Cybersecurity

Penetration Test Report

Rekall Corporation

Penetration Test Report

Confidentiality Statement

This document contains confidential and privileged information from Rekall Inc. (henceforth known as Rekall). The information contained in this document is confidential and may constitute inside or non-public information under international, federal, or state laws. Unauthorized forwarding, printing, copying, distribution, or use of such information is strictly prohibited and may be unlawful. If you are not the intended recipient, be aware that any disclosure, copying, or distribution of this document or its parts is prohibited.

Table of Contents

Contact Information

Company Name	BreakoutRoom4 Penetration Testing
Contact Name	Thomas Fior
Contact Title	Pen Tester

Document History

Version	Date	Author(s)	Comments
001	4/18/2023	Thomas Fior	Action Required

Introduction

In accordance with Rekall policies, our organization conducts external and internal penetration tests of its networks and systems throughout the year. The purpose of this engagement was to assess the networks' and systems' security and identify potential security flaws by utilizing industry-accepted testing methodology and best practices.

For the testing, we focused on the following:

- Attempting to determine what system-level vulnerabilities could be discovered and exploited with no prior knowledge of the environment or notification to administrators.
- Attempting to exploit vulnerabilities found and access confidential information that may be stored on systems.
- Documenting and reporting on all findings.

All tests took into consideration the actual business processes implemented by the systems and their potential threats; therefore, the results of this assessment reflect a realistic picture of the actual exposure levels to online hackers. This document contains the results of that assessment.

Assessment Objective

The primary goal of this assessment was to provide an analysis of security flaws present in Rekall's web applications, networks, and systems. This assessment was conducted to identify exploitable vulnerabilities and provide actionable recommendations on how to remediate the vulnerabilities to provide a greater level of security for the environment.

We used our proven vulnerability testing methodology to assess all relevant web applications, networks, and systems in scope.

Rekall has outlined the following objectives:

Table 1: Defined Objectives

Objective
Find and exfiltrate any sensitive information within the domain.
Escalate privileges.
Compromise several machines.

Penetration Testing Methodology

Reconnaissance

We begin assessments by checking for any passive (open source) data that may assist the assessors with their tasks. If internal, the assessment team will perform active recon using tools such as Nmap and Bloodhound.

Identification of Vulnerabilities and Services

We use custom, private, and public tools such as Metasploit, hashcat, and Nmap to gain perspective of the network security from a hacker's point of view. These methods provide Rekall with an understanding of the risks that threaten its information, and also the strengths and weaknesses of the current controls protecting those systems. The results were achieved by mapping the network architecture, identifying hosts and services, enumerating network and system-level vulnerabilities, attempting to discover unexpected hosts within the environment, and eliminating false positives that might have arisen from scanning.

Vulnerability Exploitation

Our normal process is to both manually test each identified vulnerability and use automated tools to exploit these issues. Exploitation of a vulnerability is defined as any action we perform that gives us unauthorized access to the system or the sensitive data.

Reporting

Once exploitation is completed and the assessors have completed their objectives, or have done everything possible within the allotted time, the assessment team writes the report, which is the final deliverable to the customer.

Scope

Prior to any assessment activities, Rekall and the assessment team will identify targeted systems with a defined range or list of network IP addresses. The assessment team will work directly with the Rekall POC to determine which network ranges are in-scope for the scheduled assessment.

It is Rekall's responsibility to ensure that IP addresses identified as in-scope are actually controlled by Rekall and are hosted in Rekall-owned facilities (i.e., are not hosted by an external organization). In-scope and excluded IP addresses and ranges are listed below.

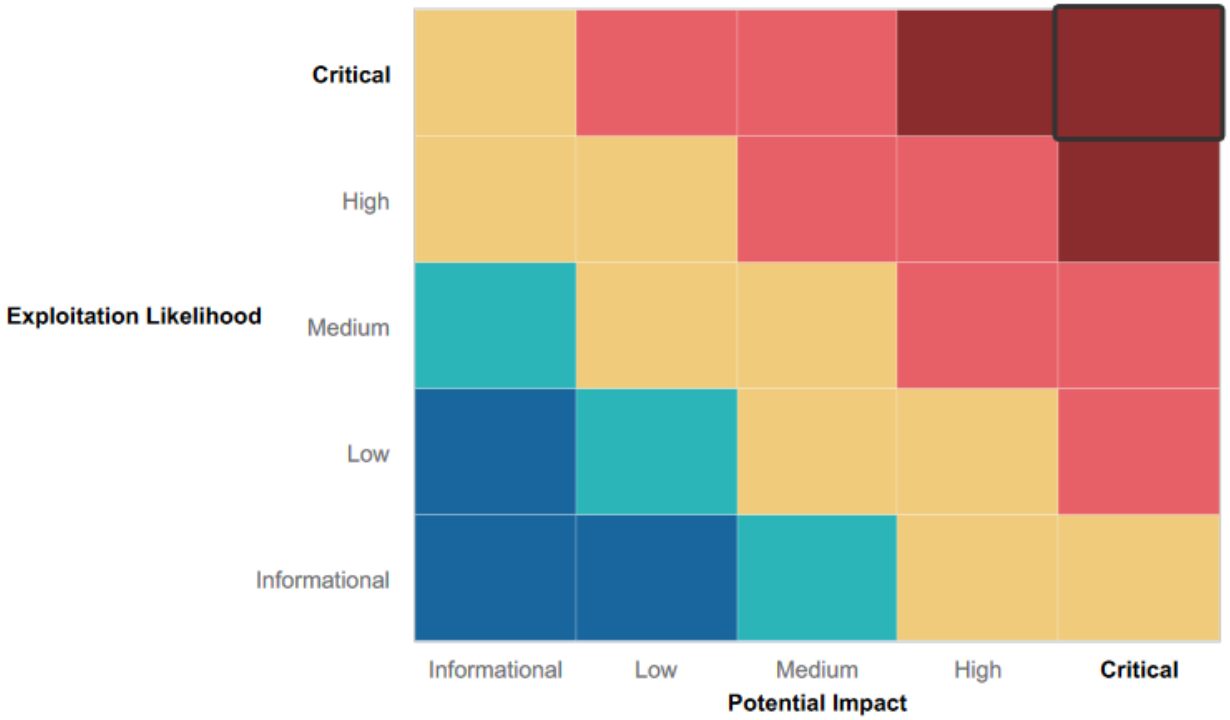
Executive Summary of Findings

Grading Methodology

Each finding was classified according to its severity, reflecting the risk each such vulnerability may pose to the business processes implemented by the application, based on the following criteria:

Critical: Immediate threat to key business processes.
High: Indirect threat to key business processes/threat to secondary business processes.
Medium: Indirect or partial threat to business processes.
Low: No direct threat exists; vulnerability may be leveraged with other vulnerabilities.
Informational: No threat; however, it is data that may be used in a future attack.

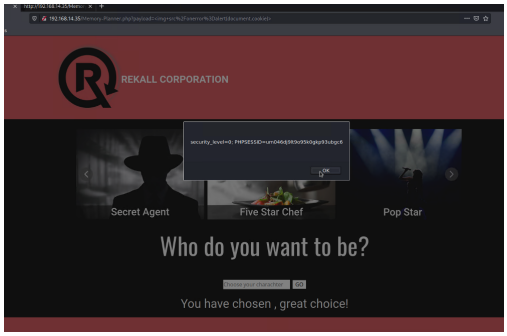
As the following grid shows, each threat is assessed in terms of both its potential impact on the business and the likelihood of exploitation:

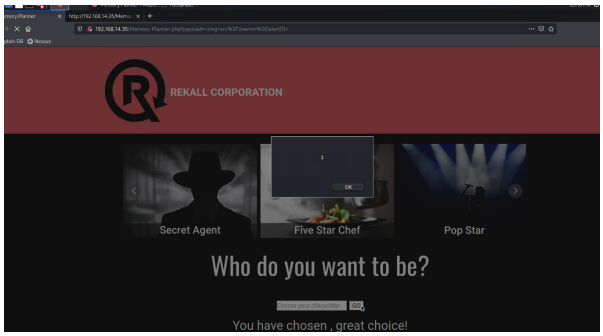


Summary of Strengths

While the assessment team was successful in finding several vulnerabilities, the team also recognized several strengths within Rekall's environment. These positives highlight the effective countermeasures and defenses that successfully prevented, detected, or denied an attack technique or tactic from occurring.

- Input Validation on VR Planner page to reduce XSS Payload exploits





Summary of Weaknesses

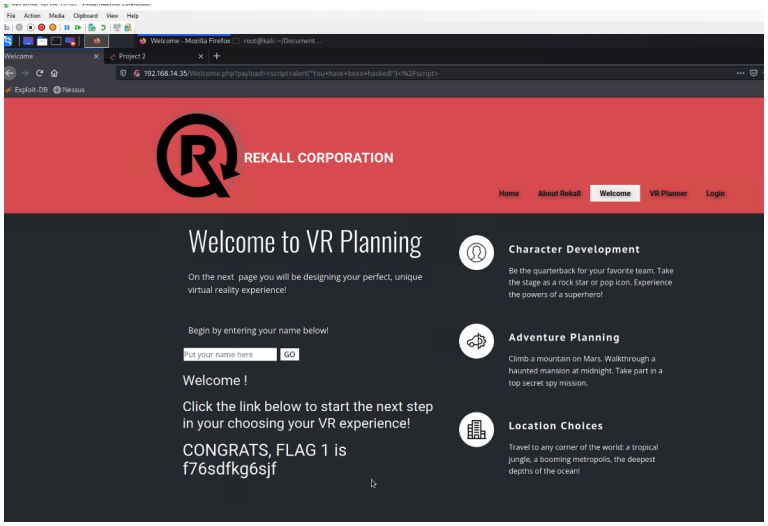
We successfully found several critical vulnerabilities that should be immediately addressed in order to prevent an adversary from compromising the network. These findings are not specific to a software version but are more general and systemic vulnerabilities.

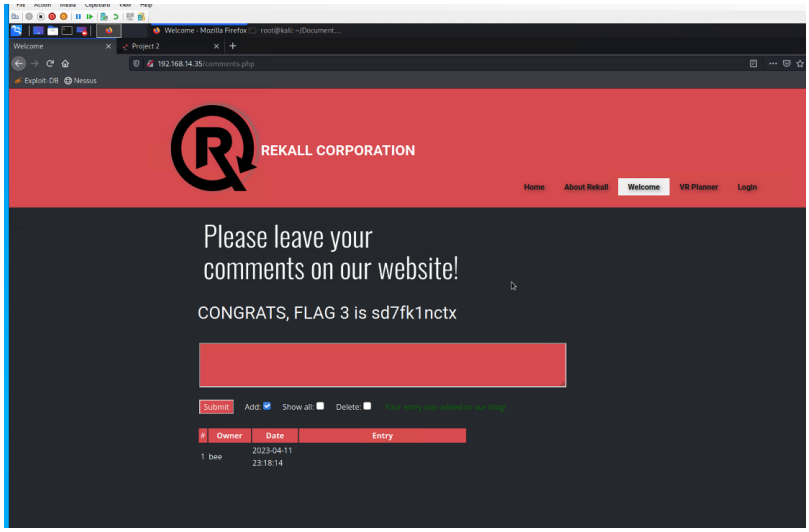
- XSS or Cross Site Scripting Reflected Vulnerabilities
- XSS or Cross Site Scripting Stored Vulnerabilities
- Local File Inclusion Vulnerabilities
- SQL Injection Vulnerabilities
- Sensitive Information such as usernames and passwords in viewable Source Code
- Command Injection Vulnerabilities
- Weak Certificate
- Brute Force Login was successful twice due to weak passwords
- Bash is outdated and is Exploitable
- SLMail Service is outdated and is Exploitable

Executive Summary

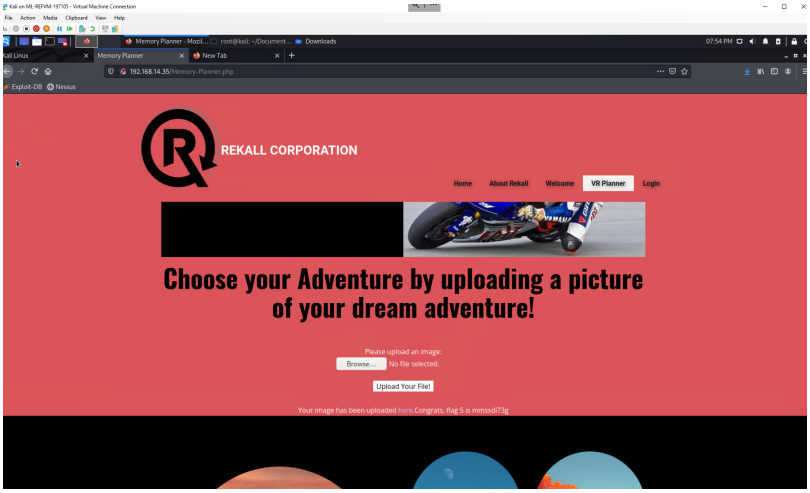
Day 1:Web Application

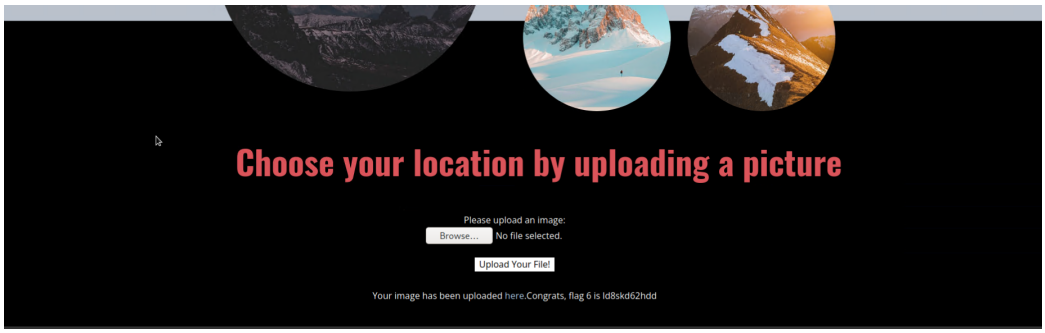
We began by attempting an XSS Injection for a pop up on the Welcome Page and the Comment Posting space, and it was successful on both pages as shown below:



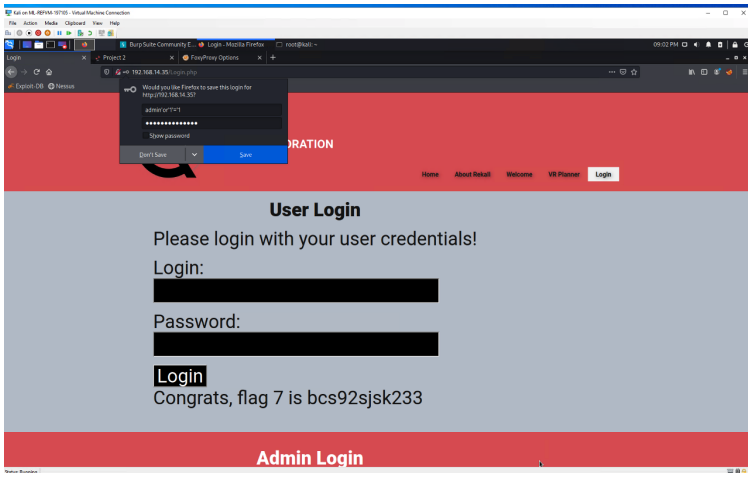


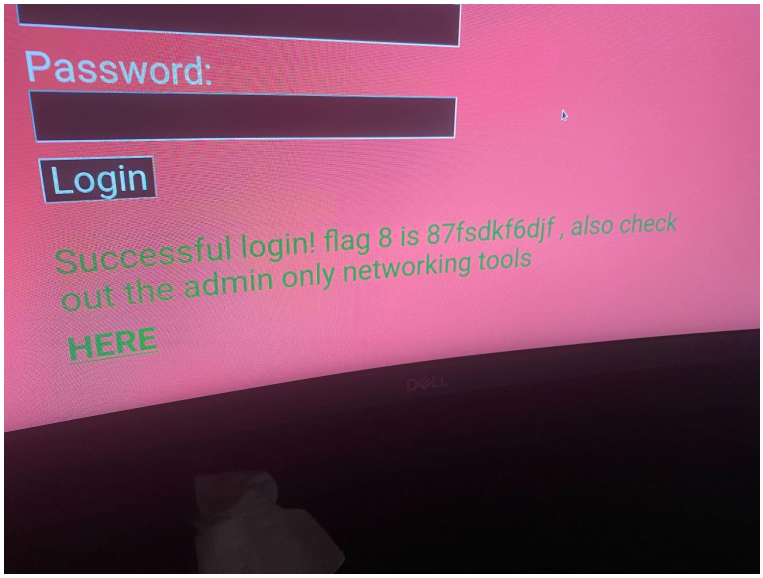
We then attempted Local File Inclusion on the VR Planner in both the Adventure and Location “Browse…” tabs and were successful on both attempts as seen below:



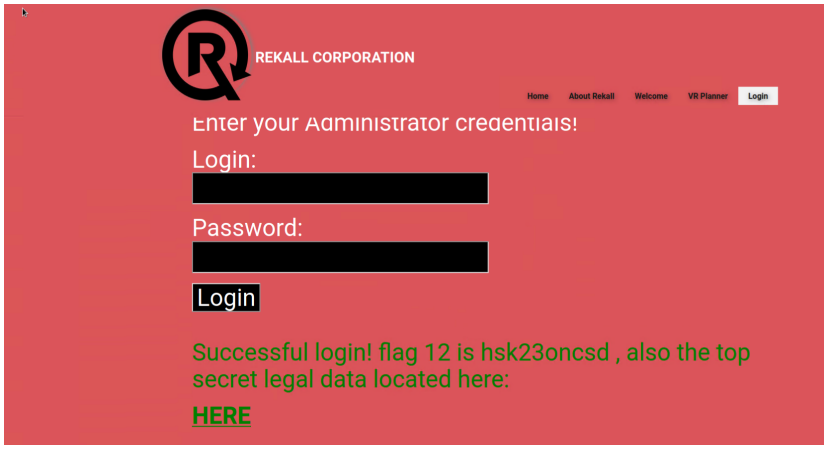


We were then able to apply an SQL Injection, a Brute Force Attack on the admin user, and view the pages source code on the login page which contained login credentials to force logins as shown below:

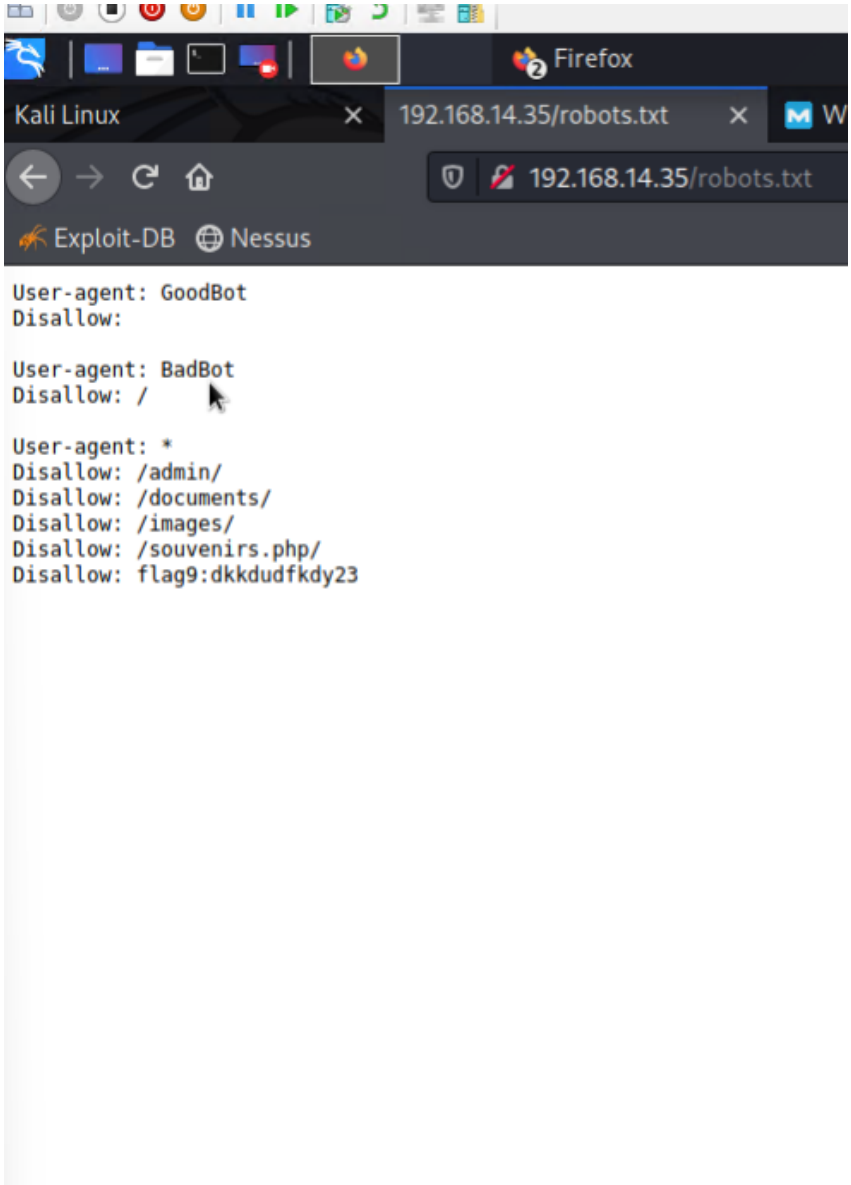


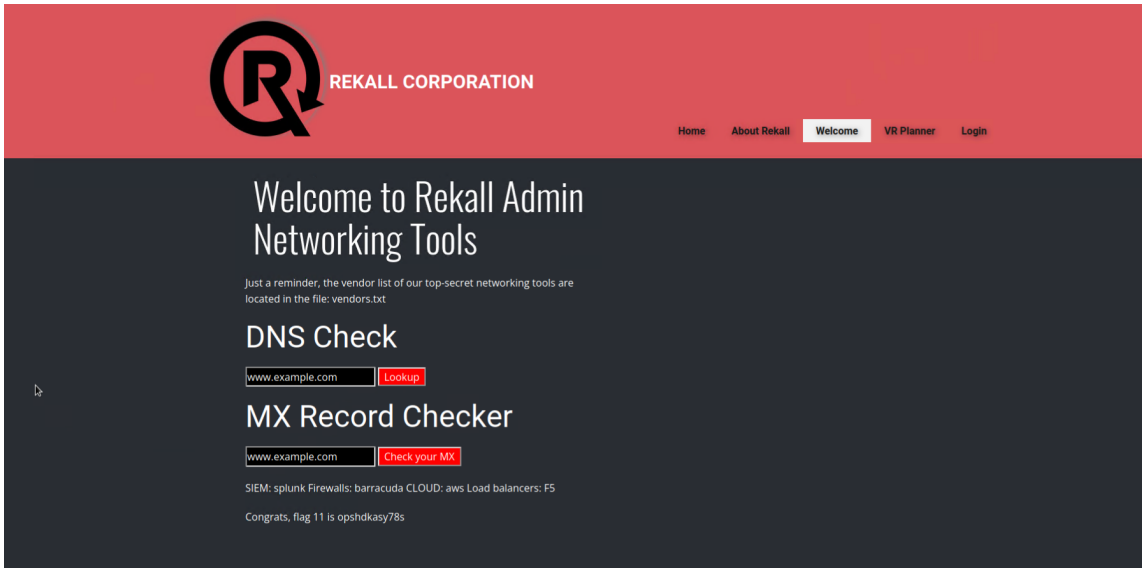


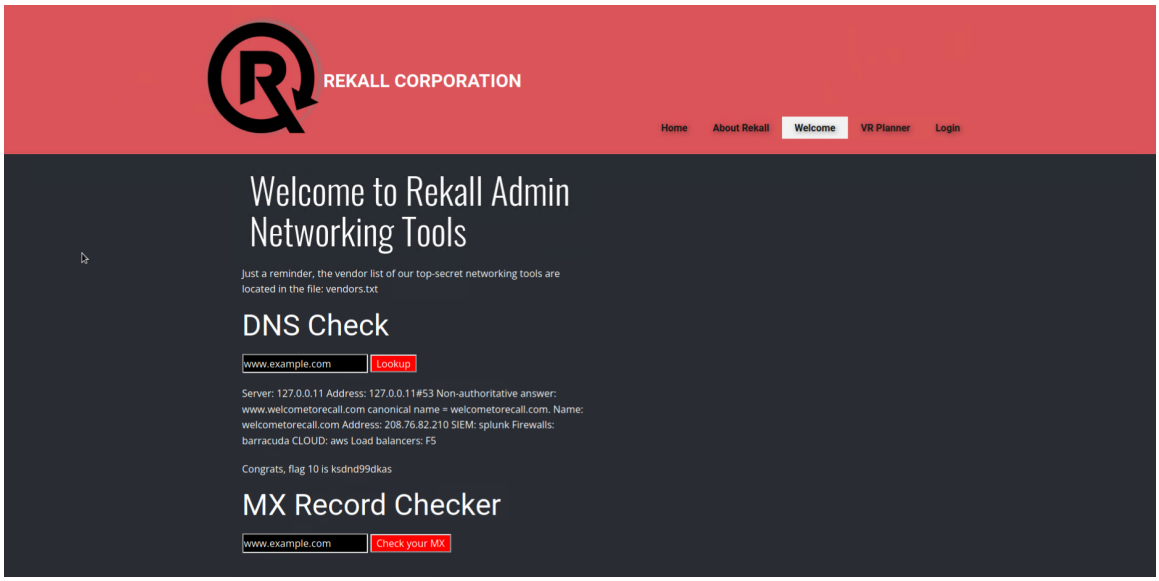

```
126   background-color: black;
127   color: white;
128 }
129 button[type=submit]{
130   background-color: black;
131   color: white;
132 }
133 </style>
134
135     <form action="/Login.php" method="POST">
136
137         <p><label for="login">Login:</label><font color="#DB545A">dougquaid</font><br />
138         <input type="text" id="login" name="login" size="20" /></p>
139
140         <p><label for="password">Password:</label><font color="#DB545A">kuato</font><br />
141         <input type="password" id="password" name="password" size="20" /></p>
142
143         <button type="submit" name="form" value="submit" background-color="black">Login</button>
144
145     </form>
146
147     </br >
148
149 </div>
150
151
```



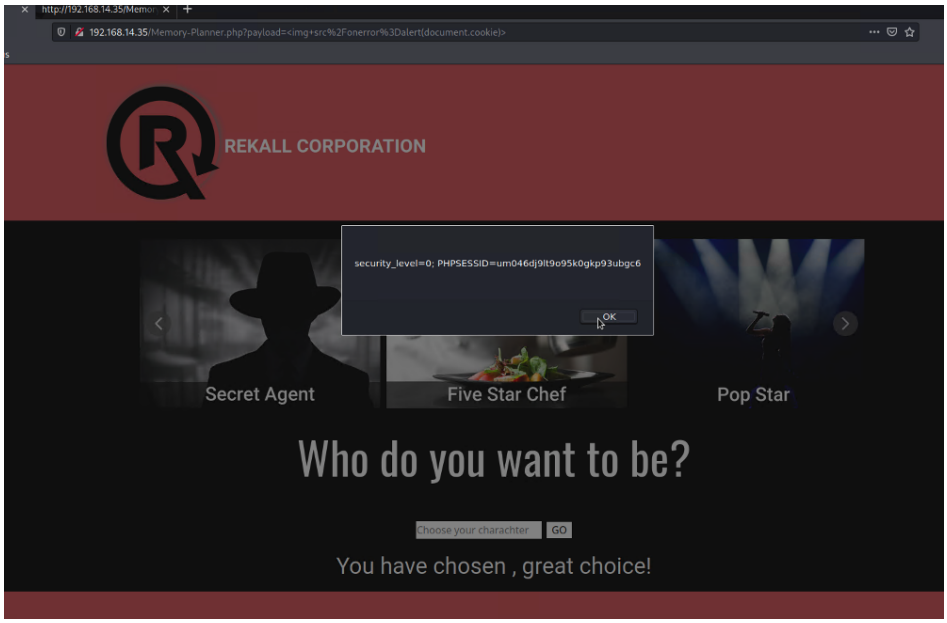
We were then able to expose sensitive information such as the web crawls and roots by adding the robots.txt file into the URL, and we were also able to perform Command Injection to provide us with the vendors.txt file in both the DNS and MX (input validation enabled) fields as seen below:

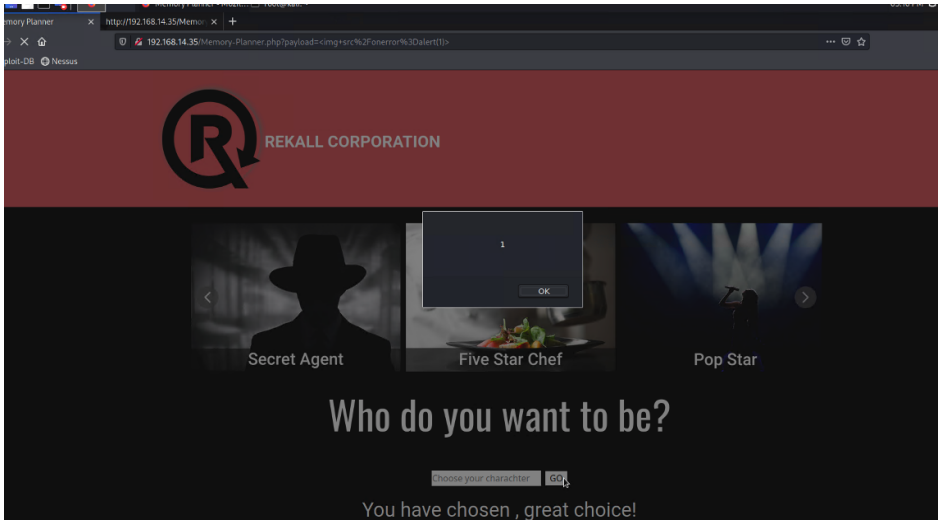






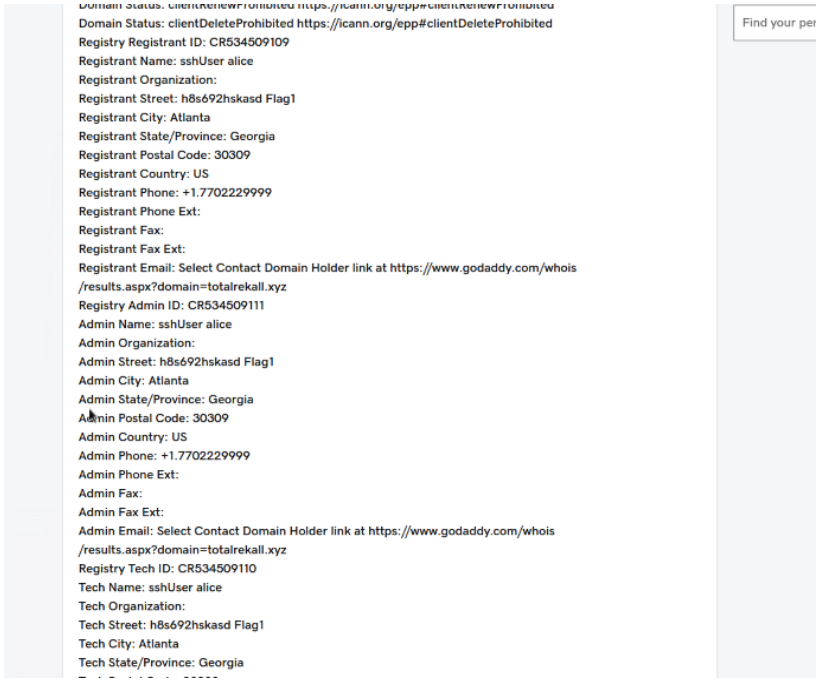
We did encounter a hold up with attempting an XSS Payload on the VR Planner Page, pop ups were initiated to bypass the input validation, but no flag was revealed. See below:

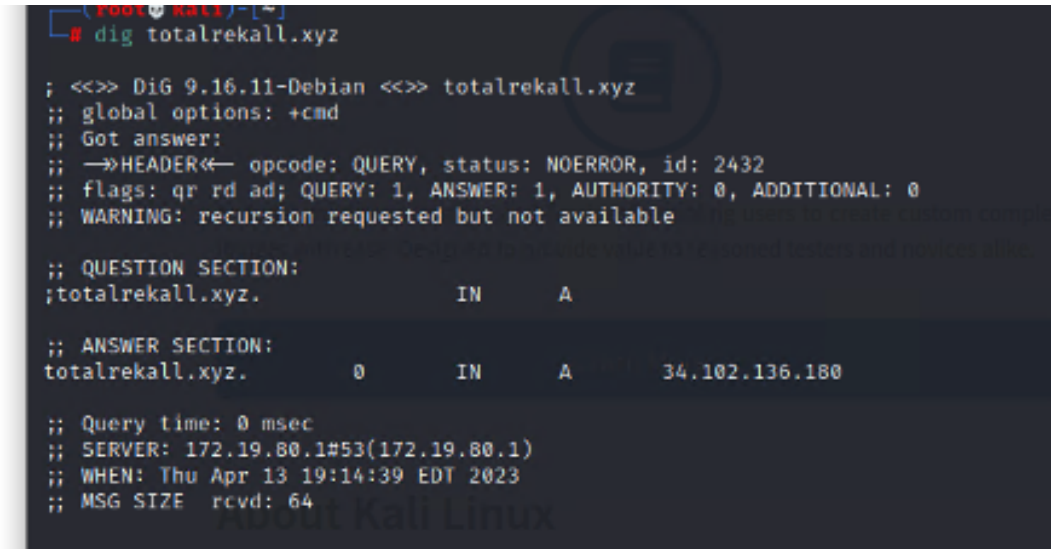


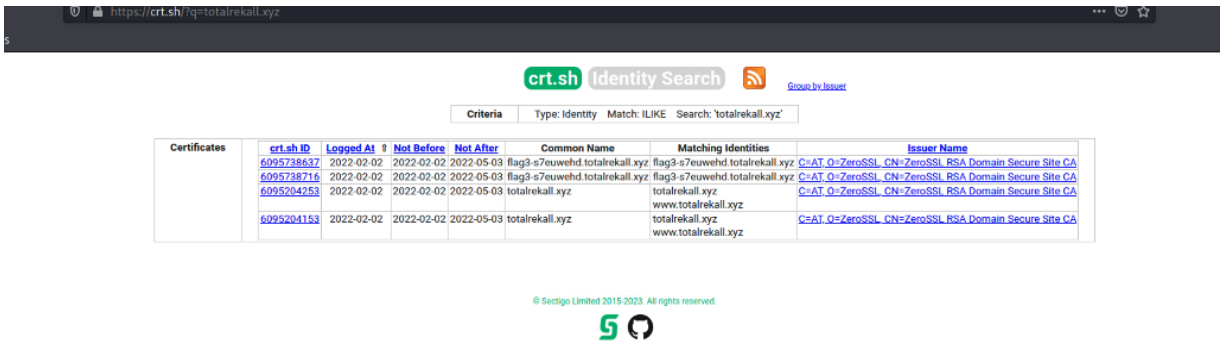


Day 2:Linux

During our Recon Phase on Day 2 we performed the following; whois command against totalrekall.xyz, dig against totalrekall.xyz, and an SSL crt.sh lookup on totalrekall.xyz as shown below:



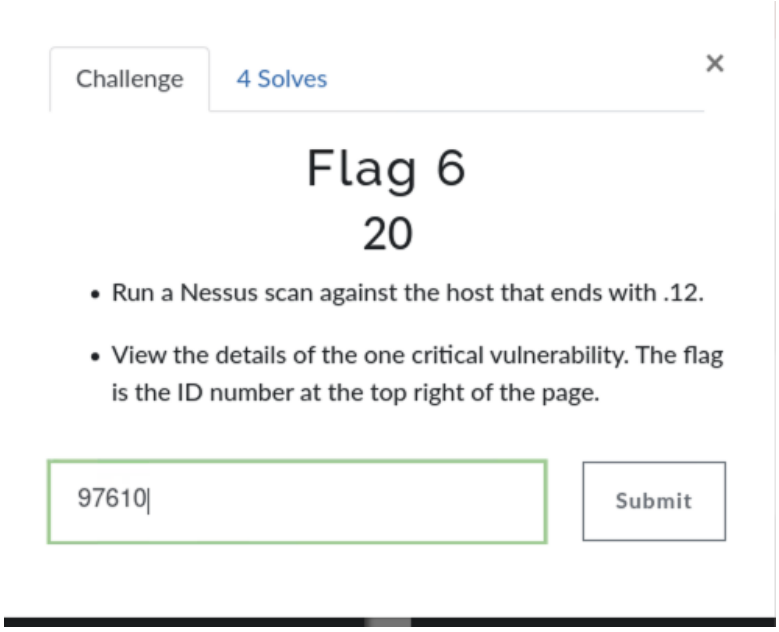




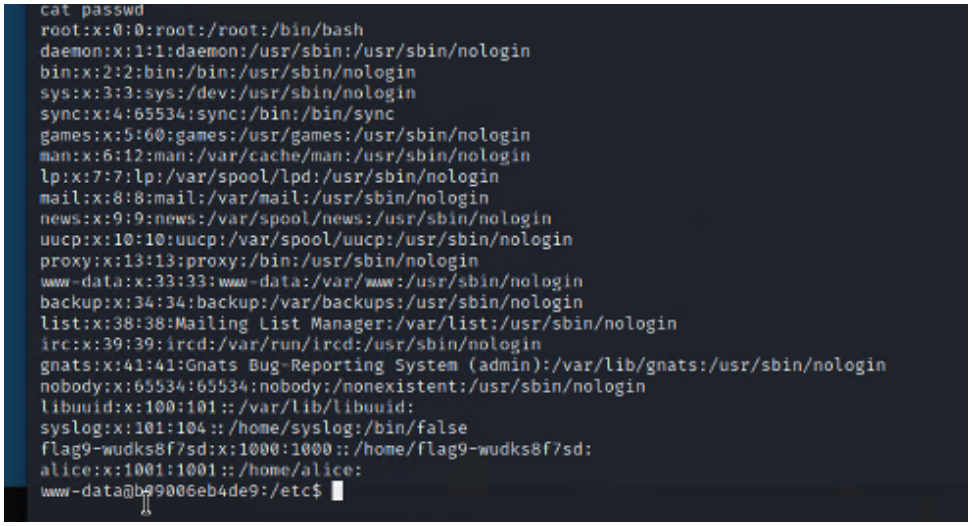
During our scanning phase of Day 2 we ran an nmap scan against the IP range of 192.168.13.0/24, an aggressive scan against 192.168.13.0/24 to find an IP running Drupal (192.168.13.13), and a Nessus Scan against the IP of 192.168.13.12 to find vulnerabilities. See below:

```
┌──(root💀kali)-[~]
└─# nmap -sV 192.168.13.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2023-04-13 19:27 EDT
Nmap scan report for 192.168.13.10
Host is up (0.0000080s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
8009/tcp open  ajp13   Apache Jserv (Protocol v1.3)
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 02:42:C0:A8:0D:0A (Unknown)

Nmap scan report for 192.168.13.11
Host is up (0.0000080s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
MAC Address: 02:42:C0:A8:0D:0B (Unknown)

Nmap scan report for 192.168.13.12
Host is up (0.0000080s latency).
Not shown: 999 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 02:42:C0:A8:0D:0C (Unknown)

Nmap scan report for 192.168.13.13
Host is up (0.0000080s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.25 ((Debian))
MAC Address: 02:42:C0:A8:0D:0D (Unknown)

Nmap scan report for 192.168.13.14
Host is up (0.0000080s 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)
MAC Address: 02:42:C0:A8:0D:0E (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.13.1
Host is up (0.00008[I]80s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE    SERVICE          VERSION
5901/tcp  open     vnc              VNC (protocol 3.8)
6001/tcp  open     X11              (access denied)
10000/tcp filtered snet-sensor-mgmt
10001/tcp filtered scp-config

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (6 hosts up) scanned in 43.22 seconds
```

```
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-generator: Drupal 8 (https://www.drupal.org)
|_http-title: Home | Drupal CVE-2019-6340
MAC Address: 02:42:C0:A8:0D:0D (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
```



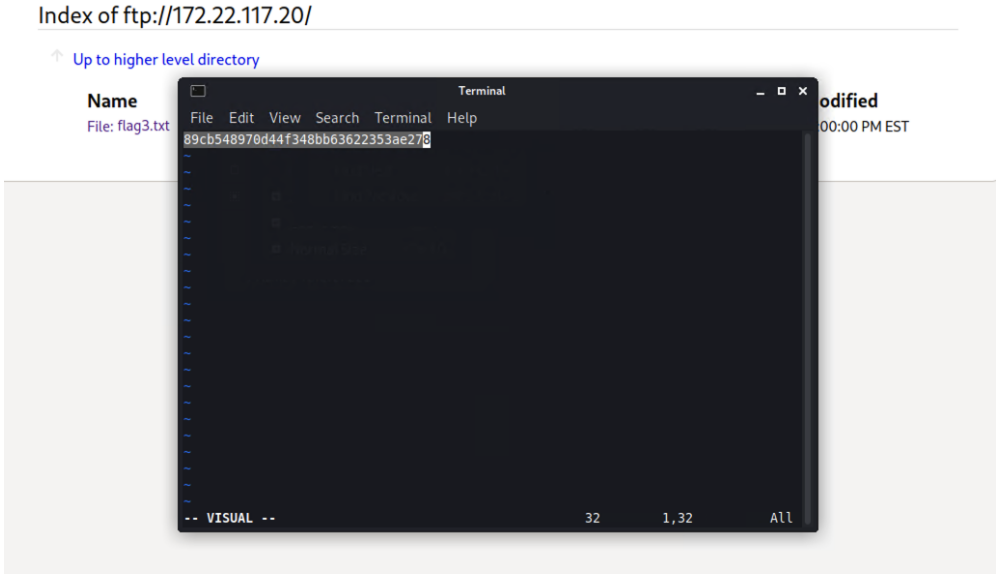
During the Exploitation and Post Exploitation Phases of Day 2 we were able to use Metasploit to exploit Bash on host .11 to gain access and view the etc/passwd file, and we were able to brute force ssh into host .14 as user:alice password:alice as shown below:

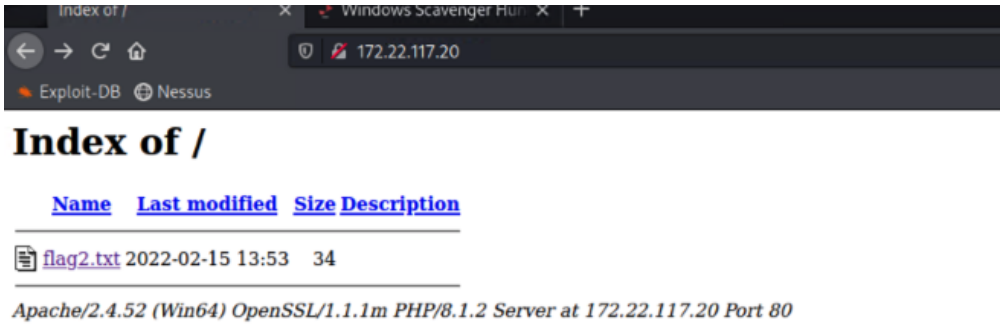


```
┌──(root💀kali)-[/]
└─# ssh alice@192.168.13.14
alice@192.168.13.14's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.10.0-kali3-amd64 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Could not chdir to home directory /home/alice: No such file or directory
```

Day 3:Windows

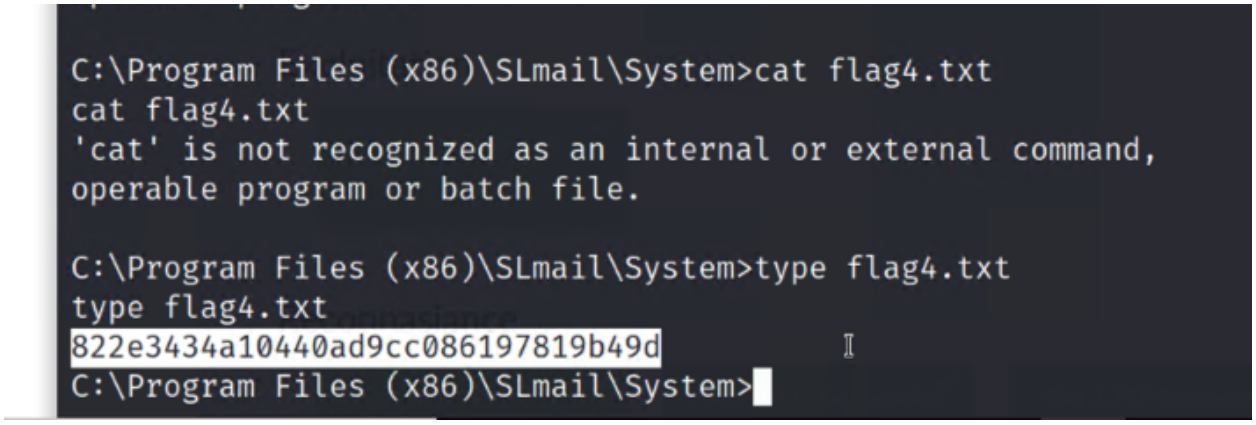
During our Recon Phase of Day 3 we started by searching for github repositories of totalrekall, we used our findings to log in as user:trivera password:Tanya4life on the IP 172.22.117.20, and then used FTP Enumeration on 172.22.117.20 IP as shown below:



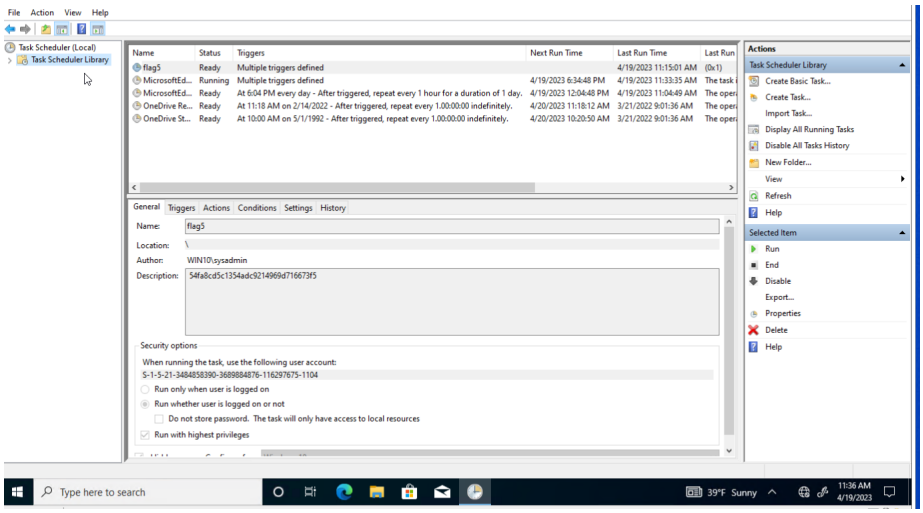


Vulnerability 20 Findings Title Port Scan of Subnet Type (Web App / Linux OS / WIndows OS) Web App Risk Rating Critical Description Using credentials gained from Github repo to login, there was a single file there named flag2.txt containing the flag Method/Payload to Exploit:
 Nmap 172.22.117.0/24 172.22.117.20 has port 80 open Opened 172.22.117.20 in a web browser Provide credentials from Flag 1 (trivera Tanya4life) to log in File flag2.txt is located in root directory Image

During our Exploitation Phase of Day 3 we had used Metasploit to gain entry through a vulnerability in the SLMail Service as shown below:



During our Post Exploitation Phase of Day 3 we had accessed the task scheduler in the exploited machine to escalate privileges, we gathered all hashed passwords, and enumerated files amongst accessible users as shown below:




```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > run post/windows/gather/hashdump

[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY 5746a193a13db189e63aa2583949573f...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hints...

No users with password hints on this system

[*] Dumping password hashes...


Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:6c49ebb29d6750b9a34fee28fadb3577:::
sysadmin:1001:aad3b435b51404eeaad3b435b51404ee:1e09a46bffe68a4cb738b0381af1dc96:::
flag6:1002:aad3b435b51404eeaad3b435b51404ee:50135ed3bf5e77097409e4a9aa11aa39:::
```


```
 Directory of C:\Users\Public\Documents

02/15/2022  03:02 PM    <DIR>          .
02/15/2022  03:02 PM    <DIR>          ..
02/15/2022  03:02 PM                32 flag7.txt
               1 File(s)             32 bytes
               2 Dir(s)   3,414,077,440 bytes free

C:\Users\Public\Documents>type flag7.txt
type flag7.txt
6fd73e3a2c2740328d57ef32557c2fdc
C:\Users\Public\Documents>
```

Summary Vulnerability Overview

Vulnerability	Severity
XSS Reflected	Medium
XSS Stored	Medium
Local File Inclusion	High
SQL Injection	Critical
Sensitive Information in Page Source Code	High
Command Injection	High
Certificate	Medium
Brute Force Password Guessing Attack	Critical
Bash Apache ShellShock	Critical
Brute Force SSH Login	Critical
SLMail Service	Critical

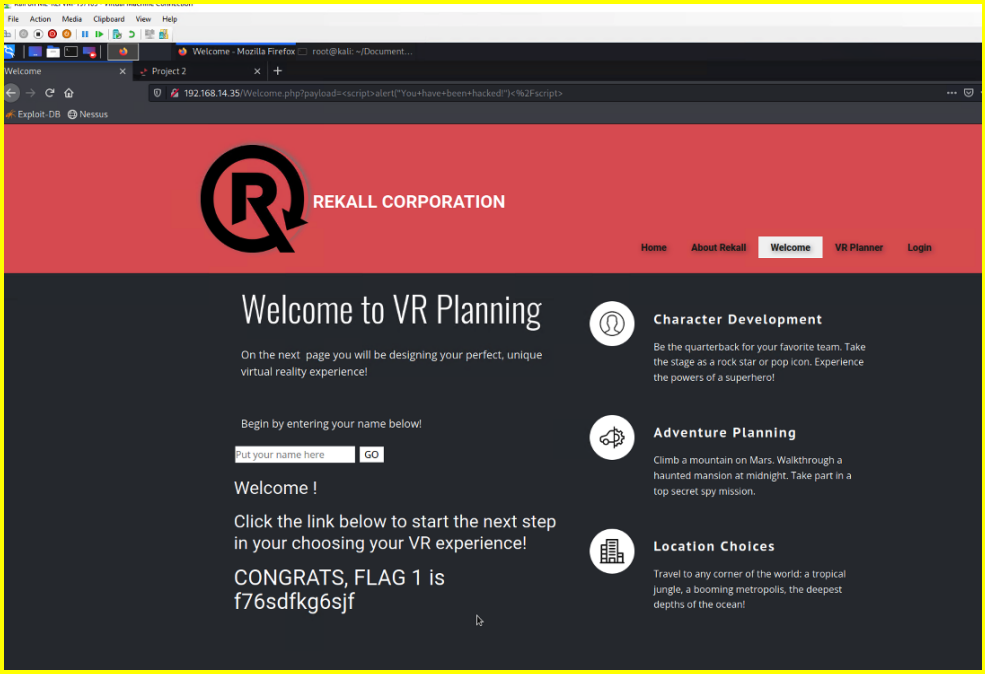
The following summary tables represent an overview of the assessment findings for this penetration test:

Scan Type	Total
Hosts	5 (172.22.117.20, 172.22.117.10, 192.168.13.11, 192.13.14, 192.168.14.35)
Ports	3 (Ports 80, 21, 22)

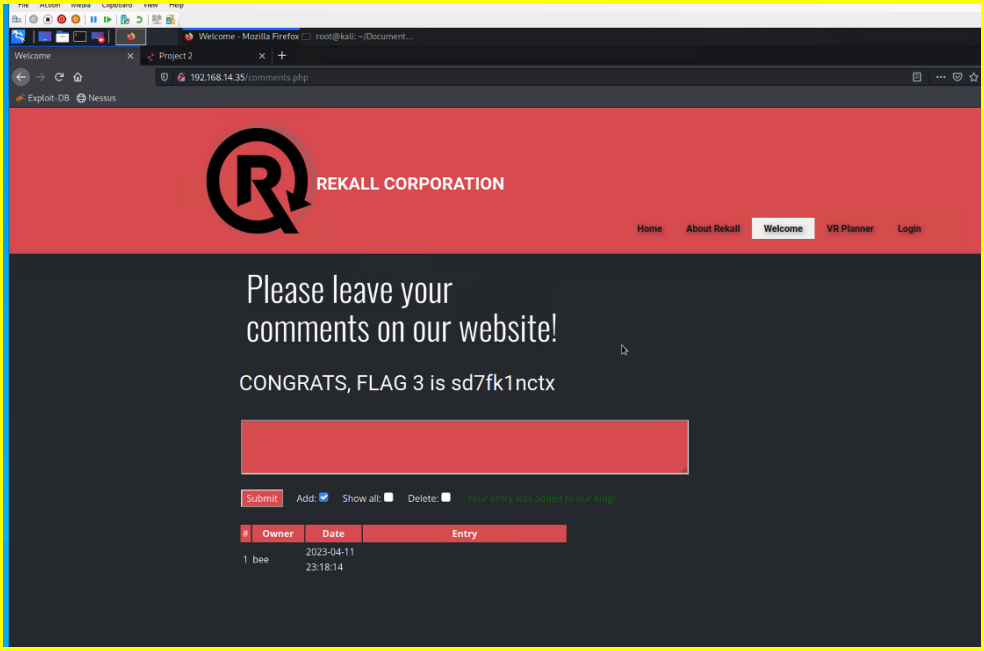
Exploitation Risk	Total
Critical	5

High	3
Medium	3
Low	0

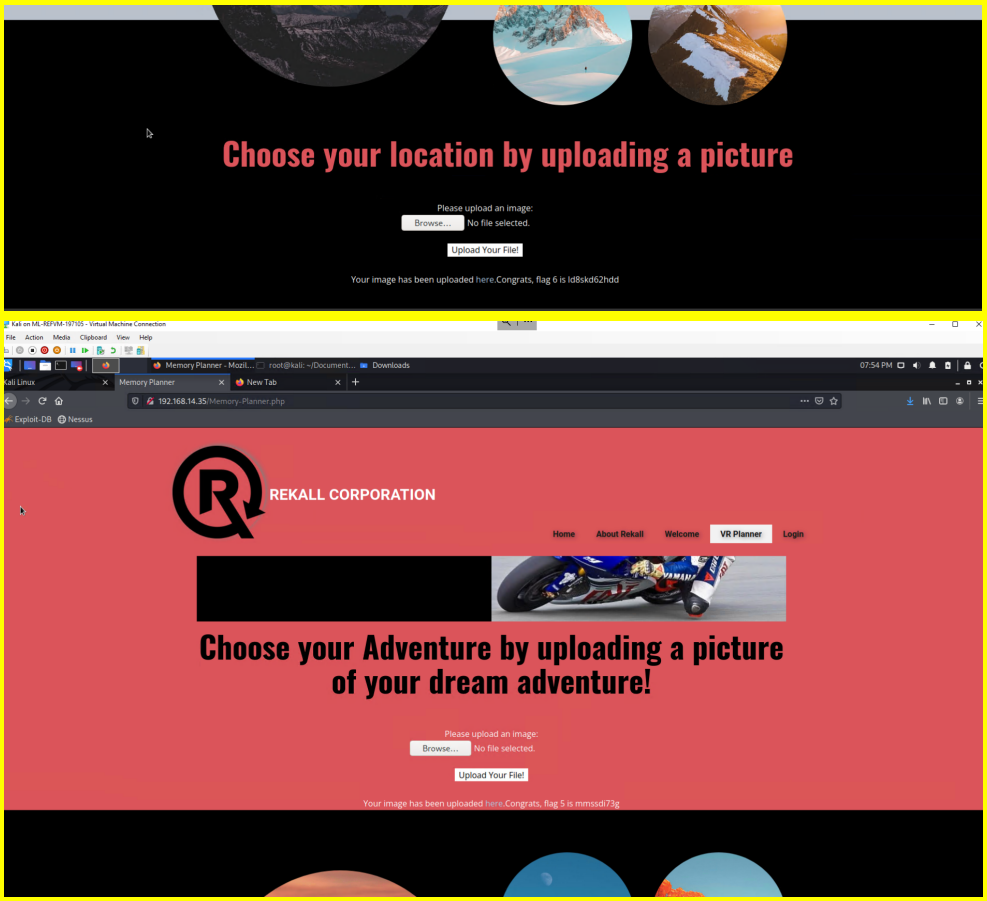
Vulnerability Findings

Vulnerability 1	Findings
Title	XSS Reflected
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Medium
Description	Javascript payloads are able to be placed on home page to force a pop up
Images	
Affected Hosts	192.168.14.35
Remediation	Input validation, Output encoding, HTML Sanitization

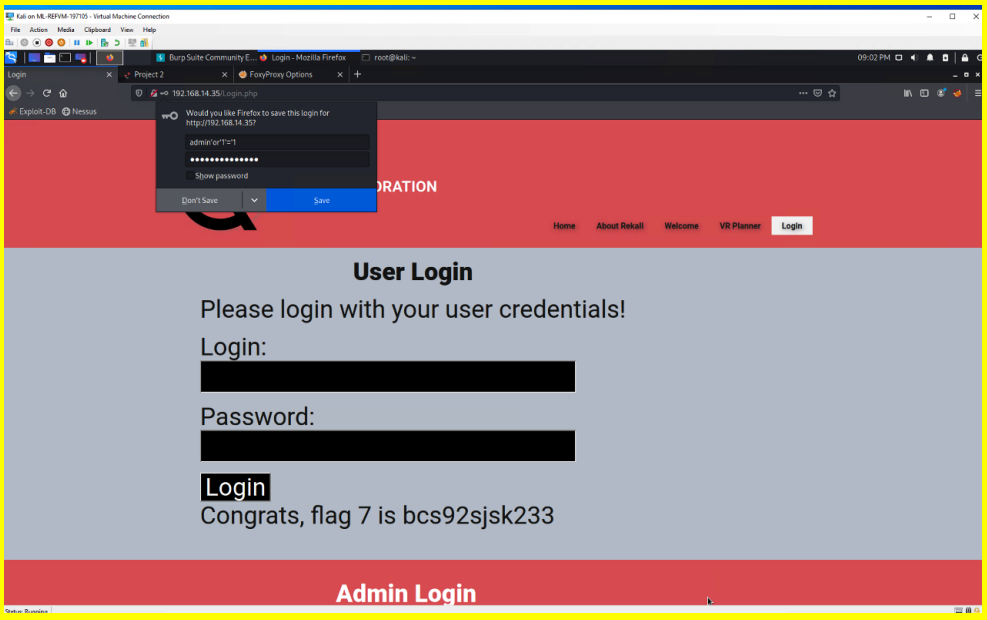
Vulnerability 2	Findings
Title	XSS Stored

Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Medium
Description	Javascript payloads are able to be placed in comments section to force a pop up
Images	 lication
Affected Hosts	192.168.14.35
Remediation	Input validation, Output encoding, HTML Sanitization

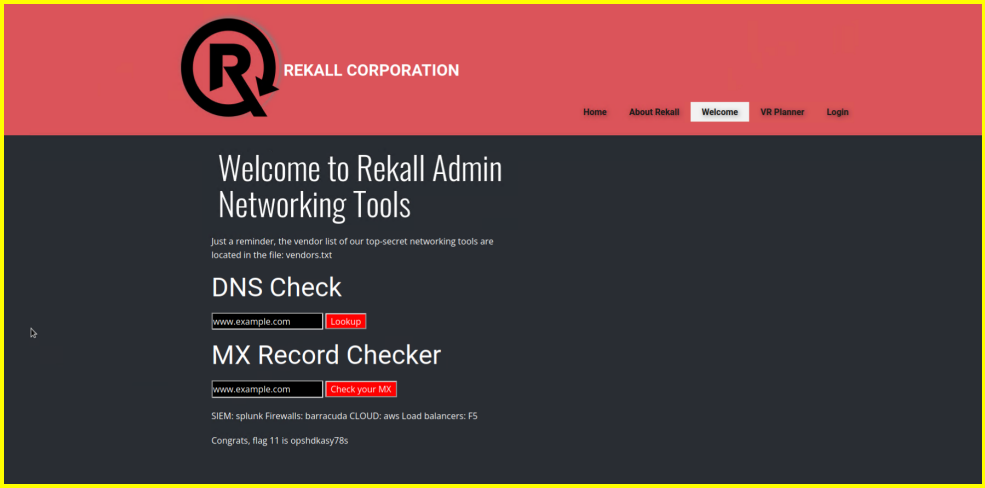
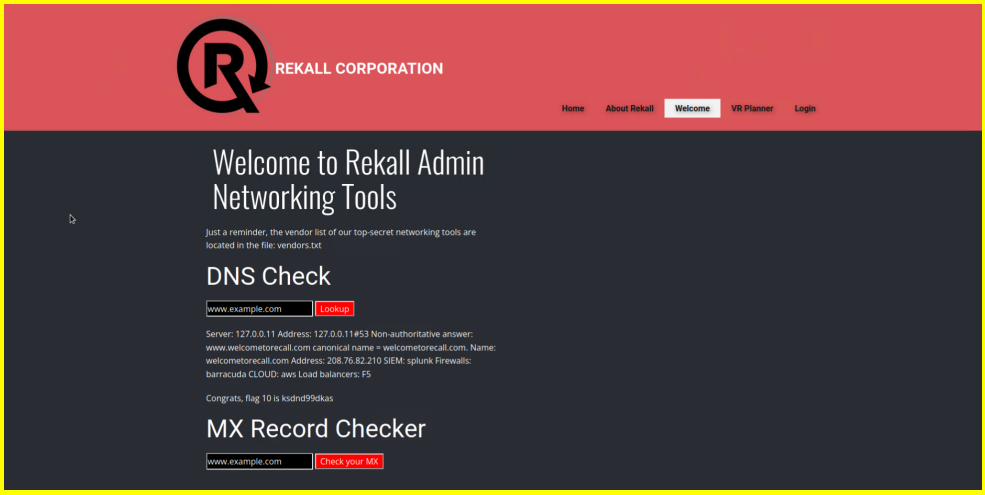
Vulnerability 3	Findings
Title	Local File Inclusion
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	Ability to upload a .php file to expose sensitive files

Images	
Affected Hosts	192.168.14.35
Remediation	Server-side validation, Sanitization, Store files in a database, Limit the API

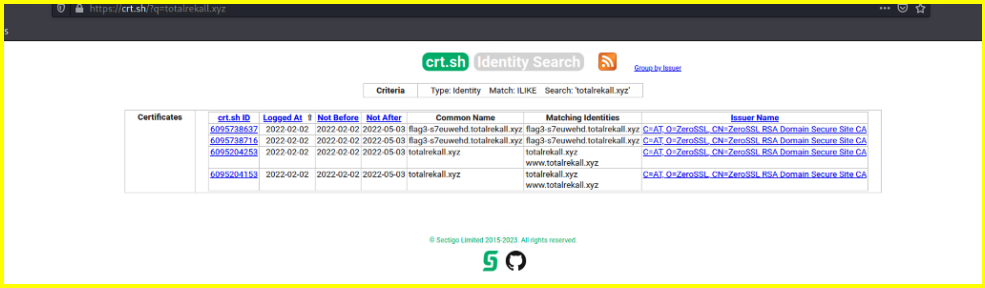
Vulnerability 4	Findings
Title	SQL Injection
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Input a malicious payload on the login page to force a login

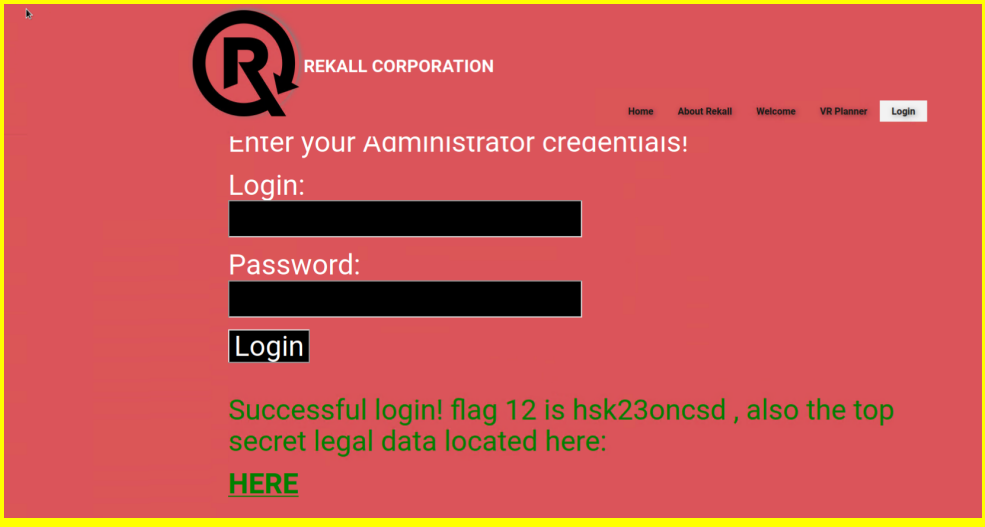
Images	
Affected Hosts	192.168.14.35
Remediation	Server-side validation, Parameterized Queries,

Vulnerability 5	Findings
Title	Sensitive Information in Page Source Code
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	Control + U click on
Images	
Affected Hosts	192.168.14.35
Remediation	Create Source Code protection policies, Incorporate Access Controls, Encrypt Sensitive Source Code info

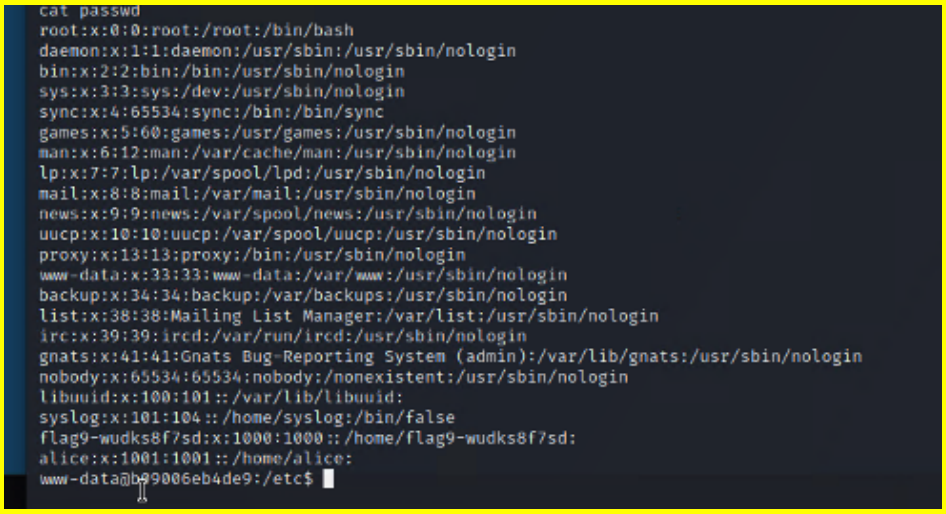
Vulnerability 6	Findings
Title	Command Injection
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	High
Description	Executing a payload in a vulnerable field to access sensitive information
Images	 
Affected Hosts	192.168.14.35
Remediation	Input Validation, White list of approved inputs

Vulnerability 7	Findings
Title	Certificate
Type (Web app / Linux OS / WIndows OS)	Linux

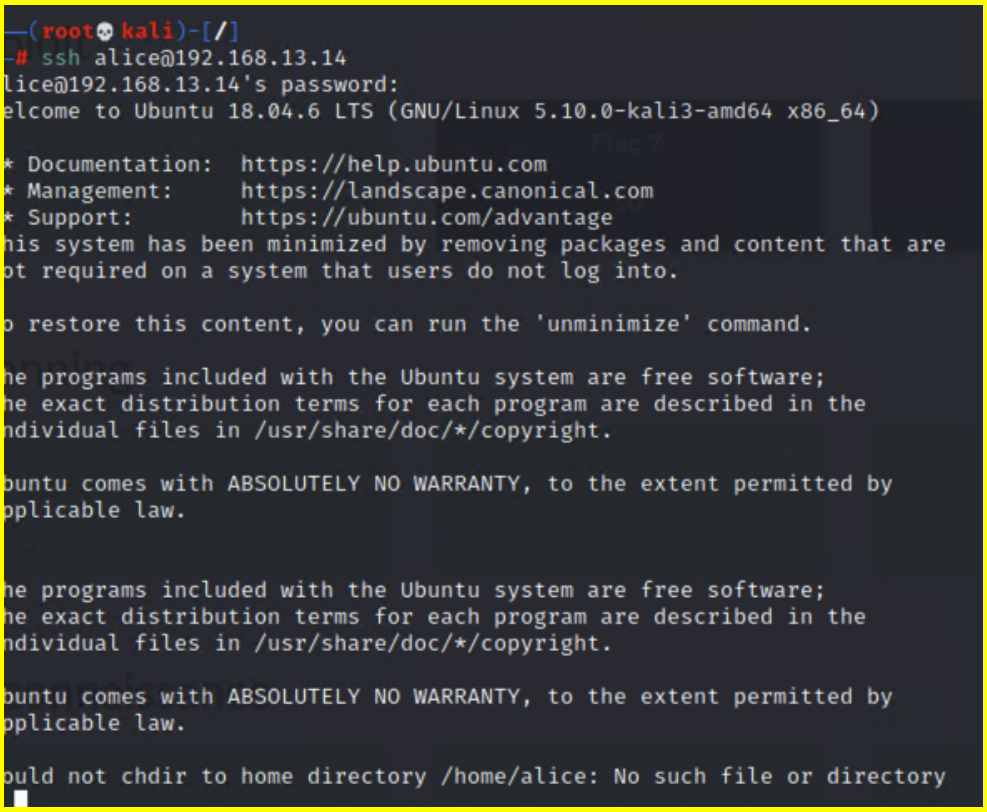
Risk Rating	Medium
Description	Use crt.sh to research SSL Certificates and look for vulnerabilities
Images	
Affected Hosts	192.168.14.35
Remediation	Make sure certificates are up to date, Check configuration of certificates

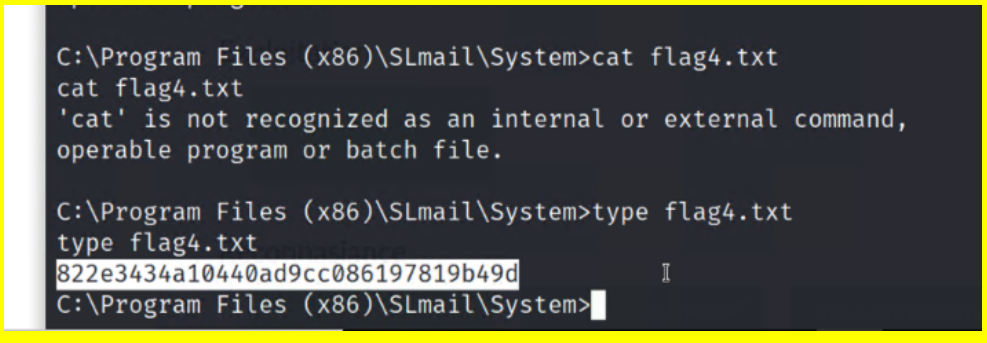
Vulnerability 8	**Findings**
Title	Brute Force Password Guessing
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	Guessing a users password until login is successful
Images	
Affected Hosts	192.168.14.35
Remediation	Password attempt restrictions, Require frequent password changes

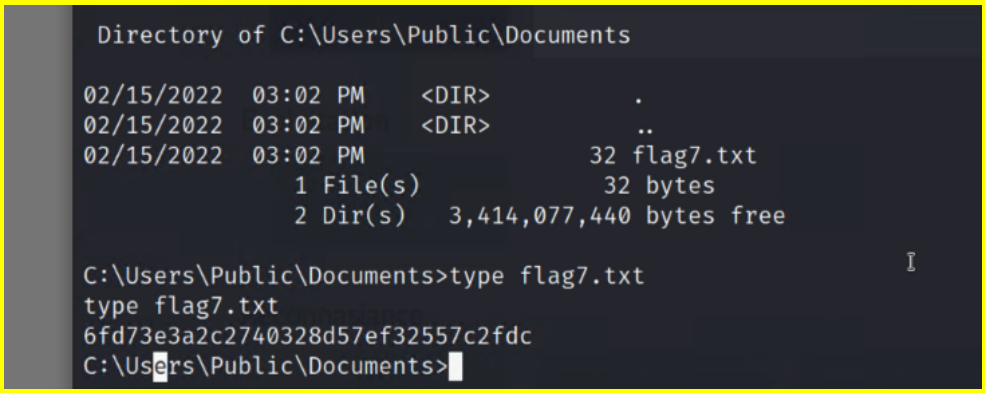
Vulnerability 9	**Findings**
Title	Apache Shell Shock

Type (Web app / Linux OS / WIndows OS)	Linux
Risk Rating	Critical
Description	Exploit a vulnerability in bash with Metasploit to gain access to the .11 host on the network
Images	 (/etc/passwrd file of 192.168.13.11 after exploiting and gaining entry)
Affected Hosts	192.168.13.11
Remediation	Update Bash Version

Vulnerability 10	Findings
Title	Brute Force SSH Login
Type (Web app / Linux OS / WIndows OS)	Linux
Risk Rating	Critical
Description	Using SSH service on Port 22 to guess password of user Alice until logged in

Images	
Affected Hosts	192.168.13.14
Remediation	Password attempt restrictions, Require frequent password changes

Vulnerability 11	Findings
Title	SLMail Service
Type (Web app / Linux OS / WIndows OS)	Windows
Risk Rating	Critical
Description	The SLMail Service version is exploitable with metasploit, we were able to drop into a meterpreter shell to execute code
Images	

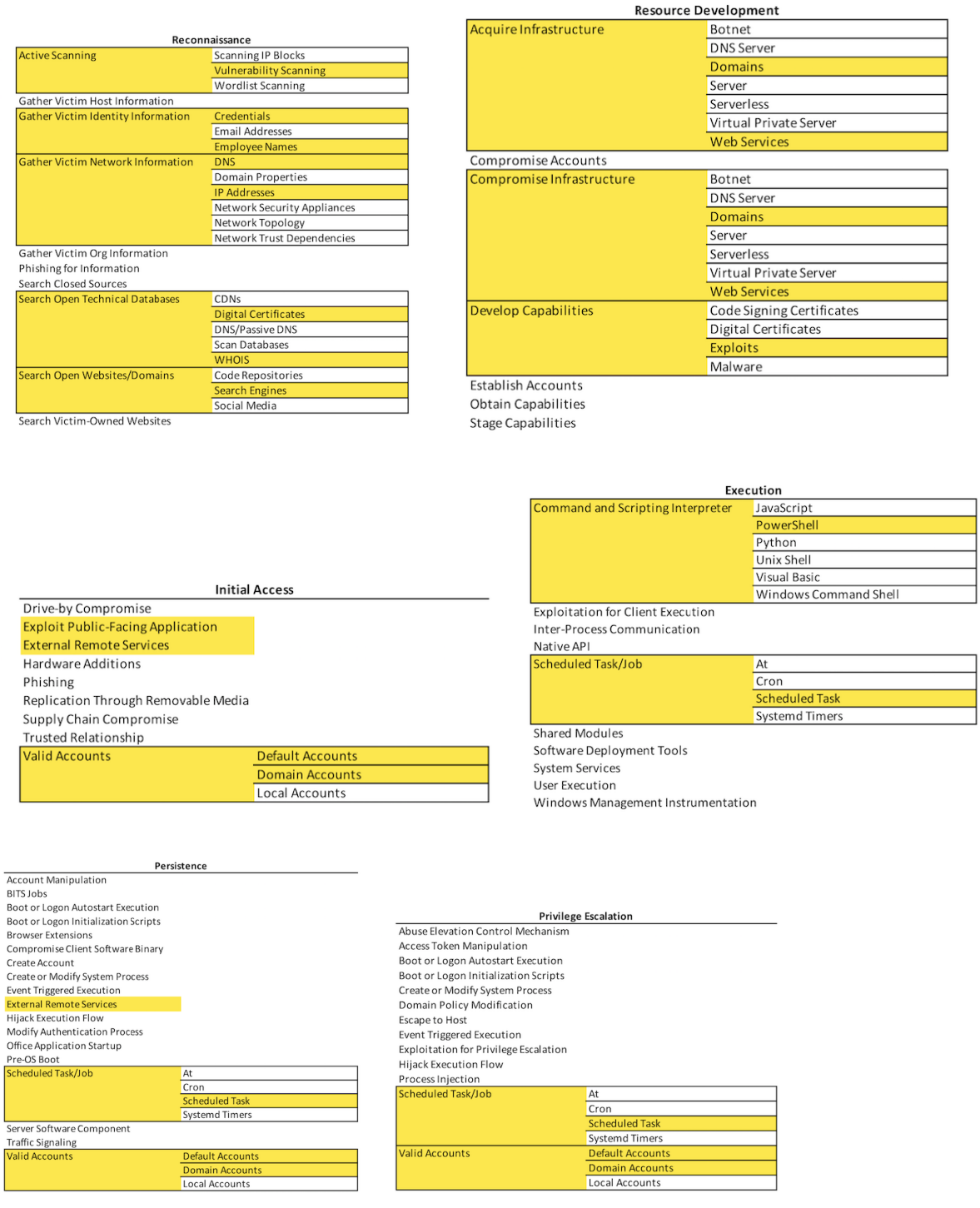
	
Affected Hosts	172.22.117.20
Remediation	Upgrade to current version of SLMail Service

MITRE ATT&CK Navigator Map

The following completed MITRE ATT&CK navigator map shows all of the techniques and tactics that BreakoutRoom4 used throughout the assessment.

Legend:

Performed successfully
Failure to perform



Defense Evasion	
Abuse Elevation Control Mechanism	
Access Token Manipulation	
BITS Jobs	
Debugger Evasion	
Deobfuscate/Decode Files or Information	
Direct Volume Access	
Domain Policy Modification	
Execution Guardrails	
Exploitation for Defense Evasion	
File and Directory Permissions Modification	
Hide Artifacts	
Hijack Execution Flow	
Impair Defenses	
Indicator Removal	
Indirect Command Execution	
Masquerading	
Modify Authentication Process	
Modify Registry	
Obfuscated Files or Information	
Pre-OS Boot	
Process Injection	
Reflective Code Loading	
Rogue Domain Controller	
Rootkit	
Subvert Trust Controls	
System Binary Proxy Execution	
System Script Proxy Execution	
Template Injection	
Traffic Signaling	
Trusted Developer Utilities Proxy Execution	
Use Alternate Authentication Material	
Valid Accounts	Default Accounts
	Domain Accounts
	Local Accounts
Virtualization/Sandbox Evasion	
XSL Script Processing	

Credential Access	
Adversary-in-the-Middle	
Brute Force	Credential Stuffing
	Password Cracking
	Password Guessing
	Password Spraying
Credentials from Password Stores	Credentials from Web Browsers
	Password Managers
	Securityd Memory
	Windows Credential Manager
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	
Steal or Forge Authentication Certificates	
Steal or Forge Kerberos Tickets	
Steal Web Session Cookie	
Unsecured Credentials	

Discovery	Lateral M
Account Discovery	Exploitation of Remote Services
Application Window Discovery	Internal Spearphishing
Browser Bookmark Discovery	Lateral Tool Transfer
Debugger Evasion	Remote Service Session Hijacking
Domain Trust Discovery	Remote Services
File and Directory Discovery	
Group Policy Discovery	
Network Service Discovery	
Network Share Discovery	
Network Sniffing	
Password Policy Discovery	Replication Through Removable Media
Peripheral Device Discovery	Software Deployment Tools
Permission Groups Discovery	Taint Shared Content
Process Discovery	Use Alternate Authentication Material
Query Registry	
Remote System Discovery	
Software Discovery	
System Information Discovery	
System Location Discovery	
System Network Configuration Discovery	
System Network Connections Discovery	
System Owner/User Discovery	
System Service Discovery	
System Time Discovery	
Virtualization/Sandbox Evasion	

ovement	Collection
	Adversary-in-the-Middle
	Archive Collected Data
	Audio Capture
	Automated Collection
Distributed Component Object Model	Browser Session Hijacking
Remote Desktop Protocol	Clipboard Data
SMB/Windows Admin Shares	Data from Information Repositories
SSH	Data from Local System
VNC	Data from Network Shared Drive
Windows Remote Management	Data from Removable Media
	Data Staged
	Email Collection
	Input Capture
	Screen Capture
	Video Capture

Command and Control	
Application Layer Protocol	DNS
	File Transfer Protocols
	Mail Protocols
	Web Protocols
Communication Through Removable Media	
Data Encoding	
Data Obfuscation	
Dynamic Resolution	
Encrypted Channel	
Fallback Channels	
Ingress Tool Transfer	
Multi-Stage Channels	
Non-Application Layer Protocol	
Non-Standard Port	
Protocol Tunneling	
Proxy	
Remote Access Software	
Traffic Signaling	
Web Service	

Exfiltration
Automated Exfiltration
Data Transfer Size Limits
Exfiltration Over Alternative Protocol
Exfiltration Over C2 Channel
Exfiltration Over Other Network Mediur
Exfiltration Over Physical Medium
Exfiltration Over Web Service
Scheduled Transfer