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

Contact Information

Company Name	Looney Toons
Contact Name	Kabir Athwal
Contact Title	Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	11/14/2022	Kabir Athwal	First version

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.

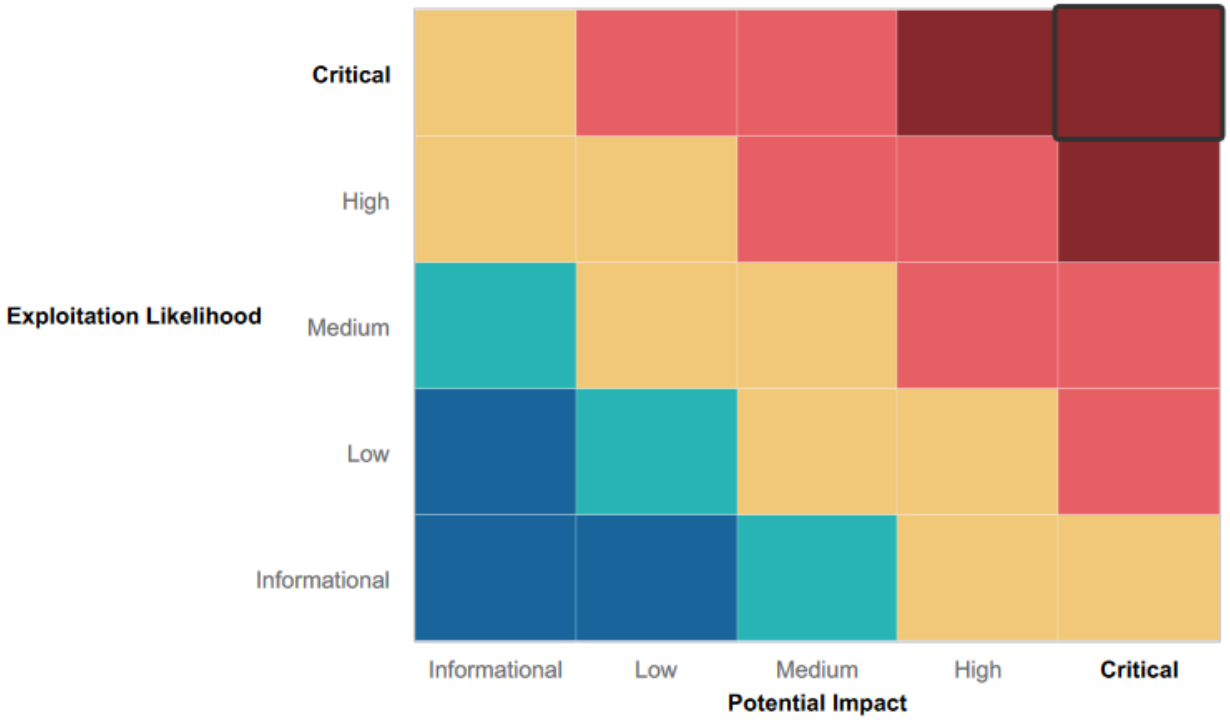
Executive Summary of Findings

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.

- There are no readily apparent strengths within Rekall's environment

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.

- No input validation on user inputs
- Sensitive data is exposed on certain pages
- Remote code execution is possible by malicious actors
- Shellshock vulnerabilities can be exploited by malicious actors
- Sensitive data is exposed on github site repository
- Port 110 (pop3) and 25 (tcp) are vulnerable, making a buffer overflow attack possible
- FTP port is insecure

Executive Summary

We split this penetration test into three different sections, testing the web application, testing the Linux hosts, and testing the Windows hosts. We tested all three of these sections separately, but information found from one was used on another

On the web application, we started by conducting cross site scripting (XSS) attacks on user-input fields on the site. XSS attacks target user-input fields to inject code that will have an effect on only an individual's site (reflected XSS) or on anybody that visits the site (stored XSS) . We conducted both a reflected and stored XSS attack, and both were successful. We used the command <script>alert(“xss”)</script>, once on an input field that only affects the user, and once on an input field that can be seen by anybody. Both times, these XSS attacks were successful and sent a popup alert on the site. Next, we went to the login.php page and inspected it. Viewing the html code, we were able to see login credentials embedded in it. The login username was ‘dougquaid’, with the password being ‘kuato’. These login credentials successfully logged us into their site and gave us access to admin only networking tools. On this admin only networking tools site, we conducted a command injection attack on the DNS user-input field. We injected the command, www.example.com | cat vendors.txt, which then showed the vendors.txt file, indicating the attack was successful.

On the Linux hosts, we first searched publicly available information about our site, totalrekall.xyz. We found the site’s WHOIS record, IP address, and certificates associated with the site. Next, we ran a zenmap scan on the entire network, giving us info about ports, services, and hosts associated with the network. Zenmap is a security scanner GUI. We also ran a Nessus scan on one of the hosts found from the zenmap scan, 192.168.13.12. Nessus is a security vulnerability scanning tool. The scan report showed one critical vulnerability in Apache Struts, which means that the system could be vulnerable to a remote code execution (RCE) exploit. We then entered a terminal session and used metasploit to exploit the RCE vulnerability. We used the package exploit/multi/tomcat_jsp_upload_bypass, as this uses Tomcat exploits, a version of an RCE vulnerability. Using the host used in the Nessus scan, 192.168.13.12, we were able to run this exploit and gain access into the host via a meterpreter shell. From here we are able to freely parse the host’s terminal. We also conducted a Shellshock attack, which remotely takes control of a host and executes arbitrary code. Again in metasploit, we found a package with shellshock in it, and used exploits/multi/http/apache_mod_cgi_bash_env_exec. We set the targeturi to /cgi-bin/shockme.cgi, a script that will exploit the host. We set the host to 192.168.13.11 and ran it, which allowed us to go through the server’s files using the cat command, accessing sensitive data such as the /etc/passwd and /etc/sudoers files.

Finally, we conducted penetration testing on the Windows hosts. We first searched the Github site repository for TotalRekall, and found a file (xampp.users) that contained hashed login credentials. We used John the Ripper tool on the terminal to crack the hash, giving us a username ‘trivera’ with a password ‘Tanya4life’. Using these credentials, we then went to 172.22.117.20, the local windows host, and successfully logged in using the credentials. Next, we exploited FTP port vulnerabilities on the Windows hosts. We saw in a previous scan that 172.22.117.20 has FTP port 21 open, so we ran ftp on that host in the terminal. From here, we were able to download and read sensitive files pulled from the host. Finally, we exploited a remote buffer overflow vulnerability found from seeing SLMail is running on port 25 (TCP) and port 110 (pop3) was open. So, we used a metasploit session to exploit this vulnerability. We used the package exploit/windows/pop3/seattlelab_pass, as this contained SLMail, and set the host to 172.22.117.20, the host with the ports needed open. Running it granted us access to the host’s terminal session, allowing us to search through the host’s filesystem.

Summary Vulnerability Overview

Vulnerability	Severity
Reflected XSS vulnerability on website user-input field	Medium
Stored XSS vulnerability on website user-input field	Medium
Sensitive data exposure on website's login page	Critical
Command injection on website DNS user-input field	High
Open source exposed data	Informational
Zenmap scan results	Informational
Nessus scan results	Low
Remote code execution (RCE) vulnerability	Critical
Shellshock vulnerability	Critical
Open source exposed data	High
FTP vulnerability	Critical
Remote buffer overflow vulnerability	Critical

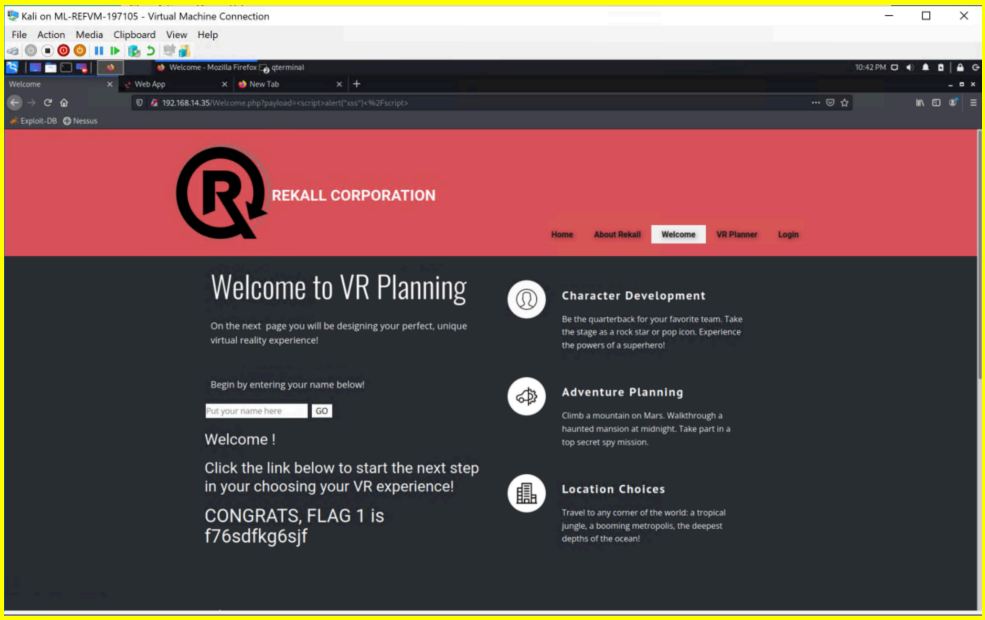
The following summary tables represent an overview of the assessment findings for this penetration test:

Scan Type	Total
Hosts	6
Ports	4

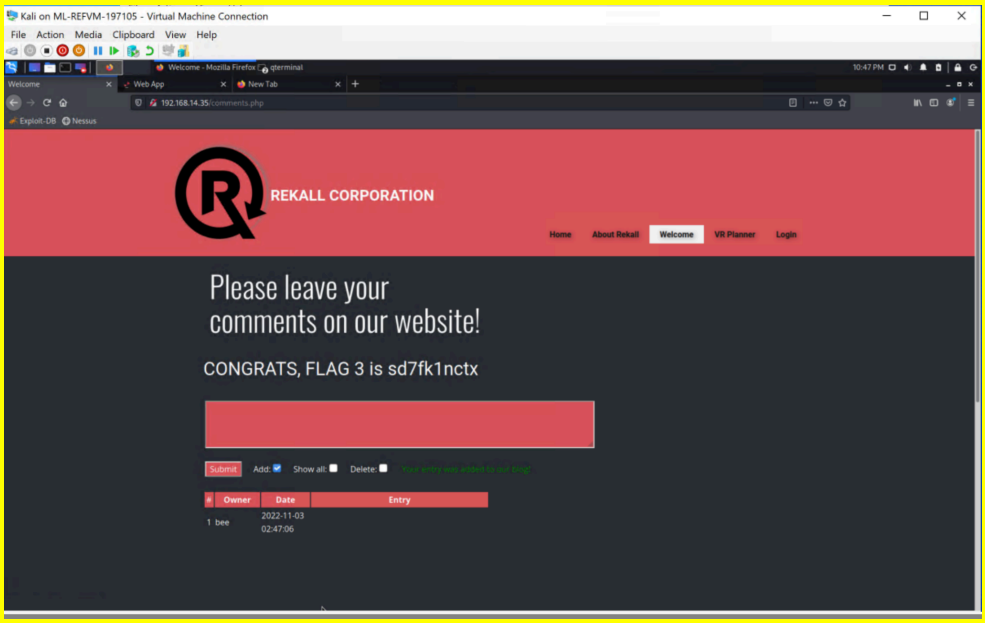
Exploitation Risk	Total
Critical	5
High	2
Medium	2
Low	1

Vulnerability Findings

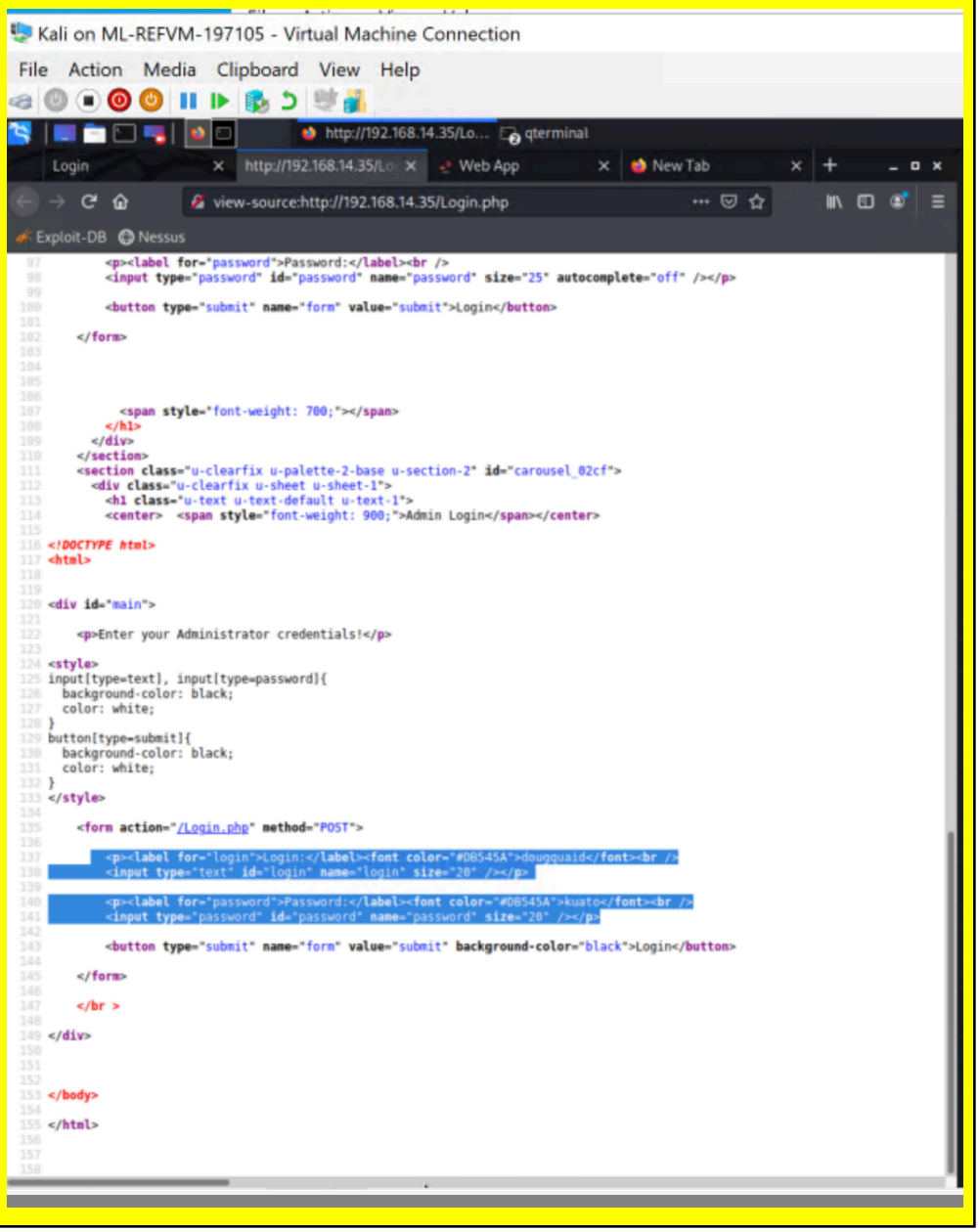
Vulnerability 1	Findings
Title	Reflected XSS vulnerability on website user-input field

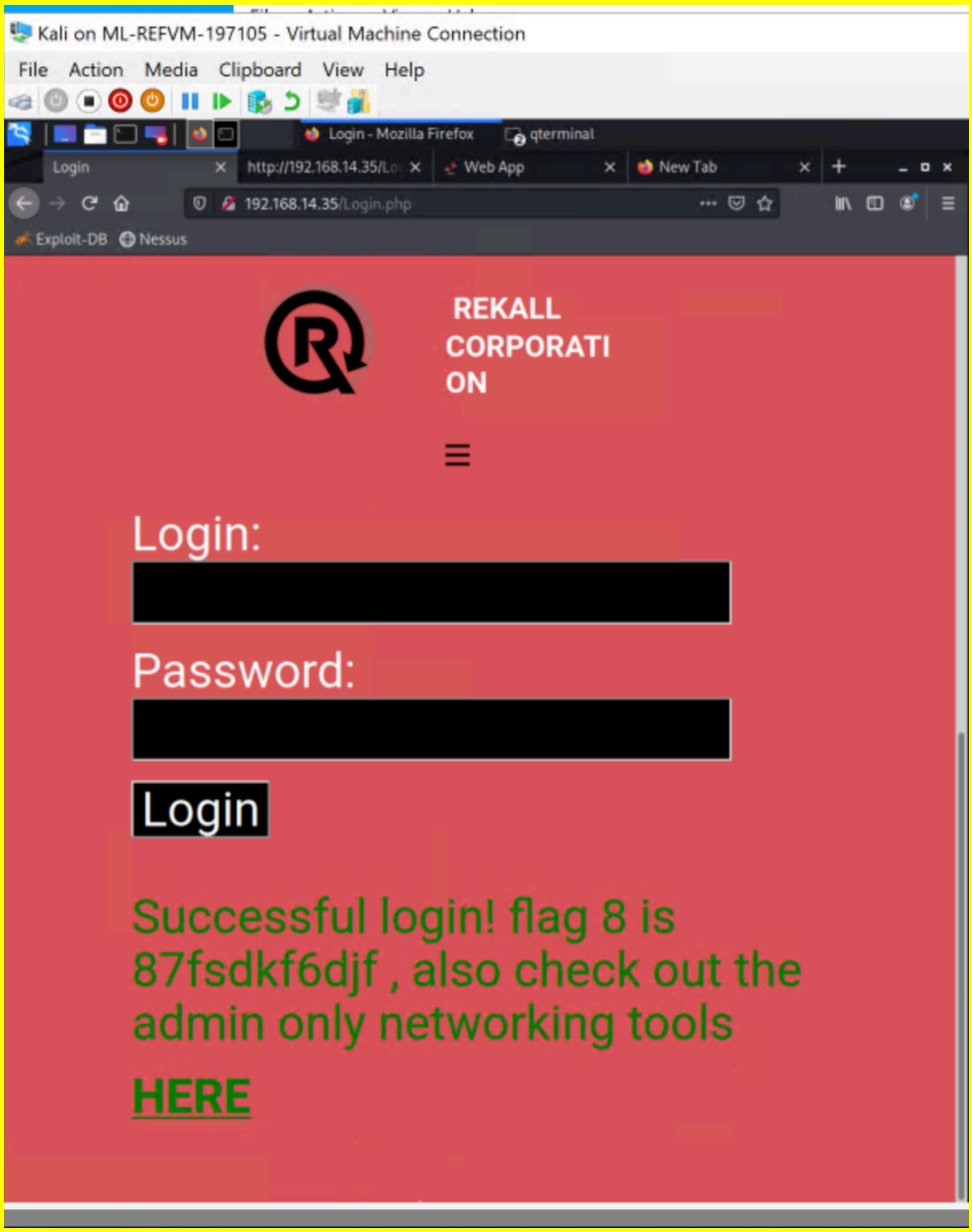
Type (Web app / Linux OS / WIndows OS)	Web app vulnerability
Risk Rating	Medium
Description	The site, Welcome.php is used to greet users to the Rekall Corporation. It contains a user-input field where you can enter your name into the site. This site is susceptible to a reflected XSS attack as there is no apparent input validation on the site, making a reflected XSS input validation attack possible. So, we were able to insert the command, <script>alert(“xss”)</script> , into the input field and it successfully returned an alert on the site.
Images	
Affected Hosts	192.168.14.35/Welcome.php
Remediation	• Set up input validation on the site, allowing only certain inputs to be submitted • Use more secure HTTP response headers, such as the Content-Type and X-Content-Type-Options headers

Vulnerability 2	Findings
Title	Stored XSS vulnerability on website user-input field
Type (Web app / Linux OS / WIndows OS)	Web app vulnerability
Risk Rating	Medium
Description	The site, Comments.php is used to allow users to submit comments about the website. It contains a user-input field where you can submit a comment on the website. This site is susceptible to a stored XSS attack as there is no apparent input validation on the site, making a stored XSS input validation attack possible. So, we were able to insert the same command as the previous

	vulnerability, <script>alert(“xss”)</script> , into the input field and it successfully returned an alert on the site.
Images	
Affected Hosts	192.168.14.35/Comments.php
Remediation	• Set up input validation on the site, allowing only certain inputs to be submitted • Use more secure HTTP response headers, such as the Content-Type and X-Content-Type-Options headers

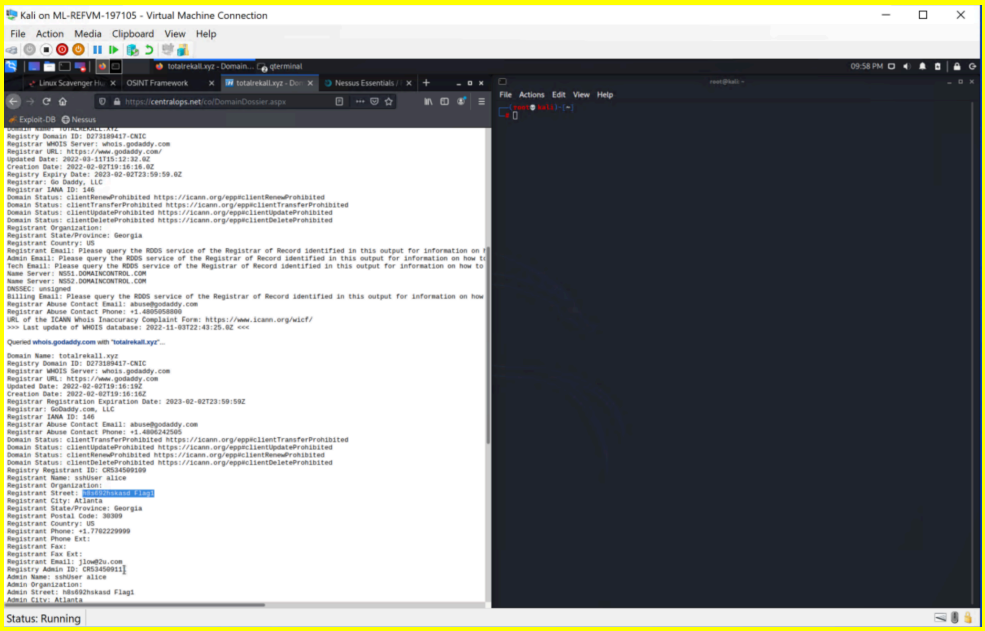
Vulnerability 3	Findings
Title	Sensitive data exposure on website’s login page
Type (Web app / Linux OS / WIndows OS)	Web app vulnerability
Risk Rating	Critical
Description	The site, Login.php is used to log in to your Rekall Corporation account. When inspecting it, we were able to find login credentials embedded in the web page’s code. The login username was ‘dougquaid’, with the password being ‘kuato’. These login credentials successfully logged us into their site and gave us access to admin only networking tools.

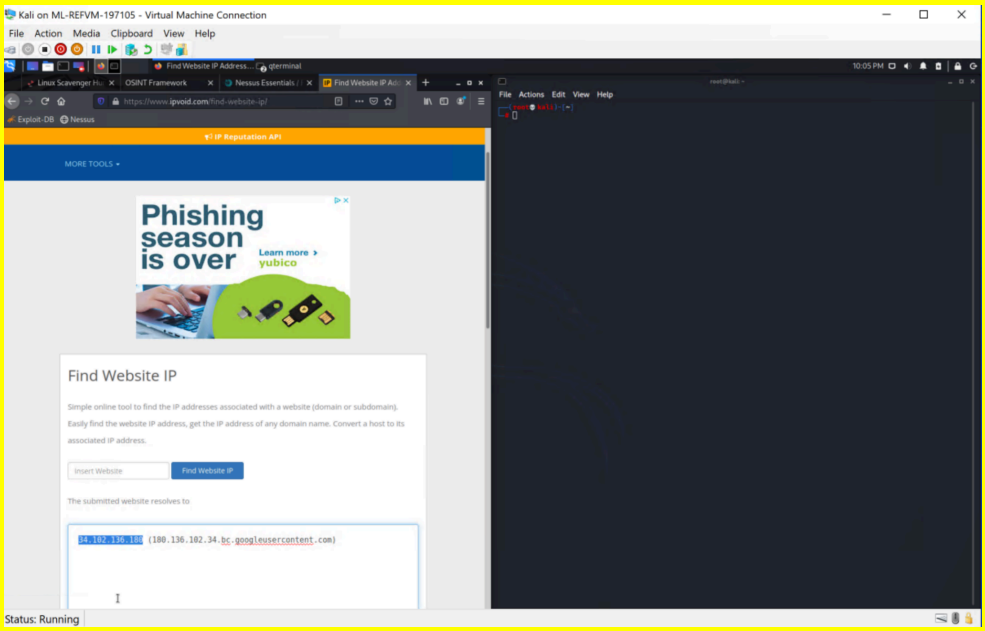
Images	

	
Affected Hosts	192.168.14.35/Login.php
Remediation	• Set up more secure web code, using encryption • Make it harder for a user to inspect element using code snippets that disable certain features when on the site

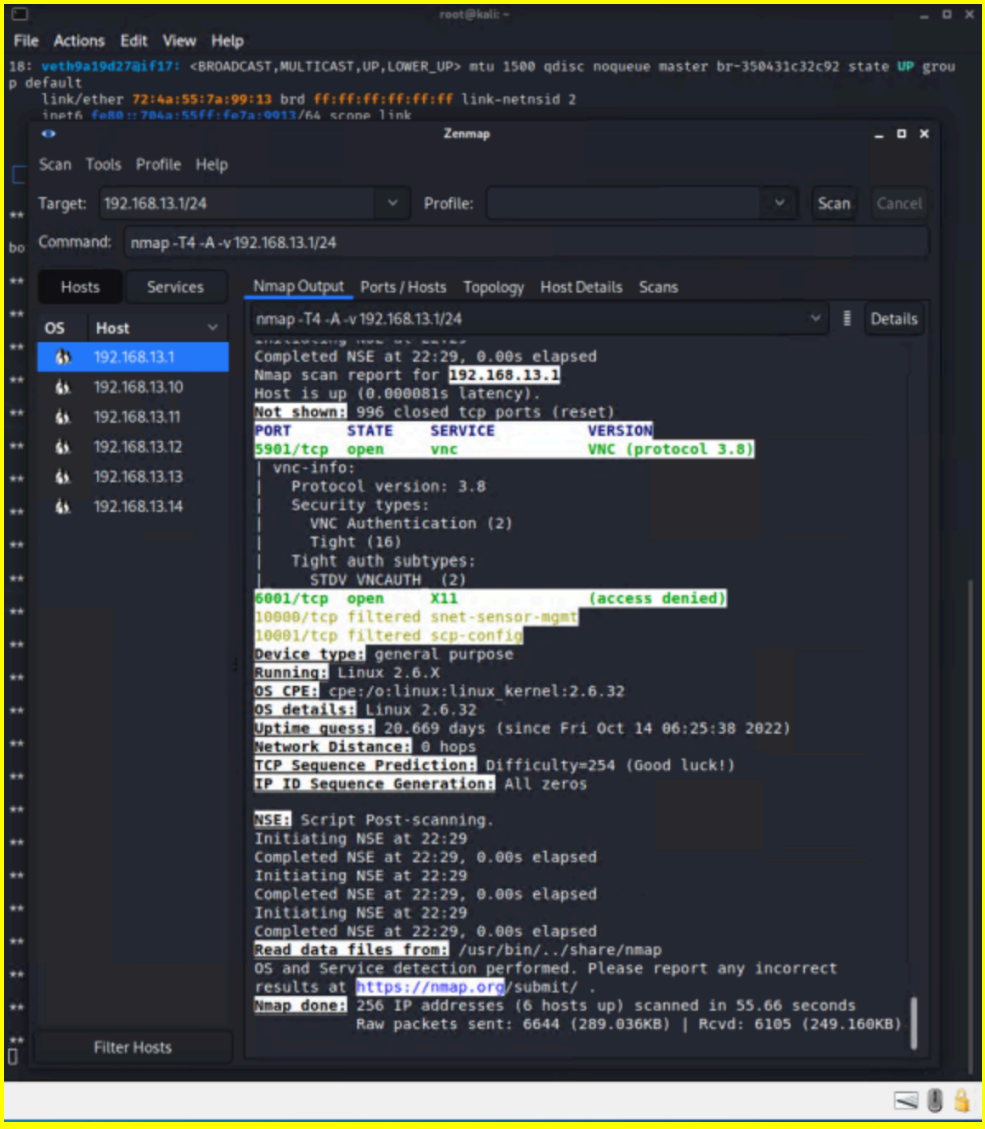
Vulnerability 4	Findings
Title	Command injection on website DNS user-input field
Type (Web app / Linux OS / WIndows OS)	Web app vulnerability
Risk Rating	High

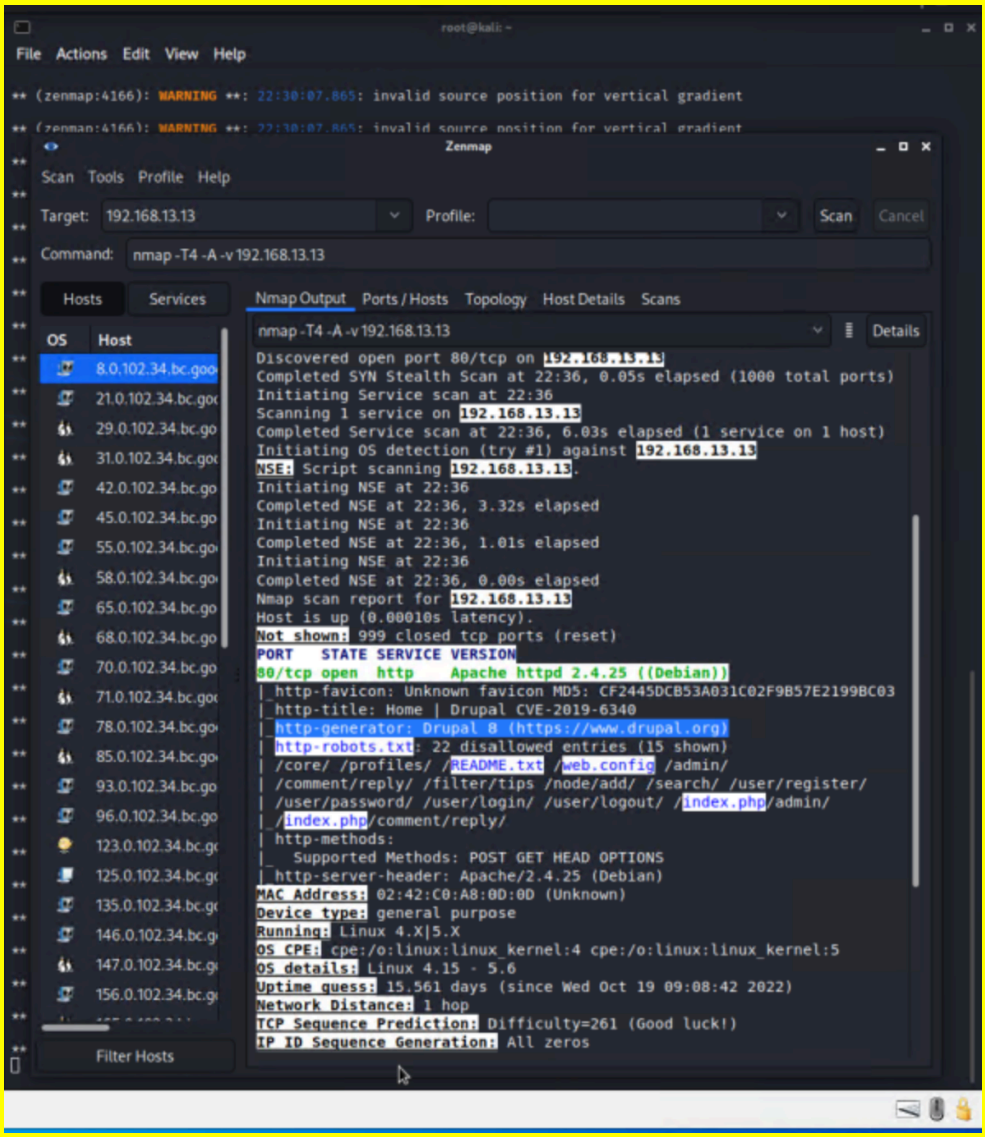
Description	Through the login credentials found in the previous vulnerability, we have access to the Networking.php site. This site has networking tools, available for use. It contains a user-input field for a DNS check. This site is susceptible to a command injection attack, as the text box reaches into its database for results. Inserting the command, www.example.com | cat vendors.txt , we were able to view the contents of the vendors.txt file.
Images	
Affected Hosts	192.168.14.35/Networking.php
Remediation	• Set up input validation on the site, allowing only certain inputs to be submitted • Use more secure HTTP response headers, such as the Content-Type and X-Content-Type-Options headers

Vulnerability 5	Findings
Title	Open source exposed data
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Informational
Description	The site, https://centralops.net/co/DomainDossier.aspx, found on the OSINT site (https://osintframework.com/), allows you to look up the WHOIS record of any domain. So, we searched totalrekall.xyz and found its domain WHOIS record. We also found the IP address of totalrekall.xyz, off an IP lookup website. crt.sh is a website that allows you to view the certificates of a domain. So, using that site we searched for totalrekall.xyz and found all certificates that they have.
Images	

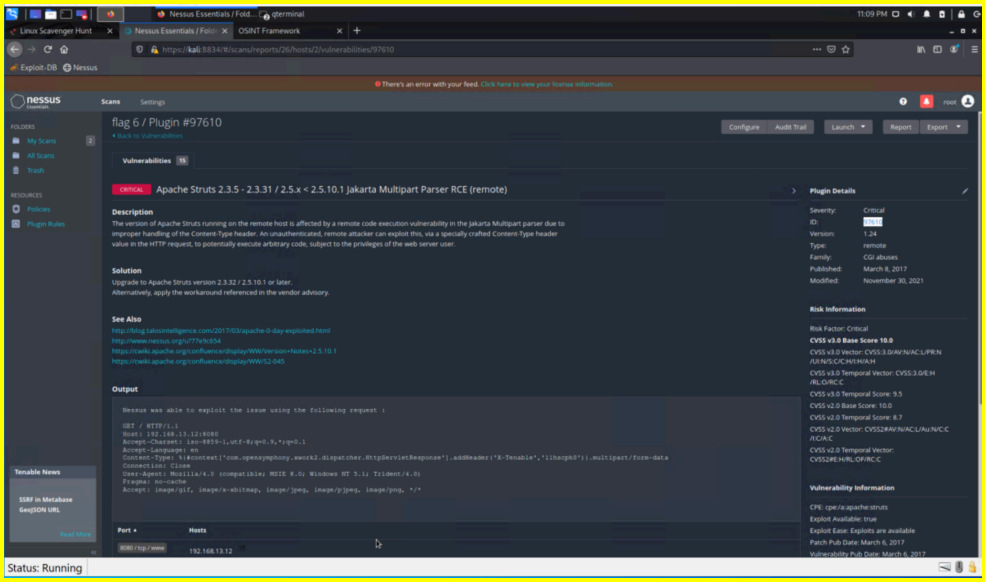


Affected Hosts	totalrekall.xyz
Remediation	• No real remediation, as all this info can be found publicly online

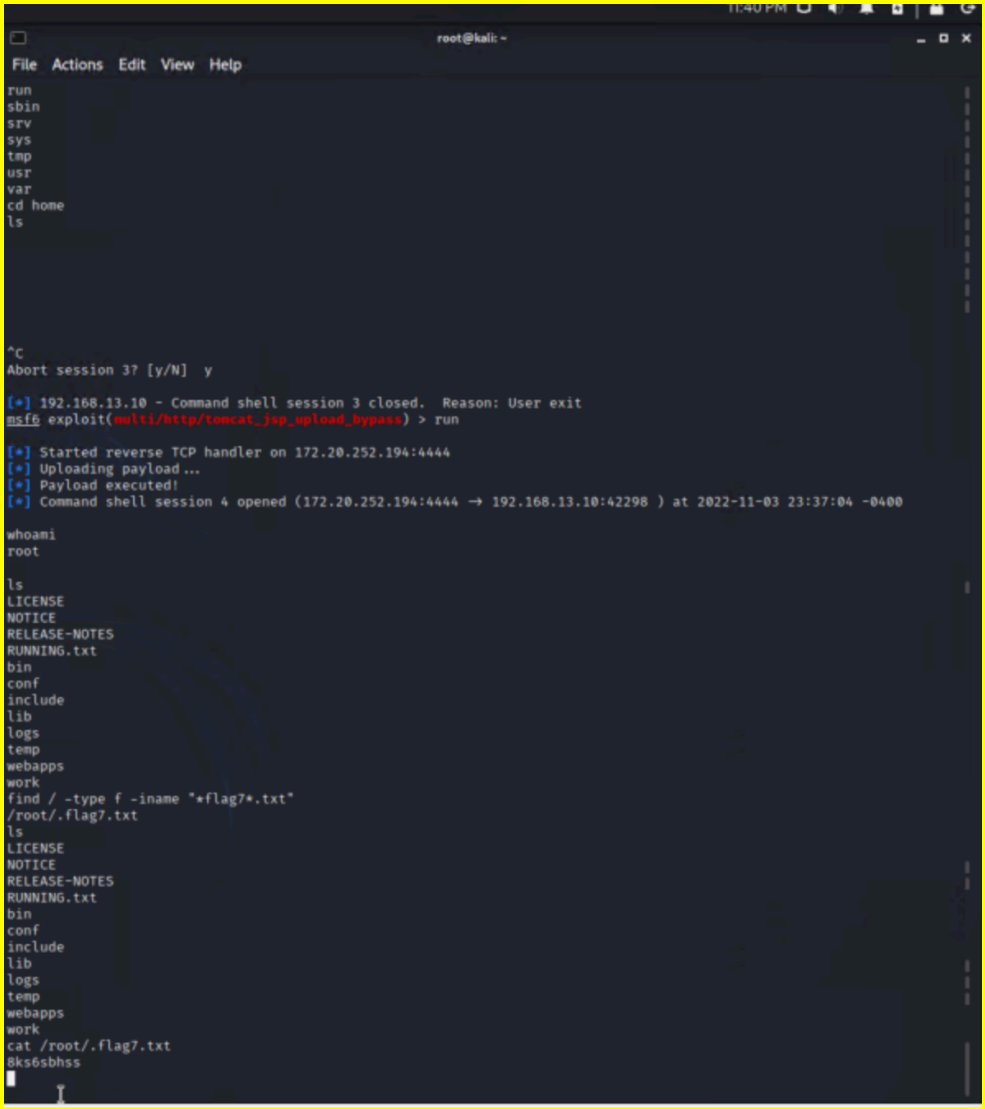
Vulnerability 6	Findings
Title	Zenmap scan results
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Informational
Description	Using zenmap, an nmap security scanner GUI, we were able to run a scan on the network, 192.168.13.1/24, finding that there are 6 total hosts running, 5 excluding the host we scanned from. The IPs we found are the following: 192.168.13.1, 192.168.13.10, 192.168.13.11, 192.168.13.12, 192.168.13.13, and 192.168.13.14. In search of a host that runs Drupal for later testing, we ran an aggressive zenmap scan on each. This allowed us to see open ports, services, and many other things about each host. Using this, we were able to find that the host 192.168.13.13 runs Drupal.
Images	

	
Affected Hosts	192.168.13.1, 192.168.13.10, 192.168.13.11, 192.168.13.12, 192.168.13.13, and 192.168.13.14
Remediation	• Set up a secure firewall to prevent zenmap scans • Spoof your IPs to hide them during a scan

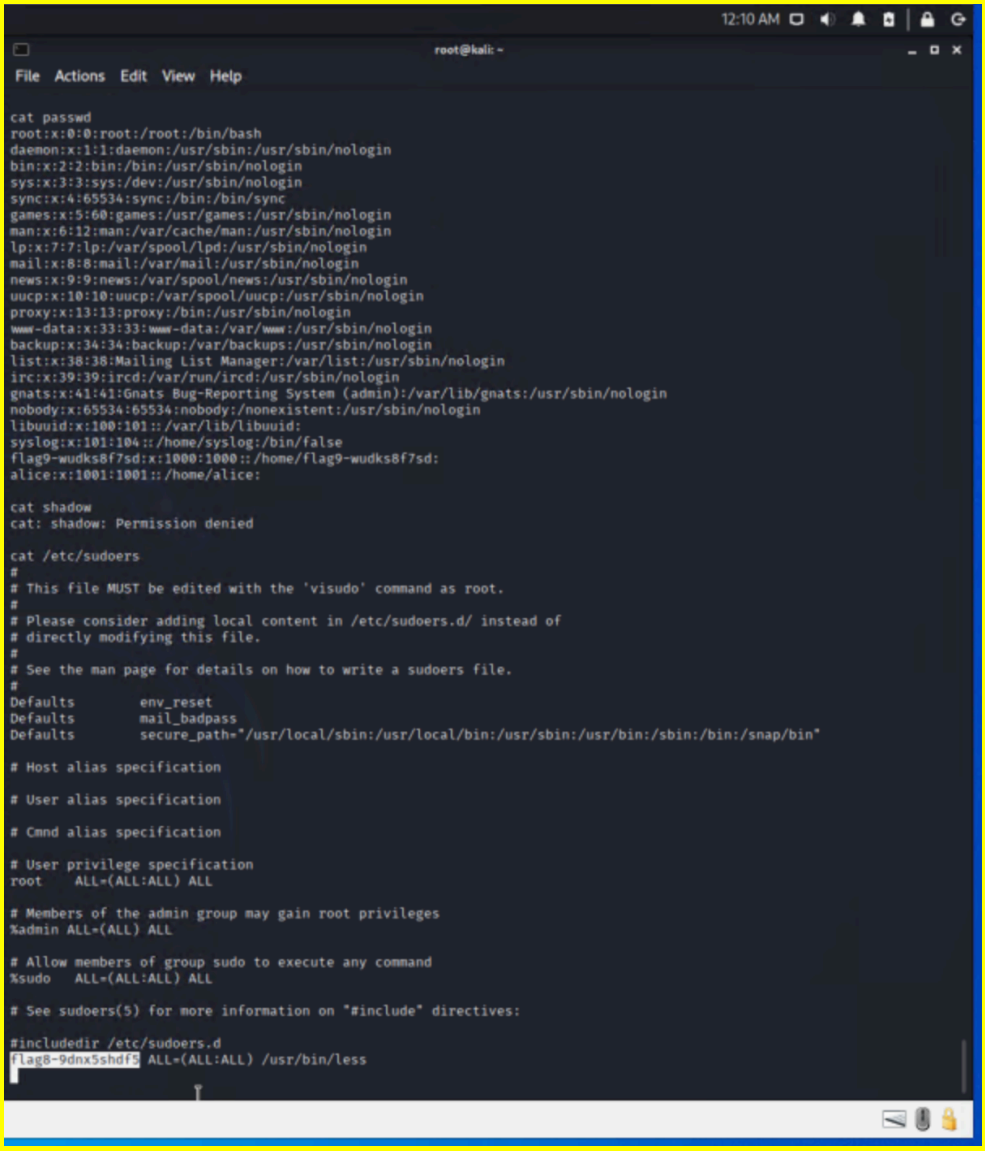
Vulnerability 7	Findings
Title	Nessus scan results
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Low
Description	Using nessus, a security vulnerability scanning tool, we ran an advanced nessus scan on the port 192.168.13.12. Opening up the scan report showed that there was one critical vulnerability found in Apache Struts. This

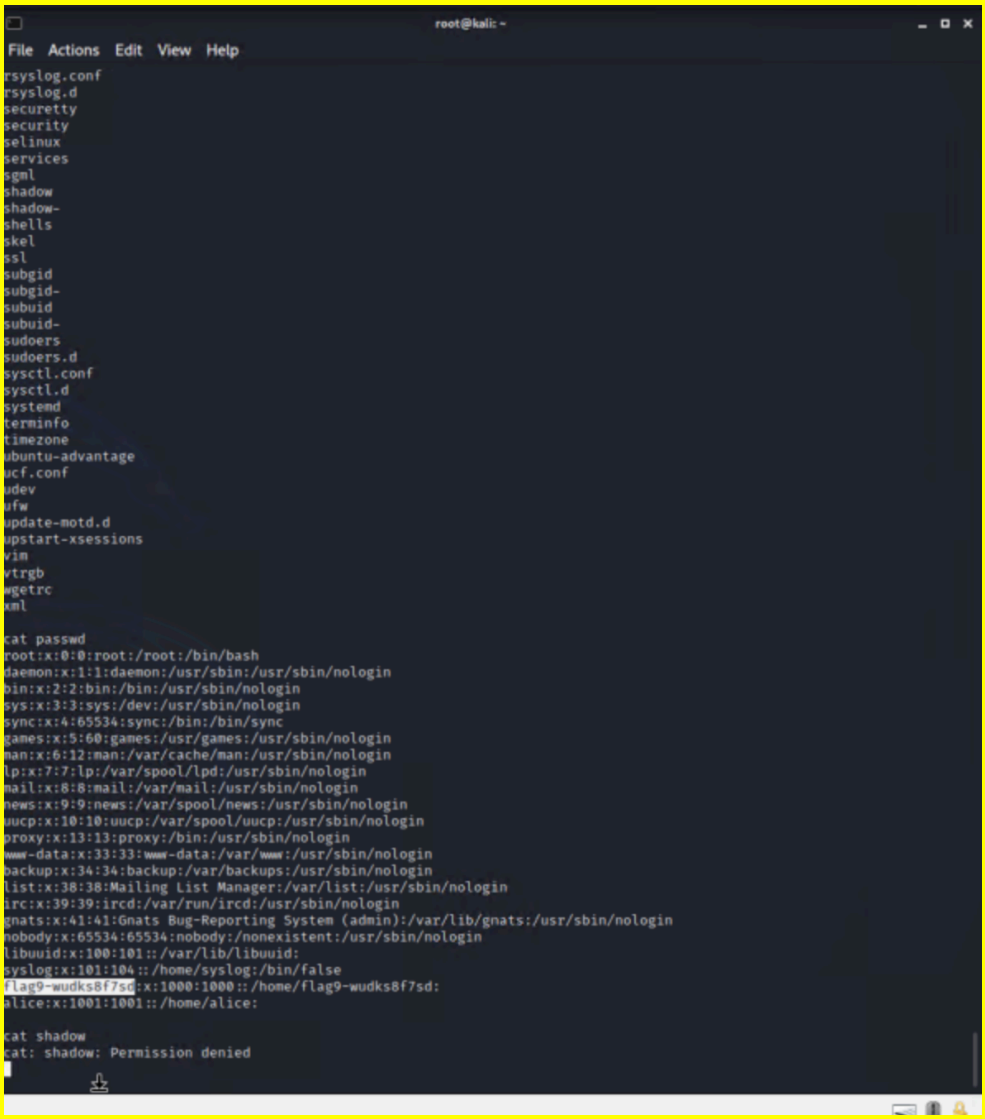
	vulnerability means that the system could be vulnerable to a remote code execution (RCE) exploit.
Images	
Affected Hosts	192.168.13.12
Remediation	• Set up a secure firewall to prevent nessus scans • Spoof your IPs to hide them during a scan

Vulnerability 8	Findings
Title	Remote code execution (RCE) vulnerability
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	In a terminal session, we entered the metasploit console. From here, we searched for an exploit with Tomcat. We found and used the exploit package exploit/multi/tomcat_jsp_upload_bypass. We set the RHOSTS to 192.168.13.10, one of the hosts on the network. We ran the exploit and was given access to the host through a meterpreter shell. We were then able to search through the host's terminal, searching for sensitive information/files.

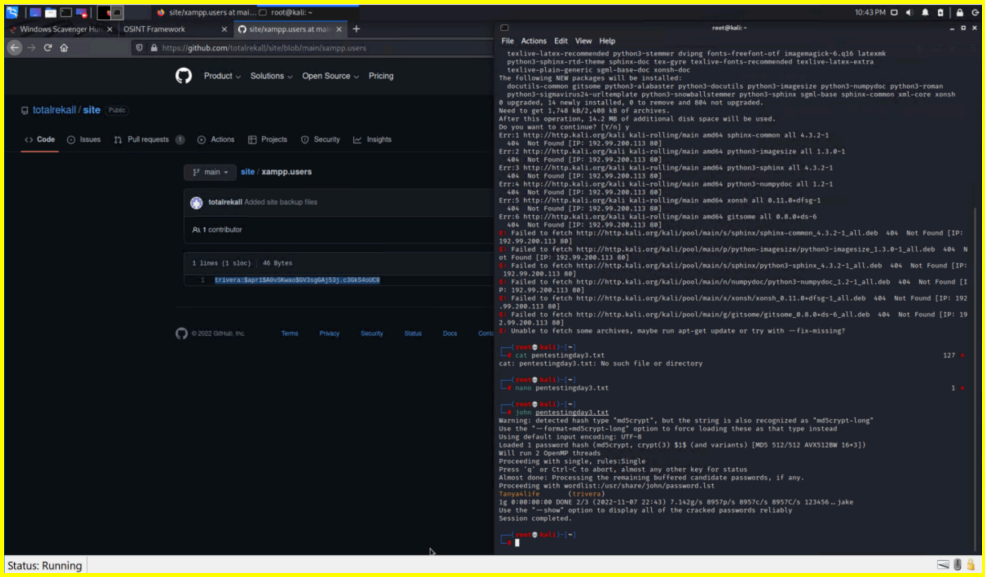
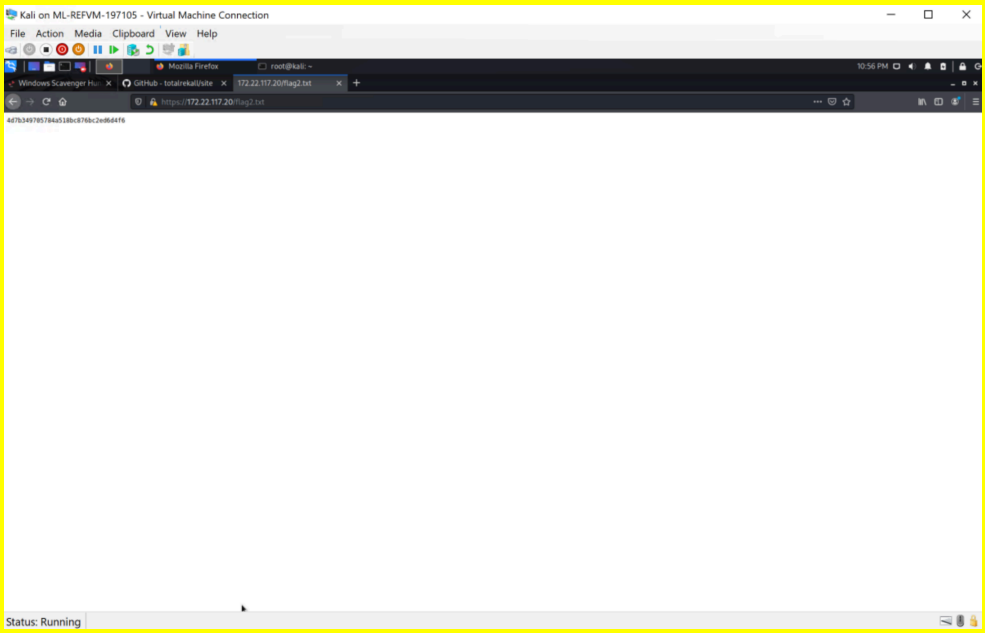
Images	
Affected Hosts	192.168.13.10
Remediation	• Sanitize user data frequently • Set up a secure firewall (WAF) to prevent RCE attacks

Vulnerability 9	Findings
Title	Shellshock vulnerability
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Continuing to use metasploit, we searched for exploits with shellshock. We found and used the exploit package exploits/multi/http/apache_mod_cgi_bash_env_exec. We set the TARGETURI

	to /cgi-bin/shockme.cgi. This is a shellshock script, which allows us to execute commands on the host terminal. We set RHOSTS to 192.168.13.11, the host IP, and ran it. This granted us access to the host through a meterpreter shell. From here, we were able to access and go through the server's files using the cat command, accessing sensitive data such as the /etc/passwd and /etc/sudoers files.
Images	

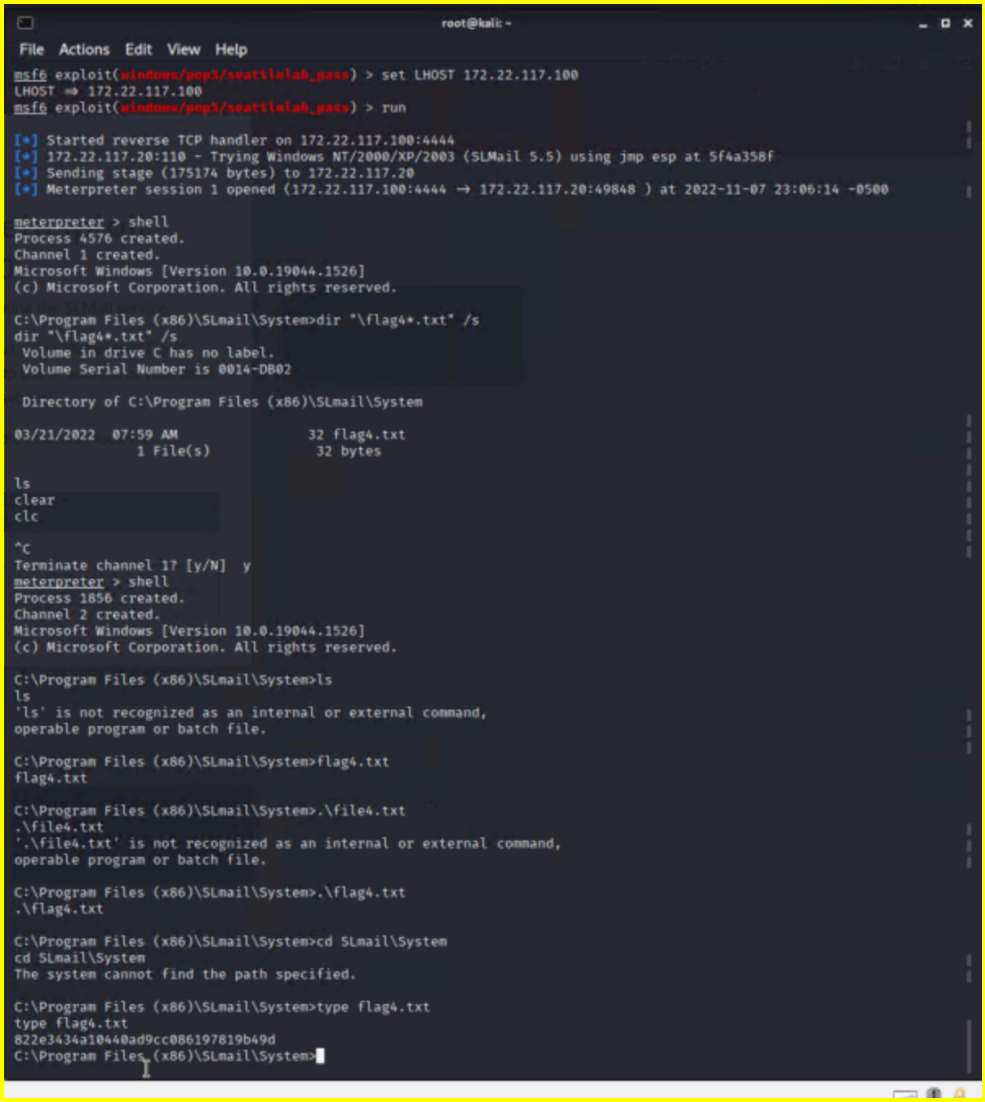
	
Affected Hosts	192.168.13.11
Remediation	• Set up a secure firewall (WAF) to prevent RCE attacks • Keep bash versions up to date • Monitor system logs to detect and alert proper personnel when anomalies are spotted

Vulnerability 10	Findings
Title	Open source exposed data
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	Searching Github, we were able to find TotalRekall's Github site repository. Searching through it, we found the xampp.users page, which contained login credentials of user 'trivera' with a hashed password. Using John the ripper tool

	on a terminal session, we cracked the password with it being ‘Tanya4life’. Using the aggressive zenmap scan results created previously, we found there was a WIN10 machine on 172.22.117.20 with port 80 open and a server machine on 172.22.117.10. Going to 172.22.117.20 on a web browser, we can enter the login credentials we found for trivera and successfully log in.
Images	 
Affected Hosts	172.22.117.20, github.com/totalrekall
Remediation	• Don’t keep site repositories publicly available on Github or on any other public website/repository

Vulnerability 11	Findings
Title	FTP vulnerability

Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	Using the aggressive zenmap scan created previously, we know that the machine on 172.22.117.20 has the FTP port 21 open. Having this port open allows anonymous ftp logins, making it susceptible to an FTP exploit. So, we ran ftp on 172.22.117.20 in a terminal, logging in as anonymous with no password. From this point, we had the ability to download and read files pulled from the machine on 172.22.117.20.
Images	
Affected Hosts	172.22.117.20
Remediation	• Use strong encryption and hashing to secure the FTP protocol • Set up a secure firewall that prevents FTP exploits • Require passwords with strong complexity and length • Keep backend databases on a separate server

Vulnerability 12	Findings
Title	Remote buffer overflow vulnerability
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	Using the aggressive zenmap scan created previously, we know that on host 172.22.117.20, SLMail is running on tcp (port 25) and pop3 (port 110). We entered a metasploit console on the terminal, and searched for an exploit package that uses SLMail. We found and used exploit/windows/pop3/seattlelab_pass. We set RHOSTS to 172.22.117.20 (the host that has SLMail running) and set LHOSTS to 172.22.117.100. Running it granted us access to the host through a meterpreter shell. From here, we have the ability to search through the host's filesystem.
Images	
Affected Hosts	172.22.117.20
Remediation	• Configure stacks as not executable so malicious actors can't use buffer

	overflow exploits ● Keep programming code clean and secure ● Randomize layouts of address space so a malicious actor struggles to coordinate the buffer overflow exploits