White Hat Lucas

Penetration Testing Report

Rekall Corporation

Penetration Test Report

Confidentiality Statement

This document contains mock penetration testing information from the fictional organization Rekall Inc. (henceforth known as Rekall). The information contained in this document is not confidential in any way. This penetration test was conducted as an educational exercise and all target machines and applications were hosted offline within a virtual machine controlled by WhiteHatLucas. Forwarding, printing, copying, distribution, or use of the information within this report is permitted.

IF this were a real penetration test for a corporate client, the Confidentiality Statement would look something like this:

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

Introduction

White Hat Lucas conducted external penetration tests of Rekall's networks and systems. The purpose of this engagement was to assess the security of an Apache web server, Windows 10 workstations, and the TotalRekall web application. We sought to identify potential security flaws by utilizing industry-accepted testing methodology and best practices.

Penetration Testing efforts were 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: Objectives

Objective
Find and exfiltrate any sensitive information within the domain.
Escalate privileges.
Compromise multiple 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 including 192.168.13.0/24, 172.22.117.20 and the totalrekall.xyz domain at 192.168.14.35. 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.

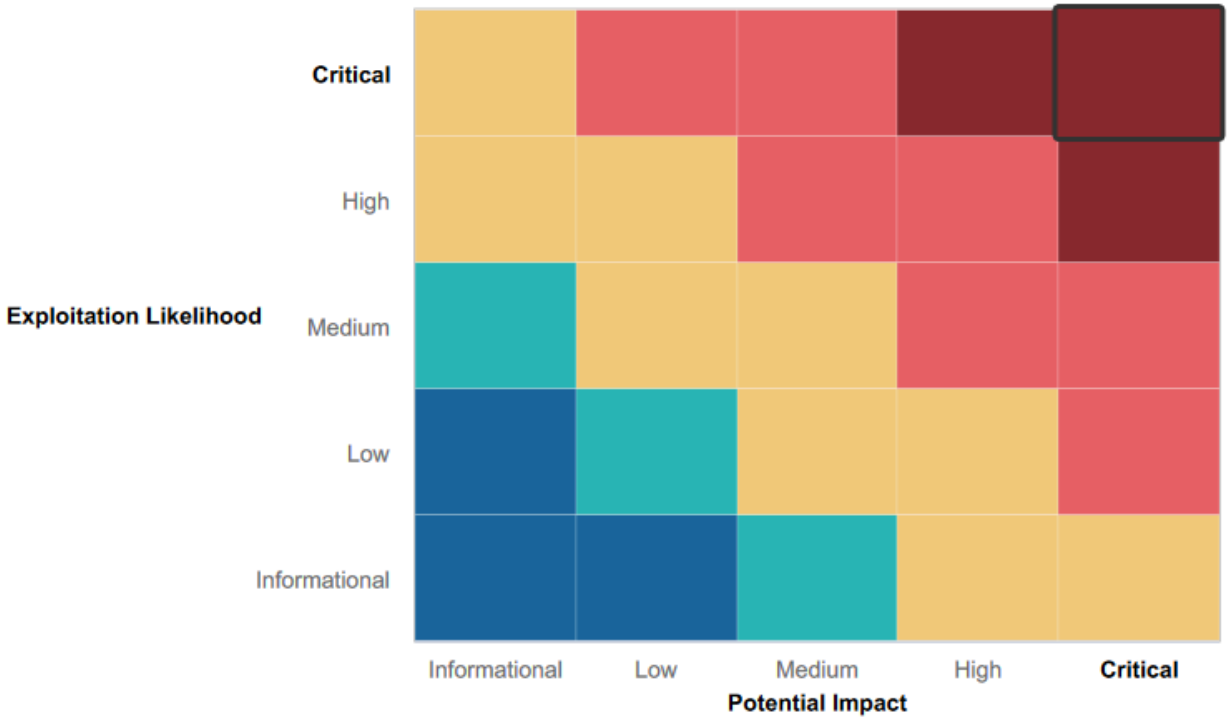
Summary of Findings

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.

- Certain assets such as the Memory Planner on the web-application deployed security measures such as input validation
- The image upload page checks the file type to ensure it's a .jpg
- The admin login at totalrekall.xyz/Login.php was not vulnerable to brute force password attacks

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 was vulnerable to common attack methods such as Command Injection, XSS, File Inclusion and SQL Injection
- Admin login credentials were stored in clear text in the html page
- Sensitive data was stored in the readable HTTP header
- Apache web server was vulnerable to multiple exploits
- Multiple user passwords were weak/crackable
- Poor access controls allowed for rampant gathering of password hashes

Executive Summary

As of the termination of penetration testing activities, our investigation discovered 21 vulnerabilities of variable severity. We exploited 7 different hosts on 10 different ports, and discovered login credentials for multiple user accounts. Existing security measures were present, and often delayed the achievement of penetration testing goals, however all goals stated in the proposed objectives were ultimately achieved. The vulnerability summary below is organized by severity and will cover responsible disclosure of the vulnerabilities we found and the exploits leveraged against them. The executive summary will cover the broad strokes of our findings and will organize findings by target type (Web App, Linux host, Windows host).

The web application was the first target and was attacked on 05/20/2024. The web application contained 4 critical vulnerabilities, 3 high risk vulnerabilities, and 2 medium risk vulnerabilities. The web application can be summarized as needing to employ user-input sanitization across the board. Multiple vulnerabilities were exploited by inserting code that was not the intended usage at various inputs on different pages on the Total Rekall site. Rekall would also drastically improve their defense posture by implementing and enforcing a user password policy that requires unique and long passwords.

Multiple Linux machines were targeted and exploited on 05/22/2024. There were 3 critical vulnerabilities, 3 high risk vulnerabilities, and 1 medium risk vulnerability exploited on Linux machines. Broadly speaking, the Linux environment at Rekall is in need of software updates. Every critical vulnerability we discovered relied on exploits that have already been patched; furthermore, each of these exploits can be leveraged by novice hackers using the free and open-source Metasploit framework. All three of the high risk vulnerabilities pertain to poor file permissions/access controls and could be remediated by deploying Role Based Access Controls (RBAC) to categorize users and files based on the needs of the organization and user type. A password policy would also improve the defense posture of the Linux environment.

The Windows environment was attacked on 05/23/2024. We discovered 1 critical vulnerability, 1 high risk vulnerability, 1 medium risk vulnerability, and 2 low risk vulnerabilities. It should be noted that we only compromised one Windows machine, however we discovered a great deal of information regarding system info and username/password combinations that may have lead to further workstation compromise if that were in-scope. The critical vulnerability and the medium risk vulnerability can both be remediated by installing OS updates (and continuing to do so in the future). The high risk vulnerability and one of the low risk vulnerabilities are a product of poor access control policy and missing firewall rules. The other low risk vulnerability can be remediated by improving access controls and obfuscating certain administrative functions like task scheduling.

In summary, the Rekall corporation penetration test achieved all assessment objectives and has yielded 21 remediation recommendations. The White Hat Group would like to commend Rekall Corporation on their decision to seek out proactive penetration testing and would like to extend an offer to provide a follow up assessment at the leisure of Rekall Corporation.

Summary Vulnerability Overview

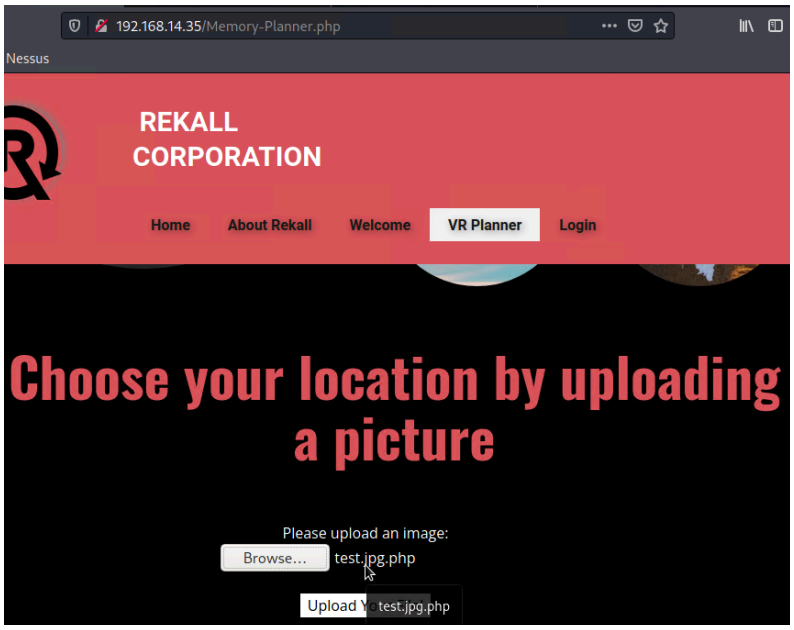
Vulnerability	Severity
Local file inclusion @ totalrekall.xyz/Memory-Planner.php	Critical
Stored cross-site scripting @ totalrekall.xyz/comments.php	Critical
Command injection @ totalrekall.xyz/networking.php	Critical
Login credentials stored in clear text within HTML page source @ totalrekall.xyz/Login.php	Critical
Remote code execution via known jsp_upload_bypass exploit @ 192.168.13.10/8080	Critical
Remote shell achieved via known apache_mod_cgi exploit @ 192.168.13.11/80	Critical
Remote shell achieved via known drupal_restws_unserialize exploit @ 192.168.13.13/80	Critical
Remote shell achieved via know SLMail exploit @ 172.22.117.20/110	Critical
Reflected cross-site scripting @ totalrekall.xyz/Memory-Planner.php	High
SQL injection @ totalrekall.xyz/Login.php	High
PHP injection @ totalrekall.xyz/souvenirs.php	High
Unauthorized access to /etc/sudoers.d @ 192.168.13.11	High
Unauthorized access to /etc/passwd @ 192.168.13.11	High
Escalated privileges to root using known sudo -u#-1 exploit @ 192.168.13.13	High
Unrestricted FTP access @ 172.22.117.20/21	High
Unauthorized directory traversal @ totalrekall.xyz/robots.txt	Medium
Brute force password attack success @ totalrekall.xyz/Login.php	Medium
Weak login credentials discovered and used to login @ 192.168.13.13	Medium
NTLM hashes discovered and cracked @ 172.22.117.20	Medium
Established persistence using schtasks exploit @ 172.22.117.20	Low
Executed remote curl requests to acquire sensitive data using stolen credentials @ 172.22.117.20	Low

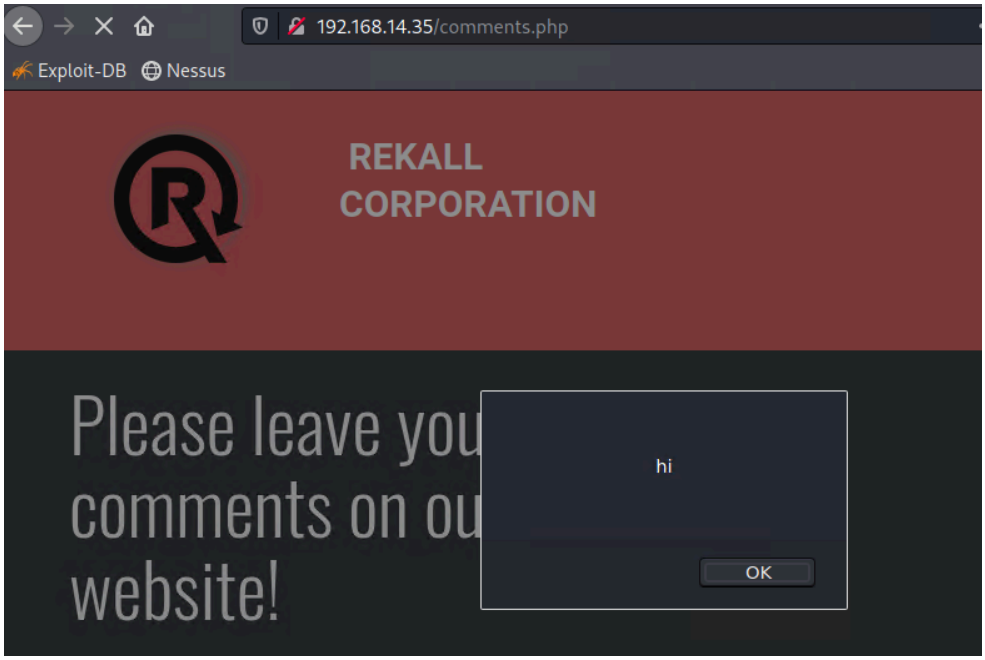
The following summary tables represent an overview of the assessment findings for this penetration test:

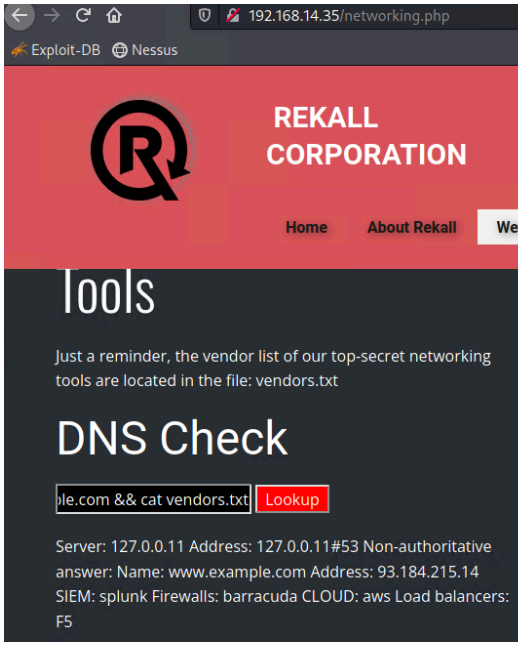
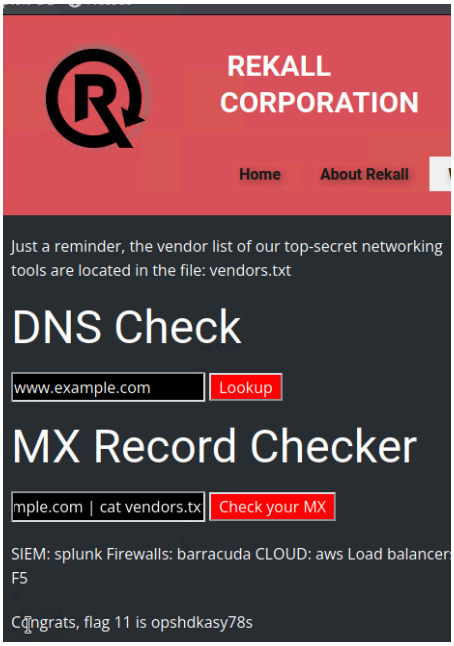
Scan Type	Total	
Hosts	192.168.13.1 192.168.13.10 192.168.13.12 192.168.13.13	192.168.13.14 192.168.14.35 172.22.117.20
Ports	21 22 80 110 5091	6001 8009 8080 10000 10001

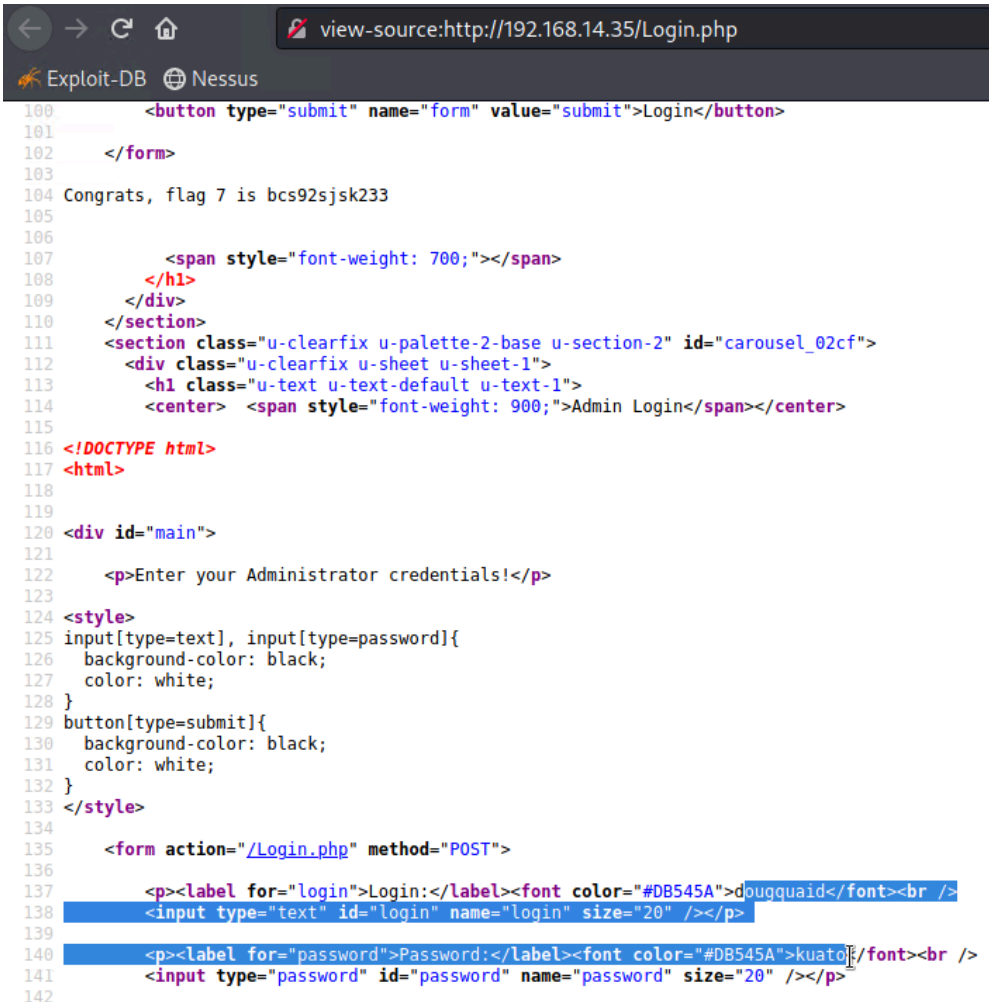
Exploitation Risk	Total
Critical	8
High	7
Medium	4
Low	2

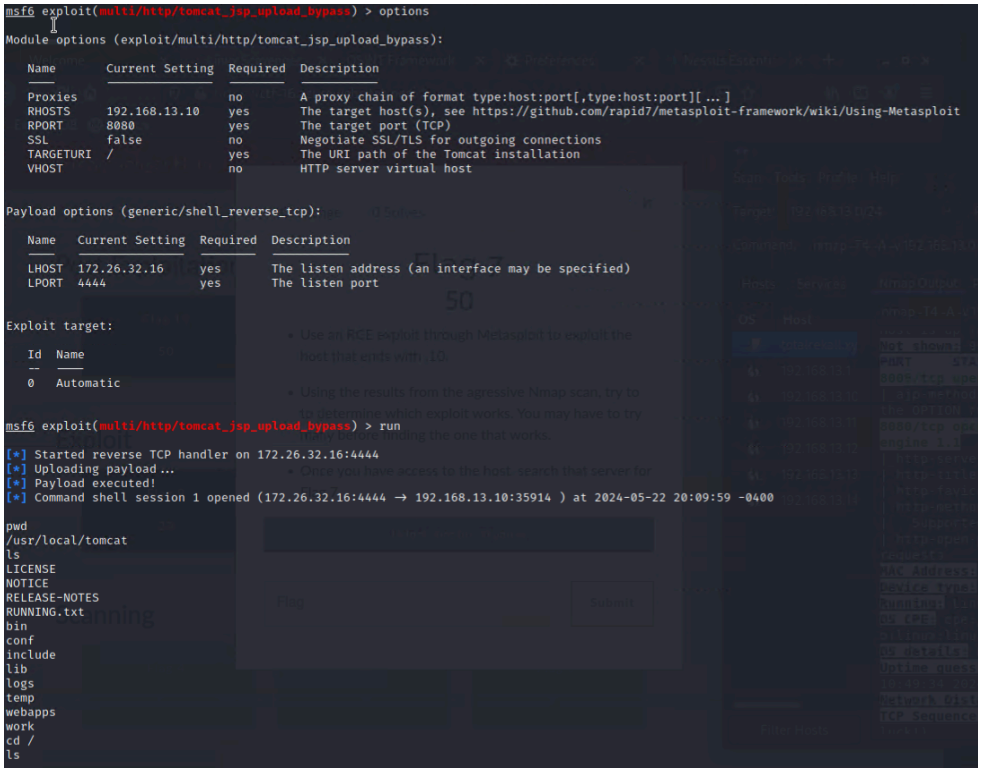
Vulnerability Findings

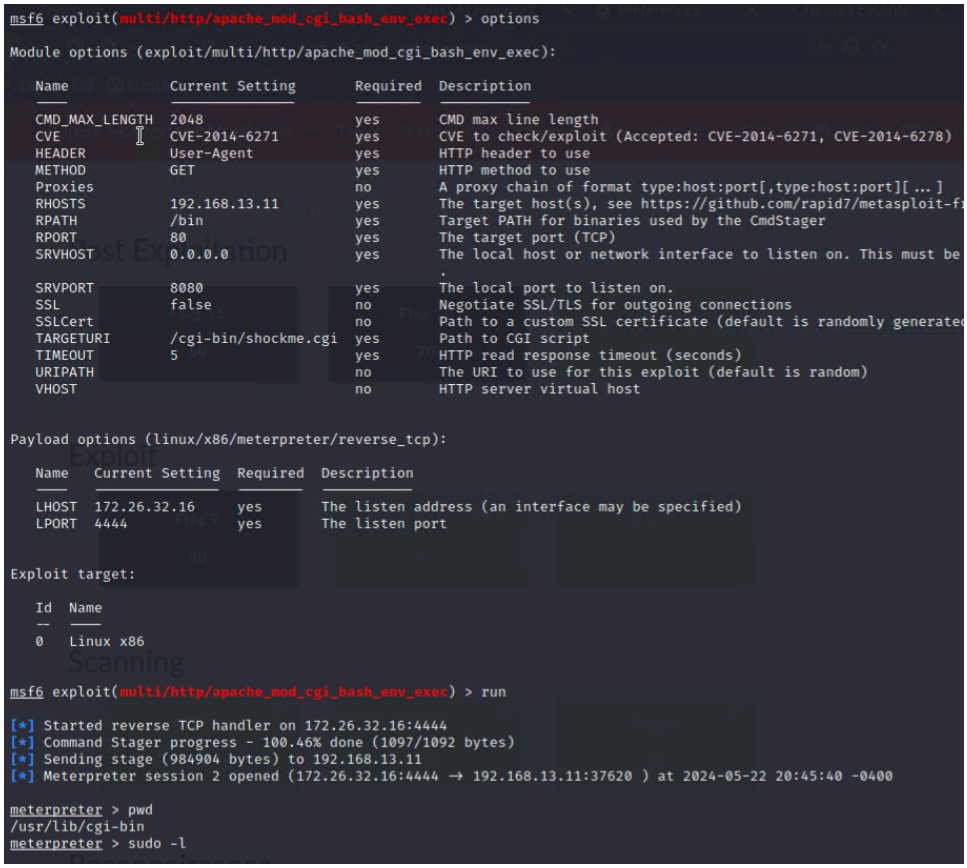
Vulnerability 1	Findings
Title	Local file inclusion
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Critical**
Description	We were able to upload a malicious .php script where the web application was intended to accept .jpg files. This was accomplished by including .jpg.php at the end of the filename to bypass the .jpg filter.
Images	
Affected Hosts	192.168.14.35 totalrekall.xyz/Memory-Planner.php
Remediation	Make the extension filter more robust by requiring the last 4 characters to be .jpg. You should also insert null bytes (%00) after the .jpg to prevent anything after that from being read. User uploads should also be encrypted, which would render any malicious code inert until decrypted. This gives you team time to analyze log data and possibly discover the malicious code before damage is done.

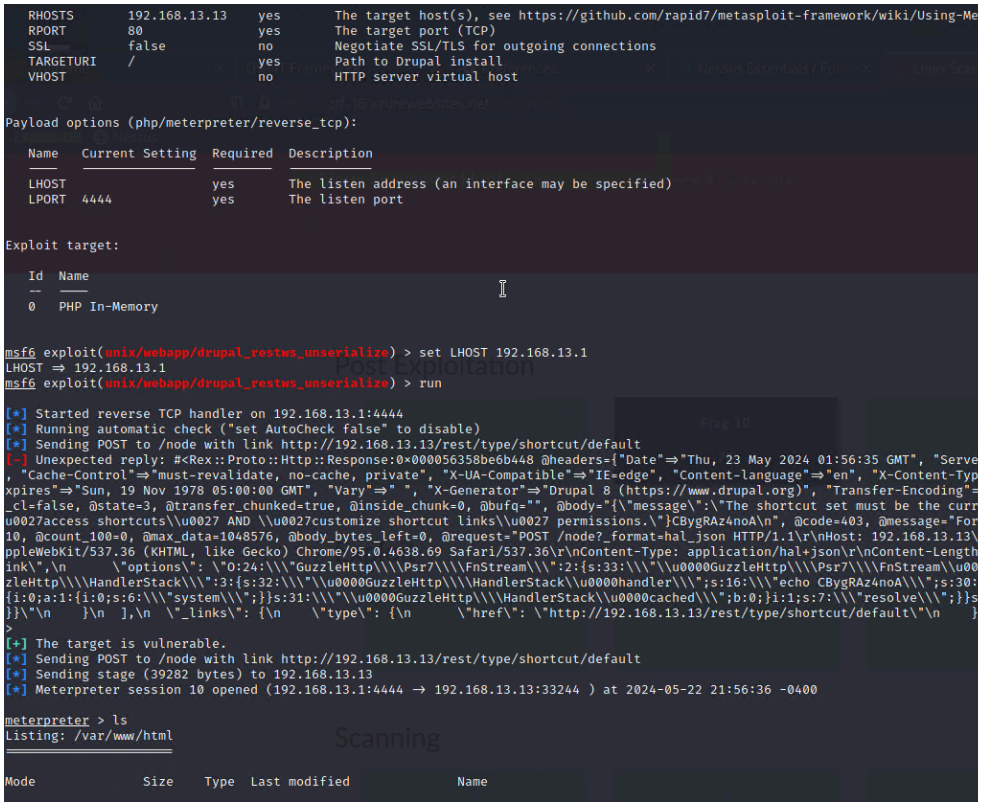
Vulnerability 2	Findings
Title	Stored cross-site scripting
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Critical**
Description	We were able to store malicious scripts in the comments section of the Web App by typing <script> alert('hi') </script> into the comments box. This would populate an alert for any user who visited the comments page.
Images	
Affected Hosts	192.168.14.35 totalrekall.xyz/comments.php
Remediation	User input should never be executable. For example, in the comments section the raw user input should be added to displayable text boxes, rather than being fed into a command.

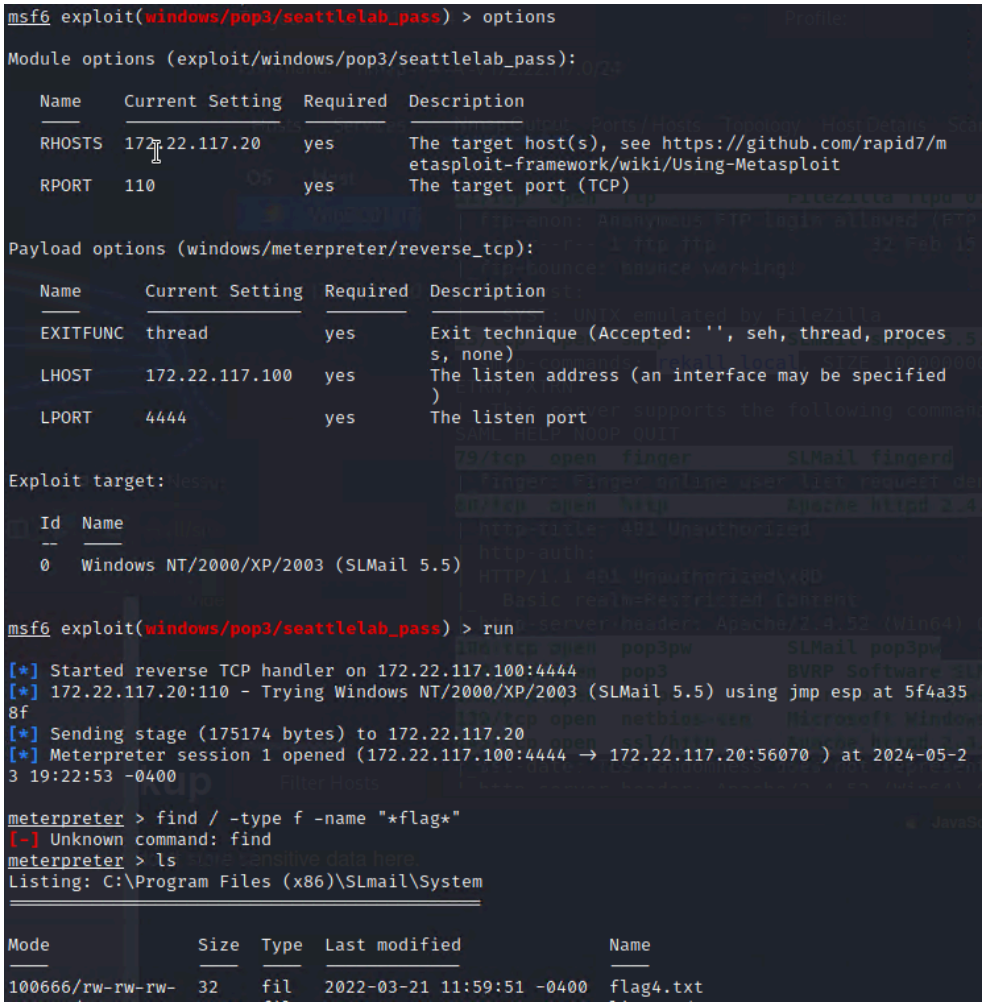
Vulnerability 3	Findings
Title	Command injection
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Critical**
Description	We were able to inject commands to exfiltrate sensitive data via the DNS Check and MX Record Checker tools on the Web App by utilizing && and | after the intended usage.
Images	 
Affected Hosts	192.168.14.35 totalrekall.xyz/networking.php
Remediation	Disallow special characters that would not be necessary for the intended use and require the user query to end with the correct extension (i/e .com).

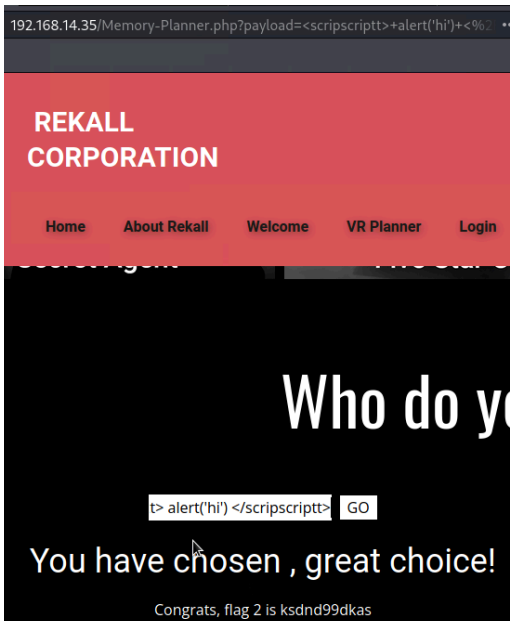
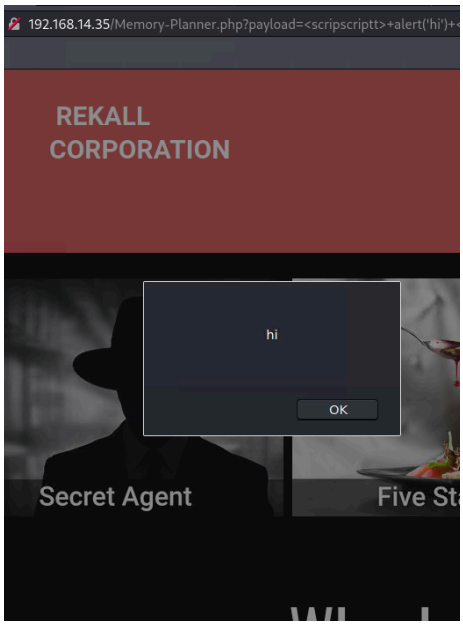
Vulnerability 4	Findings
Title	Login credentials stored in clear text within HTML page source
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Critical**
Description	User login's are validated in-browser meaning the credentials must be stored in the page source html file. This allowed us to discover valid login credentials by viewing the page source.
Images	
Affected Hosts	192.168.14.35 totalrekall.xyz/Login.php
Remediation	Log credentials cannot be securely validated within the client browser. The password should be hashed and salted, and the string consisting of a username/hash pair should be encrypted, and passed to a secure server with verifiable integrity to be compared against stored copies of the username and salted password hash.

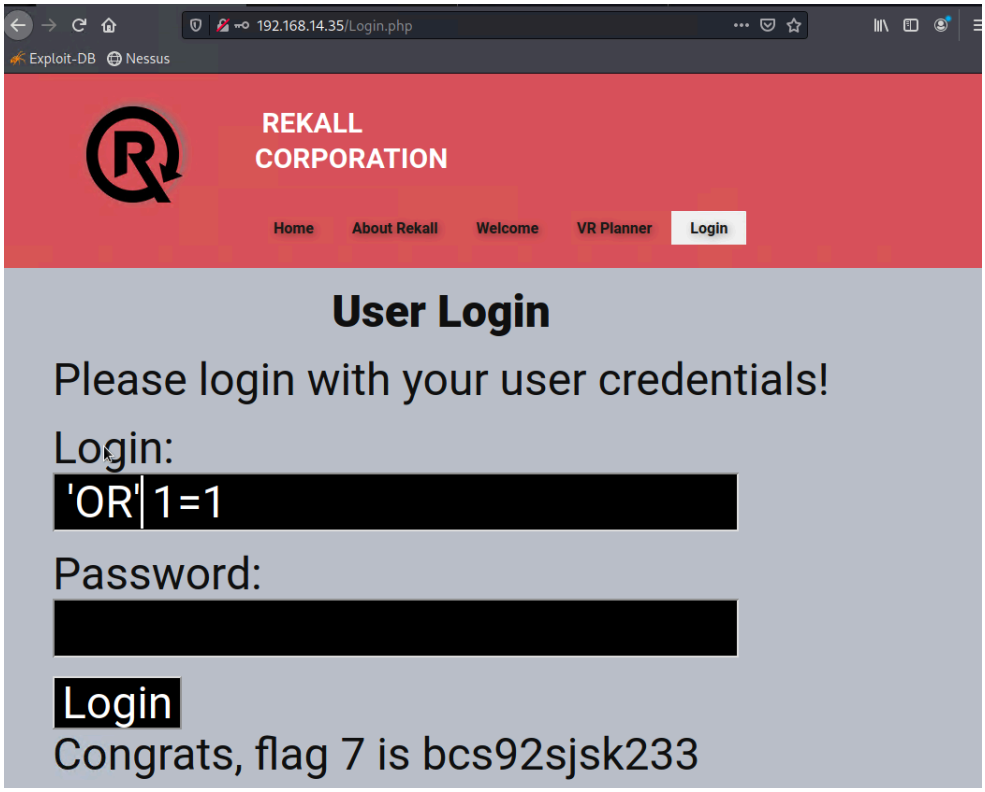
Vulnerability 5	Findings
Title	Remote code execution via known jsp_upload_bypass exploit
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Critical**
Description	We gained unauthorized remote code execution on the Apache web server by deploying the jsp_upload_bypass module in metasploit.
Images	
Affected Hosts	192.168.13.10 Port 8080
Remediation	Updating the OS of the server will close this vulnerability.

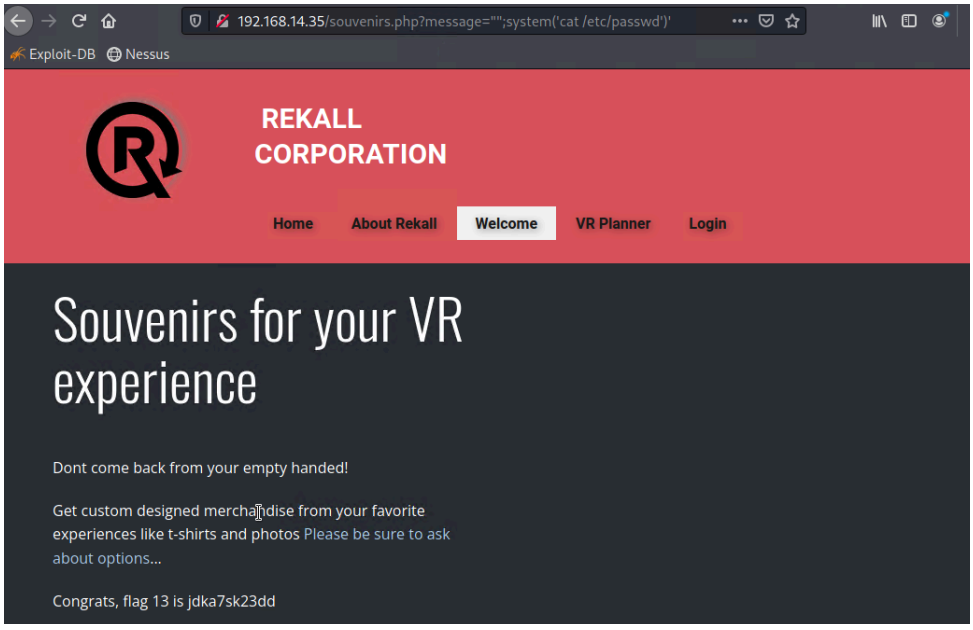
Vulnerability 6	Findings
Title	Remote shell achieved via known apache_mod_cgi_bash_env_exec exploit
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Critical**
Description	We gained an unauthorized remote shell by deploying the apache_mod_cgi_bash_env_exec module in metasploit.
Images	
Affected Hosts	192.168.13.11 Port 80
Remediation	Updating the OS of the server will close this vulnerability.

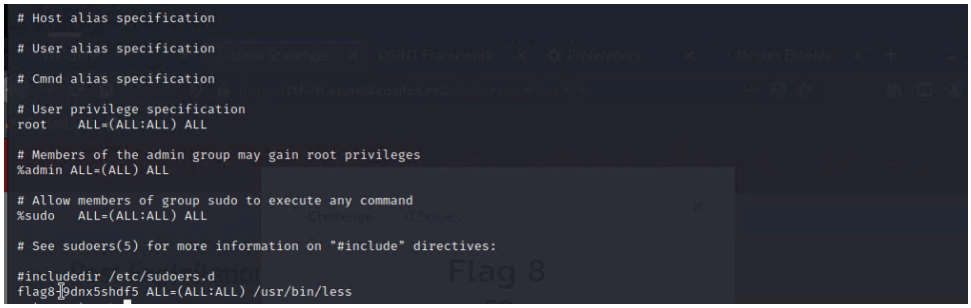
Vulnerability 7	Findings
Title	Remote shell achieved via known drupal_restws_unserialize exploit
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Critical**
Description	We gained an unauthorized remote shell by deploying the drupal_restws_unserialize module in metasploit.
Images	
Affected Hosts	192.168.13.13 Port 80
Remediation	Updating the OS of the server will close this vulnerability.

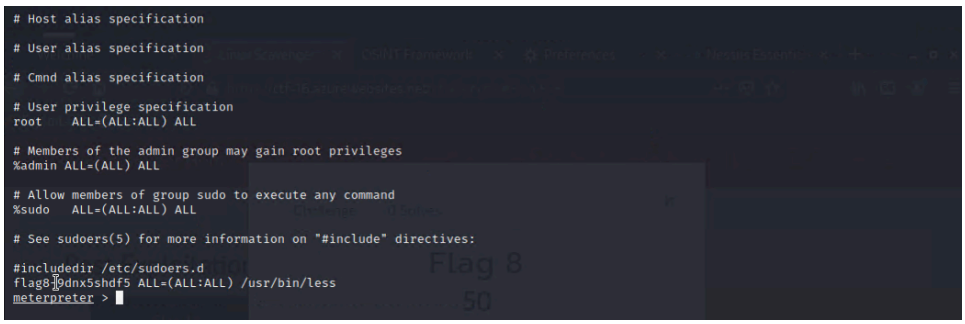
Vulnerability 8	Findings
Title	Remote shell achieved via know SLMail exploit
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	**Critical**
Description	We gained an unauthorized remote shell by deploying the seattlelab_pass module in metasploit. We used this access to exfiltrate sensitive data.
Images	
Affected Hosts	172.22.117.20 Port 110
Remediation	Updating the OS of the device will close this vulnerability.

Vulnerability 9	Findings
Title	Reflected cross-site scripting
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**High**
Description	We executed reflected cross-site scripting (XSS) by inserting <scripscriptt> alert(hi) </scripscriptt> into the box after the intended input. This vulnerability could be combined with other methods to run arbitrary code on a user's web browser.
Images	 
Affected Hosts	192.168.14.35 totalrekall.xyz/Memory-Planner.php
Remediation	User input should not be run as code.

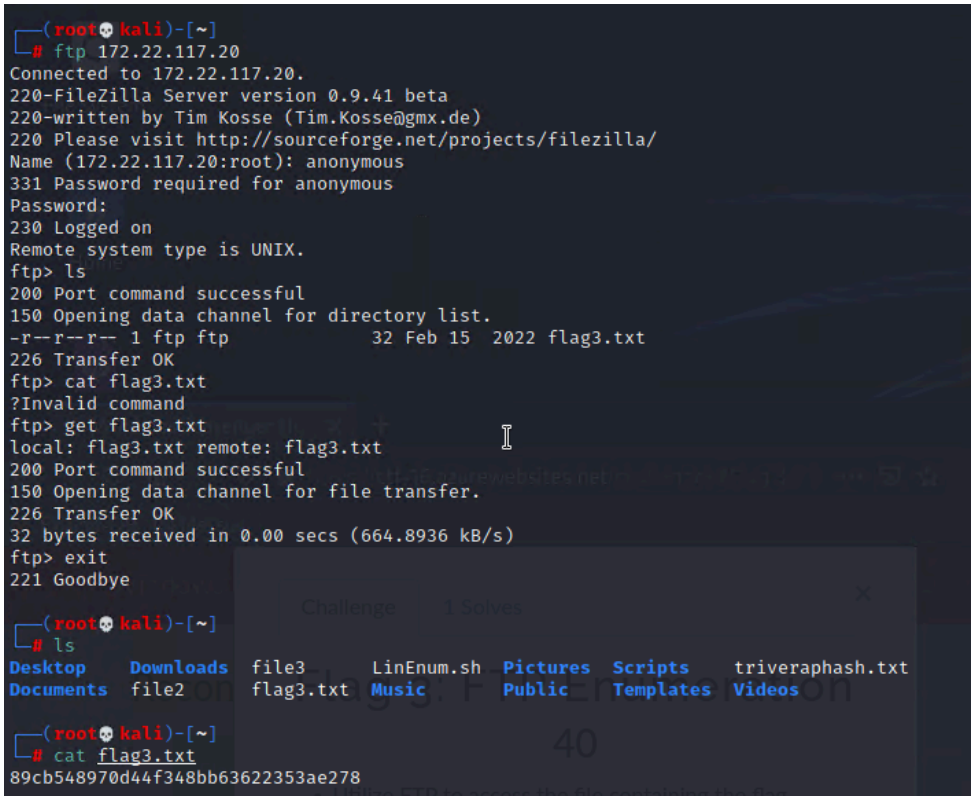
Vulnerability 10	Findings
Title	SQL injection
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	By including 'OR' 1=1 (an always true statement) we proved that the login page is susceptible to SQL Injection and may potentially disclose sensitive information such as all user names.
Images	
Affected Hosts	192.168.14.35 totalrekall.xyz/Login.php
Remediation	The input for the username field should be sanitized based on the parameters of your organization's usernames. For example, if your organization's usernames consist of letters and numbers, only those characters should be allowed in the input box for usernames. Furthermore, the authentication process shouldn't be running login credentials as code. The password should be hashed and salted, and the strings consisting of a username/hash pair should be encrypted, and passed to a secure server with verifiable integrity to be compared against stored copies of the username and salted password hash.

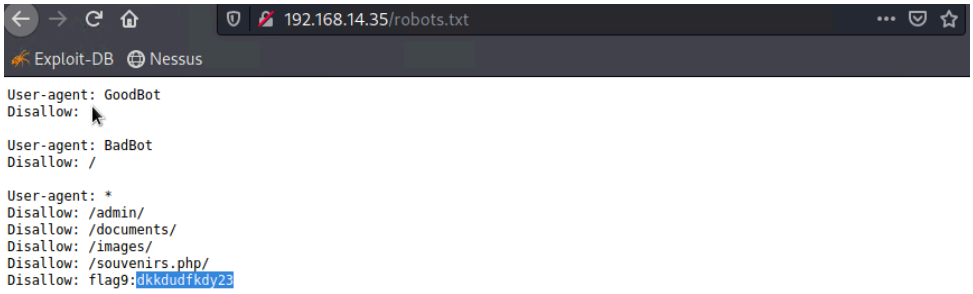
Vulnerability 11	Findings
Title	PHP injection
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**High**
Description	By adding commands to the URL line, we were able to exfiltrate sensitive information (/etc/passwd file) from the web server, potentially disclosing login usernames.
Images	
Affected Hosts	192.168.14.35 totalrekall.xyz/souvenirs.php
Remediation	Automatically inserting null bytes (%00) after souvenirs.php would prevent anything added to the end of the URL from being read. Additionally, verbose error codes should be disabled so that an attacker is not able to test inputs against the web application.

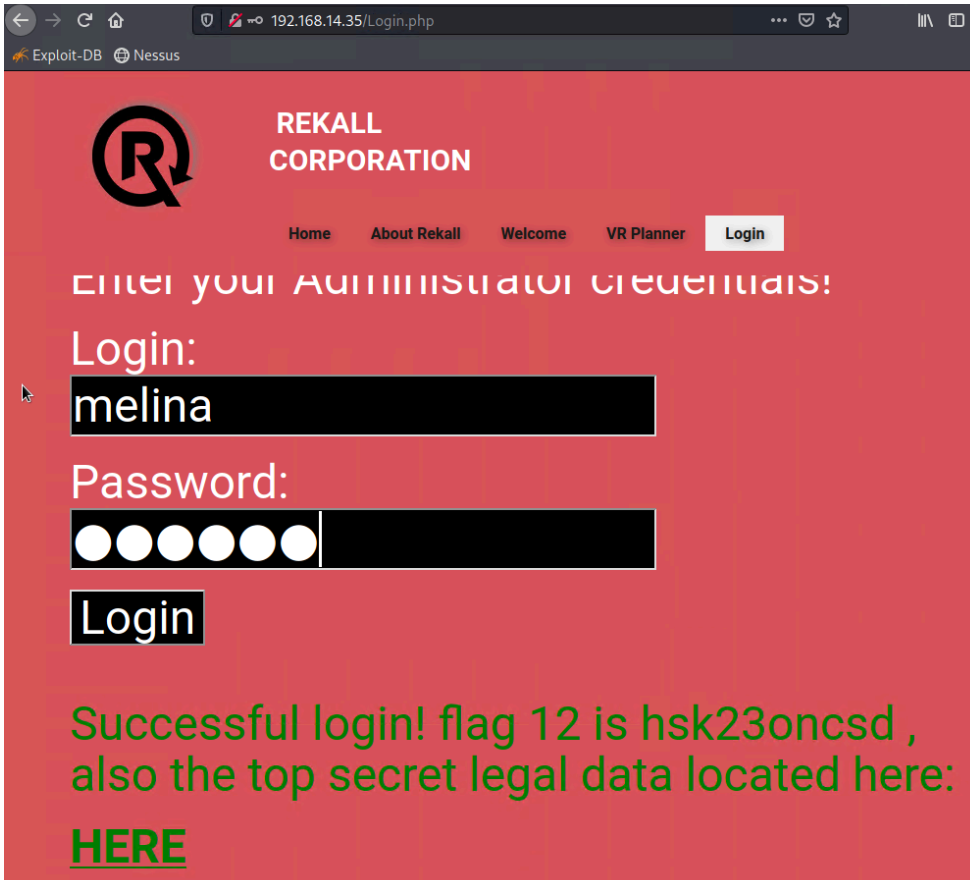
Vulnerability 12	Findings
Title	Unauthorized access to /etc/sudoers.d
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	After gaining access to the linux host via the apache_mod_cgi exploit, we were able to access the /etc/sudoers.d file with the base privileges we were granted. This could allow for privilege escalation.
Images	
Affected Hosts	192.168.13.11
Remediation	Set file permissions to prohibit access of sensitive files by low-privilege users.

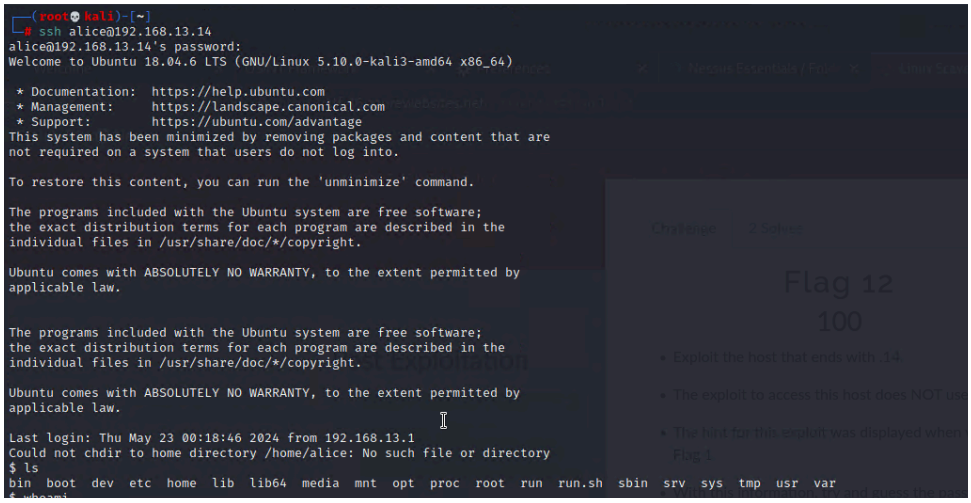
Vulnerability 13	Findings
Title	Unauthorized access to /etc/passwd
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	After gaining access to the linux host via the apache_mod_cgi exploit, we were able to access the /etc/passwd file with the base privileges we were granted. This yielded additional login info and may lead to further account compromise.
Images	
Affected Hosts	192.168.13.11
Remediation	Set file permissions to prohibit access of sensitive files by low-privilege users.

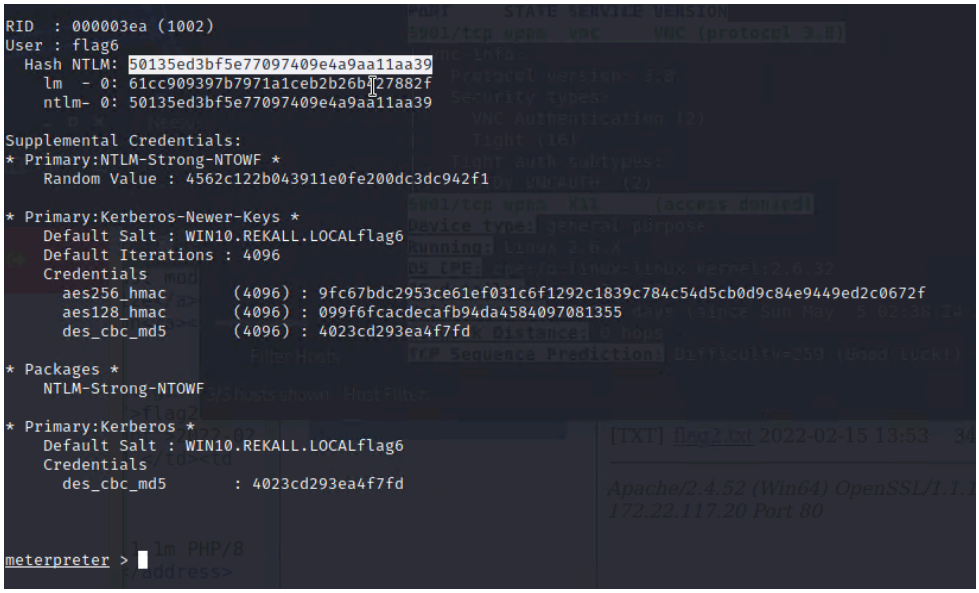
Vulnerability 14	Findings
Title	Escalated privileges to root using known sudo -u#-1 exploit
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**High**
Description	After using previously compromised login credentials to ssh into the linux host, we were able to leverage a known exploit to escalate to root privilege by issuing the command sudo -u#-1 and entering the user's password when prompted. This gave us total machine authority and could lead to sensitive data exposure and further account compromise.
Images	
Affected Hosts	192.168.13.13
Remediation	Enforcing a more stringent password policy may have prevented the compromise of this particular user account. Additionally, a Linux OS update exists to patch this vulnerability and should be installed.

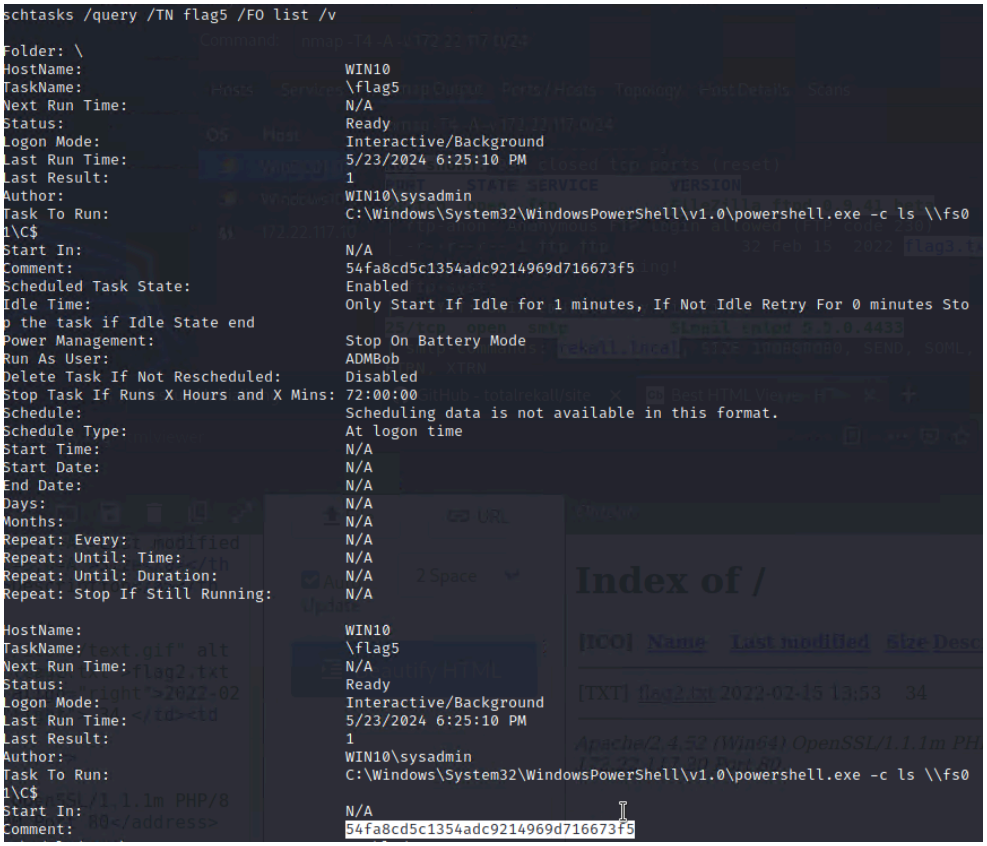
Vulnerability 15	Findings
Title	Unrestricted FTP access
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	The host was allowing un-authenticated FTP requests, and allowed us to connect and exfiltrate data without authorization.
Images	
Affected Hosts	172.22.117.20 Port 21
Remediation	Ports 20 and 21 should be closed unless absolutely critical to business function. If they must be open, anonymous access needs to be disabled and all FTP requests must be authenticated. Additionally, file permissions should be set in accordance with the principle of least privilege.

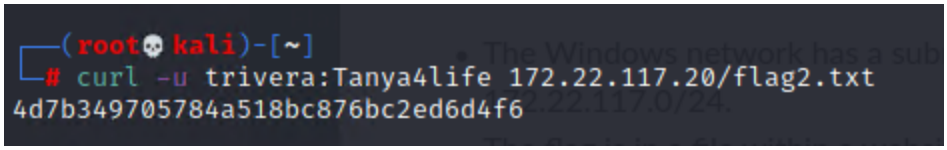
Vulnerability 16	Findings
Title	Unauthorized directory traversal
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Medium**
Description	We were able to gain unauthorized access to sensitive data by changing the url path.
Images	
Affected Hosts	192.168.14.35 totalrekall.xyz/robots.txt
Remediation	Sensitive data should be placed in separate directories with proper access controls. User's accessing these directories must be forced to authenticate prior to loading resources within restricted directories. Additionally, file names should not be displayed in the url, instead the url path should reference anonymized strings of random characters and those strings can be aliased to specific resources loaded on the webpage. This reduces the likelihood that an attacker guesses the filename of sensitive data.

Vulnerability 17	Findings
Title	Brute force password attack success
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	**Medium**
Description	Discovered valid password by performing a brute force password attack after learning valid usernames from the previously accessed /etc/passwd file.
Images	
Affected Hosts	192.168.14.35 totalrekall.xyz/Login.php
Remediation	Enforce strong password policy and employ a blacklist of known weak passwords (a file such as rockyou.txt is a good starter for the blacklist). Strong and unique passwords shouldn't be susceptible to brute force attacks since the attack largely depends of lists of previously exposed passwords.

Vulnerability 18	Findings
Title	Weak login credentials discovered and used to login
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	**Medium**
Description	The user alice has a password of alice. When combined with previous host discovery, we were able to use these credentials to sign into host.
Images	
Affected Hosts	192.168.13.13
Remediation	A password policy with technical controls preventing blacklisted passwords to enforce the policy is the best way to mitigate weak passwords usage by users.

Vulnerability 19	Findings
Title	NTLM hashes discovered and cracked
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	**Medium**
Description	After acquiring a remote shell by the previously mentioned SLMail exploit, we were able to access the kerberos NTLM password hashes. We were then able to view usernames and NTLM hashed passwords. These hashes could be cracked and this may lead to further account compromise.
Images	
Affected Hosts	172.22.117.20
Remediation	NTLM hashing is outdated and has been deprecated by Microsoft. Apply OS updates to remediate this vulnerability.

Vulnerability 20	Findings
Title	Sensitive data discovered using known schtasks exploit
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	**Low**
Description	After acquiring a remote shell by the previously mentioned SLMail exploit, we were able to schedule tasks to ensure persistent access to the compromised host.
Images	
Affected Hosts	172.22.117.20
Remediation	All scheduled tasks should be hidden in order to prevent unauthorized users from access the scheduled tasks list via schtasks.

Vulnerability 21	Findings
Title	Executed remote curl requests to acquire sensitive data using stolen credentials
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	**Low**
Description	By using previously stolen login credentials, we were able to send curl requests to retrieve sensitive data from employee workstations via HTML.
Images	
Affected Hosts	172.22.117.20
Remediation	IP filtering could be employed to prevent untrusted IP addresses from making HTTP requests from assets that do not serve web pages or other HTTP data. Additionally, enforcing strong password requirements may have prevented the compromise of this particular user account.