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

Rekall Corporation

Penetration Test Report

# AngelSecurity, LLC

## 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** | AngelSecurity, LLC |
| --- | --- |
| **Contact Name** | Angel Perez |
| **Contact Title** | Penetration Tester |
| **Contact Phone** | 555.224.2411 |
| **Contact Email** | blue\_cyberangel@proton.me |

## 

## Document History

| **Version** | **Date** | **Author(s)** | **Comments** |
| --- | --- | --- | --- |
| 001 | 03/07/2024 | Angel Perez |  |

# 

## Introduction

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

For the testing, we focused on the following:

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

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

### Assessment Objective

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

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

Rekall has outlined the following objectives:

Table 1: Defined Objectives

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

# 

## Penetration Testing Methodology

### Reconnaissance

### 

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

### Identification of Vulnerabilities and Services

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

### Vulnerability Exploitation

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

### Reporting

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

# 

## Scope

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

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

# 

# 

## Executive Summary of Findings

## 

### Grading Methodology

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

**Critical**: Immediate threat to key business processes.

**High**: Indirect threat to key business processes/threat to secondary business processes.

**Medium**: Indirect or partial threat to business processes.

**Low**: No direct threat exists; vulnerability may be leveraged with other vulnerabilities.

Informational: No threat; however, it is data that may be used in a future attack.

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

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.

* Many input fields on the Rekall website are using input validation
* Numerous files and directories were not accessible due to effective user access controls on Rekall’s Linux servers.

### 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.

* Most critical vulnerabilities are located on or related to the TotalRekall website including sensitive data exposure, command injection, XSS, Local File inclusion and SQLi and more sensitive data found on GitHub site
* Weak password users
* Linux and Windows machines have several services running with old versions with vulnerabilities including SLMail pop3d, Drupal, and Apache Tomcat Remote Code Execution.
* OSINT tool “WHOIS” found data that could be used for attackers
* Kiwi found several important usernames and password’s hashes with weak passwords

## Executive Summary

AngelSecurity (AS) executed three (3) intensive days of investigation and attempts to break into the Rekall Corporation’s network and determine its vulnerabilities

### Day 1 (Web Application)

We attempted to exploit vulnerabilities on the organization’s web application at http://192.168.14.35 and started by Cross-Site Scripting (XSS) Reflected: Injecting Malicious Scripts on the Welcome page[Flag\_1.png](https://drive.google.com/file/d/1Sk48M_Z48_BRmc9WdFzdBopSHPjPQ_dV/view?usp=sharing).

Then we looked further to see if any more XSS Reflected scripts would work across different vulnerable pages and we found on the Memory-Planner page there is an input validation but we could break it by encapsulating our script [Flag\_2.png](https://drive.google.com/file/d/1YkhYkwp_CXirZATcxVEI5Zn-8u9kDBBI/view?usp=sharing).

Also we discovered a very danger XSS Stored vulnerability within the Comments page since an attacker can store malicious code on the host server [Flag\_3.png](https://drive.google.com/file/d/1cAnKxq6vVIQce12_xTa74_fCnUvPOu08/view?usp=sharing).

Next we came across with Local File Inclusion (LFI) back-end component vulnerability found on the Memory-Planner page and we uploaded a php script that can let us to send command remotely to the web app [Flag\_5.png](https://drive.google.com/file/d/1YhmAkFyvN_prHA-nwMaZnzZ0x3vbm8eI/view?usp=sharing). Then further down the page was better protected against this vulnerability since it only allowed jpg files but we were able to bypass this by ending the file .php.jpg. [Flag\_6.png](https://drive.google.com/file/d/1rmLuBB5JsJ-Iidhwzsv7sk529_JK3l5t/view?usp=sharing).

After that we attempted to perform SQL injection on Login page and we were able to break the logic of the database and logins since there are not input validation [Flag\_7.png](https://drive.google.com/file/d/1pt_qW1zzik96-xzxJq1G7fNEEoBosV0-/view?usp=sharing). Also we found sensitive data exposure located within the page highlighting the webpage we see credentials, allowing us to login [Flag\_8.png](https://drive.google.com/file/d/1YERzGbEFxU5mhkY6nthus0QE7VtfhKe1/view?usp=sharing).

Then we looked for exclusion standard settings and successfully add “robots.txt” to the end of the web to reveal sensitive data [Flag\_9.png](https://drive.google.com/file/d/1vv-JAEK7yoNBi_7rkWXz9PnyYzjNydyM/view?usp=sharing).

Next, looking the page that Flag\_8 redirects us Networking page, and after many attempts to command injection on DNS Check we were able to find sensitive data on vendors.txt [Flag\_10.png](https://drive.google.com/file/d/19QgqJqSezkHb9tDy4Uh-djDhh9zUl9Iu/view?usp=sharing) and just below there is another field called MX Record Checker that has better protections but was still compromised by using | instead of & because that character wasn’t in the list of input validation [Flag\_11.png](https://drive.google.com/file/d/1IAgEjxaV1ManvwpnDPGkXfcEhZL1zDQI/view?usp=sharing).

### Day 2 (Linux Servers)

In this day we did some reconnaissance against the website TotalRekall.xyz using a Dossier open source tool found within https://osintframework.com/ to find information about the WHOIS domain and we found useful information [Flag\_1.png](https://drive.google.com/file/d/1YgbfYqkh7_58lcO9xbt0Asc1hZJ61Hgs/view?usp=sharing), more information was found looking in txt records [Flag\_2.png](https://drive.google.com/file/d/1Q0AbwUa-t2nindGpk4nUfcdtRYH1VTLM/view?usp=sharing) and SSL certificate using crt.sh website [Flag\_3.png](https://drive.google.com/file/d/1QAGq_Z_-CRJkX-Ch70c7QZru8IBmv2xI/view?usp=sharing).

Then we attempted to run nmap against the entire network of the target 192.168.13.0/24 (the subnet /24 to run a scan across 254 possible host machines). We found that there were 5 IPs with open ports excluding our IP (192.168.13.1) which are 192.168.13.10, 192.168.13.11, 192.168.13.12, 192.168.13.13, 192.168.13.14 [Flag\_4.png](https://drive.google.com/file/d/1CWT7rG5EfDRIp18BgdvFL1Sv9z_Kynxf/view?usp=sharing). Then we ran another scan but this time with option -A to run an aggressive scan and we found a host machine running drupal located at 192.168.13.13

[Flag\_5.png](https://drive.google.com/file/d/1v3jcbq7zsXAkMF4IrGTt-f-vNOnjldtF/view?usp=sharing) with a vulnerability CVE-2019-6340 [Drupal < 8.5.11 / < 8.6.10 - RESTful Web Services unserialize() Remote Command Execution (Metasploit)](https://www.exploit-db.com/exploits/46510) .

We ran a Nessus scan against the host 192.168.13.12 and found a critical vulnerability ID 97610 [Flag\_6.png](https://drive.google.com/file/d/1CD1JsSNBdNP6-h4i_gSsgJ9UV1OVIQKk/view?usp=sharing).

Taking in consideration that 192.168.13.10 is running on port 8080 Apache tomcat/Coyote JSP engine 1.1 we searched on metasploit and attempt to exploit with multi/http/tomcat\_jsp\_upload\_bypass and successfully got access then we use find command to search for all flag’s file name [Flag\_7.png](https://drive.google.com/file/d/1O4_INVB1G8Na-0MDWTtKa9xi_l4CEuk4/view?usp=sharing) [Tomcat - Remote Code Execution via JSP Upload Bypass (Metasploit)](https://www.exploit-db.com/exploits/43008).

Next we exploited 192.168.13.11searching for shock in metasploit and using the next module exploit/multi/http/apache\_mod\_cgi\_bash\_env\_exec and target URI: /cgi-bin/shockme.cgi we were able to access to the system [Flag\_8.png](https://drive.google.com/file/d/1Y9VZ0DT7ae-DBEvB6FVi7PYJrqXmcsnb/view?usp=sharing) and then looking in sudoers file and passwd file we can see sensitive data [Flag\_9.png](https://drive.google.com/file/d/182jU3NWFKsX1z5_xWVDBqI2o0oVpoLgp/view?usp=sharing) [Apache mod\_cgi - 'Shellshock' Remote Command Injection](https://www.exploit-db.com/exploits/34900) .

Looking carefully again [Flag\_1.png](https://drive.google.com/file/d/1YgbfYqkh7_58lcO9xbt0Asc1hZJ61Hgs/view?usp=sharing) we noticed that Admin Name: sshUser Alice and attempting to get access to 192.168.13.14 with that username and password guessing we finally got success with alice as password. ssh alice@192.168.13.14 password: alice. Looking for vulnerability CVE-2019-14287 we exploited this server by sudo -u#-1 to escalate privileges as root and see sensitive data on /root/ file [Flag\_12.png](https://drive.google.com/file/d/18OOWBFPlXeo8qBu90hCIt7hs2Ck8IMuo/view?usp=sharing) [sudo 1.8.27 - Security Bypass - Linux local Exploit](https://www.exploit-db.com/exploits/47502).

### Day 3 (Windows Servers)

AS began this day by investigating into totalrekall github to gather all information possible that could help us. We discovered a huge vulnerability due credentials were stored here easily accessible to the public and then we were able to crack the hash using john [Flag\_1.png](https://drive.google.com/file/d/1VFx8BrXP4SdFbbjeiLk3cBHKwG_XcJzd/view?usp=sharing).

Running nmap -A on the entire network 172.22.117.0/24 (the subnet /24 to run a scan across 254 possible host machines) we found 3 IP one from the machine is scanning 172.22.117.100, Windows 10 machine 172.22.117.20 and Windowserver2019 172.22.117.10 [nmap.png](https://drive.google.com/file/d/1deREIqwckU9JH_7Ep5ICsep1kmH__MJ_/view?usp=sharing). Noticing that Win10 has Apache port 80 open we attempt to connect to that site and we were prompt for username and password and introducing the credentials for trivera password Tanya4life found on the prior investigation, we gran access and be able to see sensitive data [Flag\_2.png](https://drive.google.com/file/d/1PKnNNCQVHjT2ev7Vqwg5_NQqQs3UY7Qn/view?usp=sharing).

Returning to the port scan [nmap.png](https://drive.google.com/file/d/1deREIqwckU9JH_7Ep5ICsep1kmH__MJ_/view?usp=sharing)we also noticed that FTP port 21 is running on 172.22.117.20 and also FTP anonymous access is possible and password: guest [ftp.png](https://drive.google.com/file/d/1KcESMH3anfsan3BSZEfFlmjdUdlwfgAx/view?usp=sharing). Once logged into FTP as anonymous we downloaded sensitive data and able to read it in our machine [Flag\_3.png](https://drive.google.com/file/d/1t5jC30RteFLnXUt7UiEmSu_QWCP3-c3b/view?usp=sharing)

Also Port 110 is running pop3 SLMAIL. We search SLmail on metasploit and display exploit /windows/pop3/seattlelab\_pass, setting the target 172.22.117.20 and host machine 172.22.117.100 we were able to access and open a Meterpreter been able to see sensitive data [Flag\_4.png](https://drive.google.com/file/d/1zSsEKC_cStFuyVrZEizusL7ycCpeYkTr/view?usp=sharing) then using the search command inside the Meterpreter will reveal more data [Flag\_7.png](https://drive.google.com/file/d/1T2y7Dr3xNO_xJKkJGV8BFqxn1Eraru64/view?usp=sharing) [Seattle Lab Mail (SLmail) 5.5 - POP3 'PASS' Remote Buffer Overflow (1)](https://www.exploit-db.com/exploits/638) .

Within the same Meterpreter we looked at scheduled tasks on the system for more information and be able to find an interesting tasks and read it with schtasks /query /TN flag5 /FO list /v command [Flag\_5.png](https://drive.google.com/file/d/1v6UApl9LZySqMcPJarw1itAYNHjhF_kb/view?usp=sharing)

After compromising SLMail the Meterpreter shell will be System user then we loaded kiwi and attempted to dump the contents of SAM (Security Accounts Manager) by using lsa\_dump\_sam and crack the hash with john in our host machine [Flag\_6.png](https://drive.google.com/file/d/1dK0wTX4sOI3X44tIB9nETWTwNY6Sjbuf/view?usp=sharing)

Next we use kiwi again to dump the cached credentials on Win10 to reveal administrator, AMDBob his credentials cached then using john to crack it [8.png](https://drive.google.com/file/d/1RPTAfmKuBw1CCOri0i61sZq5HeSbiTqE/view?usp=sharing) and use this information to lateral move to Windows server 2019 172.22.117.10 with ADMBob credentials [Flag\_8.png](https://drive.google.com/file/d/1XdPiMmjEFcjkT6k7UDcsalIxUSwrMiK9/view?usp=sharing) we successfully got granted to WinDC and user net user to list all the user for this domain

Then login into WinDC2019 we were able to break into the system [Flag\_9.png](https://drive.google.com/file/d/1iCQ4s5gEYJYTqyqhTWk5hLusVqLA2tMn/view?usp=sharing)

## 

## Summary Vulnerability Overview

| **Vulnerability** | **Severity** |
| --- | --- |
| Web Site |  |
| Reflected XSS on different web pages | **High** |
| Stored XSS on Comment page | **Critical** |
| Local File Inclusion (LFI) on Memory-Planner page | **Critical** |
| SQL injection on login page | **Critical** |
| Sensitive data (credentials) exposure on login page | **Critical** |
| Exclusion standard settings exposure (robots.txt) | **Low** |
| Command injection on Networking page | **Critical** |
| Linux |  |
| Open Source Data exposure about ssh alice | **High** |
| DNS txt records | **Informational** |
| SSL certificate | **Informational** |
| Drupal CVE-2019-6340 | **Critical** |
| Apache Struts 2.3.5-2.3.31/2.5x<2.5.10.1 Jakarta Multipart parser RCE (remote) (CVE-2014-6271) | **Critical** |
| Apache Tomcat Remote Code Execution Vulnerability (CVE-2017-12617) | **Critical** |
| Shellshock Vulnerability (CVE-2014-6271) | **Critical** |
| Weak password for Alice user | **High** |
| Sudo Security Bypass (CVE-2019-14287) | **Critical** |
| Windows |  |
| Sensitive data (credentials) exposure on totalrekall public GitHub site | **Critical** |
| Weak Password for trivera user | **High** |
| FTP Anonymous Login and weak password | **Critical** |
| SLMail pop3d Exploit (CVE-2003-0264) | **Critical** |
| Scheduled Task on the Windows system | **Informational** |
| Extraction and dumping of credentials with kiwi | **Medium** |
| Weak password for ADMBob user | **Critical** |

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

| **Scan Type** | **Total** |
| --- | --- |
| Hosts | 192.168.14.35 → 15.197.148.33 – totalrekall.xyz  192.168.13.10 - Linux  192.168.13.11 - Linux  192.168.13.12 - Linux  192.168.13.13 - Linux  192.168.13.14 - Linux  192.168.13.1 – Linux  172.22.117.20 – Windows10  172.22.117.10 – WinDC10  172.22.117.100 – Linux host |
| Ports | 21(FTP), 25(SMTP), 80 (HTTP) 110 (POP3), 135  (RPC), 443 (HTTPS), 8080(HTTP-Apache Tomcat/JSP) |

| **Exploitation Risk** | **Total** |
| --- | --- |
| **Critical** | 13 |
| **High** | 4 |
| **Medium** | 2 |
| **Low** | 1 |
| **Informational** | 3 |

## Vulnerability Findings

| **Vulnerability 1** | **Findings** |
| --- | --- |
| **Title** | Reflected Cross Site Scripting (XSS) - OWASP 3 Injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | **High** |
| **Description** | We were able to insert alerts into input fields in Welcome and Memory-Planner pages. This vulnerability is classified as High due to allowing an attacker to perform scripts that are reflected off the web server and executed in the context of the victim's browser. |
| **Images** | [Flag\_1.png](https://drive.google.com/file/d/1Sk48M_Z48_BRmc9WdFzdBopSHPjPQ_dV/view?usp=sharing) [Flag\_2.png](https://drive.google.com/file/d/1YkhYkwp_CXirZATcxVEI5Zn-8u9kDBBI/view?usp=sharing) |
| **Affected Hosts** | 15.197.148.33 – totalrekall.xyz |
| **Remediation** | Input validation on both the client-side and the server-side ensures that the input is what the app is expecting. |

| **Vulnerability 2** | **Findings** |
| --- | --- |
| **Title** | Stored Cross Site Scripting (Persistent XSS) - OWASP 3 Injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | **Critical** |
| **Description** | We were able to insert alerts into the Comment page which are stored in the database.The most damaging type of XSS is Stored XSS because scripts permanently are into the application's database. |
| **Images** | [Flag\_3.png](https://drive.google.com/file/d/1cAnKxq6vVIQce12_xTa74_fCnUvPOu08/view?usp=sharing) |
| **Affected Hosts** | 15.197.148.33 – totalrekall.xyz |
| **Remediation** | Input validation on both the client-side and the server-side ensures that the input is what the app is expecting. |

| **Vulnerability 3** | **Findings** |
| --- | --- |
| **Title** | Local File Inclusion (LFI) - OWASP 1 Broken Access Control, 2 Cryptographic Failures, 5 Security Misconfiguration |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | **Critical** |
| **Description** | We were able to successfully create a .php script file and upload it into the Memory Planner page. Resulting in malicious scripts being run against the database to read, modify, and/or delete data. |
| **Images** | [Flag\_5.png](https://drive.google.com/file/d/1YhmAkFyvN_prHA-nwMaZnzZ0x3vbm8eI/view?usp=sharing)[Flag\_6.png](https://drive.google.com/file/d/1rmLuBB5JsJ-Iidhwzsv7sk529_JK3l5t/view?usp=sharing) |
| **Affected Hosts** | 15.197.148.33 – totalrekall.xyz |
| **Remediation** | Restrict what type of files are able to be storage into the database (any php) |

| **Vulnerability 4** | **Findings** |
| --- | --- |
| **Title** | SQL injection - OWASP 3 Injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | **Critical** |
| **Description** | Bypassing the logic query used by the login process to gain unauthorized access |
| **Images** | [Flag\_7.png](https://drive.google.com/file/d/1pt_qW1zzik96-xzxJq1G7fNEEoBosV0-/view?usp=sharing) |
| **Affected Hosts** | 15.197.148.33 – totalrekall.xyz |
| **Remediation** | Input validation on both the client-side and the server-side ensures that the input is what the app is expecting. |

| **Vulnerability 5** | **Findings** |
| --- | --- |
| **Title** | Command injection - OWASP 3 Injection |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | **Critical** |
| **Description** | We were able on the Networking page to inject the command cat Vendors.txt to reveal sensitive data and also the passwd file of linux which contains all the usernames. |
| **Images** | [Flag\_10.png](https://drive.google.com/file/d/19QgqJqSezkHb9tDy4Uh-djDhh9zUl9Iu/view?usp=sharing)[Flag\_11.png](https://drive.google.com/file/d/1IAgEjxaV1ManvwpnDPGkXfcEhZL1zDQI/view?usp=sharing) |
| **Affected Hosts** | 15.197.148.33 – totalrekall.xyz |
| **Remediation** | Input validation on both the client-side and the server-side ensures that the input is what the app is expecting. |

| **Vulnerability 6** | **Findings** |
| --- | --- |
| **Title** | Open Source Data Exposure |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS  Web App |
| **Risk Rating** | **High** |
| **Description** | Through OSINT tools we gather information on the target machine and web page that was useful during our investigation. We found the IP of the website totalrekall.xyz and also the protocol and username to potentially get access (ssh Alice). |
| **Images** | [Flag\_1.png](https://drive.google.com/file/d/1YgbfYqkh7_58lcO9xbt0Asc1hZJ61Hgs/view?usp=sharing)[Flag\_2.png](https://drive.google.com/file/d/1Q0AbwUa-t2nindGpk4nUfcdtRYH1VTLM/view?usp=sharing)[Flag\_3.png](https://drive.google.com/file/d/1QAGq_Z_-CRJkX-Ch70c7QZru8IBmv2xI/view?usp=sharing) |
| **Affected Hosts** | 15.197.148.33 – totalrekall.xyz |
| **Remediation** | DNS txt records and SSL don’t have any info that can be useful for an attacker but it is very important to modify the domain and erase SSH alice this info should not be able to view publicly. |

| **Vulnerability 7** | **Findings** |
| --- | --- |
| **Title** | Drupal, Apache Struts 2.3.5-2.3.31/2.5x, Apache Tomcat Remote Code Execution Vulnerability and Shellshock Vulnerability |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Critical** |
| **Description** | We noticed at the moment we ran an aggressive nmap scan -A that Drupal has vulnerability CVE-2019-6340 potentially a threat. Apache Tomcat Remote we were able to exploited with a module on metasploit as well using Shellshock vulnerability we were able to get access to this linux machine |
| **Images** | [Flag\_5.png](https://drive.google.com/file/d/1v3jcbq7zsXAkMF4IrGTt-f-vNOnjldtF/view?usp=sharing)[Flag\_6.png](https://drive.google.com/file/d/1CD1JsSNBdNP6-h4i_gSsgJ9UV1OVIQKk/view?usp=sharing)[Flag\_7.png](https://drive.google.com/file/d/1O4_INVB1G8Na-0MDWTtKa9xi_l4CEuk4/view?usp=sharing)[Flag\_8.png](https://drive.google.com/file/d/1Y9VZ0DT7ae-DBEvB6FVi7PYJrqXmcsnb/view?usp=sharing)[Flag\_9.png](https://drive.google.com/file/d/182jU3NWFKsX1z5_xWVDBqI2o0oVpoLgp/view?usp=sharing) |
| **Affected Hosts** | 192.168.13.10 - Linux  192.168.13.11 - Linux  192.168.13.12 - Linux  192.168.13.13 - Linux |
| **Remediation** | It’s priority to \*always upgrade to the latest version of the service that you are running  \*sometime the latest version comes with bug so is recommendable to have backups for the prior version but never keep very old versions |

| **Vulnerability 8** | **Findings** |
| --- | --- |
| **Title** | Sudo 1.8.27 - Security Bypass |
| **Type (Web app / Linux OS / WIndows OS)** | Linux OS |
| **Risk Rating** | **Critical** |
| **Description** | With information found about ssh alice and guessing password alice (weak password) we found a vulnerability on this linux that allows us to escalate privileges to root being able to read, modify or delete everything in that machine. |
| **Images** | [Flag\_12.png](https://drive.google.com/file/d/18OOWBFPlXeo8qBu90hCIt7hs2Ck8IMuo/view?usp=sharing) |
| **Affected Hosts** | 192.168.13.14 - Linux |
| **Remediation** | Change password for Alice and have good training for employees to have a healthy security culture  Check for updates and upgrade the OS once a week (automatic task) |

| **Vulnerability 9** | **Findings** |
| --- | --- |
| **Title** | Sensitive Data exposed public GitHub site and Weak Password |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | **Critical** |
| **Description** | We discovered that totalrekall has GitHub site free public access that has exposed hash for a user. We ran John against that hash and in seconds was able to crack it. After that we use that credentials to break into port 80 172.22.117.20 Win10 and be able to see more sensitive data. |
| **Images** | [Flag\_1.png](https://drive.google.com/file/d/1VFx8BrXP4SdFbbjeiLk3cBHKwG_XcJzd/view?usp=sharing)[Flag\_2.png](https://drive.google.com/file/d/1PKnNNCQVHjT2ev7Vqwg5_NQqQs3UY7Qn/view?usp=sharing) |
| **Affected Hosts** | 172.22.117.20 – Windows10 |
| **Remediation** | It is a priority to delete that file on the GitHub repository and make sure whoever handles this data can’t post sensitive data on the internet.  Require a strong password complexity that requires passwords to be over 12 characters long, upper+lower case, & include a special character and change every 6 months. |

| **Vulnerability 10** | **Findings** |
| --- | --- |
| **Title** | FTP Anonymous Login |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **Critical** |
| **Description** | Through the nmap scan we see Port 21 FTP is running and allow Anonymous we successfully login by password guessing “guest” and download sensitive data |
| **Images** | [ftp.png](https://drive.google.com/file/d/1KcESMH3anfsan3BSZEfFlmjdUdlwfgAx/view?usp=sharing)[Flag\_3.png](https://drive.google.com/file/d/1t5jC30RteFLnXUt7UiEmSu_QWCP3-c3b/view?usp=sharing) |
| **Affected Hosts** | 172.22.117.20 – Windows10 |
| **Remediation** | Close port 21 if is not in use or disable Anonymous authentication |

| **Vulnerability 11** | **Findings** |
| --- | --- |
| **Title** | SLMail pop3d Exploit |
| **Type (Web app / Linux OS / WIndows OS)** | Windows OS |
| **Risk Rating** | **Critical** |
| **Description** | Using metasploit we searched for vulnerability on SLMail and ran an exploit module that open a Meterpreter session with System access and able to see sensitive data |
| **Images** | [Flag\_4.png](https://drive.google.com/file/d/1zSsEKC_cStFuyVrZEizusL7ycCpeYkTr/view?usp=sharing)[Flag\_7.png](https://drive.google.com/file/d/1T2y7Dr3xNO_xJKkJGV8BFqxn1Eraru64/view?usp=sharing) |
| **Affected Hosts** | 172.22.117.20 – Windows10 |
| **Remediation** | Update to IMAP using port 143 instead of SLMail Pop3 and close port 110 this also has the benefit of preventing data from being lost, as it backs up emails on a server. |

| **Vulnerability 12** | **Findings** |
| --- | --- |
| **Title** | Extraction and dumping of credentials with kiwi |
| **Type (Web app / Linux OS / WIndows OS)** | Windows 10 |
| **Risk Rating** | **Medium** |
| **Description** | After gaining a Meterpreter shell we dumped many different user and password hashes with kiwi and were able to crack them with John. The password is very weak |
| **Images** | [Flag\_6.png](https://drive.google.com/file/d/1dK0wTX4sOI3X44tIB9nETWTwNY6Sjbuf/view?usp=sharing) |
| **Affected Hosts** | 172.22.117.20 – Windows10 |
| **Remediation** | The vulnerability is found on weak password more than dumping credentials on SAM  Priority = change password for this users and send them to cybersecurity training  Require a strong password complexity that requires passwords to be over 12 characters long, upper+lower case, & include a special character and change every 6 months. |

| **Vulnerability 13** | **Findings** |
| --- | --- |
| **Title** | Lazy Admin Weak Password |
| **Type (Web app / Linux OS / WIndows OS)** | Windows 10 Server |
| **Risk Rating** | **Critical** |
| **Description** | After gaining a Meterpreter shell with System access and obtaining credentials from dump cached credentials we found ADMBob with a weak password then we proceeded to lateral move to Windows server 2019 and got access to Windows Domain Controller. |
| **Images** | [8.png](https://drive.google.com/file/d/1RPTAfmKuBw1CCOri0i61sZq5HeSbiTqE/view?usp=sharing)[Flag\_6.png](https://drive.google.com/file/d/1dK0wTX4sOI3X44tIB9nETWTwNY6Sjbuf/view?usp=sharing) |
| **Affected Hosts** | 172.22.117.10 – WinDC10 |
| **Remediation** | Priority = change password for this users and send them to cybersecurity training  Require a strong password complexity that requires passwords to be over 12 characters long, upper+lower case, & include a special character and change every 6 months. |