

Home of CTF: <https://2023.knightctf.com/>

Team: 0bi0n3

Info

[+] Your team can submit a maximum of 3 incorrect flags per minute!

[+] Your team will have a maximum of 10 flag submission attempts per challenge.

Basic

Please join our Discord server and read the rules to get your flag.

```
01001011 01000011 01010100 01000110 01111011 01110111
00110011 01001100 01100011 00110000 01001101 00110011
01011111 01010100 00110000 01011111 01001011 01101110
01101001 01100111 01101000 01110100 01000011 01010100
01000110 01111101
```

Converted from Binary

```
KCTF{w3Lc0M3_T0_KnightCTF}
```

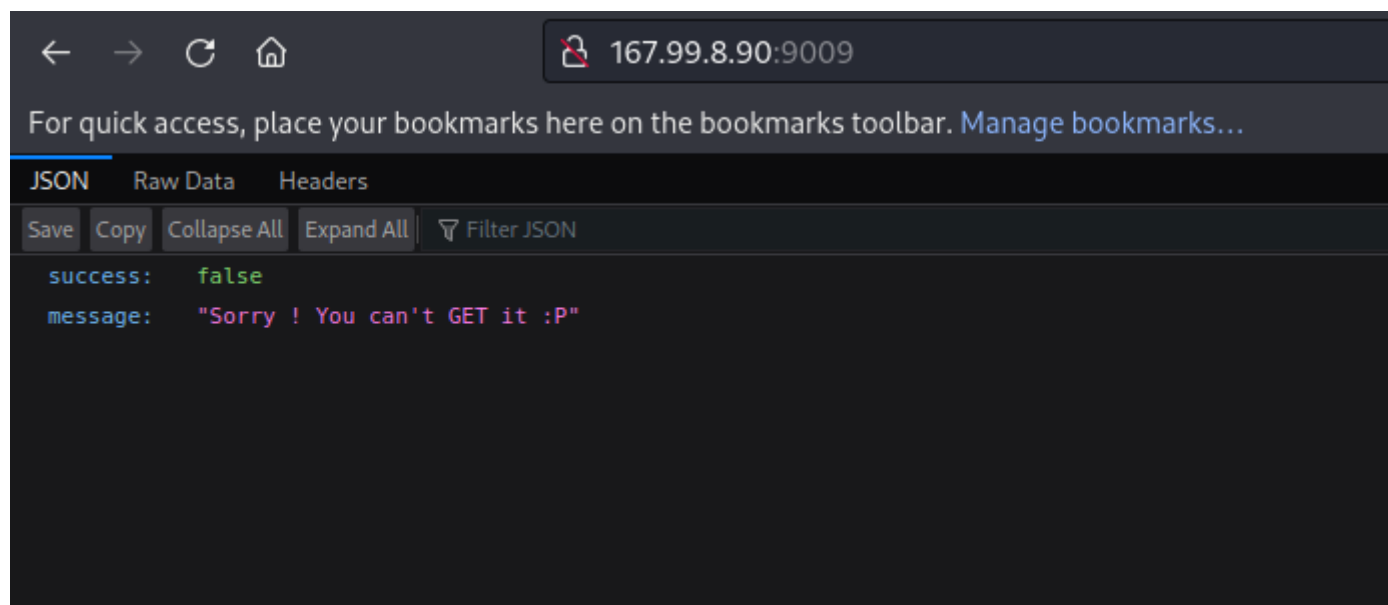
Web/API

#API

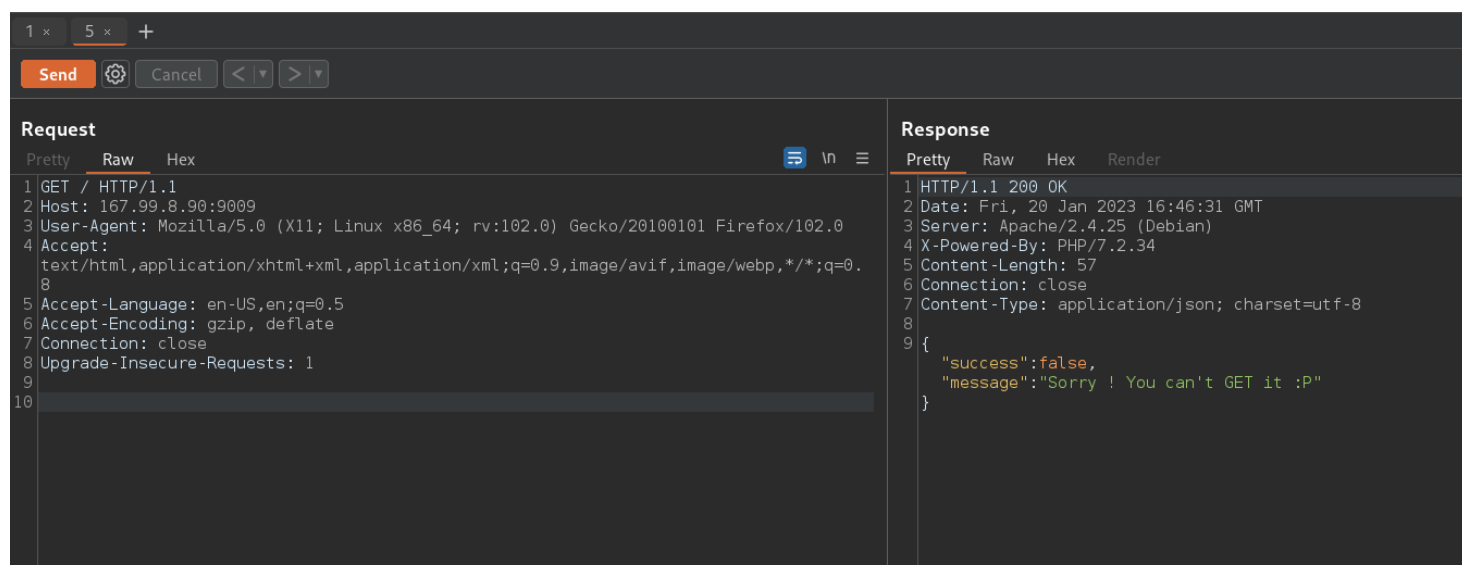
Can you GET the flag from the API ?

```
Link provided: http://167.99.8.90:9009/
```

We would go to the site and get this.



We took it to Repeater in Burp to play around



I tried to changed

OSINT

An orchestra was held where this product was mined. What is the name of the organization the conductor of the orchestra leads as a president?

Demo Flag: KCTF{Flag_HeRe}

*File Provided



Info: <http://philharmonic.lg.ua/en/>

Crypto

Factorie

Have you ever heard of prime factors? The file `challenge.txt` contains a number that has two prime factors. Can you find them?

n: 2174096211032823084932239036566496093206280423

Demo Flag: KCTF{small-number_big-number}

Resource:

<https://www.thecalculator.co/math/Factoring-Calculator-39.html>

Resource: <https://www.random-science-tools.com/maths/prime-factors.htm>

KCTF{2-19}

KCTF{2-491}

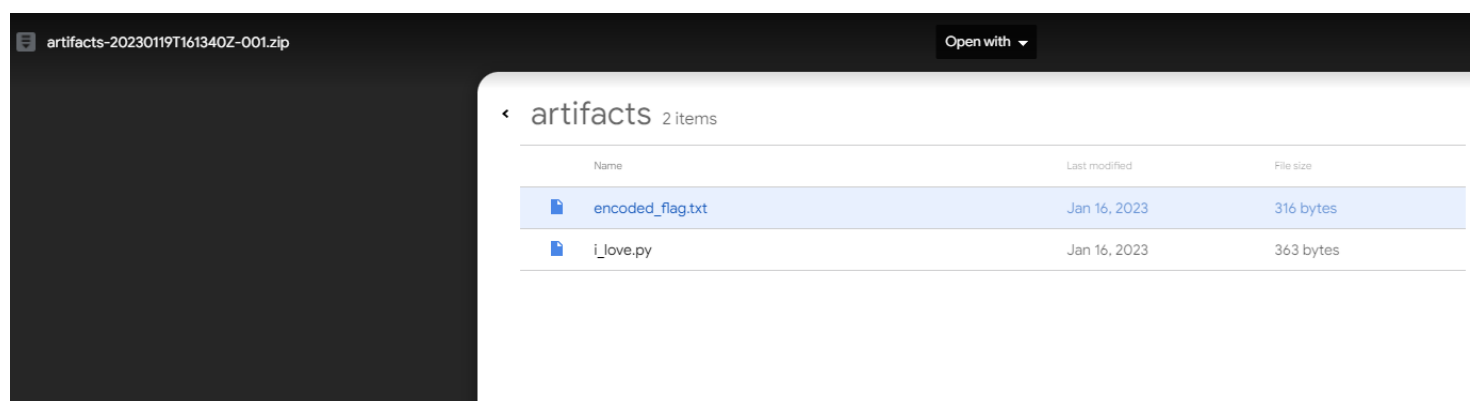
```
KCTF{2-19}  
KCTF{2-1879}  
KCTF{7-23}  
KCTF{7-37875710256959}  
KCTF{2-3}  
KCTF{2-23}  
KCTF{2-37875710256959}
```

I Love Pi

Isaac Newton left me this piece of code and a message. Can you help me decode this...

POC

We get some files and they look to be dealing in python.



I take it over to Geany to see what the py file does

```

import base64

lengths = [--REDACTED--]
flag = "KCTF{*****}"

# len(lengths) = 10
# len(flag) = 39

s = 0
encoded_flag = ""
for l in lengths:
    seg = flag[s:s+l]
    for _ in range(len(seg)):
        seg = base64.b64encode(seg.encode('ascii')).decode('ascii')
    s+=l
    encoded_flag += seg

print(encoded_flag)

```

I can see that it take a file and it decodes the base64 to ASCII, looks to be encoded in a way. Lets run it and see what it says

```

(kali㉿kali)-[~/Desktop/artifacts]
$ python ./i_love.py encoded_flag.txt
File "/home/kali/Desktop/artifacts/./i_love.py", line 3
    lengths = [--REDACTED--]
              ^
SyntaxError: invalid syntax

(kali㉿kali)-[~/Desktop/artifacts]
$

```

From the output there is an issue with line 3. Lets look again.

```

1  import base64
2
3  lengths = [--REDACTED--]
4  flag = "KCTF{*****}"
5
6  # len(lengths) = 10
7  # len(flag) = 39

```

From what I can tell there is variables that are commented out. In the lengths variable I changed it from Redacted to 10.

i_love.py x

```
1 import base64
2
3 lengths = [10]
4 flag = "KCTF{*****}"
5
6 # len(lengths) = 10
7 # len(flag) = 39
```

Lets run it again

```
(root@kali)-[/home/kali/Desktop/artifacts]
# python ./i_love.py encoded_flag.txt
Vm0wd2QyVkZOVWhUV0d4V1YwZG9WRl13Wkc5V01WbDNXa2M1V0ZadGVibFhhMXBQVmpGYWRHVkVRbUZxVjFKSVdWZDRZV014VG50W6JGcFhaV3RhU1Z
adGVHRlpWMMUpJVm10a2FGSnRhRmxWTUZZaTFYxWmtXR1JIUmXSTmF6VjVWRlpVjFZeVNrAGhSbXhXVFVaYVRGUnRlR0ZqTVdSMFVteGtUbFp1UWxoV1
JscFhWakpHU0ZadVJScSldSM001

(root@kali)-[/home/kali/Desktop/artifacts]
# echo "Vm0wd2QyVkZOVWhUV0d4V1YwZG9WRl13Wkc5V01WbDNXa2M1V0ZadGVibFhhMXBQVmpGYWRHVkVRbUZxVjFKSVdWZDRZV014VG50W6JGc
FhaV3RhU1ZadGVHRlpWMMUpJVm10a2FGSnRhRmxWTUZZaTFYxWmtXR1JIUmXSTmF6VjVWRlpVjFZeVNrAGhSbXhXVFVaYVRGUnRlR0ZqTVdSMFVteGtU
bFp1UWxoV1JscFhWakpHU0ZadVJScSldSM001" | base64 -d
Vm0wd2VFNUhTWGxWV0doVFYwZG9WMVl3Wkc5WFZteHlXa1pPVjFadGVEQmFWV1JlWVd4YWMxTnNXbFpXZWtaSVZteGFZV1JlVmtkaFJtaFLVMFZLV1Z
kWGRHRlRNazV5VFZaV1YySkhhRmxWTUZZaTFRteGFjMWR0UmXkTLZuQlhWRlpXVjJGSFZuRlJWR3M5

(root@kali)-[/home/kali/Desktop/artifacts]
# echo "Vm0wd2VFNUhTWGxWV0doVFYwZG9WMVl3Wkc5WFZteHlXa1pPVjFadGVEQmFWV1JlWVd4YWMxTnNXbFpXZWtaSVZteGFZV1JlVmtkaFJta
FLVMFZLV1ZkWGRHRlRNazV5VFZaV1YySkhhRmxWTUZZaTFRteGFjMWR0UmXkTLZuQlhWRlpXVjJGSFZuRlJWR3M5" | base64 -d
Vm0weE5HSXlVWGhTV0doV1YwZG9XVmxYkZOV1ZteDBaVWRHYWxac1NsWlZWekZlVmxayWRHVkdhRmhyU0VKVWdXGdGFTmk5yTVZWV2JHaFLVMFZLTmx
ac1dtRldNVnBXVFZWV2FHVnFRVGs9

(root@kali)-[/home/kali/Desktop/artifacts]
# echo "Vm0weE5HSXlVWGhTV0doV1YwZG9XVmxYkZOV1ZteDBaVWRHYWxac1NsWlZWekZlVmxayWRHVkdhRmhyU0VKVWdXGdGFTmk5yTVZWV2JHa
FLVMFZLTmxac1dtRldNVnBXVFZWV2FHVnFRVGs9" | base64 -d
Vm0xNGIyUXhSWGhWV0doVWlrZFNWVmx0ZUdGalZsSlZVVzFHVlZadGVGaFhXSEJYWwtaS2NrMVVWbGhYU0VKNLZsWmFWMVpWTVVWaGVqQTk=

(root@kali)-[/home/kali/Desktop/artifacts]
# echo "Vm0xNGIyUXhSWGhWV0doVWlrZFNWVmx0ZUdGalZsSlZVVzFHVlZadGVGaFhXSEJYWwtaS2NrMVVWbGhYU0VKNLZsWmFWMVpWTVVWaGVqQ
Tk=" | base64 -d
Vm14b2QxRXhVWGhYYkdSVVlteGFjVlJVUW1GVVZteFhXWHBXYkZKck1UVlhXSEJ6VlZaV1ZVMUVhejA9

(root@kali)-[/home/kali/Desktop/artifacts]
# echo "Vm14b2QxRXhVWGhYYkdSVVlteGFjVlJVUW1GVVZteFhXWHBXYkZKck1UVlhXSEJ6VlZaV1ZVMUVhejA9" | base64 -d
Vmxod1ExUXhXbGRUYmxacVRUQmFUVmxXWpWbFJrMTVXWHBzVVZWVU1Eaz0=

(root@kali)-[/home/kali/Desktop/artifacts]
# echo "Vmxod1ExUXhXbGRUYmxacVRUQmFUVmxXWpWbFJrMTVXWHBzVVZWVU1Eaz0=" | base64 -d
VlhWQ1QxWldTb1ZqTTBaTVlWYzVlRk15WXpsUVVUMDk=

(root@kali)-[/home/kali/Desktop/artifacts]
# echo "VlhWQ1QxWldTb1ZqTTBaTVlWYzVlRk15WXpsUVVUMDk=" | base64 -d
VXpCT1ZWSnVjM0ZMYVc5eFMyYz1QUT09

(root@kali)-[/home/kali/Desktop/artifacts]
# echo "VXpCT1ZWSnVjM0ZMYVc5eFMyYz1QUT09" | base64 -d
UzBOVVJuc3FLaW9xS2c9PQ=

(root@kali)-[/home/kali/Desktop/artifacts]
# echo "UzBOVVJuc3FLaW9xS2c9PQ=" | base64 -d
S0NURnsqKioqKg=

(root@kali)-[/home/kali/Desktop/artifacts]
# echo "S0NURnsqKioqKg=" | base64 -d
KCTF{*****

(root@kali)-[/home/kali/Desktop/artifacts]
#
```

Hmmm. I hit a wall. Let go back and look at the code.

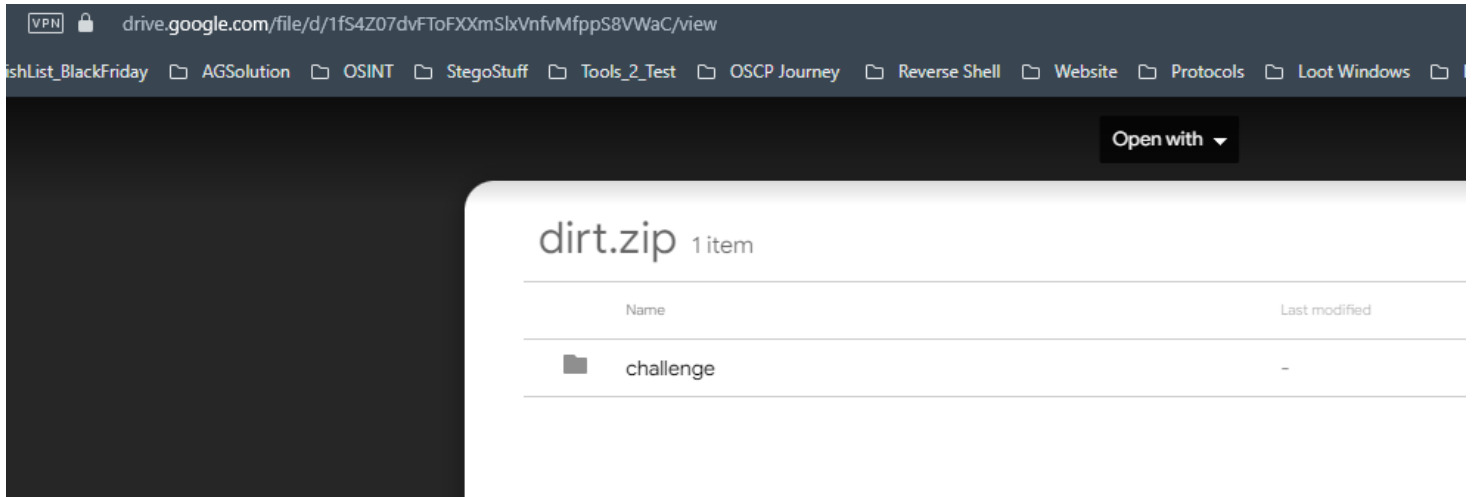
Misc

Dirt

My friend loves to travel. Can you help him get the flag?

POC

We got a zip file



Then I took it to call to see what is inside of the file

```
tree challenge
```

```
(kali㉿kali)-[~/Desktop]
└─$ tree challenge
challenge
├── }
│   ├── s
│   │   ├── r
│   │   │   ├── 3
│   │   │   │   ├── d
│   │   │   │   │   ├── l
│   │   │   │   │   │   ├── 0
│   │   │   │   │   │   │   ├── f
│   │   │   │   │   │   │   │   ├── 3
│   │   │   │   │   │   │   │   │   ├── d
│   │   │   │   │   │   │   │   │   ├── 1
│   │   │   │   │   │   │   │   │   │   ├── 5
│   │   │   │   │   │   │   │   │   │   │   ├── n
│   │   │   │   │   │   │   │   │   │   │   │   ├── 1
│   │   │   │   │   │   │   │   │   │   │   │   │   ├── s
│   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── r
│   │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── 3
│   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── d
│   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── l
│   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── 0
│   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── f
│   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── {
│   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── F
│   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── T
│   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── c
│   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── K
29 directories, 0 files

(kali㉿kali)-[~/Desktop]
└─$
```

The flag is backwards.

Flag

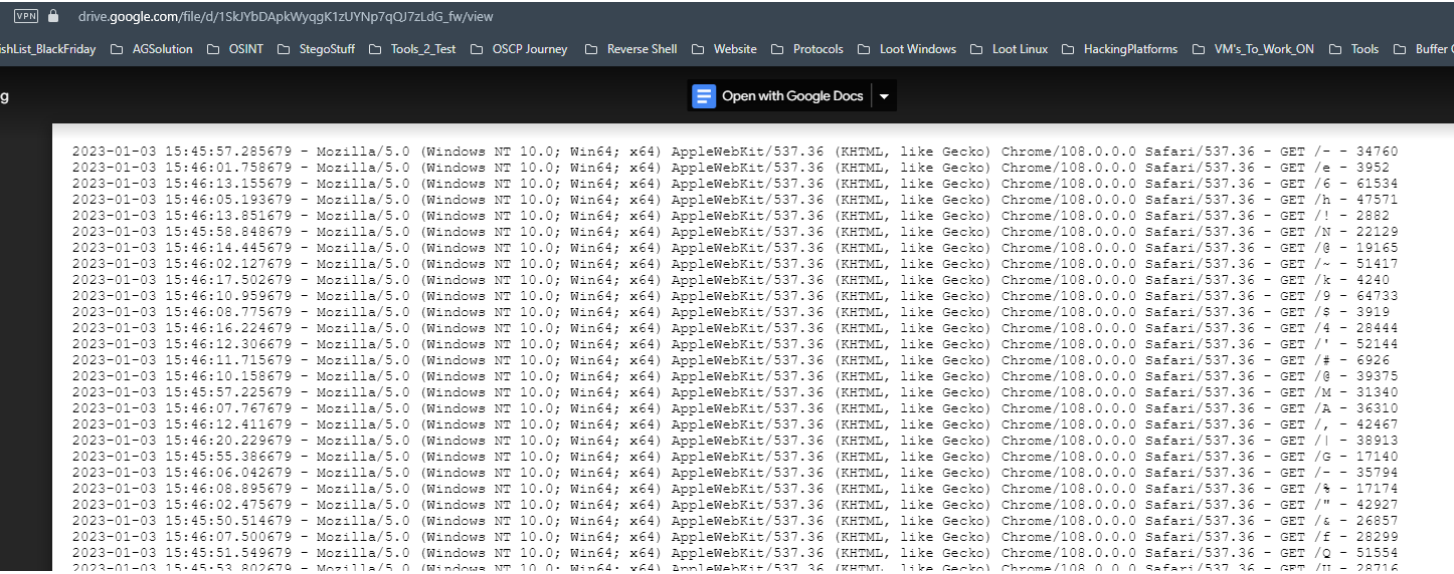
```
KCTF{f0ld3rs_1n51d3_f0ld3rs}
```

Logger

Sysadmin found an access log that contains some requests to weird endpoints. It's said that the flag lies among them. Are you smart enough to find it?

POC

We check out the file that is downloadable



From what we can see and the info about the challenge we have a logg file to sort through. Let take it to Kali and see if we can put it together.

*First thing I want to do is sort it by

Flag

Network

Sir vignere came to my dreams and sent me this packet capture and told me to find the flag from it which is the key to my success. I am a noob in these cases. So I need your help. Please help me find the flag. Will you?

POC

We have a pcap file. I want to see what protocols are being used the most.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDU's
Frame	100.0	85	100.0	9728	10 k	0	0	0	85
Ethernet	100.0	85	12.2	1190	1235	0	0	0	85
Internet Protocol Version 4	100.0	85	17.5	1700	1765	0	0	0	85
User Datagram Protocol	84.7	72	5.9	576	598	0	0	0	72
Domain Name System	84.7	72	47.0	4572	4747	72	4572	4747	72
Internet Control Message Protocol	15.3	13	17.4	1690	1754	0	0	0	13
Domain Name System	15.3	13	12.6	1222	1268	13	1222	1268	13

I see that DNS and UDP are the most used protocols here. Lets see what is working.

find-me.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

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

No.	Time	Source	Destination	Protocol	Length	Info
43	7.324893232	10.0.2.15	192.168.0.1	DNS	73	Standard query 0x1ccf A knightctf.com
44	8.572662969	192.168.0.1	10.0.2.15	DNS	425	Standard query response 0x1ccf A knightctf.com A 104.21.72.32 A 172.67.174.103 NS keaton.ns.cloudflare.com NS
45	8.597157315	10.0.2.15	104.21.72.32	ICMP	43	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 46)
46	8.676532963	104.21.72.32	10.0.2.15	ICMP	60	Echo (ping) reply id=0x0000, seq=0/0, ttl=54 (request in 45)
47	8.708385127	10.0.2.15	192.168.0.1	DNS	73	Standard query 0x9d86 A knightctf.com
48	8.803868148	192.168.0.1	10.0.2.15	DNS	425	Standard query response 0x9d86 A knightctf.com A 104.21.72.32 A 172.67.174.103 NS keaton.ns.cloudflare.com NS
49	8.905394035	10.0.2.15	104.21.72.32	ICMP	43	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 50)
50	8.920119450	104.21.72.32	10.0.2.15	ICMP	60	Echo (ping) reply id=0x0000, seq=0/0, ttl=54 (request in 49)
51	8.958090762	10.0.2.15	8.8.8.8	DNS	75	Standard query 0x0000 A V.knightctf.com
52	9.185207545	8.8.8.8	10.0.2.15	DNS	136	Standard query response 0x0000 No such name A V.knightctf.com SOA brenda.ns.cloudflare.com
53	9.185399127	10.0.2.15	8.8.8.8	ICMP	164	Destination unreachable (Port unreachable)
54	9.226257970	10.0.2.15	8.8.8.8	DNS	75	Standard query 0x0000 A V.knightctf.com
55	9.390731665	8.8.8.8	10.0.2.15	DNS	136	Standard query response 0x0000 No such name A V.knightctf.com SOA brenda.ns.cloudflare.com
56	9.390792732	10.0.2.15	8.8.8.8	ICMP	164	Destination unreachable (Port unreachable)
57	9.440792352	10.0.2.15	8.8.8.8	DNS	75	Standard query 0x0000 A B.knightctf.com
58	9.697815962	8.8.8.8	10.0.2.15	DNS	136	Standard query response 0x0000 No such name A B.knightctf.com SOA brenda.ns.cloudflare.com

> Frame 53: 164 bytes on wire (1312 bits), 164 bytes captured (1312 bits) on interface eth0, id 0

> Ethernet II, Src: PcsCompu_22:46:4f (08:00:27:22:46:4f), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)

> Internet Protocol Version 4, Src: 10.0.2.15, Dst: 8.8.8.8

> Internet Control Message Protocol

Type: 3 (Destination unreachable)

Code: 3 (Port unreachable)

Checksum: 0x1993 [correct]

[Checksum Status: Good]

Unused: 00000000

> Internet Protocol Version 4, Src: 8.8.8.8, Dst: 10.0.2.15

> User Datagram Protocol, Src Port: 53, Dst Port: 53

0000 52 54 00 12 35 02 08 00 27 22 46 4f 08 00 45 c0 RT...S...''FO...E

0010 00 96 b4 95 00 00 40 01 a8 f3 0a 00 02 0f 08 08 @.....@.....E...z...

0020 08 08 03 03 19 93 00 00 00 00 45 00 00 7a 01 d4 @.....@.....E...z...

0030 00 00 40 11 5c 81 08 08 08 0a 00 02 0f 00 35 @.....@.....E...z...

0040 00 35 00 66 c5 ff 00 00 81 83 00 01 00 00 00 01 5 f.....V.knightctf.c

0050 00 00 01 56 09 6b 6e 69 67 68 74 63 74 66 03 63 om.....om.....1-bren da ns cl

0060 6f 6d 00 00 01 00 01 c0 0e 00 06 00 01 00 00 06 oudflare ...dns-7

0070 e1 00 31 06 62 72 65 6e 64 61 02 6e 73 0a 63 6c ...7... ..

0080 6f 75 64 66 6c 61 72 65 c0 18 03 64 6e 73 c0 37

0090 88 e7 89 37 00 00 27 10 00 00 09 60 00 09 3a 80

00a0 00 00 0e 10

What I am interested is what is .15 sending to the DNS server 8.8.8.8, We can follow the UPD stream

Wireshark · Follow UDP Stream (udp.stream eq 12) · find-me.pcapng




.....V knightctf.com.....V knightctf.com.....1.brenda.ns		
cloudflare...dns.7..7..V knightctf.com.....V knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..B knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..C knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..T knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..H knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..t knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..v knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..M knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..V knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..9 knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..t knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..c knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..j knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..N knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..h knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..X knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..2 knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..V knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..u knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..M knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..F knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..9 knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..o knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..a knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..z knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..N knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..f knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..a knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..T knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..B knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..o knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..f knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..Q knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..= knightctf.com.....1.brenda.ns	
cloudflare...dns.7..7..= knightctf.com.....1.brenda.ns	

Looking at the format it looks like its bas64

VVBCTHtvMV9tcjNhX2VuMF9oazNfaTBofQ==

Last build: A month ago

Recipe



From Base64

Alphabet

A-Za-z0-9+/=

☒ Remove non-alphabet chars

☐ Strict mode

Input

VVBCTHtvMV9tcjNhX2VuMF9oazNfaTBofQ==

Output

UPBL{o1_mr3a_en0_hk3_i0h}

We get some what a decoded output. We got a hint that this might be a vignere cipher

Reverse Engineer

KrackME

Find the right flag from the binary.

Demo Flag: KCTF{FL4g_H3r3}

POC

We download a file from the site and move it over to kali and check what it is.

```
(kali㉿kali)-[~/Desktop]
└─$ file krackme_1.out
krackme_1.out: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=d2bfb9e6b0a06b19109c3dd4b5367d8e7b2a3c00, for GNU/Linux 4.4.0, with debug_info, not stripped

(kali㉿kali)-[~/Desktop]
└─$ ./krackme_1.out
=====
|               KnightCTF 2023               |
|   Organized by Knight Squad   |
|=====
kali , Welcome To KrackMe 1.0...

You don't have access to KrackMe 1.0 !
Since you are here let me ask you something...
Did you know, Bangladesh has the longest natural beach?...

(kali㉿kali)-[~/Desktop]
└─$
```