

Packet_Sniffing

bi0s{w1r35h4rk_exp0rts_1s_c00l}

08 July 2023 12:30

Challenge description

[Challenges / Network Security /](#)

Packet_Sniffing

1000 Points

Solved

Submit Flag

bi0s{...}

🚩

Download Attachments

Files/URLs required for the challenge

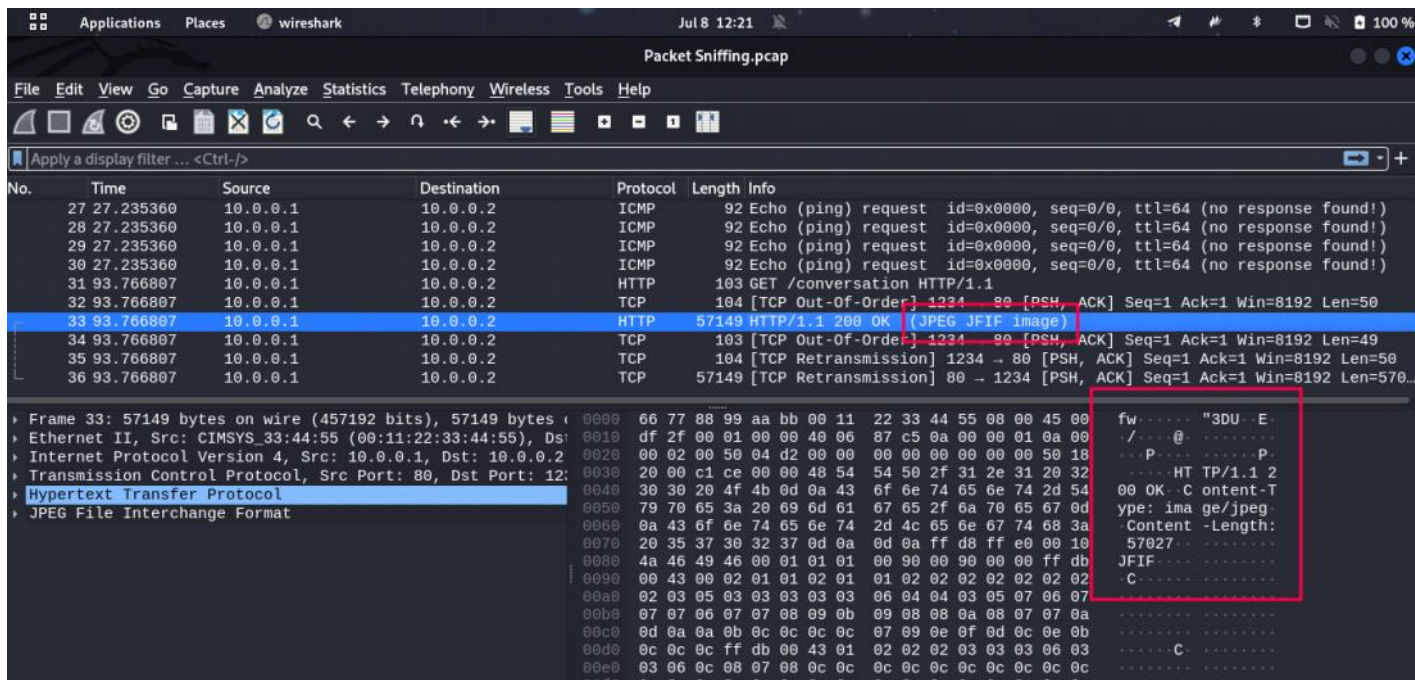
DESCRIPTION

Welcome to the world of packet capture analysis. This is where you go through the captured packets trying to find data which will help you analyse the packet capture. Packet capture analysis is the process of examining network traffic to gain insights into what is happening on a network.

You have been presented with a packet capture, which is essentially a record of network traffic containing packets exchanged. Your task is to analyze these packets meticulously, searching for critical information that can provide insights into the scenario at hand.

FLAG FORMAT: bi0s{...}

Download the given attachments.
And open it in wireshark



Go to File --> Export Objects --> HTTP
You get an image

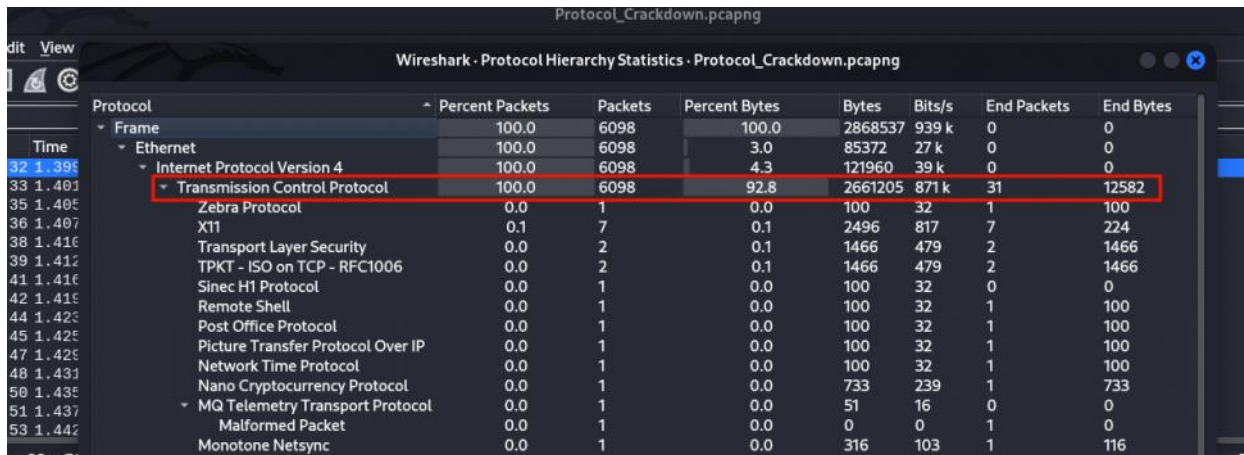
bi0s{w1r35h4rk_exp0rts_1s_c00l}

Protocol_Crackdown

bi0s{lt_w4snt_th4t_h4rd_for_y0u_ch4mp}

08 July 2023 12:31

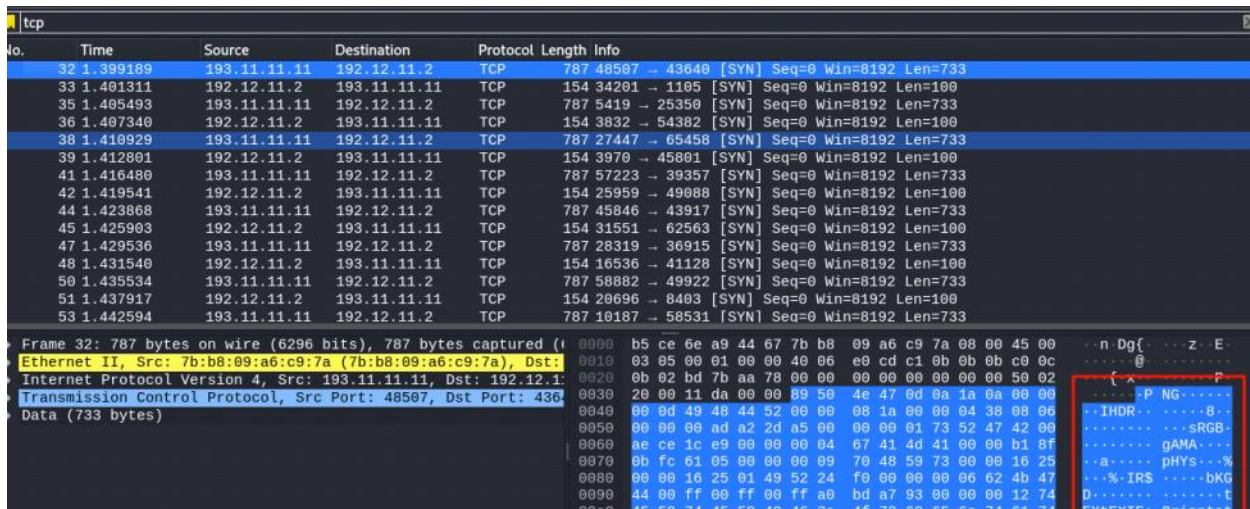
Download the given files and analyse the protocol hierarchy by
Statistics -> Protocol Hierarchy



The screenshot shows the Wireshark Protocol Hierarchy Statistics window. The 'Transmission Control Protocol' is highlighted with a red box, showing it accounts for 92.8% of the bytes and 31 of the end packets.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes
Frame	100.0	6098	100.0	2868537	939 k	0	0
Ethernet	100.0	6098	3.0	85372	27 k	0	0
Internet Protocol Version 4	100.0	6098	4.3	121960	39 k	0	0
Transmission Control Protocol	100.0	6098	92.8	2661205	871 k	31	12582
Zebra Protocol	0.0	1	0.0	100	32	1	100
X11	0.1	7	0.1	2496	817	7	224
Transport Layer Security	0.0	2	0.1	1466	479	2	1466
TPKT - ISO on TCP - RFC1006	0.0	2	0.1	1466	479	2	1466
Sinec H1 Protocol	0.0	1	0.0	100	32	0	0
Remote Shell	0.0	1	0.0	100	32	1	100
Post Office Protocol	0.0	1	0.0	100	32	1	100
Picture Transfer Protocol Over IP	0.0	1	0.0	100	32	1	100
Network Time Protocol	0.0	1	0.0	100	32	1	100
Nano Cryptocurrency Protocol	0.0	1	0.0	733	239	1	733
MQ Telemetry Transport Protocol	0.0	1	0.0	51	16	0	0
Malformed Packet	0.0	1	0.0	0	0	1	0
Monotone Netsync	0.0	1	0.0	316	103	1	116

Let us look at the TCP packets



The screenshot shows the Wireshark packet list and details for a TCP packet. The packet details pane shows the 'Transmission Control Protocol' section, which is highlighted with a red box. The packet is from 193.11.11.11 to 192.12.11.2, port 48507 to 4360.

No.	Time	Source	Destination	Protocol	Length	Info
32	1.399189	193.11.11.11	192.12.11.2	TCP	787	48507 -> 4360 [SYN] Seq=0 Win=8192 Len=733
33	1.401311	192.12.11.2	193.11.11.11	TCP	154	34201 -> 1105 [SYN] Seq=0 Win=8192 Len=100
35	1.405493	193.11.11.11	192.12.11.2	TCP	787	5419 -> 25350 [SYN] Seq=0 Win=8192 Len=733
36	1.407340	192.12.11.2	193.11.11.11	TCP	154	3832 -> 54382 [SYN] Seq=0 Win=8192 Len=100
38	1.410929	193.11.11.11	192.12.11.2	TCP	787	27447 -> 65458 [SYN] Seq=0 Win=8192 Len=733
39	1.412801	192.12.11.2	193.11.11.11	TCP	154	3970 -> 45801 [SYN] Seq=0 Win=8192 Len=100
41	1.416480	193.11.11.11	192.12.11.2	TCP	787	57223 -> 39357 [SYN] Seq=0 Win=8192 Len=733
42	1.419541	192.12.11.2	193.11.11.11	TCP	154	25959 -> 49088 [SYN] Seq=0 Win=8192 Len=100
44	1.423868	193.11.11.11	192.12.11.2	TCP	787	45846 -> 43917 [SYN] Seq=0 Win=8192 Len=733
45	1.425903	192.12.11.2	193.11.11.11	TCP	154	31551 -> 62563 [SYN] Seq=0 Win=8192 Len=100
47	1.429536	193.11.11.11	192.12.11.2	TCP	787	28319 -> 36915 [SYN] Seq=0 Win=8192 Len=733
48	1.431540	192.12.11.2	193.11.11.11	TCP	154	16536 -> 41128 [SYN] Seq=0 Win=8192 Len=100
50	1.435534	193.11.11.11	192.12.11.2	TCP	787	58882 -> 49922 [SYN] Seq=0 Win=8192 Len=733
51	1.437917	192.12.11.2	193.11.11.11	TCP	154	20696 -> 8403 [SYN] Seq=0 Win=8192 Len=100
53	1.442594	193.11.11.11	192.12.11.2	TCP	787	10187 -> 58531 [SYN] Seq=0 Win=8192 Len=733

Frame 32: 787 bytes on wire (6296 bits), 787 bytes captured (6296 bits) on interface 0
Ethernet II, Src: 7b:b8:09:a6:c9:7a (7b:b8:09:a6:c9:7a), Dst: 02:00:00:00:00:00
Internet Protocol Version 4, Src: 193.11.11.11, Dst: 192.12.11.2
Transmission Control Protocol, Src Port: 48507, Dst Port: 4360
Data (733 bytes)

This hints towards a .png file in the packets

Using this code we can extract the .png file

```
from scapy.all import *
```

```
packets = rdpcap('Protocol_Crackdown.pcapng')
```

```
png_data = b''
```

```
for packet in packets:  
    if IP in packet and packet[IP].src == '193.11.11.11' and TCP in packet:  
        png_data += bytes(packet[TCP].payload)
```

```
with open('solution.png', 'wb') as image_file:  
    image_file.write(png_data)
```



Protocol_Crackdown.pcapng



solution.png



solution.py

```
mactavish@oracle: ~/Documents/CTF files and writeups/C...  
└─(mactavish@oracle)~[~/Documents/CTF files and writeups/Cyseck CTF/protocol c  
rackdown]  
└─$ python3 solution.py  
  
└─(mactavish@oracle)~[~/Documents/CTF files and writeups/Cyseck CTF/protocol c  
rackdown]  
└─$
```



23% ↓



solution.png



Digital_Vault

bi0s{sc4py_scr1pt1ng_m4k3s_extr4cti0n_3a513rx0}

08 July 2023 15:58

```
from scapy.all import *

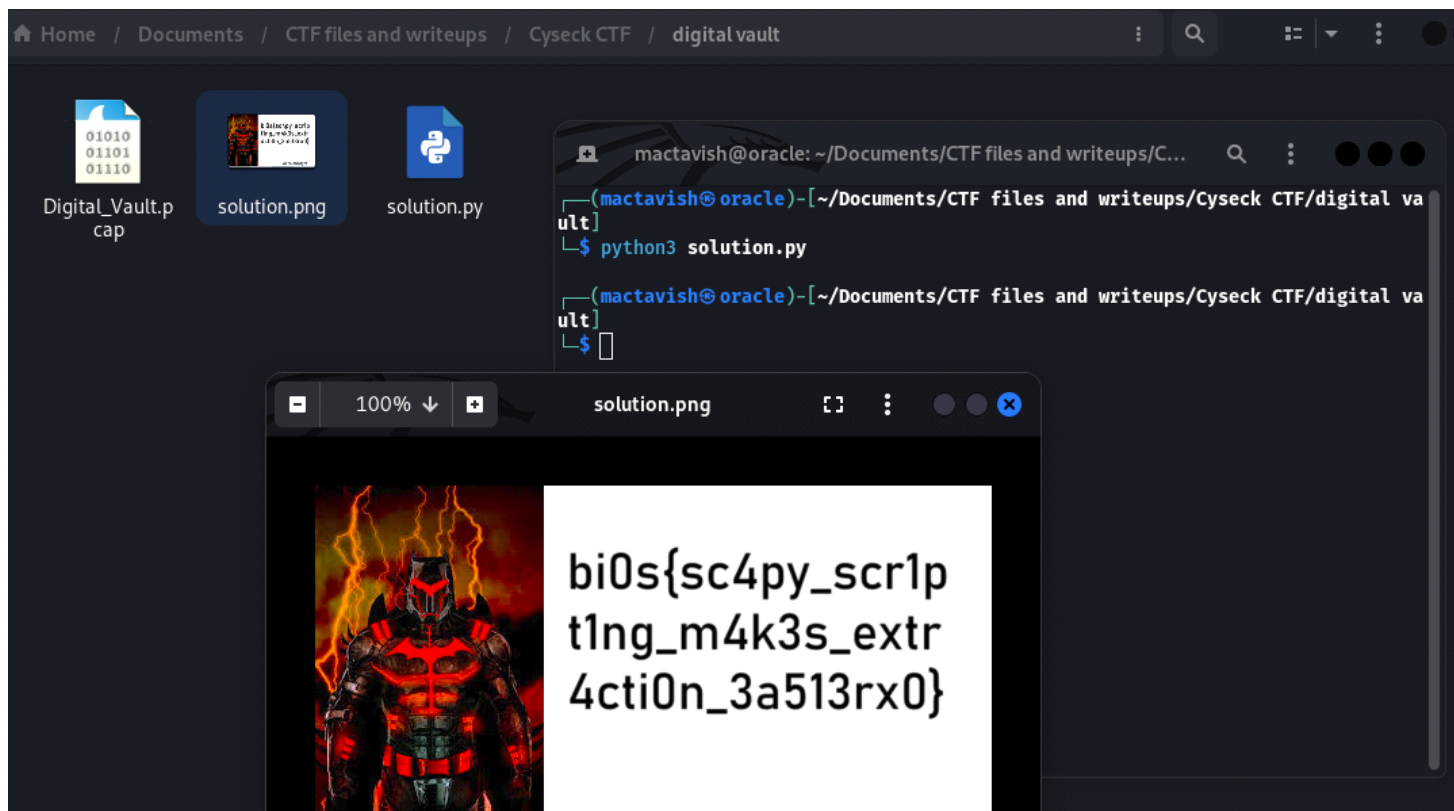
packets = rdpcap(r'Digital_Vault.pcap')

png_data = b''

counter = 0

for packet in packets[6:]:
    if TCP in packet:
        png_data += bytes(packet[TCP].payload)
        counter += 1

with open(r'solution.png', 'wb') as image_file:
    image_file.write(png_data)
```



Apply a display filter ... <Ctrl>F

Time	Source	Destination	Protocol	Length	Info
136 0.278303	192.12.11.2	193.11.11.11	ICMP	43	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response found!)
137 0.279876	192.12.11.2	193.11.11.11	ICMP	43	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response found!)
138 0.282174	192.12.11.2	193.11.11.11	ICMP	43	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response found!)
139 0.283693	192.12.11.2	193.11.11.11	ICMP	43	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response found!)
140 0.285081	192.12.11.2	193.11.11.11	ICMP	43	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response found!)
141 0.287433	192.12.11.2	193.11.11.11	ICMP	43	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response found!)
142 0.289262	192.12.11.2	193.11.11.11	ICMP	43	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response found!)
143 0.290675	192.12.11.2	193.11.11.11	ICMP	43	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response found!)
144 0.292334	192.12.11.2	193.11.11.11	ICMP	43	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response found!)
145 0.294426	192.12.11.2	193.11.11.11	ICMP	43	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response found!)
146 0.295972	192.12.11.2	193.11.11.11	ICMP	43	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response found!)

Frame 140: 43 bytes on wire (344 bits), 43 bytes captured

Ethernet II, Src: 13:e8:b9:c1:0d:9d (13:e8:b9:c1:0d:9d),

Internet Protocol Version 4, Src: 192.12.11.2, Dst: 193.11.11.11

Internet Control Message Protocol

0000 15 fc 8f 2f 14 09 13 e8 b9 c1 0d 9d 08 00 45 00 .../... ..E

0010 00 1d 00 01 00 00 40 01 e3 ba c0 0c 0b 02 c1 0b ...@... ..

0020 0b 0b 08 00 7c ff 00 00 00 00 7b ...|... {

Solve this until you get the full flag

Decrypt_The_Secrets

bi0s{n3tw0rk_interception_g0es_b00mx0x0}

08 July 2023 15:59

DecryptTheSecrets.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

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

No.	Time	Source	Destination	Protocol	Length	Info
5	0.035791	192.12.11.2	193.11.11.11	TCP	101	63545 → 54545 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=47
6	0.038471	193.11.11.11	192.12.11.2	TCP	122	44146 → 28861 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=68
7	0.041558	192.12.11.2	193.11.11.11	TCP	116	15751 → 23325 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=62
8	0.046907	193.11.11.11	192.12.11.2	TCP	95	57406 → 24570 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=41
9	0.050131	192.12.11.2	193.11.11.11	TCP	124	41506 → 25931 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=70
10	0.053202	193.11.11.11	192.12.11.2	TCP	115	28583 → 16096 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=61
11	0.065107	192.12.11.2	193.11.11.11	TCP	126	1361 → 35936 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=72
12	0.070756	193.11.11.11	192.12.11.2	TCP	129	39430 → 52890 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=75
13	0.077268	192.12.11.2	193.11.11.11	TCP	83	32664 → 43251 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=29
14	0.082224	193.11.11.11	192.12.11.2	TCP	86	49427 → 13192 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=32
15	0.088663	192.12.11.2	193.11.11.11	TCP	86	58999 → 18439 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=32
16	0.099039	193.11.11.11	192.12.11.2	TCP	105	584 → 6183 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=51
17	0.113222	192.12.11.2	193.11.11.11	TCP	69	24959 → 37685 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=15

Frame 16: 105 bytes on wire (840 bits), 105 bytes capture
Ethernet II, Src: 6a:5a:9e:1a:7f:d6 (6a:5a:9e:1a:7f:d6),
Internet Protocol Version 4, Src: 193.11.11.11, Dst: 192.
Transmission Control Protocol, Src Port: 584, Dst Port: 6
Data (51 bytes)

0000 ef 45 1f a4 75 3e 6a 5a 9e 1a 7f d6 08 00 45 00 E..u>jZE
0010 00 5b 00 01 00 00 40 06 e3 77 c1 0b 0b 0b c0 0c [....@..w.....
0020 0b 02 02 48 18 27 00 00 00 00 00 00 00 50 18 ..H.'.....P..
0030 20 00 24 5f 00 00 6d 74 62 20 66 67 74 7a 79 20 \$.mt b fgtzy
0040 67 0e 30 78 7b 73 33 79 62 30 77 70 5f 6e 73 79 gn0x{\$3y b0wp_nsy
0050 6a 77 68 6a 75 79 6e 74 73 5f 6c 30 6a 78 5f 67 jwhjuynt s_l0jx_g
0060 30 30 72 63 30 63 30 7d 3f 00rc0c0} ?

dCODE

Search for a tool

★ SEARCH A TOOL ON dCODE BY KEYWORDS:
e.g. type 'boolean'

★ BROWSE THE FULL dCODE TOOLS' LIST

Results

Brute-Force mode: the 25 shifts (for the alphabet ABCDEFGHIJKLMNOPQRSTUVWXYZ) are tested and sorted from most probable to least probable.

↑↓	↑↓
→5 (←21)	bi0s{n3tw0rk_interception_g0es_b00mx0x0}?
→9 (←17)	xe0o{J3ps0ng_ejpanyalpekj_c0ao_x0oit0t0}?
→15 (←11)	ry0i{d3jm0ha_ydjuhsufjyed_w0ui_r00cn0n0}?
→18 (←8)	ov0f{a3gj0ex_vagreprcgvba_t0rf_o00zk0k0}?
→24 (←2)	ip0z{u3ad0yr_pualyj_lwapvu_n0l_z_i00te0e0}?

CAESAR CIPHER
Cryptography · Substitution Cipher · Caesar Cipher

CAESAR CIPHER DECODER

★ CAESAR SHIFTED CIPHERTEXT (?)
gn0x{\$3yb0wp_nsyjwhjuynt s_l0jx_g00rc0c0}?

Test all possible shifts (26-letter alphabet A-Z)

▶ DECRYPT (BRUTEFORCE)

MANUAL DECRYPTION AND PARAMETERS

★ SHIFT/KEY (NUMBER): AZRA...

☒ USE THE ENGLISH ALPHABET (26 LETTERS FROM A TO Z)

☐ USE THE ENGLISH ALPHABET AND ALSO SHIFT THE DIGITS 0-9

☐ USE THE LATIN ALPHABET IN THE TIME OF CAESAR (23 LETTERS, NO J, U OR W)

☐ USE THE ASCII TABLE (0-127) AS ALPHABET

☐ USE A CUSTOM ALPHABET (A-Z0-9 CHARS ONLY)

0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ

▶ DECRYPT