

# **Cyber Security Community**



**Nama Lengkap : Satya Kusuma & Vincentius Farrel**

## C0rrupt – 250 – Forensic

Hint: Try fixing the file header

c0rrupt

250 points

Tags: picoCTF 2019 Forensics

AUTHOR: DANNY

Description

We found this [file](#). Recover the flag.

Hints 1

3,310 solves / 3,452 users attempted (96%)

63% Liked

picoCTF{FLAG}

Submit Flag

Pada soal diberikan sebuah “file”

```
(lonce@lonce)-[~/Downloads]
$ file mystery\1\
mystery(1): data
```

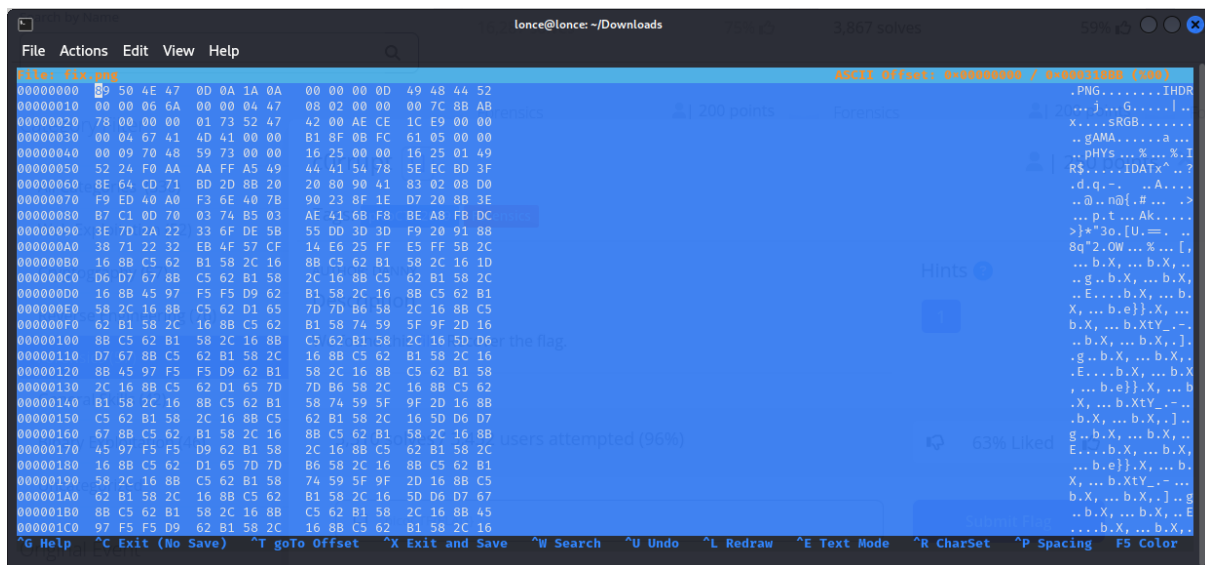
Ketika dicek tidak memberikan tipe file yang spesifik, jadi saya coba buat melihat hex filenya.

```
(lonce@lonce)-[~/Downloads]
$ xxd -g 4 mystery\1\
00000000: 89654e34 0d0ab0aa 0000000d 43224452 .eN4.....C"DR
00000010: 0000066a 00000447 08020000 007c8bab ... j ... G.... | ..
00000020: 78000000 01735247 4200aece 1ce90000 x....sRGB.....
00000030: 00046741 4d410000 b18f0bfc 61050000 .. gAMA.....a ...
00000040: 00097048 5973aa00 16250000 16250149 .. pHys ... % ... %.I
00000050: 5224f0aa aaffa5ab 44455478 5eecbd3f R$.....DETx^.. ?
00000060: 8e64cd71 bd2d8b20 20809041 830208d0 .d.q.-. .. A....
00000070: f9ed40a0 f36e407b 90238f1e d7208b3e .. @ .. n@{ .# ... .>
```

Dari bentuk struktur hex, file ini merupakan file .PNG contohnya seperti hex file .PNG seperti dibawah ini:

```
(lonce@lonce)-[~/Downloads]
$ xxd -g 4 Ninja-and-Prince-Genji-Ukiyoe-Utagawa-Kunisada.flag.png
00000000: 89504e47 0d0a1a0a 0000000d 49484452 .PNG.....IHDR
00000010: 00000432 000005dc 08020000 005ff239 ... 2....._.9
00000020: 31000100 00494441 54789c9c fdd99624 1....IDATx....$
00000030: 39762008 5e0022aa 66e6114c d6d2d57d 9v .^." .f.. L ... }
00000040: ce7cc57c c84c378b 6c9e4a32 63f77077 .|.|.L7.l.J2c.pw
00000050: 3377db75 9505b880 88aa9979 4426a7ba 3w.u.....yD& ..
00000060: 6a1ea6e7 653ea79f bac92e92 c564666c j ... e>.....dfl
00000070: bedabee8 ae220260 1e2022aa 1e995553 ..... ".` " ... US
```

Kurang lebih hex .PNG yang benar seperti itu dan tahap selanjutnya untuk memperbaiki hex file “mystery”nya.



Dengan menggunakan hexeditor untuk perbaiki .PNG, file sudah bisa di buka tapi ada error ketika ngebuka file .PNG ini.

Fatal error reading PNG image file: PNG unsigned integer out of range

Ketika di periksa dengan menggunakan pngcheck terdapat error di bagian chunk pHYS

```
(lonce@lonce)-[~/Downloads]
$ pngcheck fix2.png
fix2.png CRC error in chunk pHYS (computed 38d82c82, expected 495224f0)
ERROR: fix2.png
```

Dan ternyata benar valuenya sangat tinggi

```
Pixels Per Unit X      : 2852132389
Pixels Per Unit Y      : 5669
Pixel Units             : meters
```

Ini format dasar pHYS

The **pHYS** chunk specifies the intended pixel size or aspect ratio for display of the image. It contains:

Pixels per unit, X axis	4 bytes (PNG unsigned integer)
Pixels per unit, Y axis	4 bytes (PNG unsigned integer)
Unit specifier	1 byte

Karena setiap unit per pixels hanya berbeda 1 bytes jadi lebih masuk akal kalo disain sama yang Y, dan dengan hex yang sekarang yang menjadi penyebab kenapa nilai X sangat besar.

```
(lonce@lonce)-[~/Downloads]
$ pngcheck fix2.png
fix2.png invalid chunk length (too large)
ERROR: fix2.png
```

masih bermasalah tapi sekarang enggak spesifik kayak tadi.

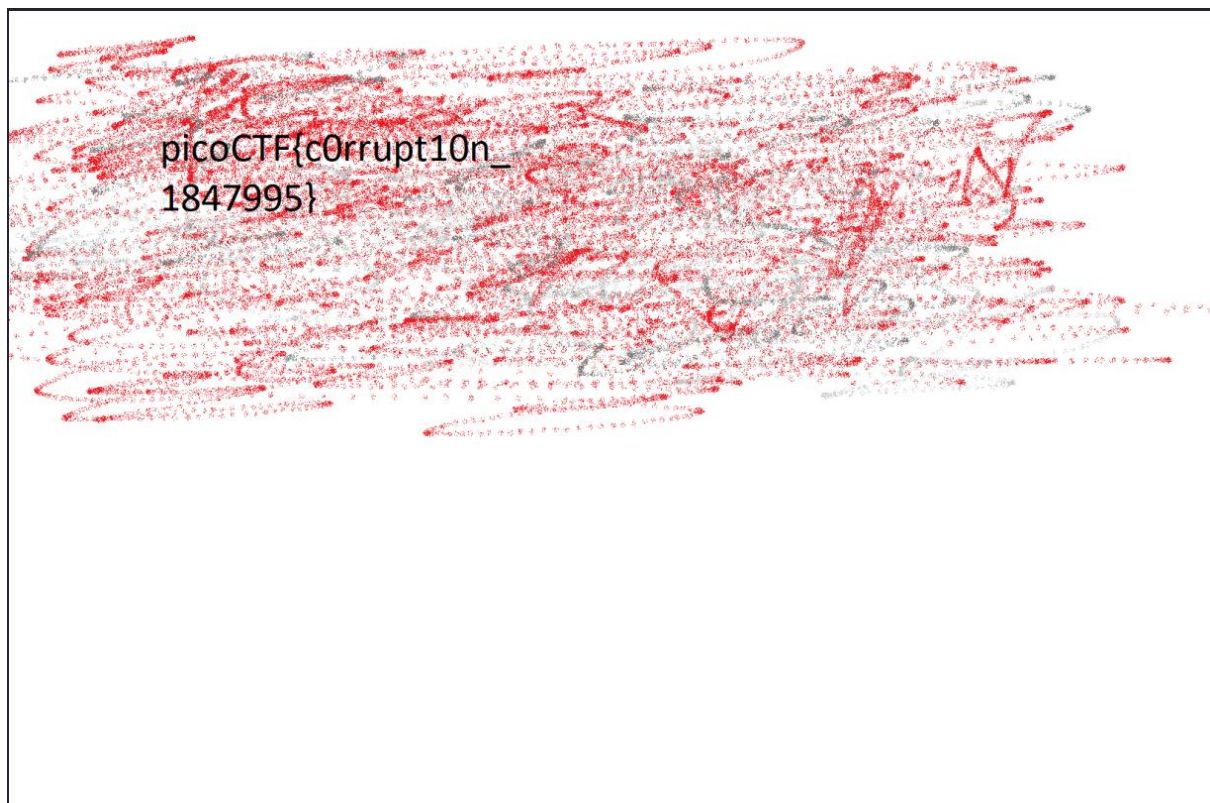
```
00000030 00 04 67 41 4D 41 00 00 B1 8F 0B FC 61 05 00 00 ..gAMA.....a...
00000040 00 09 70 48 59 73 00 00 16 25 00 00 16 25 01 49 ..pHYs...%...%.I
00000050 52 24 F0 AA AA FF A5 49 44 41 54 78 5E EC BD 3F R$....IDATx^..3
00000060 8E 64 CD 71 BD 2D 8B 20 20 80 90 41 83 02 08 D0 .d.q.-. ..A...
00000070 F9 ED 40 A0 F3 6E 40 7B 90 23 8F 1E D7 20 8B 3E ..@..n@{.#.....?
00000080 B7 C1 0D 70 03 74 B5 03 AE 41 6B F8 BE A8 FB DC ...p.t...Ak....
```

Dapat dilihat bytes setelah pHYS lumayan besar 0xAAAAFFA5, karena sifat IDAT berurutan kita bisa hitung size yang tepat. Dengan bantuan format chunk yaitu 4 byte length, 4 byte chunk type, dan 4 byte CRC.

```
(lonce@lonce)-[~/Downloads]
$ binwalk -R "IDAT" fix2.png
```

DECIMAL	HEXADECIMAL	DESCRIPTION
87	0x57	Raw signature (IDAT)
65544	0x10008	Raw signature (IDAT)
131080	0x20008	Raw signature (IDAT)
196616	0x30008	Raw signature (IDAT)

Pertama kurangi IDAT yang kedua dengan yang pertama lalu kurangi dengan format chunk 4,4,4. Jadi seperti ini:  $0x10008 - 0x57 - 4 - 4 - 4 = 0xFFA5$  karena kebetulan value yang awal sama 0xFFA5 tinggal ganti jadi 0000.



Dan file dapat dibuka.

Flag: picoCTF{c0rrupt10n\_1847995}

## Sleuthkit Apprentice – 200 – Forensic

Hints: no hint

### Sleuthkit Apprentice

| 200 points

Tags: picoCTF 2022 Forensics disk

AUTHOR: LT 'SYREAL' JONES

Hints

#### Description

(None)

Download this disk image and find the flag.

Note: if you are using the webshell, download and extract the disk image into `/tmp` not your home directory.

- [Download compressed disk image](#)

5,663 solves / 6,018 users attempted (94%)

91%  
Liked

picoCTF{FLAG}

Submit Flag

Diberikan sebuah disk image, pada soalnya hanya disuruh download (unzip juga) dan mencari flagnya di dalam disk image.

Cara menganalisa file disk image ini dengan menggunakan tools yaitu The Sleuth Kit (TSK).

```
(lonce@lonce) - [~/Downloads]
$ mmls disk.flag.img
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

Slot      Start      End      Length  Description
000:  Meta      0000000000  0000000001  0000000001 Primary Table (#0)
001:  _____ 0000000000  0000002047  0000002048 Unallocated
002:  000:000  0000002048  0000206847  0000204800 Linux (0x83)
003:  000:001  0000206848  0000360447  0000153600 Linux Swap / Solaris x86 (0x82)
004:  000:002  0000360448  0000614399  0000253952 Linux (0x83)
```

Pertama dump partion table dengan command 'mmls' dan mencari offsetnya yang akan digunakan untuk display list file dengan offset pada colom Start. Pada offset 2048 tidak ada isinya dan 206848 juga tidak ada output apapun. Pada offset 360448 terdapat list file.

```

(lonce@lonce)-[~/Downloads]
$ fls -o 360448 disk.flag.img
d/d 451:      home
d/d 11:  lost+found
d/d 12:  boot
d/d 1985:    etc
d/d 1986:    proc
d/d 1987:    dev
d/d 1988:    tmp
d/d 1989:    lib
d/d 1990:    var
d/d 3969:    usr
d/d 3970:    bin
d/d 1991:    sbin
d/d 1992:    media
d/d 1993:    mnt
d/d 1994:    opt
d/d 1995:    root
d/d 1996:    run
d/d 1997:    srv
d/d 1998:    sys
d/d 2358:    swap
V/V 31745:   $OrphanFiles

```

Dari banyak folder, folder root yang menarik (ya karena root sangat penting). Menggunakan fls lagi dengan inode 1995.

```

(lonce@lonce)-[~/Downloads]
$ fls -o 360448 disk.flag.img 1995
r/r 2363:    .ash_history
d/d 3981:    my_folder

```

Lalu dengan command yang sama tapi inode 3981.

```

(lonce@lonce)-[~/Downloads]
$ fls -o 360448 disk.flag.img 3981
r/r * 2082(realloc):  flag.txt
r/r 2371:    flag.uni.txt

```

Karena tinggal file .txt, jadi saya coba untuk extract datanya dengan menggunakan icat.

```



(lonce@lonce)-[~/Downloads]
$ icat -o 360448 disk.flag.img 2371
picoCTF{by73_5urf3r_3497ae6b}

```

Flag: **picoCTF{by73\_5urf3r\_3497ae6b}**

## Picobrowser – 200 – Web Exploitation

picobrowser 

 | 200 points 

Tags: picoCTF 2019 Web Exploitation

AUTHOR: ARCHIT

### Description

This website can be rendered only by **picobrowser**, go and catch the flag!

<https://jupiter.challenges.picoctf.org/problem/26704/> ([link](#)) or

<http://jupiter.challenges.picoctf.org:26704>

Hints 

1

You don't need to download a new web browser

23,636 solves / 24,726 users attempted (96%)



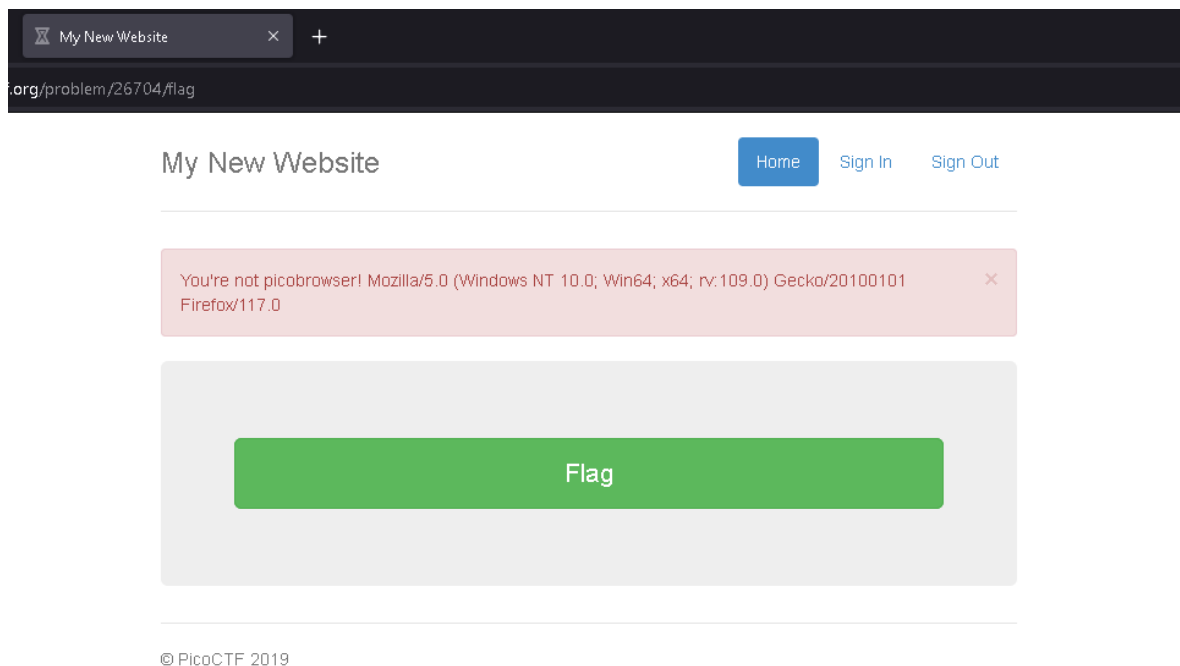
89% Liked



 picoCTF{FLAG}

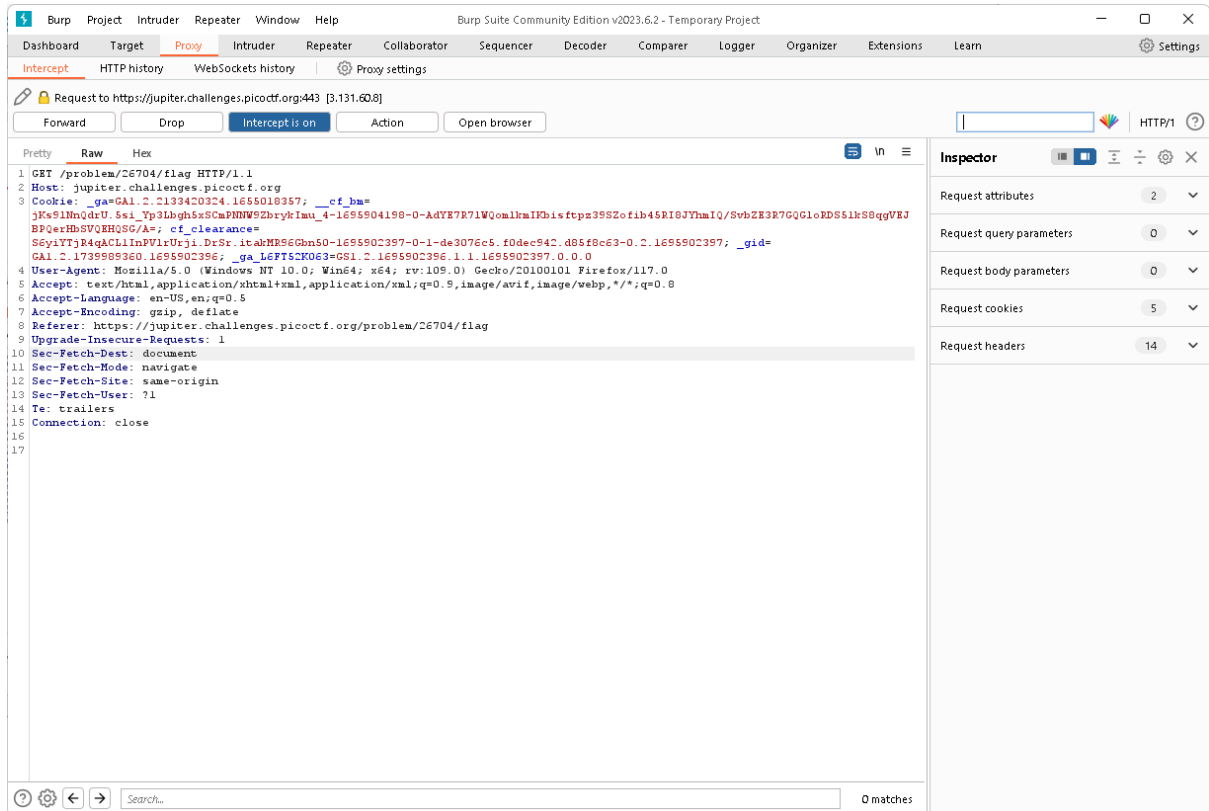
Submit Flag

Buka dulu websitenya dan nanti akan muncul seperti gambar di bawah. (Gambar di bawah sudah diklik button “Flag”nya maka muncul notifikasi merah di atasnya.

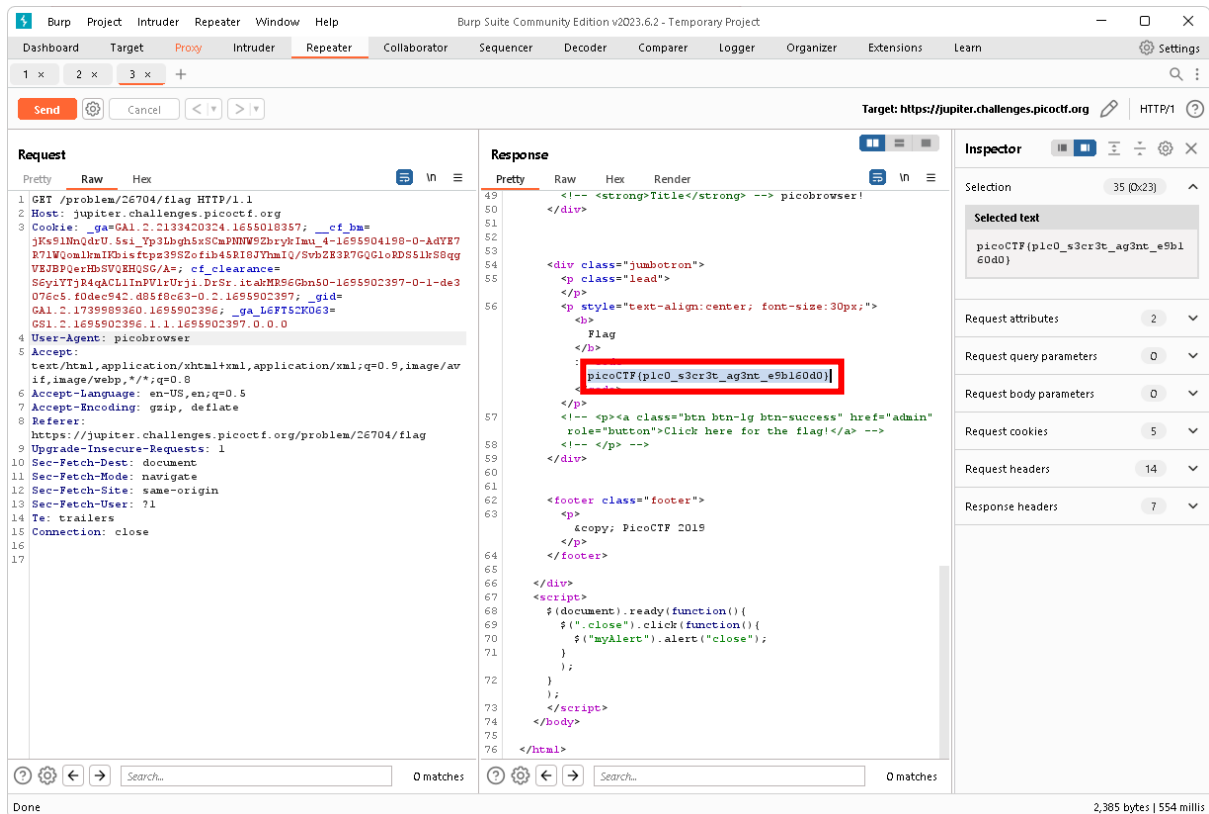


Seperti yang diberitahu di notifikasinya, dikatakan bahwa browsernya bukan “picobrowser” sehingga website/flagnya tidak dapat digunakan. Jadi langsung saja kita buka burpsuite buat intercept requestnya.







Nah dari intercept burpsuite di atas, kita kirimkan saja ke repeater (ctrl + r) lalu di send requestnya dan nanti akan muncul di kanan berupa response. Jika responsenya di scroll, maka akan muncul flagnya seperti yang ada di gambar bawah.



Flag: `picoCTF{p1c0_s3cr3t_ag3nt_e9b160d0}`

# FileAndOpen – 200 – Forensic

FindAndOpen 

 | 200 points 

Tags: picoCTF 2023 Forensics

AUTHOR: MUBARAK MIKAIL

## Description

Someone might have hidden the password in the trace file.

Find the key to unlock [this file](#). [This tracefile](#) might be good to analyze.

Hints 

1 2

Don't try to use a password cracking tool, there are easier ways here.

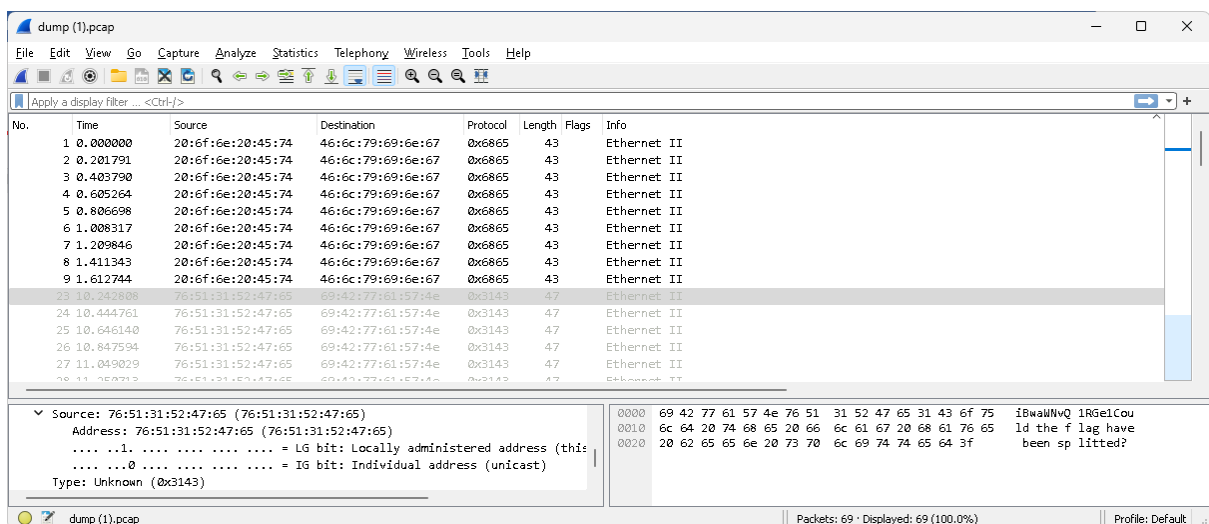
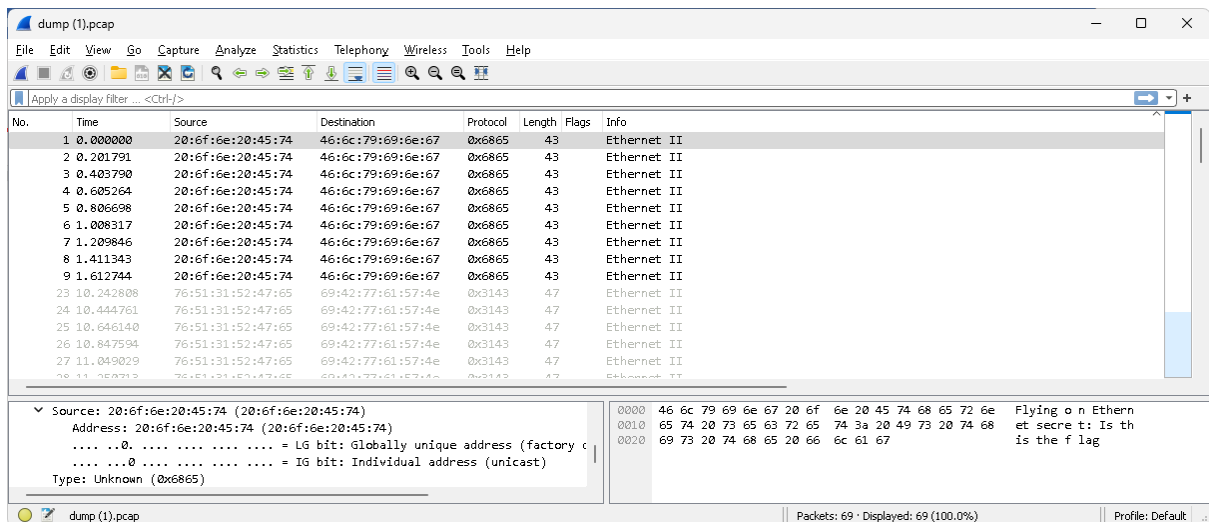
3,074 solves / 3,684 users attempted (83%)

 30% Liked 

 picoCTF{FLAG}

Submit Flag

Setelah didownload kedua file tersebut merupakan locked zip bernama flag dan dump.pcap. Seperti yang kita ketahui kalau file pcap merupakan file wireshark, jadi langsung saja kita buka di wireshark. Di dalam log tersebut terdapat beberapa hal yang menarik perhatian seperti gambar-gambar di bawah ini.



dump (1).pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Flags	Info
41	13.868619	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
42	14.069692	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
43	14.270997	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
44	14.472105	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
45	14.673443	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
46	14.874381	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
47	15.075729	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
48	24.240874	50:4a:47:54:46:52	41:41:42:42:48:48	0x4c4b	70		Ethernet II
49	48.459761	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
50	48.661266	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
51	48.862635	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
52	49.064175	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
53	49.265187	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
54	49.466470	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
55	49.668273	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
56	49.870142	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
57	50.071670	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
58	54.127340	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
59	54.329337	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
60	54.531171	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
61	54.733158	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
62	54.934646	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
63	55.136085	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
64	55.339602	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II

.....1. .... = LG bit: Locally administered address (this) .....0 ..... = IG bit: Individual address (unicast)

Type: Unknown (0x7361)

Data (35 bytes)

Data: 62616261626b6a6141534b424b534241435656415653444453535344534b4a424a53

0000 50 42 77 61 57 55 76 51 31 52 47 65 73 61 62 61 P8waUwQ 1R6saba  
0010 62 61 62 6b 6a 61 41 53 4b 42 4b 53 42 41 43 56 babkjaS KBKSBACV  
0020 56 41 56 53 44 44 53 53 53 53 44 53 4b 4a 42 4a VAVSDOSS S5DSK3BJ  
0030 53 S

dump (1).pcap

Packets: 69 · Displayed: 69 (100.0%)

Profile: Default

dump (1).pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Flags	Info
50	48.661266	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
51	48.862635	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
52	49.064175	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
53	49.265187	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
54	49.466470	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
55	49.668273	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
56	49.870142	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
57	50.071670	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
58	54.127340	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
59	54.329337	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
60	54.531171	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
61	54.733158	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
62	54.934646	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
63	55.136085	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
64	55.339602	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II

.....1. .... = LG bit: Locally administered address (this) .....0 ..... = IG bit: Individual address (unicast)

Type: Unknown (0x3144)

Data (32 bytes)

Data: 617962652074727920636865636b696e6720746865206f746865722066696c65

0000 50 42 77 61 57 55 76 51 31 52 47 65 31 4d 61 79 P8waUwQ 1R6e1May  
0010 62 65 20 74 72 79 20 63 68 65 63 6b 69 6e 67 20 be try c hecking  
0020 74 68 65 20 6f 74 68 65 72 20 66 69 6c 65 the othe r file

dump (1).pcap

Packets: 69 · Displayed: 69 (100.0%)

Profile: Default

dump (1).pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Flags	Info
50	11.403620	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
51	11.404952	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
52	11.856446	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
53	12.287812	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
54	12.409421	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
55	12.568226	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
56	12.818861	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
57	13.003255	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
58	13.205042	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
59	13.406110	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
60	13.607794	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
61	13.808619	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
62	14.009602	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
63	14.210997	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
64	14.472105	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
65	14.673443	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
66	14.874381	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
67	15.075729	76:51:31:52:47:65	69:42:77:61:57:4e	0x3143	47		Ethernet II
49	48.459761	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
50	48.661266	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
51	48.862635	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
52	49.064175	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
53	49.265187	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
54	49.466470	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
55	49.668273	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
56	49.870142	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
57	50.071670	76:51:31:52:47:65	50:42:77:61:57:55	0x7361	49		Ethernet II
58	54.127340	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
59	54.329337	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
60	54.531171	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
61	54.733158	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
62	54.934646	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
63	55.136085	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
64	55.339602	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
65	55.538161	76:51:31:52:47:65	50:42:77:61:57:55	0x3144	46		Ethernet II
66	56.484203	172.16.93.1	224.0.0.251	NMOS	132		Standard query 0x0000 PTR _googlecast._tcp.local, "Q" question TXT ChromeCast-18c2a8da3045b73bacc93a7a1f1998a _googlecast._tcp.local, "Q" question
67	56.484469	1680:4085:5fff:fe0: ffd1:1b	NMOS	152			Standard query 0x0000 PTR _googlecast._tcp.local, "Q" question TXT ChromeCast-18c2a8da3045b73bacc93a7a1f1998a _googlecast._tcp.local, "Q" question
68	56.490587	172.16.93.2	224.0.0.251	NMOS	414		Standard query response 0x0000 PTR _googlecast._tcp.local, "Q" question PTR ChromeCast-18c2a8da3045b73bacc93a7a1f1998a _googlecast._tcp.local, "Q" question
69	56.490587	172.16.93.2	224.0.0.251	NMOS	327		Standard query response 0x0000 TXT ChromeCast-18c2a8da3045b73bacc93a7a1f1998a _googlecast._tcp.local, "Q" question TXT

Coloring Rule Strings: eth[0] & 1

Ethernet II, Src: 50:4a:47:54:46:52 (50:4a:47:54:46:52), Dst: 41:41:42:42:48:48 (41:41:42:42:48:48)

Destination: 41:41:42:42:48:48 (41:41:42:42:48:48)

Address: 41:41:42:42:48:48 (41:41:42:42:48:48)

.....0 ..... = LG bit: Globally unique address (factory default)

.....1 ..... = 30 bits: Group address (multicast/broadcast)

Source: 50:4a:47:54:46:52 (50:4a:47:54:46:52)

Address: 50:4a:47:54:46:52 (50:4a:47:54:46:52)

.....0 ..... = LG bit: Globally unique address (factory default)

.....1 ..... = 30 bits: Individual address (unicast)

Type: Unknown (0x0c4b)

Data (56 bytes)

Data: 50476870637962706379623061475567633256663656304f69427761574e7051315247

0000 41 41 42 42 48 48 50 4a 47 54 46 52 4c 4b 4c 4d ADDHWP 0TRFLW  
0010 18 c2 a8 da 30 45 b7 3b acc 9 3a 7a 1f 19 98 a 0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0020 50 4a 47 54 46 52 4c 4b 4c 4d 41 41 42 42 48 48 50 4a 47 54 46 52 4c 4b 4c 4d 41 41 42 42 48 48  
0030 41 41 42 42 48 48 50 4a 47 54 46 52 4c 4b 4c 4d 41 41 42 42 48 48 50 4a 47 54 46 52 4c 4b 4c 4d 41 41 42 42 48 48

dump (1).pcap

Packets: 69 · Displayed: 69 (100.0%)

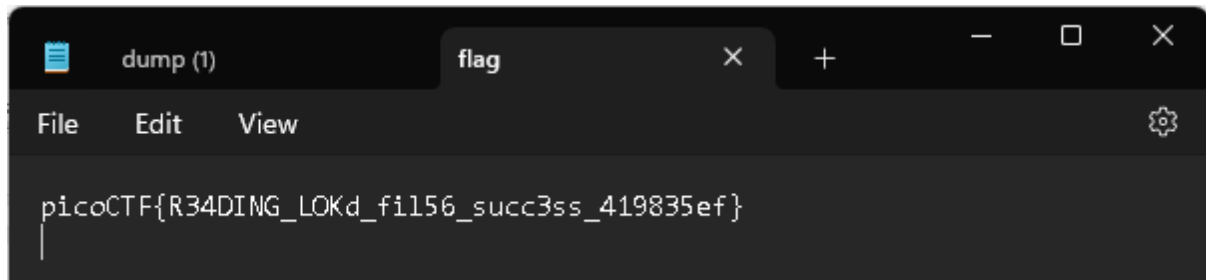
Profile: Default

Nah dari sekian banyak yang kita lihat, di yang gambar terakhir ini ternyata terdapat base64.

Setelah memasukkannya ke cyberchef, didapatkan arti:

**This is the secret: picoCTF{R34DING\_LOKd\_**

Lalu karna kita sudah mendapatkan awalan flagnya, kita coba saja untuk memasukkannya sebagai password zip tadi. Dan ya berhasil!

A screenshot of a text editor window with a dark theme. The window has a title bar with a tab labeled 'dump (1)' and a sub-tab labeled 'flag'. Below the title bar is a menu bar with 'File', 'Edit', and 'View' options, and a settings gear icon on the right. The main text area contains the string 'picoCTF{R34DING\_LOKd\_fil56\_succ3ss\_419835ef}' with a cursor at the end of the line.

```
picoCTF{R34DING_LOKd_fil56_succ3ss_419835ef}
```

Flag: **picoCTF{R34DING\_LOKd\_fil56\_succ3ss\_419835ef}**

# Secrets – 200 – Web Exploitation

Secrets

| 200 points

Tags: picoCTF 2022 Web Exploitation

AUTHOR: GEOFFREY NJOGU

Hints

Description

1

We have several pages hidden. Can you find the one with the flag?

folders folders folders

The website is running [here](#).

12,684 solves / 13,117 users attempted (97%)

68% Liked

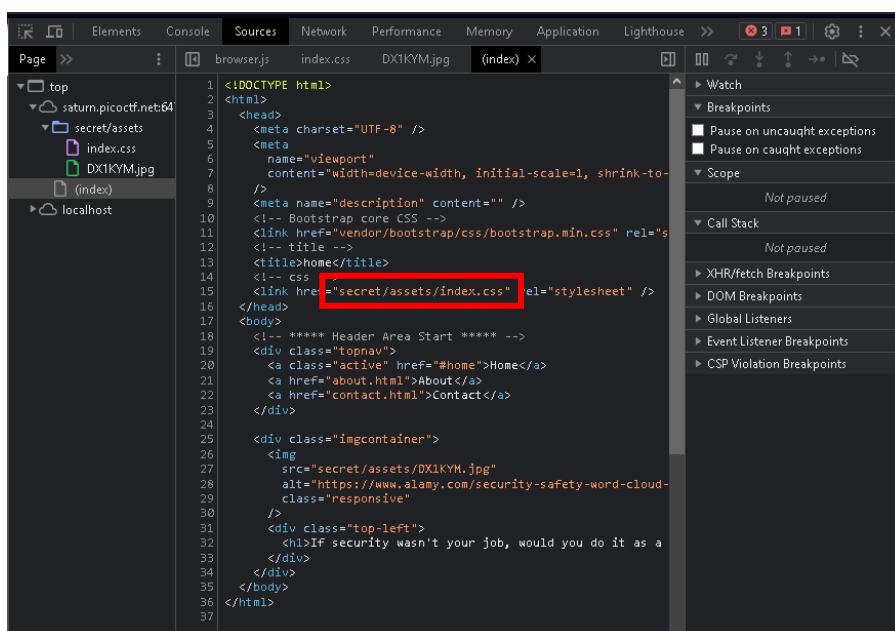
picoCTF{FLAG}

Submit Flag

Buka dulu websitenya dan akan muncul tampilan seperti di bawah.



Seperti biasa kita akan inspect dulu webnya, dan disini kita menemukan ada file index.html dan folder secret/assets. Karna hints dari soalnya “folders folders folders” jadi kita coba aja masukan href yang ada di index.html ke url kita.



Hasilnya tidak muncul apa-apa. Kalau begitu kita coba untuk memundurkan 1 folder menjadi *secret/assets/* saja dan ternyata masih tidak bisa. Yang terakhir hanya tersisa *secret/* saja dan ketika dicoba berhasil, tampilannya akan seperti gambar di bawah.

Finally. You almost found me. you are doing well



Masih dengan teknik yang sama, inspect element, kita akan mencoba melihat kembali ke dalam sourcenya dan kali ini kita menemukan folder hidden dan href lagi yang mengarah ke *hidden/file.css*

```
Page >> (index) x file.css
▼ top
  ▼ saturn.picocft.net:647
    ▼ secret
      ▼ hidden
        file.css
        (index)
  media1.tenor.com
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title></title>
5     <link rel="stylesheet" href="hidden/file.css" />
6   </head>
7
8   <body>
9     <h1>Finally. You almost found me. you are doing well</h1>
10    
7   </head>
8   <body>
9     <form>
10      <div class="container">
11        <form method="" action="/secret/assets/popup.js">
```

Langsung saja kita masukkan */superhidden/* ke dalam url kita. Boom, lagi-lagi berhasil dan menampilkan tampilan di bawah.

## Finally. You found me. But can you see me

Memang flagnya tidak kelihatan karena diberikan warna putih untuk fontnya. Disini ada dua cara untuk melihatnya, pertama dengan memblok tulisannya dan yang kedua melihat inspect element.

## Finally. You found me. But can you see me



picoCTF{succ3ss\_@h3n1c@10n\_790d2615}

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <link rel="stylesheet" href="mycss.css">
    <style type="text/css">...</style>
    <style type="text/css" id="operaUserStyle">...</style>
  </head>
  ... <body> == $0
    <h1>Finally. You found me. But can you see me</h1>
    <h3 class="flag">picoCTF{succ3ss_@h3n1c@10n_790d2615}</h3>
  </body>
</html>
```

Flag: picoCTF{succ3ss\_@h3n1c@10n\_790d2615}

## Web Gauntlet – 200 – Web Exploitation

Web Gauntlet 

 | 200 points 

Tags: picoCTF 2020 Mini-Competition Web Exploitation

AUTHOR: MADSTACKS

If the flag is not displayed after completing this challenge, try clearing your cookies. Cookies set by other challenges may prevent the flag from displaying properly.

Hints 

1 2 3 4 5

For some filters it may be hard to see the characters, always (always) look at the raw hex in the response.

### Description

Can you beat the filters? Log in as admin <http://jupiter.challenges.picoctf.org:54319/>  
<http://jupiter.challenges.picoctf.org:54319/filter.php>

8,928 solves / 9,383 users attempted (95%)

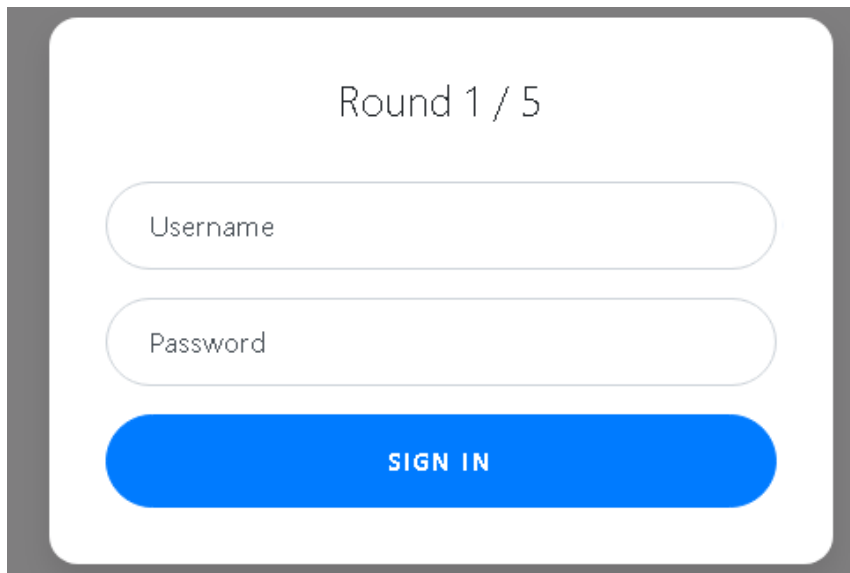
 83% Liked 

 picoCTF{FLAG}

Submit Flag

Source ngerjain: <https://www.invicti.com/blog/web-security/sql-injection-cheat-sheet/>

Di soal dikasih tau kalo kita harus login sebagai admin dan dikasih 2 website yang dimana website pertama merupakan web untuk kita mengerjakan soalnya (ada 5 soal) dan yang website ke dua adalah web untuk informasi tentang filternya (symbol atau syntax yang ga bisa dipake buat ngerjain). Untuk tampilan web 1 dan 2 seperti yang ada di bawah ini.



Round 1 / 5

Username

Password

SIGN IN

Round1: or

Seperti yang sudah diketahui, filternya adalah **or**. Karna kita harus login sebagai admin, disini kita akan menggunakan username “admin” dan password “a” (a for asal keisi).



Disini kita coba aja untuk masukkin payloadnya **admin' --**

#1 kenapa gitu sih payloadnya? Ok ini dia penjelasannya

Sebelumnya kita udah tau dari hint kalo ini sqlite dan kalo sengaja/memang salah payloadnya, websitenya ini bakal kasih tau SQL querynya dan disini gw coba aja masukin admin doang di usernamenya dan a di passwordnya.

```
SELECT * FROM users WHERE username='admin' AND password='a'
```

```
SELECT * FROM users WHERE username='admin' AND password='a'
```

Nah kalo kita masukin payload kita yang **admin' --** bakal jadi kayak gini hasilnya

```
SELECT * FROM users WHERE username='admin'-- AND password='a'
```

Nah hasil query di atas ini bakal ngebaca sampe adminnya aja, bagian AND ke belakang ga dibaca atau lebih tepatnya dicomment. FYI -- itu merupakan line comment sama kayak #.

```
SELECT * FROM users WHERE username='admin'
```

Round 1 / 5

Username

admin' --|

Password

•

SIGN IN

Lanjut ke round 2.

Kali ini filternya ga boleh pake -- lagi.

Round2: or and like = --

Waduh gimana ini? Haha tenang saja masih ada penggantinya 🤓

Round 2 / 5

Congrats! On to round 2

Username

admin' /\*

Password

•

SIGN IN

#2 kenapa gitu sih payloadnya? Ok ini dia penjelasannya

Seperti yang sudah kita ketahui, querynya akan dikasih tau sama websitenya dan masih sama.

Nah kalo kita masukin payload kita yang **admin'/\*** bakal jadi kayak gini hasilnya

```
SELECT * FROM users WHERE username='admin'/* AND password='a'
```

Hasilnya ini bakal sama aja kayak query sebelumnya karna /\* merupakan bagian dari /\*\*/ yang dimana merupakan inline comment. Makanya querynya itu bakal jadi gini.

```
SELECT * FROM users WHERE username='admin'
```

Lanjuttt round 3

Dan ya filternya nambah beberapa.

Round3: or and = like > < --

Karna kayaknya ga ada masalah di filternya, jadi disini kita coba masukin payload round 2, namun hasilnya berkata lain. Ya benar sekali, payloadnya tidak bisa digunakan ga tau kenapa. Tapi jangan sedih jangan risau karena kita masih punya senjata lain.

Round 3 / 5

Username

admin';

Password

\*

SIGN IN

#3 kenapa gitu sih payloadnya? Ok ini dia penjelasannya

Nah kalo kita masukin payloadnya, querynya bakal jadi kayak gini.

```
SELECT * FROM users WHERE username='admin'; AND password='a'
```

Mulai dari bagian AND ke belakang itu ga dibaca atau mungkin lebih tepatnya bakal jadi statement lain. Hal ini biasa disebut stacking queries yang dimana ngestop querynya dulu terus mulai query baru lagi.

Mantap lanjut ke round 4

Hadeh kata admin udah ga boleh dipake... Kek mana??

—

Round4: or and = like > < -- admin

Keep calm, keep chill.

Round 4 / 5

Congrats! On to round 4

Username

ad' || 'min';

Password

•

SIGN IN

#4 kenapa gitu sih payloadnya? Ok ini dia penjelasannya

Nah kalo kita masukin payloadnya, querynya bakal jadi kayak gini.

```
SELECT * FROM users WHERE username='ad' || 'min'; AND password='a'
```

Penjelasan setelah tanda titik koma (;) masih sama kayak payload round 3 dan kali ini ada tambahan symbol pipe dua kali (||) yang artinya merupakan string concatenation atau penggabungan string dan efek yang diberikan sama saja dengan tanda plus (+). Jadi hasilnya akan sama saja dengan ad + min = admin.

Asik dikit lagi beres, round 5

,

Round5: or and = like > < -- union admin

Tampak tidak ada masalah dengan payload round 4, jadi kita coba masukin aja.

Round 5 / 5

Congrats! On to round 5

Username

ad' || 'min';

Password

•

SIGN IN

Round 6 / 5

Congrats! You won! Check out filter.php

Username

Password

SIGN IN

Duar beneran bisa ga kayak sebelumnya. Langsung kita cek di web satunya lagi lalu kita dipertemukan dengan code yang udah nyusahin kita dan flagnya 🥳.

```

<?php
session_start();

if (!isset($_SESSION["round"])) {
    $_SESSION["round"] = 1;
}
$round = $_SESSION["round"];
$filter = array("");
$view = ($_SERVER["PHP_SELF"] == "/filter.php");

if ($round == 1) {
    $filter = array("or");
    if ($view) {
        echo "Round1: ".implode(" ", $filter)."<br/>";
    }
} else if ($round == 2) {
    $filter = array("or", "and", "like", "=", "--");
    if ($view) {
        echo "Round2: ".implode(" ", $filter)."<br/>";
    }
} else if ($round == 3) {
    $filter = array(" ", "or", "and", "=", "like", ">", "<", "--");
    // $filter = array("or", "and", "=", "like", "union", "select", "insert", "delete", "if", "else", "true", "false", "admin");
    if ($view) {
        echo "Round3: ".implode(" ", $filter)."<br/>";
    }
} else if ($round == 4) {
    $filter = array(" ", "or", "and", "=", "like", ">", "<", "--", "admin");
    // $filter = array(" ", "/*", "--", "or", "and", "=", "like", "union", "select", "insert", "delete", "if", "else", "true", "false", "admin");
    if ($view) {
        echo "Round4: ".implode(" ", $filter)."<br/>";
    }
} else if ($round == 5) {
    $filter = array(" ", "or", "and", "=", "like", ">", "<", "--", "union", "admin");
    // $filter = array("0", "unhex", "char", "/*", "*/", "--", "or", "and", "=", "like", "union", "select", "insert", "delete", "if", "else", "true", "false", "admin");
    if ($view) {
        echo "Round5: ".implode(" ", $filter)."<br/>";
    }
} else if ($round >= 6) {
    if ($view) {
        highlight_file("filter.php");
    }
} else {
    $_SESSION["round"] = 1;
}



// picoCTF{y0u_m4d3_1t_a5f58d5564fce237fbcc978af033c11b}

```

Flag: **picoCTF{y0u\_m4d3\_1t\_a5f58d5564fce237fbcc978af033c11b}**

## Client-side-again – 200 – Web Exploitation

Client-side-again 

 | 200 points 

Tags: picoCTF 2019 Web Exploitation

AUTHOR: DANNY

### Description

Can you break into this super secure portal?

<https://jupiter.challenges.picoctf.org/problem/56816/> ([link](#)) or

<http://jupiter.challenges.picoctf.org:56816>

Hints 

1

What is obfuscation?

15,270 solves / 17,367 users attempted (88%)



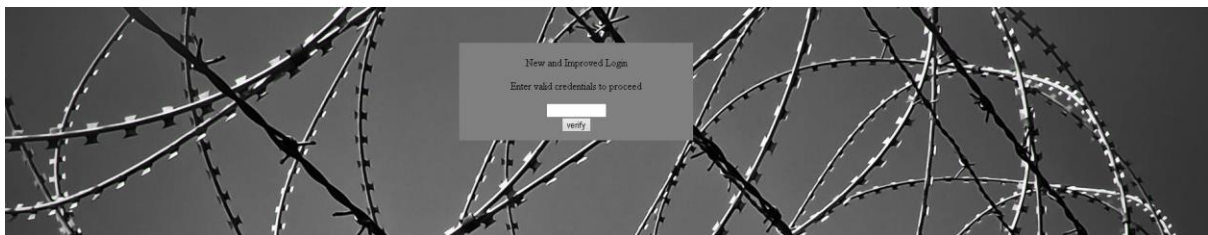
72% Liked



picoCTF{not\_this\_37115}

Submit Flag

Kedua website yang disediakan disol terlihat identic namun ada perbedaan pada bagian /problem/. Setelah dibuka pun keduanya memiliki tampilan yang sama.



Seperti biasa kita lihat inspect element dulu. Hint dari soal ini ada obfuscation, jadi di codenya pasti ada code yang sudah di obfuscate (biasanya di js). Pada inspect element kita menemukan adanya keberadaan javascript dalam bentuk `<script></script>`.

```
1 <html>
2 <head>
3 <title>Secure Login Portal V2.0</title>
4 </head>
5 <body background="barbed_wire.jpeg" >
6 <!-- standard MD5 implementation -->
7 <script type="text/javascript" src="md5.js"></script>
8
9 <script type="text/javascript">
10   var _0x5a46=['37115','_again_3','this','Password\x20Verify
11 </script>
12 <div style="position:relative; padding:5px;top:50px; left:38%;
13 <div style="text-align:center">
14 <p>New and Improved Login</p>
15
16 <p>Enter valid credentials to proceed</p>
17 <form action="index.html" method="post">
18 <input type="password" id="pass" size="8" />
19 <br/>
20 <input type="submit" value="verify" onclick="verify(); return
21 </form>
22 </div>
23 </div>
24 </body>
25 </html>
26
```

Kita coba untuk mengcopy dan melakukan beautify code agar javascriptnya bisa dibaca lebih mudah. (<https://codebeautify.org>)

```

Output
1 var _0x5a46 = ['37115'], '_again_3', 'this', 'PasswordVerified', 'Incorrect password', 'getElementById', 'value', 'substring', 'picoCTF', 'not_this'];
2 (function(_0x4bd822, _0x2bd6f7) {
3   var _0x4b5b = function(_0x1d68f6) {
4     while (!_0x1d68f6) {
5       _0x4bd822['push'](_0x4bd822['shift']());
6     }
7   };
8   _0x4b5b(++_0x2bd6f7);
9 }(_0x5a46, _0x1d68f6));
10 var _0x4b5b = function(_0x2d8f05, _0x4b81bb) {
11   _0x2d8f05 = _0x2d8f05 - 0x0;
12   var _0x4d74cb = _0x5a46[_0x2d8f05];
13   return _0x4d74cb;
14 };
15
16 function verify() {
17   checkpass = document[_0x4b5b('0x0')]['pass'][_0x4b5b('0x1')];
18   split = 0x0;
19   if (checkpass[_0x4b5b('0x2')](0x0, split * 0x2) == _0x4b5b('0x3')) {
20     if (checkpass[_0x4b5b('0x2')](0x7, 0x9) == '{n}') {
21       if (checkpass[_0x4b5b('0x2')](split * 0x2, split * 0x2 * 0x2) == _0x4b5b('0x4')) {
22         if (checkpass[_0x4b5b('0x2')](0x3, 0x6) == 'oCT') {
23           if (checkpass[_0x4b5b('0x2')](split * 0x3 * 0x2, split * 0x4 * 0x2) == _0x4b5b('0x5')) {
24             if (checkpass[_0x4b5b('0x2')](0x6, 0xb) == 'F{not}') {
25               if (checkpass[_0x4b5b('0x2')](split * 0x2 * 0x2, split * 0x3 * 0x2) == _0x4b5b('0x6')) {
26                 if (checkpass[_0x4b5b('0x2')](0xc, 0x10) == _0x4b5b('0x7')) {
27                   alert(_0x4b5b('0x8'));
28                 }
29             }
30         }
31     }
32 }
33 }
34 } else {
35   alert(_0x4b5b('0x9'));
36 }
37 }
38 }

```

Dari sini kita bisa mengetahui bahwa pada function verify terdapat `_0x4b5b('0x?')` yang obfuscated. Untuk dapat mengetahuinya, kita bisa menggunakan console yang ada pada inspect element.

```

> _0x4b5b
< f (_0x2d8f05, _0x4b81bb){_0x2d8f05=_0x2d8f05-0x0;var _0x4d74cb=_0x5a46[_0x2d8f05];return _0x4d74cb;}
> _0x4b5b("0x0")
Uncaught SyntaxError: missing ) after argument list
> _0x4b5b("0x0")
< 'getElementById'
> _0x4b5b("0x1")
< 'value'
> _0x4b5b("0x2")
< 'substring'
> _0x4b5b("0x3")
< 'picoCTF{'
> _0x4b5b("0x4")
< 'not_this'
> _0x4b5b("0x5")
< '37115'
> _0x4b5b("0x6")
< '_again_3'
> _0x4b5b("0x7")
< 'this'
> _0x4b5b("0x8")
< 'Password Verified'
> _0x4b5b("0x9")
< 'Incorrect password'

```

Nah dari sini kita sudah mengetahui masing-masing valuenya. Dan sekarang saatnya kita untuk mengubah value di codenya agar lebih mudah lagi untuk menyusun flagnya.

```

if (checkpass[substring])(0, split * 2) == picoCTF{
  if (checkpass[substring])(7, 9) == "{n}" {
    if (checkpass[substring])(split * 2, split * 2 * 2) == not_this) {
      if (checkpass[substring])(3, 6) == "oCT") {
        if (checkpass[substring])(split * 3 * 2, split * 4 * 2) == 37115) {
          if (checkpass["substring"](6, 11) == "F{not") {
            if (checkpass[substring])(split * 2 * 2, split * 3 * 2) == _again_3) {
              if (checkpass[substring])(12, 16) == this) {
                alert(Password Verified);
              }
            }
          }
        }
      }
    }
  }
}

```

Flag: `picoCTF{not_this_again_337115}`