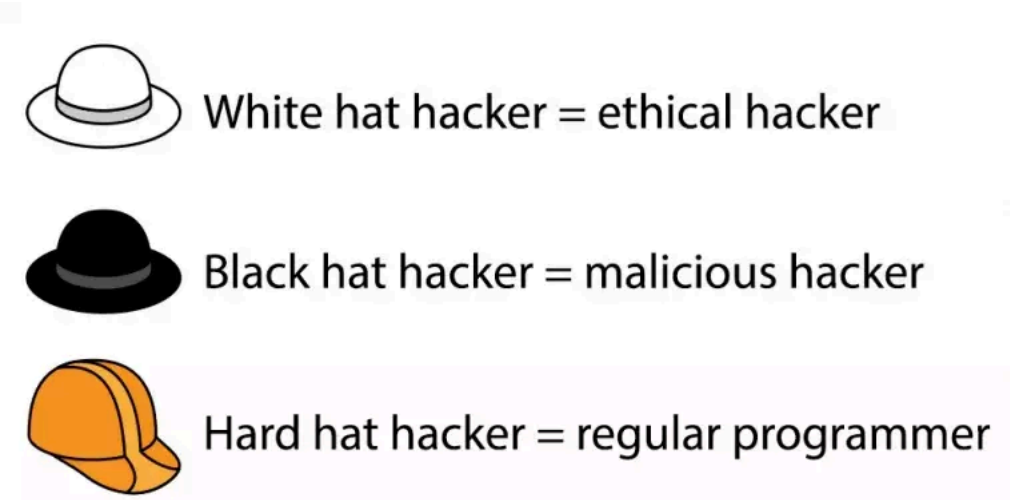
Cybersecurity

Penetration Test Report

Rekall Corporation

Penetration Test Report

<u>Student Note</u>: Complete all sections highlighted in yellow.



Confidentiality Statement

This document contains confidential and privileged information from Rekall Inc. (henceforth known as Rekall). The information contained in this document is confidential and may constitute inside or non-public information under international, federal, or state laws. Unauthorised forwarding, printing, copying, distribution, or use of such information is strictly prohibited and may be unlawful. If you are not the intended recipient, be aware that any disclosure, copying, or distribution of this document or its parts is prohibited.

Table of Contents

Contact Information

Company Name	WhitHack
Contact Name	Cameron Whitford
Contact Title	Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	14/10/2024	Cameron J Whitford	Initial Draft
002	19/10/2024	Cameron J Whitford	Finalised Document

Introduction

In accordance with Rekall policies, our organisation conducts external and internal penetration tests of its networks and systems throughout the year. The purpose of this engagement was to assess the networks' and systems' security and identify potential security flaws by utilising industry-accepted testing methodology and best practices.

For the testing, we focused on the following:

- Attempting to determine what system-level vulnerabilities could be discovered and exploited with no prior knowledge of the environment or notification to administrators.
- Attempting to exploit vulnerabilities found and access confidential information that may be stored on systems.
- Documenting and reporting on all findings.

All tests took into consideration the actual business processes implemented by the systems and their potential threats; therefore, the results of this assessment reflect a realistic picture of the actual exposure levels to online hackers. This document contains the results of that assessment.

Assessment Objective

The primary goal of this assessment was to provide an analysis of security flaws present in Rekall's web applications, networks, and systems. This assessment was conducted to identify exploitable vulnerabilities and provide actionable recommendations on how to remediate the vulnerabilities to provide a greater level of security for the environment.

We used our proven vulnerability testing methodology to assess all relevant web applications, networks, and systems in scope.

Rekall has outlined the following objectives:

Table 1: Defined Objectives

Objective
Find and exfiltrate any sensitive information within the domain.
Escalate privileges.
Compromise several machines.

Penetration Testing Methodology

Reconnaissance

We begin assessments by checking for any passive (open source) data that may assist the assessors with their tasks. If internal, the assessment team will perform active recon using tools such as Nmap and Bloodhound.

Identification of Vulnerabilities and Services

We use custom, private, and public tools such as Metasploit, hashcat, and Nmap to gain perspective of the network security from a hacker's point of view. These methods provide Rekall with an understanding of the risks that threaten its information, and also the strengths and weaknesses of the current controls protecting those systems. The results were achieved by mapping the network architecture, identifying hosts and services, enumerating network and system-level vulnerabilities, attempting to discover unexpected hosts within the environment, and eliminating false positives that might have arisen from scanning.

Vulnerability Exploitation

Our normal process is to both manually test each identified vulnerability and use automated tools to exploit these issues. Exploitation of a vulnerability is defined as any action we perform that gives us unauthorised access to the system or the sensitive data.

Reporting

Once exploitation is completed and the assessors have completed their objectives, or have done everything possible within the allotted time, the assessment team writes the report, which is the final deliverable to the customer.

Scope

Prior to any assessment activities, Rekall and the assessment team will identify targeted systems with a defined range or list of network IP addresses. The assessment team will work directly with the Rekall POC to determine which network ranges are in-scope for the scheduled assessment.

It is Rekall's responsibility to ensure that IP addresses identified as in-scope are actually controlled by Rekall and are hosted in Rekall-owned facilities (i.e., are not hosted by an external organisation). In-scope and excluded IP addresses and ranges are listed below.

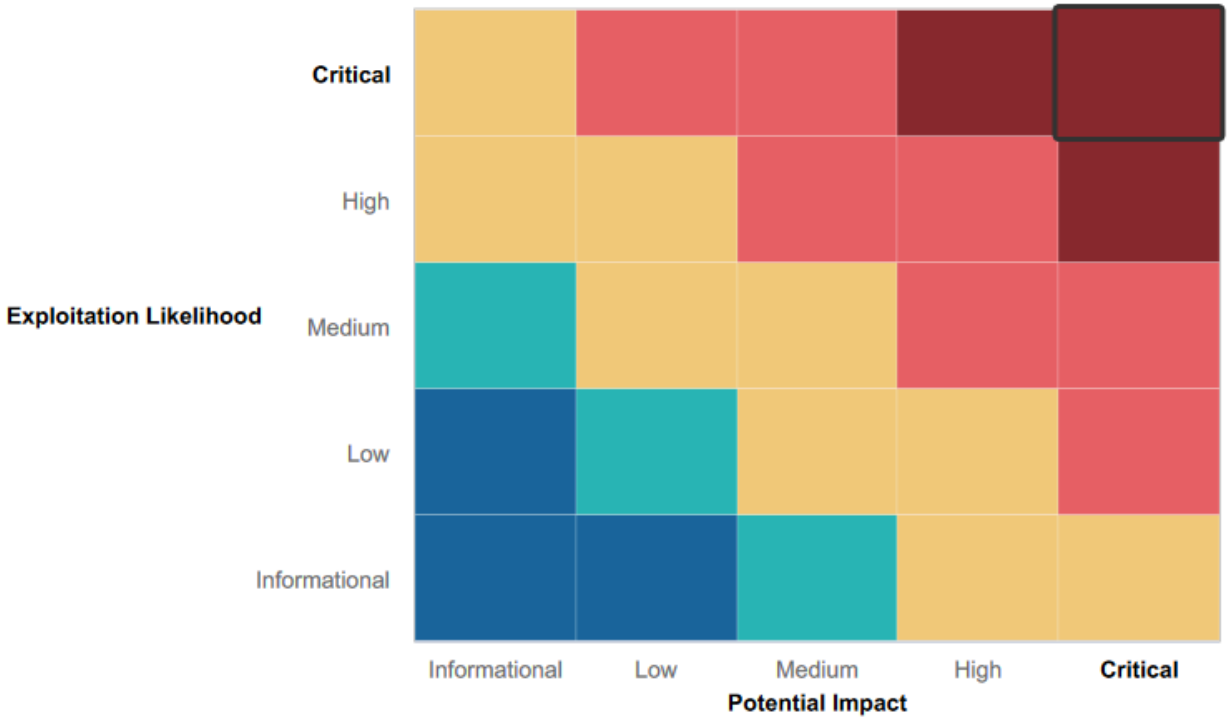
Executive Summary of Findings

Grading Methodology

Each finding was classified according to its severity, reflecting the risk each such vulnerability may pose to the business processes implemented by the application, based on the following criteria:

Critical: Immediate threat to key business processes.
High: Indirect threat to key business processes/threat to secondary business processes.
Medium: Indirect or partial threat to business processes.
Low: No direct threat exists; vulnerability may be leveraged with other vulnerabilities.
Informational: No threat; however, it is data that may be used in a future attack.

As the following grid shows, each threat is assessed in terms of both its potential impact on the business and the likelihood of exploitation:



Summary of Strengths

While the assessment team was successful in finding several vulnerabilities, the team also recognized several strengths within Rekall's environment. These positives highlight the effective countermeasures and defences that successfully prevented, detected, or denied an attack technique or tactic from occurring.

Web App
- WhitHack Corp had difficulties while attempting SQL injections on some of the input fields within the web application.
- Basic protections were used creating some difficulty when executing exploits like Cross-Site Scripting (XSS) or Local File Inclusion on some pages

Linux
- A number of the ports were not exploitable with currently known exploits
- A number of directories and files had correct permissions assigned

Windows
- A number of the ports were not exploitable with currently known exploits
- Some users had stronger passwords that would be difficult to brute force

Summary of Weaknesses

We successfully found several critical vulnerabilities that should be immediately addressed in order to prevent an adversary from compromising the network. These findings are not specific to a software version but are more general and systemic vulnerabilities.

Web App
- Areas of the Welcome.php, Memory-Planner.php and comments.php was vulnerable to cross-site scripting (XSS)
- Sensitive data is exposed within HTTP Response headers on About-Rekall.php and in the HTML of the Login.php page
- Multiple fields of Memory-Planner.php vulnerable to Local file inclusions
- SQL Injections are possible in certain fields of the Login.php
- Robots.txt is publically available
- Command injections were possible on the networking.php due to lacks of input validation
- Weak log-in credentials created possibilities for brute force attacks
- Hidden web pages with sensitive data were still available
- Directories not protected and easily traversable

Linux
- Data publicly exposed through OSINTs Domain Dossier web page and certificate look-up tools
- Scanning the subnet revealed a number of open ports across the machines with known vulnerabilities
- Weak user passwords allowed for successful brute force attacks
- A lack of updated software allowed for security bypass exploits to be used

Windows
- Users password hashes available publicly through github page
- Scanning the subnet revealed a number of open ports with known vulnerabilities
- Anonymous FTP login allowed
- Lack of management/moderation of scheduled tasks
- Lack of permissions assigned to a number of sensitive directories and files within the systems making things such as password hashes available

Executive Summary

This report summarises the findings from a comprehensive penetration testing engagement conducted on various components of the Rekall infrastructure. The objective was to identify vulnerabilities, assess security controls, and provide recommendations for remediation. The assessment uncovered multiple security weaknesses that could potentially be exploited by malicious actors.

Key Findings

1. Information Disclosure

Sensitive data was discovered within publicly accessible directories and repositories, indicating a lack of proper access controls. This exposure included user credentials, which could lead to unauthorised access to critical systems.

2. Credential Exploitation

The assessment identified opportunities for credential theft and exploitation, including weak password policies and inadequate protections for stored user credentials. Compromised accounts could allow attackers to gain unauthorised access to sensitive resources.

3. Remote File Access

Insecure configurations allowed for unauthorised file access via FTP and other services, exposing sensitive information that should have been restricted from public view.

4. Vulnerabilities in Services

Several services were found to have vulnerabilities that could be exploited, including known weaknesses in email protocols and web applications. These vulnerabilities pose significant risks to the overall security posture of the organisation.

5. Privilege Escalation

The testing revealed potential paths for privilege escalation, indicating that unauthorised users could gain elevated access to sensitive areas of the infrastructure, undermining the principle of least privilege.

6. Code Injection and Cross-Site Scripting

Multiple instances of code injection and cross-site scripting vulnerabilities were identified, allowing for unauthorised execution of scripts that could compromise user data and application integrity.

Conclusion

The penetration testing engagement highlighted numerous vulnerabilities within the Rekall infrastructure, including data exposure, ineffective access controls, and exploitable service vulnerabilities. It is crucial for the organisation to address these issues promptly through a robust remediation strategy, enhancing security controls and implementing best practices in vulnerability management.

Recommendations

- Implement Strong Access Controls: Restrict access to sensitive directories and files, ensuring only authorised personnel have the necessary permissions.
- Enhance Input Validation: Strengthen validation mechanisms to mitigate risks associated with code injection and cross-site scripting vulnerabilities.
- Regular Vulnerability Scanning: Conduct routine vulnerability assessments and penetration tests to identify and remediate potential weaknesses in the environment.
- User Education and Awareness: Train staff on security best practices, including recognizing phishing attempts and safeguarding credentials.
- Incident Response Planning: Develop and maintain an incident response plan to effectively respond to future security incidents.

By taking these steps, the organisation can significantly improve its security posture and reduce the risk of data breaches and other cyber threats.

Flag Summary

Web App

Flag 1
Found by inserting "<script>alert('x')</script>" into the input field on the Welcome.php page, exploiting Cross-Site Scripting (XSS).

Flag 2
Input validation removed the word "script," but it can be bypassed by entering ‘sscriptcript’. Doing this on the Welcome.php page reveals flag 2.

Flag 3
On the comments.php page, using "<script>alert('x')</script>" again allows access to flag 3 via XSS.

Flag 4
Flag 4 is hidden in the HTTP response headers of the About-Rekall.php page. Access it through network tools or command line tools like curl.

Flag 5
Local File Inclusion (LFI) on the Memory-Planner.php page reveals flag 5 by uploading a script.php file.

Flag 6
Similar to flag 5, LFI is present on another page. The input validation checks for ".jpg," which can be bypassed by naming the file script.jpg.php, revealing flag 6.

Flag 7
On the Login.php page, a basic SQL injection with x' OR '1'='1 allows successful login and access to flag 7.

Flag 8
The admin login credentials are visible in the HTML of the Login.php page.

Flag 9
Access the robots.txt file through directory traversal on the disclaimers.php page to find flag 9.

Flag 10
After logging in via the credentials from flag 8, code injection using && or ; in the URL allows reading the vendors.txt file, revealing flag 10.

Flag 11
In the second input field of the networking.php page, input validation prevents & and ;. Using | instead allows access to flag 11.

Flag 12
Using directory traversal, I accessed the passwd file in the etc directory, discovering the username “melina”. A brute force attack with Burp Suite using the credentials melina:melina led to access to admin_legal_data.php and flag 12.

Flag 13
Based on the robots.txt from flag 9, the souvenirs.php page was accessed. PHP code injections in the URL revealed flag 13.

Flag 14
Using the credentials melina:melina, I accessed admin_legal_data.php. By testing various session IDs via Burp (repeater or intruder), I gained access to the restricted area where flag 14 was found.

Flag 15
Exploring all available files through command injection revealed a directory named "old_disclaimers." Within it, a file called disclaimer_1.txt was found, which contained the final flag.

Linux Systems

Flag 1
Using a Domain Dossier Tool on the provided totalrekall.xyz revealed exposed data, including flag 1 listed as “Registrant Street.”

Flag 2
Ping the totalrekall.xyz domain to find the IP address. It can also be found using online address lookup tools. (Note: The IP has changed since the flag setup, so flag 2 was not retrievable.)

Flag 3
Searching for totalrekall.xyz on a certificate lookup site (like crt.sh) exposes several certificates, which reveals flag 3.

Flag 4
Running an Nmap scan on the network 192.168.13.0/24 identified 5 live hosts, excluding the host being scanned.

Flag 5
An aggressive Nmap scan indicated that the host 192.168.13.13 runs Drupal.

Flag 6
A Nessus scan of 192.168.13.12 identified one critical vulnerability with plugin ID #97610.

Flag 7
Using msfconsole to search for JSP exploits, I found exploit/multi/http/tomcat_jsp_upload_bypass. After setting the parameters, I gained shell access and retrieved flag 7.

Flag 8
After testing various exploits, I discovered the 'shellshock' vulnerability. By running the exploit, I gained shell access and viewed the sudoers file, where flag 8 was hidden.

Flag 9
While in a Meterpreter shell session, I accessed the passwd file in the /etc directory and found flag 9.

Flag 10
Research indicated the host was vulnerable to the ‘Struts’ exploit. After configuring and using this exploit, I created a session and downloaded a flag zip file. Upon unzipping it, I found flag 10.

Flag 11
Searching for ‘Drupal’ exploits in msfconsole led to an exploitable vulnerability. After setting the correct parameters and running it, I initiated a Meterpreter session on the host and retrieved the username using ‘getuid’.

Flag 12
The whois data from flag 1 revealed ‘sshuser Alice.’ I SSHed into the server with ssh alice@192.168.13.14, guessing the password (‘alice’) to gain shell access. From there, I exploited a

known vulnerability to escalate privileges and accessed the root directory, where flag 12 was located.

Windows System

Flag 1
Searching through GitLab revealed the /xampp.users directory, containing the username and password hash for "trivera." Using John the Ripper to crack this hash provided the login credentials for trivera.

Flag 2
Conducting a port scan of the subnet identified two additional hosts. One open port on the Windows10 machine was HTTP, which prompted for login credentials. Using the credentials from flag 1, I gained access and found flag 2.

Flag 3
An aggressive port scan revealed that flag3.txt was hidden on the WinDC01 machine and accessible via anonymous FTP. I connected to the host IP (ftp 162.22.117.20) with the credentials anonymous:anonymous and downloaded flag3.txt.

Flag 4
Reviewing the port scan of the Windows10 host showed open ports, including port 25, which was vulnerable to the pop3/seatlelab_pass exploit. I found this exploit by searching for “slmail” in msfconsole. After configuring the options and gaining access, I obtained flag 4.

Flag 5
After gaining shell access to the host, I examined scheduled tasks where flag 5 was hidden. By adding options to filter the output (/TN for Task Name, /fo LIST for List Format, and /v for Verbose), I located the flag in the task comment.

Flag 6
Returning to the Meterpreter session, I loaded Kiwi and performed an lsa_dump_sam to find users and NTLM hashes. I discovered a user named "flag6." Using John, I cracked the hash and obtained the user's login credentials.

Flag 7
I searched the system for files containing “flag7” in their names and found flag7.txt in the C:\Users\Public\Documents directory, which contained flag 7.

Flag 8
Within the Meterpreter session, I ran kiwi_cmd lsadump::cache and found the user "ADMBob" along with his password hash in the MsCacheV2 format. Cracking this hash with John allowed lateral movement. I accessed another machine using the window/smb/psexec exploit through msfconsole. Executing net users on this host revealed flag 8.

Flag 9
Searching the system for files named “flag9” led to the discovery of flag9.txt located at the root of the C:\ drive, containing the flag.

Flag 10
By loading Kiwi on this machine and using dcsync_ntlm for the Administrator account, I obtained the NTLM hash for the user, which is the tenth and final flag.

Summary Vulnerability Overview

Vulnerability	Severity
Web App	
Reflected/Stored XSS Vulnerabilities across multiple web pages	Critical
Command Injection Vulnerabilities in a number of input fields	Critical
Open Source Data Exposure (Sensitive Data)	High
Local File Inclusion	Critical
SQL Injections	Critical
Weak Credentials (Weak to brute force attacks)	High
Linux System	
Open Source Data Exposure (Sensitive Data)	High
Weak Credentials (Weak to brute force attacks)	High
CVE-2017-5638 (Remote Code Execution)	Critical
CVE-2017-12615 (Apache Tomcat HTTP PUT method vulnerability)	Critical
CVE-2018-7600 (Remote code execution due to improper input validation)	Critical
CVE-2014-6271 (Shellshock)	Critical
CVE-2019-14287 (Security Bypass)	High
Windows System	
Open Source Data Exposure (Sensitive Data)	Critical
Anonymous FTP is enabled	Medium
CVE-2003-0264 (Buffer overflow in the SLMail POP3 server)	Critical
Lateral Movement	High

The following summary tables represent an overview of the assessment findings for this penetration test:

Scan Type	Total
Hosts	192.168.14.35 172.22.117.10 172.22.117.20 192.168.13.10 192.168.13.11 192.168.13.12 192.168.13.13 192.168.13.14
Ports	21 22 80 8080 443 110 445

Exploitation Risk	Total
Critical	10
High	6
Medium	1
Low	0

Vulnerability Findings

1.1
Critical Severity Vulnerability: Reflected/Stored XSS

Description	Injection of potentially malicious scripts through insecure input fields.
Impact	Potential for data theft
System	Web Application
Affected Host(s)	Welcome.php, Memory-Planner.php, Comments.php
References	
Remediations	Adding input validation or output encoding to ensure the input is rendered as text

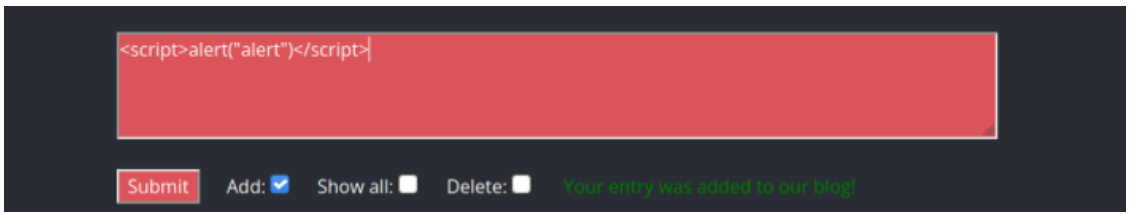
Proof Of Concept:
By inserting “<script>alert(“x”)</script>” within the input field on Welcome.php page, as shown in Exhibit 1.1, flag 1 can be found.This is a form of Reflected Cross-Site Scripting (XSS).

This can also be done on the Comments.php page as a form of Stored XSS as shown in Exhibit 1.2.

Exhibit 1.1

Inserting “<script>alert(“x”)</script>” within the input field on the Welcome.php web page

Exhibit 1.2



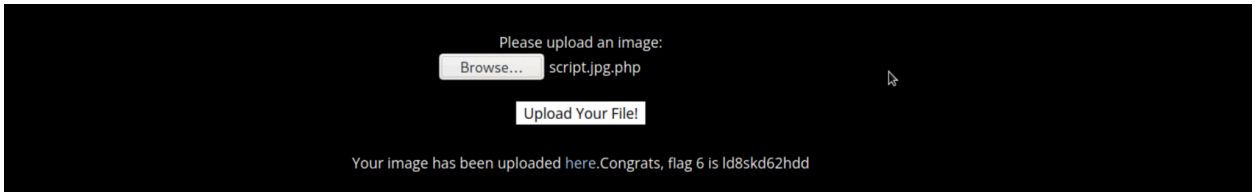
Inserting “<script>alert(“x”)</script>” within the input field on the Comments.php web page

1.2
Critical Severity Vulnerability: Local File Inclusion

Description	File uploads without proper validation or sanitization
Impact	Potential for malicious payloads to be uploaded and executed
System	Web Application
Affected Host(s)	Memory-Planner.php
References	
Remediation	Input validation and sanitization

Proof Of Concept:
Local File Inclusion can be used on the Memory-Planner.php page. By uploading a script.php file as below flag 5 is revealed.
Local File Inclusion can also be used to find flag 6. However, a form of input validation is used to check for the presence of ".jpg" within the uploaded file. This can be easily bypassed by naming the file script.jpg.php as shown in exhibit 1.3. Doing this will reveal flag 6.

Exhibit 1.3



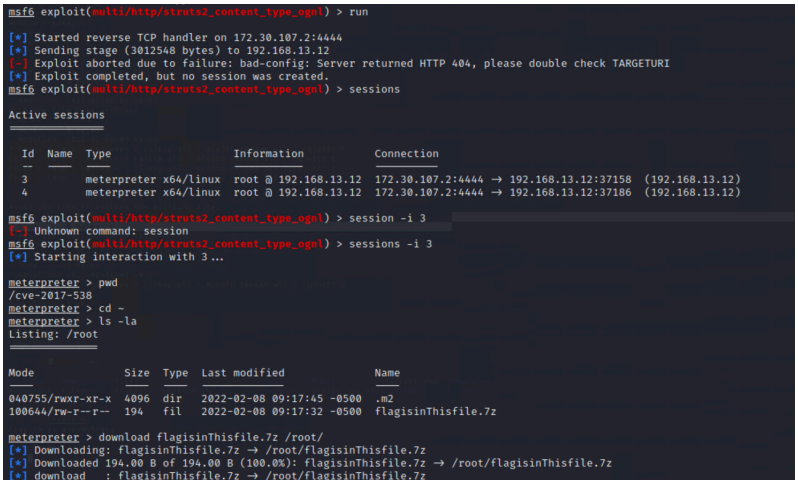
Uploading the potentially malicious payload 'script.jpg.php'

2.1
Critical Severity Vulnerability: CVE-2017-5638 (Remote Code Execution)

Description	Exploits Apache Struts 2 improper handling of OGNL expressions in Content-Type header for RCE.
Impact	Easily gain root shell access onto the host
System	Linux OS
Affected Host(s)	192.168.13.12
References	https://www.exploit-db.com/exploits/41614
Remediation	Upgrade to the latest version of Apache Struts 2 and apply security patches.

Proof Of Concept:
After completing a Nessus scan of 192.168.13.12 we can find one critical vulnerability. This vulnerability's plugin id is #97610. After using and setting the options for this exploit, as shown in Exhibit 2.1, we were able to create a session of which we could access to find and download the flag zip file. After unzipping and reading the contents we are able to find flag 10.

Exhibit 2.1



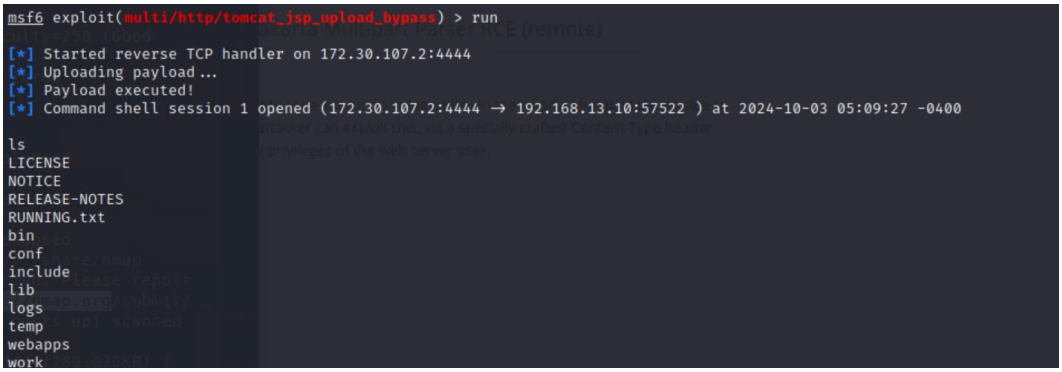
Using 'Jakarta' Multipart Parser OGNL Injection through Metasploit to gain shell access through Metasploit

2.2
Critical Severity Vulnerability: CVE-2017-12615 (Apache Tomcat HTTP PUT method vulnerability)

Description	Exploits Apache Tomcat HTTP PUT method misconfiguration to upload and execute JSP files for RCE.
Impact	Easily gain shell access into host system
System	Linux OS
Affected Host(s)	192.168.13.10
References	https://www.exploit-db.com/exploits/42953
Remediation	Disable HTTP PUT method or upgrade Tomcat to a version higher than 7.0.79

Proof Of Concept:
After running msfconsole and doing a search for exploits containing JSP we found ‘exploit/multi/http/tomcat_jsp_upload_bypass’ after setting the correct parameters we were able to successfully gain shell access and find and cat flag 7.

Exhibit 2.2



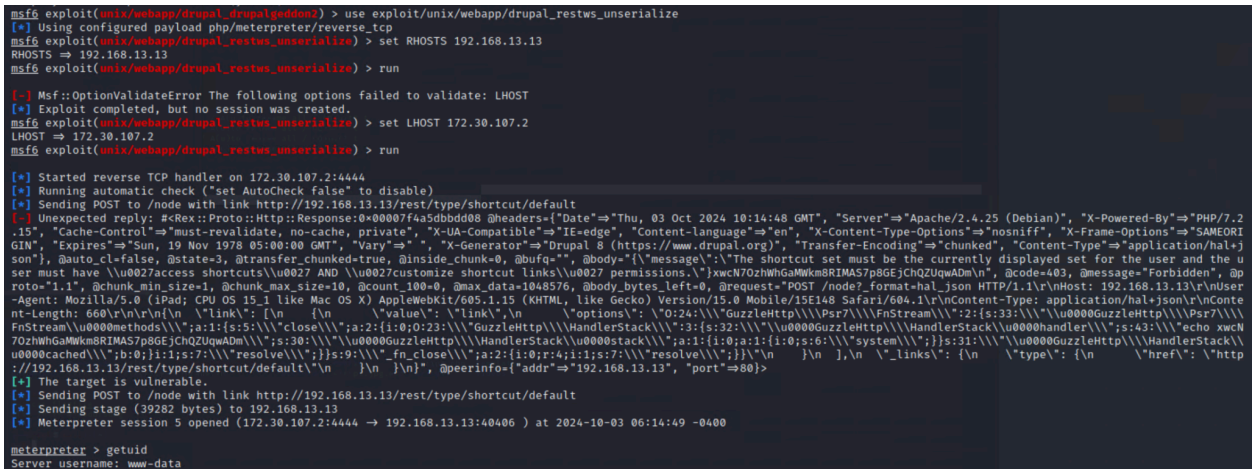
Executing Apache Tomcat HTTP PUT method vulnerability through Metasploit.

2.3
Critical Severity Vulnerability: CVE-2018-7600 (Remote code execution due to improper input validation)

Description	Exploits improper input validation in Drupal, leading to remote code execution (RCE).
Impact	Easily gain shell access into host system
System	Linux OS
Affected Host(s)	192.168.13.13
References	https://www.exploit-db.com/exploits/44482
Remediation	Update to the latest secure version of Drupal (7.58, 8.3.9, 8.4.6, or 8.5.1 or higher).

Proof Of Concept:
After searching for exploits for ‘drupal’ through the msfconsole, we were able to find an exploit. By setting the correct parameters and running the exploit we were able to start a meterpreter session within the host. From here we were able to get the username by ‘getuid’ as shown in exhibit 2.3.

Exhibit 2.3



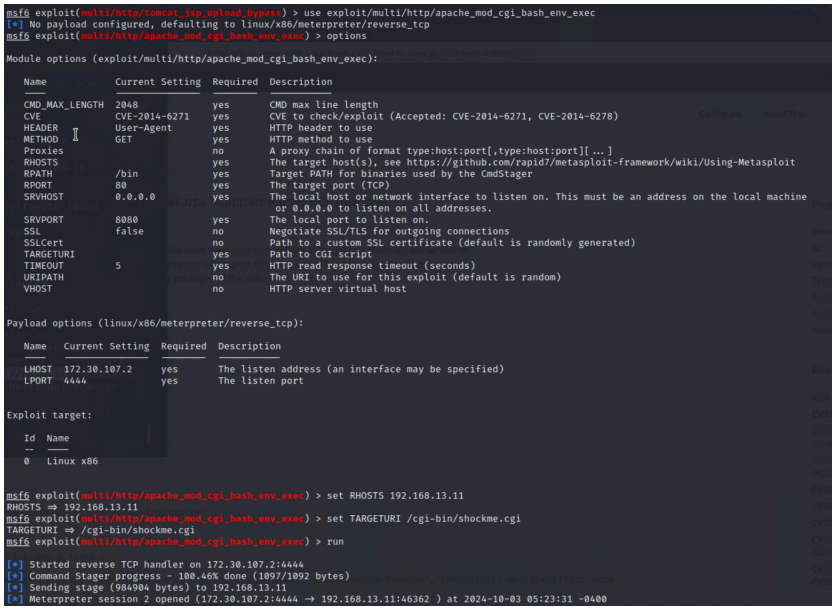
Execution of the drupal exploit through Metasploit

2.4
Critical Severity Vulnerability: CVE-2014-6271 (Shellshock)

Description	Exploits a Bash vulnerability (Shellshock) in Apache's CGI handling, allowing RCE.
Impact	Easily gain shell access into host system
System	Linux OS
Affected Host(s)	192.168.13.11
References	https://www.exploit-db.com/exploits/38849
Remediation	Update Bash to a patched version and restrict CGI script execution where unnecessary.

Proof Of Concept:
After trialling a number of exploits and searching for a ‘shocking’ vulnerability we were able to find the ‘shellshock’ vulnerability. After setting the needed parameters and running the exploit we were able to gain shell access into the host, as shown in Exhibit 2.4. From here we were able to view the sudoers file where flag 8 was hidden.

Exhibit 2.4



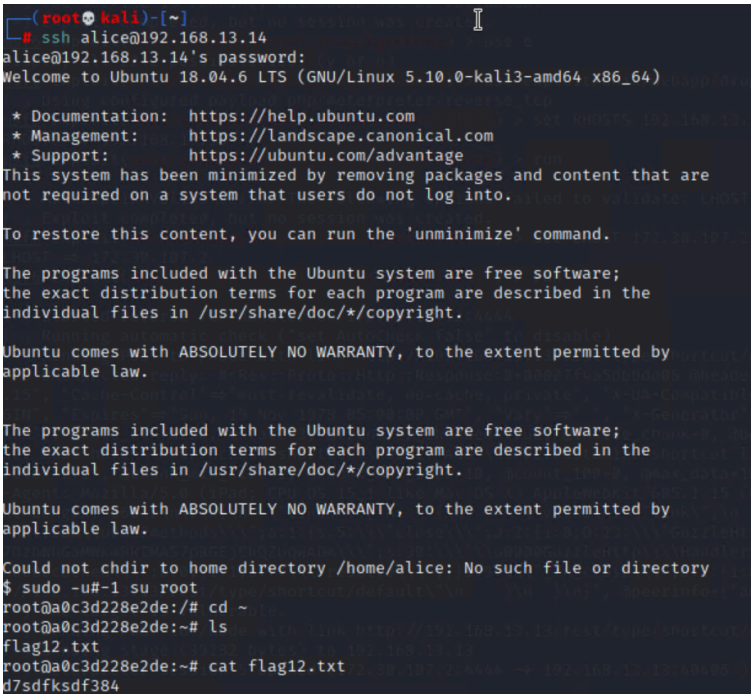
Execution of Shellshock vulnerability through Metasploit

2.5
High Severity Vulnerability: CVE-2019-14287 (Security Bypass)

Description	Allows privilege escalation on Linux systems via incorrect handling of sudo user privileges.
Impact	Any user can easily gain super user
System	Linux OS
Affected Host(s)	192.168.13.14
References	https://www.exploit-db.com/exploits/47502
Remediation	Update sudo to a patched version that addresses the vulnerability.

Proof Of Concept:
By being the whois data from flag 1 we are able to see ‘sshuser Alice’. By using SSH into the server (ssh alice@192.168.13.14) and guessing the correct password (‘alice’) we were able to gain shell access to the host. From here we were able to use a known vulnerability to escalate our privileges and gain access to the root directory where flag 12 can be found, as shown in Exhibit 2.5 below
.

Exhibit 2.5



Execution of privilege escalation vulnerability.

3.1
Medium Severity Vulnerability: Anonymous FTP is enabled

Description	Misconfigured FTP servers allowing anonymous:anonymous login, enabling access to files.
Impact	Unprotected files vulnerable to be stolen via anonymous login
System	Windows OS
Affected Host(s)	172.22.117.20
References	
Remediation	Disable anonymous FTP access, configure authentication, and restrict access to necessary directories.

Proof Of Concept:
By doing an aggressive port scan we can see the flag3.txt is hidden within the WinDC01 machine and is accessible via anonymous FTP access. Running an FTP for the host ip (ftp 162.22.117.20) and using the credentials anonymous:anonymous we are able to access the machine and download flag3.txt to our local machine as shown in Exhibit 3.1.

Exhibit 1.1

```
┌──(root💀kali)-[~]
└─# ftp 172.22.117.20
Connected to 172.22.117.20.
220-FileZilla Server version 0.9.41 beta
220-written by Tim Kosse (Tim.Kosse@gmx.de)
220 Please visit http://sourceforge.net/projects/filezilla/
Name (172.22.117.20:root): anonymous
331 Password required for anonymous
Password:
230 Logged on
Remote system type is UNIX.
ftp> ls
200 Port command successful
150 Opening data channel for directory list.
-r--r--r-- 1 ftp ftp            32 Feb 15  2022 flag3.txt
226 Transfer OK
ftp> get flag3.txt
local: flag3.txt remote: flag3.txt
200 Port command successful
150 Opening data channel for file transfer.
226 Transfer OK
32 bytes received in 0.00 secs (37.2024 kB/s)
ftp> exit
221 Goodbye

┌──(root💀kali)-[~]
└─# ls
Desktop    Downloads  file3      flagfile           hash.txt    Music     Public          script.php  Templates
Documents  file2      flag3.txt  flagisinThisfile.7z  LinEnum.sh  Pictures  script.jpg.php  Scripts     Videos

┌──(root💀kali)-[~]
└─# cat flag3.txt
89cb548970d44f348bb63622353ae278
```

Using anonymous:anonymous to gain ftp access.

3.2
Critical Severity Vulnerability: CVE-2003-0264 (Buffer overflow in the SLMail POP3 server)

Description	Buffer overflow in Seattle Lab Mail (SLMail) POP3 PASS command allows RCE
Impact	Easily gain shell access into host system
System	Windows OS
Affected Host(s)	178.22.117.20
References	https://www.exploit-db.com/exploits/16399
Remediation	Upgrade or discontinue use of SLMail, or apply a workaround by disabling the vulnerable POP3 service.

Proof Of Concept:
By looking at our port scan of the Windows10 host we are able to see the open and potentially vulnerable ports. This includes port 25 which after some research is vulnerable to the pop3/seatlelab_pass exploit. This can be found by searching ‘slmail’ on the msfconsole. After setting the options and gaining access to the host we are able to obtain flag 4. This can be seen in Exhibit 3.2 below.

Exhibit 3.2

```
root@kali: ~
File  Actions  Edit  View  Help
msf6 exploit(windows/pop3/seattlelab_pass) > set RHOSTS 172.22.117.20
RHOSTS => 172.22.117.20
msf6 exploit(windows/pop3/seattlelab_pass) > set payload windows/meterpreter/bind_tcp
payload => windows/meterpreter/bind_tcp
msf6 exploit(windows/pop3/seattlelab_pass) > run

[*] 172.22.117.20:110 - Trying Windows NT/2000/XP/2003 (SLMail 5.5) using jmp esp at 5f4a358f
[*] Started bind TCP handler against 172.22.117.20:4444
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Meterpreter session 2 opened (172.22.117.100:44501 → 172.22.117.20:4444 ) at 2024-10-03 20:14:36 -0400

meterpreter > ls
Listing: C:\Program Files (x86)\SLmail\System
=============================================

Mode              Size    Type  Last modified              Name
----              ----    ----  -------------              ----
100666/rw-rw-rw-  32      fil   2022-03-21 11:59:51 -0400  flag4.txt
100666/rw-rw-rw-  3358    fil   2002-11-19 13:40:14 -0500  listrcrd.txt
100666/rw-rw-rw-  1840    fil   2022-03-17 11:22:48 -0400  maillog.000
100666/rw-rw-rw-  3793    fil   2022-03-21 11:56:50 -0400  maillog.001
100666/rw-rw-rw-  4371    fil   2022-04-05 12:49:54 -0400  maillog.002
100666/rw-rw-rw-  1940    fil   2022-04-07 10:06:59 -0400  maillog.003
100666/rw-rw-rw-  1991    fil   2022-04-12 20:36:05 -0400  maillog.004
100666/rw-rw-rw-  2210    fil   2022-04-16 20:47:12 -0400  maillog.005
100666/rw-rw-rw-  2831    fil   2022-06-22 23:30:54 -0400  maillog.006
100666/rw-rw-rw-  1991    fil   2022-07-13 12:08:13 -0400  maillog.007
100666/rw-rw-rw-  2366    fil   2024-10-01 00:06:24 -0400  maillog.008
100666/rw-rw-rw-  2366    fil   2024-10-01 04:23:36 -0400  maillog.009
100666/rw-rw-rw-  2147    fil   2024-10-03 00:24:23 -0400  maillog.00a
100666/rw-rw-rw-  2366    fil   2024-10-03 04:13:30 -0400  maillog.00b
100666/rw-rw-rw-  31862   fil   2024-10-03 20:14:34 -0400  maillog.txt
100666/rw-rw-rw-  418748  fil   2024-10-03 19:58:44 -0400  task.txt

meterpreter > cat flag4.txt
822e3434a10440ad9cc086197819b49dmeterpreter > cat flag4.txt
```

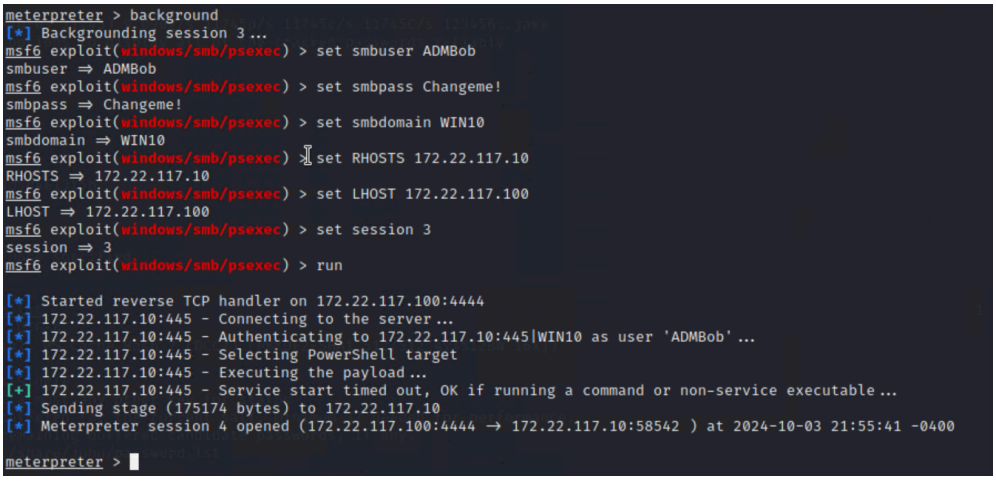
Execution of SLMail exploit through Metasploit

3.3
High Severity Vulnerability: Lateral Movement

Description	Moving laterally across systems or networks after initial compromise, exploiting trust or weak credentials.
Impact	Being able to gain more and more access once infiltrated into the system
System	Windows OS
Affected Host(s)	172.22.117.10
References	
Remediation	Implement network segmentation, strong authentication policies, and monitoring to detect lateral movement.

Proof Of Concept:
Within a meterpreter session on 172.22.117.20, we are also able to run a kiwi_cmd lsadump::cache where we are able to find the user ADMBob and his password hash in the MsCacheV2 format. Using John we are able to crack this hash and get the password. Lateral movement is now possible, being able to access the other machine with the window/smb/psexec exploit through the msfconsole. This can be seen in Exhibit 3.3 below. Entering a shell session on this host and running 'net users' to see the list user gives us flag 8.

Exhibit 3.3



Execution of lateral movement through Metasploit