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	Hitch Cyber Solutions, LLC
Contact Name	Devin Hitch
Contact Title	Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	2/08/2024-2/21/2024	Devin Hitch	Web Application Vulnerability Testing
002	2/12/2023-2/21/2024	Devin Hitch	Linux Systems Vulnerability Testing
003	2/14/2023-2/21/2024	Devin Hitch	Windows Systems Vulnerability Testings

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.

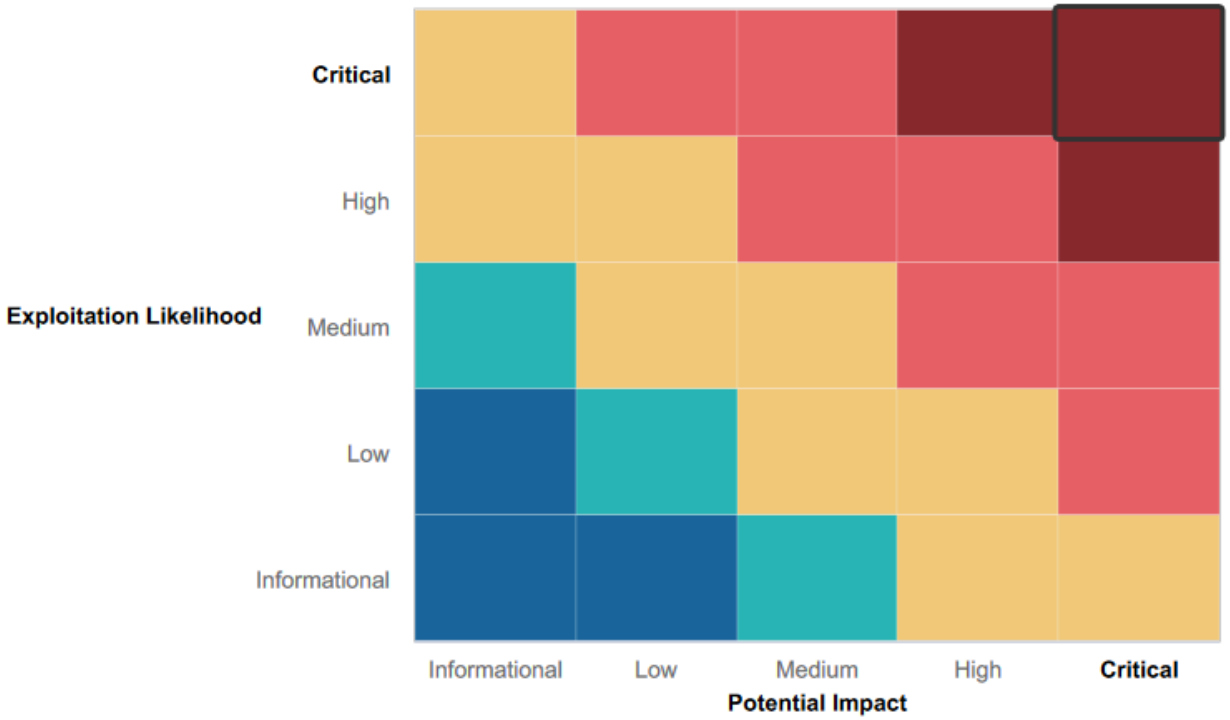
Executive Summary of Findings

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.

- Throughout the web application, there were several input validation methods that were able to protect against cross site scripting and injection methods. For example, the 2nd field for image upload is protected by only allowing .jpg extensions. The MX look-up tool has input validation to block '&' which prevents basic compound command injection attacks.
- All users require passwords to login
- All systems have some number of filtered ports (behind a firewall)
- Several metasploit modules failed to establish a connection.

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.

- The web applications vulnerabilities involve many command injection methods, and sanitization of user inputs would greatly improve the applications security.
- In addition, the website is vulnerable to XSS, and these issues directly effect customers using Rekall Web Application, and should be remedied immediately.
- There is a lot of information that is public facing that should not be. Company policies regarding what is private needs to be revisited.
- Vulnerable ports allowed remote access to Linux and Windows machines, and updated firewall policies should be implemented immediately. A deny all as default, followed by whitelisting necessary IPs and actions would be highly beneficial.
- Passwords used by both regular users and administration should be far more complex. Adding a character minimum of 12 to passwords would go a long way in preventing hashes from being cracked quickly.
- Passwords should be required to update at least every 6 months, as several exploits were carried out using cached user credentials.
- Defense in depth methods should be in place, as after initial access was gained, privileges were escalated on both linux and windows machines. As use of the windows10 credentials allowed access to administrator accounts, which in turn allowed access to the DC and active directory. In the hands of an attacker, the entirety of Rekall's confidentiality, integrity, and availability would very likely be compromised.

Executive Summary

During the penetration test, carried out between February 8th, 2024 through February 21st, 2024, I investigated the potential vulnerabilities on Rekall Corporation's web application, their Linux systems, as well as their windows systems. All testing was done from the MST time zone, therefore timestamps are recorded as GMT -7:00 throughout this report.

The wep application was rather vulnerable to various methods of attack. Cross site scripting (stored) was able to be implemented on the comments section, which is harmful to customers attempting to use the web application. Command injections were able to be carried out, and sensitive company data was exposed. PHP injection was possible through uploading images to the server, and better protection methods are needed, as requiring the presence of .jpg in media uploads can easily be worked around by attackers. It was also possible to navigate to web pages that were clearly not meant to be available, through the use of directory traversal tactics. Overall, the web application poses an immediate threat to the Rekall company and their customers. A WAF (Web Application Firewall) should strongly be considered.

The subnet containing the linux machines was enumerated with OSINT and scanning tools. OSINT gave basic information regarding certificates and DNS information, and the scans revealed several open ports which were used to gain access to the system.. Open port 8080 and 80 leave the systems vulnerable to HTTP (web) requests, which were used to gain C2 and execute code remotely. Port 22 is open for SSH, which is needed for the business, however the password strength needs to be improved to prevent unauthorized users from accessing the system. As the username and password of a specific user, alice, was found in OSINT, and the password was guessed. Several metasploit modules were loaded with information gained from scanning phases, and the systems were vulnerable to these methods which granted root access. After initial access was gained, further enumeration of the system revealed more sensitive data and credentials. The details of these actions can be found in the Vulnerability Findings section of this report. Overall, updating the firewall policies will be the most effective method in protecting the Linux machines, as the majority of initial access was gained through these open ports.

Open source intelligence revealed publicly available password hash and username. The windows machines contained open ports for FTP, SMB, TCP and UDP. These open ports were used to gain remote access to low privileged users. These exploits were found in metasploit. After initial access was gained, various exploits were used to enumerate system info. This information was used to escalate privileges to administrator and system. Through lateral movement and post initial access enumeration, administrator login credentials were obtained [username: administrator password: Changeme!]. In addition, credentials were cached and able to be accessed for privilege escalation and continued enumeration of the systems sensitive data. Eventually, using the administrator credentials, access to the DC was obtained. This allows for full enumeration of all users and passwords, and allows attackers to establish multiple avenues for persistent access. Proper logging methods would allow for security teams to locate and remote attackers from the system before they have a chance to escalate privileges or reveal sensitive company information.

After concluding this penetration test, the primary goals of Rekall's blue team should be, in order of priority:

1. Update firewall policies and implement Intrusion Detection and Prevention systems (IDS/IPS)
2. Implement logging for better detection of attacks, as well as understanding of how attacks were implemented. This will assist security teams in mitigating the damage caused by attackers.
3. Improve password policies company wide, forcing the use of symbols, numbers, capitals, lowercase, and minimum of at least 10 characters.

Summary Vulnerability Overview

Vulnerability	Severity
XSS reflected, Welcome.php, Field name: 'enter your name below" [Web App Flag 1]	High
XSS reflected, Memory planner.php, field name: "choose your character" [Web App Flag 2]	High
XSS stored, location comments.php, field name: "Please leave your comments on our website" [Web App Flag 3]	Critical
Sensitive data exposure, Location: About-rekall.php, [Web App Flag 4]	Medium
Local file inclusion, Location Memory-Planner.php, second field, upload picture [Web App Flag 5]	Critical
Local file inclusion, Location Memory-Planner.php, third field, upload picture [Web App Flag 6]	Critical
SQL injection, Location: login.php, first field [Web App Flag 7]	Critical
Sensitive data exposure, Location login.php, second field. [Web App Flag 8]	Critical
Sensitive data exposure, Location: robots.txt [Web App Flag 9]	Low
Command injection, Location: networking.php, first field [Web App Flag 10]	Critical
Command injection, Location: networking.php, second field [Web App Flag 11]	Critical
PHP injection, Location: souvenirs.php [Web App Flag 13]	Critical
Directory traversal, Location: Disclaimer.php [Web App Flag 15]	Critical
Open source exposed data, https://centralops.net/co/DomainDossier.aspx [Linux Flag 1]	Low
OSINT, txtlookup of totalrekall.xyz [Linux Flag 2]	Informational
Open source exposed data/certificate transparency [Linux Flag 3]	Informational
Network scan, nmap 192.168.13.0/24 [Linux Flag 4]	Informational
Network scan, aggressive nmap scan 192.168.13.0/24 [Linux Flag 5]	Informational
Nessus scan reveals critical vulnerability [Linux Flag 6]	Informational
Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617), Host 192.168.13.10 [Linux Flag 7]	Critical
Shellshock vulnerability, IP addr 192.168.13.11 [Linux Flags 8 & 9]	Critical
Struts - CVE-2017-5638 Location: 192.168.13.12 [Linux Flag 10]	Critical
Drupal - CVE-2019-6340 [Linux Flag 11]	Critical
CVE-2019-14287 & username as password [Linux Flag 12]	Critical
Password hashes publicly available [Windows Flag 1]	Critical
Port scan enumeration, open HTTP [Windows Flag 2]	Critical
FTP anonymous vulnerability [Windows Flag 3]	Critical
SLMail vulnerability [Windows Flag 4]	Critical
Scheduled tasks persistence vulnerability [Windows Flag 5]	Medium
Kiwi exploit & weak password [Windows Flag 6]	Critical
Post exploitation enumeration of sensitive data [Windows Flag 7]	High

Cached credentials & DCsync vulnerability [Windows Flag 8]	**Critical**
Post exploit enumeration [Windows Flag 9]	**Critical**
DC Administrator login found via DCsync vulnerability [Windows Flag 10]	**Critical**

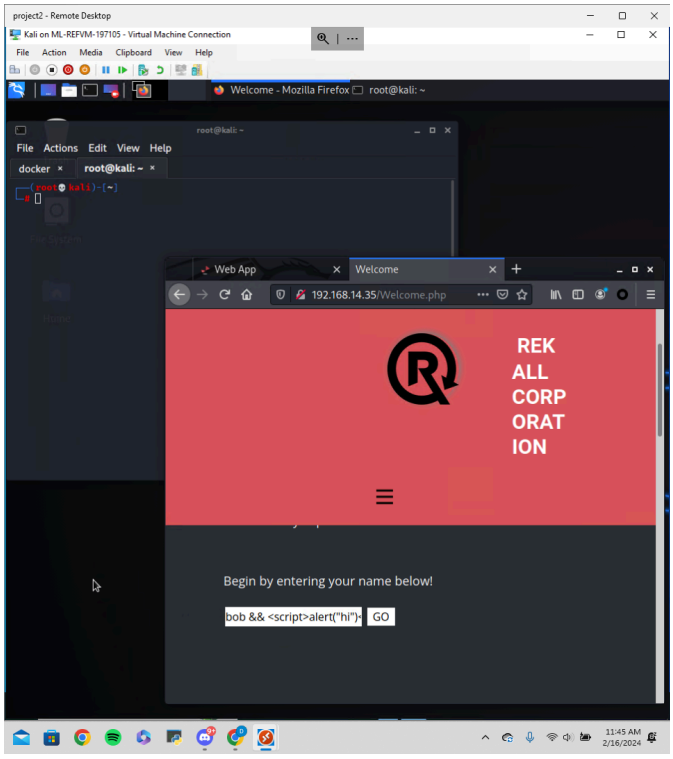
The following summary tables represent an overview of the assessment findings for this penetration test:

Scan Type	Total
Hosts	1 Web Application (192.168.14.35) 5 Linux Machines (192.168.13.10) (192.168.13.11) (192.168.13.12) (192.168.13.13) (192.168.13.14) 2 Windows Machines (172.22.117.20) (172.22.117.20) 8 Total Hosts
Ports	8080 HTTP 80 HTTP 22 SSH 110 POP3 21/TCP FTP 445 SMB 139/TCP 138/UDP 8 Total Ports

Exploitation Risk	Total
Critical	22
High	3
Medium	2
Low	2

Vulnerability Findings

Vulnerability 1	Findings
Title	XSS reflected, Location: Welcome.php

Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	On Rekall's web application, the input field to "enter your name" is vulnerable to a cross site scripting reflected attack. In this instance, a script can be run on the webpage using the input: <script>alert('Hi')</script> successfully creating a javascript popup on the web app. The script used in this example is rather harmless, however a malicious script could be uploaded just as easily, deeming this a critical risk.
Images	# Image 1 shows example of XSS payload  # Image 2 shows a successful XSS attack

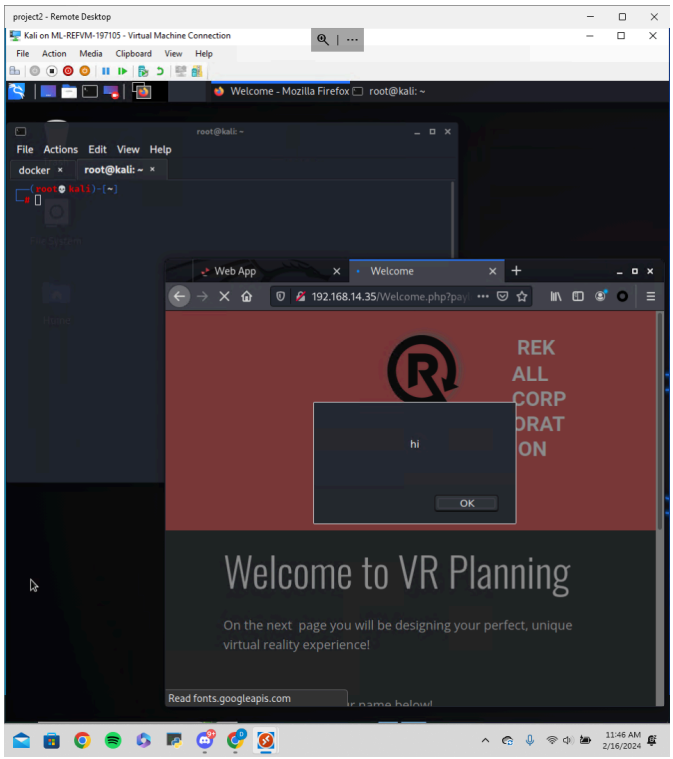
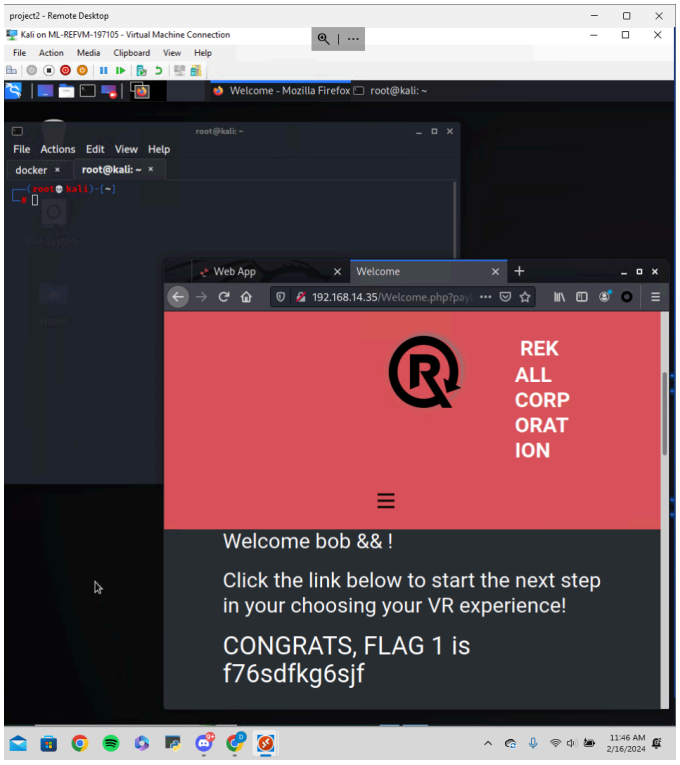
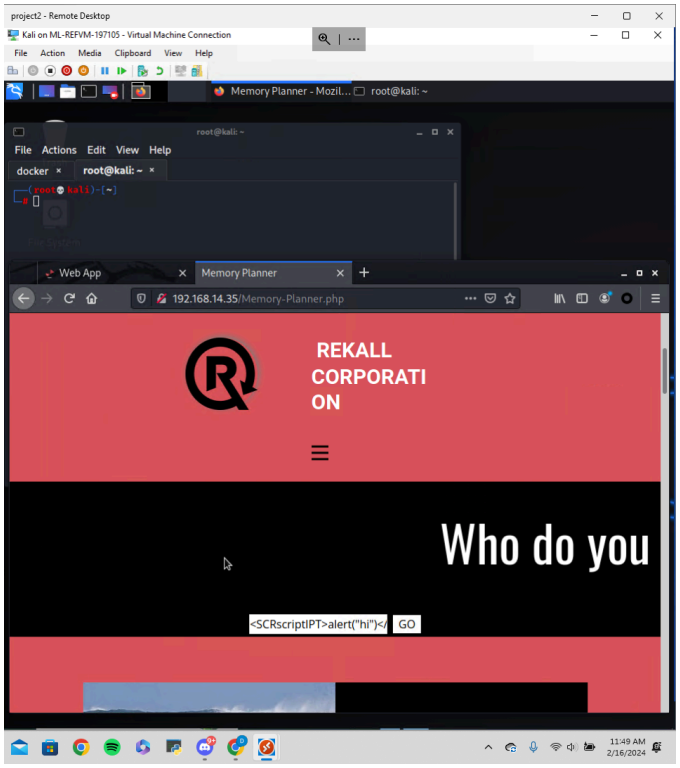


Image 3 shows the flag



Affected Hosts	192.168.14.35/Welcome.php
Remediation	Input validation is needed, a name only contains letters, spaces, and in some cases an apostrophe (‘). These are the only characters that should be allowed.

Vulnerability 2	Findings

Title	XSS reflected, Location: Memory planner.php
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	On Rekall's web application, the input field to "who do you want to play as" is vulnerable to a cross site scripting reflected attack. This field is more secure than the XSS reflected attack in Vulnerability 1, as there is some input validation preventing the use of the word 'script'. However, clever attackers can work around this, in this instance, surrounding one 'script' with another, ie <SCRscriptIPT> This work around causes the input validation in place to remove the script in the middle, leaving a <SCRIPT> behind which is then run on the web application. As there is more security in this field than the previously mentioned XSS reflected attack, the risk rating is set to high.
Images	# Image 1 shows example of XSS payload  # Image 2 shows a successful XSS attack

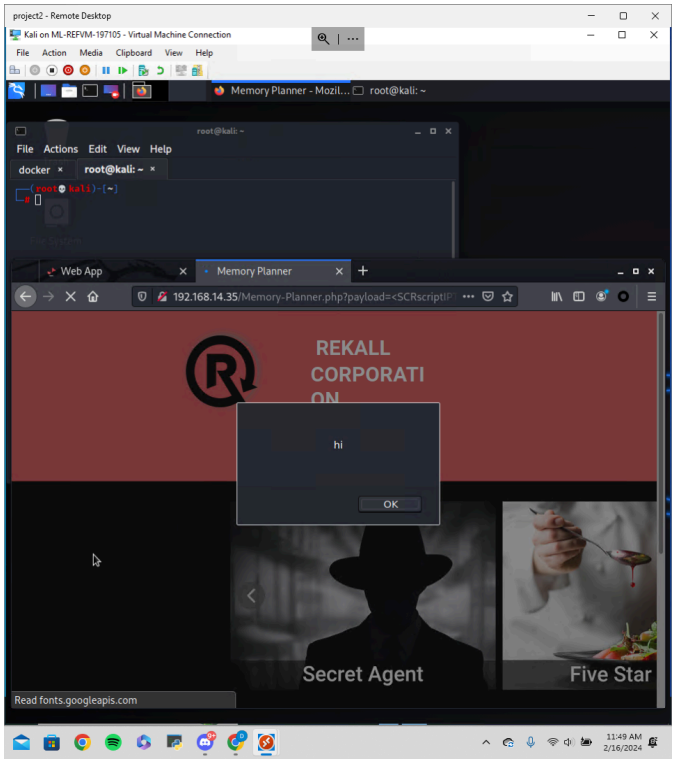
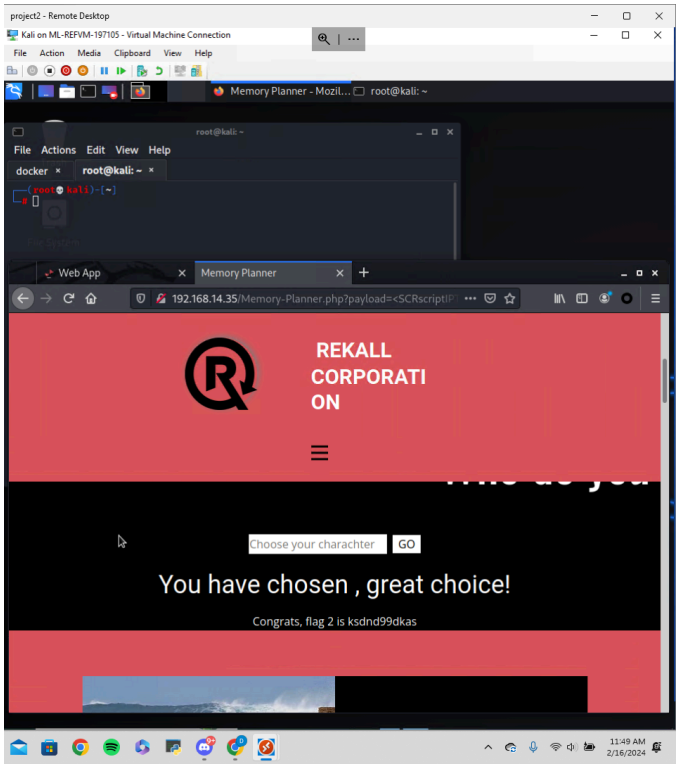
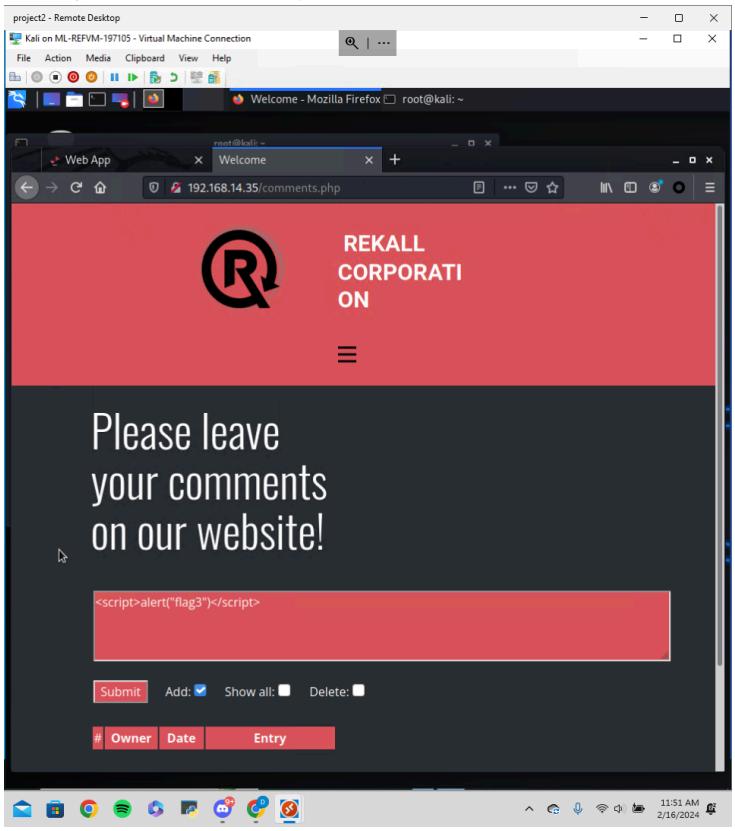
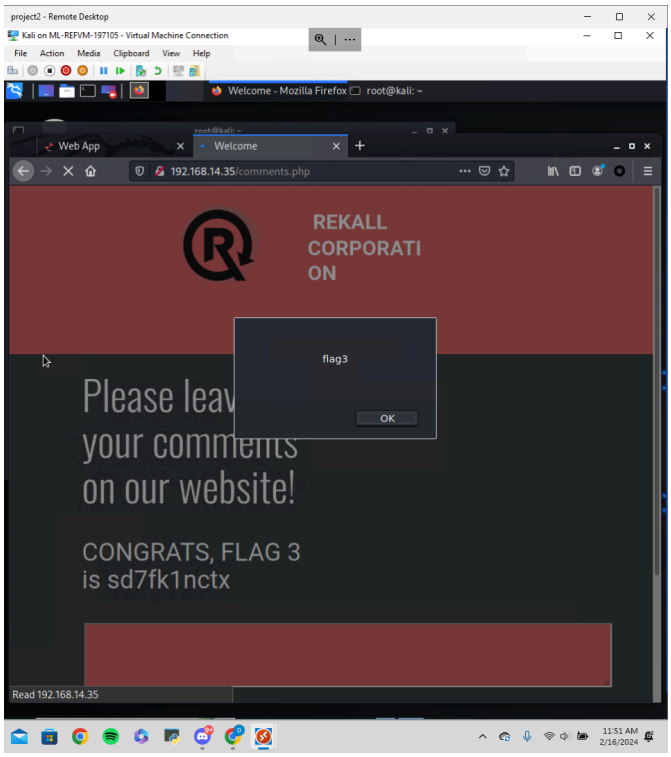


Image 3 shows the flag

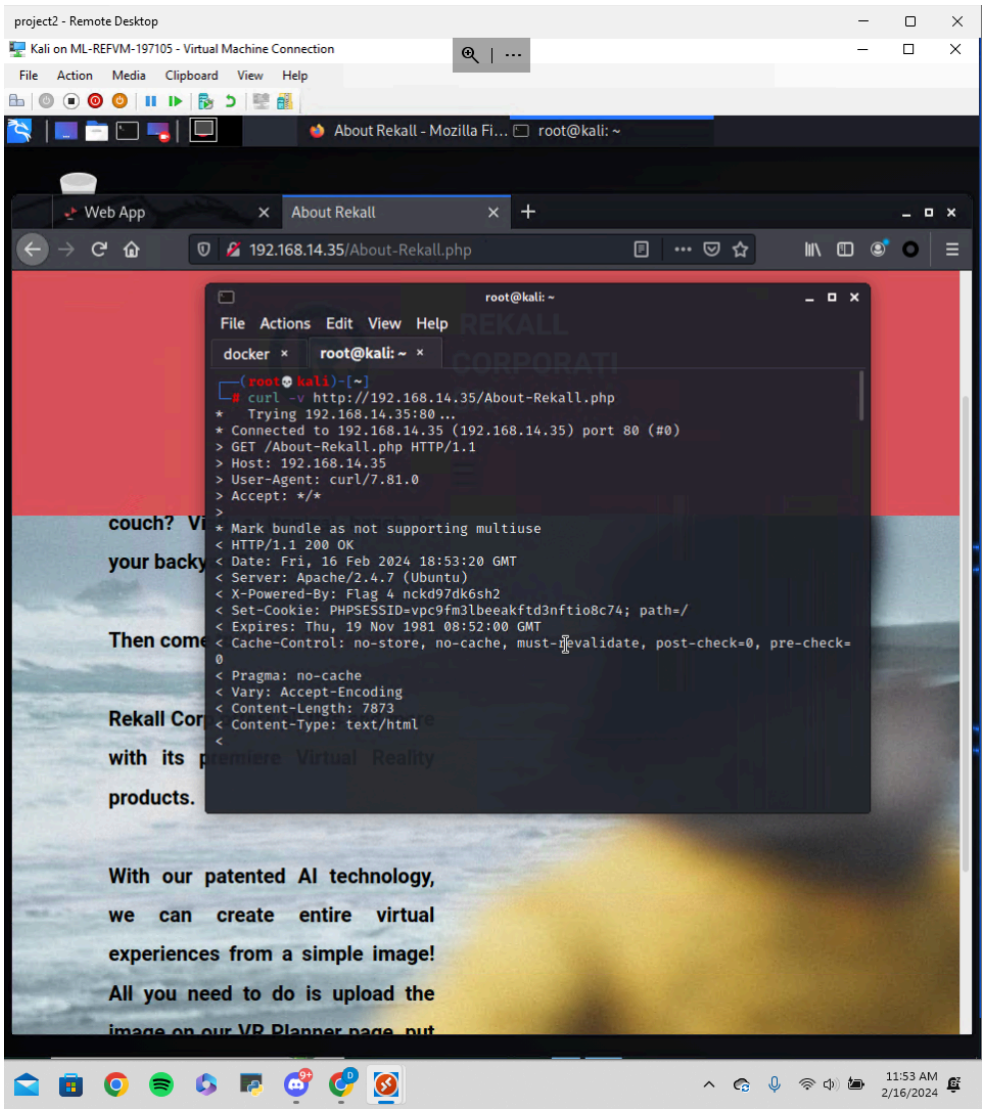


Affected Hosts	192.168.14.35/Memory-Planner.php
Remediation	More advanced input validation, removing the ability to use ‘<’ or ‘>’. As this field is to select from a list of options, it would be best practice to switch this input from a text box to a drop-down menu.

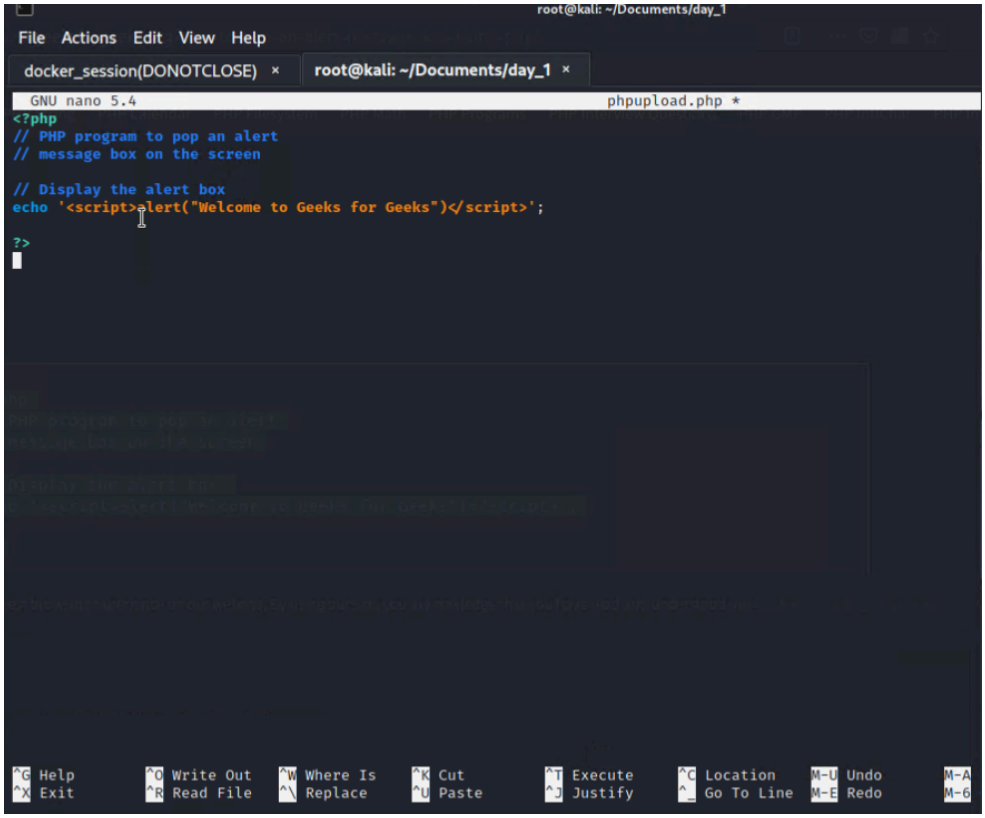
Vulnerability 3	Findings
Title	XSS stored, field name: “Please leave your comments on our website”
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	The comments section allows for a XSS stored attack. An attacker can enter a comment in the form of a malicious script, that is stored within the web apps HTML source code. When other users visit this page, a popup containing malicious will appear and potentially infect customers’ systems. The script used was <script>alert(‘flag3’)</script>
Images	# Image 1 shows the payload to be uploaded  # Image 2 shows a successful XSS and reveals the flag

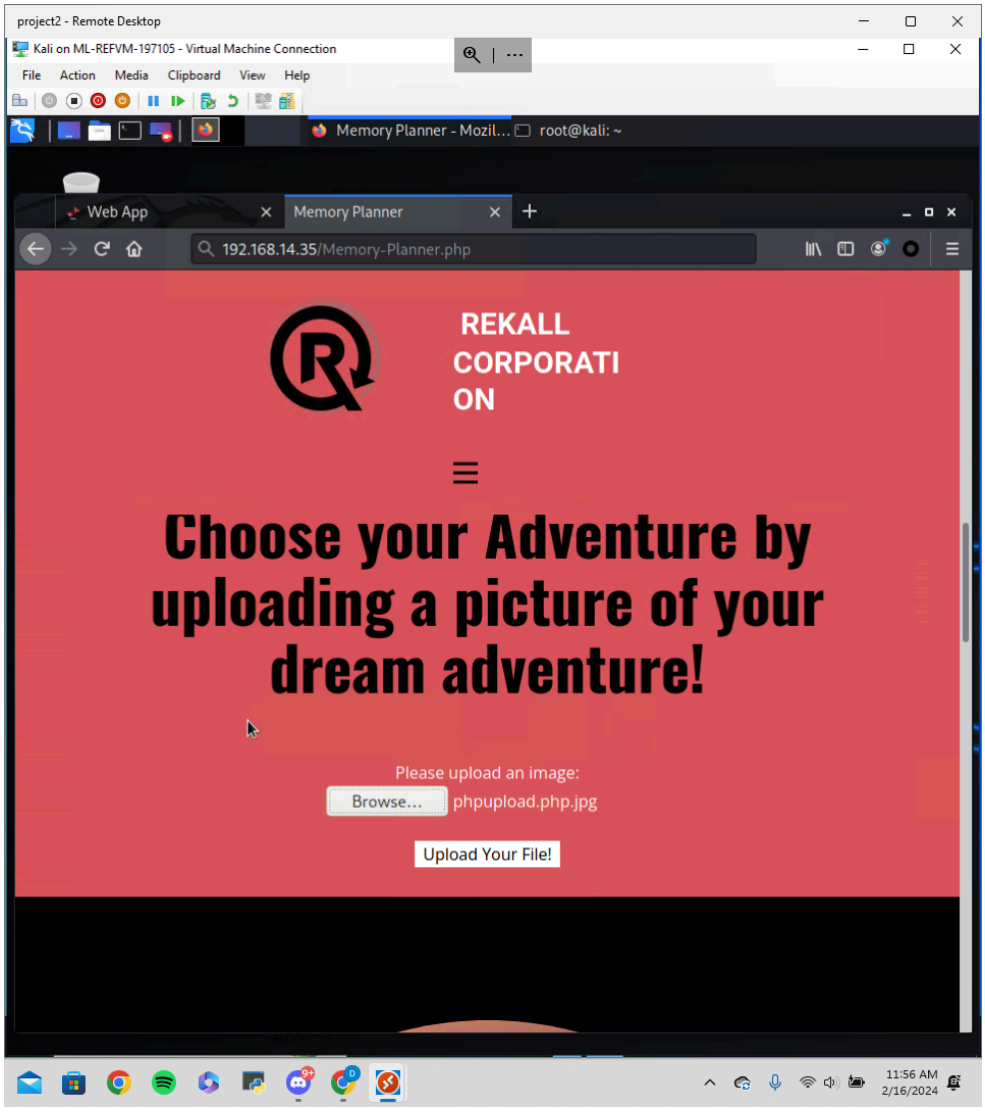
	
Affected Hosts	192.168.14.35/comments.php
Remediation	Input validation, only allow letters, numbers, commas, apostrophes.

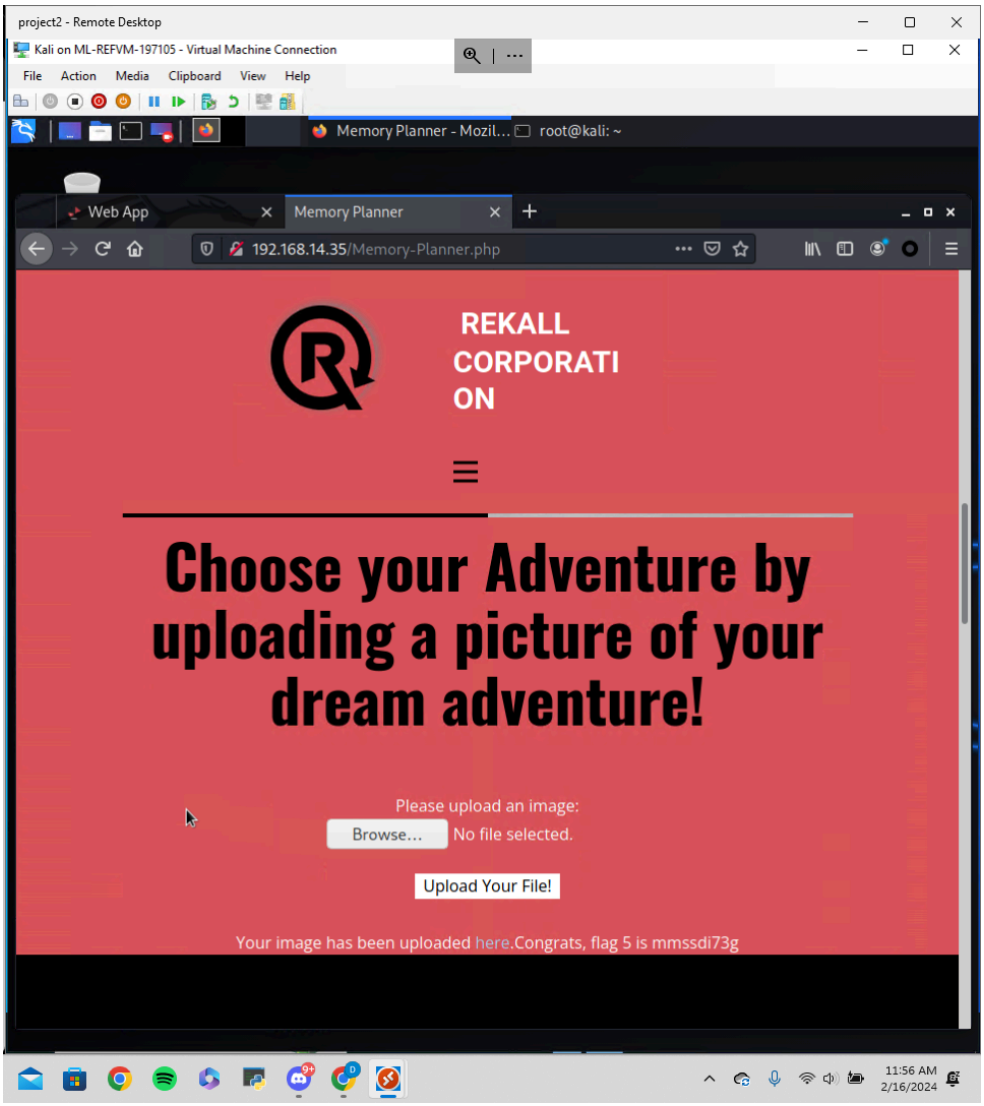
Vulnerability 4	Findings
Title	Sensitive data exposure, Location: About-rekall.php
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Medium
Description	Using the curl command, information about the web app that should be private can be accessed by unauthorized users. Using curl, I was successfully able to exploit a vulnerability in the API to return sensitive data.
Images	# Image shows the results of the curl command, exposing sensitive data and revealing the flag

	
Affected Hosts	192.168.14.35/About-rekall.php
Remediation	Only the minimum amount of data necessary should be exposed through the API, and properly validate and enforce permissions on API calls to guarantee only the authorized users can access sensitive data. Use of SSL protocols can authenticate and encrypt data to prevent sensitive data exposure.

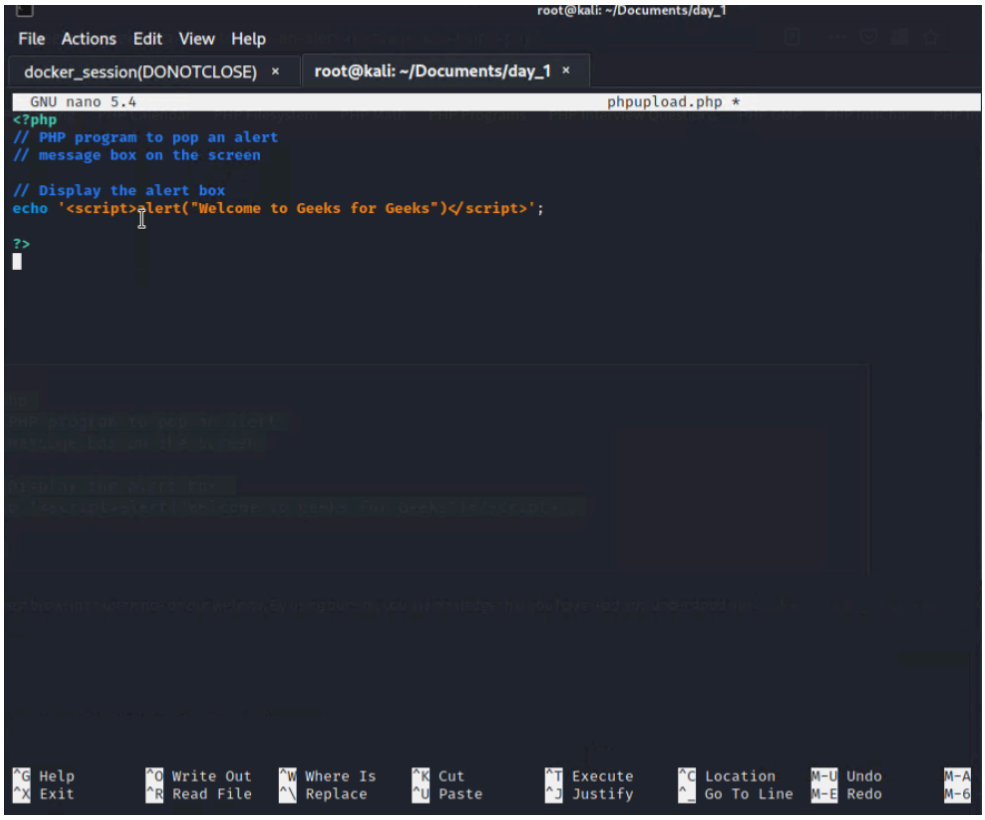
Vulnerability 5	Findings
Title	Local file inclusion, Location Memory-Planner.php
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Local file inclusion is a method to trick a web application into running or exposing files within the web server. In more severe cases, this can lead to cross-site scripting and remote code execution. Within rekall's web application,

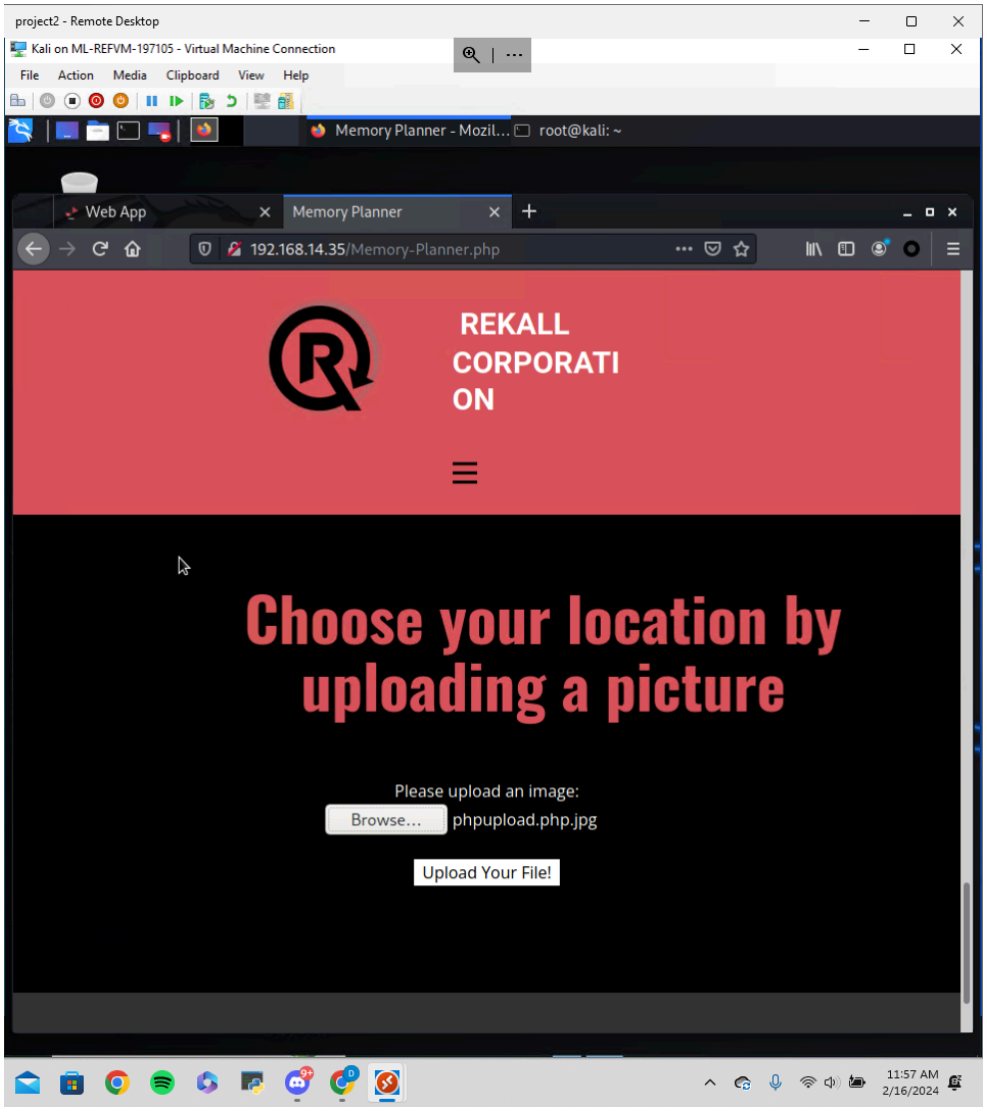
	the “upload your image field” is vulnerable to a php script upload, allowing attackers to run any server-side malicious code they want. For example, enumerate files stored within the web server.
Images	

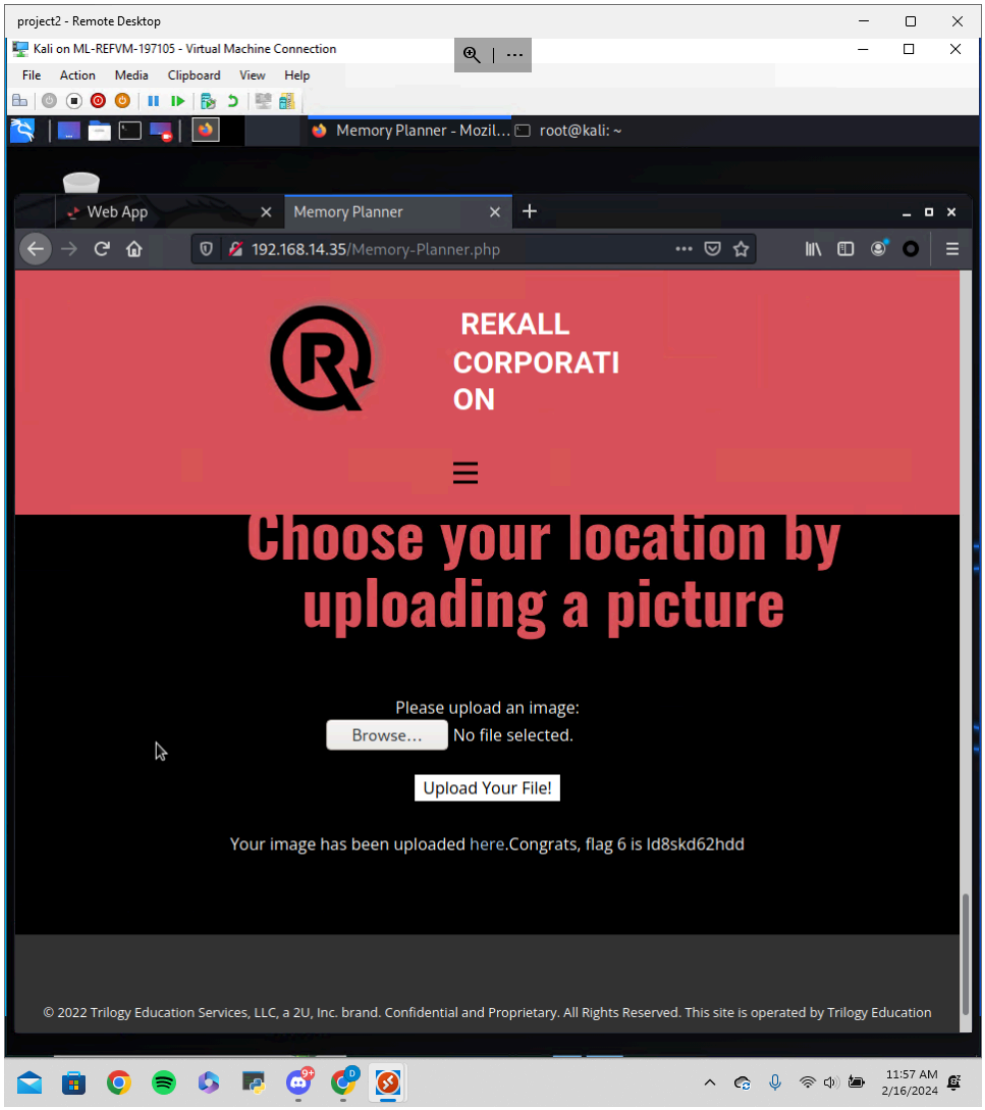


	
Affected Hosts	192.168.14.35/Memory-Planner.php
Remediation	There are several remediation methods to implement. One method would be to use a database instead of files on the web server that can be compromised. Another method would be to use whitelisting, only allowing specific actions and preventing anything else.

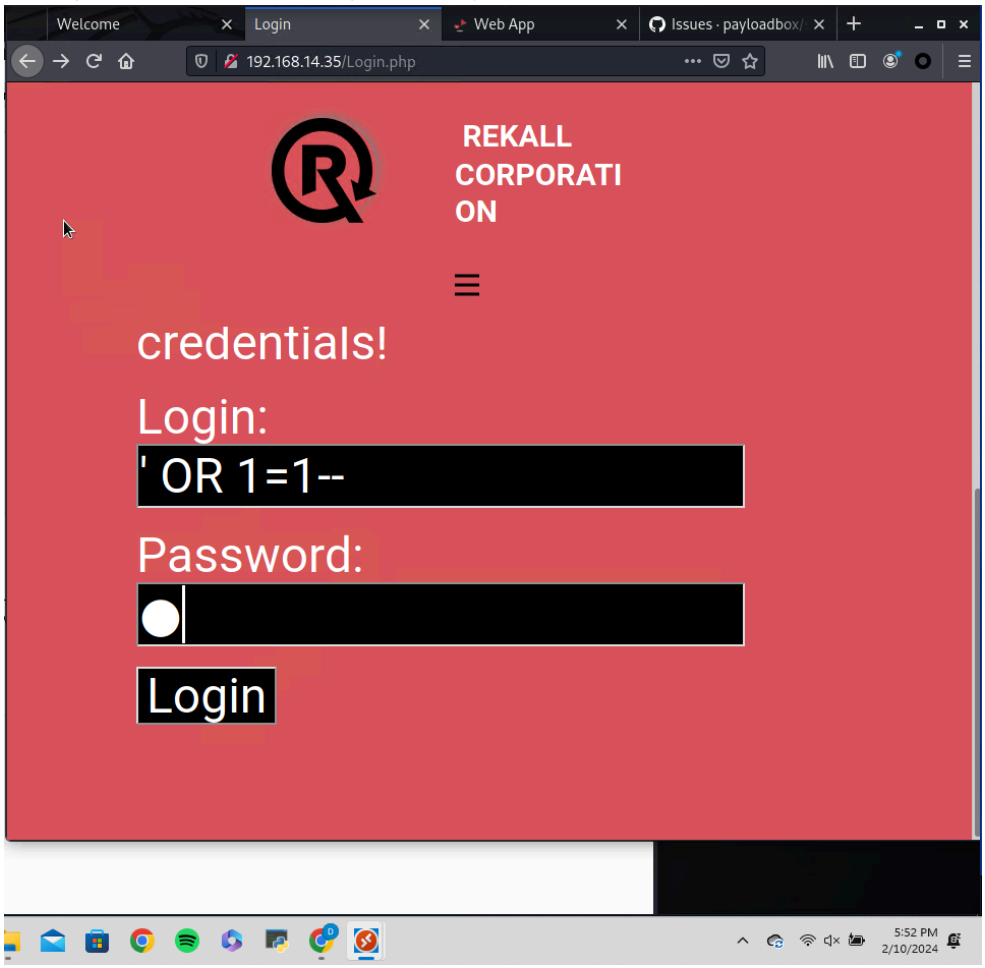
Vulnerability 6	Findings
Title	Local file inclusion, Location Memory-Planner.php
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	The security team attempted to use input validation, to look specifically for .jpg file uploads. However, this is easily circumvented by simply adding a .jpg extension to a php script. This will bypass the input validation, and allow for the

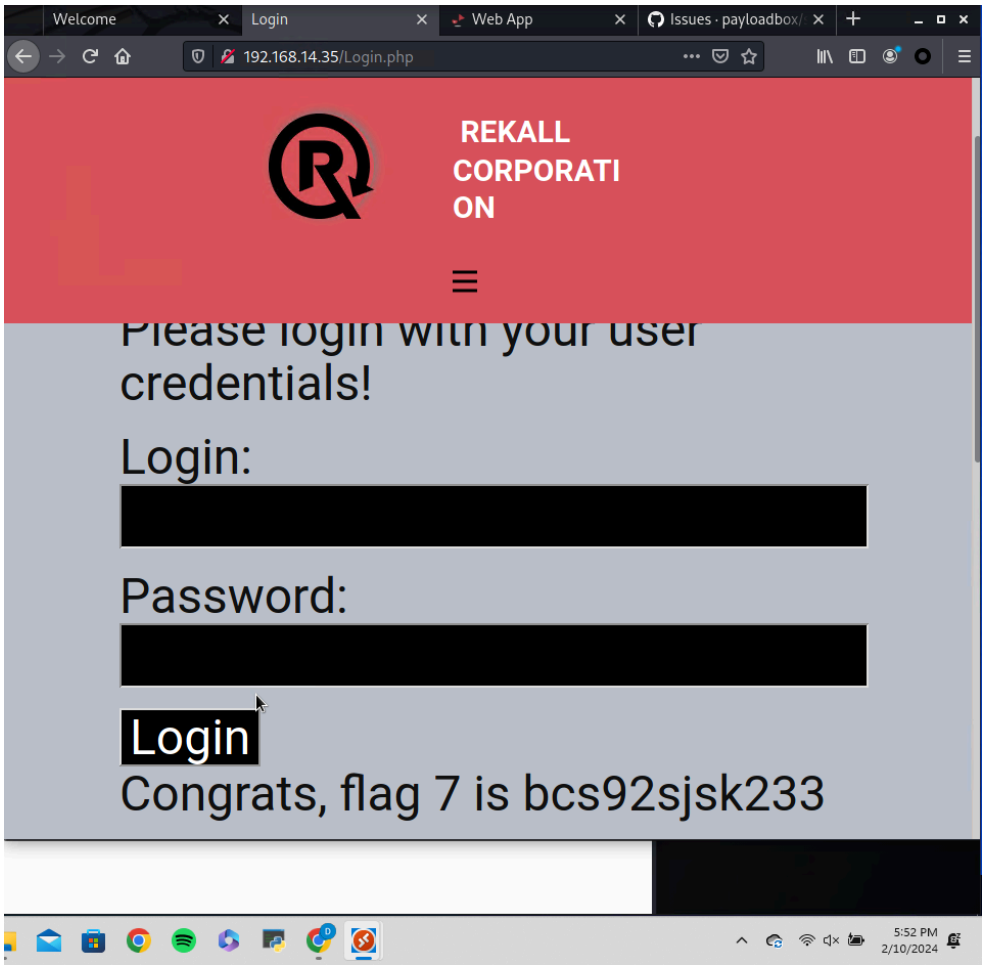
	exploit to be run without restrictions, exposing the web application to the same vulnerabilities as outlined in Vulnerability 5.
Images	 



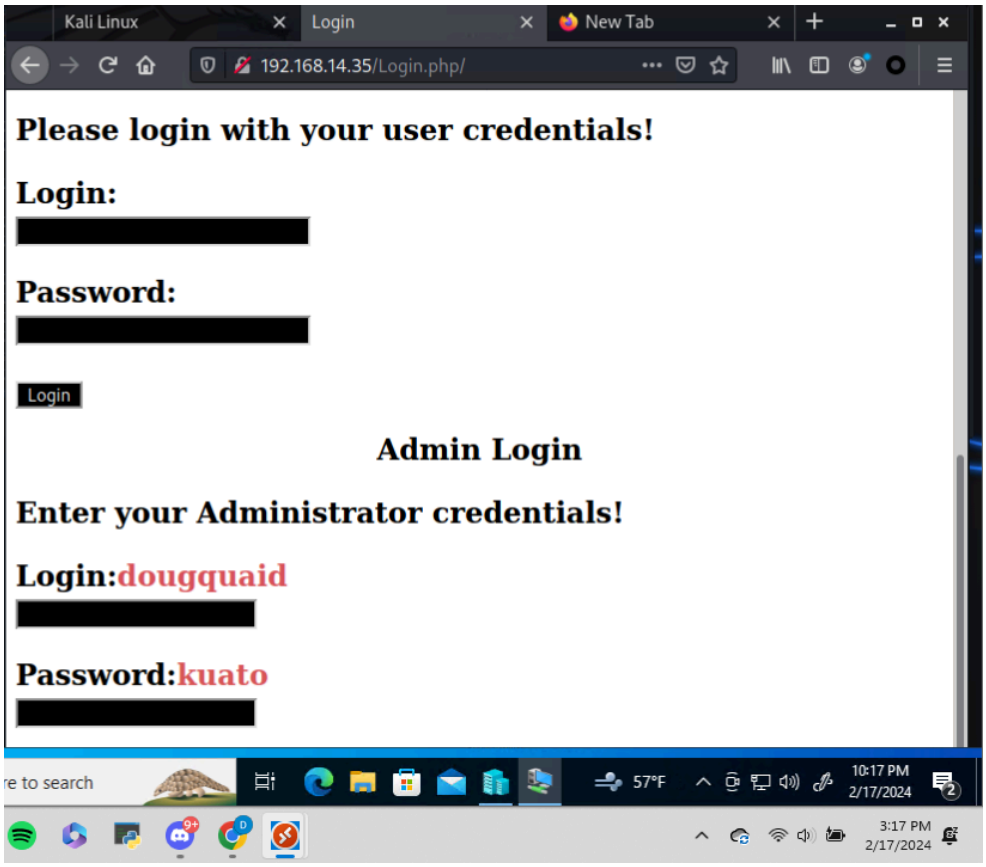
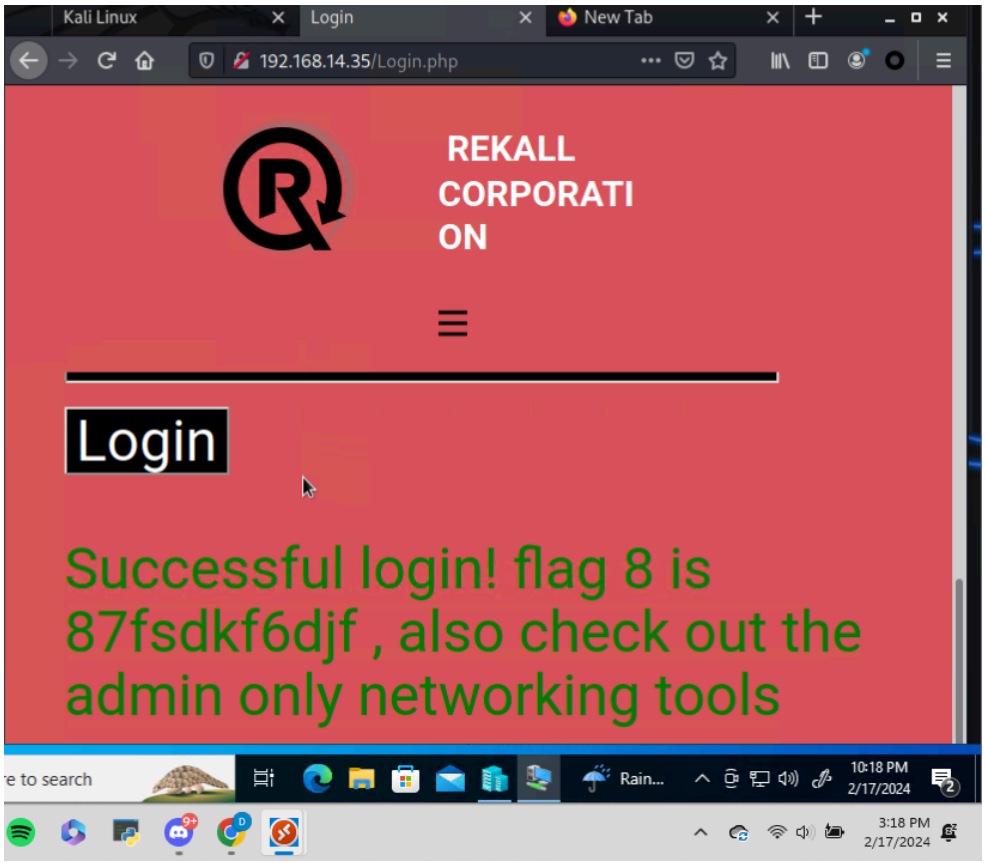
	
Affected Hosts	192.168.14.35/Memory-Planner.php
Remediation	The security team attempted to use input validation, to look specifically for .jpg file uploads. However, this is easily circumvented by simply adding a .jpg extension to a php script.

Vulnerability 7	Findings
Title	SQL injection, Location: login.php
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	The input for login is vulnerable to an SQL injection. By inputting ‘ OR 1=1 -- and a ‘space’ for password, an attacker is able to view the database from this field. They could also delete, encrypt, or add data to/from the database allowing for ransomware, DoS, and compromise of the entire CIA triad.

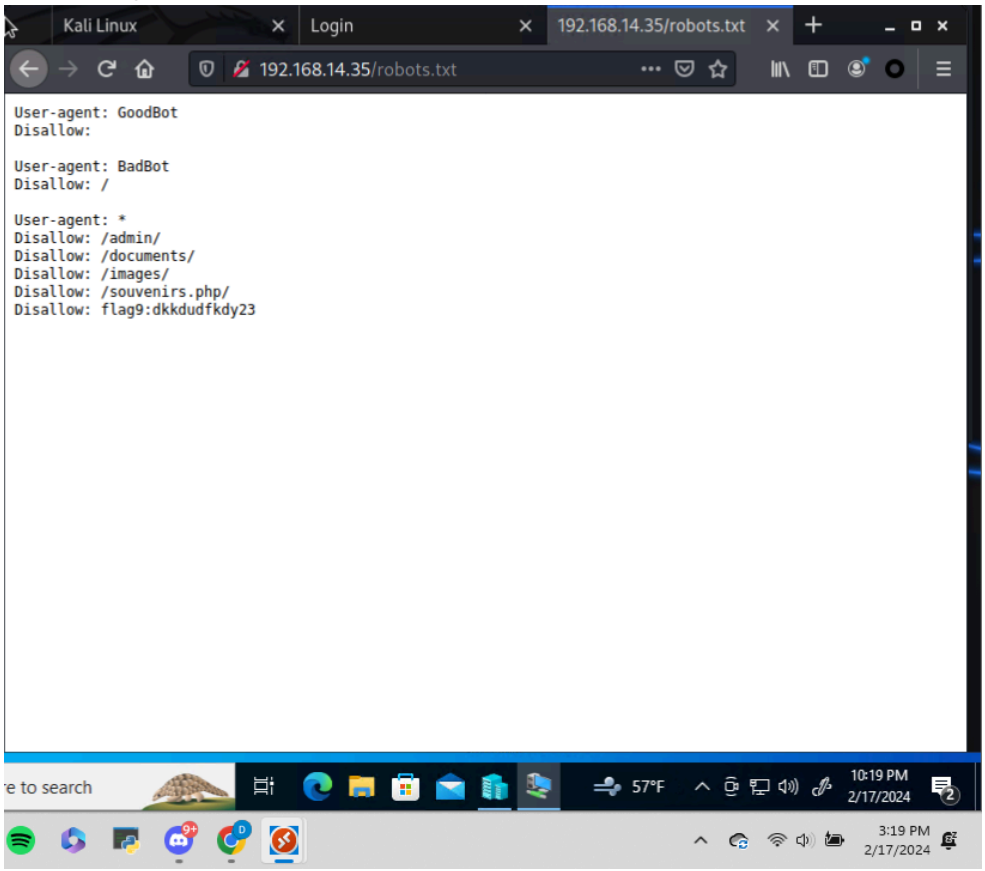
Images	# Image 1 shows the SQL injection payload  # Image 2 shows the successful exploit and reveals the flag

	
Affected Hosts	192.168.14.35/Login.php
Remediation	Input validation, allowing only letters and numbers in usernames. Another method would be parameterized queries, so in this case the parameterized query would look for a literal match of the entire string, ‘ OR 1=1 --. Thus protecting the database from malicious SQL injection.

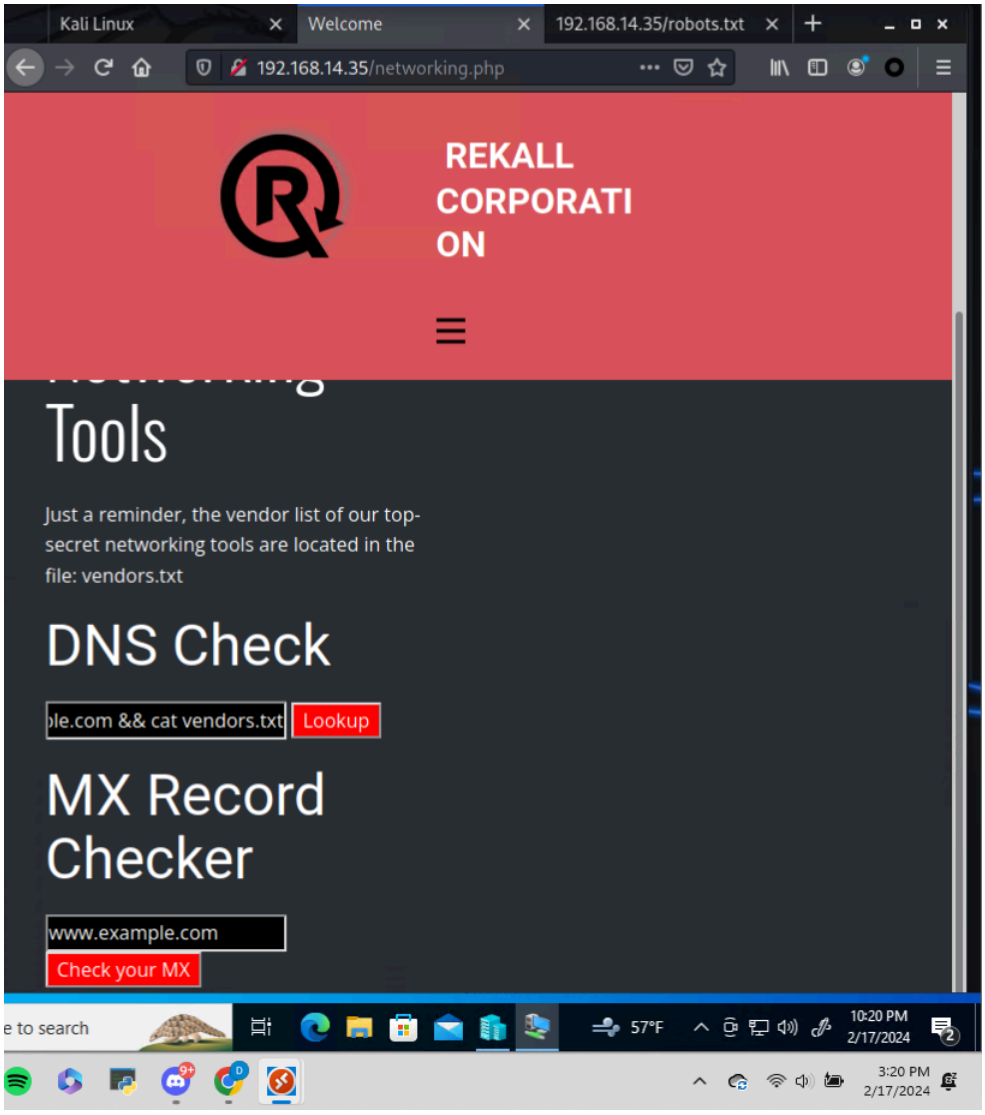
Vulnerability 8	Findings
Title	Sensitive data exposure, Location login.php, second field
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	By viewing the login.php page, accessed by switching the URL, the login page displays the username and password for an admin user to login. These login credentials are authentic and can be used for a successful login to admin account.

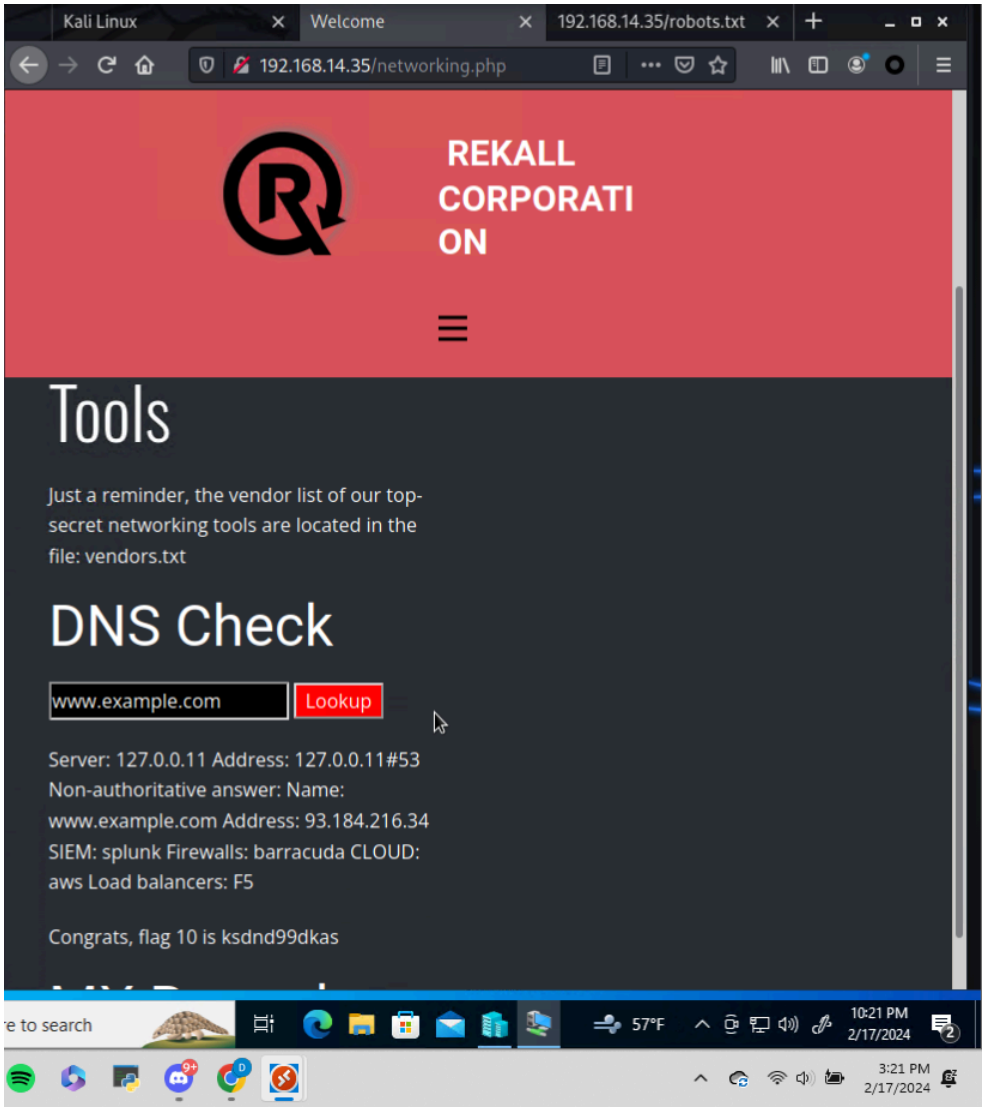
Images	 
Affected Hosts	192.168.14.35/Login.php
Remediation	Edit the HTML of the file and remove the login details. In general, caching

	should be disabled on pages that contain sensitive information, such as login info. Ensure that passwords are stored with an algorithm specifically designed for password protection, ie SHA-256 hashing. Adding salts to hashes will create added difficulty for an attacker to decrypt a hash if they are able to gain access to it. More complex password policies also will contribute to more difficult to crack password hashes.

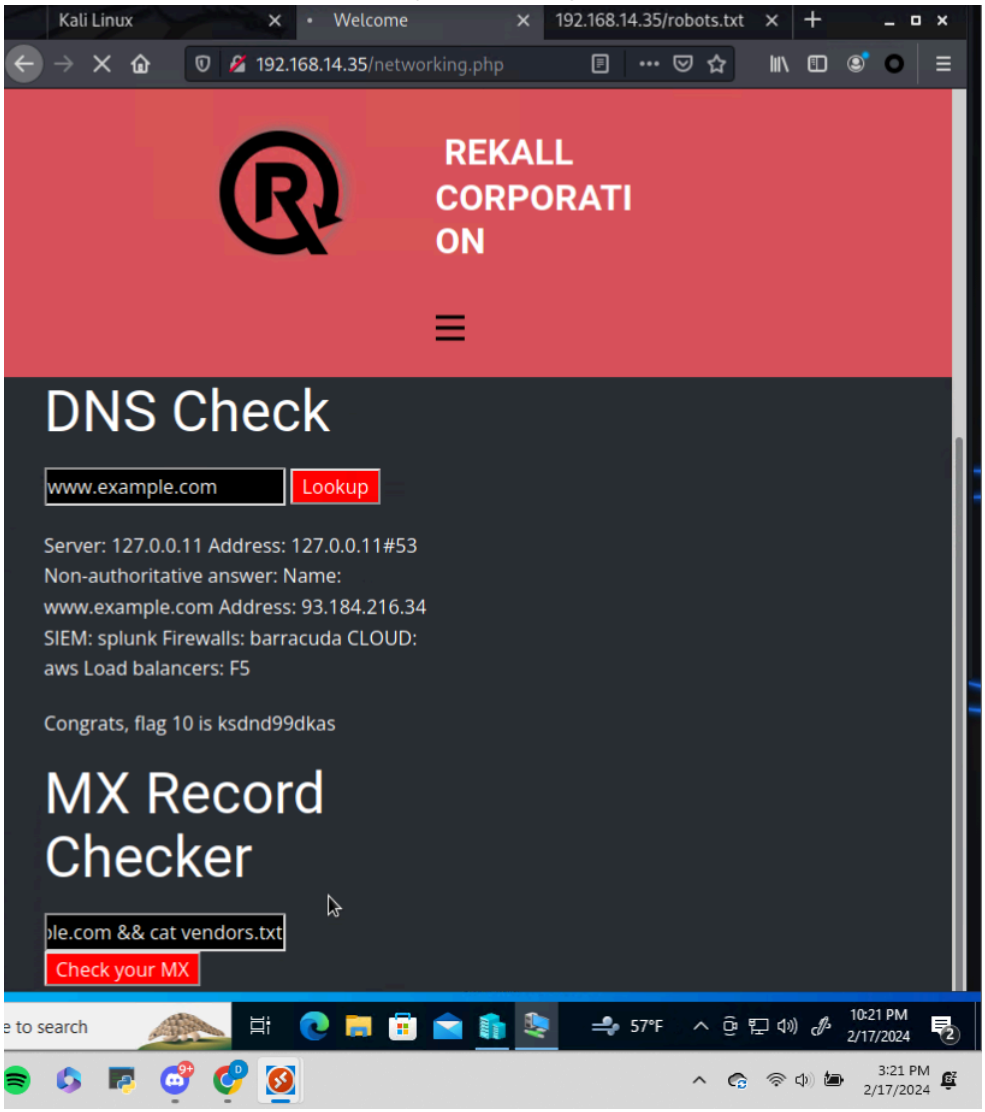
Vulnerability 9	Findings
Title	Sensitive Data Exposure
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Low
Description	The robots.txt file is accessible by the public. This file contains information regarding which pages can or cannot be indexed by web crawlers. However, it also reveals target URL extensions that may be of interest to an attacker. In Rekall's case, this file provides a roadmap to sensitive information on /admin, /documents, /images, and /souvenirs.php as these files are listed as 'disallow.'
Images	# This image shows the 
Affected Hosts	192.168.14.35
Remediation	For pages that should be private and NOT publicly accessible, password

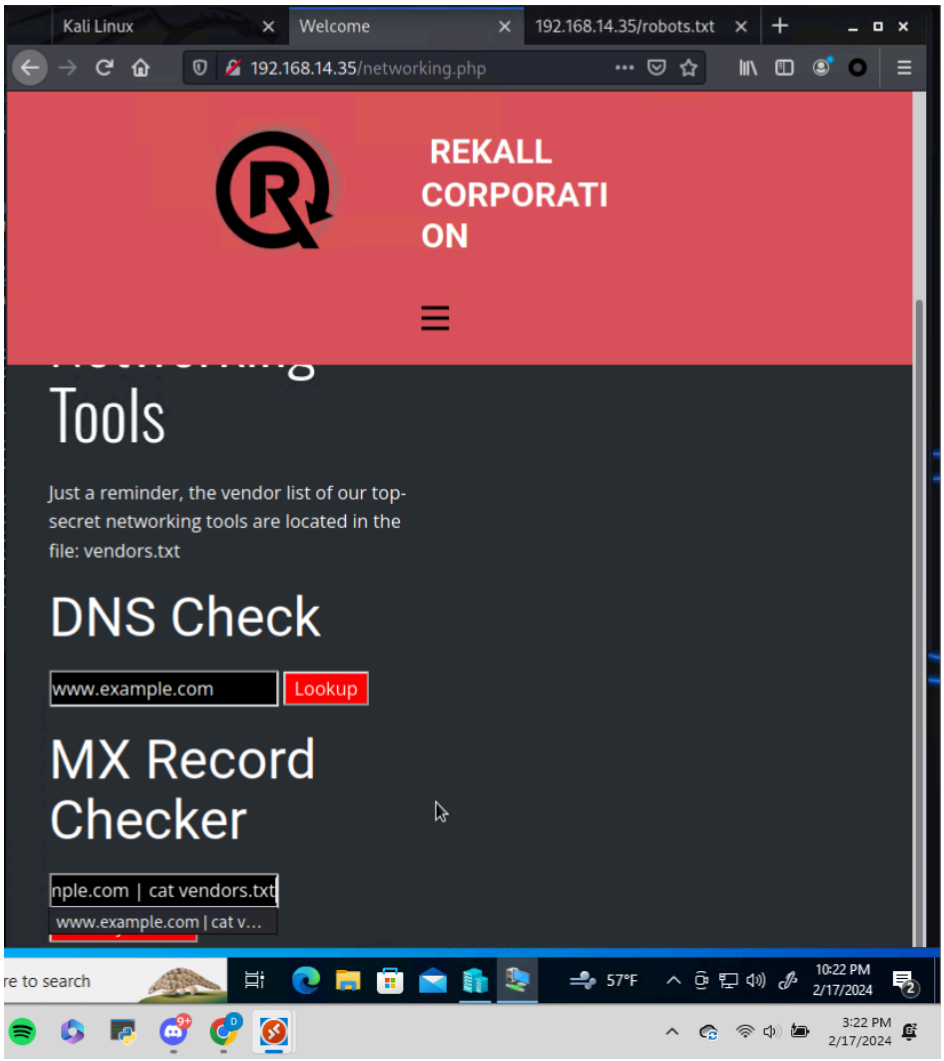
	protection of these pages or IP whitelisting is the best practice. The adoption of SSL protocol to authenticate and encrypt data could further protect sensitive data.

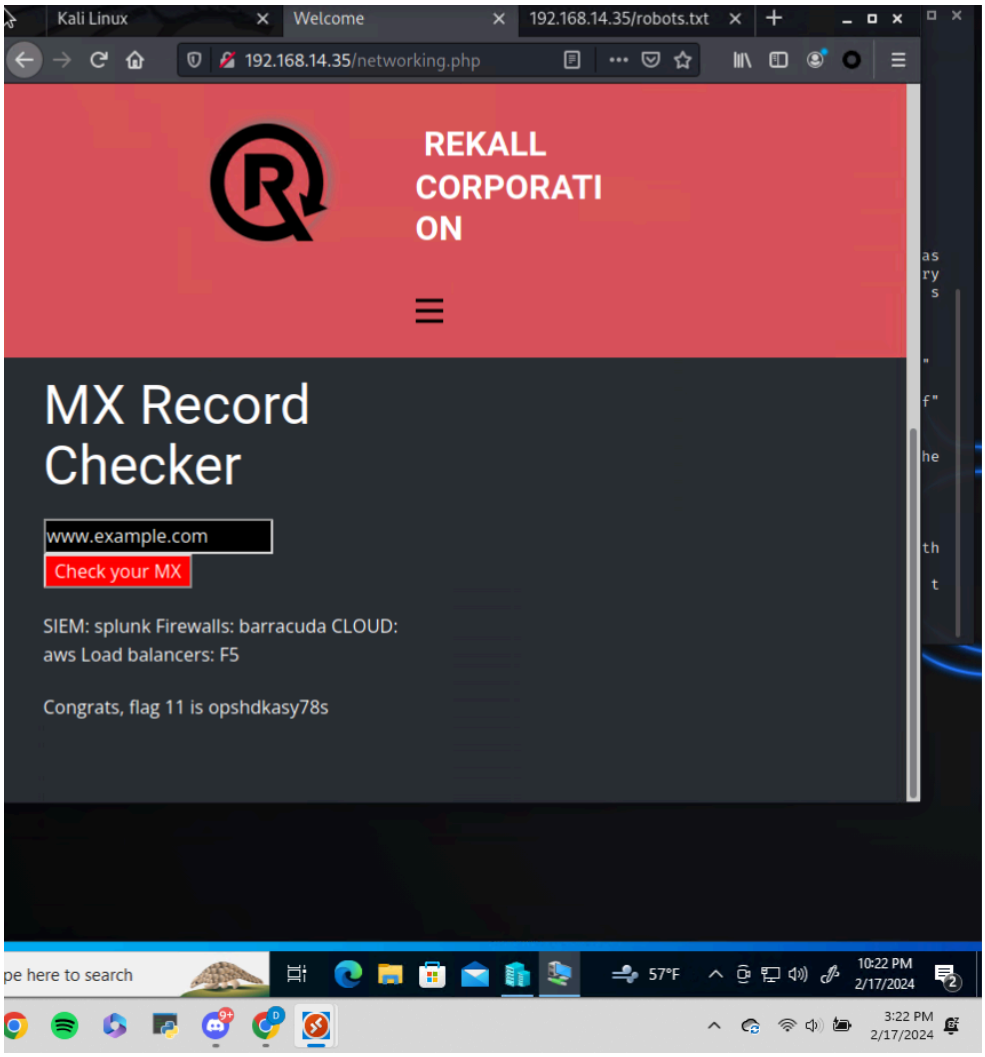
Vulnerability 10	Findings
Title	Command Injection
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	This web application is in communication with the OS, and the DNS check field is vulnerable to command injection. By submitting a url following by &&, a compound command is issued and information about the company can be revealed. From this field, entering the command: www.example.com && cat vendors.txt I was able to successfully reveal sensitive information from the web application.
Images	

	
Affected Hosts	192.168.14.35/networking.php
Remediation	Input validation, in the form of whitelisting letters, numbers, and periods. Disallow the rest.

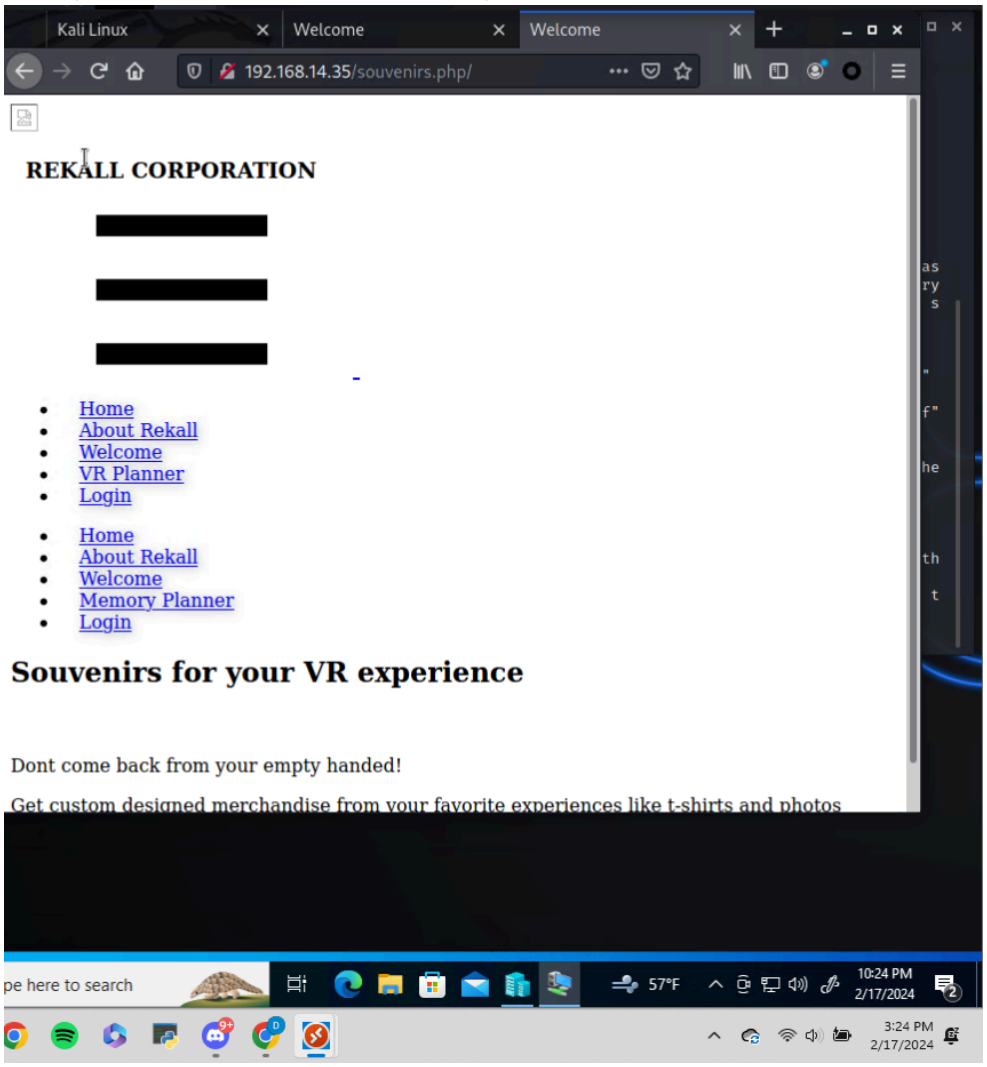
Vulnerability 11	Findings
Title	Command Injection
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Critical
Description	The MX Record check field is vulnerable to command injection. There is some input validation in place on the web application, disallowing the use of ampersands. However, compound commands can also be issued using a pipe |. This leaves the field vulnerable and possesses the same risks as those outlined in Vulnerability 10.

Images	# The first attempted command injection, using &&, resulted in a failed attempt  # The second image shows the payload using a pipe to issue the compound command



	 # Image 3 (above) shows the successful command injection on the MX Record Checker
Affected Hosts	192.168.14.35/networking.php
Remediation	Input validation, in the form of whitelisting letters, numbers, and periods. Disallow the rest.

Vulnerability 12	Findings
Title	PHP Injection
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Using information gained in the robots.txt, the URL extension /souvenirs.php was added. This page offers the ability to use PHP injection to enumerate sensitive data from the system. I was able to use PHP injection by altering the URL to http://192.168.13.35/souvenirs.php?message=""; system('cat /etc/passwd')

	This resulted in the /etc/passwd file being displayed on the page.
Images	# Image 1 shows the souvenirs.php page  # Image 2 shows that clicking a link sends a message and reveals the possibility of a PHP injection

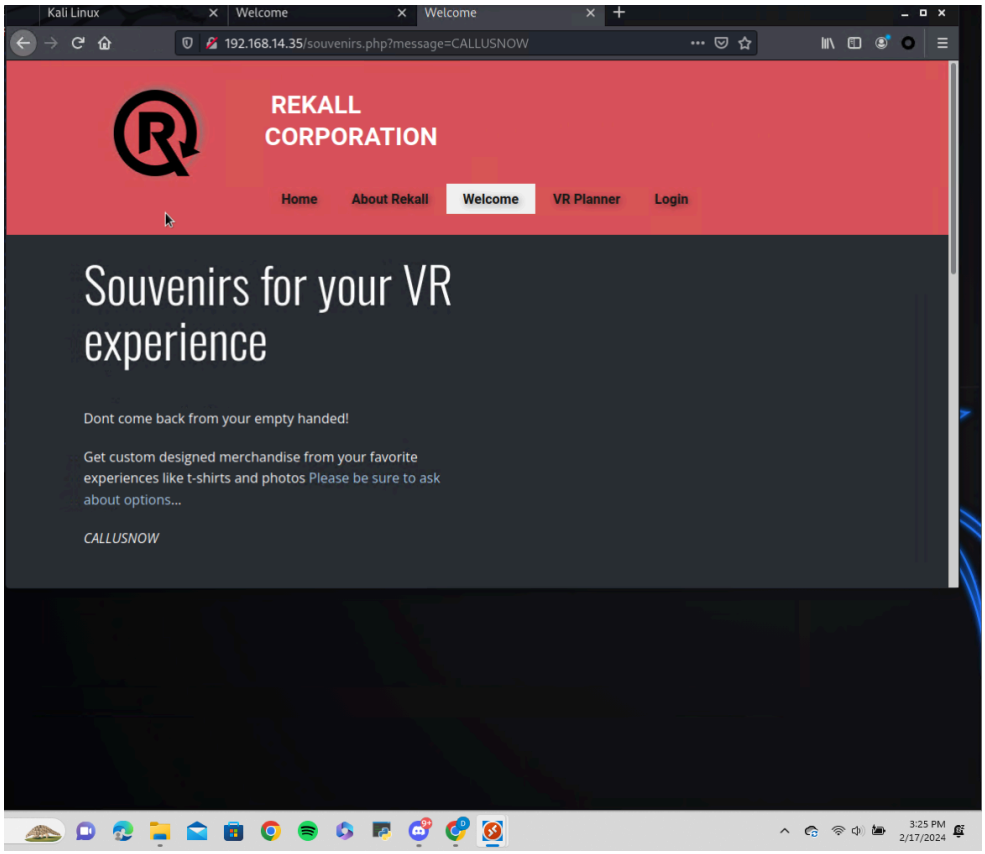


Image 3 shows the PHP injection payload used

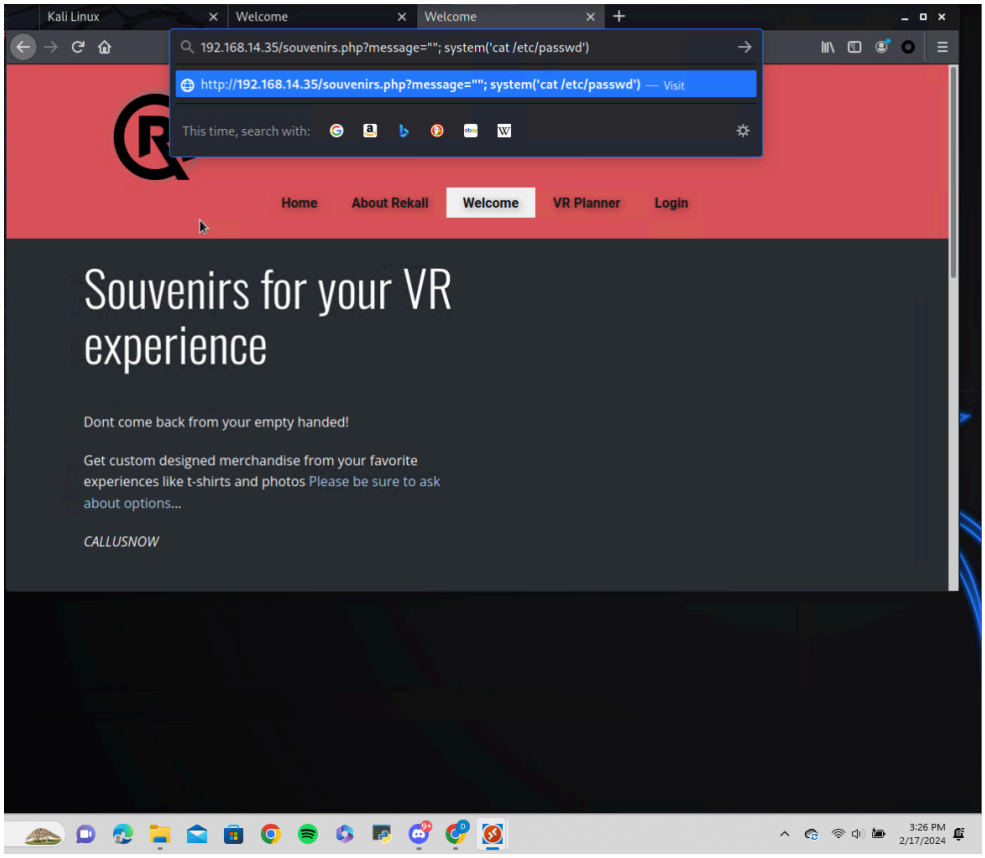
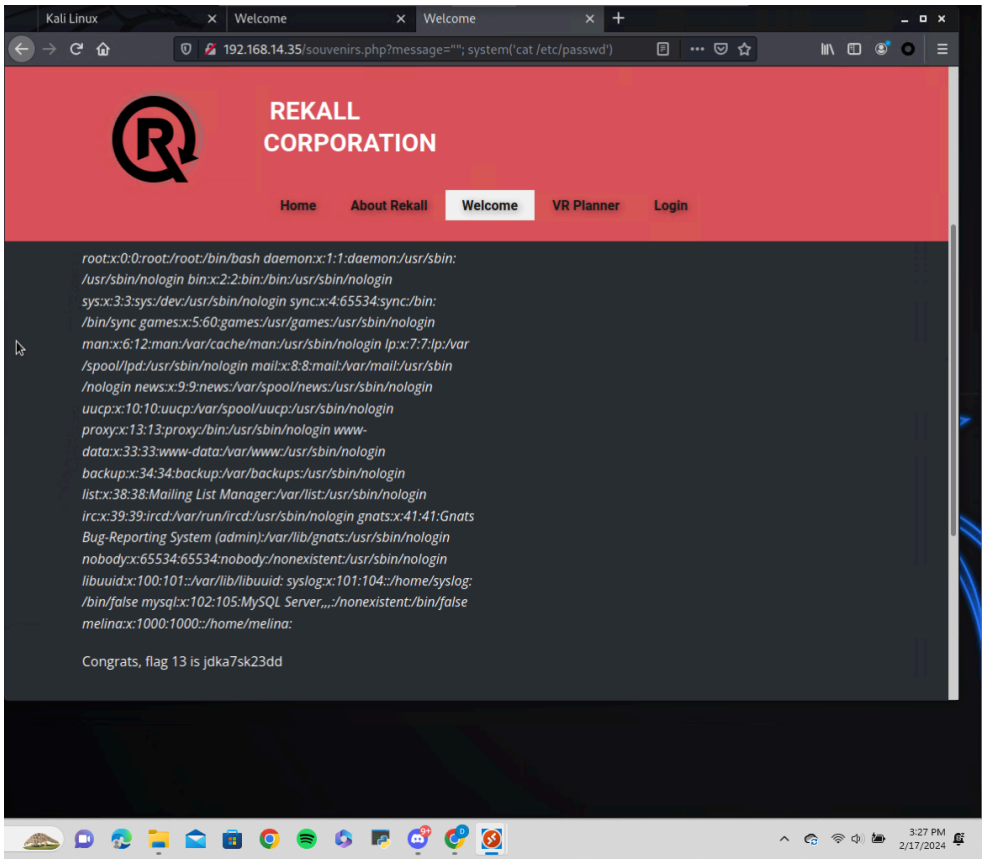


Image 4 shows the successful PHP injection, revealing the contents of /etc/passwd as well as flag 13.

	
Affected Hosts	192.168.14.35/souvenirs.php
Remediation	Issue a SSL certificate to change the webpage from HTTP to HTTPS. Sanitize everything coming in, and as a fundamental rule, avoid the use of shell_exec(), exec(), system(), and passthru() where possible, as these operations execute at the OS level. The code can be run through software built for code injection protection to highlight any areas of risk within the PHP code. The application should also avoid displaying verbose error messages, as these can be used by malicious actors to identify sensitive information related to the PHP application and web server.

Vulnerability 13	Findings
Title	Directory Traversal
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Utilizing the vulnerabilities found within the DNS and MX record checkers, we can issue the compound command ls to list the contents of the directory. Within the results of this ls command, I was able to find the ‘old_disclaimers’ directory. Another command ‘ls ./old_disclaimers’ revealed that the directory contains a file called disclaimer_1.txt. Using directory traversal within the URL, the contents of this file were revealed.

Images

Image 1 shows the LS command used

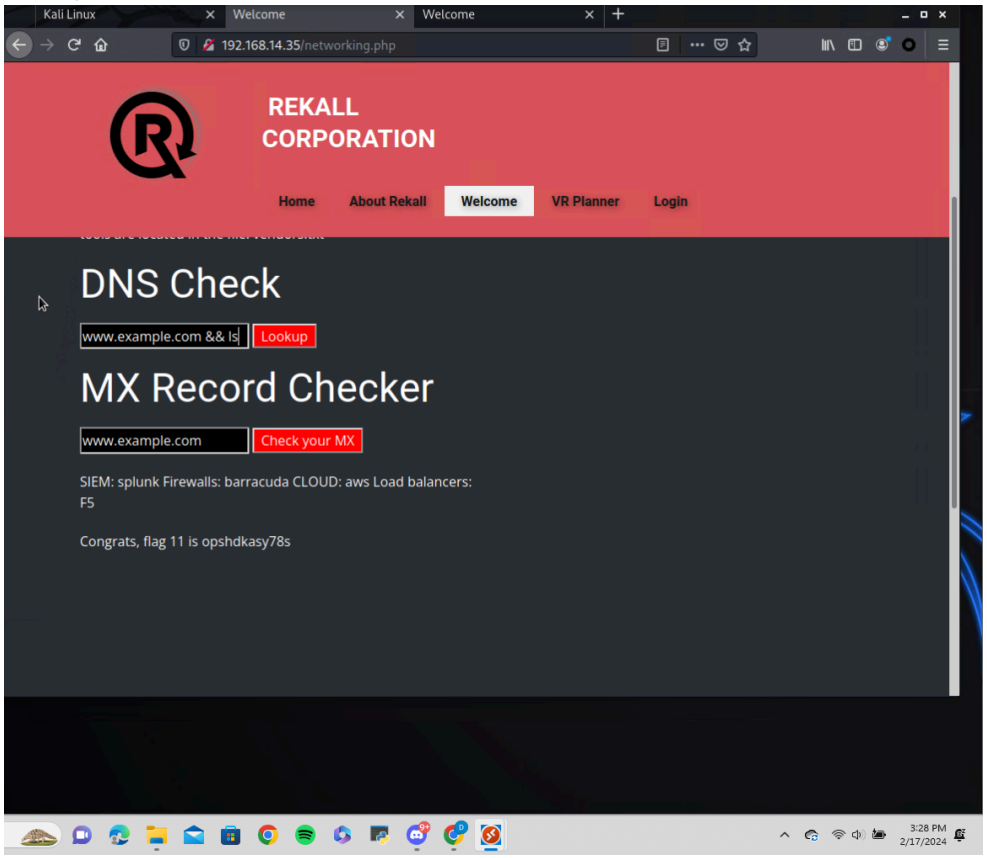


Image 2 shows the results of this ls, with old_disclaimers highlighted

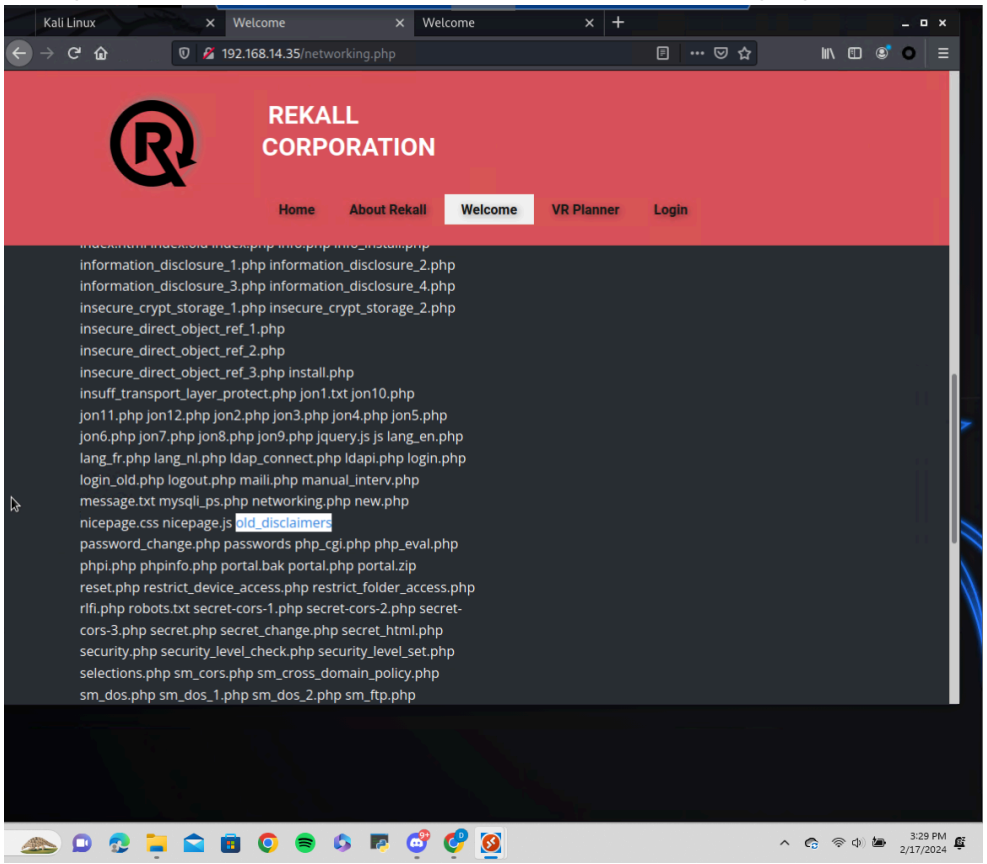


Image 3 shows the results of running ls ./old_disclaimers, in other words reveals the contents of that directory.

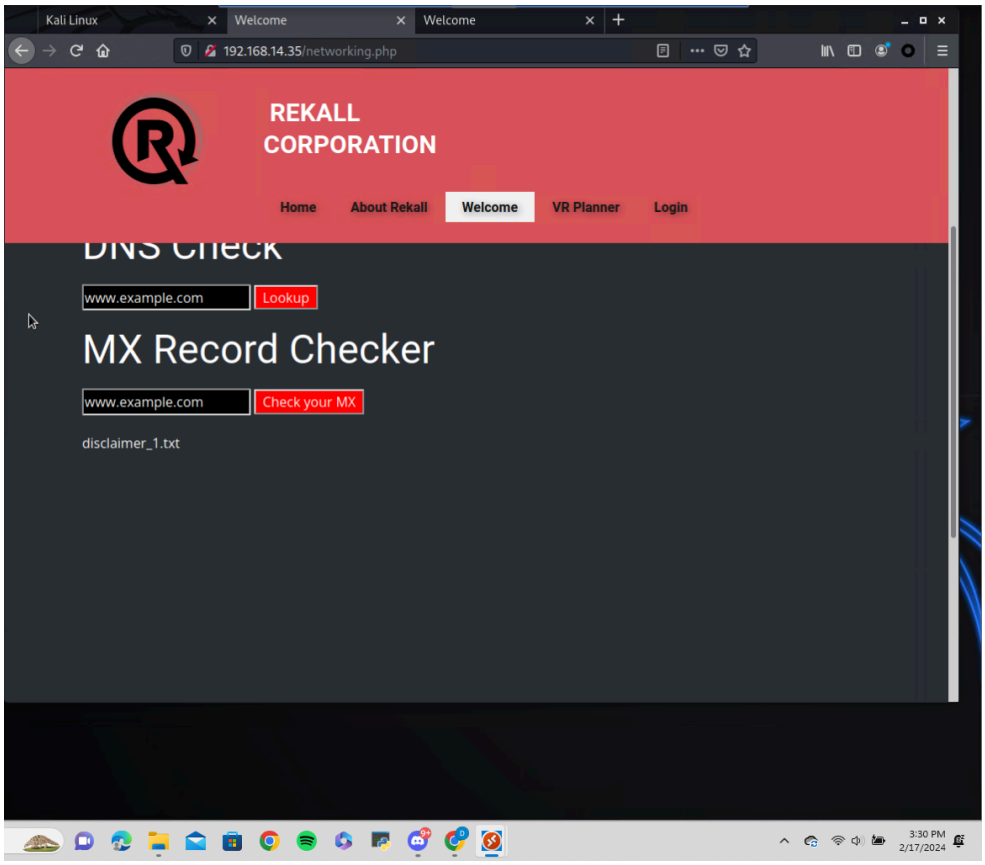


Image 4 shows the directory traversal input used

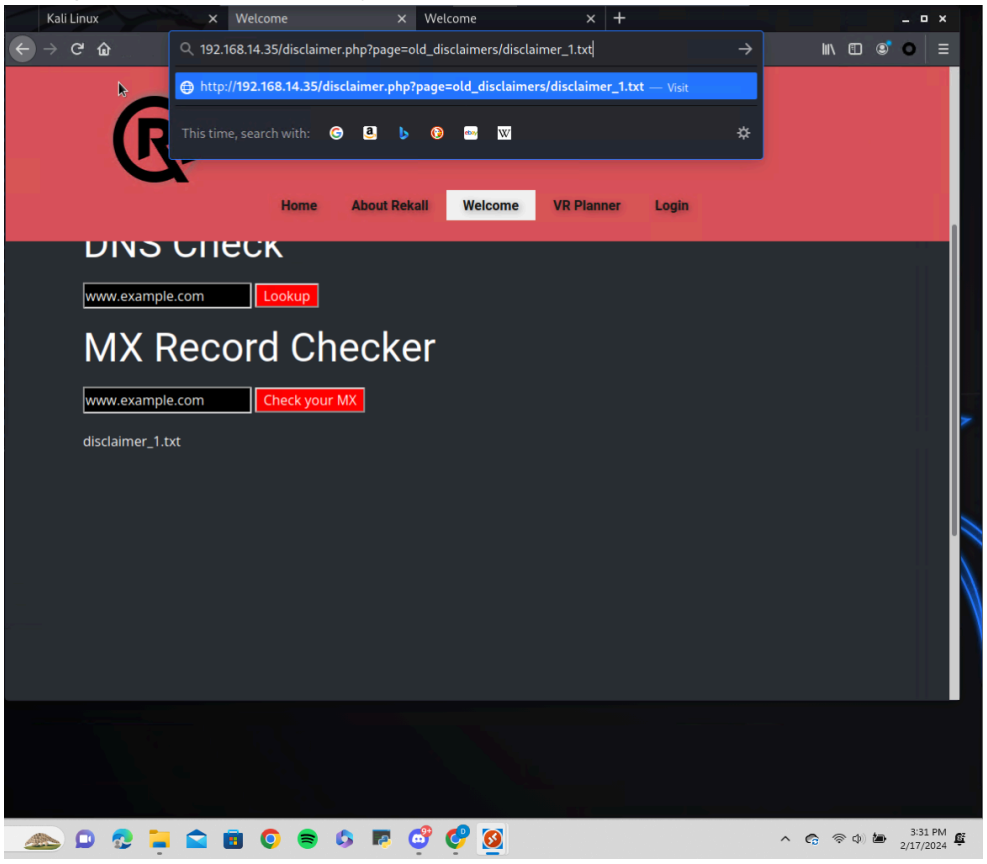
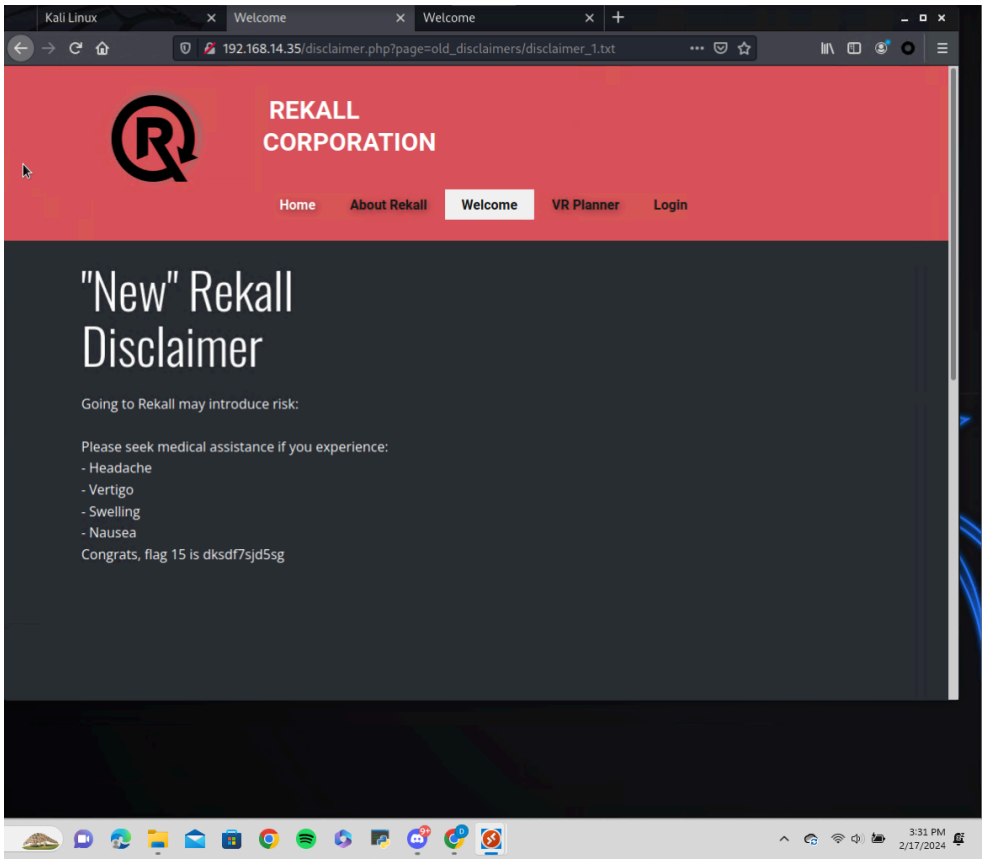
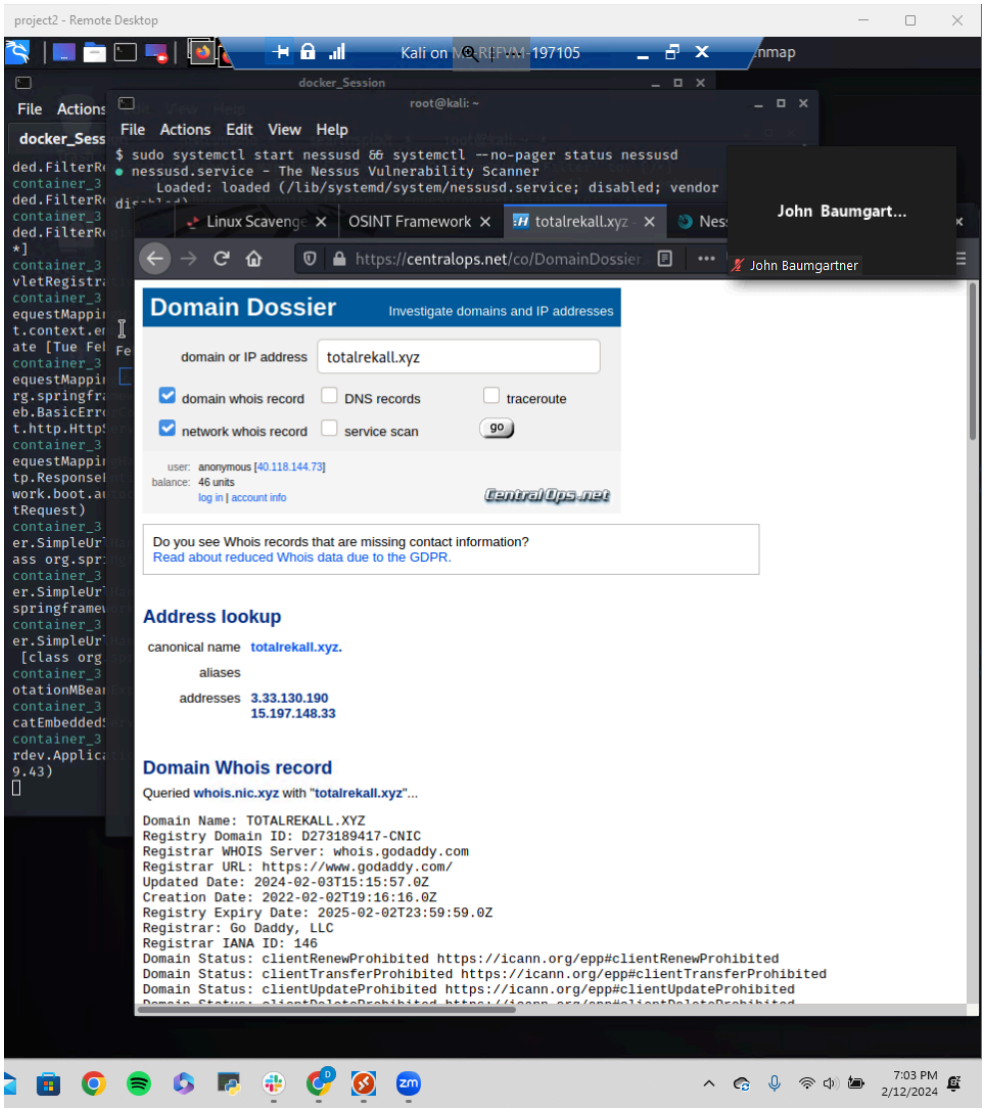
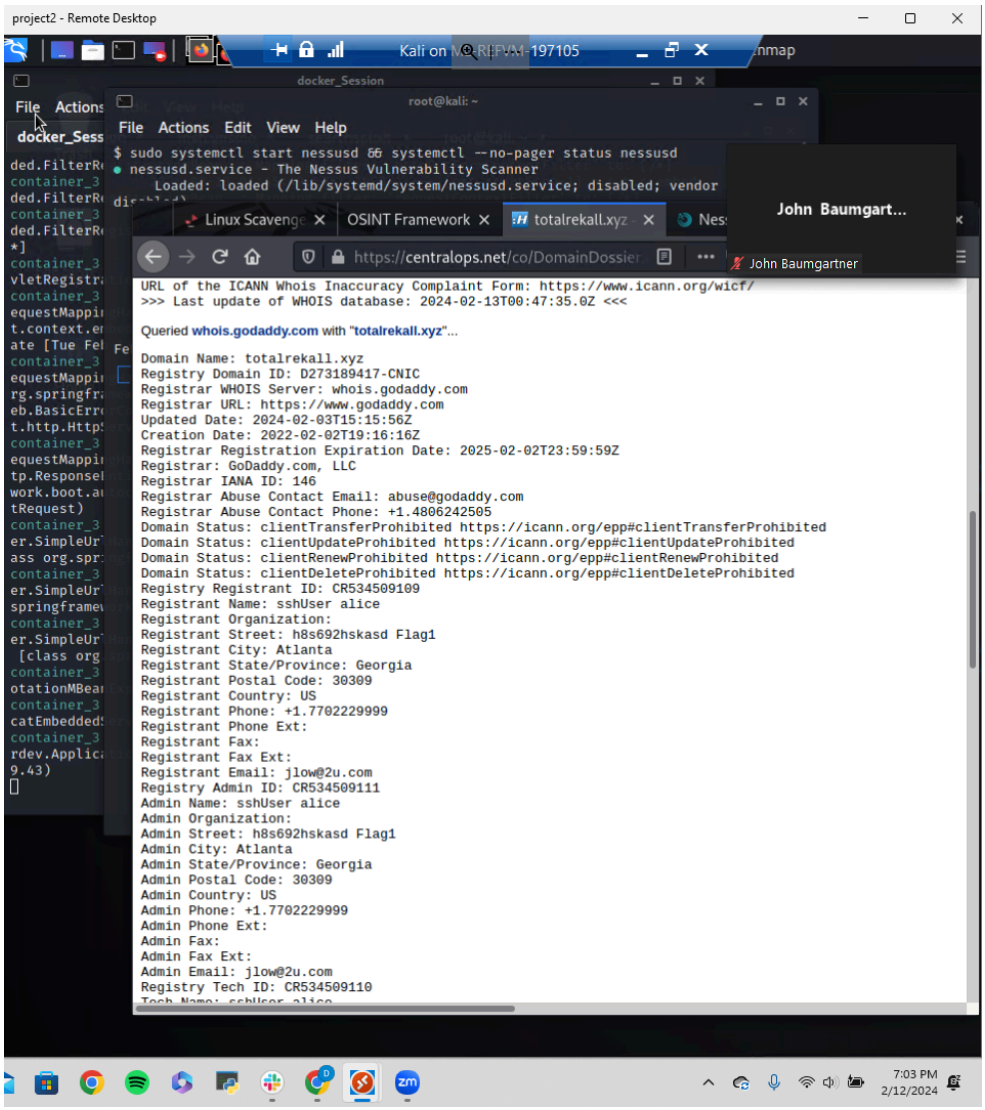


Image 5 shows the successful exploit and reveals flag 15

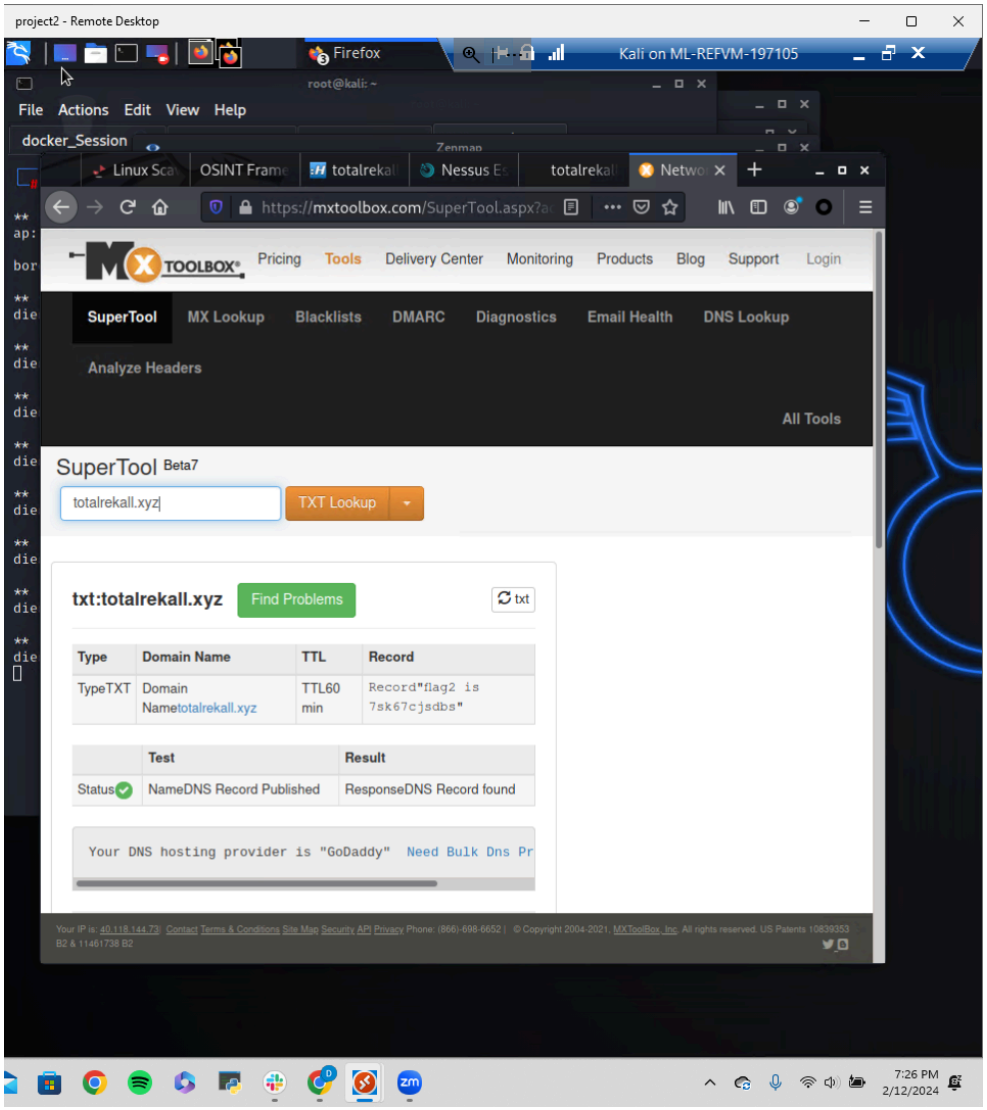
	
Affected Hosts	192.168.14.35/disclaimer.php
Remediation	Ensure all user inputs are validated and sanitized before they are processed. Reject any suspicious characters, such as ../ or ..\ Use the principle of least privilege, limiting what the attacker may be able to access. Deploy a web application firewall that examines HTTP traffic and identifies patterns that match known attack vectors. Update and patch systems regularly, many updates provide updates that fix security risks, including directory traversal attacks.

Vulnerability 14	Findings
Title	Open source exposed data
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Low
Description	Using Dossier to lookup rekall's publicly available data, I was able to enumerate a large amount of data. Although this vulnerability on its own may not be worthy of critical risk rating, the data that is exposed can be used to

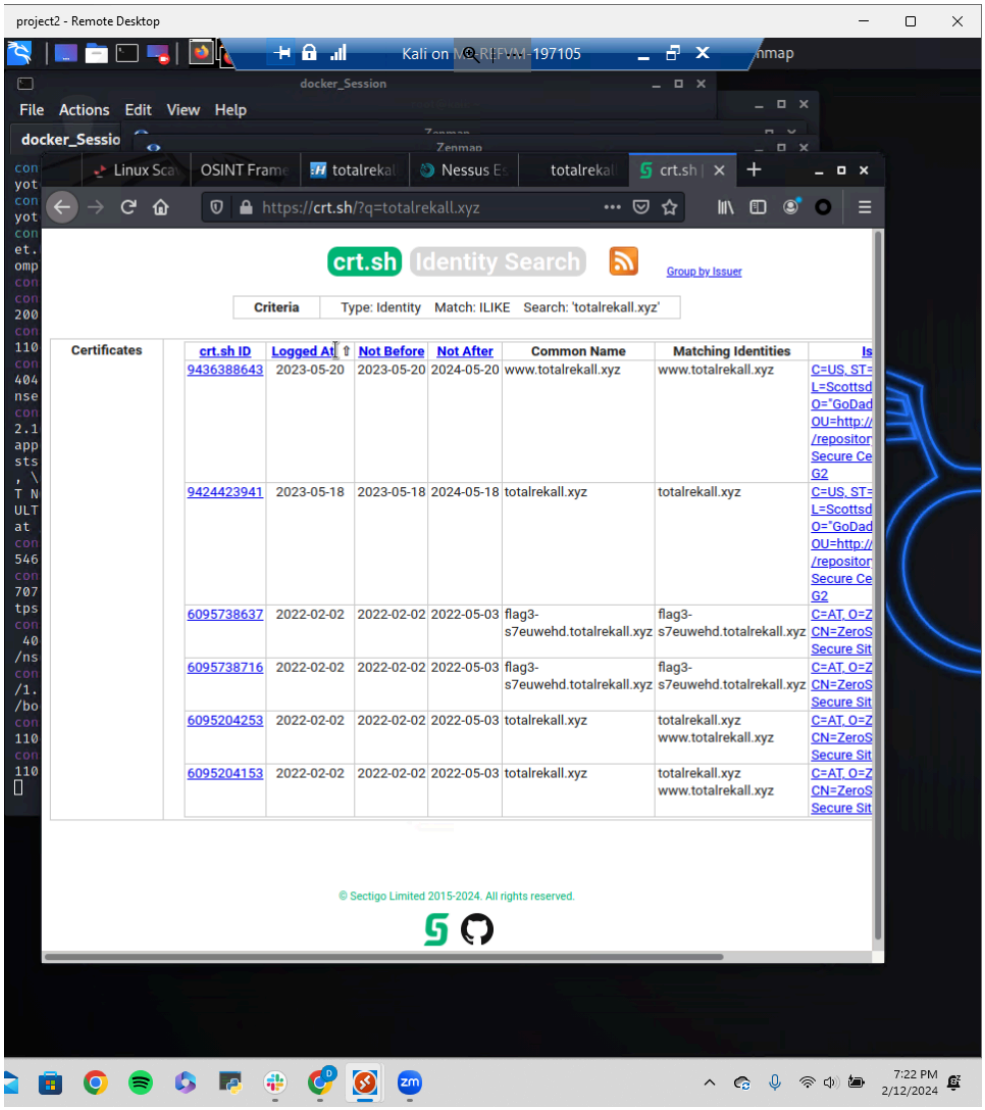
	gain admin access to the system via SSH, as well as poses a risk in revealing physical addresses and admin emails.
Images	

	
Affected Hosts	Domain: totalrekall.xyz
Remediation	Change what information is public facing, and set it to private.

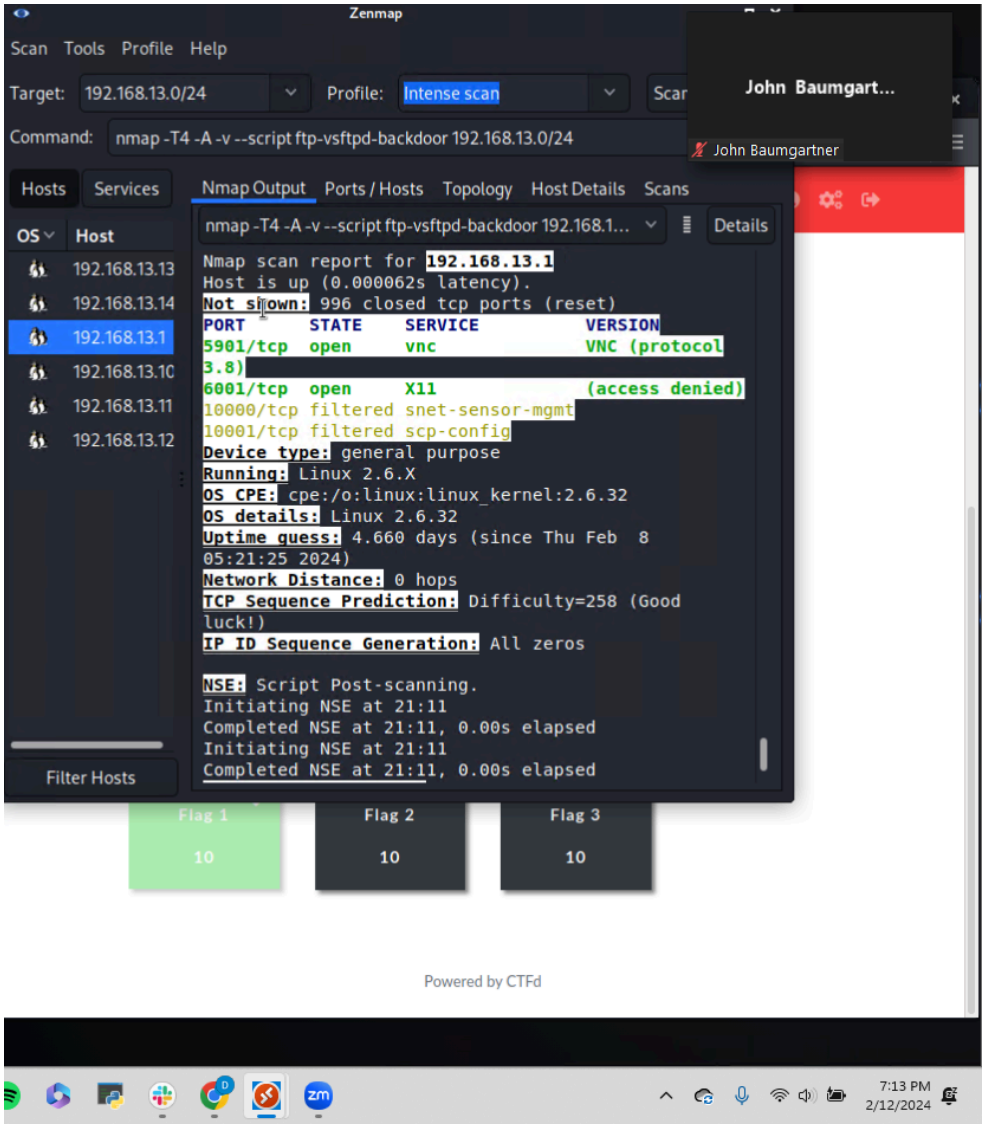
Vulnerability 15	Findings
Title	OSINT, txtlookup of totalrekall.xyz
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	informational
Description	Using txtlookup tool, some reconnaissance information was gathered.

Images	
Affected Hosts	Domain: totalrekall.xyz
Remediation	Sensitive data should not be included in the txt record as it is publicly available.

Vulnerability 16	Findings
Title	Open source exposed data/certificate transparency
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	informational
Description	The use of crt.sh can find records of domain certificates, and helps the attack to enumerate the network and search for subdomains. This is dangerous due to the fact that the attacker can keep their identity secret while gathering information on the target.

Images	
Affected Hosts	Domain: totalrekall.xyz
Remediation	As certificates are needed for customers' browsers to connect to the webpage, this cannot be removed. They can sometimes cause more harm than good, especially in the hands of a skiller cyber criminal. However, having proper detection methods in place can help defenders detect and prevent attacks against their domain.

Vulnerability 17	Findings
Title	Network scan, zenmap
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	informational
Description	Using zenmap, the details of the network and subnet can be enumerated in great detail, Revealing which hosts are on the system. From this can, we can see my IP (192.168.13.1) as well as 5 other Linux hosts. This scan contains

	details as to which ports and services are open and running.
Images	
Affected Hosts	192.168.13.0/24
Remediation	The best defense against scanning is a well configured firewall. The primary rule of firewalls, deny by default, should be applied. After everything is blocked, override that rule's priority to only allow essential traffic. Also, logging port scans can prove useful. Although not all scans signify an attack, having the log records in place can be useful to recognize patterns, a single IP performing multiple scans for example, could then be blocked by the firewall. Finally, a good offense sometimes leads to the best defense. Regular scanning of your company's network can illuminate what areas might be vulnerable to an attack.

Vulnerability 18	Findings