Forensics CTF 3

Platform: picoCTF 2025

Challenge Name: Ph4nt0m 1ntrud3r

Category: Forensics

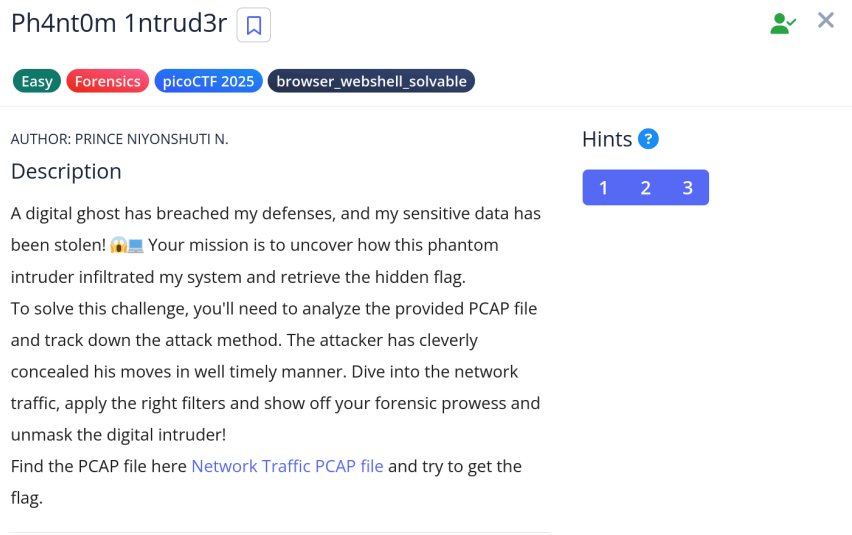
Difficulty: Easy

Submitted By: Gurleen Kaur Brar

Objective

The goal of the challenge was to analyze a network packet capture (PCAP) file to identify how an intruder exfiltrated sensitive data and extract the hidden flag. This required filtering TCP traffic, inspecting packet lengths, and decoding the hidden payload.

Challenge Description



Files and Tools Used

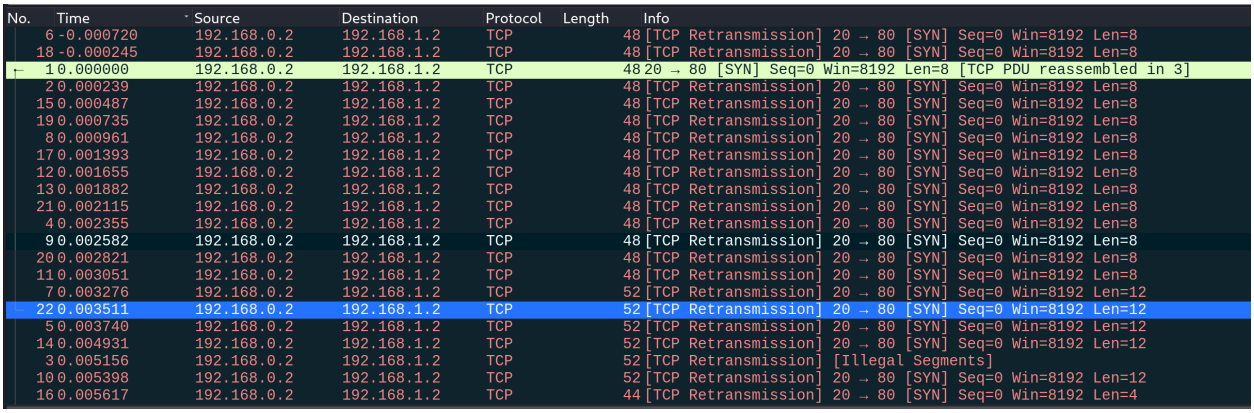
- **File Provided:** Network Traffic PCAP file
- **Tools Used:**

- Wireshark (for PCAP analysis)
- Kali Linux

Step-by-Step Process

Step 1: Load and Rearrange Timestamps in Wireshark

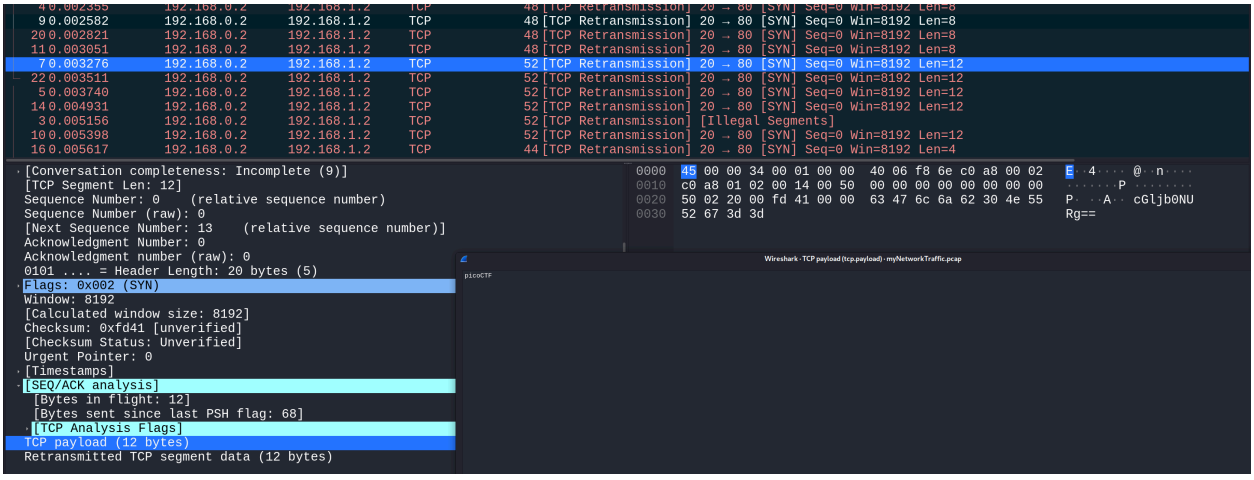
Opened the PCAP file in Wireshark and rearranged the timestamp column to chronological order. This allowed for better visibility of packet sequences and retransmissions.



No.	Time	Source	Destination	Protocol	Length	Info
6	-0.000720	192.168.0.2	192.168.1.2	TCP	48	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
18	-0.000245	192.168.0.2	192.168.1.2	TCP	48	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
1	0.000000	192.168.0.2	192.168.1.2	TCP	48	20 → 80 [SYN] Seq=0 Win=8192 Len=8 [TCP PDU reassembled in 3]
2	0.000239	192.168.0.2	192.168.1.2	TCP	48	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
15	0.000487	192.168.0.2	192.168.1.2	TCP	48	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
19	0.000735	192.168.0.2	192.168.1.2	TCP	48	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
8	0.000961	192.168.0.2	192.168.1.2	TCP	48	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
17	0.001393	192.168.0.2	192.168.1.2	TCP	48	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
12	0.001655	192.168.0.2	192.168.1.2	TCP	48	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
13	0.001882	192.168.0.2	192.168.1.2	TCP	48	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
21	0.002115	192.168.0.2	192.168.1.2	TCP	48	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
4	0.002355	192.168.0.2	192.168.1.2	TCP	48	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
9	0.002582	192.168.0.2	192.168.1.2	TCP	48	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
20	0.002821	192.168.0.2	192.168.1.2	TCP	48	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
11	0.003051	192.168.0.2	192.168.1.2	TCP	48	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
7	0.003276	192.168.0.2	192.168.1.2	TCP	52	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
22	0.003511	192.168.0.2	192.168.1.2	TCP	52	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
5	0.003740	192.168.0.2	192.168.1.2	TCP	52	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
14	0.004931	192.168.0.2	192.168.1.2	TCP	52	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
3	0.005156	192.168.0.2	192.168.1.2	TCP	52	[TCP Retransmission] [Illegal Segments]
10	0.005398	192.168.0.2	192.168.1.2	TCP	52	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
16	0.005617	192.168.0.2	192.168.1.2	TCP	44	[TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=4

Step 2: Filter Packets by Length = 12

Used Wireshark's filter to isolate TCP packets with a length of 12 bytes, a sign of possible hidden payloads or covert messages.



9	0.002582	192.168.0.2	192.168.1.2	TCP	48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
20	0.002821	192.168.0.2	192.168.1.2	TCP	48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
11	0.003051	192.168.0.2	192.168.1.2	TCP	48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
7	0.003276	192.168.0.2	192.168.1.2	TCP	52 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
22	0.003511	192.168.0.2	192.168.1.2	TCP	52 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
5	0.003740	192.168.0.2	192.168.1.2	TCP	52 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
14	0.004931	192.168.0.2	192.168.1.2	TCP	52 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
3	0.005156	192.168.0.2	192.168.1.2	TCP	52 [TCP Retransmission] [Illegal Segments]
10	0.005398	192.168.0.2	192.168.1.2	TCP	52 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
16	0.005617	192.168.0.2	192.168.1.2	TCP	44 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=4

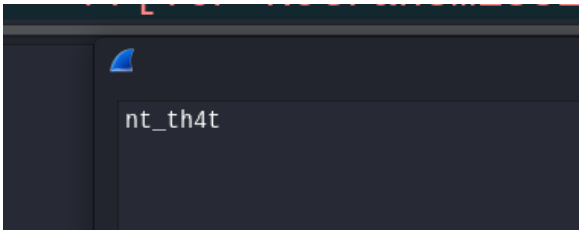
```
[Conversation completeness: Incomplete (9)]
[TCP Segment Len: 12]
Sequence Number: 0    (relative sequence number)
Sequence Number (raw): 0
[Next Sequence Number: 13    (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
0101 .... = Header Length: 20 bytes (5)
Flags: 0x002 (SYN)
Window: 8192
[Calculated window size: 8192]
Checksum: 0xfd41 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
[SEQ/ACK analysis]
  [Bytes in flight: 12]
  [Bytes sent since last PSH flag: 68]
[TCP Analysis Flags]
TCP payload (12 bytes)
Retransmitted TCP segment data (12 bytes)
```

```
0000  45 00 00 34 00 01 00 00  40 06 f8 6e c0 a8 00 02   E··4···· @··n····
0010  c0 a8 01 02 00 14 00 50  00 00 00 00 00 00 00 00   ·······P ········
0020  50 02 20 00 fd 41 00 00  63 47 6c 6a 62 30 4e 55   P· ··A·· cGljb0NU
0030  52 67 3d 3d                                       Rg==
```

Step 3: Inspect Packet Bytes

Right-clicked each of these 12-byte packets → Show packet bytes. The payload revealed ASCII fragments of the flag.

The fragments were decoded and reassembled into a full string.



Flag Submitted

```
picoCTF{1t_w4snt_th4t_34sy_tbh_4r_af160980}
```

The flag was successfully submitted and marked correct.