

挖矿流量分析

笔记本： 应急

创建时间： 2022/12/8 8:29

一、概述

在对常规的挖矿流量进行分析时，常用的方法是通过抓取流量中的域名、ip地址然后丢到威胁情报平台去分析，查看是否具有挖矿木马行为标记，但是这里可能存在的问题是假如威胁情报平台更新不及时，就无法及时准确判断出挖矿木马了。最近在参加一个ctf比赛时，正好遇到一个对流量进行分析，识别出挖矿流量，由于给出的流量包已经对一些ip或域名进行模糊，无法通过常规的方法进行筛选。所以只能依靠挖矿木马的一些流量特征进行筛选，总结了挖矿木马常见的一些特征。

二、常见特征

挖矿木马常用的协议为Stratum协议，而挖矿的流程主要是矿池和矿机之间进行交换，常见的交互过程如下：

1

矿机->矿池数据传输

Method字段	功能
mining.authorize	矿机登录认证
mining.extranonce.subscribe	向矿池表面矿机支持set_extranonce方法
mining.get_transactions	获取作业ID
mining.submit	矿机提交挖矿结果
mining.suggest_difficulty	挖矿难度的偏好
mining.suggest_target	挖矿目标的偏好
mining.capabilities(DRAFT)	矿机通知矿池其拥有的能力和可选项

矿池->矿机的数据传输

Method	功能
client.get_version	获取矿机版本信息
client.reconnect	等待指定时间(s)后重连
client.show_message	矿机展示信息
mining.notify	响应矿机的mining.subscribe请求
mining.set_difficulty	矿池更新难度
mining.set_extranonce	矿池更新extranonce
mining.set_goal(DRAFT)	通知矿机未来的工作目标(尚未使用)

给出的题目主要内容如下：

近期，校园网开展了整治虚拟货币“挖矿”专项活动。选手拿到的是一段时间的校园网网关部分流量数据，其中目标域名、IP已经过匿名化处理。请选手审计所给的数据包，实现一个不依赖于IOC的挖矿流量检测分析引擎，并找出数据包中的所有挖矿流量。

判题脚本说明

选手需要上传一个txt文件，文件的每一行代表挖矿流量在pcap文件包中的编号数字，如图所示：

下载相关流量数据包，进行分析，第一个感觉是无从下手，因为确实还没从没有这么细致的流量包进行数据分析

```
2022-10-24 12:15:18.248010 23.200.148.194 192.168.60.78 TCP 1394 [TCP ACKed unseen segment] [TCP segment of a reassembled PDU]
2022-10-24 12:15:18.288140 23.200.148.194 192.168.60.78 TLSv1.2 1394 Ignored Unknown Record
2022-10-24 12:16:32.102408 50.63.161.93 192.168.60.78 TCP 1394 [TCP ACKed unseen segment] [TCP Retransmission] 443 -> 63273 [ACK] Seq=1 Ack=518 Win=3
2022-10-24 12:16:32.103429 50.63.161.93 192.168.60.78 TCP 1394 [TCP ACKed unseen segment] 443 -> 63273 [ACK] Seq=1341 Ack=518 Win=30464 Len=1340
2022-10-24 12:16:32.103458 50.63.161.93 192.168.60.78 TLSv1.2 1394 [TCP ACKed unseen segment] , Ignored Unknown Record
2022-10-24 12:16:32.124915 50.63.161.93 192.168.60.78 TCP 1394 [TCP ACKed unseen segment] [TCP Retransmission] 443 -> 63275 [ACK] Seq=1 Ack=518 Win=3
2022-10-24 12:16:32.124915 50.63.161.93 192.168.60.78 TCP 1394 [TCP ACKed unseen segment] 443 -> 63275 [ACK] Seq=1341 Ack=518 Win=30464 Len=1340
2022-10-24 12:16:32.124915 50.63.161.93 192.168.60.78 TLSv1.2 1394 [TCP ACKed unseen segment] , Ignored Unknown Record
2022-10-24 12:16:32.229231 50.63.161.93 192.168.60.78 TCP 1394 [TCP ACKed unseen segment] [TCP Retransmission] 443 -> 63286 [ACK] Seq=1 Ack=518 Win=3
2022-10-24 12:16:32.229231 50.63.161.93 192.168.60.78 TCP 1394 [TCP ACKed unseen segment] 443 -> 63286 [ACK] Seq=1341 Ack=518 Win=30464 Len=1340
2022-10-24 12:16:32.229231 50.63.161.93 192.168.60.78 TLSv1.2 1394 [TCP ACKed unseen segment] , Ignored Unknown Record
2022-10-24 12:15:45.186883 52.10.191.40 192.168.60.78 TLSv1.2 1394 [TCP ACKed unseen segment] , Server Hello
2022-10-24 12:15:45.186883 52.10.191.40 192.168.60.78 TCP 1394 [TCP ACKed unseen segment] 443 -> 62501 [ACK] Seq=1341 Ack=518 Win=28032 Len=1340 [TCP
2022-10-24 12:15:45.187483 52.10.191.40 192.168.60.78 TCP 1394 [TCP ACKed unseen segment] 443 -> 62501 [ACK] Seq=2681 Ack=518 Win=28032 Len=1340 [TCP
2022-10-24 12:15:45.196626 52.10.191.40 192.168.60.78 TLSv1.2 1394 [TCP ACKed unseen segment] , Server Hello
2022-10-24 12:15:45.196626 52.10.191.40 192.168.60.78 TCP 1394 [TCP ACKed unseen segment] 443 -> 62502 [ACK] Seq=1341 Ack=518 Win=28032 Len=1340 [TCP

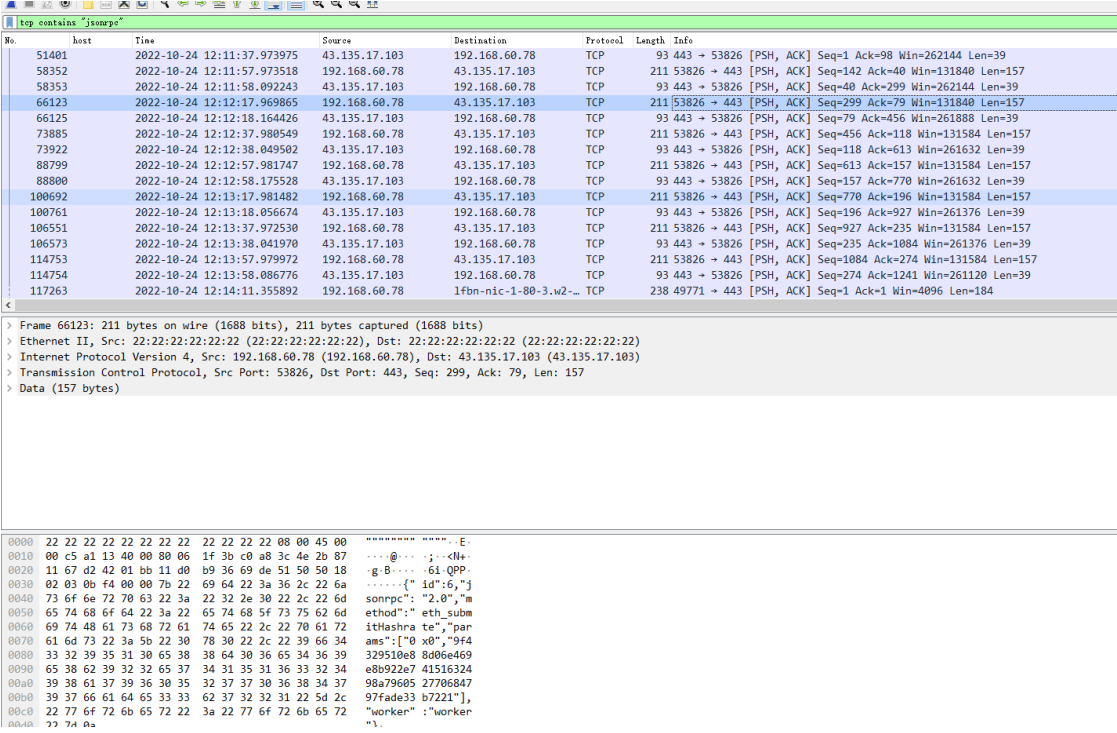
Window: 219
[Calculated window size: 28032]
[Window size scaling factor: 128]
Checksum: 0x9b47 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
> [Timestamps]
v [SEQ/ACK analysis]
  [Bytes in flight: 1340]
  [Bytes sent since last PSH flag: 4020]
  > [TCP Analysis Flags]
    TCP payload (1340 bytes)
    [Reassembled PDU in frame: 150772]
    [TCP segment data (1340 bytes)]
0000 22 22 22 22 22 22 22 22 22 22 22 22 22 08 00 45 04 .....E.
0010 05 64 33 e7 40 00 17 06 3a 80 34 0a bf 28 c0 a8 -d3@...:4-(-
0020 3c 4e 01 bb f4 25 fe 71 ef 37 90 f0 04 63 50 10 <H-~%q 7...cP
0030 00 db 9b 47 00 00 00 27 30 25 a0 23 a0 21 06 1f --Gp0%#-+
0040 58 74 74 70 3a 2f 2f 63 72 6c 2e 65 6e 74 72 75 http://c1.entru
0050 73 74 2e 6e 65 74 2f 67 32 63 61 2e 63 72 6c 30 st.net/g 2ca.crl0
0060 3b 06 03 55 1d 20 04 34 30 32 30 30 06 04 55 1d ;..U. -4 0280..U.
0070 20 00 30 28 30 26 06 08 2b 06 01 05 05 07 02 01 .0(0%..+.....
0080 16 1a 68 74 74 70 3a 2f 2f 77 77 77 2e 65 6e 74 ..http://www.ent
```

从网上找的部分挖矿流量特征：

```
'id','method','jsonrpc','params','result','login','pass','agent','job_id','seed_h
```

3、特征筛查

1、有了特征就会想到怎么从流量中去筛选，可直接在wireshak可以直接利用wireshark的搜索语法进行搜索，比如tcp contains "jsonrpc"、http contains等内容



2、还可尝试利用脚本对pcap文件进行分析，脚本主要是利用 pyshark函数，部分代码如下：

```
capture =
pyshark.FileCapture('F:\ctf\cryptomining.pcap', tshark_path="E:\ProgramFiles\Wire

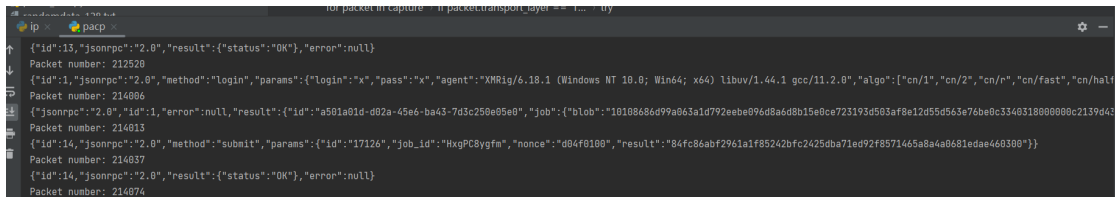
for packet in capture:
    # Print the packet info
    print("Packet number:", packet.number)
    print("Timestamp:", packet.sniff_time)
    print("Protocol:", packet.transport_layer)
    print("Source:", packet.ip.src)
```

```

print("Destination:", packet.ip.dst)
#提取tcp data部分
if packet.transport_layer == 'TCP':
    payload_hex=str(packet.tcp.payload).replace(':',',')
    payload_bytes = bytes.fromhex(payload_hex)
    payload_str = payload_bytes.decode('utf-8')

```

通过这几行代码基本就能分析出pcap包中的tcp源端口、目的端口、编号、tcp的数据等部分内容，然后在根据挖矿的流量特征进行匹配就可以了，打印出部分结果如下：



The image shows a Wireshark packet capture window with the filter 'tcp.flags.reset == 0'. The packet list shows several packets, with packet 214813 selected. The packet details pane shows the following JSONRPC data:

```

{"id":13,"jsonrpc":"2.0","result":{"status":"OK"},"error":null}
Packet number: 212520
{"id":1,"jsonrpc":"2.0","method":"login","params":{"login":"x","pass":"x","agent":"XMRig/6.18.1 (Windows NT 10.0; Win64; x64) libuv/1.44.1 gcc/11.2.0","algo":["cn/1","cn/2","cn/r","cn/fast","cn/half"]}
Packet number: 214806
{"jsonrpc":"2.0","id":1,"error":null,"result":{"id":"a501a01d-d02a-45e6-ba43-7d3c259e05e0","job":{"blob":"10188686d99a063a1d792eebe096d8a6d8b15e0ce723193d583af8e12d55d563e76be0c3340318000000c2139d42"}
Packet number: 214813
{"id":14,"jsonrpc":"2.0","method":"submit","params":{"id":"17126","job_id":"HxgPC8ygfM","nonce":"d04f0100","result":"84fc86abf2961a1f85242bfc2425dba71ed92f8571465a8a4a0681edae460300"}}
Packet number: 214837
{"id":14,"jsonrpc":"2.0","result":{"status":"OK"},"error":null}
Packet number: 214874

```

这只是提供了一个针对挖矿流量脱离IOC的识别思路，可能搜集的特征库还不够全。看网上还有一种简单方法就是直接导入安全检测设备，看设备的检测结果，如果具备条件的可以一试。