|  |
| --- |
| Cybersecurity |
| Penetration Test Report |

Rekall Corporation

Penetration Test Report

**Student Note: Complete all sections highlighted in yellow.**

# 

## Confidentiality Statement

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

Table of Contents

[Confidentiality Statement 2](#_30j0zll)

[Contact Information 4](#_1fob9te)

[Document History 4](#_3znysh7)

[Introduction 5](#_2et92p0)

[Assessment Objective 5](#_3dy6vkm)

[Penetration Testing Methodology 6](#_2s8eyo1)

[Reconnaissance 6](#_17dp8vu)

[Identification of Vulnerabilities and Services 6](#_3rdcrjn)

[Vulnerability Exploitation 6](#_26in1rg)

[Reporting 6](#_lnxbz9)

[Scope 7](#_35nkun2)

[Executive Summary of Findings 8](#_44sinio)

[Grading Methodology 8](#_z337ya)

[Summary of Strengths 9](#_3j2qqm3)

[Summary of Weaknesses 9](#_1y810tw)

[Executive Summary Narrative](#_4i7ojhp) 10

[Summary Vulnerability Overview 1](#_2xcytpi)3

Vulnerability Findings [1](#_1ci93xb)4

# 

## Contact Information

|  |  |
| --- | --- |
| **Company Name** | Thunder Storm Security, LLC |
| **Contact Name** | Rishabh Gautam |
| **Contact Title** | Penetration Tester |

## 

## Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Comments** |
| 001 | 2023-07-23 | R.Gautam | Draft1 |
| 002 | 2023-07-24 | R.Gautam | Draft2 |
| 003 | 2023-07-25 | R.Gautam | Draft3 |
| 004 | 2023-07-26 | R.Gautam | Initial Review |
| 005 | 2023-07-27 | R.Gautam | Final Review |
|  |  |  |  |
|  |  |  |  |

# 

## Introduction

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

For the testing, we focused on the following:

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

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

### Assessment Objective

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

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

Rekall has outlined the following objectives:

Table 1: Defined Objectives

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

# 

## Penetration Testing Methodology

### Reconnaissance

### 

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

### Identification of Vulnerabilities and Services

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

### Vulnerability Exploitation

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

### Reporting

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

# 

## Scope

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

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

# 

|  |  |
| --- | --- |
| IP Address/URL | Description |
| 192.168.13.0/24 totalrekall.xyz | Rekall internal domain, range and website |

## 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:

Chart

Description automatically generated with medium confidence

### 

### Summary of Strengths

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

* Protection against DDOS attacks: Measures have been put in place to defend against DDOS attacks, ensuring that the network remains available and responsive even under heavy traffic loads intended to overwhelm it
* Network architecture mapping: By comprehensively mapping the network architecture, vulnerabilities in open-source data penetration have been eliminated, reducing the risk of unauthorized access and data breaches.
* Utilization of advanced security tools: Tools such as Metasploit, Hashcat, and Nmap are actively employed to proactively prevent and detect any unauthorized attempts to breach the network, providing an added layer of security.
* Defense and offense strategy: The network security strategy encompasses both defensive and offensive aspects, taking proactive measures to defend against potential threats while also being prepared to respond effectively to any security incidents
* Ongoing penetration testing: Regular and continuous penetration testing is conducted to identify potential weaknesses and vulnerabilities within the network. This allows for timely mitigation of any security issues, strengthening the overall security posture

### 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.

* The web application is susceptible to Cross-Site Scripting (XSS) and SQL injection attacks. This means that attackers can potentially inject malicious code into the application and manipulate the database, leading to unauthorized access and data compromise.
* Storing credentials in the HTML source code is highly insecure. It exposes sensitive information to anyone who views the source code of the webpage, making it easy for attackers to obtain login credentials
* The Apache web server used by the application is outdated and contains known vulnerabilities, making it susceptible to exploitation by malicious actors. Updating the server to the latest version is crucial to mitigate potential risks.
* The SLMail server has security vulnerabilities that can be exploited, granting unauthorized users access to the server shell. This can lead to unauthorized control and manipulation of the server, allowing attackers to execute malicious commands
* Password hashes, which are meant to be securely stored, are accessible to unauthorized individuals. This exposes the passwords to the risk of cracking, potentially leading to privilege escalation and unauthorized account access.
* Rekall Corp's physical address is publicly available, which could be exploited by malicious actors for targeting physical attacks or social engineering attempts.
* Credentials are being revealed during IP lookups, which poses a security risk if unauthorized individuals gain access to this information. It can lead to unauthorized access to sensitive resources.
* The IP addresses within Rekall's IP range are displaying potential vulnerabilities, such as open ports and other weaknesses, when scanned. This makes it easier for attackers to identify potential entry points for unauthorized access

## Executive Summary

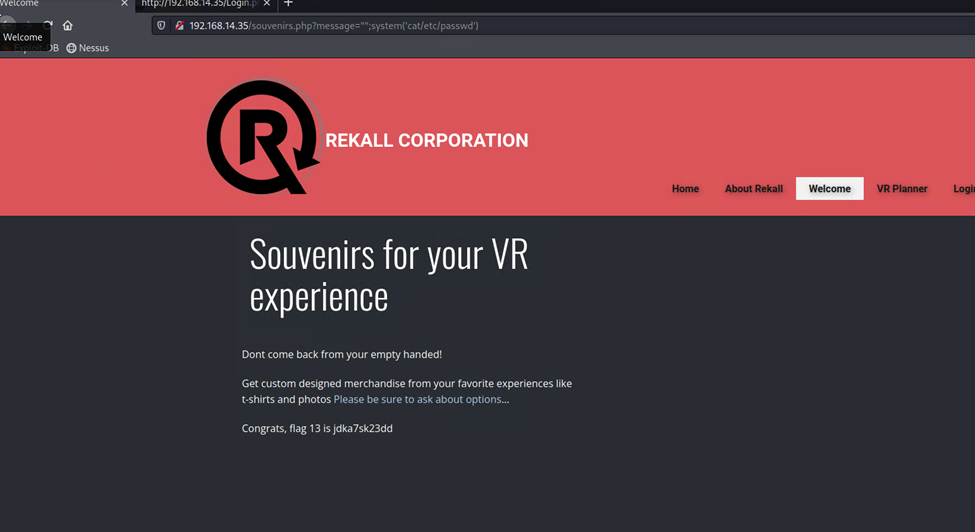
## 

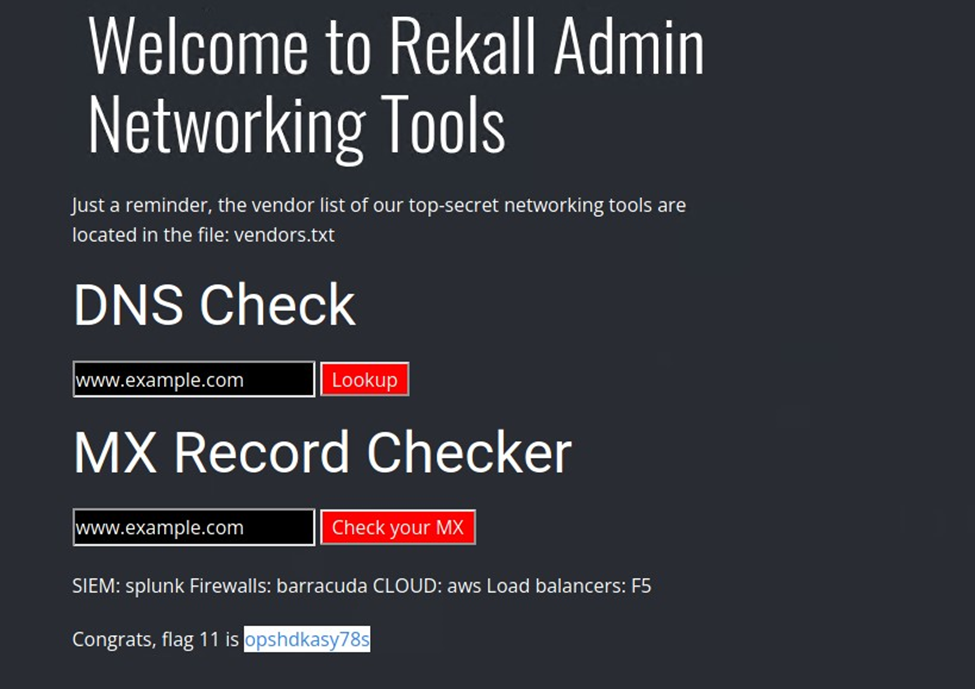
During the examination of Rekall's IT; Thunder Storm Security, LLC identified numerous vulnerabilities, including critical ones, which could have severe consequences on the company's revenue and reputation. They successfully accessed Rekall's assets, extracted essential data, and escalated privileges across several systems. The initial testing by Thunder Storm Security, LLC revealed susceptibility to XSS attacks on the Rekall Web Application's homepage, allowing the execution of malicious scripts.  
  
 Moreover, the web app's VR Planner website allowed file uploads, making it susceptible to local file inclusion. The Comments page had an XSS Stored vulnerability enabling the execution of scripting code, while the Login.php toolbar was open to SQL Injection attacks, and the Networking.php page was vulnerable to Command Injection attacks. Thunder Storm Security, LLC used OSINT to discover visible open-source data and a stored certificate in crt.sh. Alarming discoveries included user login credentials exposed in the HTML source code of the Login.php page and an open robots.txt file. Unauthorized access to the web host's files and folders resulted from user credentials found in a Github repository, and an outdated Apache server vulnerability was found.   
  
  
Exploring the Windows OS environment, Thunder Storm Security found insecurely open FTP ports (21 and 110) used for SLMail service, leading to password hash file access and the creation of a reverse shell using Metasploit. Vulnerabilities in the Windows 10 Machine Task Scheduler and Metepreter's ability to display directories on open Windows folders were also identified.   
  
  
Thunderstorm Security, LLC discovered 5 publicly accessible IP addresses with vulnerabilities within the Linux system, and one of the servers was running Drupal. They gained access to one host and escalated privileges to root using stolen credentials. Additionally, using Meterpreter, a well-known shell RCE execution vulnerability was found. The Metasploit Shellshock attack allowed access to the sudoers file. To protect the company's assets and operations, Thunder Storm Security, LLC provided comprehensive recommendations to address these vulnerabilities effectively.  
  
  
Thunder Storm Security has provided details steps of the penetration testing was performed on day to day basis with screenshots.The penetration test commenced with a series of attacks on the company's online application, such as cross-site scripting, SQL injection, and local file inclusion. The objective was to gather information about the business by identifying specific confidential webpages and credentials.

## 

## 

## 





Subsequently, our focus shifted to attempting unauthorized access to their Linux machines and locating user credentials stored within the systems. We used to dig totalrekall.xyz to find the open source information.   
  
Next we ran a nmap scam scan on the company’s network to see how many hosts were up (5).

A screenshot of a computer

Description automatically generated

Nessus scan was ran on the IP 192.168.13.12 to find a critical vulnerability

A computer screen shot of a computer

Description automatically generated

Metasploit was used to exploit the vulnerability to gain shell session   
  
  
  
Subsequently, leveraging the previously discovered struts vulnerability, we exploited another loophole using a metasploit exploit, granting us an additional shell session on the victim's IP address 192.168.13.12. Consequently, we regained access to the files once more.  
  
A screenshot of a computer

Description automatically generated  
During a prior nmap scan, it was identified that the Drupal service was operational at IP address 192.168.13.13. Consequently, we leveraged an exploit to initiate a shell session, successfully obtaining unauthorized access to the victim's system.

A computer screen shot of a computer code

Description automatically generated

During the reconnaissance phase, we identified a user named Alice on the host 192.168.13.14. We gained SSH access to her account because of a weak password. Using this access, we exploited an unpatched version of sudo, granting us root privileges and full control over the system. Further reconnaissance led us to a GitHub page containing a user's name and hashed password, which we successfully cracked to gain additional access.

A screenshot of a computer screen

Description automatically generated

We have obtained the users' credentials, we gained entry to the company network using the online VPN. Being aware of the presence of anonymous FTP on the webpage, we successfully established a connection through FTP.

A screenshot of a computer

Description automatically generated

After discovering the availability of SLmail as a service, we proceeded to utilize an exploit that allowed us to gain unauthorized access through this service.

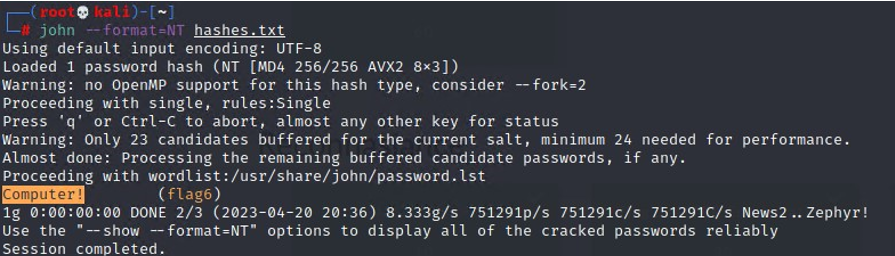
A screenshot of a computer

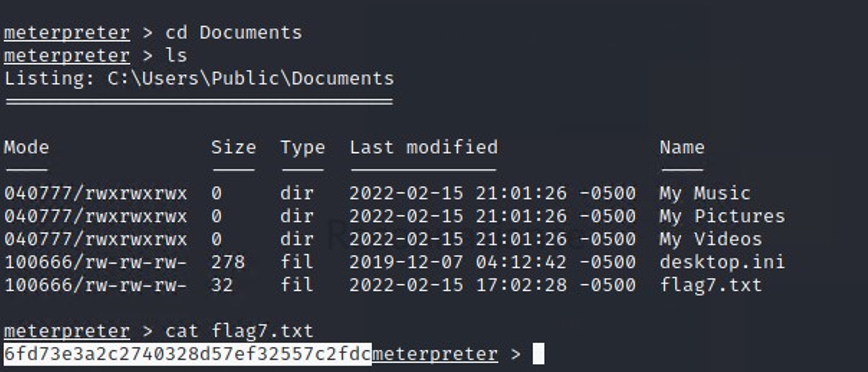
Description automatically generated

After gaining access, we scheduled a task to establish persistence within the network. By exploring the scheduled tasks, we discovered additional valuable information

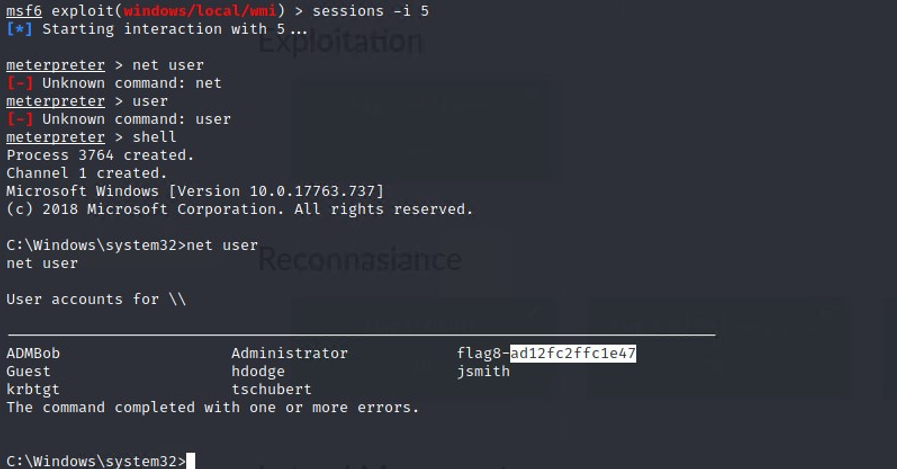
A screenshot of a computer

Description automatically generated

During our ongoing session, we utilized "Kiwi" to extract SAM on a Windows system, enabling us to obtain a list of users along with their hashed passwords. Subsequently, we successfully cracked these passwords to gain unauthorized access  
  
  
  
  
During this active session, we successfully maintained access to the computer's filesystem, allowing us to continue exploring and searching through its contents.



We successfully captured all the users present on the system.



With the utilization of additional user credentials, we successfully elevated privileges, granting us the ability to conduct deeper searches within the system.

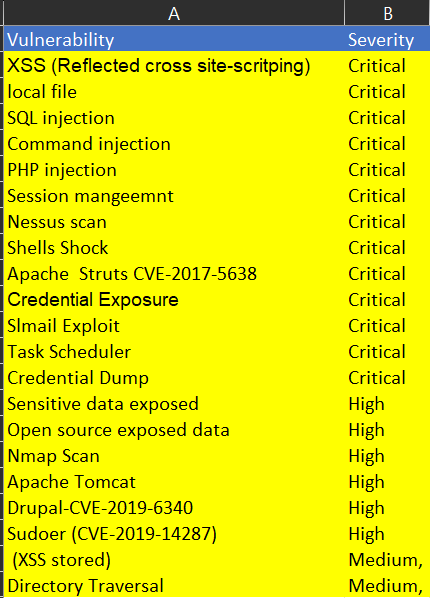
A computer screen shot of a computer

Description automatically generated

After gaining complete access to the Windows 10 machine, we employed the dcsync technique to extract the password of the user named "Administrator," whom we had identified earlier. This allowed us to acquire administrator privileges on the domain controller.

## A close up of numbers Description automatically generated

## Summary Vulnerability Overview



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

|  |  |
| --- | --- |
| **Scan Type** | **Total** |
| Hosts | 192.168.13.10, 192.168.13.11  192.168.13.12, 192.168.13.13  192.168.13.14, 192.168.14.35  172.22.117.20, 172.22.117.110 |
| Ports |  |

|  |  |
| --- | --- |
| **Exploitation Risk** | **Total** |
| **Critical** | 13 |
| **High** | 6 |
| **Medium** | 2 |
| **Low** | 0 |

## Vulnerability Findings

|  |  |
| --- | --- |
| **Vulnerability 1** | **Findings** |
| **Title** | Cross-site-scritping(XSS) |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | Critical |
| **Description** | Malicious script was successfully  executed on home page <script>alert(you'vebeenhacked)</script> <SCRscriptIPT> Secret Agent </SCRscriptIPT |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Input validation |

|  |  |
| --- | --- |
| **Vulnerability 2** | **Findings** |
| **Title** | Xss stored |
| **Type (Web app / Linux OS / WIndows OS)** | Web app |
| **Risk Rating** | medium |
| **Description** | On Comment page we entered **<script>alert(you'vebeenhacked)</script>** |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Introduce XSS protection measures to prevent the injection of script code |

|  |  |
| --- | --- |
| **Vulnerability 3** | **Findings** |
| **Title** | Local File inclusion |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | The LFI attack was carried out successfully, enabling the upload of a .php file using the toolbar found on the VR Planner page Images section.  .php file contained :**<?php $command = $\_GET['cmd']; echo system($command); ?>** |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Implement measures to disallow direct file path appending, and if feasible, restrict the API to permit inclusion solely from a specific directory and its subdirectories. This approach enhances security and helps prevent unauthorized access to files |

|  |  |
| --- | --- |
| **Vulnerability 4** | **Findings** |
| **Title** | SQL injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | During the attempt to access the /Login.php page, a payload (consisting of a name or "1=1") was entered into the password toolbar, which led to a successful exploitation. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | The web application should be configured to reject direct user input and/or implement character escaping mechanisms |

|  |  |
| --- | --- |
| **Vulnerability 5** | **Findings** |
| **Title** | Command injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | Access from /Networking.php to 192.168.14.35/disclaimer.php?page=vendors.txt was permitted by navigating through 192.168.14.35/networking.php. The ability to input "splunk" into the toolbar designed for DNS Check Images was also observed |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Apply input validation to prevent unauthorized access. |

|  |  |
| --- | --- |
| **Vulnerability 6** | **Findings** |
| **Title** | Sensitive data exposure |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | High |
| **Description** | The Robots.txt page can be accessed without any restrictions. One can examine the HTTP response headers using tools like Burp Suite or by making a curl request. Moreover, user information can be obtained by inspecting the web page or checking the HTML code of the login.php page. The Robots.txt file is accessible to anyone visiting the website |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | To address the security concerns, restrict Robots.txt access, adjust HTTP response headers, implement secure authentication, enforce HTTPS, conduct regular security testing, follow best practices, apply access controls, and provide security training to administrators. |

|  |  |
| --- | --- |
| **Vulnerability 7** | **Findings** |
| **Title** | PHP injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | Modifying the link to  **http://192.168.14.35/souvenirs.php?message=""; system('cat/etc/passwd')** made the website vulnerable to php injection |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Apply input validation and parameterized queries to user-supplied data. Ensure that the "message" parameter is sanitized and only accepts valid input. Implement proper error handling to avoid revealing sensitive system information. Additionally, restrict access and permissions for critical files like /etc/passwd. Regularly update PHP and server software to mitigate known security issues. Conduct security audits and penetration testing to identify and address potential vulnerabilities in the application. Educate developers on secure coding practices to prevent future injection vulnerabilities.A screenshot of a computer  Description automatically generated |

|  |  |
| --- | --- |
| **Vulnerability 8** | **Findings** |
| **Title** | Session Management |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | The Burp Intruder tool was utilized to identify the confidential session ID, which happened to be 87. This was achieved after obtaining the link to the admin\_legal\_data.php page and recognizing flag 12 during the process. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Strengthen session management by using secure, random, and long session IDs. Implement session expiration and regeneration after login. Ensure sensitive data is encrypted. Perform regular security testing to identify and fix vulnerabilities.  A screenshot of a computer  Description automatically generated |

|  |  |
| --- | --- |
| **Vulnerability 9** | **Findings** |
| **Title** | Directory Traversal |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Medium |
| **Description** | By exploiting the flag 10 or command injection vulnerability and executing 'ls' command, we can view all folders. Among them, accessing the "old\_disclaimers" directory is possible. By modifying the URL to http://192.168.14.35/disclaimer.php?page=old\_disclaimers/disclaimer\_1.txt, we can access previous versions of the disclaimer page |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Apply input validation and sanitization for user-supplied data on the "page" parameter. Implement proper command filtering and avoid direct execution of user input. Use whitelisting to restrict access to directories and files. Enforce strong access controls to prevent unauthorized access. Regularly update the application and underlying software to patch known security issues. Conduct security assessments and penetration testing to identify and fix any remaining vulnerabilities. Educate developers on secure coding practices to prevent future flaws. Additionally, consider implementing a Web Application Firewall (WAF) to detect and block malicious requests.  A screenshot of a computer  Description automatically generated |

|  |  |
| --- | --- |
| **Vulnerability 10** | **Findings** |
| **Title** | Open-source exposed data |
| **Type (Web app / Linux OS / WIndows OS)** | Linux os |
| **Risk Rating** | High |
| **Description** | On the Domain Dossier webpage, viewed the WHOIS data for totalrekall.xyz . it then showed the sensitive information. Upon searching for "totalrekall.xyz" on crt.sh, a stored certificate was discovered |
| **Images** | A screenshot of a computer  Description automatically generated  A screenshot of a computer  Description automatically generated |
| **Affected Hosts** | 34.102.136.180 (https://centralops.net/co/DomainDossier.aspx) |
| **Remediation** | Make certain that no confidential information is publicly accessible, perform necessary actions to sanitize WHOIS records and cert.sh site needs to be protected |

|  |  |
| --- | --- |
| **Vulnerability 11** | **Findings** |
| **Title** | Nmap Scan |
| **Type (Web app / Linux OS / WIndows OS)** | Linux os |
| **Risk Rating** | High |
| **Description** | Ran an aggressive Nmap scan: nmap -A 192.168.13.0/24 |
| **Images** |  |
| **Affected Hosts** | 192.168.13.10, 192.168.13.12, 192.168.13.13, 192.168.13.14, 192.168.13.1 |
| **Remediation** | To counter the aggressive Nmap scan, employ measures like blocking probes, limiting information disclosure, introducing delays, and providing deceptive data. These tactics aim to hinder the effectiveness of the scan, preventing adversaries from gathering accurate information about the network's hosts and services |

|  |  |
| --- | --- |
| **Vulnerability 12** | **Findings** |
| **Title** | Nessus Scan |
| **Type (Web app / Linux OS / WIndows OS)** | Linux os |
| **Risk Rating** | Critical |
| **Description** | Nessus scan was run to view the critical vulnerability for Apache Struts |
| **Images** |  |
| **Affected Hosts** | 192.168.13.12 |
| **Remediation** | Apache updates needs to be performed on a regular basis |

|  |  |
| --- | --- |
| **Vulnerability 13** | **Findings** |
| **Title** | Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617) |
| **Type (Web app / Linux OS / WIndows OS)** | Linux os |
| **Risk Rating** | High |
| **Description** | By utilizing the exploit "multi/http/tomcat\_jsp\_upload\_bypass" and configuring the RHOST to 192.168.13.10 in Metasploit, one can attain shell access. Employing the Meterpreter interface allows for executing commands, leading to obtaining flag 7. |
| **Images** |  |
| **Affected Hosts** | 192.168.14.10 |
| **Remediation** | Apply security patches and updates to Tomcat. Implement strong authentication and access controls to limit unauthorized access. Regularly monitor and analyze system logs for suspicious activities. Use intrusion detection/prevention systems to detect and block malicious attempts |

|  |  |
| --- | --- |
| **Vulnerability 13** | **Findings** |
| **Title** | Shellshock |
| **Type (Web app / Linux OS / WIndows OS)** | Linux os |
| **Risk Rating** | Critical |
| **Description** | The Shellshock exploit is executed via Metasploit using "exploit/multi/http/apache\_mod\_cgi\_bash\_env\_exec." The vulnerable web page, "/cgi-bin/shock-me.cgi," is targeted. Once shell access is obtained, flags are extracted by observing the "sudoers" file using "cat /etc/sudoers" and the "/etc/passwd" file. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.11 |
| **Remediation** | Update Apache to the latest version, fix CGI scripts, and implement input validation. Restrict access to sensitive files like "/etc/sudoers" and "/etc/passwd". Regularly scan for vulnerabilities and apply security patches promptly. |

|  |  |
| --- | --- |
| **Vulnerability 14** | **Findings** |
| **Title** | Apache Struts-CVE-2017-5638 |
| **Type (Web app / Linux OS / WIndows OS)** | Linux os |
| **Risk Rating** | Critical |
| **Description** | Using Metasploit's multi/http/struts2\_content\_type\_ognl exploit, gain a meterpreter shell on RHOST to 192.168.13.12. Download the target file to the Kali system, unzip it, and found the flag. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.12 |
| **Remediation** | Update the affected Struts version to a patched release. Apply security patches regularly, implement proper access controls, and use web application firewalls to filter malicious requests. Conduct regular security assessments to identify and address potential vulnerabilities |

|  |  |
| --- | --- |
| **Vulnerability 15** | **Findings** |
| **Title** | Drupal-CVE-2019-6340 |
| **Type (Web app / Linux OS / WIndows OS)** | Linux os |
| **Risk Rating** | High |
| **Description** | Drupal exploit (unix/webapp/drupal\_restws\_unserialize) through Metasploit , they managed to gain a meterpreter shell access but setting RHOSTS to 192.168.13.13. Once inside, executing the "getuid" command led them to discover "flag 11."  . |
| **Images** |  |
| **Affected Hosts** | 192.168.13.13 |
| **Remediation** | Apply the latest Drupal updates to patch the issue. Additionally, implement proper input validation and serialization controls in the Drupal RESTWS module to prevent unserialize attacks. Regularly monitor the system for any suspicious activity or unauthorized access |

|  |  |
| --- | --- |
| **Vulnerability 16** | **Findings** |
| **Title** | Sudoer (CVE-2019-14287) |
| **Type (Web app / Linux OS / WIndows OS)** | Linux os |
| **Risk Rating** | High |
| **Description** | Alice's account is a recognized one, and by leveraging information obtained from flag 1's WHOIS data, we were able to guess the password and gain SSH access to the server. Once inside, we utilized the sudo command with the option "-u#-1" to read the contents of the "flag12.txt" file located in the root directory. This action was made possible due to the presence of a vulnerability known as CVE-2019-14287. |
| **Images** |  |
| **Affected Hosts** | 192.168.13.14 |
| **Remediation** | Implement strong password policies for all user accounts, including Alice's. Patch the vulnerability CVE-2019-14287. Enforce least privilege principles to limit users' sudo access. Regularly review and monitor system logs for suspicious activities. Conduct security training for employees to raise awareness about social engineering risks. |

|  |  |
| --- | --- |
| **Vulnerability 17** | **Findings** |
| **Title** | Credential Exposure |
| **Type (Web app / Linux OS / WIndows OS)** | Window os |
| **Risk Rating** | Critical |
| **Description** | While searching Github led to totalrekall Github page and searching the site repository. User credentials are exposed in the HTML code of the Login.php page and become visible when the page is highlighted in a web browser. |
| **Images** | A screen shot of a computer  Description automatically generated |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Implement proper input validation and use server-side scripting to handle sensitive data securely. Store user credentials in a hashed and salted format, and avoid displaying them in HTML. Implement HTTPS and follow best practices for web application security to safeguard against unauthorized access. |

|  |  |
| --- | --- |
| **Vulnerability 18** | **Findings** |
| **Title** | SLmail Exploit |
| **Type (Web app / Linux OS / WIndows OS)** | Window os |
| **Risk Rating** | Critical |
| **Description** | The SLMail vulnerability, caused by an open port 110, was effectively exploited using the Metasploit's windows/pop3/seattlelab\_pass exploit. This exploitation led to a successful Meterpreter session being established, granting unauthorized access to the system |
| **Images** |  |
| **Affected Hosts** | 192.168.14.35 |
| **Remediation** | Close port 110, apply security patches, and implement robust access controls. Regularly update and monitor the system to prevent unauthorized exploitation and maintain a secure environment. |

|  |  |
| --- | --- |
| **Vulnerability 19** | **Findings** |
| **Title** | Task Sheduler |
| **Type (Web app / Linux OS / WIndows OS)** | Schedule Tasks Exploit |
| **Risk Rating** | Critical |
| **Description** | By utilizing the SLMail shell access obtained from flag 4, flag 5 can be acquired by querying the scheduled tasks. This suggests that accessing SLMail provides the necessary permissions to gather information about scheduled tasks, leading to the discovery of flag 5 |
| **Images** |  |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Implement the following remediation measures: 1) Secure SLMail to prevent unauthorized shell access. 2) Review and restrict access to scheduled tasks, limiting it to authorized users only. 3) Regularly update and patch software to minimize potential exploits. 4) Conduct periodic security audits to identify and address any new vulnerabilities. |
|  |  |

|  |  |
| --- | --- |
| **Vulnerability 20** | **Findings** |
| **Title** | Credential Dump |
| **Type (Web app / Linux OS / WIndows OS)** | Schedule Tasks Exploit |
| **Risk Rating** | Critical |
| **Description** | By utilizing the SLMail shell access obtained from flag 4, flag 5 can be acquired by querying the scheduled tasks. This suggests that accessing SLMail provides the necessary permissions to gather information about scheduled tasks, leading to the discovery of flag 5 |
| **Images** | A screenshot of a computer  Description automatically generated |
| **Affected Hosts** | 172.22.117.20 |
| **Remediation** | Implement the following remediation measures: 1) Secure SLMail to prevent unauthorized shell access. 2) Review and restrict access to scheduled tasks, limiting it to authorized users only. 3) Regularly update and patch software to minimize potential exploits. 4) Conduct periodic security audits to identify and address any new vulnerabilities. |