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

Confidentiality Statement 2
Contact Information 4
Document History 4
Introduction 5
  Assessment Objective 5
Penetration Testing Methodology 6
  Reconnaissance 6
  Identification of Vulnerabilities and Services 6
  Vulnerability Exploitation 6
  Reporting 6
Scope 7
Executive Summary of Findings 8
  Grading Methodology 8
  Summary of Strengths 9
  Summary of Weaknesses 9
Executive Summary Narrative 10
Summary Vulnerability Overview 13
Vulnerability Findings 14

Contact Information

Company Name	GOAT
Contact Name	Rahibullah
Contact Title	Pentesters

Document History

Version	Date	Author(s)	Comments
001	07/20/2022	Rahibullah	

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.

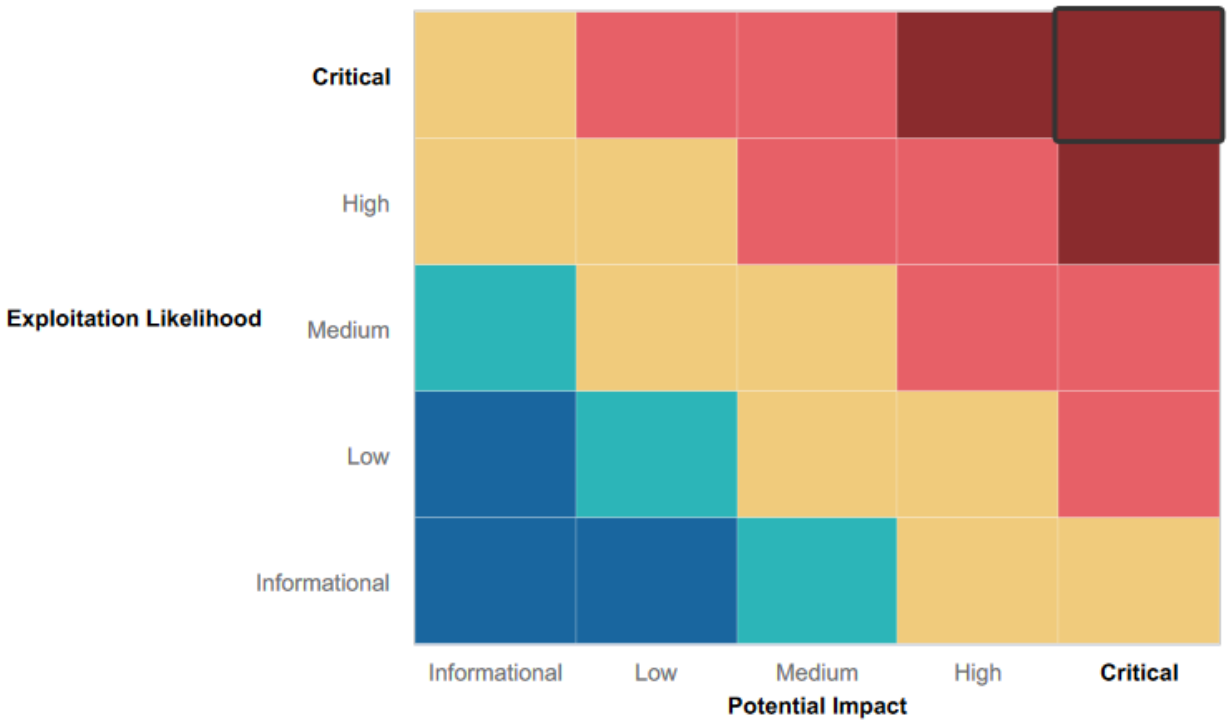
Executive Summary of Findings

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.

- couldn't ssh into any domain controller
- the active directory had no system vulnerability that was exploitable, it was kept up to data.
- The web application had a load balancer , to balance the load means it can help if there is DoS attack.

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.

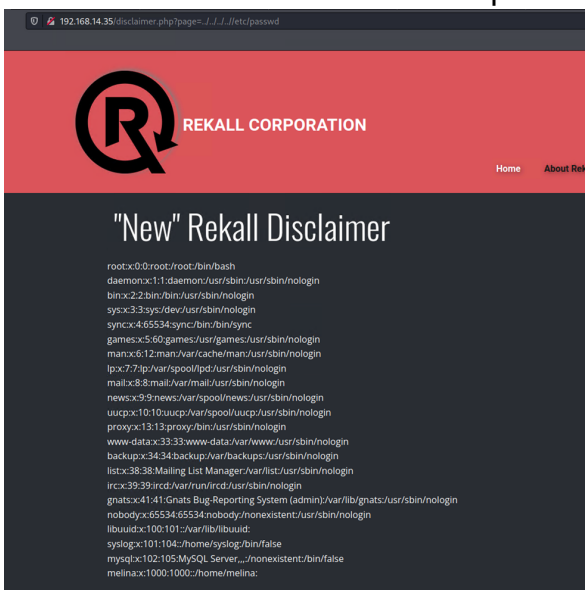
- High-level summary of weaknesses here
- web application: many input fields are vulnerable to XSS
- web application: vulnerable to RCE (remote code execution)
- windows:weak password for admin account.
- web application: PHP injection that will cause RCE
- we: application:File upload , someone can upload PHP files which then can be executed by the webapp
- web application:login input field for users is vulnerable to SQLi
- web application:path traversal , an attacker can read files that are in web application.
- ALL: Most of the vulnerabilities exist due to not updating the services and OSs.
- linux server: the linux server is running tomcat framework old version that is exploitable
- linux server: the linux server also running the old version of drupal framework

Executive Summary

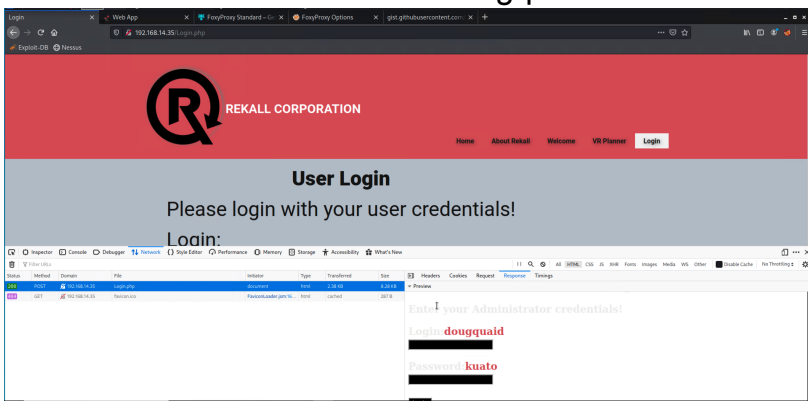
[Provide a narrative summary of your steps and findings, including screenshots. It's fine to mention specifics (e.g., used Metasploit to exploit a vulnerable version of DistCC), but do not get too technical in these specifics. This should be an A–Z summary of your assessment.]

Web App:

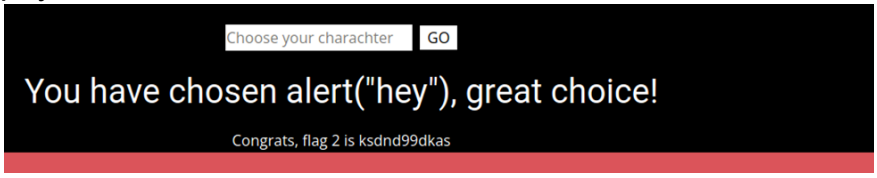
1. From web app homepage, we were able to use directory traversal to access a list of user accounts in the passwd file



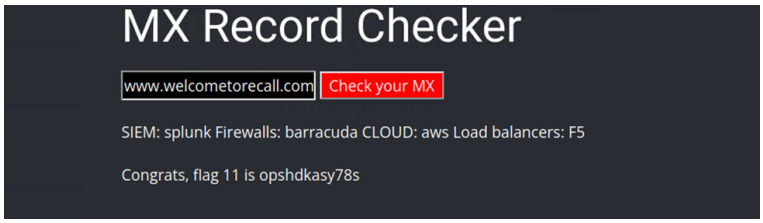
2. On the login page, we were able to obtain HTTP traffic containing user credentials for account dougquaid

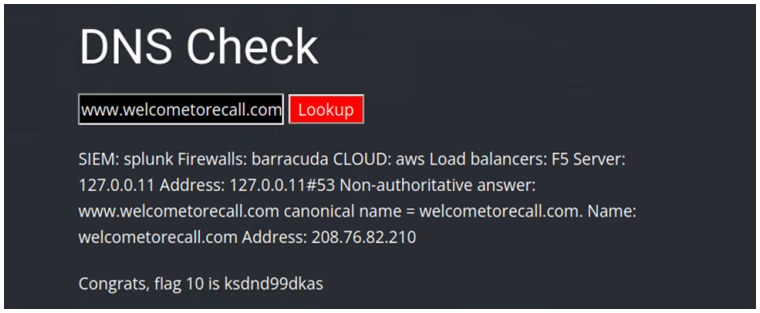


3. "Choose your Character" page is vulnerable to code injection with a simple payload



4. We were able to exploit DNS and MX record checkers through code injection to obtain useful information about the system such as firewall type and version, and different devices on the network.





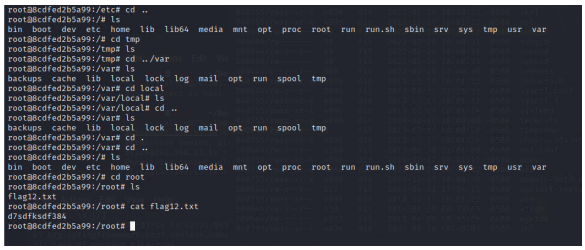
Linux Environment:

1. WHOIS record for totalrekall.xyz lists a username with SSH capabilities in plaintext

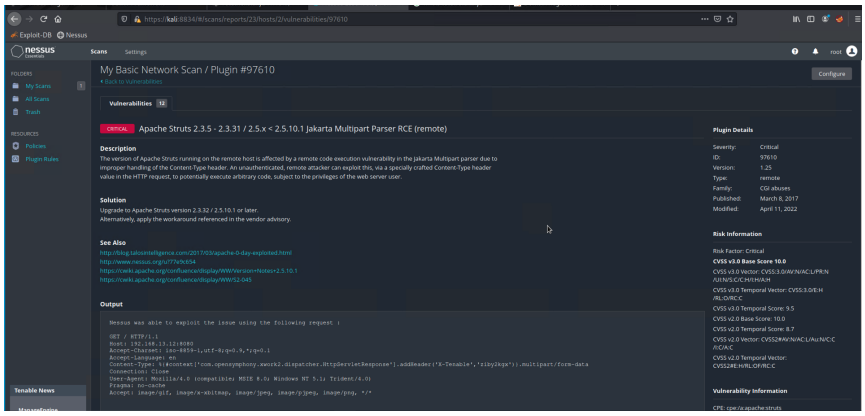
```
Admin Name: sshUser alice
Admin Organization:
Admin Street: h8s692hskasd Flag1
Admin City: Atlanta
Admin State/Province: Georgia
Admin Postal Code: 30309
Admin Country: US
Admin Phone: +1.7702229999
Admin Phone Ext:
Admin Fax:
Admin Fax Ext:
Admin Email: jlow@2u.com
Registry Tech ID: CR534509110
Tech Name: sshUser alice
Tech Organization:
```

1b. Alice had a weak, guessable password (password = alice) which allowed access into a machine with port 22 open.

1c. We were able to exploit a vulnerability through Alice's account to gain access to the root account. This was possible by setting the userID to 0, aka root



2. Apache Webserver was vulnerable to a specific HTTP request exploit which allowed us to execute code remotely.



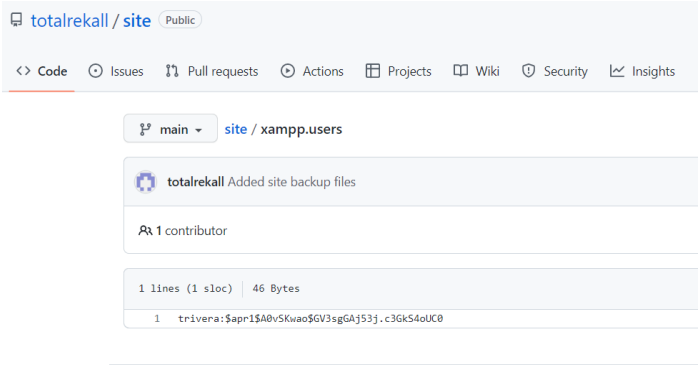
3. Linux environments were vulnerable to reverse shells (essentially remote connection where the target connects to the attacker). This allowed us to view both the passwd and sudoers files

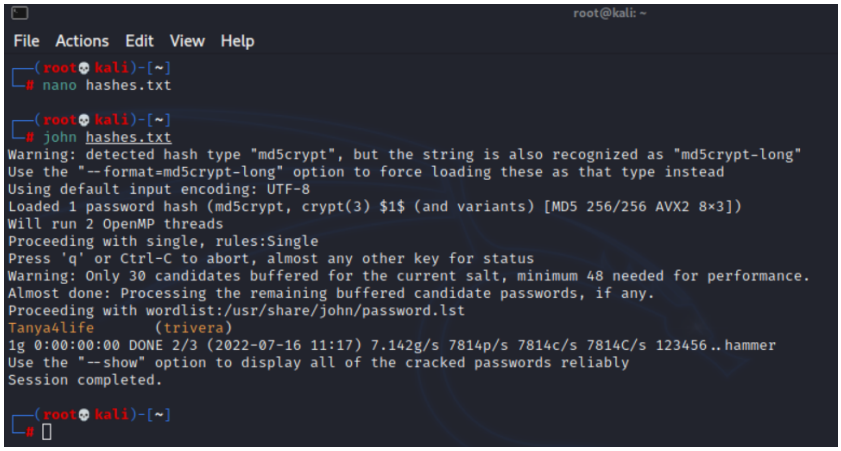
```
meterpreter > cat passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
flag9-wudks8f7sd:x:1000:1000::/home/flag9-wudks8f7sd:
alice:x:1001:1001::/home/alice:
```

```
meterpreter > cat sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
flag8-9dnx5shdf5 ALL=(ALL:ALL) /usr/bin/less
```

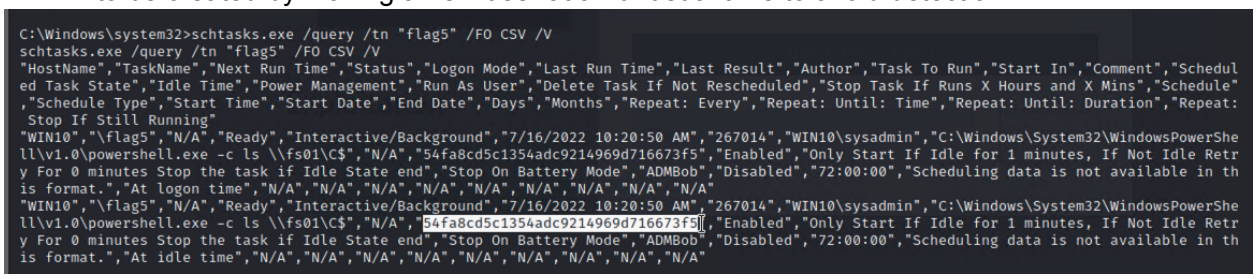
Windows Environment:

1. Username and unsalted password hash were publicly available in a github repository. Unsalted password made it easy to crack, allowing access to the system

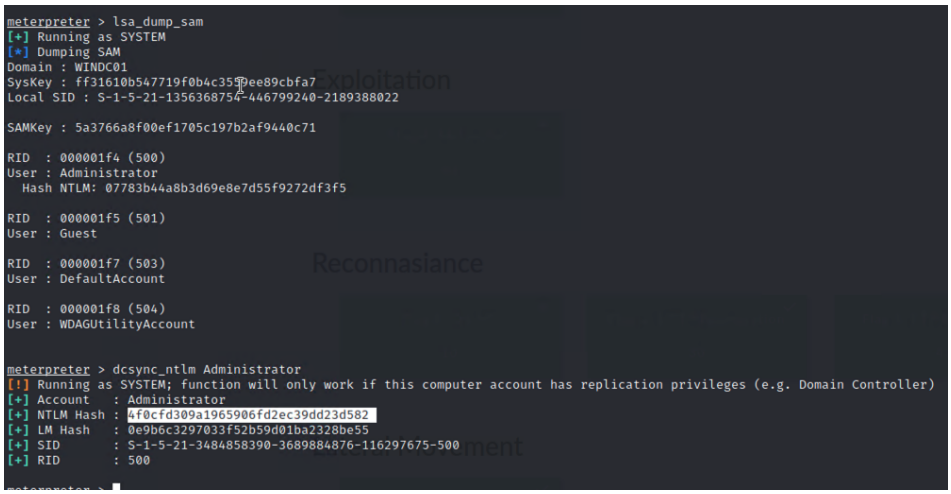




2. We were able to view and modify scheduled tasks which could potentially allow a backdoor to be created by making a new user at an unusual time to avoid detection



3. Through the use of another reverse shell, we were able to obtain the hashed password for any account, including Administrator



4. From there, we were able to escalate privilege to view a list of all users in the system, as well as view any files

```
C:\Windows\system32>net users
net users

User accounts for \\

-------------------------------------------------------------------------------
ADMBob                   Administrator            flag8-ad12fc2ffc1e47
Guest                    hdodge                   jsmith
krbtgt                   tschubert
The command completed with one or more errors.
```

```
meterpreter > ls
Listing: C:\
=============

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
040777/rwxrwxrwx  0      dir   2022-02-15 13:14:22 -0500  $Recycle.Bin
040777/rwxrwxrwx  0      dir   2022-02-15 13:01:09 -0500  Documents and Settings
040777/rwxrwxrwx  0      dir   2018-09-15 03:19:00 -0400  PerfLogs
040555/r-xr-xr-x  4096   dir   2022-02-15 13:14:06 -0500  Program Files
040777/rwxrwxrwx  4096   dir   2022-02-15 13:14:08 -0500  Program Files (x86)
040777/rwxrwxrwx  4096   dir   2022-02-15 16:27:48 -0500  ProgramData
040777/rwxrwxrwx  0      dir   2022-02-15 13:01:13 -0500  Recovery
040777/rwxrwxrwx  4096   dir   2022-02-15 16:14:31 -0500  System Volume Information
040555/r-xr-xr-x  4096   dir   2022-02-15 13:13:58 -0500  Users
040777/rwxrwxrwx  16384  dir   2022-02-15 16:19:43 -0500  Windows
100666/rw-rw-rw-  32     fil   2022-02-15 17:04:29 -0500  flag9.txt
000000/---------  0      fif   1969-12-31 19:00:00 -0500  pagefile.sys

meterpreter > cat flag9.txt
meterpreter > cat flag9.txt
f7356e02f44c4fe7bf5374ff9bcbf872meterpreter >
```

Summary Vulnerability Overview

Vulnerability	Severity
HTML Exploitation of web app	**Critical**
drupal vulnerability - linux machine	**Critical**
XSS payload vulnerability - web app	**High**
default password and username	**Critical**
command injection - web app	**Critical**

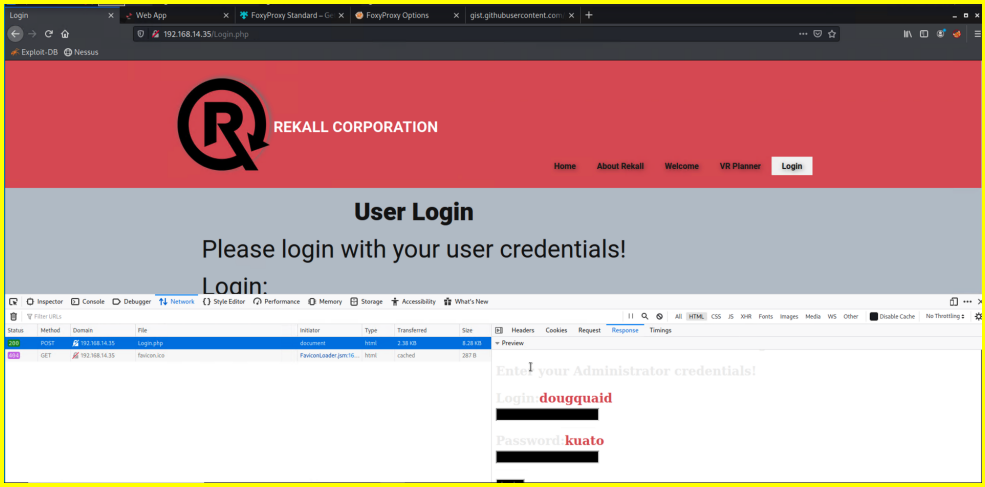
Hash dumping	High
view and modify scheduled tasks - backdoor access	**High**

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

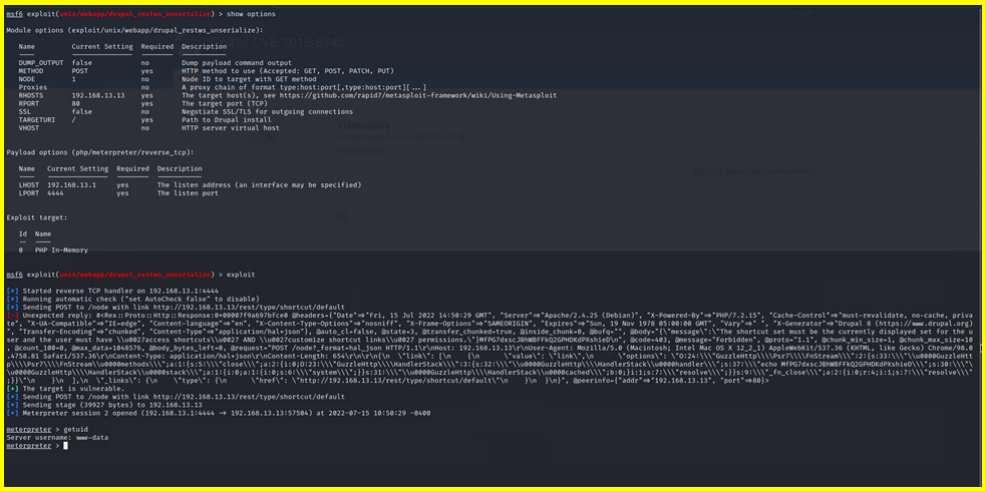
Scan Type	Total
Hosts	7 IPs (web app, windows and linux)
Ports	Most common 1000

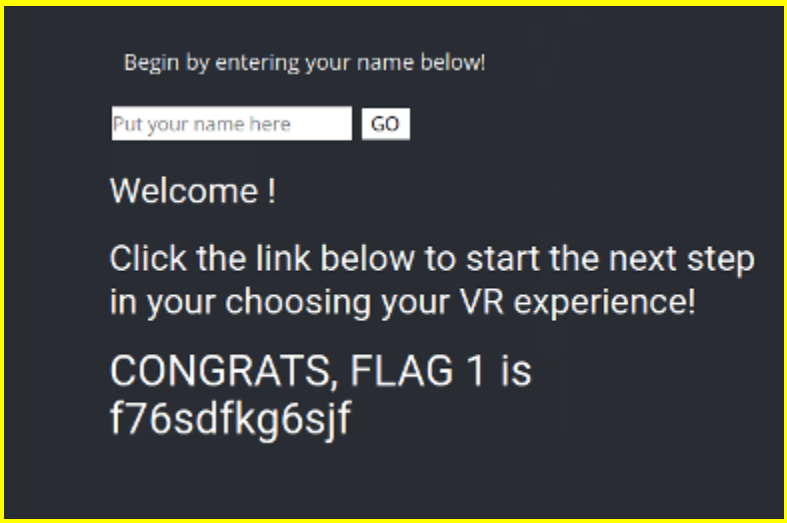
Exploitation Risk	Total
Critical	4
High	3
Medium	0
Low	0

Vulnerability Findings

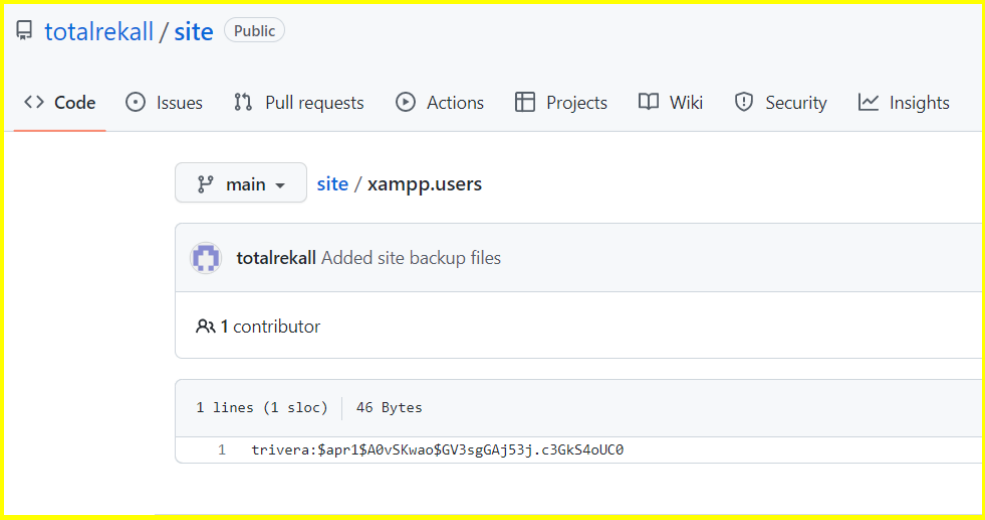
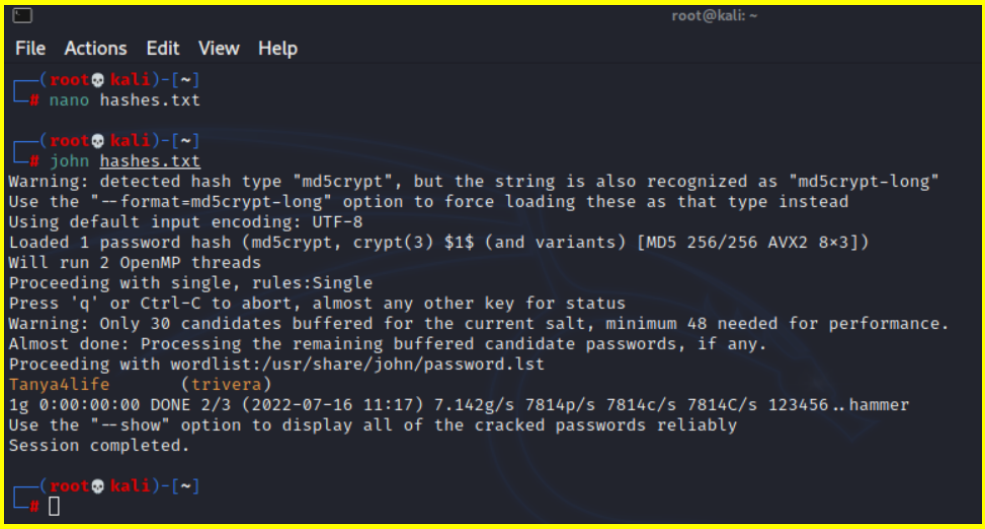
Vulnerability 1	Findings
Title	HTML Exploitation
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	Critical
Description	within the Login.php page the admin username and password were able to be obtained from the HTTP traffic in the Network Response tab when inspecting the website page
Images	
Affected Hosts	192.168.14.35/Login.php
Remediation	don't allow the username and password to be cached in the html, additionally, use two-step authentication to prevent brute force attacks

Vulnerability 2	Findings
Title	drupal vulnerability
Type (Web app / Linux OS / WIndows OS)	linux
Risk Rating	Critical
Description	It's a vulnerability that is exploited in 2019 , which give the attacker shell in the target system if using drupal 8.x

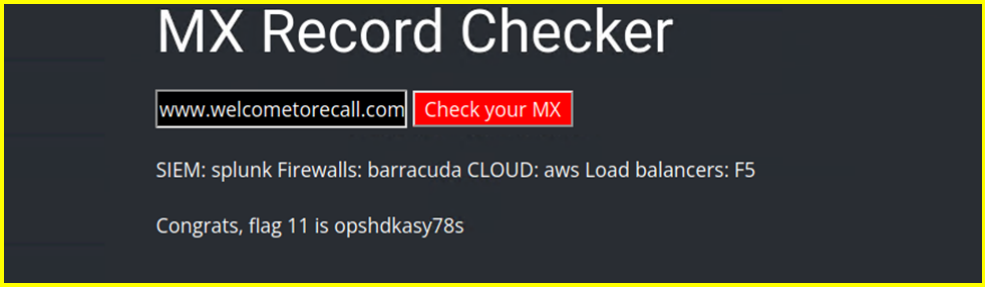
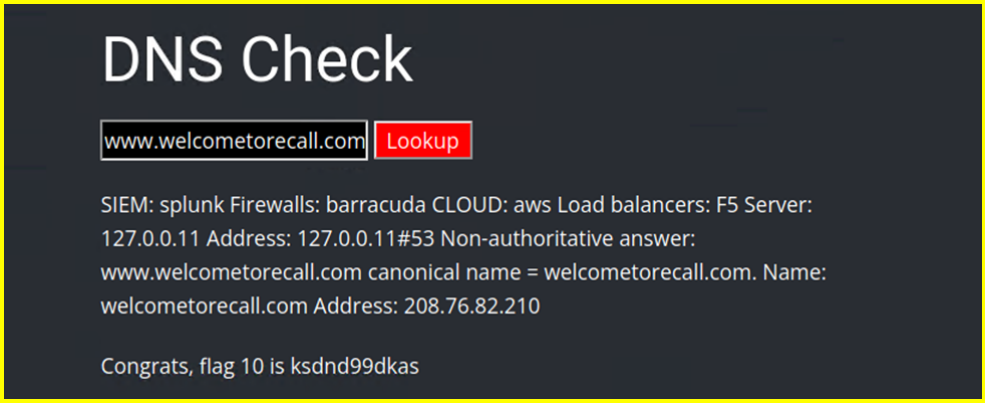
Images	
Affected Hosts	http://192.168.13.13
Remediation	to fix this vulnerability we have to update the website framework, the drupal version that is running in the system is version 8 but there is version 10.x is available , as soon as possible pleas update

Vulnerability 3	Findings
Title	XSS payload vulnerability
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	High
Description	This is where a malicious script is injected into an otherwise benign and trusted website. Our team was able to inject the payload <script>alert(“hey”)</script>
Images	
Affected Hosts	192.168.14.35/Welcome.php
Remediation	Filter input on arrival. At the point where user input is received, filter as strictly

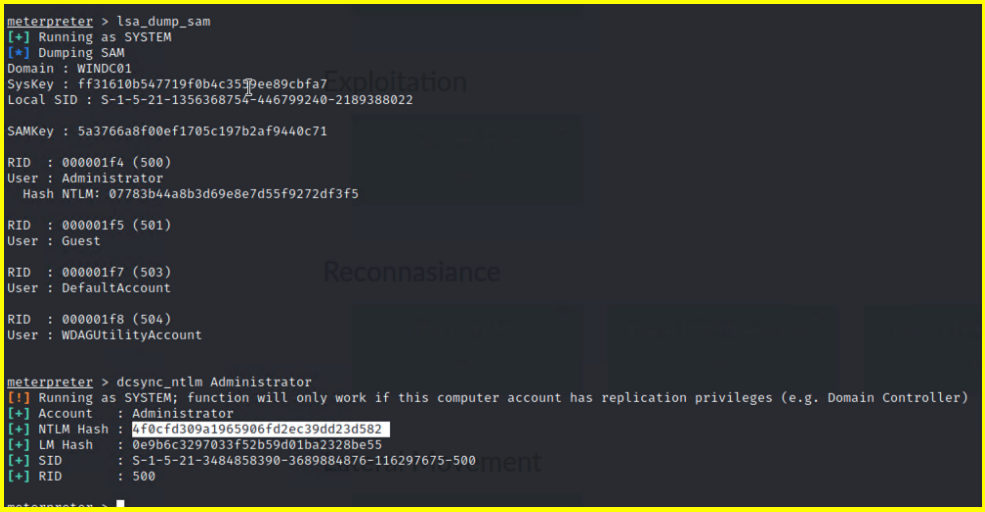
	as possible based on what is expected or valid input.

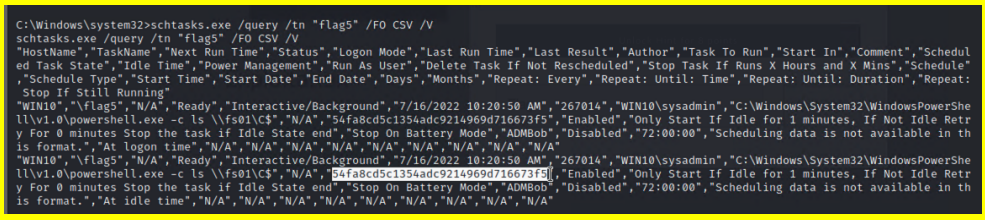
Vulnerability 4	Findings
Title	default password and username
Type (Web app / Linux OS / WIndows OS)	windows
Risk Rating	critical
Description	the repository downloaded for web application has its default username and password , after finding out what repository is used we found the default user name and the password by finding the repository in the github.com
Images	 
Affected Hosts	172.22.117.20
Remediation	change the password as soon as possible.

Vulnerability 5	Findings

Title	command injection
Type (Web app / Linux OS / WIndows OS)	web application
Risk Rating	critical
Description	a normal user can login and be able to read and look at the files , website has a feature where a user can check MX (mail exchange) records of a domain and also DNS check which execute nslookup command in the background, a malicious user can use &, | ,-a … after the nslookup command executed these symbols will execute the attack command. here we read the a file where it says what kind of firewall, SIEM, load balancer is in use
Images	 
Affected Hosts	http://192.168.14.35
Remediation	when a web application let a user to use feature like ping, traceroute, MX records …that is executing command in the OS that application is running , there should always be input validation. the user should not be allowed to input things that are not allowed.

Vulnerability 6	Findings
Title	Compromising Admin using lsa_dump_sam (hash dumping)
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	Once our team created a connection to meterpreter after exploiting the system,

	we were able to use the command lsa_dump_sam to get the Administrator's NTLM Hash. This hash can then be exploited and provide us with the password to the admin account which allows for a brute force attack
Images	
Affected Hosts	172.22.117.20
Remediation	Make sure the system is updated/patched. Use complex hashes. Make sure access controls are implemented

Vulnerability 7	Findings
Title	view and modify scheduled tasks - backdoor access
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	viewing and modifying the task scheduler is a weakness that allows attackers to escalate privileges to root if improperly configured
Images	
Affected Hosts	172.22.117.20
Remediation	audit scheduled tasks by ensuring your are logging events to watch out for new tasks being set

Add any additional vulnerabilities below.