

राक्षसCTF

CHALLENGE NAME: [Data Trail]

DEV : [Pranav Bhosale]

CATEGORY: [Digital Forensics]

LEVEL: [Medium]



Challenge Description:

We have intercepted a modern military group while they were communicating their strategies. Inspired by the encryption methods of a famous Roman general who once declared , "Veni", "Vidi", "Vici", they're using encrypted methods to protect their sensitive information. Your mission is to analyze the provided file, uncover the hidden message, and figure out what they are planning.

Solution:

Step 1)

We need to filter out the packets that are carrying data so we will apply a custom filter that looks at incoming packages with HID data present.

```
frame.len==35 and !(usb.capdata==00:00:00:00:00:00:00) and  
!(usbhid.data == 00:00:00:00:00:00:00:00)
```

After applying the is filter we will save these packets into a new file called filtered.pcapng.

Step 2)

We then need to get the 2 important parts of the packets ie usb.capdata & usb.data_len==8 where we can analyse the keystrokes by extracting them into a .txt file which we will later feed into a Python script to parse the data. We will use Tshark to extract data from our filtered.pcapng file.

```
tshark -r ./filtered.pcapng -Y 'usb.capdata && usb.data_len == 8' -T  
fields -e usb.capdata | sed 's/./:~/g2'>filtered
```

Step 3)

A popular tool used to analyse keystrokes is <https://github.com/TeamRocketIst/ctf-usb-keyboard-parser>

We used this tool to extract the following string:

```
BoznwfHAL{0v3y4a10s_I3z3ya_Za0yr}
```

Step 4)

As the description suggests we need to use caser cipher to solve further but the configuration we need to set it as is given within the pcap file itself if we observe carefully at the last few packages we see that they are coming from a USB device with address 2 and which indicates that these indeed are mouse clicks so by counting the Number of Packets where the starting bits are **01** which means that the right mouse button was pressed .

If we count the number of such packets it turns out to be 5 so we need to set our cipher up for 5 shifts.

```
Frame 181: 35 bytes on wire (280 bits), 35 bytes captured (280 bits) on interface \\.\USBPcap1, id 0
USB URB
[Source: 1.2.1]
[Destination: host]
USBPcap pseudoheader length: 27
IRP ID: 0xffffd70d91828760
IRP USBD_STATUS: USBD_STATUS_SUCCESS (0x00000000)
URB Function: URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER (0x0009)
IRP information: 0x01, Direction: PDO -> FDO
URB bus id: 1
Device address: 2
Endpoint: 0x81, Direction: IN
URB transfer type: URB_INTERRUPT (0x01)
Packet Data Length: 8
[Request in: 178]
[Time from request: 0.886930000 seconds]
[bInterfaceClass: HID (0x03)]
HID Data: 0100000000000000
```

Here we can See that this is a packet with address as 2 and HID data as 01 which means this is indeed a Mouse Click.



Search for a tool

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

★ BROWSE THE [FULL dCODE TOOLS' LIST](#)

Results

Caesar Cipher – Shift by 5
F,G,H,I,K,L,...D,E
A,B,C,D,E,F,...Y,Z

□5 (□18) VishwaCTF{0p3r4t10n_D3s3rt_St0rm}%
□5 (□18) Gtesw\NFQ{0b3d4f10z_03e3df_Ef0dy}%

Caesar Cipher - dCode
Tag(s) : Substitution Cipher

Share

dCode and more



CAESAR CIPHER

Cryptography › Substitution Cipher › Caesar Cipher

CAESAR CIPHER DECODER

★ CAESAR SHIFTED CIPHERTEXT (?)
BoznwfHAL{0v3y4a10s_I3z3ya_Za0yr}%

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

► DECRYPT (BRUTEFORCE)

MANUAL DECRYPTION AND PARAMETERS

★ SHIFT/KEY (NUMBER): 5

☐ 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

See also: [ROT Cipher](#) — [Shift Cipher](#)

Flag: VishwaCTF{0p3r4t10n_D3s3rt_St0rm}