starting nmap V. 4.68 by Fyodor@insecure,org (www.insecure.org/nmap/)
Host (192.168.1.0) seems to be a subnet broadcast
address (returned 3 extra ping).
Host (192.168.1.44) appears to be up
Host (192.168.1.255) seems to be a subnet broadcast
address (returned 3 extra ping).
Nmap run completed - - 256 IP addresses (10 host up) scanned in 21 seconds
```

The first objective of S in determining the IP addresses for the target system is accomplished. Via one of S tool such as nmap, hacker at this stage has effectively identified the IP address of the target system (192.168.1.44).

- **Determining the operating system:**

Every operating system requires technical specifications to execute and run applications on its environment. Managing applications and services and associating ports to these applications and services is one of the main functionality of operating systems. Thus, identifying operating (DOS) system that runs on the target INF is critical part in developing SHSs. We define DOS objective as:

$$\exists\, y \in S \rightarrow \text{IDENT\_OS}(y, \text{INF}) \tag{11}$$

Where IDENT_OS, stands for identifying the operating system that manages victim's INF. Going back to the case study of password guessing attack, there are number of tools that are used to identify the deployed operating system in the target's server. Active stack fingerprinting is a technology that identifies the deployed operating system based on IP stack implementation. Every operating system interprets RFC guidance differently during implementing their TCP/IP stack [24]. Thus, by spotting these differences, hacker can identify which the operating system they deal with by the response type from the target system. One of these tools is nmap. The following example illustrates how nmap identifies the deployed operating system [24].

[root] nmap –o 192.168.1.44

Starting nmap V. 4.68 by Fyodor@insecure.org

Interesting ports on shadow (192.168.1.44) :

Port	state	Protocol	Service
7	open	tcp	echo
9	open	tcp	discard
13	open	tcp	daytime
19	open	tcp	chargen
21	open	tcp	ftp
22	open	tcp	ssh
23	open	tcp	telnet
25	open	tcp	smtp
37	open	tcp	time
111	open	tcp	sunrpc
139	open	tcp	smb
512	open	tcp	exec
513	open	tcp	login
514	open	tcp	shell
2049	open	tcp	nfs
4045	open	tcp	lockd

TCP Sequence Prediction: Class=random positive increments

Difficulty=26590 (worthy challenge)

Remote operating system guess: Solaris 2.5, 2.51

- **Determining the running services:**

The following objective of S after DOS is to determine running services (DSR) on the specified target. The important of DSR is illustrated in identification of the listening ports and applications that runs on these ports. Such great deal of information can be extracted by sophisticated hackers from this step. So, we define DSR main objective as:

$$\exists\, y \in S \Rightarrow \text{IDENT\_APP}(y, \text{INF}) \wedge \text{IDENT\_SR}(y, \text{INF}) \tag{12}$$

Where IDENT_APP, stands for identifying the running application and IDENT_SR stands for identifying the running services on the intended INF. Going back to the case study of password guessing attack, there is one widely used scanning tool that identifies the running services and opening ports called netcat [24]. Netcat is a tool designed to perform port scanning on specific target systems over TCP or UDP protocols. Netcat probes the selected target to determine which service is in listening state. The following example illustrates how netcat command identifies the running services and opening ports, which is SMB on port 139 [24].

```
[root] nc –v –z –w2 192.168.1.44 1-140
[192.168.1.44] 139 (smb) open
[192.168.1.44] 135 (?) open
[192.168.1.44] 110 (pop-3) open
[192.168.1.44] 106 (?) open
[192.168.1.44] 81 (?) open
[192.168.1.44] 80 (http) open
[192.168.1.44] 79 (finger) open
[192.168.1.44] 53 (domain) open
[192.168.1.44] 42 (?) open
[192.168.1.44] 25 (smtp) open
[192.168.1.44] 21 (ftp) open
```

The above examples of the case study of password guessing attack showed how S operations provides NI about the nominated victim's system. A remote password guessing attack is nearly impossible to be accomplished without NI gained by the previous examples of S objectives. This paper focuses on worst case in which the hacker has successful executed S and extracted the NI. Thus, we refer to S as a successful process in which it ends up with NI. We define S objectives (SOJ) as:

$$SOJ = \{So_i : i \in I\} \rightarrow DTT \wedge DOS \wedge DSR \quad (13)$$

5.2.2. Scanning properties

During S, hackers intensively ENG with their victim by sending and receiving packets. Moreover, ANL the ENG to obtain NI, such as operating system information, requires broad knowledge about deployed technologies where was the case of identifying OS type and version in the previous case study. Thus, we define S properties (Sp) as:

$$Sp = \forall x \in S : ENG(x, INF) \wedge ANL(x, PACKET, INF) \quad (14)$$

The hacker at this point has defined all the windows and doors that can be routes for launching attacks. Then, they perform the last step just before launching their attacks, which is enumeration. The basic building block for most surgical SHSs relies heavily on the wealth of information gained from S in which it is impossible to develop SHSs without the obtained NI

5.3. Enumeration

At the final stage, the hacker has effectively recognized opening ports and running services preparing for hack. Before they form their SHS, they intensely probe the spotted services looking for known weaknesses and discovering new routes [24]. Enumeration (E) is also a number of executed operations by hackers. However, these operations are more intrusive than Fo and So. However, the obtained information from enumeration is not critical as information gained from So.

E is a process that includes active ENG and direct queries to the target's system, giving it a higher level of intrusiveness compared with S [24]. This process heavily relies on information gathered from S. Thus, we define E objective as:

$$\exists y \in E \rightarrow Make(y, app, INF) \wedge Model(y, app, INF) \quad (15)$$

Basically the E objectives are mostly related to the identified applications on the intended INF. The hacker in E step tries to probe the target to identify the make [Make (y, app, INF)] and model [Model (y, app, INF)] of the spotted services. At the first glance, the hacker seeks for misconfiguration of shared resources (for instance, unsecured share systems), user names as an important factor for dictionary attacks and common security vulnerabilities related to the spotted services such as remote buffer overflow for a web server [24]. E operations share similar properties as S, which we define it as:

$$\forall x \in E : ENG(x, INF) \wedge ANL(x, PACKET, INF) \quad (16)$$

6. S FOR NI

Developing SHSs require intensive knowledge about victims' INFs. It does require deep knowledge about the deployed technologies and security postures in victims' INFs. Acquiring knowledge requires successful completion of executed operations by hackers to specifically extract NI from S and additional information from F and E to assist developing of SHSs.

The main aim of F is to reduce the scope of interest to specific IP addresses and domains. Moreover, F is aimed to study how the victim operates and interrelation between the victim and allied organizations. The amount of information gained by the hacker form F encloses information about internet, intranet, remote access and extranet. Then, the hacker proceeds to next step, which is S [24].

S is the most critical pre-hacking steps because of the NI obtained from it. The hacker at this stage tries to identify points of entry of the victim's INF. Successful execution of sophisticated S may identify which system is listing for incoming traffic. Although, it can identify the running services on the specified victim's system and deployed operating system. Such great deal of information extracted from successful execution of S which will limit hackers use for specific tools, scripts and applications to that specific operating system, services, listening system. Then, the hacker completes gathering additional information by perfuming E [24].

E operations are considered most intrusive compared with S. However, the S is more critical because of the NI gained from successful execution of So. Information gained from E is mainly to specify make and model of installed application on the victim's system. This information won't be possible to gain without identifying the listening system, the operating system that runs on victim's system and the running services and opening ports.

Hence, S is the most critical pre-hacking step in which it provide NI to be essential factor for any SHS. This thesis considers the worst case, which assumes that S is executed successfully, and the NI is gained from execution of S. Thus we define the relation between S and NI as follows:

$$S \rightarrow NI \quad (17)$$

7. S AND SHS

Based on the information acquired from S, the hacker at this stage develops a suitable SHS to break into their victim's system. Counting SHSs is impractical. The sophistication in hacking techniques and the limitation associated with current security solutions escalate the success rate for many hacking attempts. However, most of experienced hackers perform S in order to break into a system.

S is performed to accomplish a great task, acquiring NI about the victims' systems. Every SHS requires specific information about how victims operate and what the deployed technologies in victims' INF, without this information developing hacking strategies is nearly impossible. So, we define the relation between NI and SHS as:

$$\neg \text{ NI} \rightarrow \neg \text{ SHS} \tag{18}$$

Communication between hackers and victims comes in form of packets. Packets are sent to victims by hackers and vice versa. These packets are impossible to deliver without specifying the IP address on the victim's system. Although, data presents in transformed packet are generate by scripts or tools designed specifically to accomplish specific function on the victim's system. So, without knowing the open ports, services running on these ports and operating system, developing specific tools and script of data forming on the transferred packets is not possible. Thus, SHS is addressed via deterring S operations. Hence, we define the importance of S to address SHS as follows:

$$\neg \text{ S} \rightarrow \neg \text{ SHS} \tag{19}$$

Nevertheless, it is important to highlight S properties in order to address SHS. So, the following section discusses S properties, which are ENG and ANL.

8. Deterring Shs Via Targeting S Properties

As it has been discussed previously in this paper, SHS requires NI in which it is impossible for SHS to be selected or implemented in the first place without it. NI includes known exactly the IP address of the victim's system, the deployed operating system in the specified victim, running services and in which ports they run on the victim's system. The NI is obtained when the hacker successful executes S. Thus, the main purpose of S is to end up with NI.

In addition, S requires some technical specification which is not possible to perform S and some network functionalities without them. These technical specification where illustrated previously in this paper as communication risk (section 3.1). These technical specifications are defined in section 5.2 as S properties. These properties are ENG and ANL. The following subsections discuss ENG and ANL in more details with the use of case study of remote password guessing attack.

8.1. ENG

ENG is sequence of communication between the hacker and their victim. It occurs when the hacker is sending threat objective packets to the victim and victim replies back to them with packets. Packets sent from the hacker are threat objective packets, which leads to NI when the hacker successful executes S.

In some cases, ENG does not occur between the hacker and their victim such as when the victims system is switch of or hanged. In this thesis, no ENG means that at least one condition of ENG is not occurring.

When the hacker is in no ENG condition, it means one of the two following conditions:

- The hacker is sending threat objective packets to the victim. The victim is not replying back to the hacker.
- The hacker is sending threat objective packets to the victim and packets are dropped or not delivered during the ENG.

The following figure illustrated the conditions of no ENG.

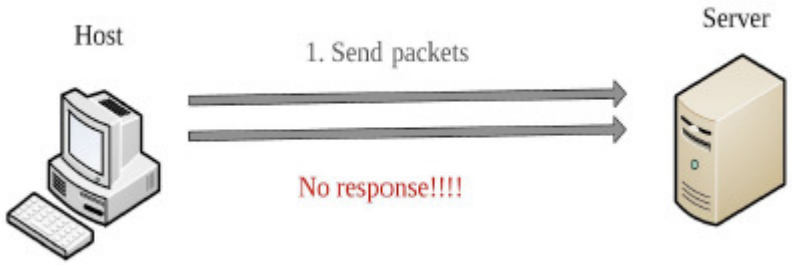

Figure 3: None ENG.

Hence, we define none ENG as:

$$\neg \text{ ENG} = \langle \text{Send(packet)}^1, \ldots\ldots \text{ Send(packet)}^n \rangle \tag{20}$$

In another words, ¬ENG is occurring when the nominated victim does not reply to the hacker. Through successful execution of S, the replied packets from the victim to hacker hold NI. Therefore, when hacker does not receive replied packets from the victim, obtaining NI is nearly impossible. We defined the relation between none ENG and S as follows:

$$\neg \text{ ENG} \rightarrow \neg \text{ S} \tag{21}$$

The following subsections illustrate the importance of ENG and ANL with the use of remote password guessing attack.

8.1.1. ENG for DTT

Going back to the case study of password guessing attack described previously, in the first objective of S which is DTT, the hacker sends threat objective packets to block of IP addresses to determine which host is alive. Identifying the exact IP address to target requires from the victim to reply to the hacker. The following figure illustrates how hackers identify which system to target.

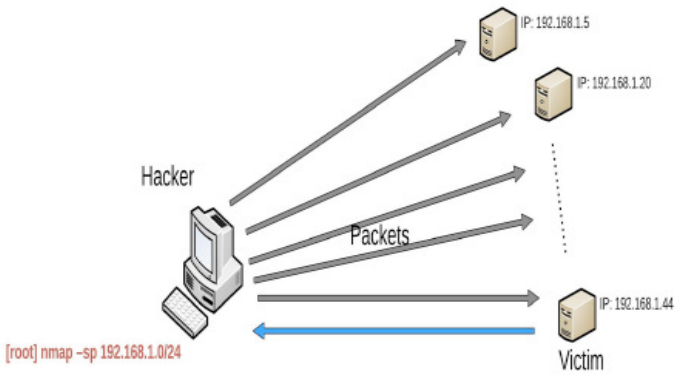

Figure 5: ENG during DTT.

DTT IP address is essential for most of the communication within INF. It is not possible hosts to communicate without sending and receiving packets. Thus, as it has been illustrated in the previous figure the hacker has effectively identified the IP address of the target INF by receiving a reply from (192.168.1.44).

8.1.2. ¬ ENG for DTT

When ENG is not satisfied, the possibility of identifying the actual IP address of the target system is impossible. ¬ENG property means that the hacker sends threat objective packets, which is the case of DTT and receives none replies. The following figure illustrates how the hacker sends packets to block of IP addresses and receives no reply.

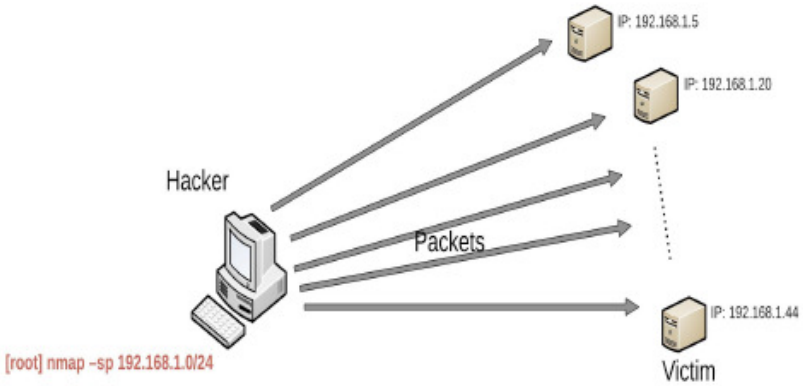

Figure 5: ¬ENG during DTT.

In this case identifying which system to target is not possible, since DTT heavily relies on receiving replied packets from the victims. Thus the expected output after execution nmap command line for DTT is:

[root] nmap –sp 192.168.1.0/24
starting nmap V. 4.68 by Fyodor@insecure,org (www.insecure.org/nmap/)
Host (192.168.1.0) seems to be a subnet broadcast address (returned 3 extra ping).
Note: Host seem down. If it is really up, but blocking our ping probes.
Host (192.168.1.255) seems to be a subnet broadcast
address (returned 3 extra ping).
Nmap run completed - - 256 IP addresses (0 host up) scanned in 30 seconds

8.1.3. ENG for DOS

DOS is critical information since SHSs vary based on the type and version of OS that manages the victim's system. Remote password guessing attack is commonly associated with Windows system. The reason behind that is Windows system is still placing reliance on passwords as main guard to for access. Hence, remote password guessing attack is effective hacking strategies directed to Windows system. The following figure shows how the hacker ENG with the victim and successful identified the OS.

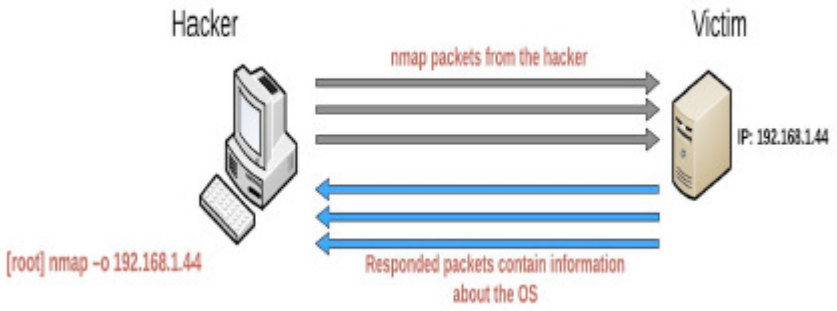

Figure 6: ENG for DOS.

As it has been shown in the previous figure, the hacker required ENG property in order to obtain the type and version of OS. It is impossible to obtain OS type and version without receiving replied packet with required information to the hacker.

8.1.4. ¬ENG for DOS

Let assume that hacker has effectively DTT of the victim, the following step of S objective is DOS. ENG property is important through DOS of the target system. It is impossible to DOS without the presence of ENG between the hacker and their victim. The following figure illustrates execution of nmap command line for DOS when ENG is not satisfied.

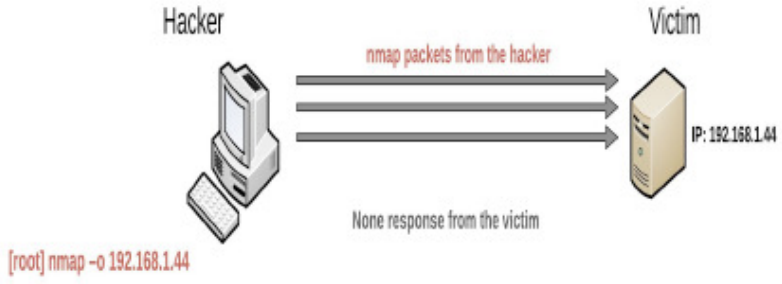

Figure 7: ¬ ENG for DOS.

The expected output of executing nmap without satisfying ENG property would appear to the hacker as:

[root] nmap –o 192.168.1.44
Starting nmap V. 4.68 by Fyodor@insecure.org
Nmap scan report of [hostname] (192.168.1.144)
Not shown: 1000 closed ports
TCP Sequence Prediction: Class=random positive increments
Difficulty=26590 (Good luck!)
Remote operating system guess: **No exact OS match for host**

8.1.5. ENG for DSR

Information is DSR. DSR is essential as DTT and DOS. Vulnerabilities within INF may appear from countless factors and running services is one of them. ENG is critical for DSR since it heavily depend on the replied packets that contain information about running services and opening ports. The following figure highlights the importance of ENG for DSR of remote password guessing attack.

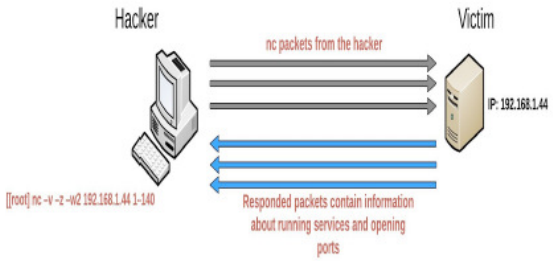

Figure 7: ENG for DSR.

8.1.6. ¬ENG for DSR

Let assume that the hacker has effectively DTT and DOS of the target victim. The last step is to obtain one final NI, which is DSR. The same concept of DTT and DOS of the remote password guessing attack is applied also on DSR. The following figure illustrate how important is ENG property for DSR.

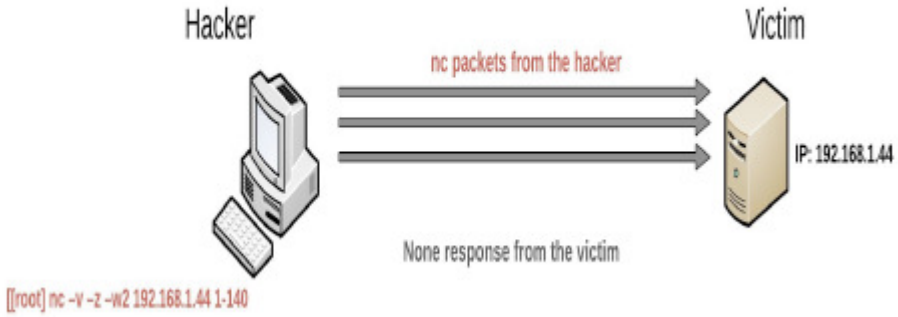

Figure 8: ¬ENG for DSR.

The expected output of running netcat command line would appear as:

[root] nc –v –z –w2 192.168.1.44 1-140
Connection to 192.168.1.44 port 1-140 failed: connection timeout

8.2 ANL

ANL is a process consists of collecting communication traffic, which is in form of packets and extracts NI information and additional information from these packets. ANL is critical step formost SHS. The collected packets may be obtained from the ENG between the hacker and their victim or from ENG between two or more legitimate users. ANL is the second S properties in which, performing a successful S that assists hackers with NI is impossible to be accomplished without ANL.

Extracting NI information from transmitted packets requires from hackers to know the protocol that defines structures and format of bits of packets. There are many protocols that widely used for communications such TCP/IP protocol or OSI seven layers network protocol. In order for two computers systems to communicate, the structure (ks) and formation (kf) of bit within packets must be agreed on or ENG is not possible. Hence, we define ANL requirements (ANLr) as:

$$ANLr = ks\ (packet^{1}, packet^{'1} \ldots packet^{n}, packet^{'n}) \wedge kf\ (packet^{1}, packet^{'1} \ldots packet^{n}, packet^{'n}) \quad (22)$$

Packets structure and format is critical requirement for the ANL process since it is impossible of hackers to understand the information presents in the transmitted packets, which is applied to the use of cryptography. So, when ks or kf are not satisfied, that leads to ¬ANL. We define the relationship between S and ANL as:

$$\neg\ ANL \rightarrow \neg\ S \quad (23)$$

The following subsection illustrates the importance of ANL for every S objective.

8.2.1. ANL for DTT

Attack described previously, the hacker sends threat objectives packets to block of IP addresses and waits for the reply, which identifies the actual IP address of the victim. Identifying the actual IP address requires that the hacker knows the deployed protocol.

Since the hacker and victim are using TCP/IP protocol, the structure and formation of the packets are already known for the hacker. ANL the packets to extract IP address in this case is simple since the source and destination IPs are presented clearly in the transmitted packet, see the following figure.

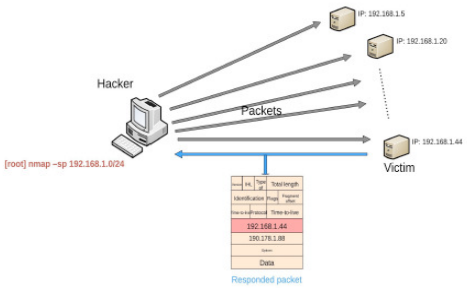

Figure 9: ANL for DTT.

8.2.2. ¬ ANL for DTT

Extracting NI from unknown packet format become and complex task for hackers. For nmap command line as an example the source and destination are presented in most network protocols that is impossible for packets to be delivered from one host to another. The following figure illustrates the ENG between the hacker and victim without an agreed network protocol.

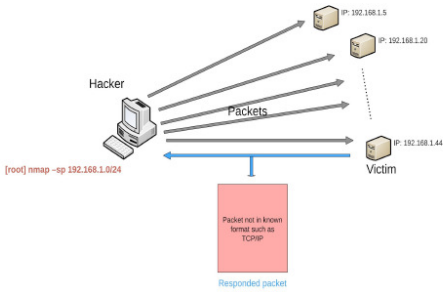

Figure 10: ¬ ANL for DTT.

However, the hacker can manually identify the target IP address by analyzing the source IP address in the sent packet but it will consume considerable time.

8.2.3. ANL for DOS

Extracting information about the deployed operating system in the victim's INF is more complex than DTT. This process requires ANL sequences of replied packets from the victim, in which they may presents a clue about the victims implementation such as interprets RFC guidance for TCP/IP. The following figure illustrates the importance of ANL for DOS. To simplify the ANL process we assume that OS information is included in data field of TCP/IP protocol.

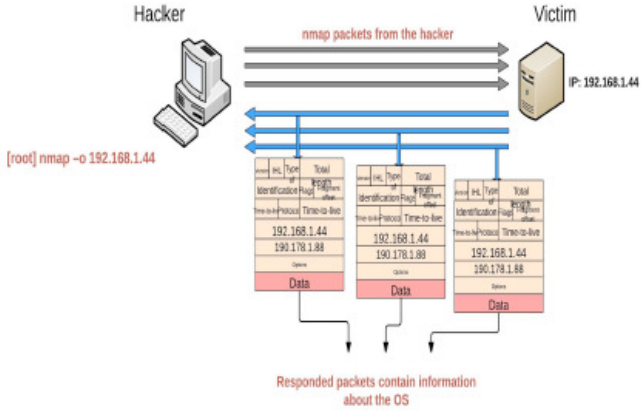

Figure 11: ANL for DOS.

8.2.4. ¬ ANL for DOS

ANL the packets to obtain DOS about the victim's system is complex, however, there are great deal of tools and scripts that assist hackers. Nevertheless, these tools require predefined and known protocol such as TCP/IP or OSI network layer protocols. Thus, the expected result of running nmap command line for the case study may appear as:

[root] nmap –o 192.168.1.44
Starting nmap V. 4.68 by Fyodor@insecure.org
Nmap scan report of [hostname] (192.168.1.144)
Not shown: 1000 closed ports
TCP Sequence Prediction: Class=random positive increments

Difficulty=26590 (Good luck!)

Remote operating system guess: No exact OS match for host

8.2.5. ANL for DSR

The importance for ANL for DSR is not less than DTT and DOS. DSR is also complex process of ANL sequence of packets between the hacker and victim. A tool such as netcat ENGs with the victim and tries all possible ports and services to see which packets they receive from the victim which is the main indicator of running services and opening ports, see the following figure.

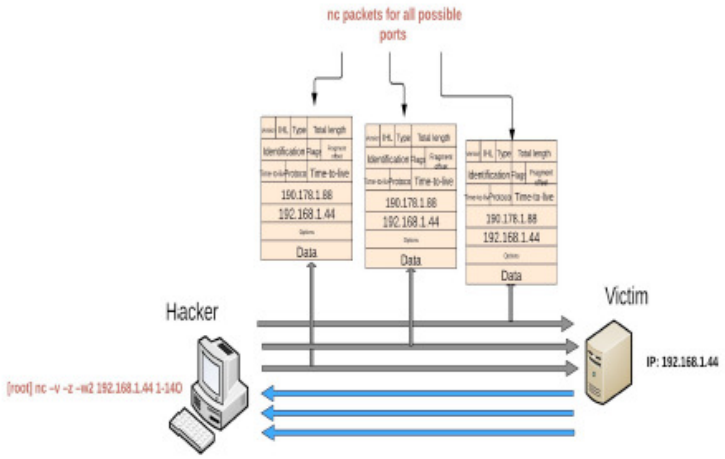

Figure 21: ANL for DSR.

8.2.6. ¬ ANL for DSR

None satisfaction of ANL for DSR makes extracting NI a complex task. The importance of ANL to DSR is illustrated in the expected output of running netcat command line for the purpose of obtaining DSR.

[root] nc –v –z –w2 192.168.1.44 1-140
Connection to 192.168.1.44 port 1-140 failed: unexpected response

8.3 Summary of ENG & ANL

The following table summarizes the pervious techniques and expected results from running some S tools while ENG and ANL are satisfied and none satisfied to illustrates the importance of these properties.

S properties	Commands	Result
ENG and ANL for DTT	[root] nmap –sp 192.168.1.0/24	Nmap run completed - - 256 IP addresses (10 host up) scanned in 21 seconds
ENG and ANL for DOS	[root] nmap –o 192.168.1.44	Remote operating system guess: **Solaris 2.5, 2.51**
¬ENG and ANL for DSR	[root] nc –v –z –w2 192.168.1.44 1-140	[192.168.1.44] 139 (smb) open
¬ ENG or ¬ ANL for DTT	[root] nmap –sp 192.168.1.0/24	Nmap run completed - - 256 IP addresses (0 host up) scanned in 30 seconds
¬ENG or ¬ ANL for DOS	[root] nmap –o 192.168.1.44	Remote operating system guess: **No exact OS match for host**
¬ENG or ¬ ANL for DSR	[root] nc –v –z –w2 192.168.1.44 1-140	Connection to 192.168.1.44 port 1-140 failed: connection timeout

Table 1: Satisfying and none satisfying of S properties.

9. Conclusion

SHS is one common threat for most INFs. SHSs differ based on countless factors such as the type of deployed technologies including hardware and software. However, most of SHSs require common NI. The possibility of developing SHS without NI is nearly none which includes, the IP address of the victim INF, the operating system that manages the victim's INF, the running services on victim's INF and the opening ports in the victim's INF.

NI can be obtained by performing S. S is one of pre-hacking steps performed by most hackers for one purpose that is acquiring NI. S requires two essential properties in which it impossible to obtain NI without them. The S properties are: ENG and ANL.

This paper examines the possibility of addressing SHS by designing a secure system that addresses S properties (ENG, ANL). Target ENG and ANL to deter S which leads to a disclosure of the NI from being obtained by hackers. Then, the possibility of developing SHS is almost none.

There are great challenges for security experts and researchers towards addressing hacking strategies. However, we believe that security solutions should be focused on controlling ENG and eliminating the possibility to ANL.

References

[1] K. Rawlinson. (2015, 9/1/2015). Available: http://www.bbc.com/news/technology-30744834

[2] Dehlawi, Z.; Abokhodair, N., "Saudi Arabia's response to cyber conflict: A case study of the Shamoon malware incident," Intelligence and Security Informatics (ISI), 2013 IEEE International Conference on , pp.73,75, 4-7 June 2013.

[3] S. Bratus, "What Hackers Learn that the Rest of Us Don't: Notes on Hacker Curriculum," Security & Privacy, IEEE, vol. 5, pp. 72-75, 2007.

[4] C. Payne and T. Markham, "Architecture and applications for a distributed embedded firewall," Computer Security Applications Conference, 2001. ACSAC 2001. Proceedings 17th Annual, pp. 329-336, 2001

[5] A. X. Liu and M. G. Gouda, "Diverse firewall design," Parallel and Distributed Systems, IEEE Transactions on, vol. 19, pp. 1237-1251, 2008.

[6] A. X. Liu and M. G. Gouda, "Firewall Policy Queries," Parallel and Distributed Systems, IEEE Transactions on, vol. 20, pp. 766-777, 2009.

[7] Y. Bartal, A. Mayer, K. Nissim, and A. Wool, "Firmato: a novel firewall management toolkit," in Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on, 1999, pp. 17- 31.

[8] F. Avolio, "Firewalls and Internet security, the second hundred (Internet) years," The Internet Protocol Journal, vol. 2, pp. 24-32, 1999.

[9] C. C. Center, "CERT Advisory CA-2003-20 W32/Blaster Worm," Available At http://www.cert.org/advisories/CA-2003-20.html, 2003.

[10] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "The spread of the sapphire/slammer worm, 2003," Available At :http://www.caida.org/outreach/papers/2003/sapphire/sapphire.html.[11] M. Roesch, "Snort-lightweight intrusion detection for networks," in Proceedings of LISA '99: 13'th Systems Administration Conference, 1999, pp. 229-238.

[12] Y. Lin, Y. Zhang, and Y.-j. Ou, "The Design and Implementation of Host-Based Intrusion Detection System," in Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium on, 2010, pp. 595-598.

[13] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection," Network, IEEE, vol. 8, pp. 26-41, 1994.

[14] D. Goldsmith and M. Schiffman, "Firewalking: A traceroute-like analysis of IP packet responses to determine gateway access control lists," Cambridge Technology Partners, vol. Available At: http://www.packetfactory.net/firewalk/firewalk- final.html, 1998.

[15] G. A. Marin, "Network security basics," Security & Privacy, IEEE, vol. 3, pp. 68-72, 2005.

[16] D. Mutz, G. Vigna, and R. Kemmerer, "An experience developing an IDS stimulator for the black-box testing of network intrusion detection systems," in Computer Security Applications Conference, 2003. Proceedings. 19th Annual, 2003, pp. 374-383.
[17] M. Sourour, B. Adel, and A. Tarek, "Environmental awareness intrusion detection and prevention system toward reducing false positives and false negatives," in Computational Intelligence in Cyber Security, 2009. CICS '09. IEEE Symposium on, 2009, pp. 107-114.
[18] T. Pietraszek and A. Tanner, "Data mining and machine learning—Towards reducing false positives in intrusion detection," Information Security Technical Report, vol. 10, pp. 169-183, 2005.
[19] L. Spitzner, Honeypots: tracking hackers: Addison-Wesley Professional, 2003.
[20] L. Spitzner, "The Honeynet Project: trapping the hackers," Security & Privacy, IEEE, vol. 1, pp. 15-23, 2003.
[21] B. Jian, J. Chang-peng, and G. Mo, "Research on network security of defense based on Honeypot," in Computer Application and System Modeling (ICCASM), 2010 International Conference on, 2010, pp. V10-299-V10-302.
[22] C. Rong and Y. Geng, "Honeypots in blackhat mode and its implications [computer security]," in Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference on, 2003, pp. 185-188. [23] L.-j. Zhang, "Honeypot-based defense system research and design," in Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on, 2009, pp.466-47.
[24] S. McClure, J. Scambray, and G. Kurtz, Hacking Exposed: Network Security Secrets and Solutions, Fourth Edition: McGraw-Hill, Inc., 2003.
[25] S. Frei, B. Tellenbach, and B. Plattner, "0-Day Patch Exposing Vendors (In) security Performance," BlackHat Europe, Amsterdam, NL, 2008.
[26] T. J. Holt and M. Kilger, "Techcrafters and Makecrafters: A Comparison of Two Populations of Hackers," in Information Security Threats Data Collection and Sharing, 2008. WISTDCS '08. WOMBAT Workshop on, 2008, pp. 67-78.
[27] S. Bratus, "Hacker Curriculum : How Hackers Learn Networking," Distributed Systems Online, IEEE, vol. 8, pp. 2-2, 2007.