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	Cyber Village Security
Contact Name	Kylan Aburo-Pratt
Contact Title	Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	2/2/2023	Ky Pratt	Web Application Vulnerability
002	2/6/2023	Ky Pratt	Linux Assessment
003	2/8/2023	Ky Pratt	Windows Assessment

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.

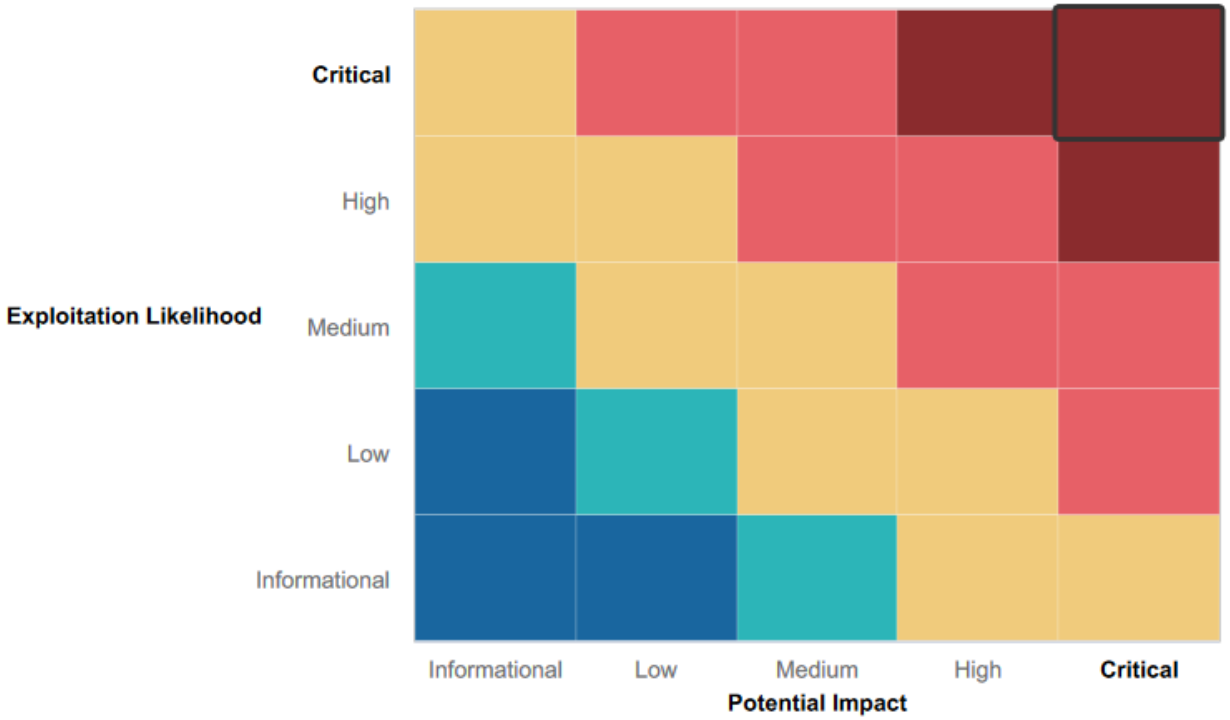
Executive Summary of Findings

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.

- Protected Shadow File
- Blind SQL Injection
- Blind XSS
- Some input validation on forms.
- Limited user accounts available (linux)
- Some passwords had sufficient depth

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 Reflected
- XSS Stored
- Unsecured HTTP Headers
- Local File Inclusion
- SQL Injection
- Exposure of Sensitive Data in HTML
- Sensitive Data in Robots.txt
- Command Injection + Exposure of Vendors.txt
- Weak Admin Passwords
- PHP Injection
- Broken Access Control
- Directory Traversal + Exposure of Disclaimer_1.txt)
- Exposure of Passwords in Code repositories
- Out of date versions of Sudo, Apache Tomcat, Drupal, Apache Struts.
- Permittance of Anonymous FTP downloads
- Out of date versions of SLMal
- Overlapping credentials
- Unmonitored commands for OS credential dumping.

Executive Summary

Web Application Assessment

Our team at Cyber Village Security first initiated our evaluation of Rekall Corporation through their customer facing web application at hxxp://192.168.14.35/. While overall a great product and experience offered, the application can be interacted with in many ways that pose risk to the company's data and overall stability of the application.

Cross-Site Scripting was executed in several page URLs and intake forms in the Welcome.php, Memory-Planner.php, and comments.php page. On the page that takes the user's name, a simple script was able to be entered and executed through the URL. No filters were in place leaving this page as a major vulnerability. Similarly the URL on the Memory Planner page was ultimately vulnerable to the same attack. However, in this case there was a small filter on the input for the word "script". We were able to bypass this by disguising the word script within the work "s***script***cript". While this was an improvement in input validation it proved not enough to stop these two reflected XSS attacks. Additionally, the comment page allowed visitors to enter executable script without any barriers. This situation was a case of stored XSS, meaning the script was persistent for any future visitors to this page. To prevent both of these situations input validation must be put in place within the structure of how the website takes data.

Sensitive data was accessible and exposed in several places throughout the web application. The HTTP response headers contained data that could prove valuable to an attacker. This was seen in the About-Rekall.php page. More information was exposed in the Robot.txt file. This file is a common element to websites that tell web crawlers what to not crawl. However, this scenario exposed the souvenirs.php page that presented its own vulnerability when accessed. The robots.txt file is not a vulnerability alone, but the Rekall team should put proper access controls on any sensitive data involved on that page.

There were 2 photo upload forms on the memory-planner.php page. While the form's usage was only for images, both were able to accept php scripts. With this vulnerability, an attacker could upload malicious scripts at will. The second of the two forms had a small filter requiring the jpg suffix, however a simple file name of script.jpg.php was able to bypass this. This, again, is a matter of input validation. To make secure, the upload forms should have a whitelist applied for only specific data types.

The user login page presented several vulnerabilities, beginning with examining the HTML of the page. Within the source code we found the admin credentials for user Doug Quaid. Credentials should never be stored in HTML and wherever they are stored they should always be encrypted and password protected. The login form on its own was exploited with an SQL Injection. By feeding the password field with a true statement to pair with a random password we were able to bypass the need for an actual password. There are many SQL techniques and they must be accounted for in data sanitization. Applying a character limit on the username and login fields can sometimes limit what types of attacks can even be attempted.

Accessing Doug Quaid's account gave reference to a new page only intended for administrators, the Networking.php page. Here on this page was a direct written header about sensitive data in the vendors.txt file. On this page there were two entry forms, A DNS checker and an MX Checker. Both forms allowed for command injection and directory traversal. The DNS form could be escaped with a double ampersand '&&' and the MX form could be escaped with a pipe character '|'. With this ability we were able to read the information of the Vendors.txt file. Additionally with our ability to explore directories we could read the etc/passwd file. To avoid command injection there needs to be a limitation of what characters can be used on the entry forms as well as a method of input validation. Path traversal can be limited by applying proper access control.

Per usual the etc/passwd file contained the users within the system. Specifically of interest was the user Melina. This user presented a major risk with the use of a weak password. Through password guessing, we were able to access Melina's admin account. All users should adopt stronger

password requirements to prevent such an easy breach. Entering Melina's account took us to a page containing the legal data for Rekall.

Admin-Legal-Data.php is a page that requires admin access to read the file. However it was vulnerable and we were able to show broken access control. The URL contained an admin=001 parameter which we were able to brute force by cycling through numbers. Admin=87 gave us admin access to read this sensitive file.

The Rekall web application contained quite the number of old pages not properly disposed of. Souvenirs.php which was exposed in the robots.txt file was still accessible to those who knew the URL. On this page we were able to inject a php script in the URL. This ability could allow an attacker to gain remote code execution which is a critical vulnerability.

The final page that was accessible was the old_disclaimer directory containing the diclaimer_1.txt. This file was accessible in multiple ways and revealed sensitive data about the side effects of the Rekall experience. From a technical vulnerability standpoint this page allowed us to abuse the page parameter to explore any file path we knew of.

This concluded our assessment of the Web Application. Our team would continue to assess Linux and Windows systems in the proceeding days. While a detailed recap follows on mitigation, we believe that a major focus should be on improving the password requirements of the users within the company and applying methods of input validation on the forms within the site.

Linux Assessment

Our gathering of details started by utilizing open source techniques. Leveraging Domain Dossier we queried the records of totalrekall.xyz. A lot of information was available through reading through the results. Specifically of interest to us was information on an administrative user named Alice and a street address. This information served value in assessing the exposure of personal data and later on a point of access. Among this information were numerous records including the IP address of totalrekall.xyz **34.102.136.180.** Investigating the results of certificate transparency we gained access to several hosts potentially in use and their certificate of authority. It is always best to be aware of what data is present about your network online, as there are hundreds of OSINT sites and tools. A regular check on what is exposed can help you be aware of what might need to be secured or even removed.

Our team discovered 5 Linux based hosts through an Nmap scan of the IP address range of 192.168.13.0/24. The machines ranged from 192.168.13.10-14. We were able to execute an aggressive scan and find more information on all the ports visible and services running on them. This is how we decided to attempt exploiting each machine. Leveraging a firewall and intrusion detection system (IDS) can allow the blocking of probes and restricting information returned by tools like Nmap. It's also recommended for the team to stay up to date on understanding what services are in use, their need, and their version vulnerabilities over time.

Another one of our tools in this endeavor was the vulnerability scanner, Nessus. Knowing the machine IPs, Nessus allowed us to cross reference common vulnerabilities and the services running on each host. Several critical vulnerabilities were suggested on each machine. Our first target was the .10 Machine which was running Apache Tomcat. Utilizing Metasploit we were able to gain root access in a command shell through the Multi/http/tomcat_jsp_upload_bypass. From here we could move around as we pleased and access any file.

Our next target was the .11 machine. With a Shellshock attack we were again able to gain access to the machine. The account we gained access to was in the Sudoers group so we had a strong amount of privileges on this machine as well. While inside these machines we could see the etc/passwd file. While there were not many users on the system there was one that was referenced to connect to earlier findings: Alice. Alice would come into play down the road as we looked to enter the .14 machine.

Working sequentially, we attempted to access the .12 machine. Vulnerability scans showed a potential critical vulnerability to Jarkarta Apache Struts. Our Metasploit tool was able to gain a shell utilizing the struts2_content_type_ognl exploit. Here we extracted a 7zip file that contained sensitive data.

The .13 Machine was running Drupal as a service. We were able to exploit this utilizing ao common exploit. Gaining entry to this system remotely, we could then look through each file and exploit sensitive data. This can be fixed with a simple upgrade of the Drupal version used.

The final machine located at .14 did not have any critical vulnerabilities that were shown. However through exploiting OSINT resources we found that it may be vulnerable to a sudo exploit. Specifically CVE-2019-14287. This left us with a need to get into the system and try it. To enter this system we leveraged information gathered earlier from the Domain Dossier. Alice was listed as an SSHuser and administrator. Her password was guessed utilizing password guessing techniques. Ultimately our ability to enter this account was due to Alice's weak password of 'alice'. While inside we could exploit the sudo exploit by tricking sudo into assuming a fake user. We were then able to gain root access and read all files.

Windows Assessment

Our team discovered 2 windows machines on the network range of 172.22.117.0/24. A windows workstation was located at 172.22.117.20 and a Windows domain controller was located at 172.22.117.10.

To begin our assessment we used OSINT techniques to search the web. In a Github repository we found a username and password for the user, trivera. The password was presented in a hash form but we were able to crack it to reveal the passwords in plaintext:Tanya4life. This credential was used to enter a website on the internal network at 172.22.117.20.

After running an aggressive nmap scan, the Windows workstation showed it allowed anonymous ftp logins and showed a file that we could read. Utilizing the ftp command we could log in and pull the file to our system to read. It is possible to limit the data shown on a scan, but time must be taken to decide the best way to do this and if it is necessary.

The Windows Workstation had several services running, one of interest was the SLMail service. We were able to exploit this service, start a remote session through meterpreter and log into the Windows Workstation as a service. From here we could migrate to any process. At this time we migrated to a process with domain system privileges. From here we could check the scheduled tasks where we found a task running that provided no value to the system.

After gaining system privileges, our team utilized our meterpreter shell and the command tool called Kiwi. With this tool we could dump the credentials of the SAM file and the cached credentials. The credentials revealed 3 users of interest: flag6, sysadmin and ADMbob. All three of these users had password hashes that were cracked.

While inside our remote session we could explore the directories and files of the workstation. Within the Documents/public/ folder was a file with sensitive data. This is a situation that comes down to user awareness. Sensitive files should only be kept in highly secure areas, not a public access folder on the network.

ADMbob was an administrator account which allowed us to exploit the SMB protocol and Windows Management Instrumentation (WMI). Utilizing auxiliary techniques we were able to scan the domain controller and find that the credentials for ADMbob overlapped. While maintaining a session on the Workstation we took advantage of WMI to laterally move over to the Domain Controller as a system process. Here, like before we could enumerate all the users on the account.

At the root of the C drive was a sensitive file that we were able to read. Given our system privileges on a Domain Controller we could execute as we pleased. The last exploit we accomplished was dumping the credentials of the users on the system. Because this was a Domain Controller we had to go about accessing the credentials differently than the local attack. A DCSync attack was performed and we gained access to the hashes of all the users on the DC. This credential dump specifically included the password for the Administrator Account. Many of the passwords cracked were revealed in little to no time because they were not strong enough. A common theme in the overall cyber security culture of Rekall Corporation that needs to change immediately.

Impact and Remediation

A detailed recap of solutions and how we conducted our exploits is in the Vulnerability Findings section below, however that is more of a technical overview. The main takeaway from our assessment is that your systems and web applications possess some critical vulnerabilities that allow for complete access of both your Windows and Linux servers. The impact and financial risk of having these vulnerabilities present is critical as an attacker could access any system and file to delete, change, or encrypt and leverage as through uploading ransomware.

Overall many of the exploit paths we took can be remedied by updating the versions of the services being used. Many of these entry vectors involve exploiting a service with a version that is out of date. It is best practice that the Rekall Defense Team monitors the cyber security climate to always stay patched and up to date on the services used to be functional.

With that being said there will ultimately be vulnerabilities that come in the future and the presence of what is known as a Zero-Day vulnerability. It is best to apply a Defense In-Depth model at Total Rekall. Staying up to date on the versions used within your networks is just one layer of defense. Additional defense can be put in place through intrusion detection, and monitoring abnormal behaviors. This additional layer of defense, if implemented correctly can allow you to catch an attacker before too much damage is done. Storing your sensitive company assets behind walls of detection, and stored separately from one another allows for a potential outcome of less damage if there is damage.

Lastly but certainly not least, the education of your employees and adoption of a cyber security culture is paramount to protecting your company in the digital world. A password strengthening initiative should be taken immediately as many of our exploits leveraged a valid credential we had access to.

Summary Vulnerability Overview

Vulnerability	Severity
Web Application	
Unsecured HTTP Headers	Medium
Local File Inclusion (LFI)	Critical
SQL Injection	Critical
Exposure of Sensitive Data in HTML (Doug Quaid)	Critical
Sensitive Data Exposure (Robots.txt)	Medium
Command Injection + Exposure of Sensitive Data (Vendors.txt)	High
Weak Admin Passwords (Melina)	Critical
PHP Injection	Critical
Broken Access Control	Critical
Directory Traversal + Exposure of Sensitive Data (Disclaimer_1.txt)	High
Linux Systems	
Exposure of Networking Information and Credentials	Medium
Weak User Passwords	Critical
Apache Tomache JSP Bypass CVE-2017-12617	High
Shellshock Exploit CVE-2014-6271 CVE-2014-6228	Critical
Apache Jakarta Struts Exploit CVE-2017-5638	Critical
Drupal Exploit CVE-2019-6340	High
Sudo Exploit CVE-2019-14287	High
Windows Systems	
Exposure of Sensitive Data in Code Repositories	Medium
Unauthorized FTP Transfer of Files	Medium
SLMail Exploit CVE-2003-0264	High
Process Migration	High
Uncommon Scheduled Tasks Running on Workstation	Medium
Unmonitored SAM and Cache Dumping	Critical
Admin Credential Overlap	Critical
DCsync Attack/Weak Passwords	High

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

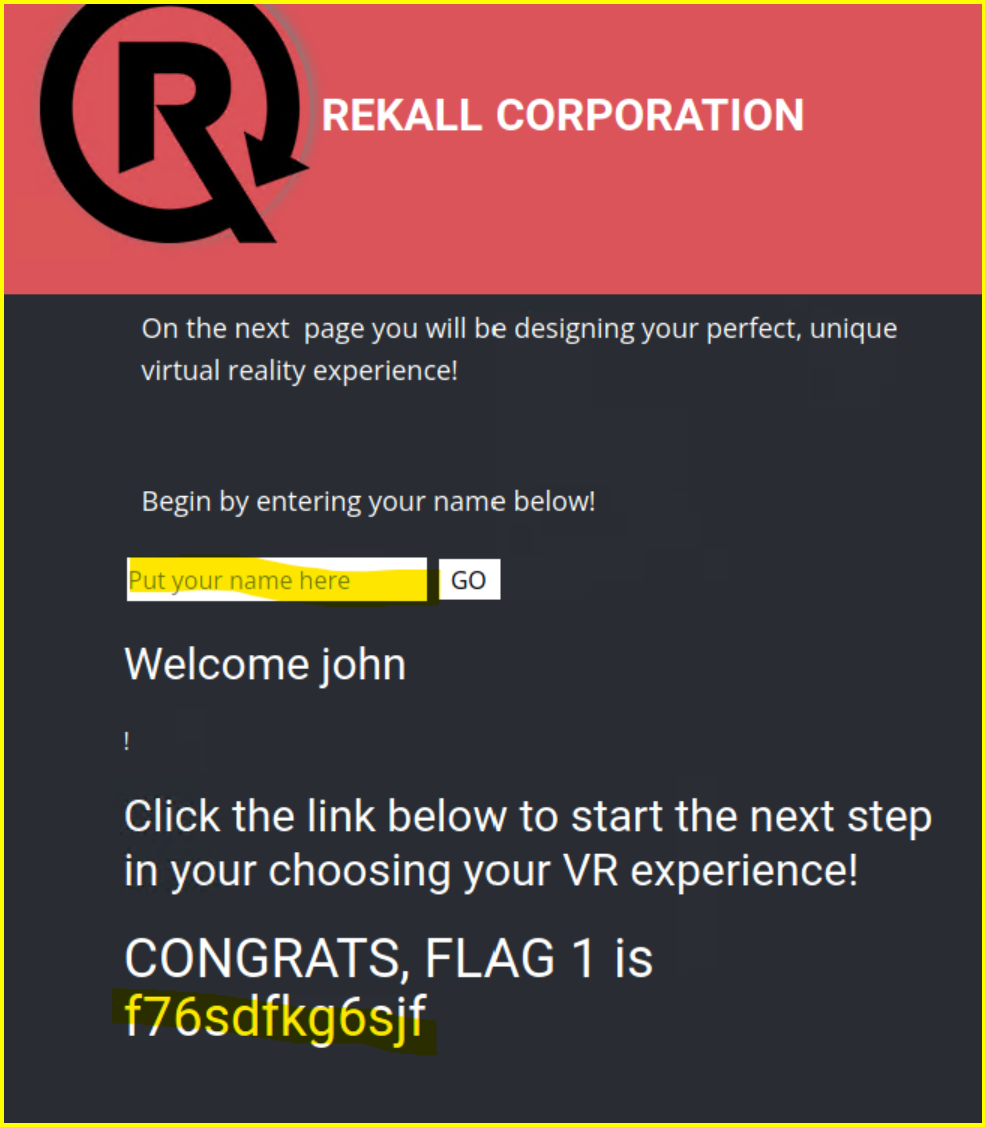
Scan Type	Total
Hosts	Windows: hxxp://172.22.117.10 hxxp://172.22.117.20 Linux: hxxp://192.168.13.10 hxxp://192.168.13.11 hxxp://192.168.13.12 hxxp://192.168.13.13 hxxp://192.168.13.14 Web Application: hxxp://192.168.14.35
Ports	Windows: 21 - ftp 25 - smtp 79 - finger SLMail 80 - http 88 - kerberos-sec 106 - pop3pw 110 - pop3 135 - msrpc 139 - netbios-ssn 389- ldap 443 - ssl/http 445 - microsoft-ds (SMB) 464 - kpasswd5? 593 - ncacn_http 636 - tcpwrapped Linux: 8009 - ajp13 Apache Jserv 8080 - Http Apache Tomcat 80 - http Apache httpd 2.4.25 Debian 22 - SSH - OpenSSH Web Application: 80 - http Apache 2.4.7 Ubuntu 3305 mysql 5.5.47

Exploitation Risk	Total
Critical	11
High	8
Medium	6

Low	0

Vulnerability Findings

Vulnerability 1	Findings
Title	Reflected XSS / Stored XSS
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	Critical
Description	Several of the intake forms and URLS were subject to Cross-Site Scripting techniques. While some had filters in place, our team was able to bypass them with various escape techniques. The URL and form on the Welcome page had no filters and simply accepted the script we entered. On the memory planner page we found an intake form “Who Do You Want To Be?” This form had a filter for the word script but we were able to bypass it utilizing this escape format : payload=<sscriptcript>alert("XSS")<%2Fsscriptcript> The comments page contains an entry form that allows for unfiltered Cross-Site Scripting. This area allows for stored XSS, so once the script is taken it remains persistent. The example shown has a script running to reveal a session cookie.
Images	**Flag 1:**

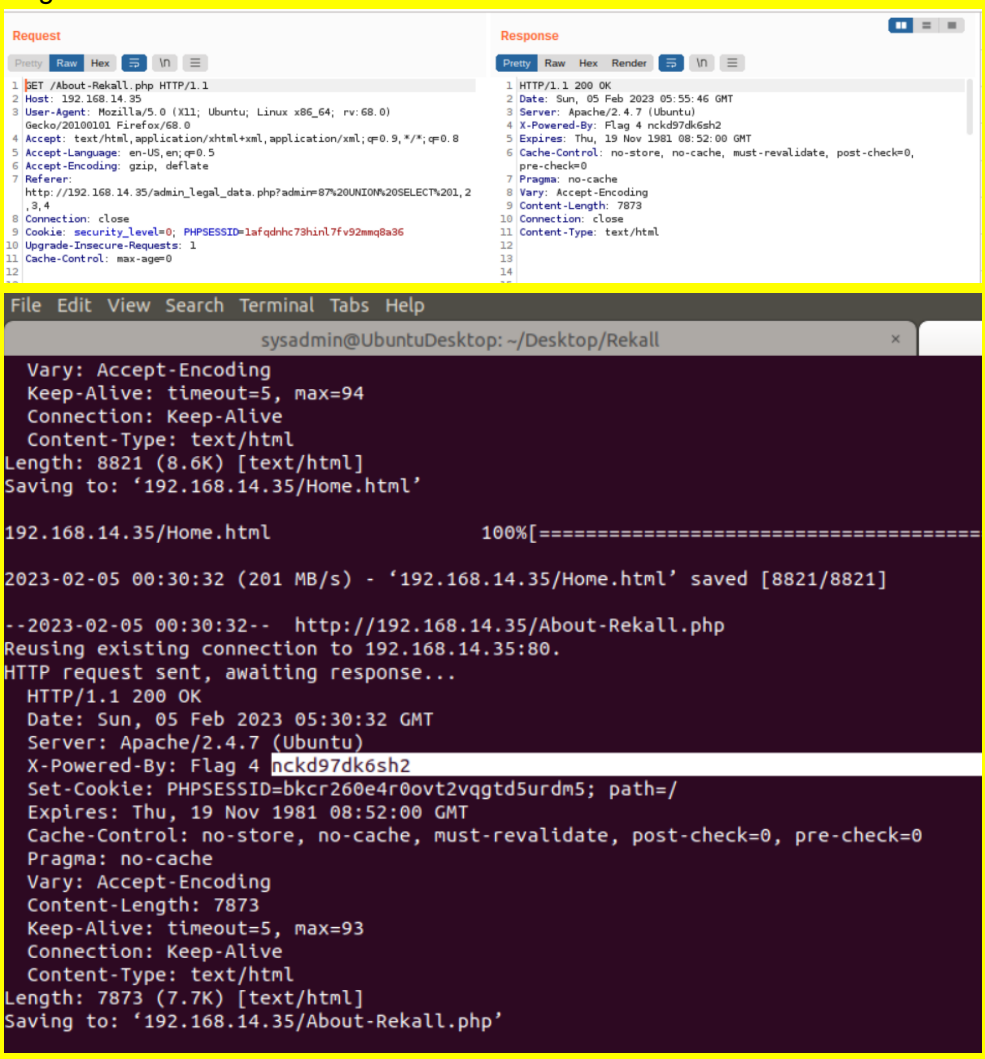



```
<form>
    <input type="text" name="payload" placeholder="Put your name here">
    <input type="submit" value="GO">
</form>
</p>
</body>

<h3>Welcome john</h3><script>alert('you've been hacked)</script>!</h3><h3>Click the link below to start the next step in your choosing your VR experience!</h3><h2>CONGRATS, FLAG 1 is f76sdfkg6sjf</b></h2>
```

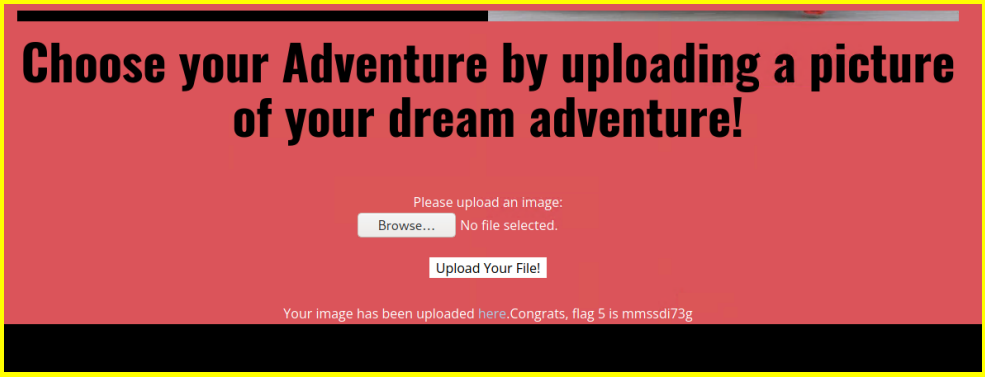
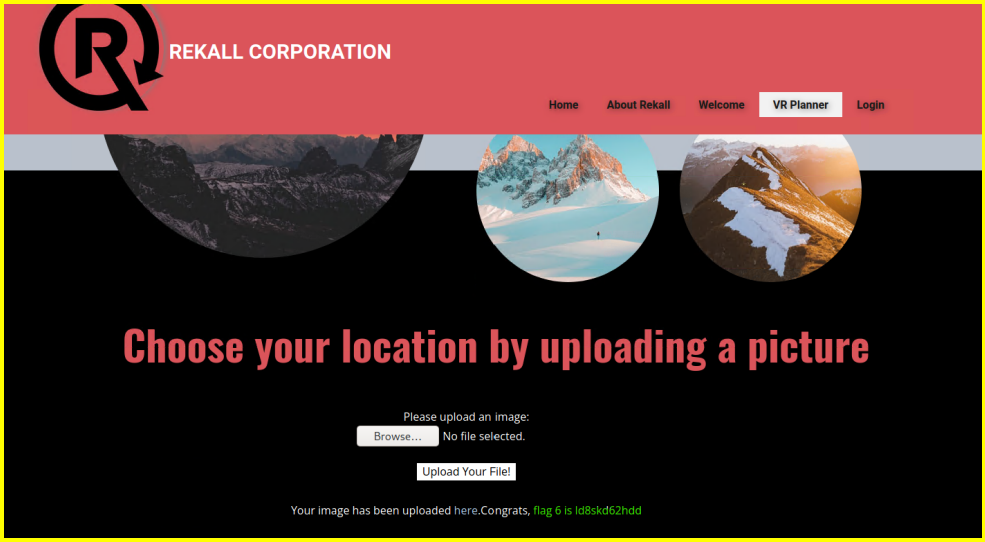
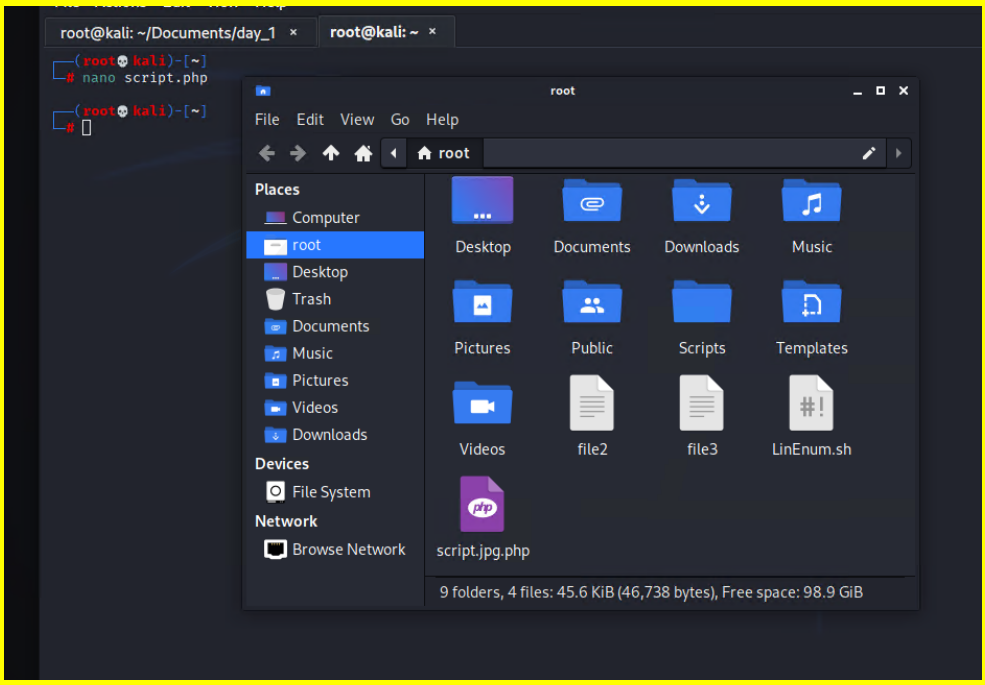
Flag 2:

	Flag 3: 
Affected Hosts	hxxp://192.168.14.35/Welcome.php hxxp://192.168.14.35/Memory-Planner.php
Remediation	• Input Validation and sanitization as rigidly as possible to only what is expected in the form or URL • Utilize Content-Type header to limit what can be requested or posted via scripting.

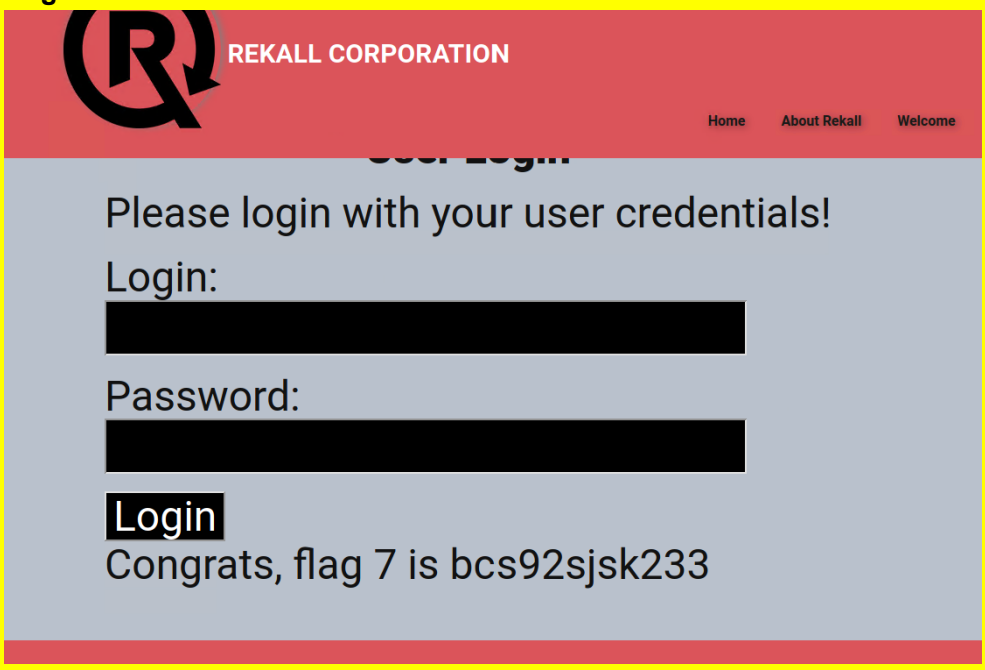
Vulnerability 2	Findings
Title	Unsecured HTTP Response Headers
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Medium
Description	After exploring and abusing the HTTP protocol, one of the pages on the site had sensitive information in the response header. This is a risk of varying severity. It depends on what is stored here by the developers of the website. With that being said if not taken into account it can be abused and could potentially be used to steal session cookies. The page in question is the About-Rekall.php page.

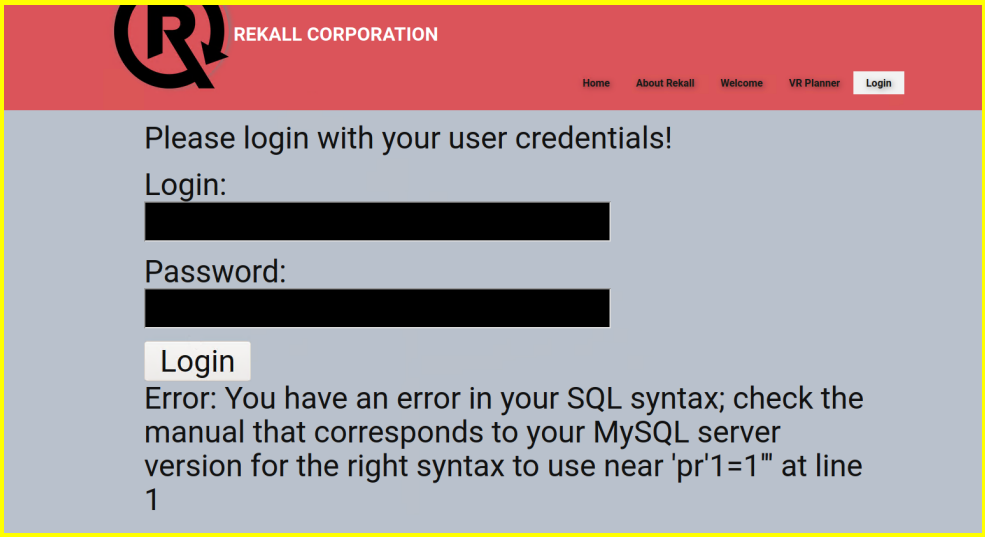
Images	Flag 4: 
Affected Hosts	hxxp://192.168.14.35/About-Rekall.php
Remediation	• Encode Output data • Limit sensitive data stored in HTTP

Vulnerability 3	Findings
Title	Local File Inclusion (LFI)
Type (Web app / Linux OS / WIndows OS)	Web APP
Risk Rating	Critical
Description	The Memory Planner page has 2 upload areas for images. However the first area has no restrictions on what can be uploaded. A simple php script was uploaded to reveal this vulnerability. The upload form on the bottom on the page has a tighter grasp of what can be

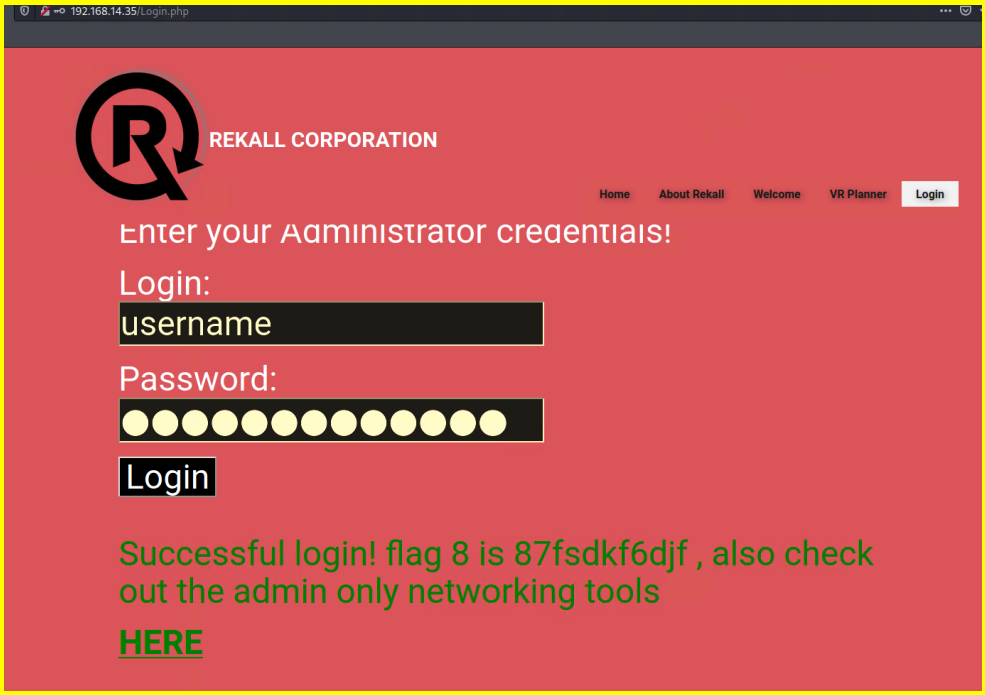
	uploaded and is limited to JPG files. However, we were able to mask our script name utilizing "Script.jpg.php" Both of these areas are vulnerable to an attacker uploading an executable script or file which is ultimately a critical situation.
Images	**Flag 5:**  **Flag 6:**  

Affected Hosts	hxxp://192.168.14.35/Memory-Planner.php
Remediation	• Whitelist file types that can be uploaded • Monitor HTTP header before accepting files • Rigid input validation on upload forms.

Vulnerability 4	**Findings**
Title	SQL Injection
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	On the Rekall Login Portal, the initial login form is subject to SQL injection. Utilizing a common injection technique we were able to submit a true statement with a dummy password. This true statement allowed us to gain entry to the login form without knowing the password. Example: FakePassword'OR '1'='1' Shown in the second picture below, the form even delivered feedback to our attempts to trigger the SQL injection vulnerability. This type of feedback can be powerful to an attacker deciding on which route to try to use in exploitation
Images	**Flag 7:** 

	
Affected Hosts	hxxp://192.168.14.35/Login.php
Remediation	• Rigid input validation • Apply character limit on password forms, complex injections require a lot of text sent. • Utilize prepared statements.

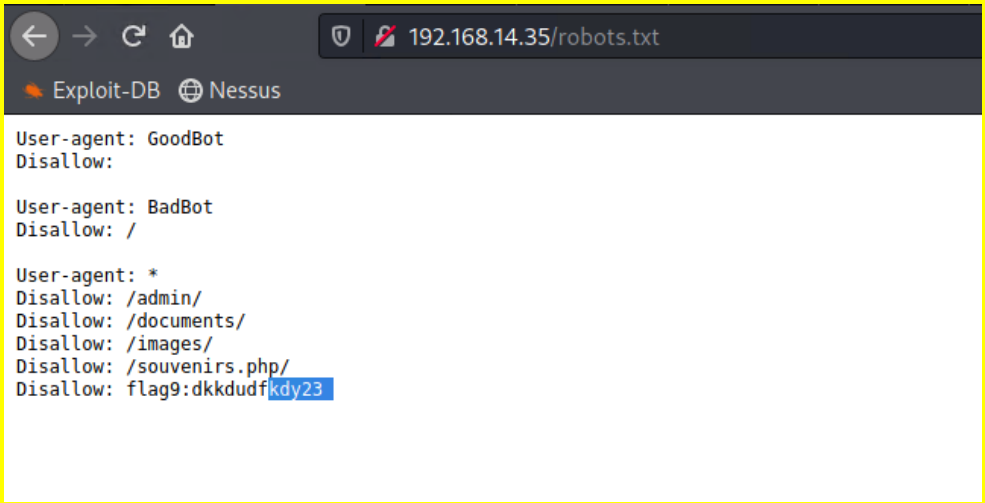
Vulnerability 5	Findings
Title	Exposure of Sensitive Data in HTML
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	On the bottom of the login page, we found and exploited the admin portal. After analyzing the Page Source HTML we found the user DougQuaid's credentials in plain sight. There was no real exploit here other than exploring. This credential is accessible to anyone who might explore the source code.
Images	**Flag 8:**



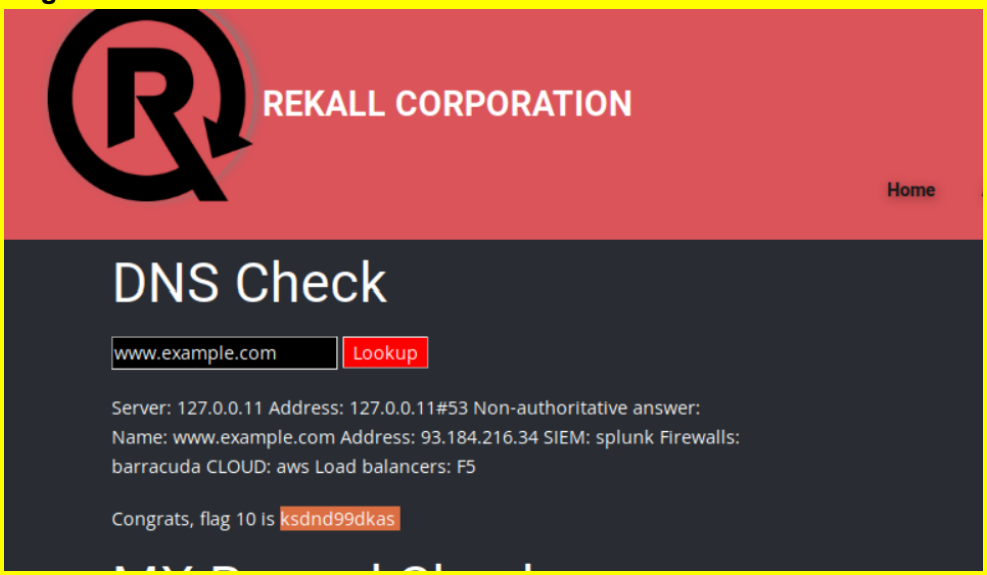
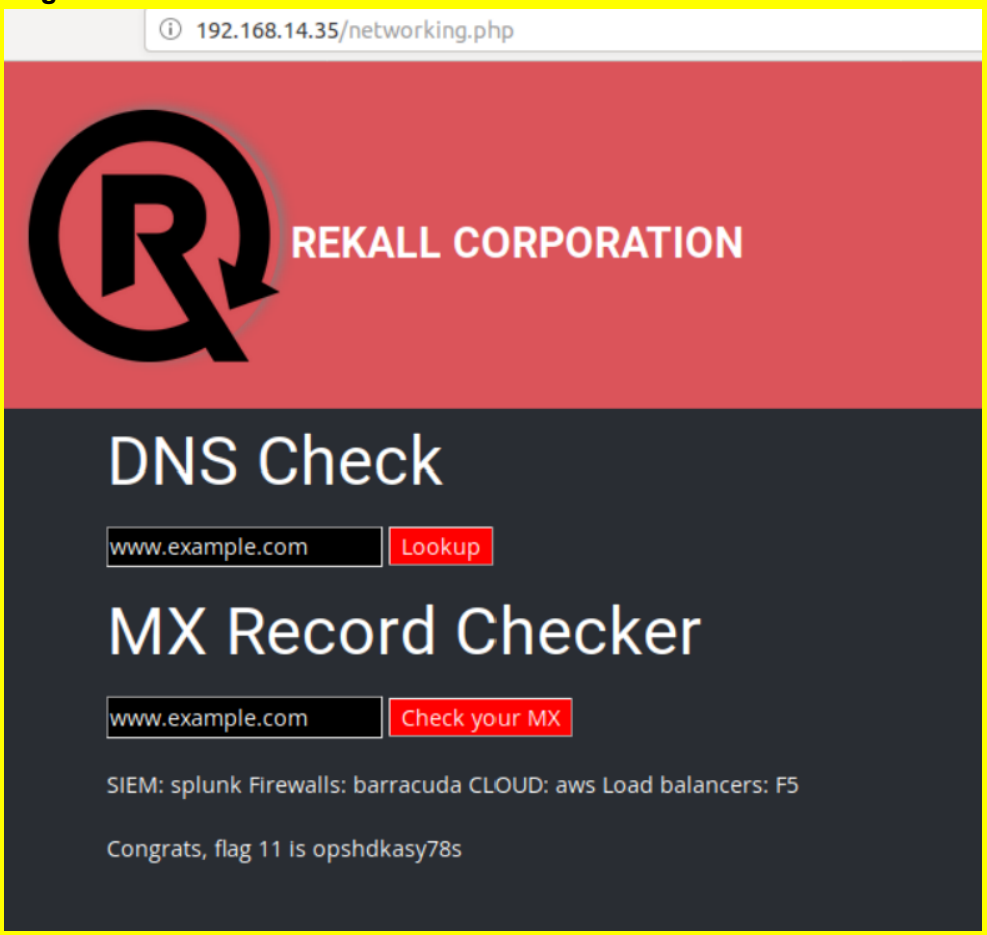

```
<div id="main">

    <p>Enter your Administrator credentials!</p>

<style>
input[type=text], input[type=password]{
  background-color: black;
  color: white;
}
button[type=submit]{
  background-color: black;
  color: white;
}
</style>

    <form action="/Login.php" method="POST">

        <p><label for="login">Login:</label><font color="#DB545A">dougquaid</font><br />
        <input type="text" id="login" name="login" size="20" /></p>

        <p><label for="password">Password:</label><font color="#DB545A">kuato</font><br />
        <input type="password" id="password" name="password" size="20" /></p>

        <button type="submit" name="form" value="submit" background-color="black">Login</button>

    </form>

    </br >
    <font color="red">Invalid credentials!</font>
</div>


</body>

</html>
```

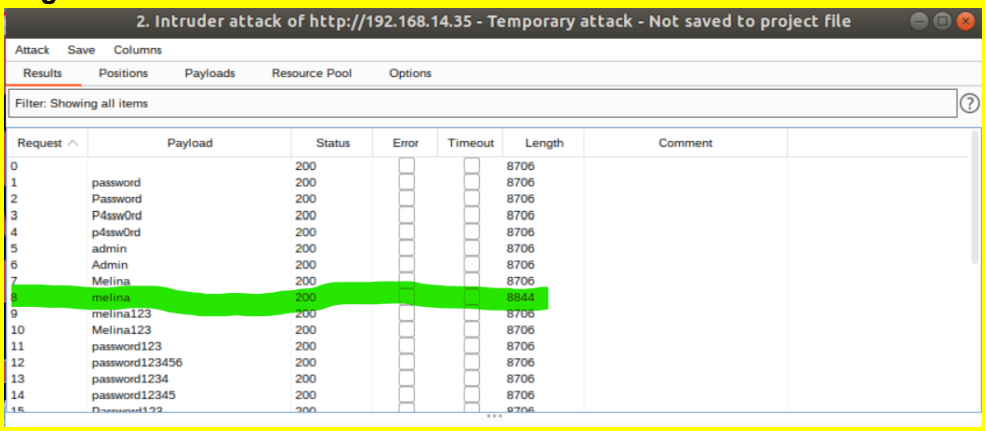
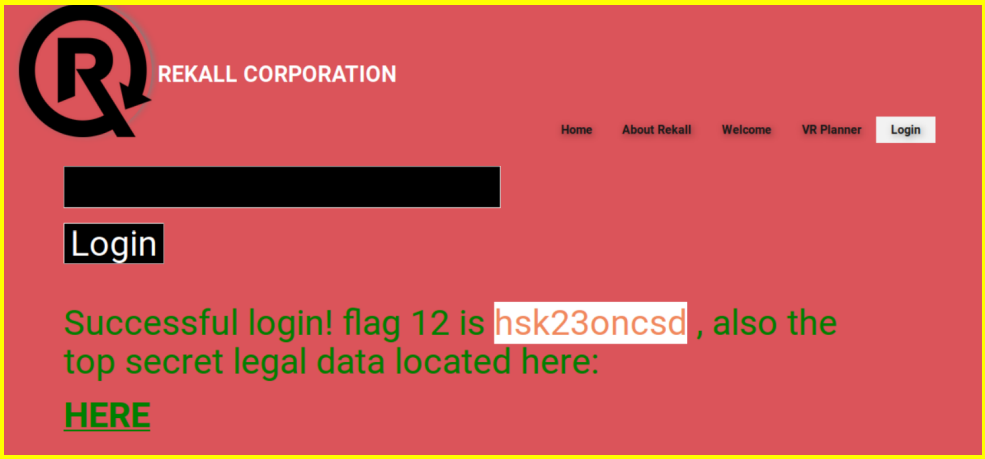
Affected Hosts	hxxp://192.168.14.35/Login.php
Remediation	• Do not store passwords in HTML

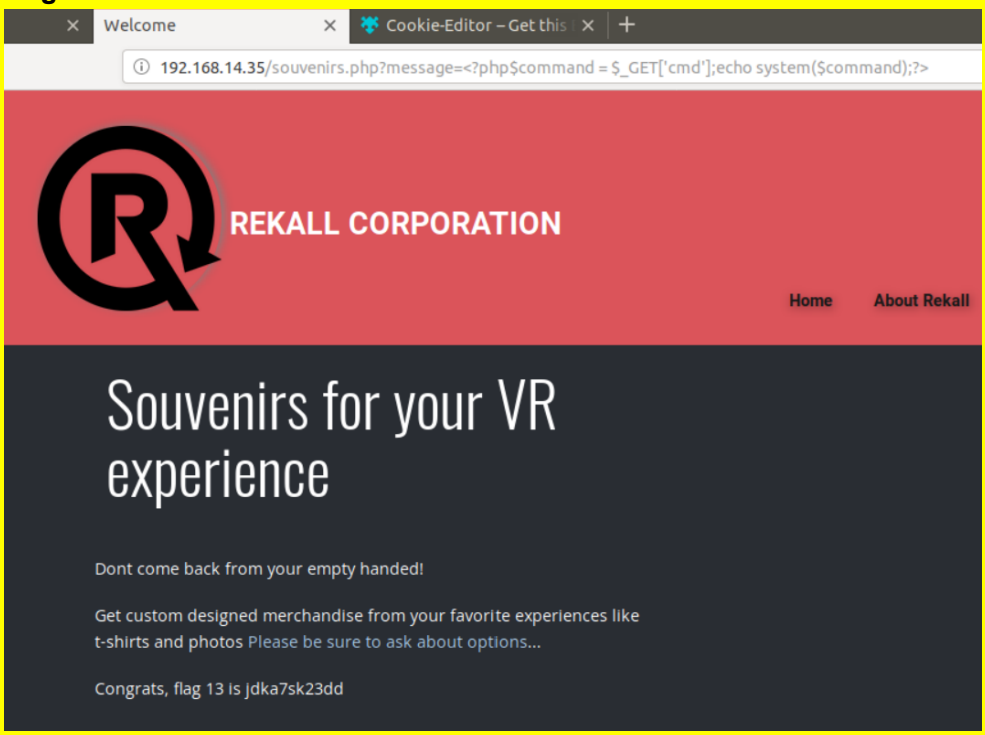
Vulnerability 6	Findings
Title	Sensitive Data Exposure (Robots.txt)
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Medium

Description	The robots.txt file was found and it contained data regarding the website. Including the Souvenirs.php page. This page was later exploited. However the robot.txt page is not necessarily a vulnerability, however what is kept here must have strict permissions applied and developers should limit sensitive data stored in this file.
Image	**Flag 9:** 
Affected Hosts	hxxp://192.168.14.35/robots.txt
Remediation	• limit sensitive data stored in robots.txt file • apply proper access controls to items that may be accessible through robots.txt findings.

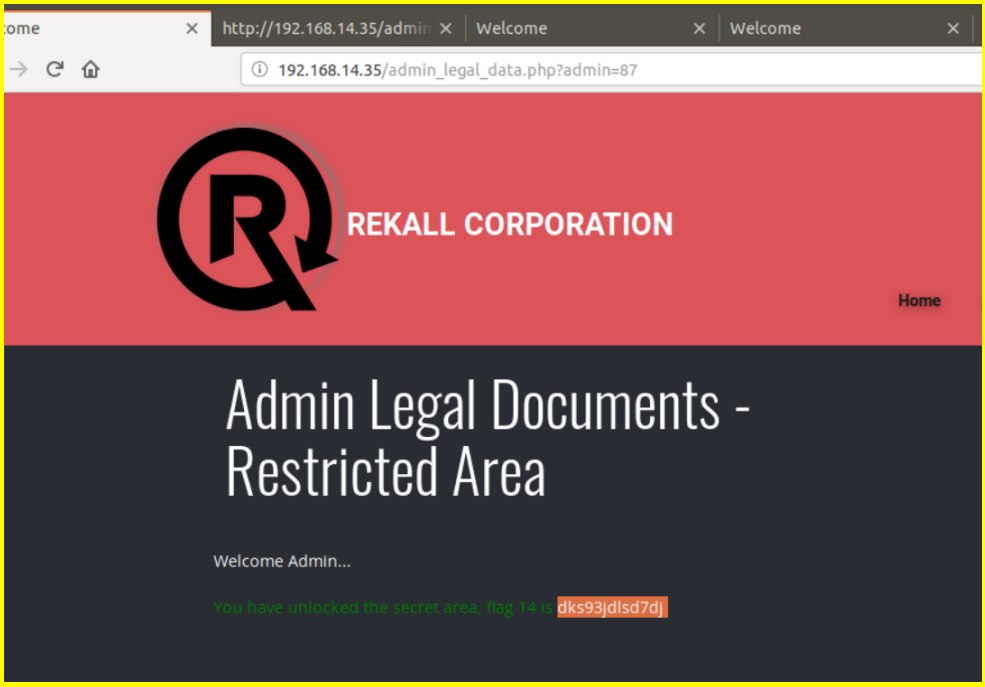
Vulnerability 7	Findings
Title	Command Injection + Directory Traversal + Exposure of Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	After accessing Doug Quaids account we were led to an admin only page called Networking.php. While this page was intended to be for admins only, it was accessible to anyone who knew of the URL. While we were there we discovered 2 input forms: DNS Checker and MX Checker. Both of these forms were subject to command injection utilizing the && notation and the | notation respectively. Through both of these forms we could traverse directories and read important files such as etc/passwd and etc/host. The Passwd file gave us light to a user named Melina which we would exploit later. What was most peculiar is directly on the page was a header that told us directly "Just a reminder, the vendor list of our top-secret networking tools are located in the file: vendors.txt " with this information we were able to read the file on both forms to reveal sensitive information about Rekall.

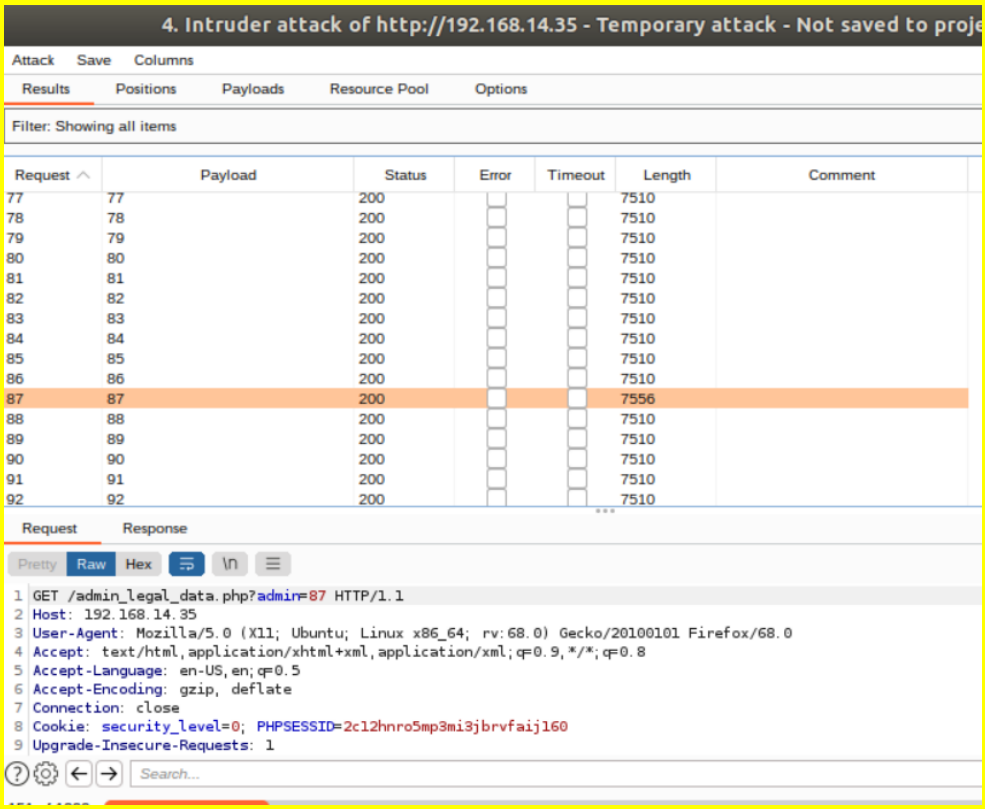
Images	**Flag 10:**  **Flag 11:** 
Affected Hosts	hxxp://192.168.14.35/Networking.php
Remediation	• Whitelist allowable characters and input • Limit input to alphabet characters

Vulnerability 8	Findings
Title	Weak Admin Passwords
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Through previous findings allowing us to read the etc/passwd folder, we found a user named Melina. Through password guessing techniques we were able to guess her password and gain access to yet another administrator account. Melina's password was incredibly weak, given it was simply her name “melina” Entry to melina’s account led the way to another page regarding legal data about Rekall Corp.
Images	**Flag 12:**  
Affected Hosts	hxxp://192.168.14.35/Login.php
Remediation	• Increase password requirements • 12 character length • Must contain uppercase, lowercase, numbers and special characters • passwords must change every 4-6 months

Vulnerability 9	Findings
Title	PHP Injection
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	The Souvenirs.php page which was revealed to use by the robots.txt file was vulnerable to php injection in the URL. Here we were able to build onto the message parameter to inject a php script. The script we executed is one that would allow command execution. Given the power granted with such a script from here an attacker could then do a lot of damage with command execution on the app.
Images	**Flag 13:** 
Affected Hosts	hxxp://192.168.14.35/Souvenirs.php
Remediation	• eliminate dynamic code execution • adopt secure code techniques in development • utilize security analysis tools for code in development

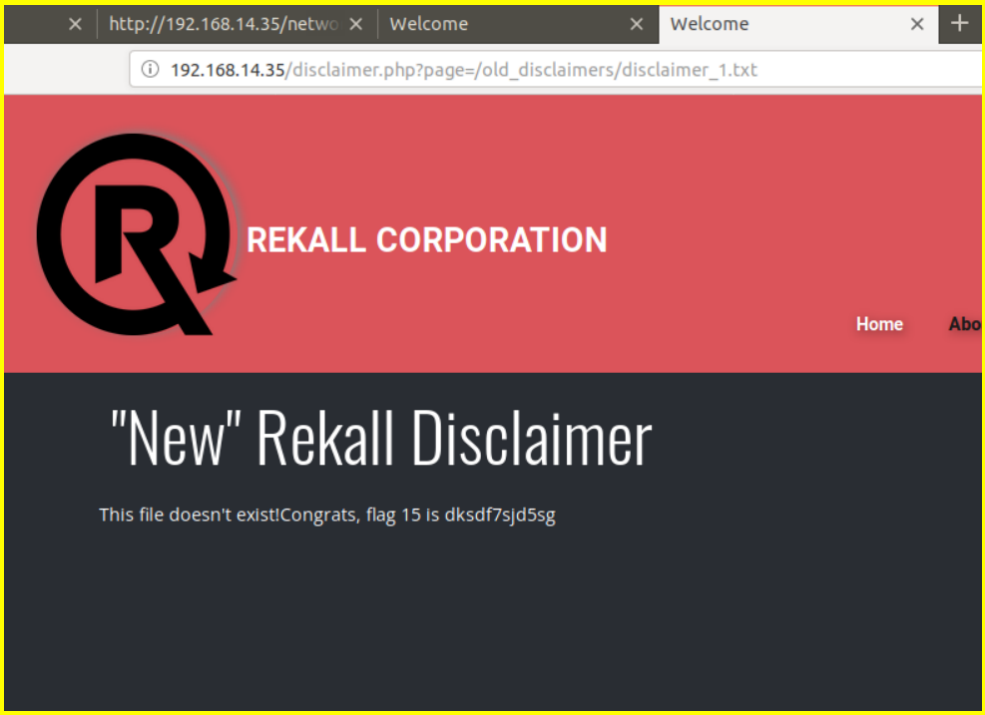
Vulnerability 10	Findings
Title	Broken Access Control

Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Entering Melina's account directed us to the Admin Legal Documents- Restricted Area. At first this page was blocked with a warning for Admins Only. However the URL was able to be manipulated by simply putting in the correct admin id number. Utilizing Burp Suite Intruder we could quickly run numbers through the field. We found that Admin=87 was a success which allowed us access to this proprietary data. This page specifically shows broken authentication and session management.
Images	**Flag 14:** 



Affected Hosts	hxxp://192.168.14.35/Admin_legal_data.php
Remediation	• Apply a default:deny security practice. • Log and monitor access control failures • Implement server-side access control to ensure access validation. • Remove session ID from being present in the URL

Vulnerability 11	**Findings**
Title	Directory Traversal + Exposure of Old Sensitive Data
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Medium
Description	The Disclaimer page gave reference to the current disclaimer being the “new” version. With a little bit of exploring we could utilize the command injection from the DNS check page to show all the pages and directories in the /app directory. In this directory was another directory named old_disclaimers. Back on the disclaimers page, the page parameter in the URL allowed us to perform directory traversal to read the disclaimer_1.txt file in the old_disclaimer directory. Not only was this an exposure of sensitive data but it was also data that could be used against Rekall Corp in a lawsuit given the side effects from the product.
Images	**Flag 15:**

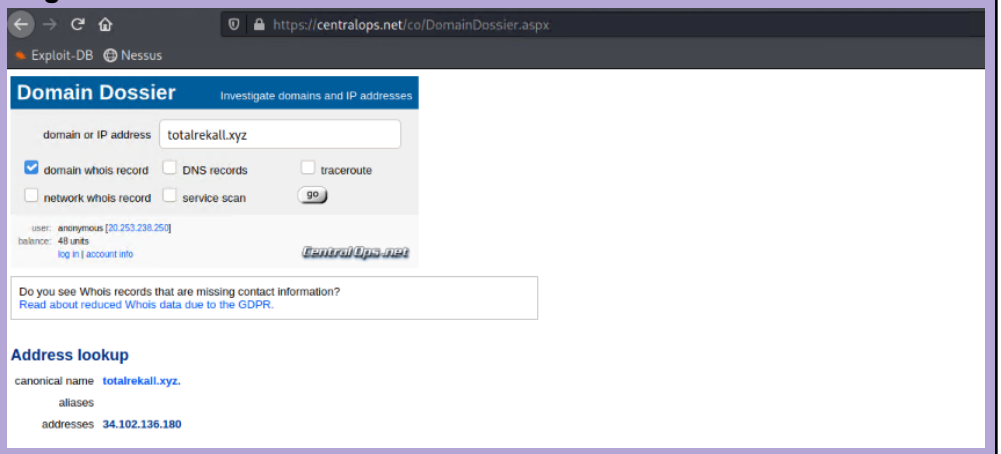
	
Affected Hosts	hxxp://192.168.14.35/disclaimer.php
Remediation	• Avoid sending user input to the application API • If this is necessary for functionality, the Rekall web app must validate the input and allow only the base file path required. For example, the disclaimer page should only allow input from the /disclaimer base file path.

Vulnerability 12	Findings
Title	Exposure of Linux Networking Information and Credentials
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Medium
Description	Various pieces of information were available through a WHOIS search on a domain dossier. We found the IP address of totalrekall.xyz, the address of the registrant, and a username of an SSH user. The latter being of the most concern. Utilizing Certificate Transparency we also found information on various sites being used for the Rekall Corporation. A Nessus Scan revealed to us a potential critical vulnerability that was exploited later on in our findings. An aggressive NMAP scan revealed a CVE for the Drupal services which was exploited as well.
Images	**Flag 1:**

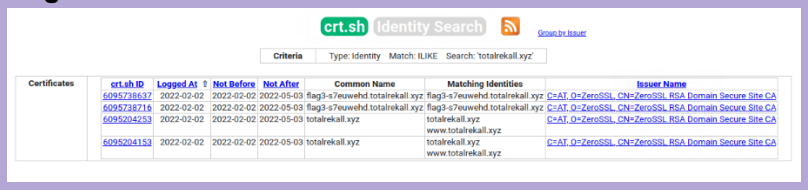
Queried whois.godaddy.com with "totalrekall.xyz"...

```
Domain Name: totalrekall.xyz
Registry Domain ID: D273189417-CNIC
Registrar WHOIS Server: whois.godaddy.com
Registrar URL: https://www.godaddy.com
Updated Date: 2023-02-03T14:04:18Z
Creation Date: 2022-02-02T19:16:16Z
Registrar Registration Expiration Date: 2024-02-02T23:59:59Z
Registrar: GoDaddy.com, LLC
Registrar IANA ID: 146
Registrar Abuse Contact Email: abuse@godaddy.com
Registrar Abuse Contact Phone: +1.4806242505
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: clientRenewProhibited https://icann.org/epp#clientRenewProhibited
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Registry Registrant ID: CR534509109
Registrant Name: sshUser alice
Registrant Organization:
Registrant Street: h8s692hskasd Flag1
Registrant City: Atlanta
Registrant State/Province: Georgia
Registrant Postal Code: 30309
```

Flag 2:



Flag 3:



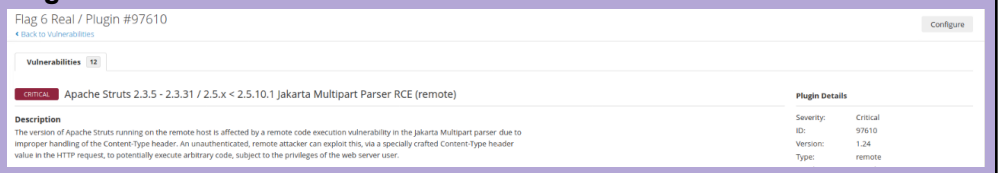
Flag 4:

```
File  Actions  Edit  View  Help
┌──(root㉿kali)-[~]
└─# nmap -sV 192.168.13.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2023-02-06 21:52 EST
Nmap scan report for 192.168.13.10
Host is up (0.000010s 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 (0.000010s 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 (0.0000090s 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 (0.000010s 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 (0.000010s 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
```

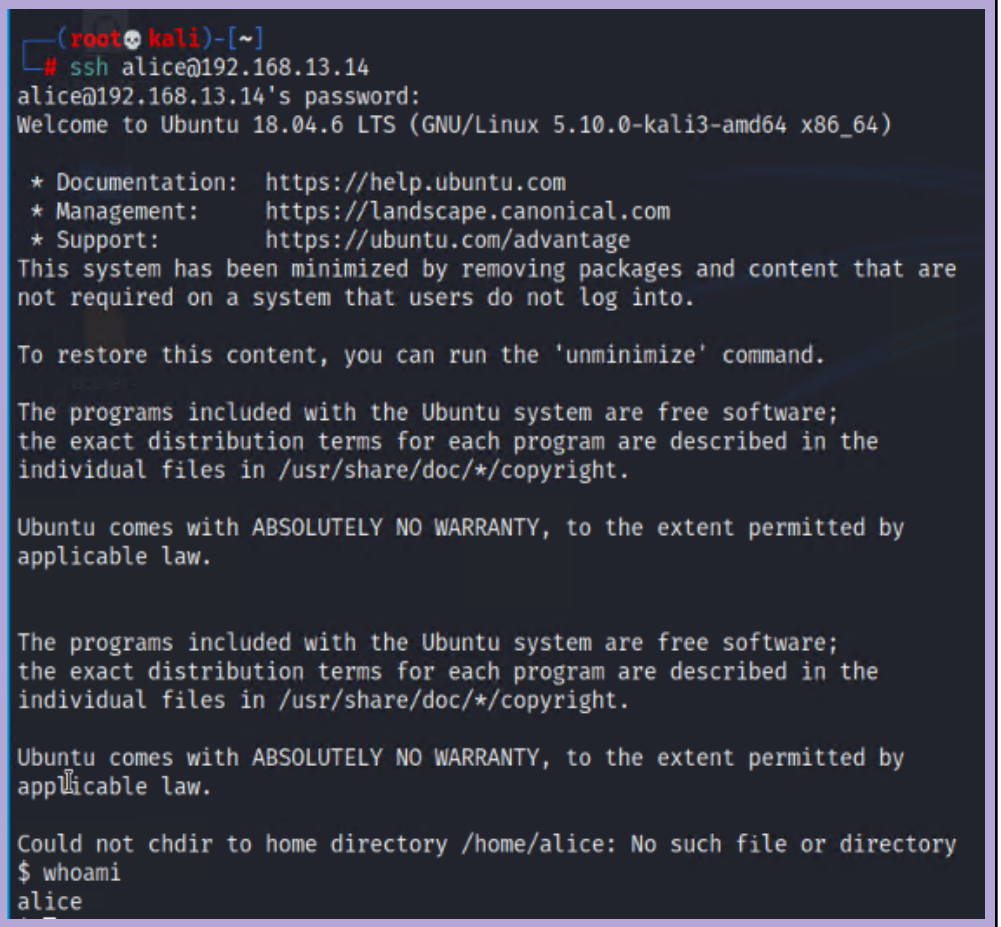
Flag 5:

```
Nmap scan report for 192.168.13.13
Host is up (0.000016s 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 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
```

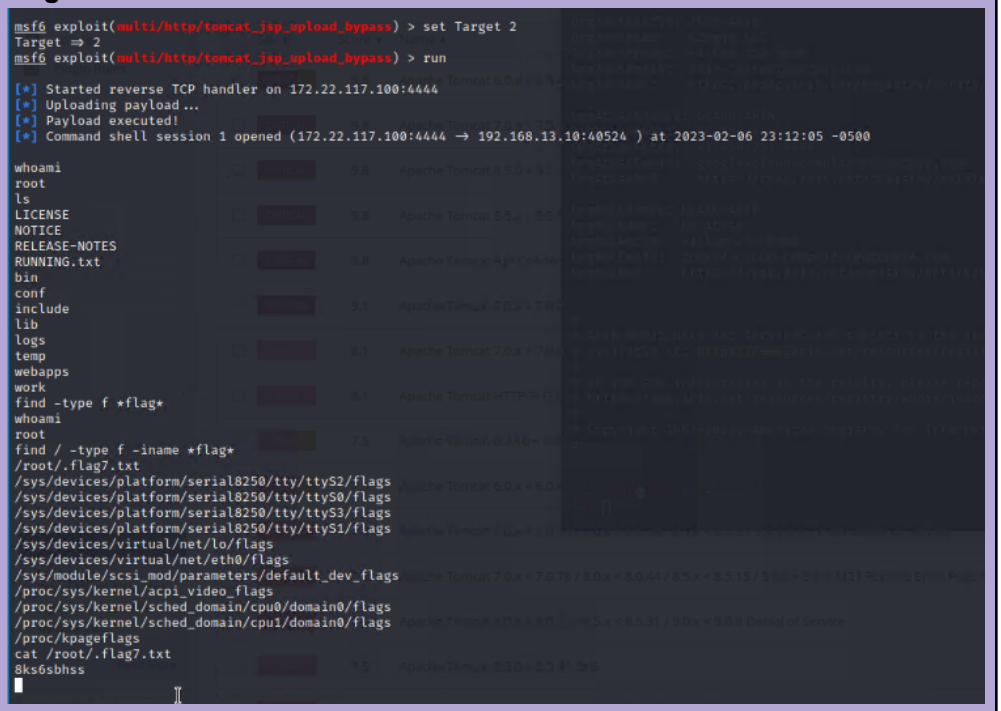
Flag 6:



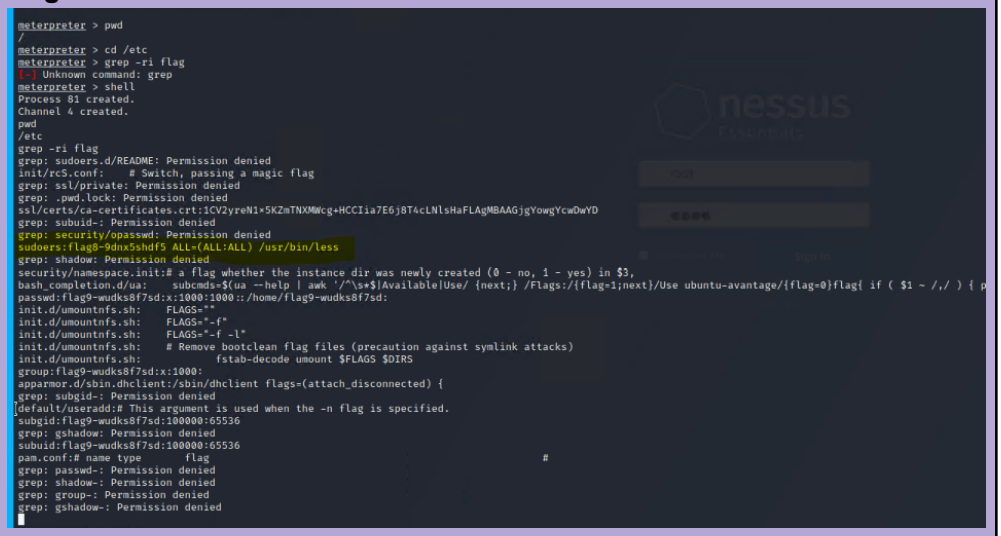
Affected Hosts	hxxp://totalrekall.xyz , hxxp:192.168.13.10-14
Remediation	• Maintain awareness of what is exposed online about the company. • Limit sensitive information displayed on OSINT sources • ensure employee passwords are strong and routinely changed if some usernames can be derived from OSINT sources. • Defense team must maintain the mindset of a hacker and understand that what can be found openly online is the first point of developing an attack.

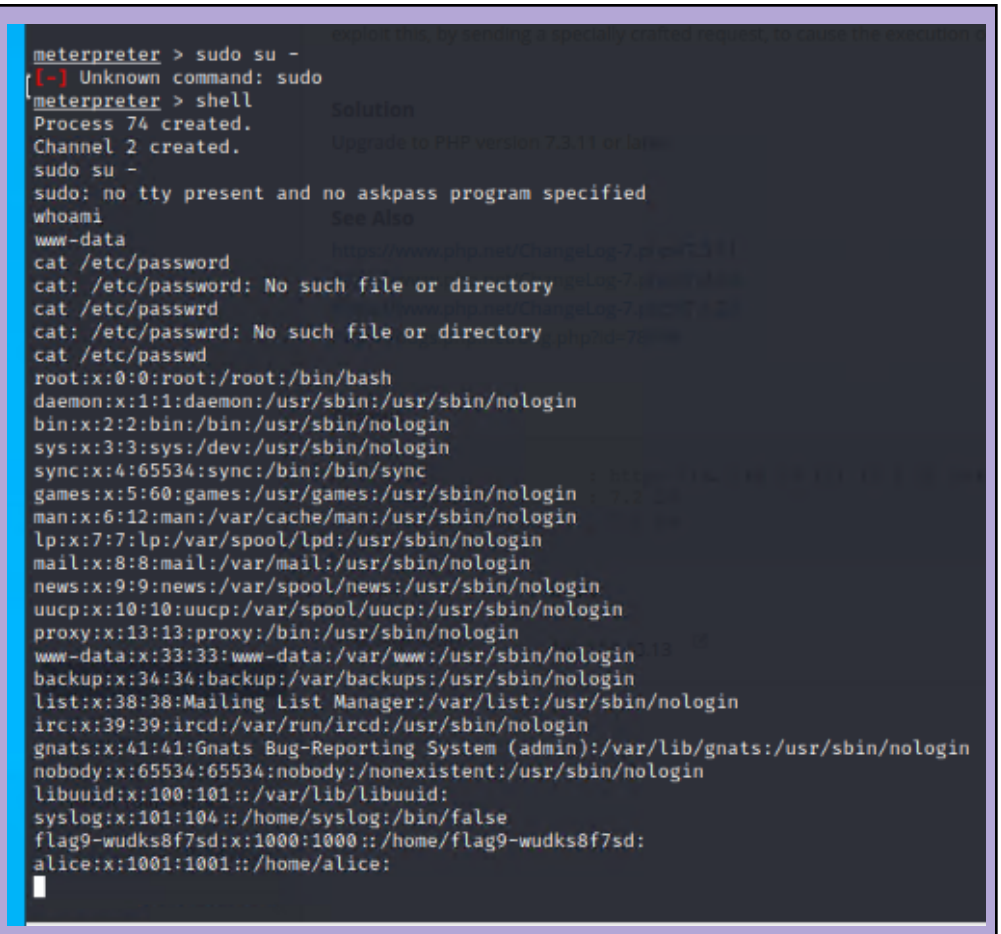
Vulnerability 13	Findings
Title	Weak User Passwords
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	The SSH user, Alice, was one of the only users we were able to find on the system. However, this was all we needed to enter the Linux System at 172.22.117.14. We were successfully able to SSH to her machine with the password of “alice”. This allowed us to exploit the machine further in post exploitation.
Images	
Affected Hosts	hxxp://192.169.13.14
Remediation	• Increase employee password requirements. • 12 character length • Must contain uppercase, lowercase, numbers and special characters • passwords must change every 4-6 months

Vulnerability 14	Findings

Title	Apache Tomache JSP Bypass CVE-2017-12617
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	This specific exploit utilizes a PUT request in order to bypass authentication and upload a jsp shell to an Apache tomcat server. Any code in the JSP could then be executed. Metasploit Module: /multi/http/tomcat_jsp_upload_bypass
Images	**Flag 7:** 
Affected Hosts	hxxp://192.168.13.10
Remediation	• This exploit is possible when running Apache Tomcat versions 9.0.0.M1 to 9.0.0, 8.5.0 to 8.5.22, 8.0.0.RC1 to 8.0.46 and 7.0.0 to 7.0.81 • It recommends upgrading to a version beyond and out of this vulnerable range. • Additional measures can be taken to introduce alerts on irregular file uploads on the system to potentially detect an intrusion as soon as possible.

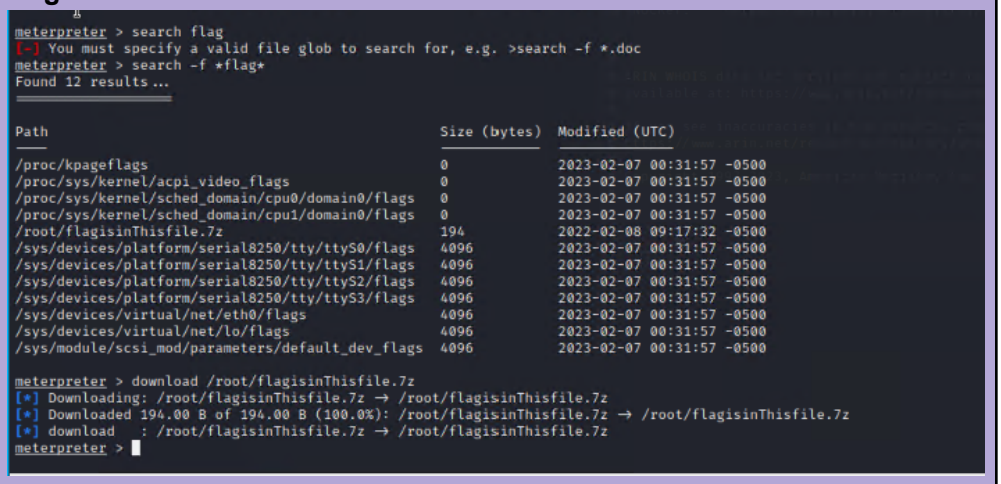
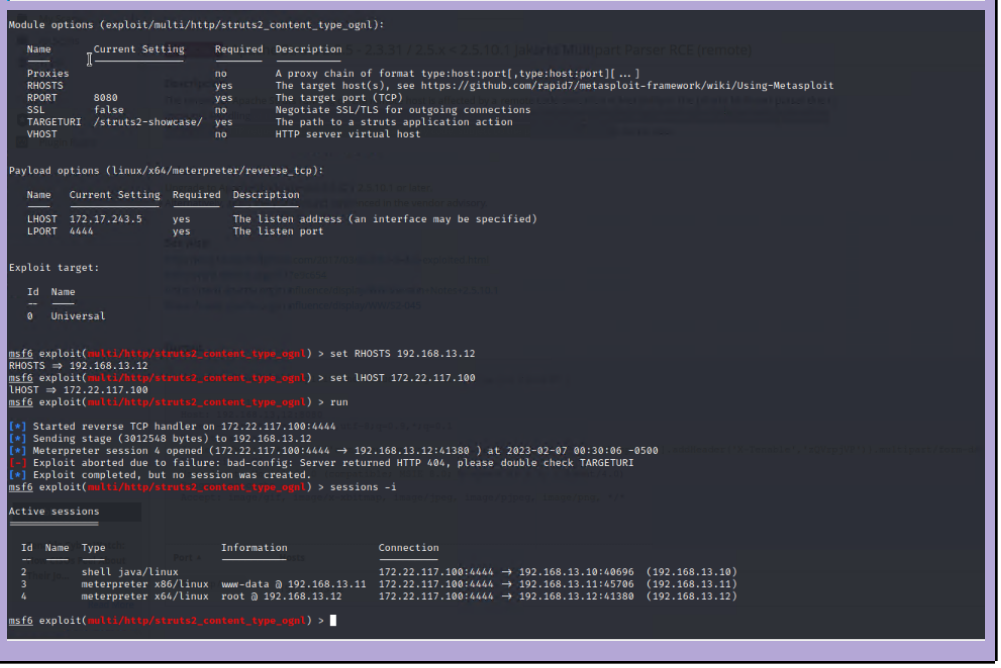
Vulnerability 15	Findings
Title	Shellshock Exploit CVE-2014-6271 CVE-2014-6228

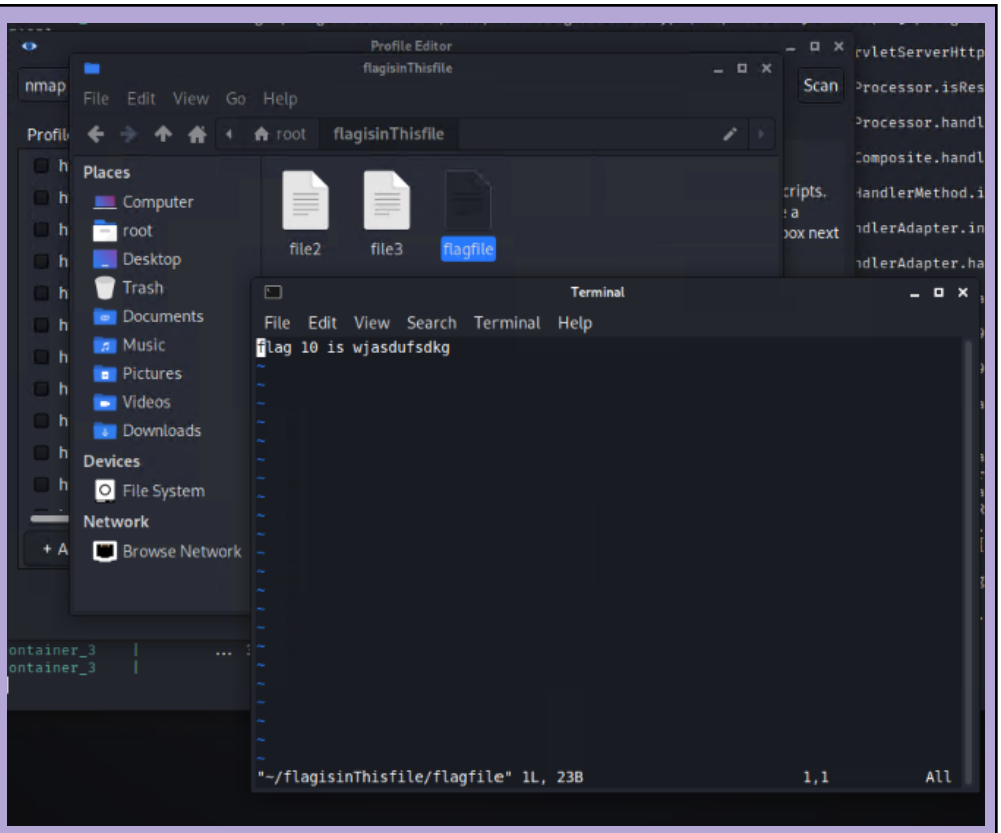
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	This host had a vulnerability to how a bash shell deals with CGI scripts in Apache. We were able to send our own CGI script to perform a Shellshock attack. While inside we had privileges to view the etc directory. Specifically we could view all files except the shadow file. A screenshot is below of the sensitive information contained in the sudoers file and the passwd file. Metasploit module: multi/http/apache_mod_cgi_bash_env_exec
Images	**Flag 8**   **Flag 9:**



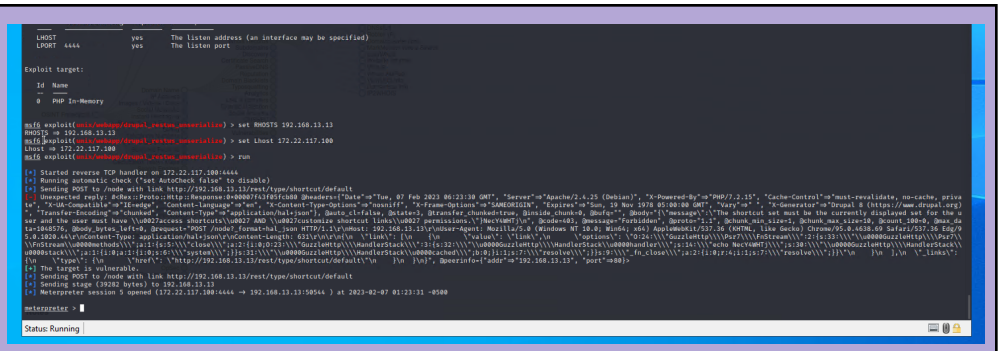
Affected Hosts	hxxp://192.168.13.11
Remediation	• upgrade to latest GNU Bash shell

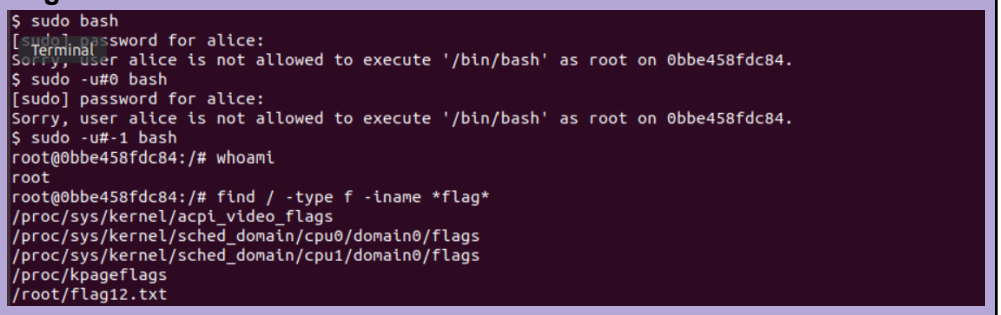
Vulnerability 16	Findings
Title	Apache Jakarta Struts Exploit CVE-2017-5638
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	Apache Struts version 2.3.5-2.3.31 and 2.5-2.5.10 are vulnerable to remote code execution. In this situation we can perform RCE through the HTTP content type header. This allows for executables to drop into the temp directory. We were able to locate a 7zip file and download it into our local host. This file

	contained sensitive company data. Metasploit Module: multi/http/struts2_Content_type_ognl
Images	**Flag 10**  

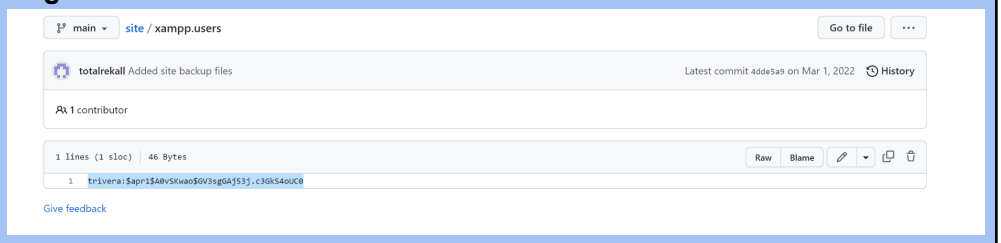
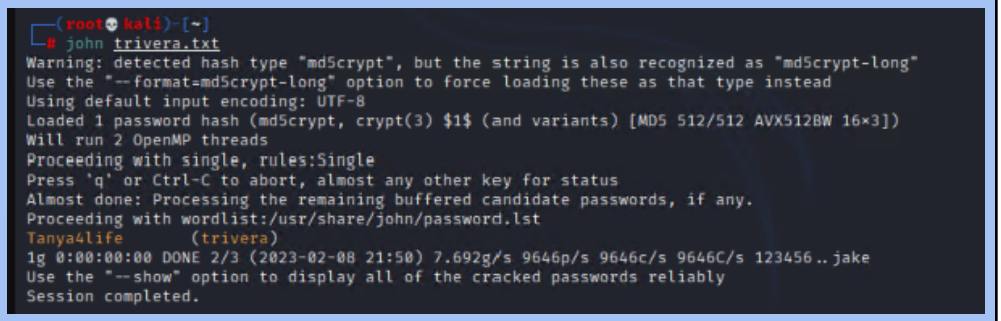
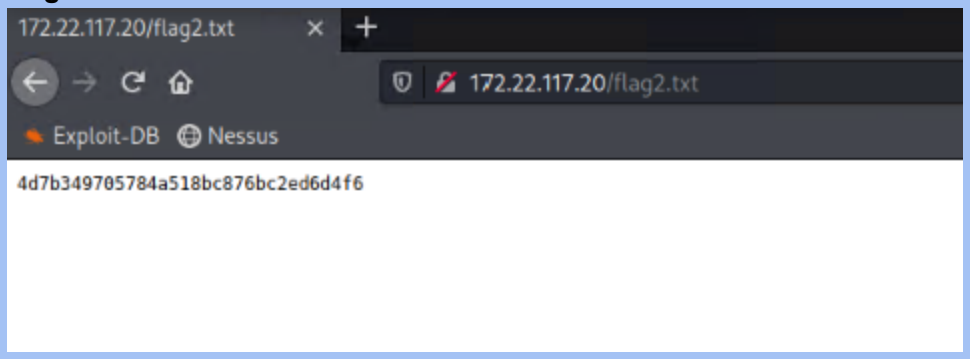
	
Affected Hosts	hxxp://192.168.13.12
Remediation	• upgrading Struts to a different parser can alleviate this vulnerability. • Utilize a Web Application Firewall. Set rules to allow valid content types. • Set another WAF rule to ban OGNL expressions. • Monitor for file downloads to unknown IP addresses to prevent extraction of files to unauthorized locations.

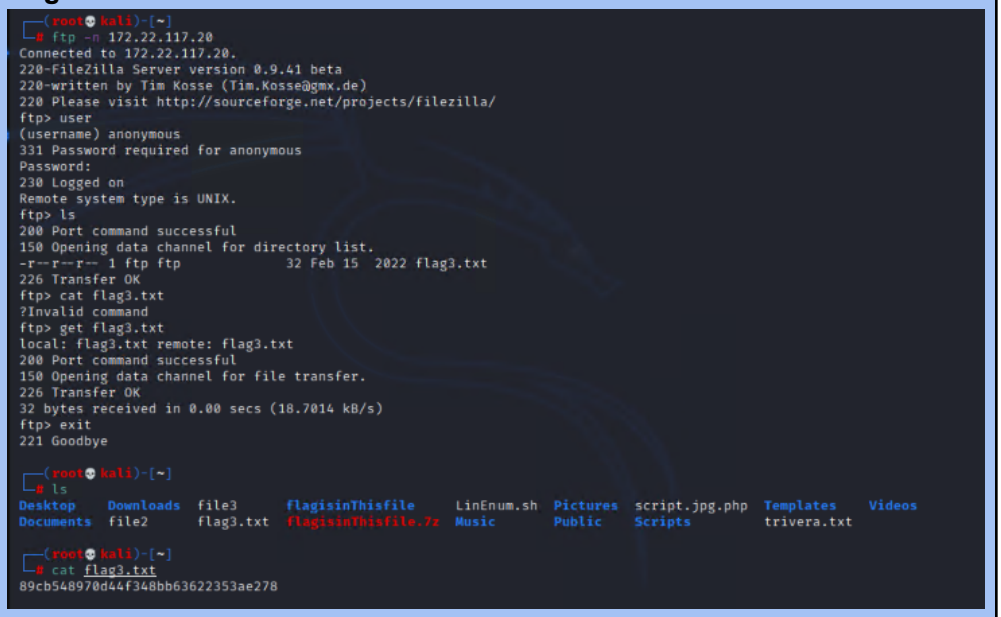
Vulnerability 17	Findings
Title	Drupal Exploit CVE-2019-6340
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	Exploitable through a PHP unserialize() vulnerability in Drupal RESTful Web Services. Metasploit can send a request to the /node REST endpoint. This is specific to when the REST API option is enabled. Metasploit Module: /unix/webapp/drupal_restws_unserialize
Images	**Flag 11**

	
Affected Hosts	hxxp://192.168.13.13
Remediation	• Recommended upgrade to version with security patches. Common versions with the security update at 8.6.10 and 8.5.11

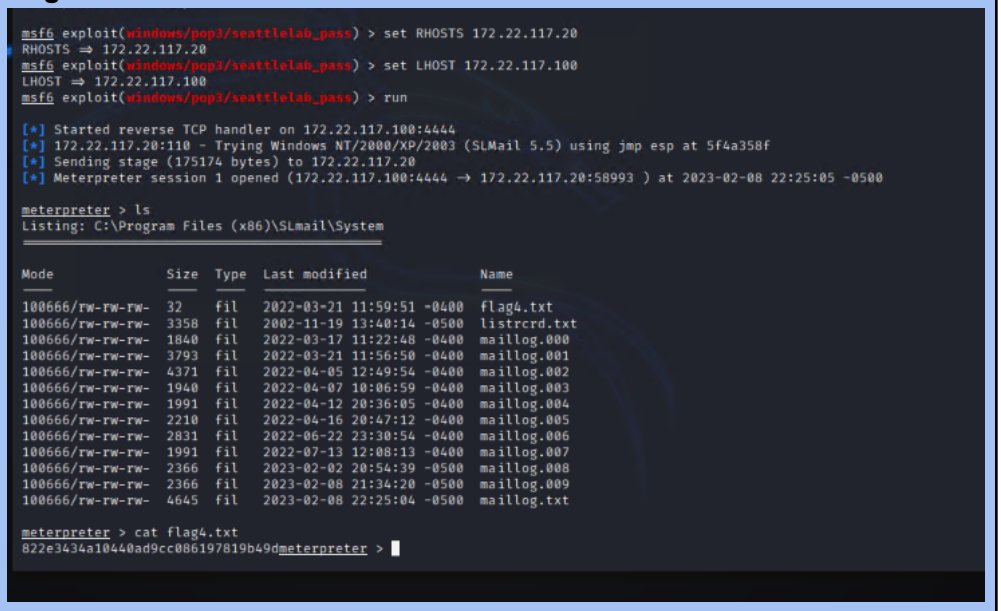
Vulnerability 18	**Findings**
Title	Sudo Exploit CVE-2019-14287
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	In versions of Sudo prior to 1.8.28 a user with Sudoers access (All:All) can bypass policies through an incorrect attempt to login. By sending a custom login ID we can bypass authentication and gain root access.
Images	**Flag 12** 
Affected Hosts	hxxp://192.168.13.14
Remediation	• check sudo version with sudo --version, then update sudo version • do not utilize (All:All) access on sudo. Implement least privilege protocol. • Blacklist executables on sudo command.

Vulnerability 19	**Findings**

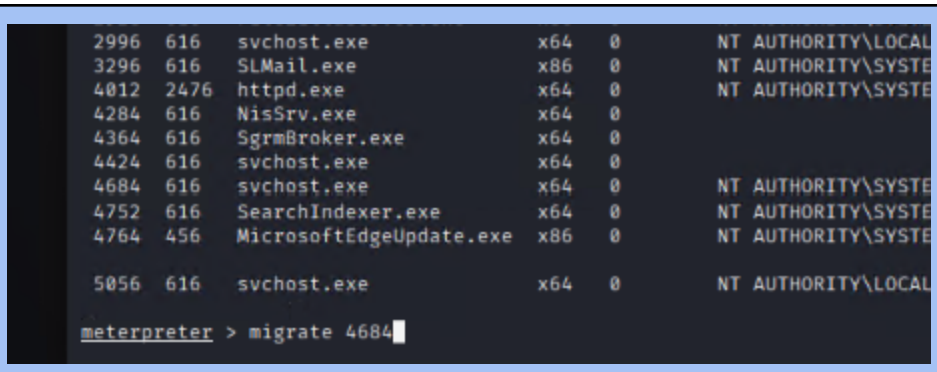
Title	Exposure of Sensitive Data in Code Repositories
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Medium
Description	Searching through Github repositories we found an account for the utser totalrekall. In this repository was a file containing a potential username and hash. We were able to successfully crack the hash. After running an NMAP scan we found two Windows machines. The password and credential allowed us to enter the web server running on the 172.22.117.20 machine.
Images	**Flag 1:**   **Flag 2:** 
Affected Hosts	hxxp://172.22.117.20
Remediation	• Ensure employees regularly update passwords and increase password requirements. • Enforce Multi-Factor Authentication on any employee accessible websites. • Enforce entry to websites through company VPN.

Vulnerability 20	Findings
Title	Unauthorized FTP Transfer of Files
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Medium
Description	Our aggressive scan allowed us to see the Windows Workstation allowed for anonymous login through FTP. We could also see a sensitive data file path on the system. By logging on as anonymous we extracted the file and read the data.
Images	**Flag 3** 
Affected Hosts	hxxp://172.22.117.20
Remediation	• Disable anonymous authentication • Require MFA for FTP login • Remove sensitive files from being accessed through FTP.

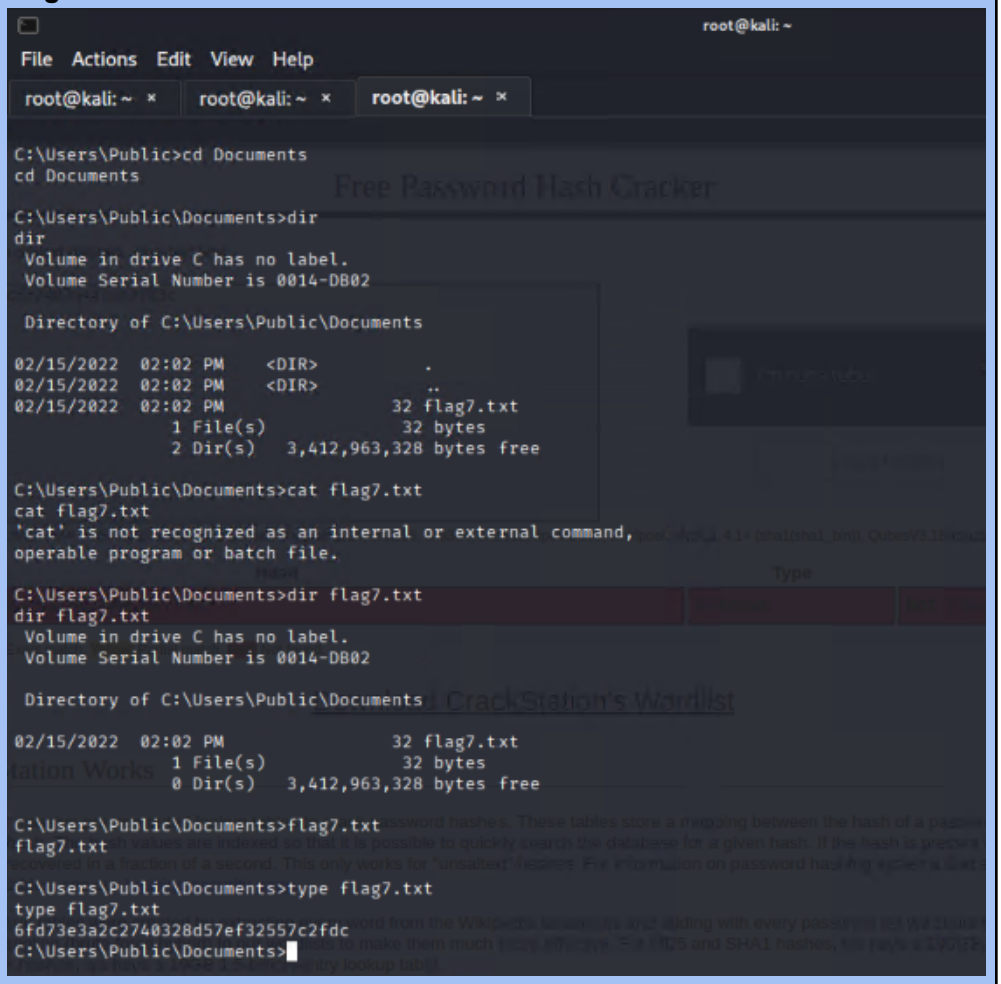
Vulnerability 21	Findings
Title	SLMail Exploit CVE-2003-0264
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	By sending a password with excessive length we can exploit a buffer overflow vulnerability in the POP3 server. This is specific to the Seattle Lab Mail 5.5.

	Through this exploit we can execute code and gain a shell on the system. Metasploit Module: /windows/pop3/seattlelab_pass
Images	**Flag 4** 
Affected Hosts	hxxp://172.22.117.20
Remediation	• Upgrading versions of SLmail eliminates this vulnerability

Vulnerability 22	Findings
Title	Process Migration
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	An attacker can exploit vulnerabilities to inject into a process. This process can then migrate to other processes in an attempt to elevate privileges. Some of the credential dumping we did required us to have SYSTEM privileges. We were able to migrate over to utilize the commands we needed to interact with Kiwi, the metasploit equivalent of MimiKatz Having this type of access allowed us to explore, each system. We did find a sensitive file in a public folder. This should be kept elsewhere.
Images	

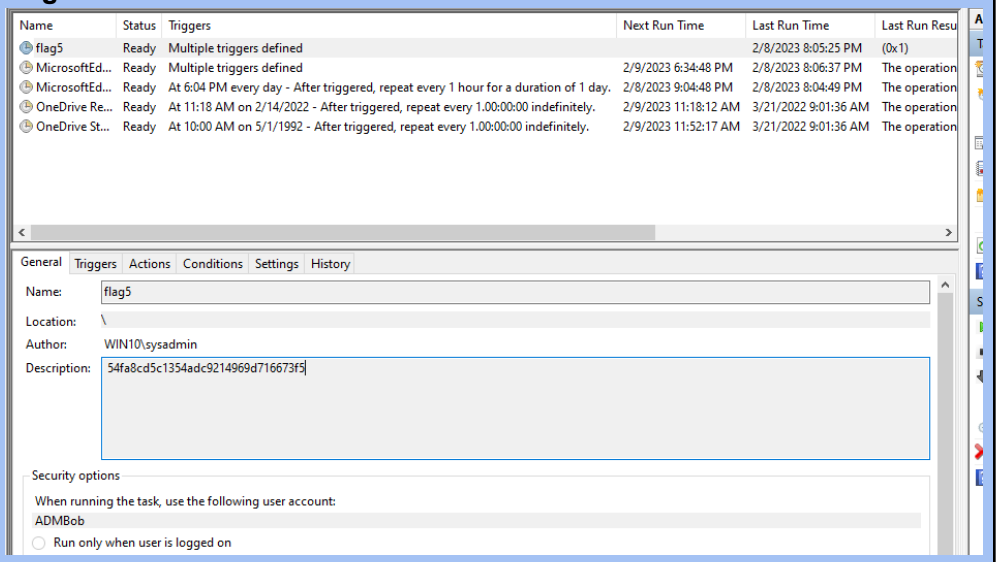


Flag 7:

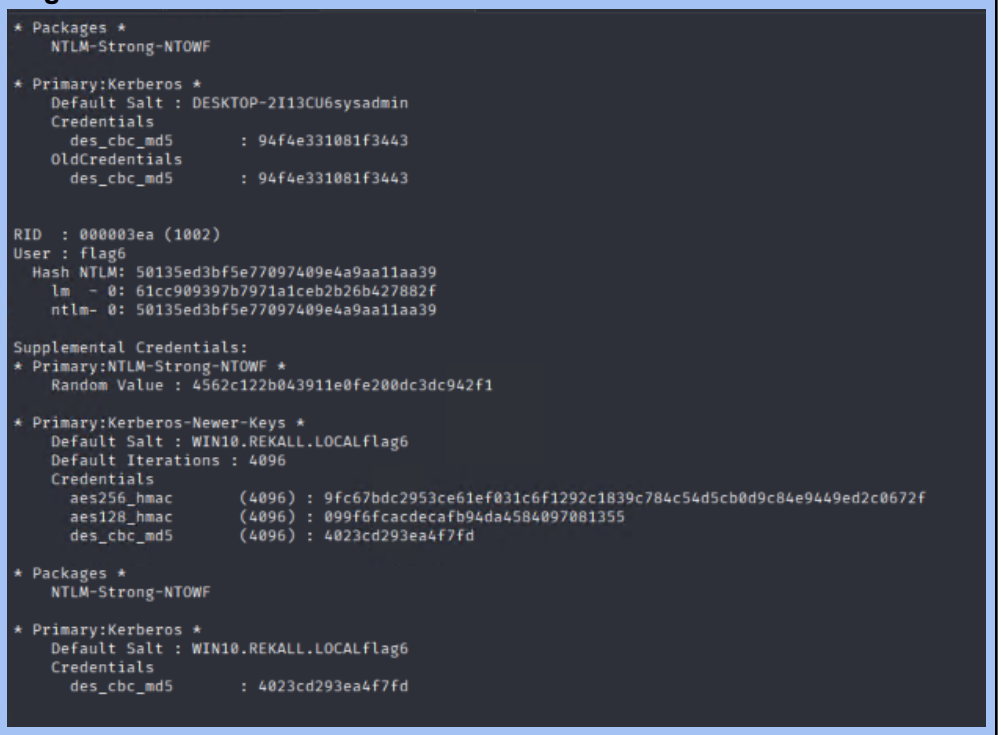
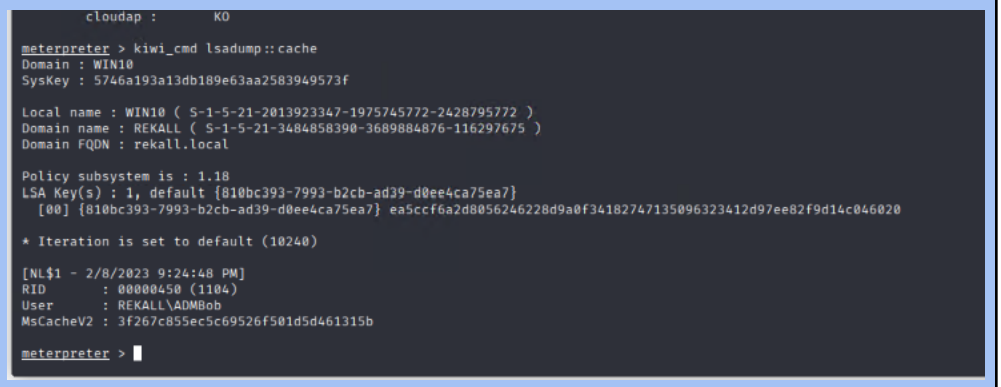
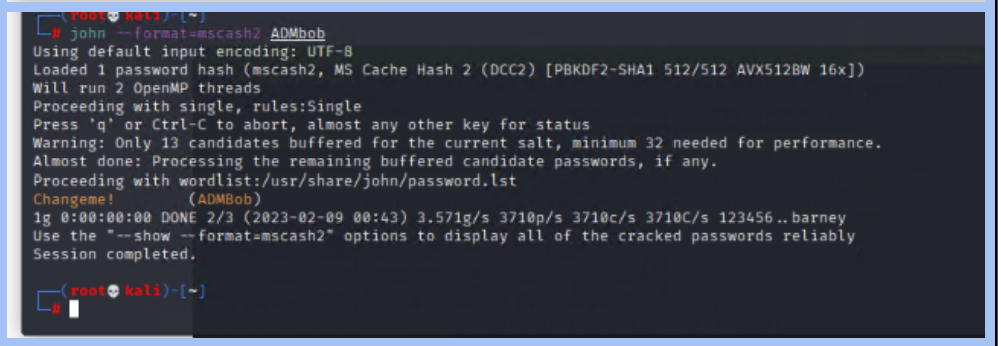


Affected Hosts	hxxp://172.22.117.10,20
Remediation	• Process migration can be very hard to stop and detect. • It's best to focus on preventing access into the system to fully eliminate this possibility. • However, there are some endpoint detection solutions that can look for some types of process injection patterns to block them.

Vulnerability 23	Findings

Title	Uncommon Scheduled Tasks Running on Workstation
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Medium
Description	Directly reporting to J.Smith of Rekall we were told there were concerns with excessive and errant tasks running within the system. When we looked through the task schedule we did find a task that seemed to not belong. With that being said an attacker would do more to obfuscate their task to make it harder to find. Depending on what the task being run is, this can be increasingly dangerous.
Images	**Flag 5:** 
Affected Hosts	hxxp://172.22.117.20
Remediation	• Regular auditing should be put in place through the defense team to scan for abnormal tasks that could potentially be used for privilege escalation. • Limiting the overall privileges of users can help attenuate vectors for privilege escalation. • Monitoring for the events of Command Execution for scheduled jobs can be a place to start defining rules for detection.

Vulnerability 24	Findings
Title	Unmonitored SAM and Cache Dumping
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical

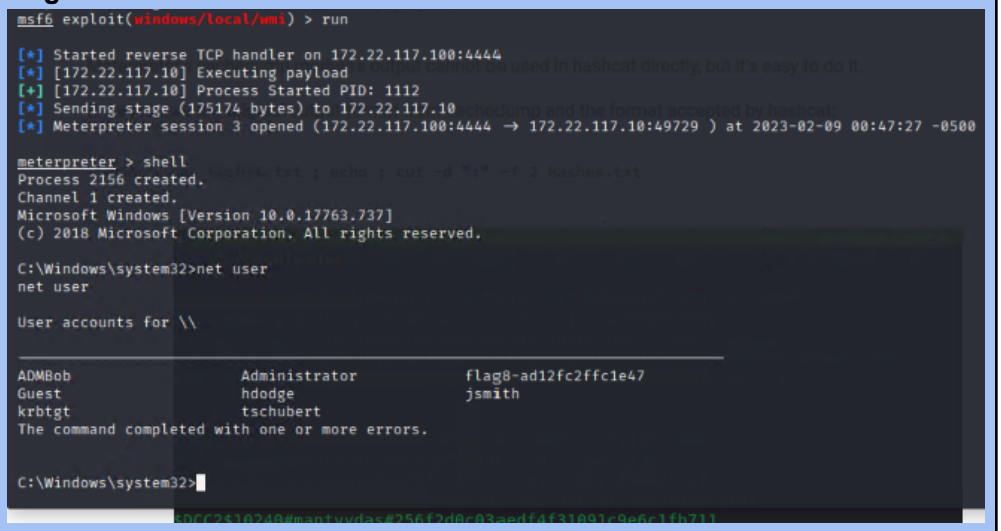
Description	Utilizing Kiwi commands we could interact with the LSASS database and the cached registry. From here we could extract usernames and hashes on the system. These credentials were all eventually cracked and utilized. Specifically the cached credential of ADMbob proved valuable as it overlapped with access to the Domain Controller.
Images	**Flag 6**   
Affected Hosts	hxxp://172.22.117.20

```
* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : DESKTOP-2I13CU6sysadmin
    Credentials
      des_cbc_md5       : 94f4e331081f3443
    OldCredentials
      des_cbc_md5       : 94f4e331081f3443


RID  : 000003ea (1002)
User : flag6
  Hash NTLM: 50135ed3bf5e77097409e4a9aa11aa39
    lm  - 0: 61cc909397b7971a1ceb2b26b427882f
    ntlm- 0: 50135ed3bf5e77097409e4a9aa11aa39

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 4562c122b043911e0fe200dc3dc942f1

* Primary:Kerberos-Newer-Keys *
    Default Salt : WIN10.REKALL.LOCALflag6
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 9fc67bdc2953ce61ef031c6f1292c1839c784c54d5cb0d9c84e9449ed2c0672f
      aes128_hmac       (4096) : 099f6fcacdecafb94da4584097081355
      des_cbc_md5       (4096) : 4023cd293ea4f7fd

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : WIN10.REKALL.LOCALflag6
    Credentials
      des_cbc_md5       : 4023cd293ea4f7fd
```

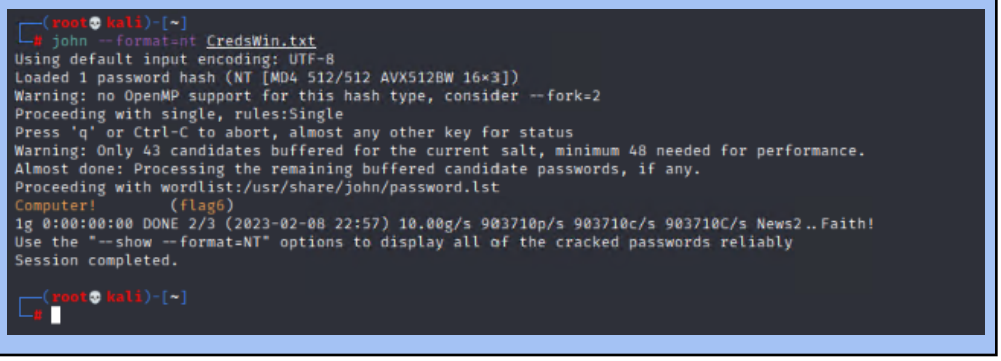
```
        cloudap :       KO

meterpreter > kiwi_cmd lsadump::cache
Domain : WIN10
SysKey : 5746a193a13db189e63aa2583949573f

Local name : WIN10 ( S-1-5-21-2013923347-1975745772-2428795772 )
Domain name : REKALL ( S-1-5-21-3484858390-3689884876-116297675 )
Domain FQDN : rekall.local

Policy subsystem is : 1.18
LSA Key(s) : 1, default {810bc393-7993-b2cb-ad39-d0ee4ca75ea7}
  [00] {810bc393-7993-b2cb-ad39-d0ee4ca75ea7} ea5ccf6a2d8056246228d9a0f34182747135096323412d97ee82f9d14c046020

* Iteration is set to default (10240)

[NL$1 - 2/8/2023 9:24:48 PM]
RID       : 00000450 (1104)
User      : REKALL\ADMBob
MsCacheV2 : 3f267c855ec5c69526f501d5d461315b

meterpreter >
```

```
┌──(root㉿kali)-[~]
└─# john --format=mscash2 ADMbob
Using default input encoding: UTF-8
Loaded 1 password hash (mscash2, MS Cache Hash 2 (DCC2) [PBKDF2-SHA1 512/512 AVX512BW 16x])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 13 candidates buffered for the current salt, minimum 32 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Changeme!        (ADMBob)
1g 0:00:00:00 DONE 2/3 (2023-02-09 00:43) 3.571g/s 3710p/s 3710c/s 3710C/s 123456..barney
Use the "--show --format=mscash2" options to display all of the cracked passwords reliably
Session completed.

┌──(root㉿kali)-[~]
└─#
```

Remediation	• Limit users from being cached in plaintext by implementing the Protected Users security group in Active Directory. • Ensure users have complex passwords • Monitor and flag for command execution that may attempt to gain cached credentials or LSASS access.

Vulnerability 25	**Findings**
Title	Admin Credential Overlap
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	We discovered that our credentials for the user ADMbob overlapped with access to the Domain Controller. This allowed us to perform lateral movement utilizing two metasploit modules. The psexec module allowed us to have system privileges on the workstation while the WMI module allowed us to laterally move over to the DC. The psexec module leverages an administrator credential to create a new service within the system. As a Domain Admin, we had full access to enumerate the machine and gain more usernames and hashes to crack. Metasploit Modules: /Windows/smb/psexec /Windows/Local/Wmi
Images	**Flag 8** 
Affected Hosts	hxxp://172.22.117.10,20
Remediation	• Require Administrators and users with access to multiple machines to have different passwords

	• Increase employee password requirements. • 12 character length • Must contain uppercase, lowercase, numbers and special characters • passwords must change every 4-6 months

Vulnerability 26	Findings
Title	DCsync Attack/Weak Passwords
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	We were able to leverage the domain controller remotely through a process called DCSync. With this we could pull password data from Active Directory. While we simply noted the hashes available, DCsync can be used to create a Golden Ticket for further account manipulation. Of the users found on the Windows system, many of them were able to be cracked utilizing wordlists and the tool John. While the vulnerability that allowed access to the hashes was another matter, the passwords were able to be cracked in reasonable type for an attacker to utilize to their advantage. The user credential pairs are as follows: • flag6:Computer! • sysadmin:Spring2022 • ADMbob:Changeme! • hdodge:Iloveyou! • schubert:Passw0rd! • jsmith:Winter2022 Once inside we were able to access sensitive data files.
Images	

```
┌──(root💀kali)-[~]
└─# john --format=nt CredsWin.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (NT [MD4 512/512 AVX512BW 16×3])
Remaining 1 password hash
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 34 candidates buffered for the current salt, minimum 48 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Spring2022       (sysadmin)
1g 0:00:00:00 DONE 2/3 (2023-02-08 23:00) 11.11g/s 14355p/s 14355c/s 14355C/s 123456..jake
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

Flag 9:

```
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
f7356e02f44c4fe7bf5374ff9bcbf872meterpreter >
```

Flag 10:

```
Success.
meterpreter > dcsync_ntlm Administrator
[+] Account  : Administrator
[+] NTLM Hash : 4f0cfd309a1965906fd2ec39dd23d582
[+] LM Hash  : 0e9b6c3297033f52b59d01ba2328be55
[+] SID      : S-1-5-21-3484858390-3689884876-116297675-500
[+] RID      : 500

meterpreter > dcsync_ntlm hdodge
[+] Account  : hdodge
[+] NTLM Hash : fc9d7c3a3a1e86f1bcc35cd887cb74d5
[+] LM Hash  : 185ef402f3232781fb8c52a203172ec6
[+] SID      : S-1-5-21-3484858390-3689884876-116297675-1108
[+] RID      : 1108

meterpreter > dcsync_ntlm tschubert
[+] Account  : tschubert
[+] NTLM Hash : fc525c9683e8fe067095ba2ddc971889
[+] LM Hash  : ac8fbe72bbeebc2064ae44843d29ee59
[+] SID      : S-1-5-21-3484858390-3689884876-116297675-1106
[+] RID      : 1106

meterpreter > dcsync_ntlm jsmith
[+] Account  : jsmith
[+] NTLM Hash : 7978dc8a66d8e480d9a86041f8409560
[+] LM Hash  : a9fa6022567b16e600341d3e85bdd2d3
[+] SID      : S-1-5-21-3484858390-3689884876-116297675-1105
[+] RID      : 1105

meterpreter > dcsync_ntlm krbtgt
[+] Account  : krbtgt
[+] NTLM Hash : fa5875a009bc010f4a210826e8dabfaa
[+] LM Hash  : d6044fe0087abda3138a7aef49d8d28b
[+] SID      : S-1-5-21-3484858390-3689884876-116297675-502
[+] RID      : 502

meterpreter >
```

Affected Hosts	hxxp://172.22.117.10,20
Remediation	• This is a situation where it is imperative that Administrator accounts • have heavily complex passwords. Overall we must Increase employee password requirements. • 12 character length

	• Must contain uppercase, lowercase, numbers and special characters • passwords must change every 4-6 months • Recall, what led to this access was an overlapping credential. Please, refrain from allowing overlapping credentials to mitigate this type of occurrence. • Detection is possible with the monitoring of activity associated with DCSync.

Resources:

- https://securiti.ai/blog/sensitive-data-exposure/
- https://blog.hubspot.com/website/api-security
- https://portswigger.net/web-security/file-path-traversal
- https://portswigger.net/web-security/cross-site-scripting
- https://brightsec.com/blog/local-file-inclusion-lfi/
- https://portswigger.net/web-security/os-command-injection
- https://www.rapid7.com/db/vulnerabilities/apache-tomcat-cve-2017-12617/
- https://www.synopsys.com/blogs/software-security/cve-2017-5638-apache-struts-vulnerability-explained
- https://medium.com/@briskinfosec/drupal-core-remote-code-execution-vulnerability-cve-2019-6340-35dee6175afa
- https://www.rapid7.com/blog/post/2019/02/21/cve-2019-6340-drupal-core-remote-code-execution-what-you-need-to-know/
- https://nvd.nist.gov/vuln/detail/CVE-2019-14287
- https://attack.mitre.org/techniques/T1003/006/
- https://attack.mitre.org/techniques/T1003/005/