| 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** | Ninja Security Consultants |
| --- | --- |
| **Contact Name** | Christopher Nnaji |
| **Contact Title** | Lead Pen tester |

## 

## Document History

| **Version** | **Date** | **Author(s)** | **Comments** |
| --- | --- | --- | --- |
| 001 | Sept 8th 2022 | C. Nnaji |  |

# 

## 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.

* High-level summary of strengths here

### Summary of Weaknesses

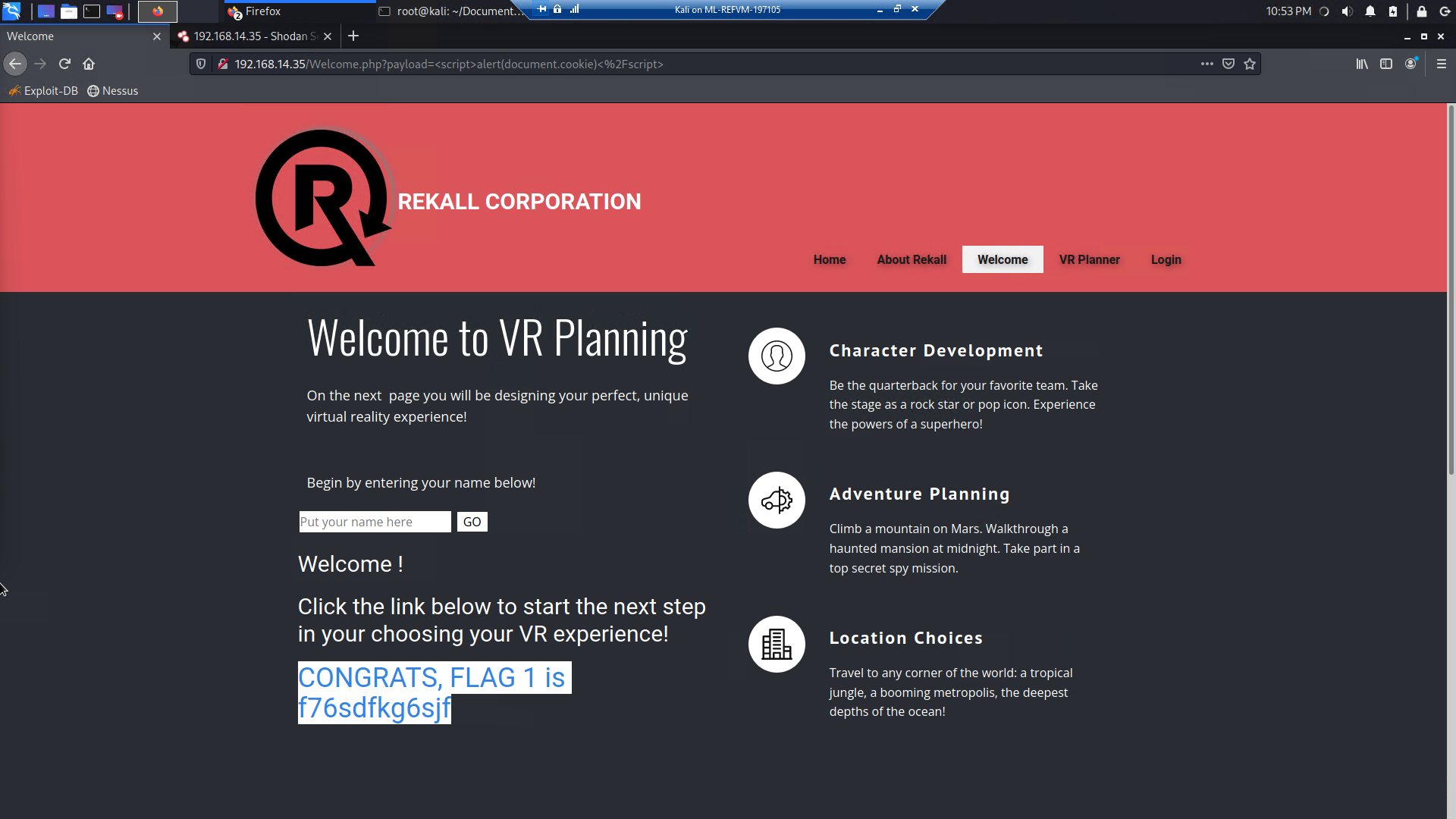
We successfully found several critical vulnerabilities that should be immediately addressed in order to prevent an adversary from compromising the network. These findings are not specific to a software version but are more general and systemic vulnerabilities.

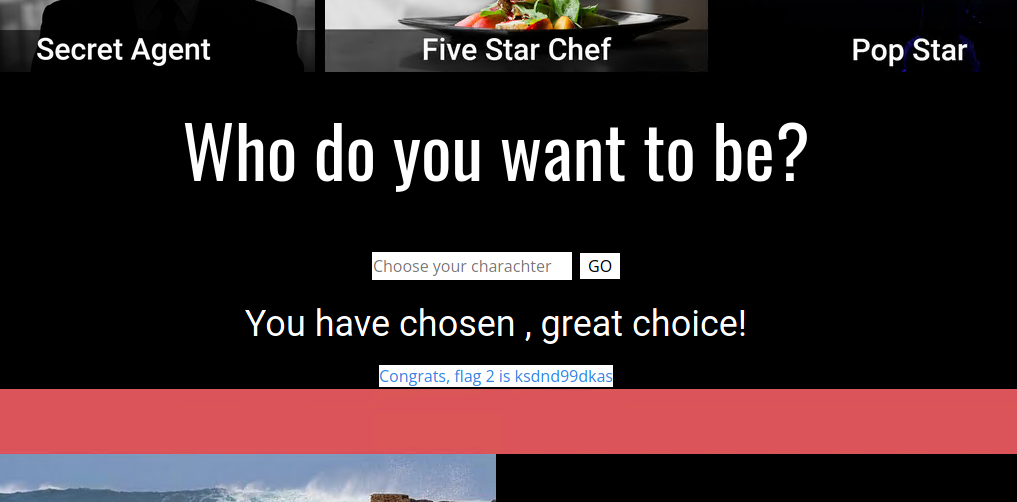
* SQL injections
* Login credentials posted directly to the site
* image uploads can be used to inject script
* open ports
* apache version out of date with know exploits
* linux has backdoor code execution
* ability to get root access using exploits
* ssh port able to using stolen credentials

## Executive Summary

Several vulnerabilities using XSS to deliver payloads and find information on the back end of the server. I’ve enclosed a few fields with vulnerabilities. Some lower priority and some critical.

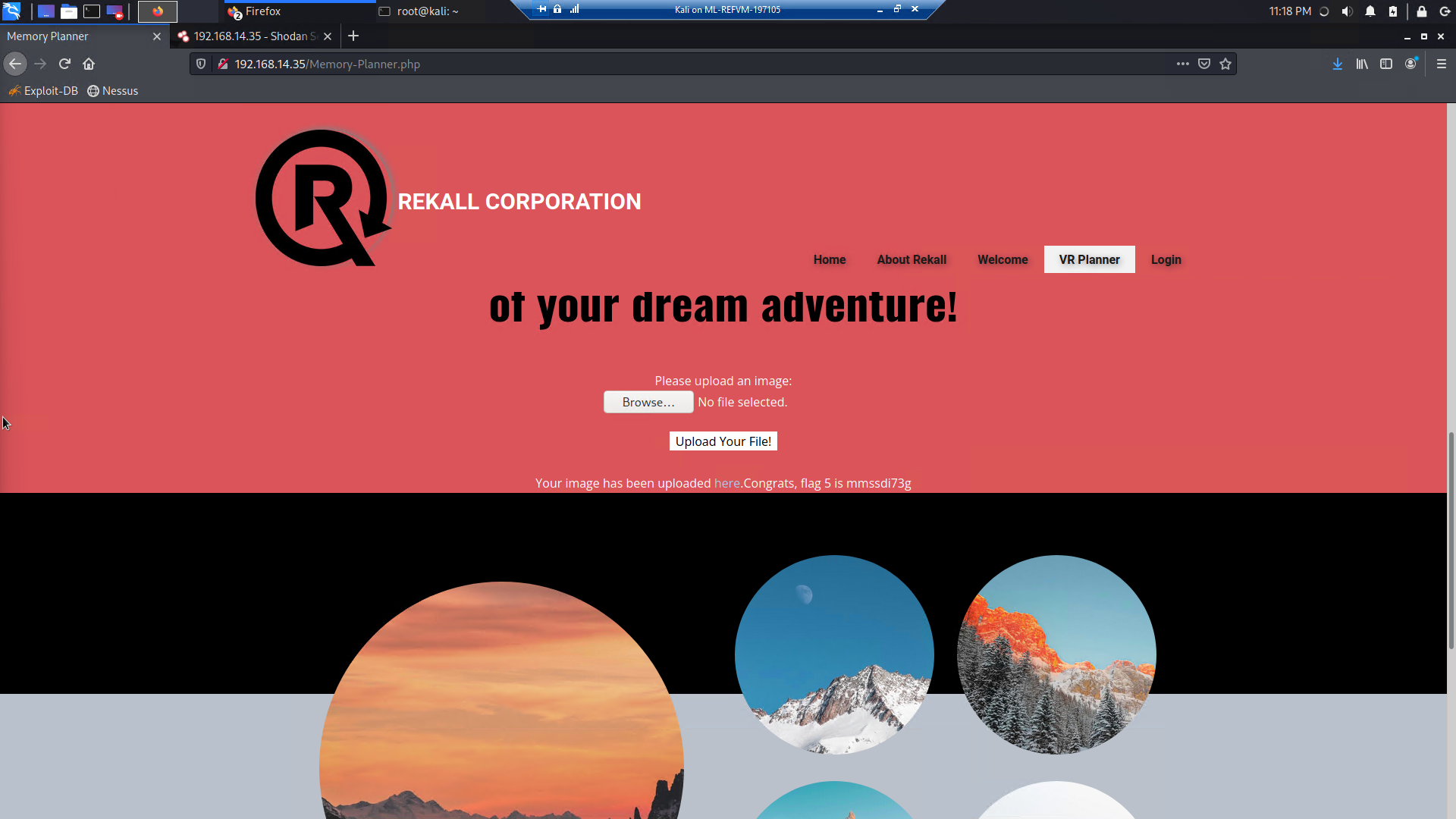
(insert flags 1-3)

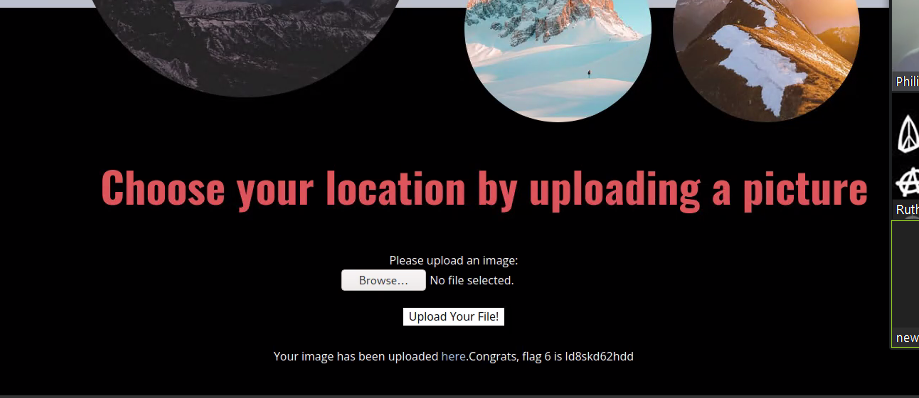






I used Local file Inclusion to upload a script into a field designated for photo uploads.



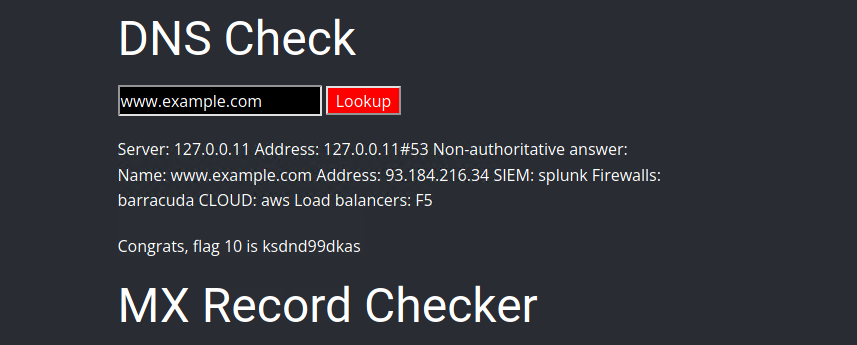


Some vulnerabilities are simply displayed on the page itself



Sensitive files easily accessible via command injections

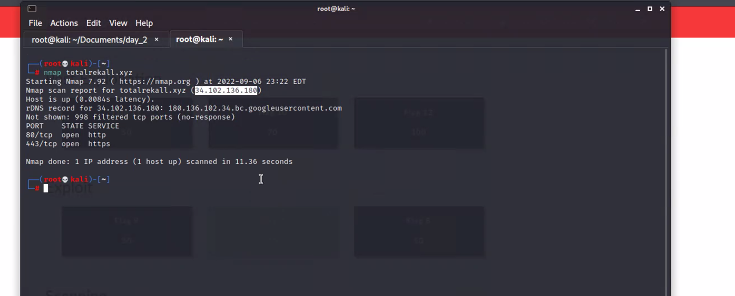
(flag 10)

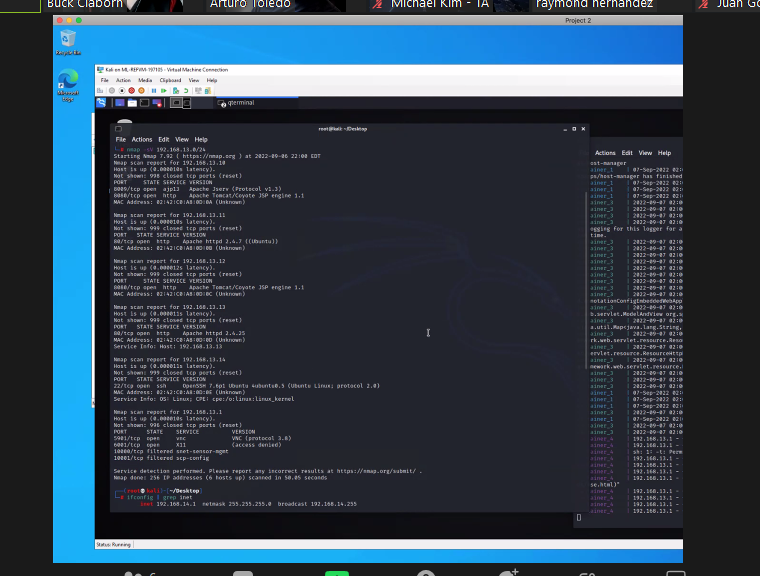


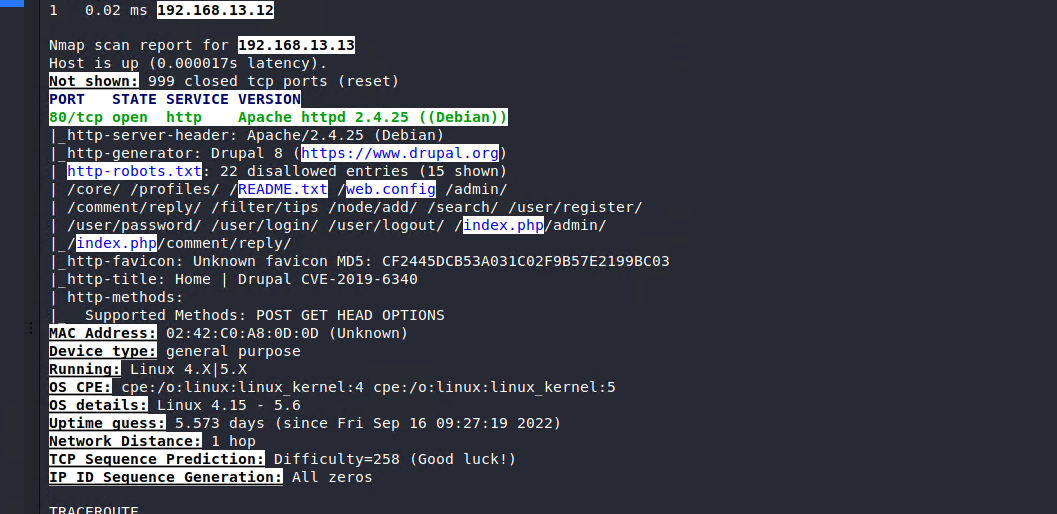
Linux ctf

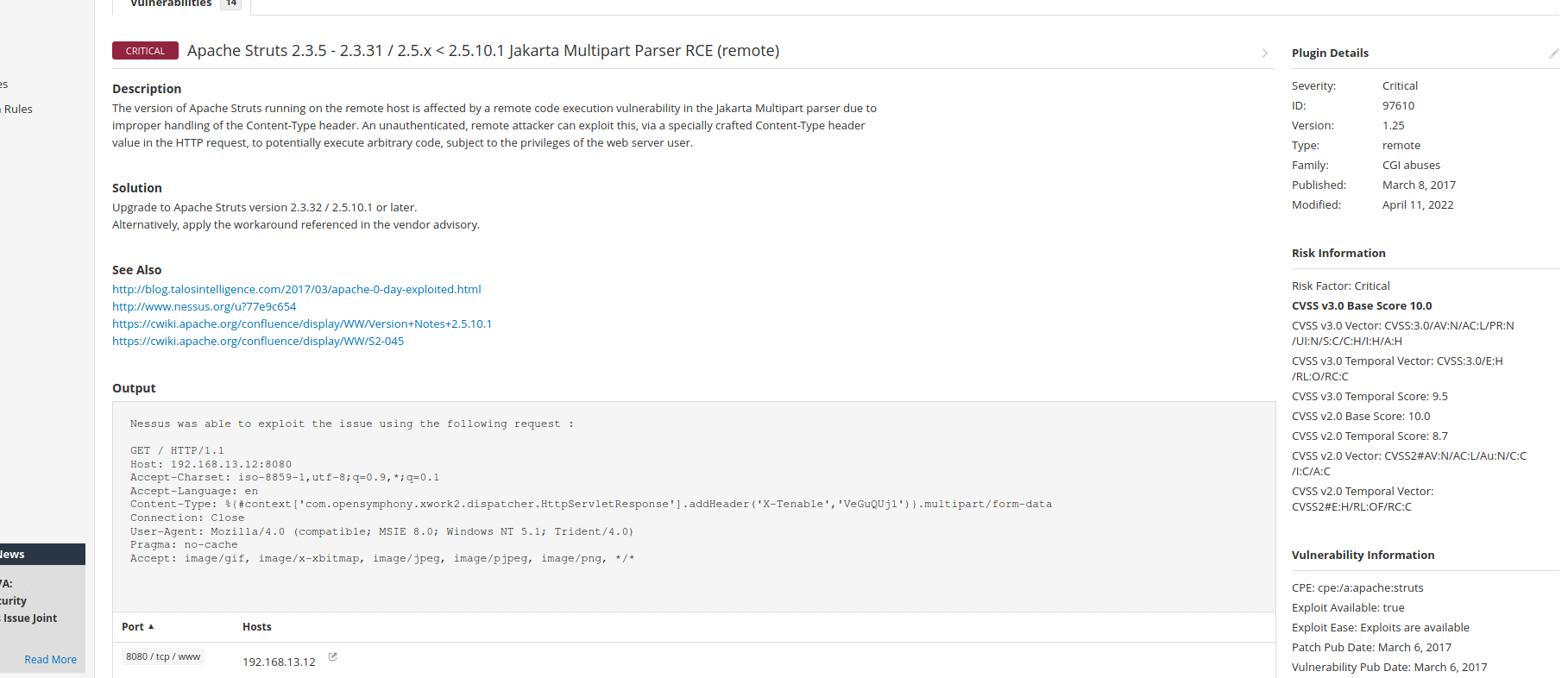
During my linux scan I used basic open source intelligence to discover critical vulnerabilities.

(flags 2, 456)





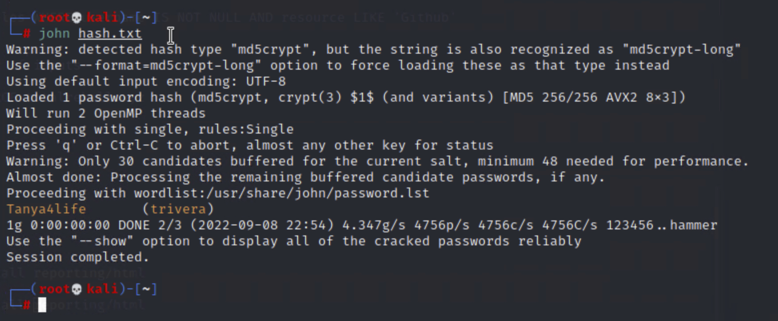


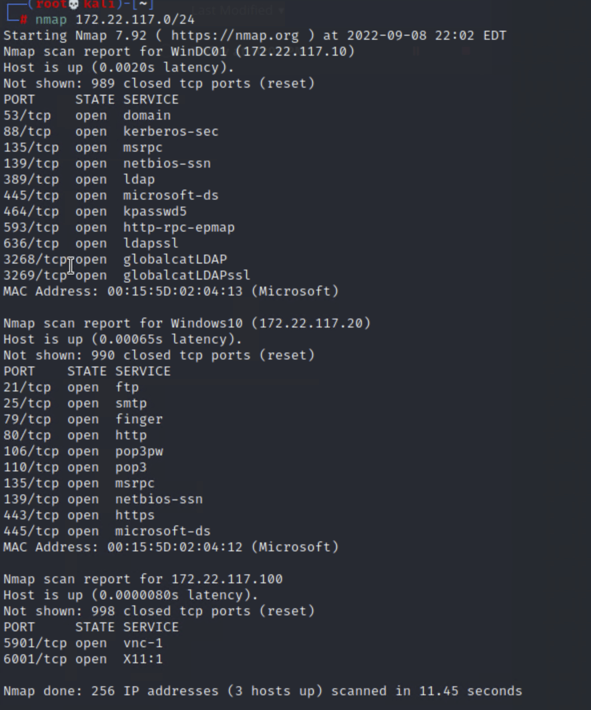
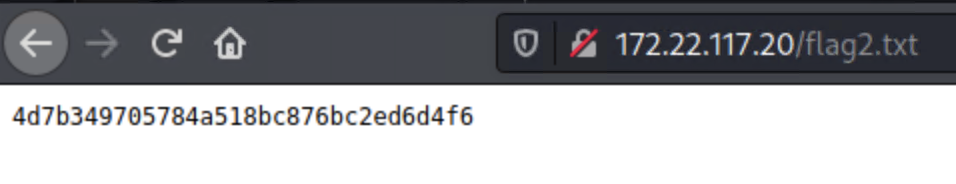
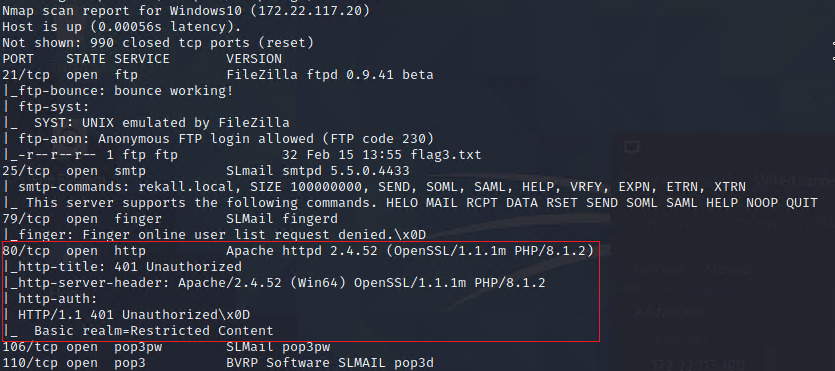
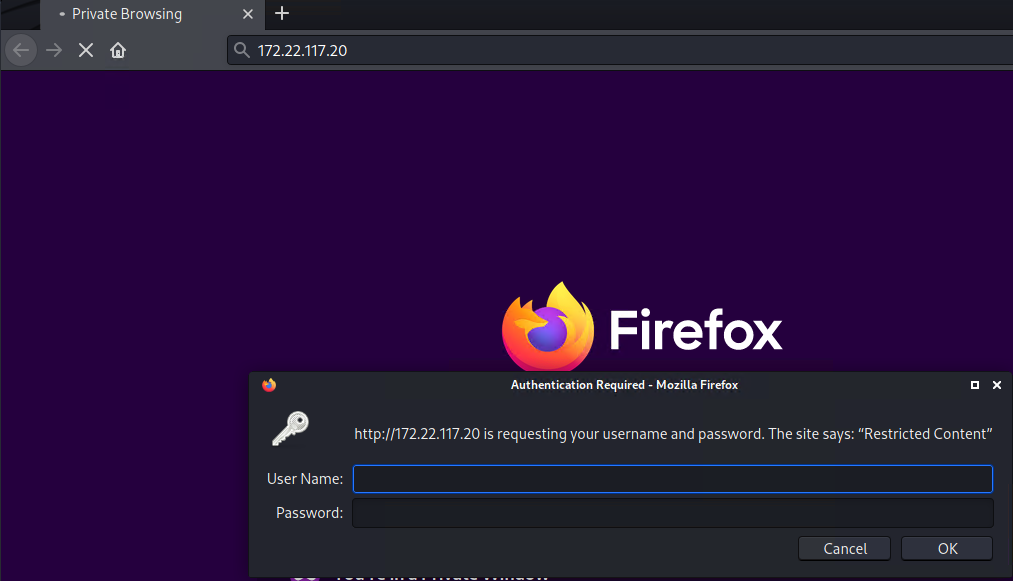


windows

Credentials held on the GitHub site. Using those credentials I gained access to the internal subnet.

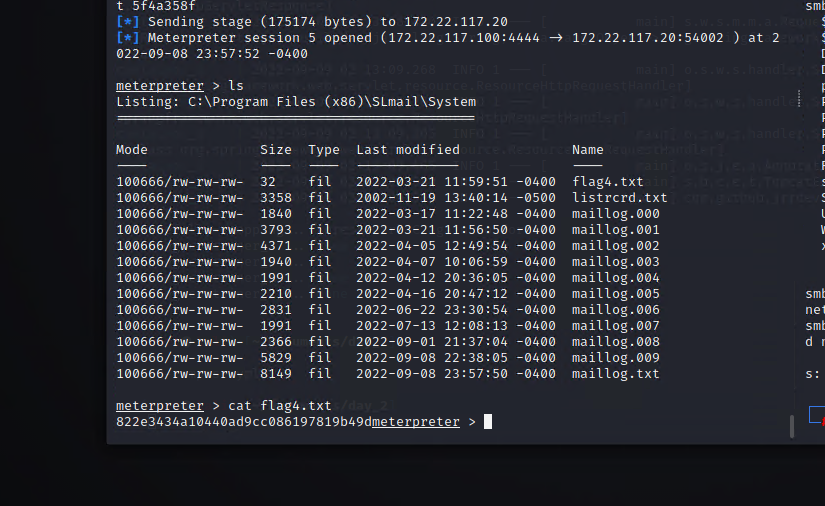
(flag 1,2)





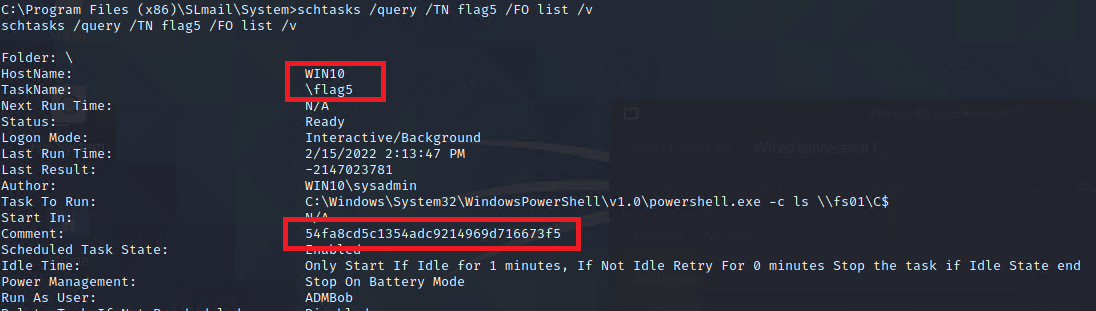
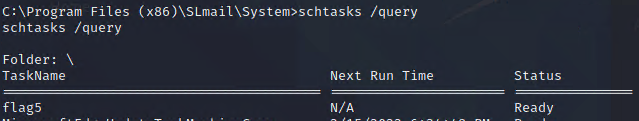
After I used FTP to anonymously link and copy a file from the subnet. Using Metasploit I was able to use an exploit to see the contents of files.

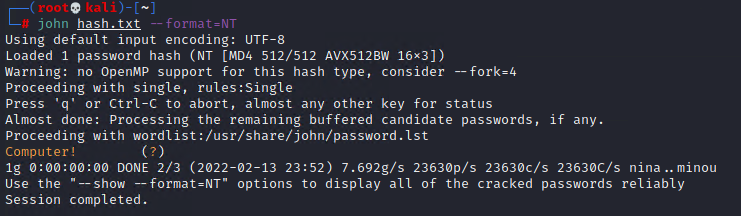
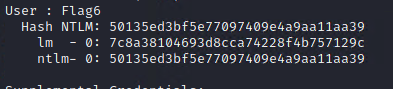
(flag4)

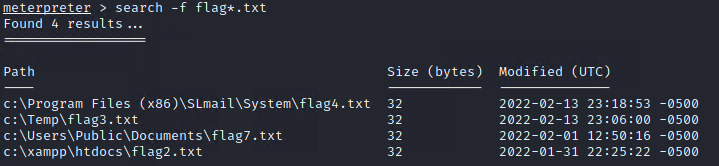


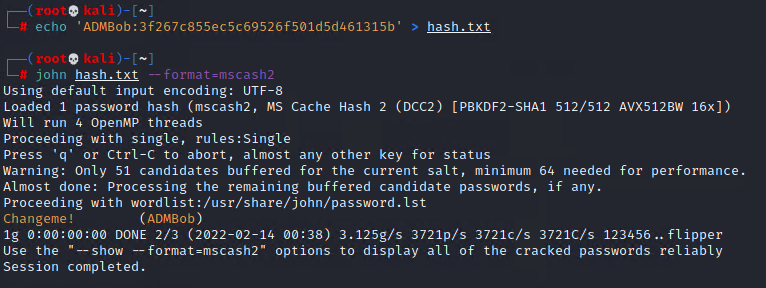
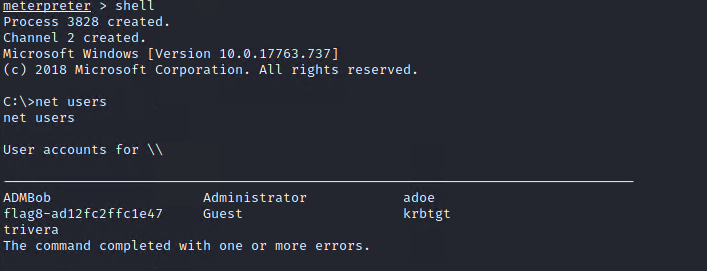
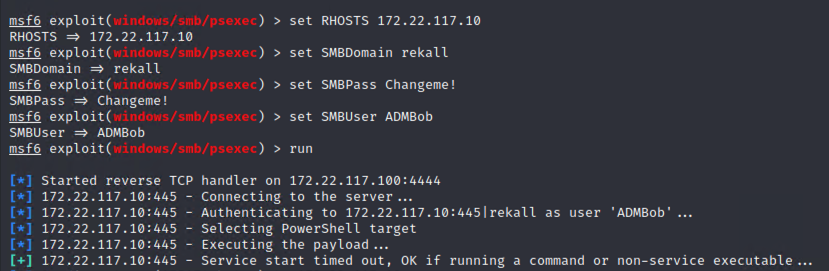
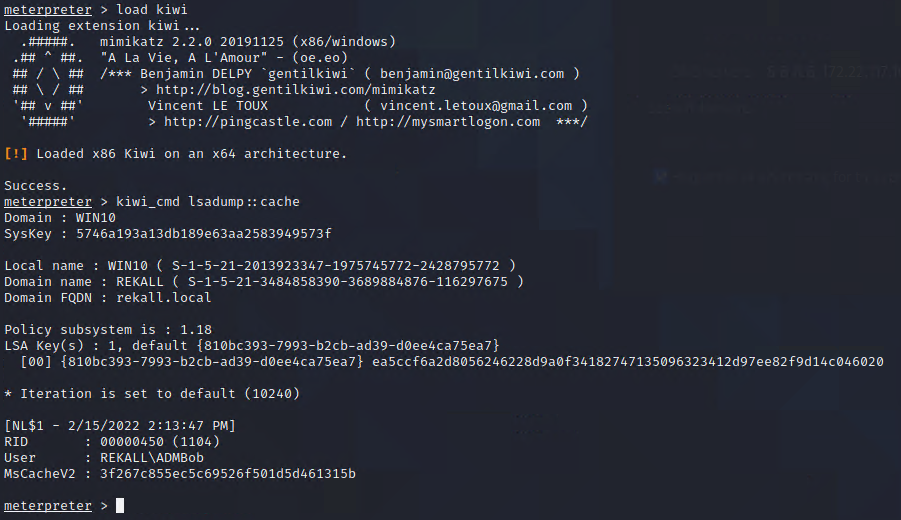
While continuing to use Metasploit I was able to dump admin credentials using kiwi. Once I had admin level credentials I was able to navigate laterally move through the system and explore files and gain access to more password hashes and files.

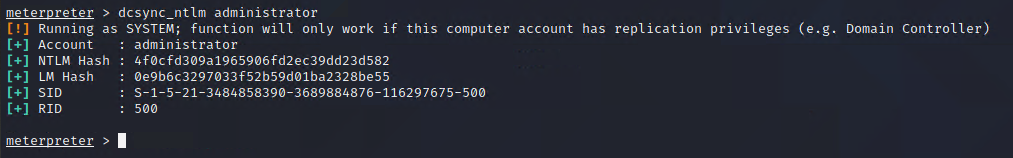
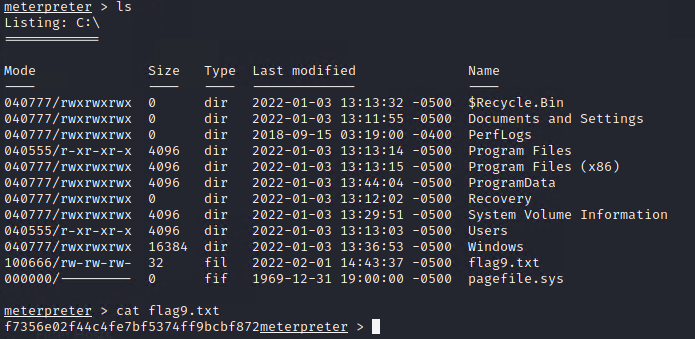
Flags 5-10











## 

## Summary Vulnerability Overview

| **Vulnerability** | **Severity** |
| --- | --- |
| Apache Struts 2.3.5 - 2.3.31 Jakarta Multi perser RCE | **Critical** |
| XSS vulnerabilities | **high** |
| failed file validation | **medium** |
| sensitive file names easily displayed | **medium** |
| Unsecure Credentials | **critical** |
| unpatched/outdated software | **high** |
| credentials outside website | **c** |
|  | **x** |
|  | **x** |
|  | **x** |
|  | **x** |
|  | **x** |
|  | **x** |
|  | **x** |
|  | **x** |
|  | **x** |
|  | **x** |
|  | **x** |
|  | **x** |
|  | **x** |
|  | **x** |

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

| **Scan Type** | **Total** |
| --- | --- |
| Hosts | 3 |
| Ports | 4 |

| **Exploitation Risk** | **Total** |
| --- | --- |
| **Critical** | 3 |
| **High** | 5 |
| **Medium** | 3 |
| **Low** | 1 |

## Vulnerability Findings

| **Vulnerability 1** | **Findings** |
| --- | --- |
| **Title** | Apache Struts 2.3.5 - 2.3.31 Jakarta Multi perser RCE |
| **Type (Web app / Linux OS / WIndows OS)** | linux |
| **Risk Rating** | Critical |
| **Description** |  |
| **Images** | see image from linux summery |
| **Affected Hosts** |  |
| **Remediation** | patch with updates |

| **Vulnerability 2** | **Findings** |
| --- | --- |
| **Title** | XSS |
| **Type (Web app / Linux OS / WIndows OS)** | web app |
| **Risk Rating** | high |
| **Description** | cross site scripting lets intruders place their own code and get results. |
| **Images** | see summary |
| **Affected Hosts** | application |
| **Remediation** | input validation |

| **Vulnerability 3** | **Findings** |
| --- | --- |
| **Title** | failed file validation |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | high |
| **Description** | file inputs can be used to input code and malicious files |
| **Images** | see summary |
| **Affected Hosts** | Web App |
| **Remediation** | File type validation |

| **Vulnerability 4** | **Findings** |
| --- | --- |
| **Title** | Credentials displayed on web app |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | Login Credentials are printed on the background of the login.php page. |
| **Images** | see summery |
| **Affected Hosts** | Web App |
| **Remediation** | Validatie that all code is sanitized before updating |

| **Vulnerability 5** | **Findings** |
| --- | --- |
| **Title** | Documents containing credentials |
| **Type (Web app / Linux OS / WIndows OS)** | Linux |
| **Risk Rating** | High |
| **Description** | File contained active credentials in plain text. |
| **Images** | See Summary |
| **Affected Hosts** | SSH |
| **Remediation** | Never store credentials in a text file on your folders. |

| **Vulnerability 6** | **Findings** |
| --- | --- |
| **Title** | Unpatched Software allows for remote code execution |
| **Type (Web app / Linux OS / WIndows OS)** | Linux |
| **Risk Rating** | medium |
| **Description** | Outdated software allows for exploitable attack surface |
| **Images** | See Summary |
| **Affected Hosts** | Linux |
| **Remediation** | patch software with update. |

| **Vulnerability 7** | **Findings** |
| --- | --- |
| **Title** | Credentials Outside website |
| **Type (Web app / Linux OS / WIndows OS)** | Web App |
| **Risk Rating** | Critical |
| **Description** | Credentials stored on separate site in plain text |
| **Images** | see summjary |
| **Affected Hosts** |  |
| **Remediation** | secure all credentials at all times. |

Add any additional vulnerabilities below.