CSCI 466: Networks

Network Forensics: Tools and example investigation

Reese Pearsall

Fall 2024

Goals of Network Forensics

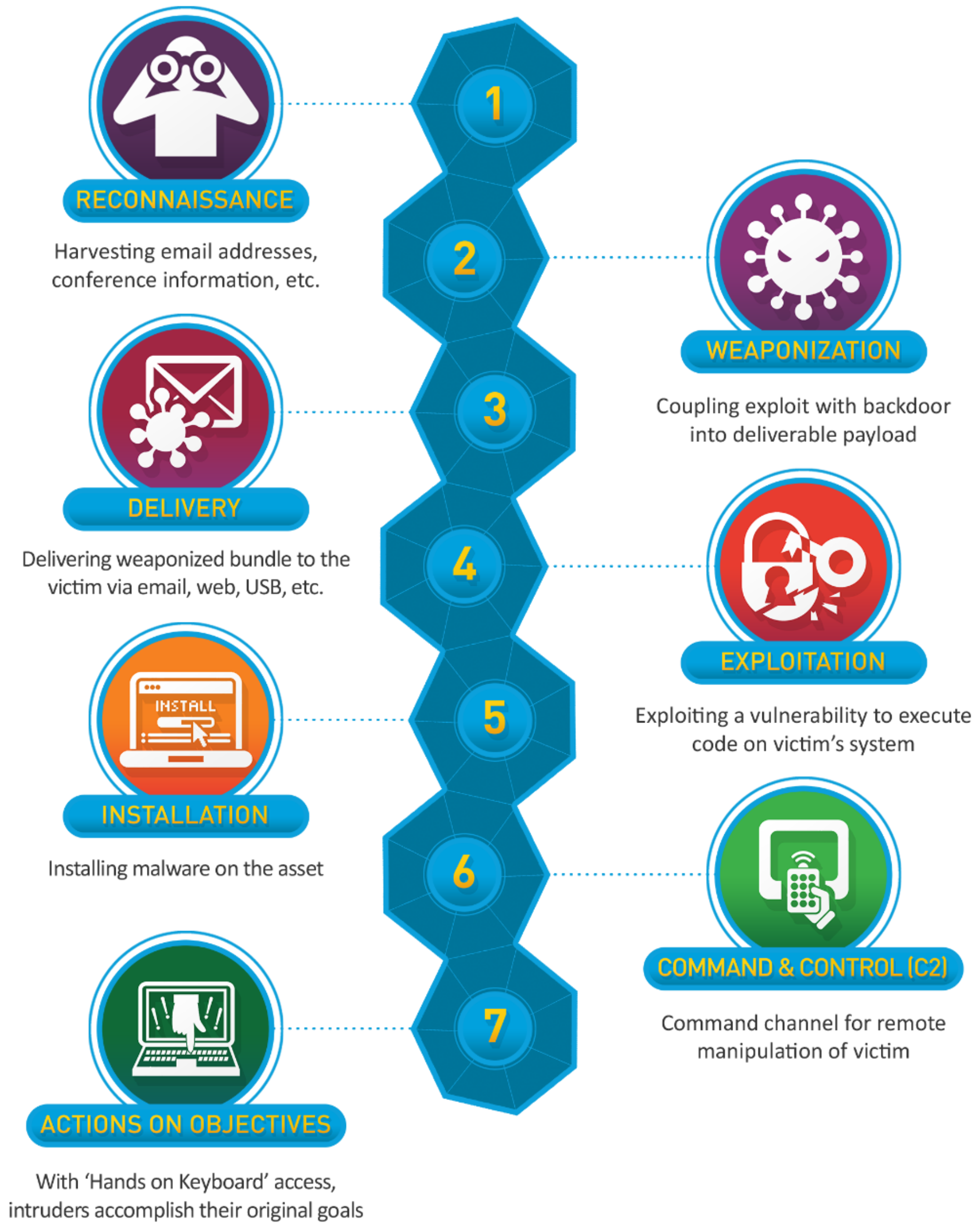
Answer important questions such as:

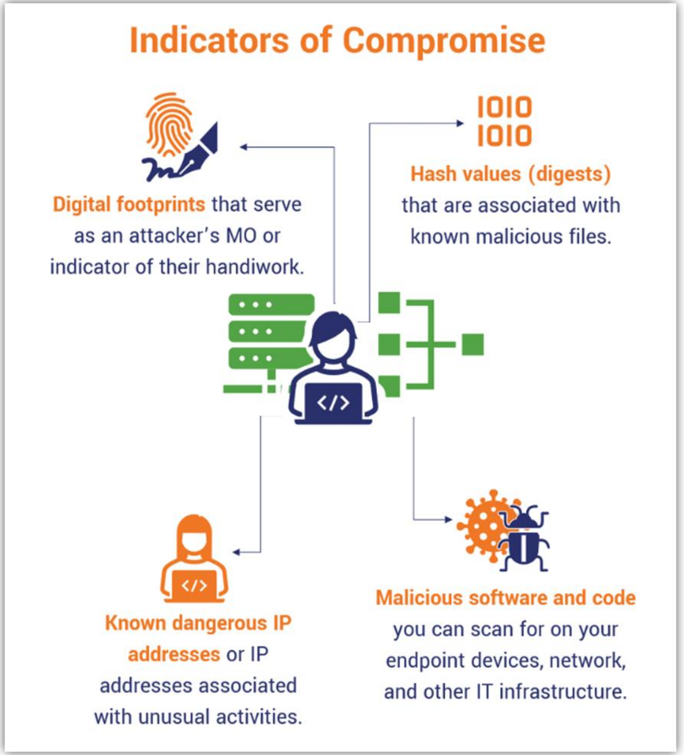
- When did the incident happen? What is the timeline?
- What is the root cause of the incident?
- Who attacked us?
- Why did they attack us?
- What is the scope of the damage?

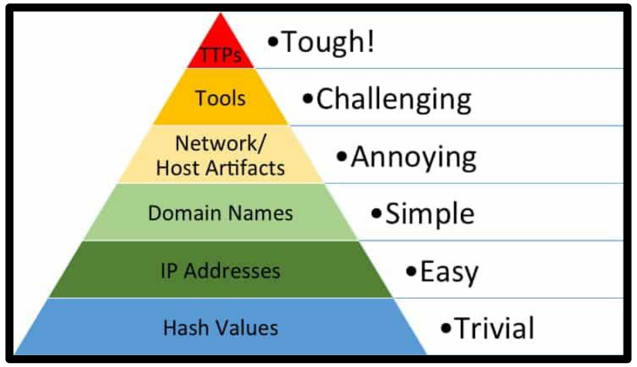
Reflection

- Did staff and organizations perform as expected?
- What will our organization do next time?
- What corrective actions need to happen?

The **Cyber Kill Chain** describes the typical steps a malicious actor carries out to conduct a cyber attack


1
RECONNAISSANCE
Harvesting email addresses, conference information, etc.
2
WEAPONIZATION
Coupling exploit with backdoor into deliverable payload
3
DELIVERY
Delivering weaponized bundle to the victim via email, web, USB, etc.
4
EXPLOITATION
Exploiting a vulnerability to execute code on victim's system
5
INSTALL
INSTALLATION
Installing malware on the asset
6
COMMAND & CONTROL (C2)
Command channel for remote manipulation of victim
7
ACTIONS ON OBJECTIVES
With 'Hands on Keyboard' access, intruders accomplish their original goals


Indicators of Compromise
Digital footprints that serve as an attacker's MO or indicator of their handiwork.
1010
1010
Hash values (digests) that are associated with known malicious files.
</>
Known dangerous IP addresses or IP addresses associated with unusual activities.
Malicious software and code you can scan for on your endpoint devices, network, and other IT infrastructure.

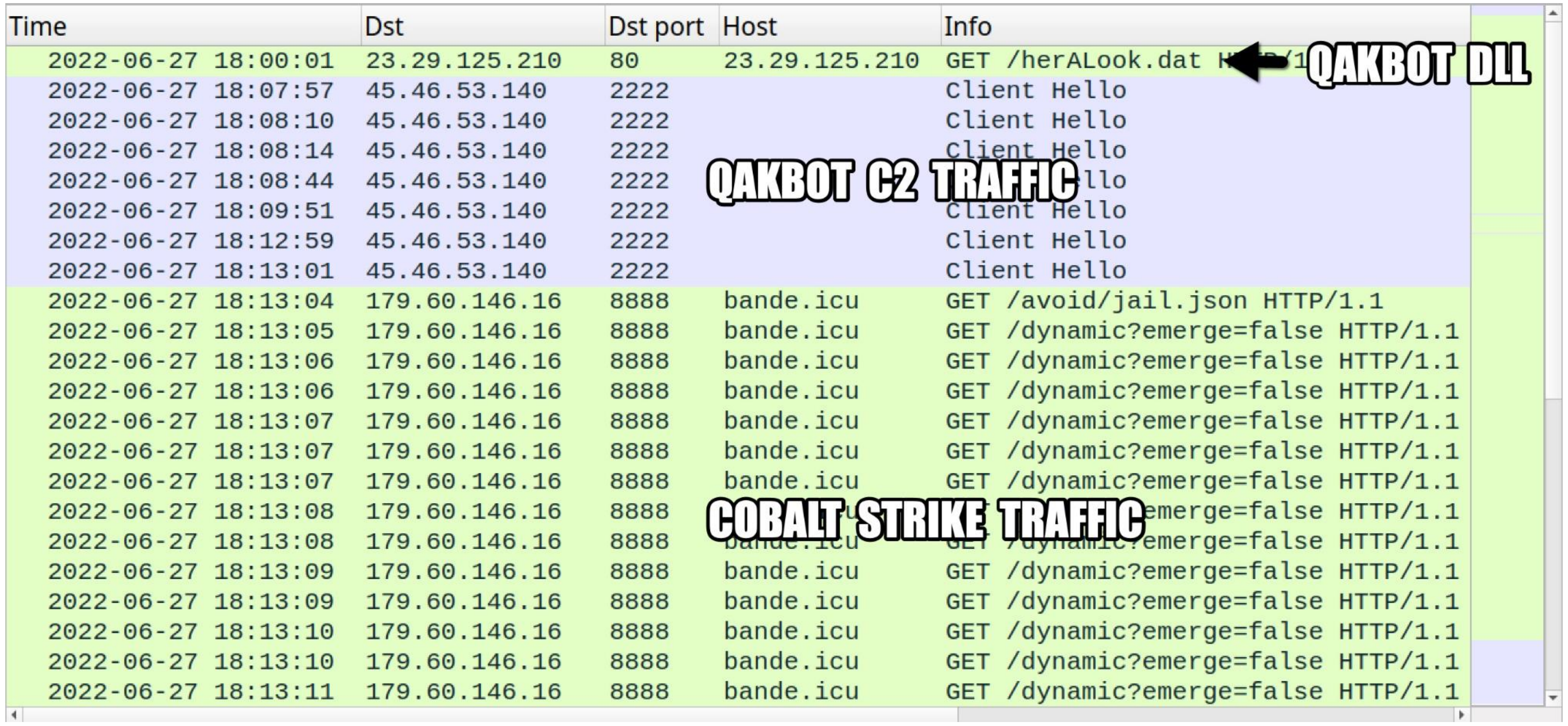


Malware File Types to Check for

- `.exe` files - Windows executable files
- `.dll` files - Dynamic Linked Libraries
- `.msi` files – Windows installers
- `.bat` files – Windows command line script
- `.vbs` scripts – Visual Basic Scripts
- `.js` scripts – Javascript file
- `.docx` files – Microsoft Word Document (can contain macros)
- `.xlsx` files – Microsoft Excel Spreadsheet (can contain macros)
- `.zip` files – Compressed Archive Files (may have scripts when unzipped)
- `.pdf` files – Can contain suspicious links, or a PDF reader vulnerability

There are many tools we can use for digital forensics, but we are covering some of the more well-known ones

Wireshark

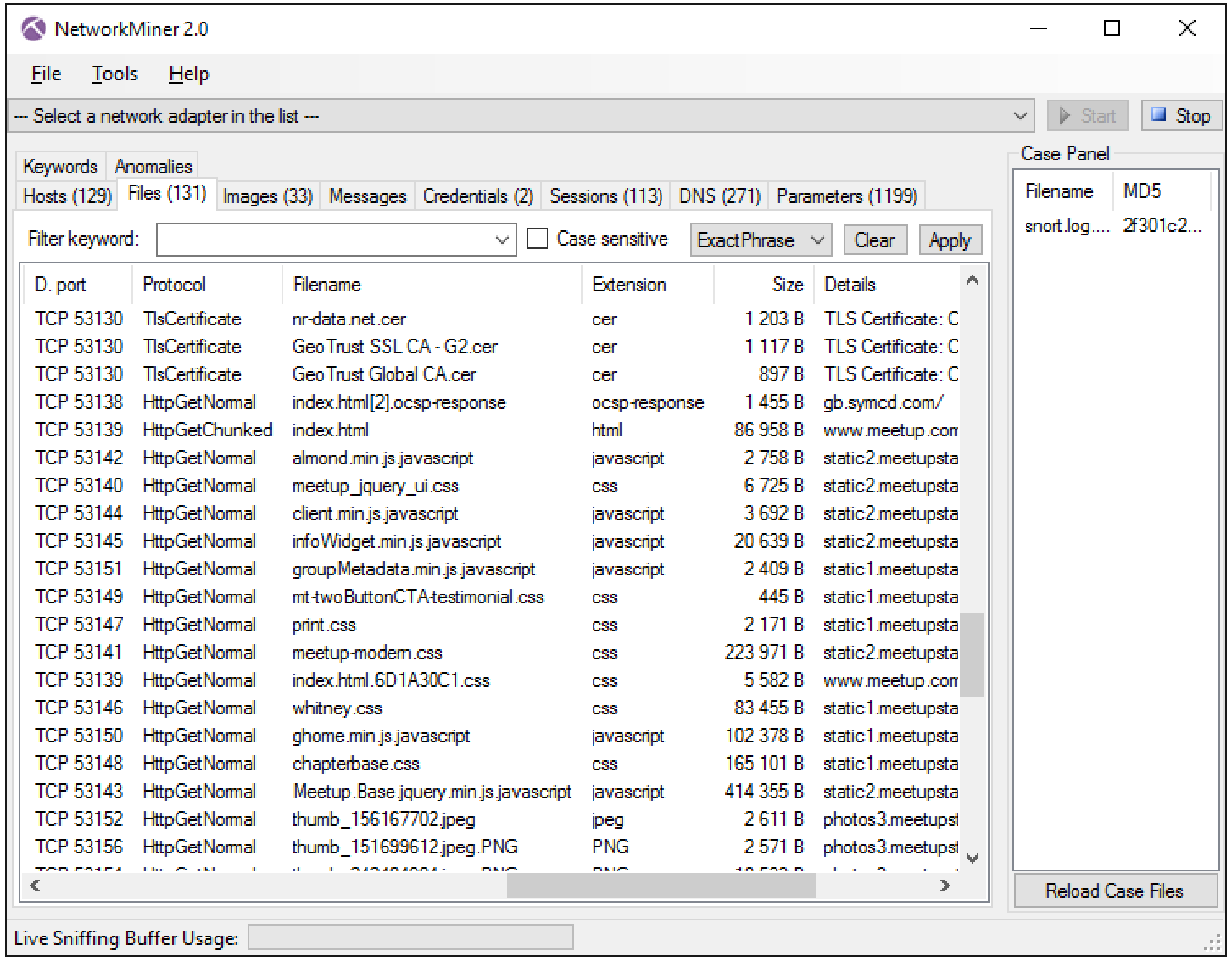


Time	Dst	Dst port	Host	Info
2022-06-27 18:00:01	23.29.125.210	80	23.29.125.210	GET /herALook.dat HTTP/1
2022-06-27 18:07:57	45.46.53.140	2222		Client Hello
2022-06-27 18:08:10	45.46.53.140	2222		Client Hello
2022-06-27 18:08:14	45.46.53.140	2222		Client Hello
2022-06-27 18:08:44	45.46.53.140	2222		Client Hello
2022-06-27 18:09:51	45.46.53.140	2222		Client Hello
2022-06-27 18:12:59	45.46.53.140	2222		Client Hello
2022-06-27 18:13:01	45.46.53.140	2222		Client Hello
2022-06-27 18:13:04	179.60.146.16	8888	bande.icu	GET /avoid/jail.json HTTP/1.1
2022-06-27 18:13:05	179.60.146.16	8888	bande.icu	GET /dynamic?emerge=false HTTP/1.1
2022-06-27 18:13:06	179.60.146.16	8888	bande.icu	GET /dynamic?emerge=false HTTP/1.1
2022-06-27 18:13:06	179.60.146.16	8888	bande.icu	GET /dynamic?emerge=false HTTP/1.1
2022-06-27 18:13:07	179.60.146.16	8888	bande.icu	GET /dynamic?emerge=false HTTP/1.1
2022-06-27 18:13:07	179.60.146.16	8888	bande.icu	GET /dynamic?emerge=false HTTP/1.1
2022-06-27 18:13:07	179.60.146.16	8888	bande.icu	GET /dynamic?emerge=false HTTP/1.1
2022-06-27 18:13:08	179.60.146.16	8888	bande.icu	GET /dynamic?emerge=false HTTP/1.1
2022-06-27 18:13:08	179.60.146.16	8888	bande.icu	GET /dynamic?emerge=false HTTP/1.1
2022-06-27 18:13:09	179.60.146.16	8888	bande.icu	GET /dynamic?emerge=false HTTP/1.1
2022-06-27 18:13:09	179.60.146.16	8888	bande.icu	GET /dynamic?emerge=false HTTP/1.1
2022-06-27 18:13:10	179.60.146.16	8888	bande.icu	GET /dynamic?emerge=false HTTP/1.1
2022-06-27 18:13:10	179.60.146.16	8888	bande.icu	GET /dynamic?emerge=false HTTP/1.1
2022-06-27 18:13:11	179.60.146.16	8888	bande.icu	GET /dynamic?emerge=false HTTP/1.1

We can use Wireshark to identify **specific** malicious packets, and find the exact moment where the infection started

C2 server information, Victim information, other relevant evidence

NetworkMiner


NetworkMiner 2.0
File Tools Help
--- Select a network adapter in the list ---
Start
Stop
Keywords Anomalies
Hosts (129) Files (131) Images (33) Messages Credentials (2) Sessions (113) DNS (271) Parameters (1199)
Filter keyword:
Case sensitive
ExactPhrase
Clear
Apply
Case Panel
Reload Case Files
Live Sniffing Buffer Usage:

D. port	Protocol	Filename	Extension	Size	Details
TCP 53130	TlsCertificate	nr-data.net.cer	cer	1 203 B	TLS Certificate: C
TCP 53130	TlsCertificate	GeoTrust SSL CA - G2.cer	cer	1 117 B	TLS Certificate: C
TCP 53130	TlsCertificate	GeoTrust Global CA.cer	cer	897 B	TLS Certificate: C
TCP 53138	HttpGetNormal	index.html[2].ocsp-response	ocsp-response	1 455 B	gb.symcd.com/
TCP 53139	HttpGetChunked	index.html	html	86 958 B	www.meetup.com
TCP 53142	HttpGetNormal	almond.min.js.javascript	javascript	2 758 B	static2.meetupsta
TCP 53140	HttpGetNormal	meetup_jquery_ui.css	css	6 725 B	static2.meetupsta
TCP 53144	HttpGetNormal	client.min.js.javascript	javascript	3 692 B	static2.meetupsta
TCP 53145	HttpGetNormal	infoWidget.min.js.javascript	javascript	20 639 B	static2.meetupsta
TCP 53151	HttpGetNormal	groupMetadata.min.js.javascript	javascript	2 409 B	static1.meetupsta
TCP 53149	HttpGetNormal	mt-twoButtonCTA-testimonial.css	css	445 B	static1.meetupsta
TCP 53147	HttpGetNormal	print.css	css	2 171 B	static1.meetupsta
TCP 53141	HttpGetNormal	meetup-modern.css	css	223 971 B	static2.meetupsta
TCP 53139	HttpGetNormal	index.html.6D1A30C1.css	css	5 582 B	www.meetup.com
TCP 53146	HttpGetNormal	whitney.css	css	83 455 B	static1.meetupsta
TCP 53150	HttpGetNormal	ghome.min.js.javascript	javascript	102 378 B	static1.meetupsta
TCP 53148	HttpGetNormal	chapterbase.css	css	165 101 B	static1.meetupsta
TCP 53143	HttpGetNormal	Meetup.Base.jquery.min.js.javascript	javascript	414 355 B	static2.meetupsta
TCP 53152	HttpGetNormal	thumb_156167702.jpeg	jpeg	2 611 B	photos3.meetupst
TCP 53156	HttpGetNormal	thumb_151699612.jpeg.PNG	PNG	2 571 B	photos3.meetupst

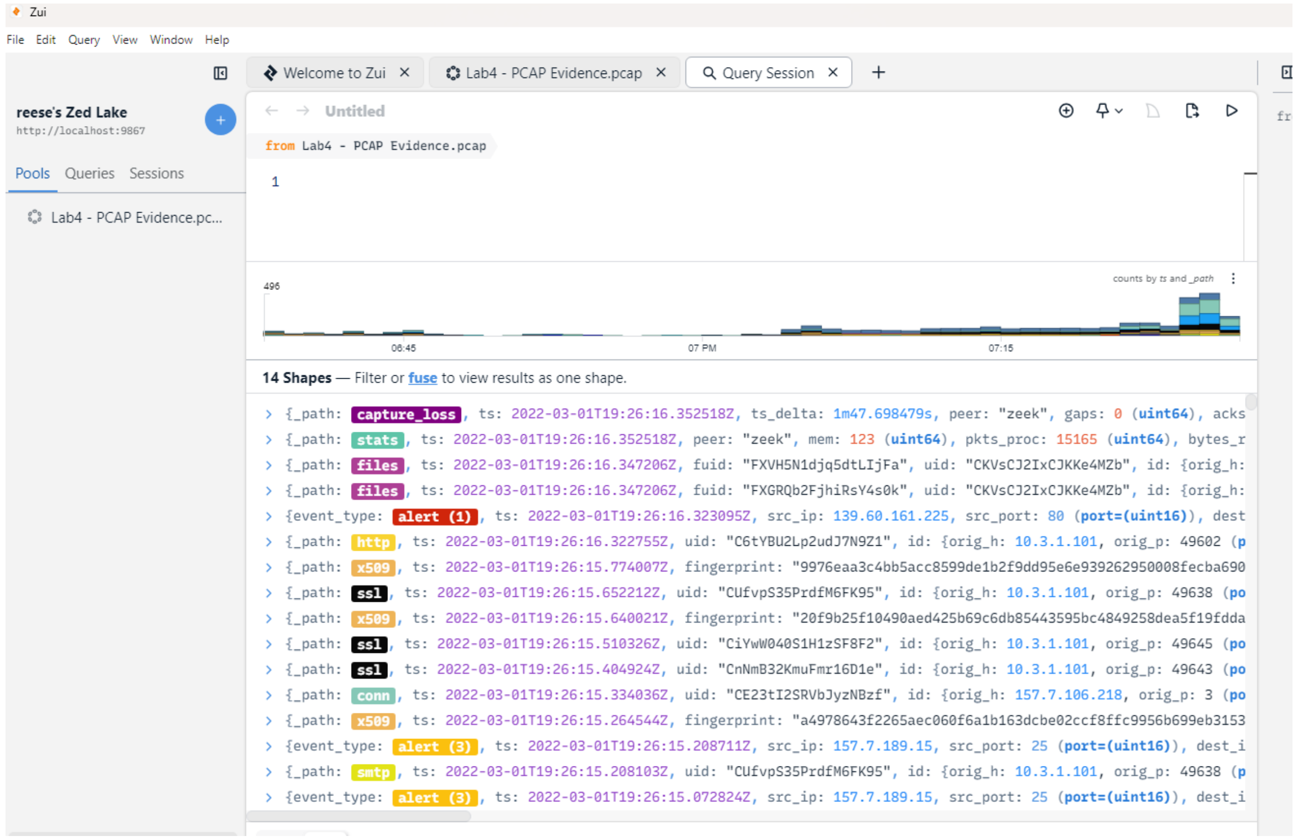
Filename	MD5
snort.log....	2f301c2...

NetworkMiner is an automated flow analysis tool that will identify all hosts, downloaded files, emails, logins from a pcap file and attempt to reassemble them for analysis

Very powerful tool that can provide many helpful insights during an investigation

NetworkMiner should always be run in a sandbox environment (VM) that is disconnected from the network

Zui



```
from Lab4 - PCAP Evidence.pcap
1
```

14 Shapes — Filter or fuse to view results as one shape.

```
> {_path: capture_loss, ts: 2022-03-01T19:26:16.352518Z, ts_delta: 1m47.698479s, peer: "zeek", gaps: 0 (uint64), acks
> {_path: stats, ts: 2022-03-01T19:26:16.352518Z, peer: "zeek", mem: 123 (uint64), pkts_proc: 15165 (uint64), bytes_r
> {_path: files, ts: 2022-03-01T19:26:16.347206Z, fuid: "FXVH5N1djq5dtLIjFa", uid: "CKVsCJ2IxCJKKe4MZb", id: {orig_h:
> {_path: files, ts: 2022-03-01T19:26:16.347206Z, fuid: "FXGRQb2FjhiRsY4s0k", uid: "CKVsCJ2IxCJKKe4MZb", id: {orig_h:
> {event_type: alert (1), ts: 2022-03-01T19:26:16.323095Z, src_ip: 139.60.161.225, src_port: 80 (port=(uint16)), dest
> {_path: http, ts: 2022-03-01T19:26:16.322755Z, uid: "C6tYBU2Lp2udJ7N9Z1", id: {orig_h: 10.3.1.101, orig_p: 49602 (p
> {_path: x509, ts: 2022-03-01T19:26:15.774007Z, fingerprint: "9976eaa3c4bb5acc8599de1b2f9dd95e6e93926295008fecba690
> {_path: ssl, ts: 2022-03-01T19:26:15.652212Z, uid: "CUfvpS35PrdfM6FK95", id: {orig_h: 10.3.1.101, orig_p: 49638 (po
> {_path: x509, ts: 2022-03-01T19:26:15.640021Z, fingerprint: "20f9b25f10490aed425b69c6db85443595bc4849258dea5f19fdda
> {_path: ssl, ts: 2022-03-01T19:26:15.510326Z, uid: "CiYwW040S1H1zSF8F2", id: {orig_h: 10.3.1.101, orig_p: 49645 (po
> {_path: ssl, ts: 2022-03-01T19:26:15.404924Z, uid: "CnNmB32KmuFmr16D1e", id: {orig_h: 10.3.1.101, orig_p: 49643 (po
> {_path: conn, ts: 2022-03-01T19:26:15.334036Z, uid: "CE23tI2SRVbJyzNBzf", id: {orig_h: 157.7.106.218, orig_p: 3 (po
> {_path: x509, ts: 2022-03-01T19:26:15.264544Z, fingerprint: "a4978643f2265aec060f6a1b163dcbe02ccf8ffc9956b699eb3153
> {event_type: alert (3), ts: 2022-03-01T19:26:15.208711Z, src_ip: 157.7.189.15, src_port: 25 (port=(uint16)), dest_i
> {_path: smtp, ts: 2022-03-01T19:26:15.208103Z, uid: "CUfvpS35PrdfM6FK95", id: {orig_h: 10.3.1.101, orig_p: 49638 (p
> {event_type: alert (3), ts: 2022-03-01T19:26:15.072824Z, src_ip: 157.7.189.15, src_port: 25 (port=(uint16)), dest_i
```

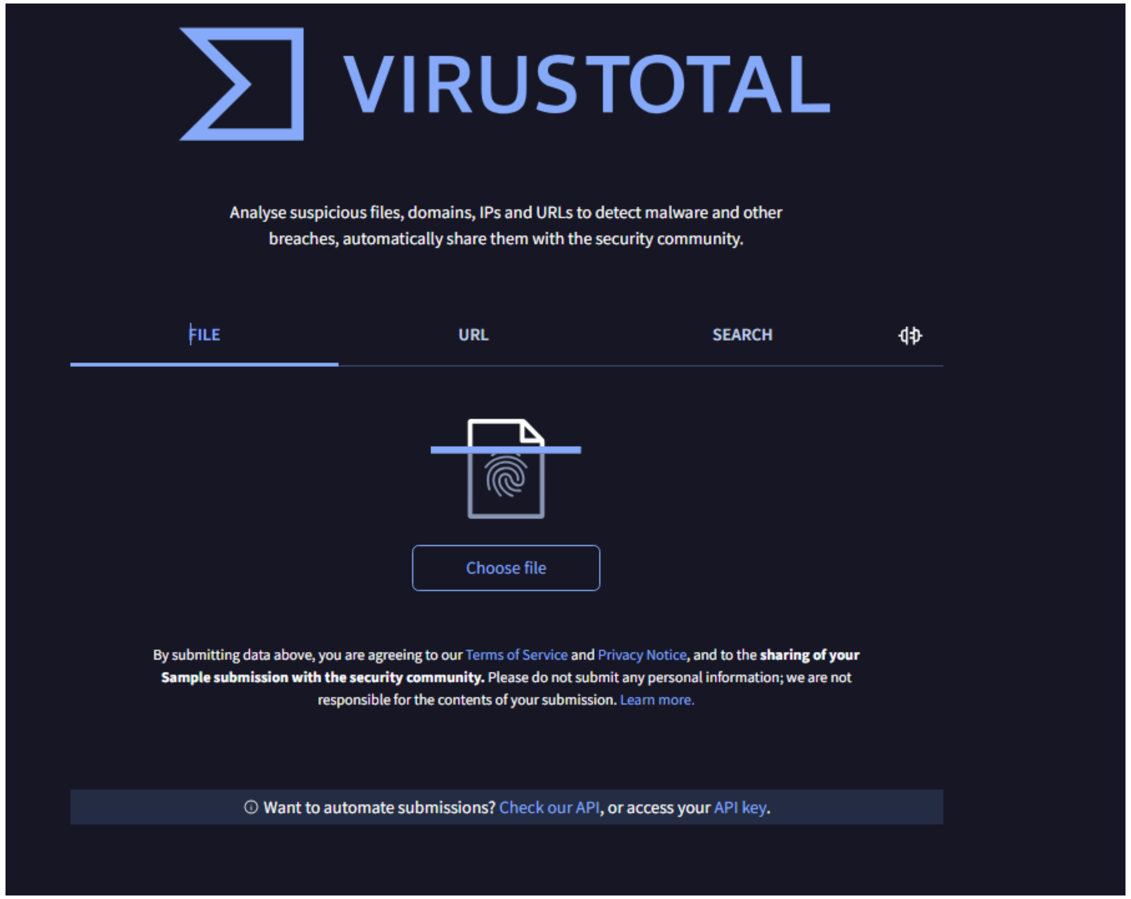
Zui (formerly known as **brim**) is a automated flow analysis tool that will identify any suspicious packets, emails, certificates, files from a pcap file and create a *timeline*

Won't assemble the files, so it will be safer to use

Can help identify potential IOCs

VirusTotal

VirusTotal is a massive database of known malware signatures and malicious fingerprints

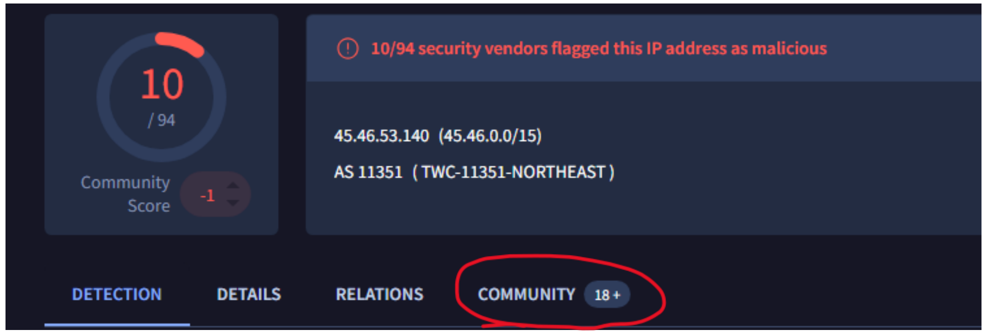

Analyse suspicious files, domains, IPs and URLs to detect malware and other breaches, automatically share them with the security community.

By submitting data above, you are agreeing to our Terms of Service and Privacy Notice, and to the **sharing of your Sample submission with the security community.** Please do not submit any personal information; we are not responsible for the contents of your submission. Learn more.

Want to automate submissions? Check our API, or access your API key.

We can provide signatures from our investigation to see if they have been flagged as malicious in the past

- File Hashes
- IP Addresses
- Domain Names

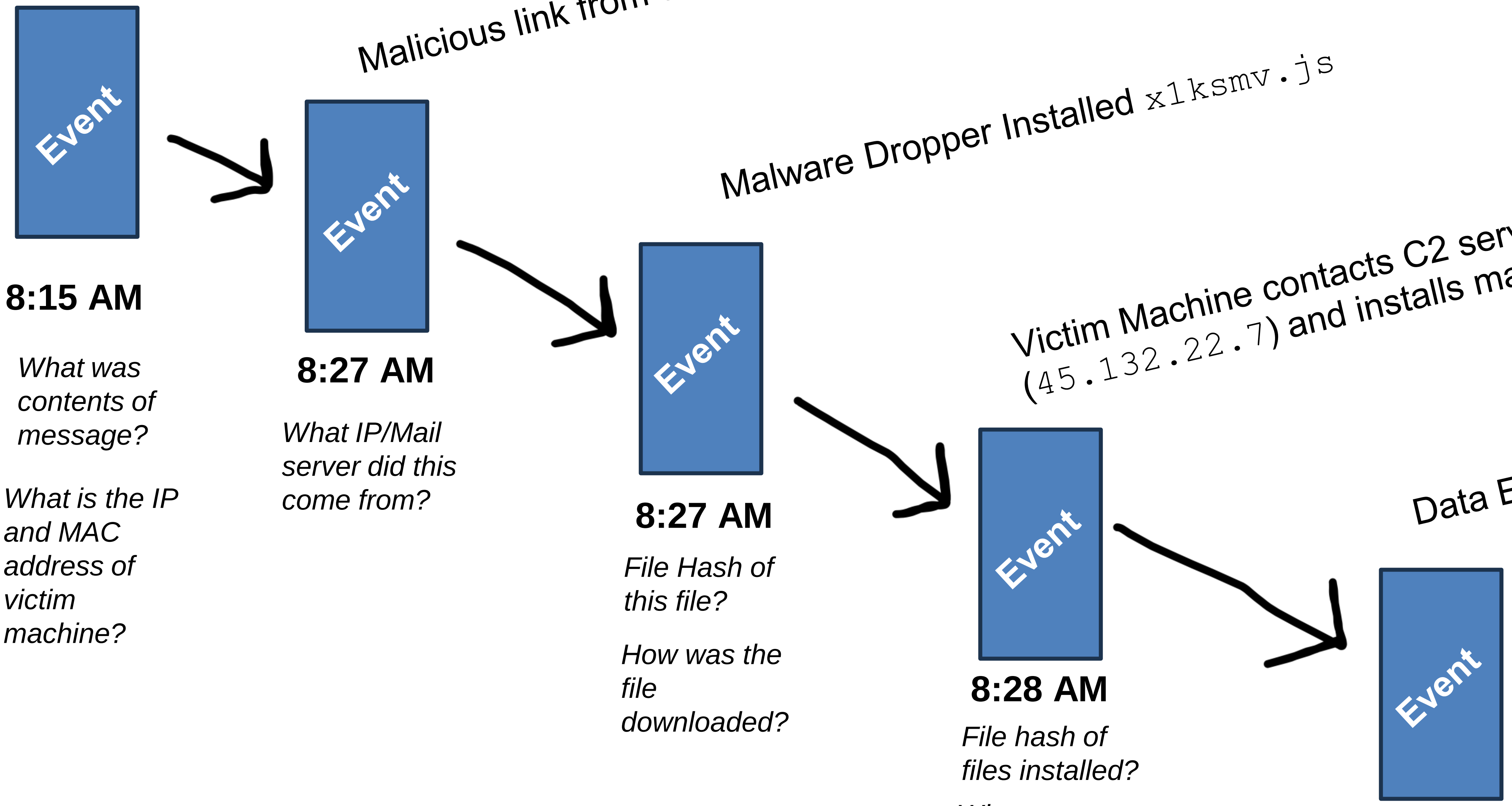

10/94 security vendors flagged this IP address as malicious

45.46.53.140 (45.46.0.0/15)

AS 11351 (TWC-11351-NORTHEAST)

The community tab will provide more context around the malicious signatures

Example Timeline



Email Received from `trustme@hacker.com`

8:15 AM

What was contents of message?

What is the IP and MAC address of victim machine?

Malicious link from email clicked on

8:27 AM

What IP/Mail server did this come from?

Malware Dropper Installed `x1ksmv.js`

8:27 AM

File Hash of this file?

How was the file downloaded?

Victim Machine contacts C2 server (`45.132.22.7`) and installs malware

8:28 AM

File hash of files installed?

What ports were used?

Data Exfiltration to C2 server

8:33 AM

What protocols were used?

What data was stolen?

Any evidence of persistence?

Let’s do an example investigation with real captured malicious traffic

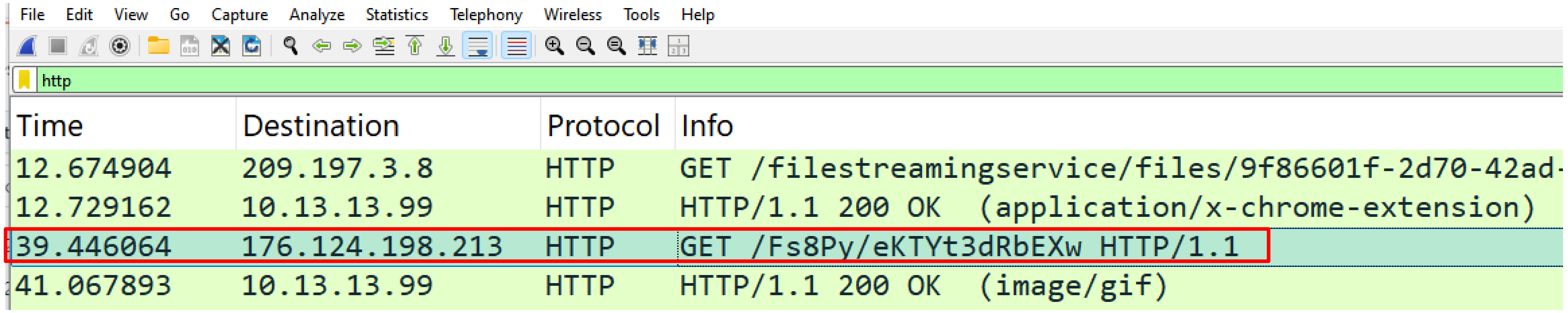
We will want to keep a notepad to put relevant information as we are analyzing the PCAP file

We will want to be constructing a timeline as we are doing the investigation

babe please stop you aren’t a hard-boiled 1940’s
network forensics investigator
beat it toots these streets ain’t what they used to be

Let’s do an example investigation with real captured malicious traffic

First, let's try to find the piece of malware that was installed

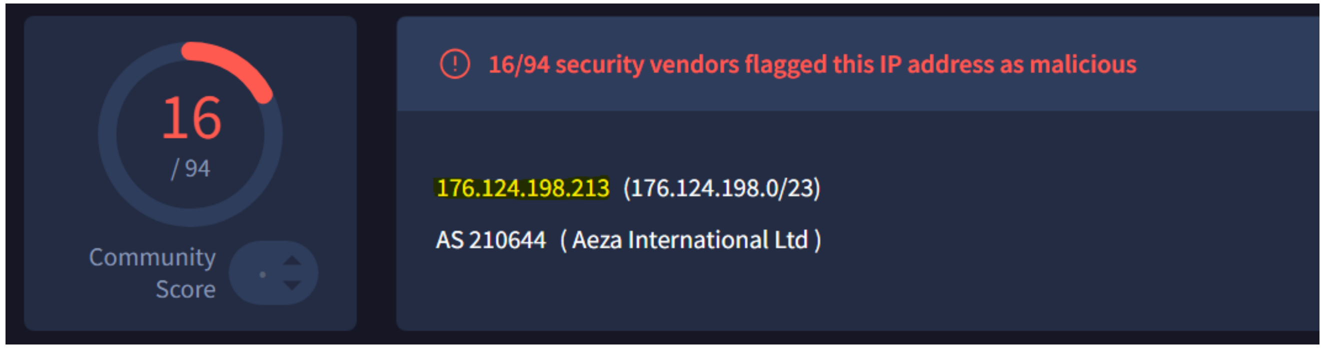

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
http
```

Time	Destination	Protocol	Info
12.674904	209.197.3.8	HTTP	GET /filestreamingservice/files/9f86601f-2d70-42ad-
12.729162	10.13.13.99	HTTP	HTTP/1.1 200 OK (application/x-chrome-extension)
39.446064	176.124.198.213	HTTP	GET /Fs8Py/eKTYt3dRbEXw HTTP/1.1
41.067893	10.13.13.99	HTTP	HTTP/1.1 200 OK (image/gif)

This HTTP Get request seems suspicious

- Strange URL in the header



16/94 security vendors flagged this IP address as malicious

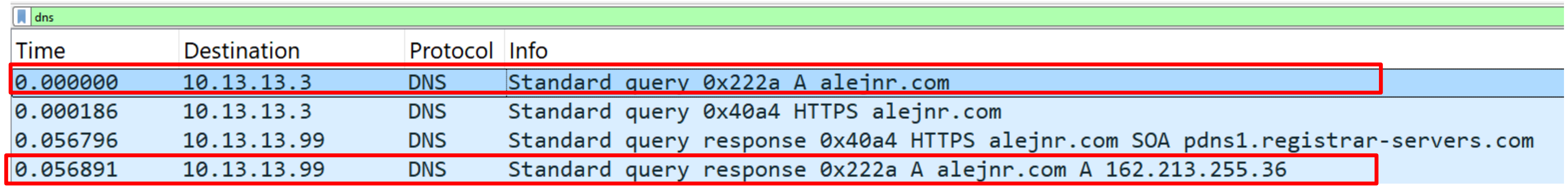
176.124.198.213 (176.124.198.0/23)

AS 210644 (Aeza International Ltd)

VirusTotal confirms that this IP address is malicious

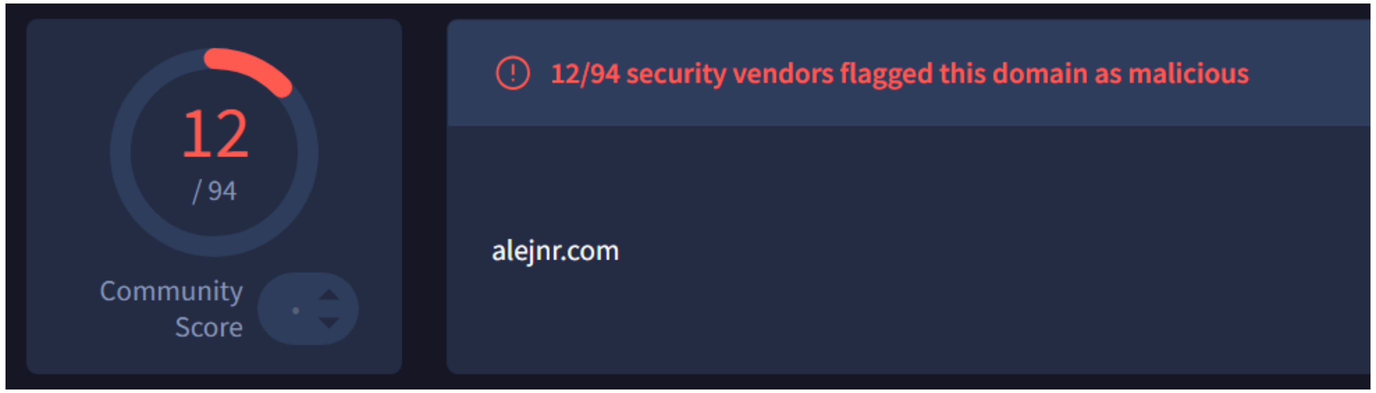
Investigating DNS

Many of the DNS requests were made to "normal" website. However, this domain seemed a bit strange and this request was made shortly before the malicious HTTP request



```
dns
Time        Destination    Protocol  Info
0.000000    10.13.13.3     DNS       Standard query 0x222a A alejnr.com
0.000186    10.13.13.3     DNS       Standard query 0x40a4 HTTPS alejnr.com
0.056796    10.13.13.99    DNS       Standard query response 0x40a4 HTTPS alejnr.com SOA pdns1.registrar-servers.com
0.056891    10.13.13.99    DNS       Standard query response 0x222a A alejnr.com A 162.213.255.36
```

This domain had been flagged as malicious in the past



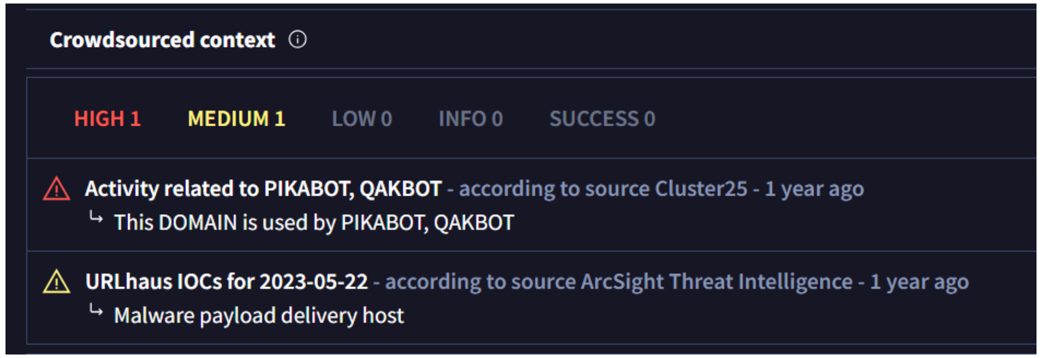
12 / 94

Community Score

12/94 security vendors flagged this domain as malicious

alejnr.com

VirusTotal also provides context as to why it was flagged as malicious



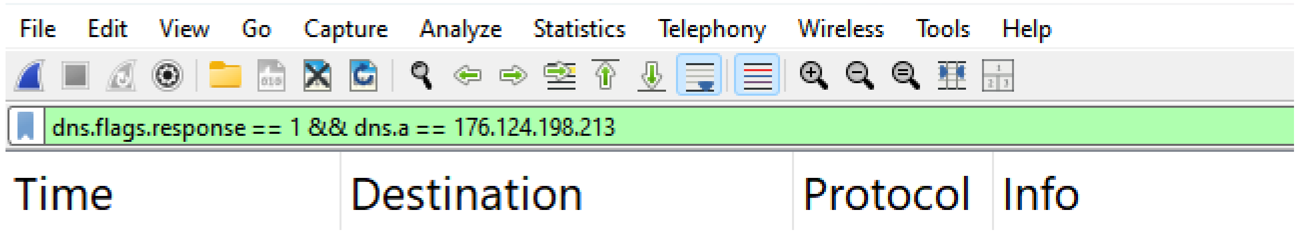
Crowdsourced context

HIGH 1   MEDIUM 1   LOW 0   INFO 0   SUCCESS 0

Activity related to PIKABOT, QAKBOT - according to source Cluster25 - 1 year ago
↳ This DOMAIN is used by PIKABOT, QAKBOT

URLhaus IOCs for 2023-05-22 - according to source ArcSight Threat Intelligence - 1 year ago
↳ Malware payload delivery host

Where did this IP address come from?

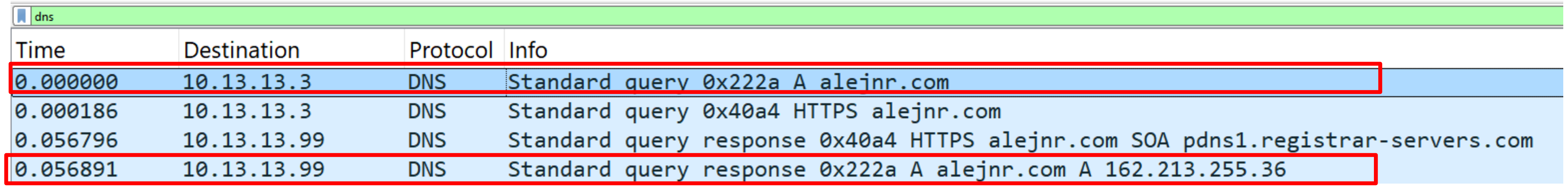

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
dns.flags.response == 1 && dns.a == 176.124.198.213
Time            Destination            Protocol   Info
```

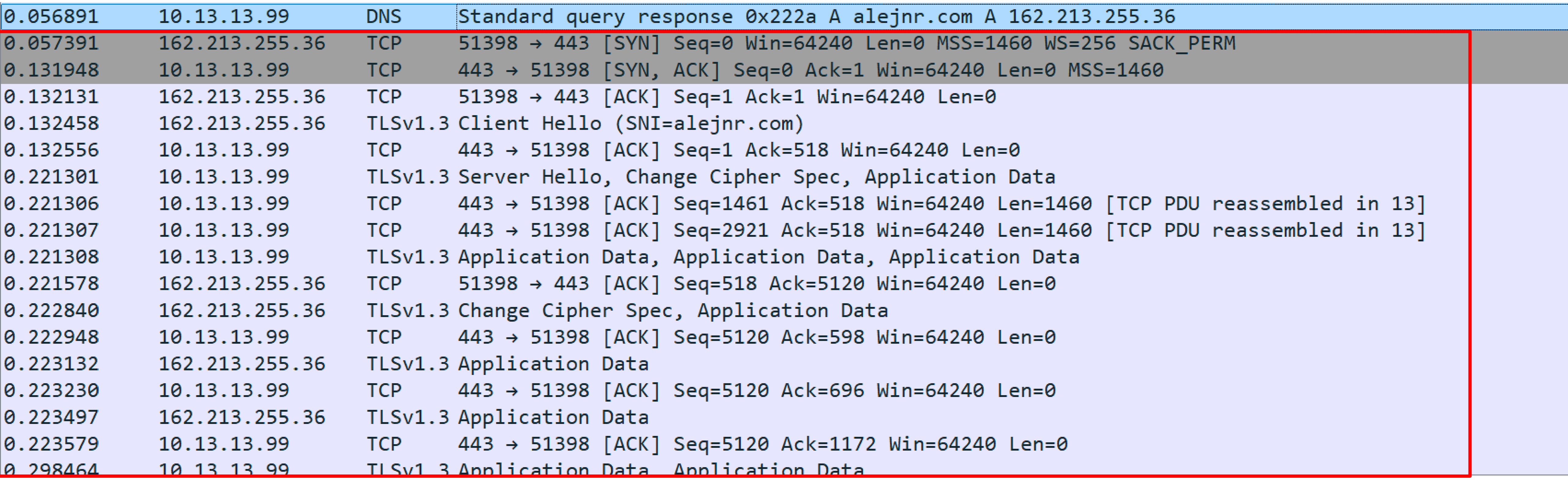
This malicious IP did not come from a DNS response.

This could be an indicator that another piece of malware contacted that IP addressed and issued the HTTP request (malware dropper)

Many of the DNS requests were made to "normal" website. However, this domain seemed a bit strange and this request was made shortly before the malicious HTTP request

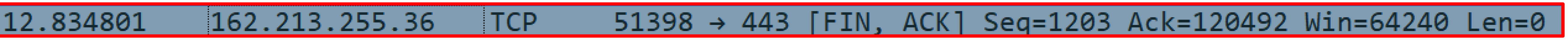


```
dns
Time        Destination    Protocol  Info
0.000000    10.13.13.3     DNS       Standard query 0x222a A alejnr.com
0.000186    10.13.13.3     DNS       Standard query 0x40a4 HTTPS alejnr.com
0.056796    10.13.13.99    DNS       Standard query response 0x40a4 HTTPS alejnr.com SOA pdns1.registrar-servers.com
0.056891    10.13.13.99    DNS       Standard query response 0x222a A alejnr.com A 162.213.255.36
```

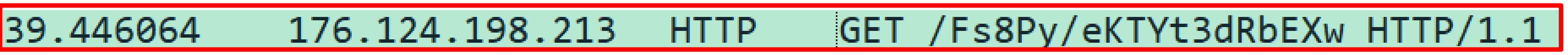


```
0.056891    10.13.13.99      DNS     Standard query response 0x222a A alejnr.com A 162.213.255.36
0.057391    162.213.255.36   TCP     51398 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
0.131948    10.13.13.99      TCP     443 → 51398 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
0.132131    162.213.255.36   TCP     51398 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
0.132458    162.213.255.36   TLSv1.3 Client Hello (SNI=alejnr.com)
0.132556    10.13.13.99      TCP     443 → 51398 [ACK] Seq=1 Ack=518 Win=64240 Len=0
0.221301    10.13.13.99      TLSv1.3 Server Hello, Change Cipher Spec, Application Data
0.221306    10.13.13.99      TCP     443 → 51398 [ACK] Seq=1461 Ack=518 Win=64240 Len=1460 [TCP PDU reassembled in 13]
0.221307    10.13.13.99      TCP     443 → 51398 [ACK] Seq=2921 Ack=518 Win=64240 Len=1460 [TCP PDU reassembled in 13]
0.221308    10.13.13.99      TLSv1.3 Application Data, Application Data, Application Data
0.221578    162.213.255.36   TCP     51398 → 443 [ACK] Seq=518 Ack=5120 Win=64240 Len=0
0.222840    162.213.255.36   TLSv1.3 Change Cipher Spec, Application Data
0.222948    10.13.13.99      TCP     443 → 51398 [ACK] Seq=5120 Ack=598 Win=64240 Len=0
0.223132    162.213.255.36   TLSv1.3 Application Data
0.223230    10.13.13.99      TCP     443 → 51398 [ACK] Seq=5120 Ack=696 Win=64240 Len=0
0.223497    162.213.255.36   TLSv1.3 Application Data
0.223579    10.13.13.99      TCP     443 → 51398 [ACK] Seq=5120 Ack=1172 Win=64240 Len=0
0.298464    10.13.13.99      TLSv1.3 Application Data, Application Data
```

Immediately after we resolve the host name of alejnr.com, we start a TCP/TLS connection with that



```
12.834801    162.213.255.36    TCP     51398 → 443 [FIN, ACK] Seq=1203 Ack=120492 Win=64240 Len=0
```



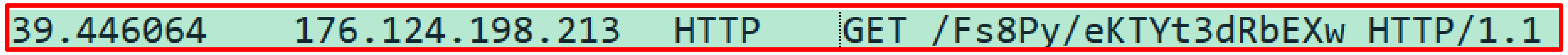
```
39.446064    176.124.198.213    HTTP    GET /Fs8Py/eKTYt3dRbEXw HTTP/1.1
```

We stop talking to that malicious domain, and then shortly after the malicious HTTP GET request is done

Theory:

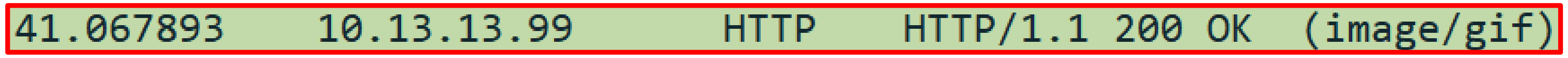
We reached out to this malicious domain alejnr.com and downloaded a malicious file which issued this HTTP request

What is in this malicious HTTP request?



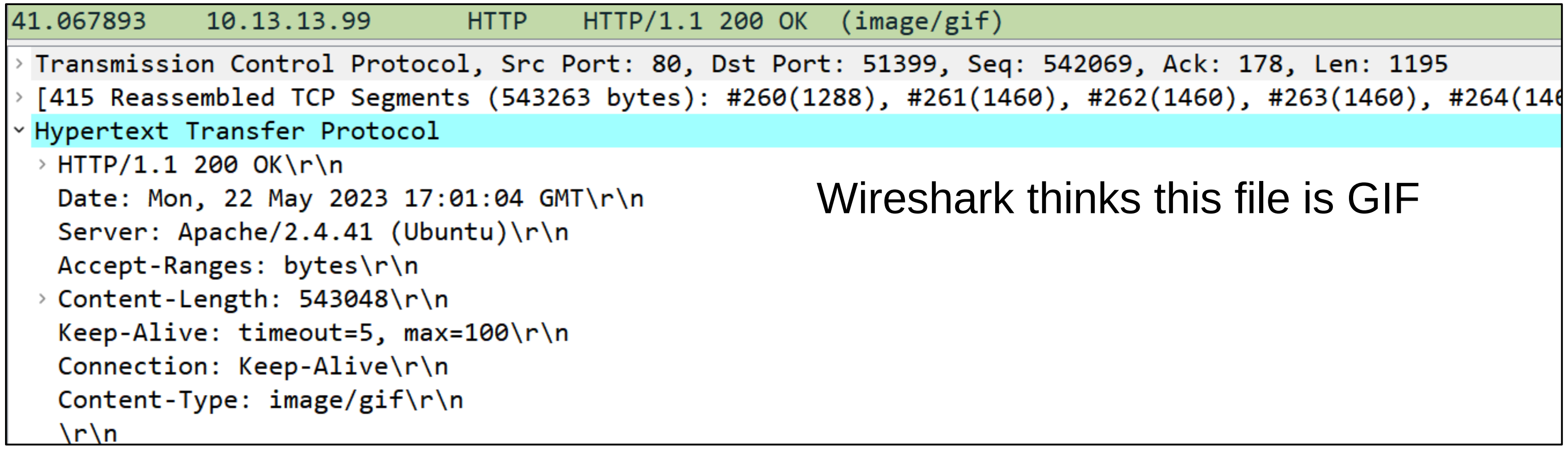
```
39.446064     176.124.198.213    HTTP     GET /Fs8Py/eKTYt3dRbEXw HTTP/1.1
```

Request



```
41.067893     10.13.13.99        HTTP     HTTP/1.1 200 OK  (image/gif)
```

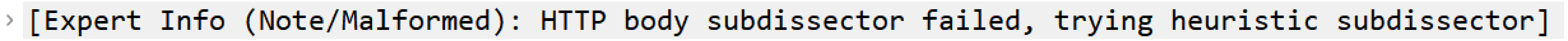
Response



```
41.067893     10.13.13.99        HTTP     HTTP/1.1 200 OK  (image/gif)
› Transmission Control Protocol, Src Port: 80, Dst Port: 51399, Seq: 542069, Ack: 178, Len: 1195
› [415 Reassembled TCP Segments (543263 bytes): #260(1288), #261(1460), #262(1460), #263(1460), #264(14
˅ Hypertext Transfer Protocol
  › HTTP/1.1 200 OK\r\n
    Date: Mon, 22 May 2023 17:01:04 GMT\r\n
    Server: Apache/2.4.41 (Ubuntu)\r\n
    Accept-Ranges: bytes\r\n
  › Content-Length: 543048\r\n
    Keep-Alive: timeout=5, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: image/gif\r\n
    \r\n
```

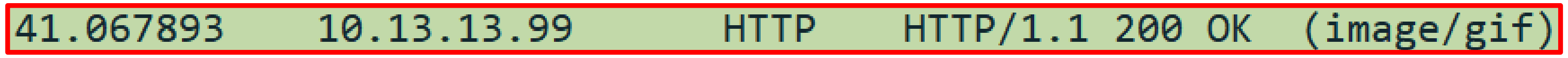
Wireshark thinks this file is GIF

Wireshark notices that there is something weird going on…



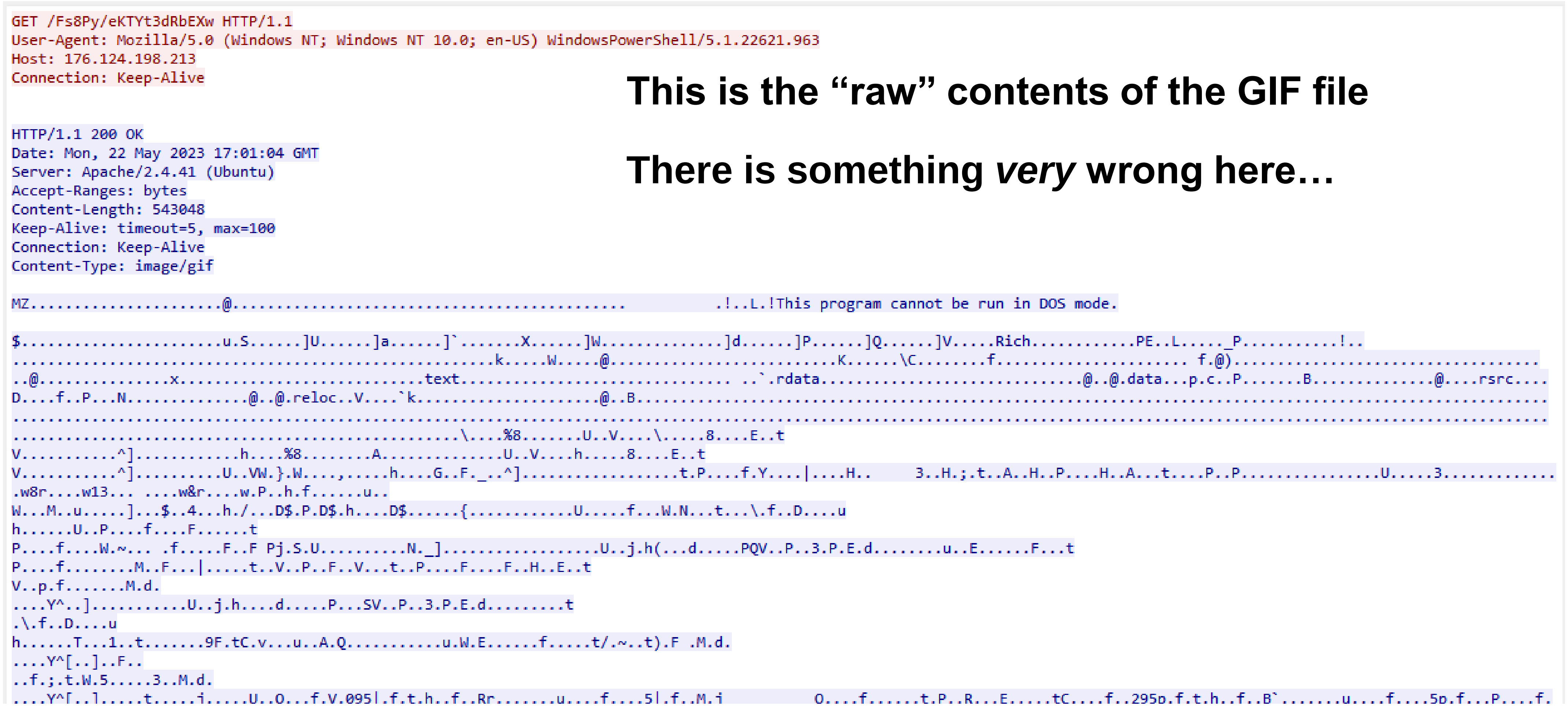
```
› [Expert Info (Note/Malformed): HTTP body subdissector failed, trying heuristic subdissector]
```

What is in this malicious HTTP request?



```
41.067893    10.13.13.99        HTTP    HTTP/1.1 200 OK  (image/gif)
```

Right click → Follow TCP Stream

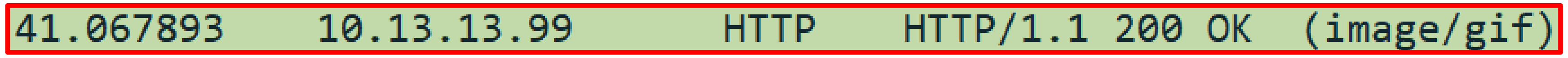


```
GET /Fs8Py/eKTYt3dRbEXw HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT; Windows NT 10.0; en-US) WindowsPowerShell/5.1.22621.963
Host: 176.124.198.213
Connection: Keep-Alive


HTTP/1.1 200 OK
Date: Mon, 22 May 2023 17:01:04 GMT
Server: Apache/2.4.41 (Ubuntu)
Accept-Ranges: bytes
Content-Length: 543048
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: image/gif

MZ......................@........................................         .!..L.!This program cannot be run in DOS mode.

$.......................u.S......]U......]a......]`.......X......]W..............]d......]P......]Q......]V.....Rich............PE..L....._P...........!..
.......................................................k.....W.....@........................K......\C........f.....................  f.@)..................................
..@...............x...........................text..............................  ..`.rdata...........................@..@.data...p.c..P.......B.............@....rsrc....
D....f..P...N..............@..@.reloc..V....`k....................@..B..................................................................................................
............................................................................................................................................................................
...................................................\....%8.......U..V....\.....8....E..t
V...........^]............h....%8........A...............U..V....h.....8....E..t
V...........^]..........U..VW.}.W....,.....h....G..F._..^]..................t.P....f.Y....|....H..     3..H.;.t..A..H..P....H..A...t....P..P................U.....3............
.w8r....w13...  ....w&r....w.P..h.f......u..
W...M..u.....]...$..4...h./...D$.P.D$.h....D$......{............U.....f...W.N...t...\.f..D....u
h......U..P....f....F......t
P....f....W.~...  .f.....F..F Pj.S.U..........N._]..................U..j.h(...d.....PQV..P..3.P.E.d........u..E......F...t
P....f.........M..F...|.....t..V..P..F..V...t..P....F....F..H..E..t
V..p.f.......M.d.
....Y^..]...........U..j.h....d.....P...SV..P..3.P.E.d.........t
.\.f..D....u
h......T...1..t.......9F.tC.v...u..A.Q...........u.W.E......f.....t/.~..t).F .M.d.
....Y^[..]..F..
..f.;.t.W.5.....3..M.d.
....Y^[..].....t.....i.....U..O...f.V.095|.f.t.h..f..Rr.......u....f....5|.f..M.i          O....f......t.P..R...E.....tC....f..295p.f.t.h..f..B`.......u....f....5p.f...P....f.
```

This is the “raw” contents of the GIF file

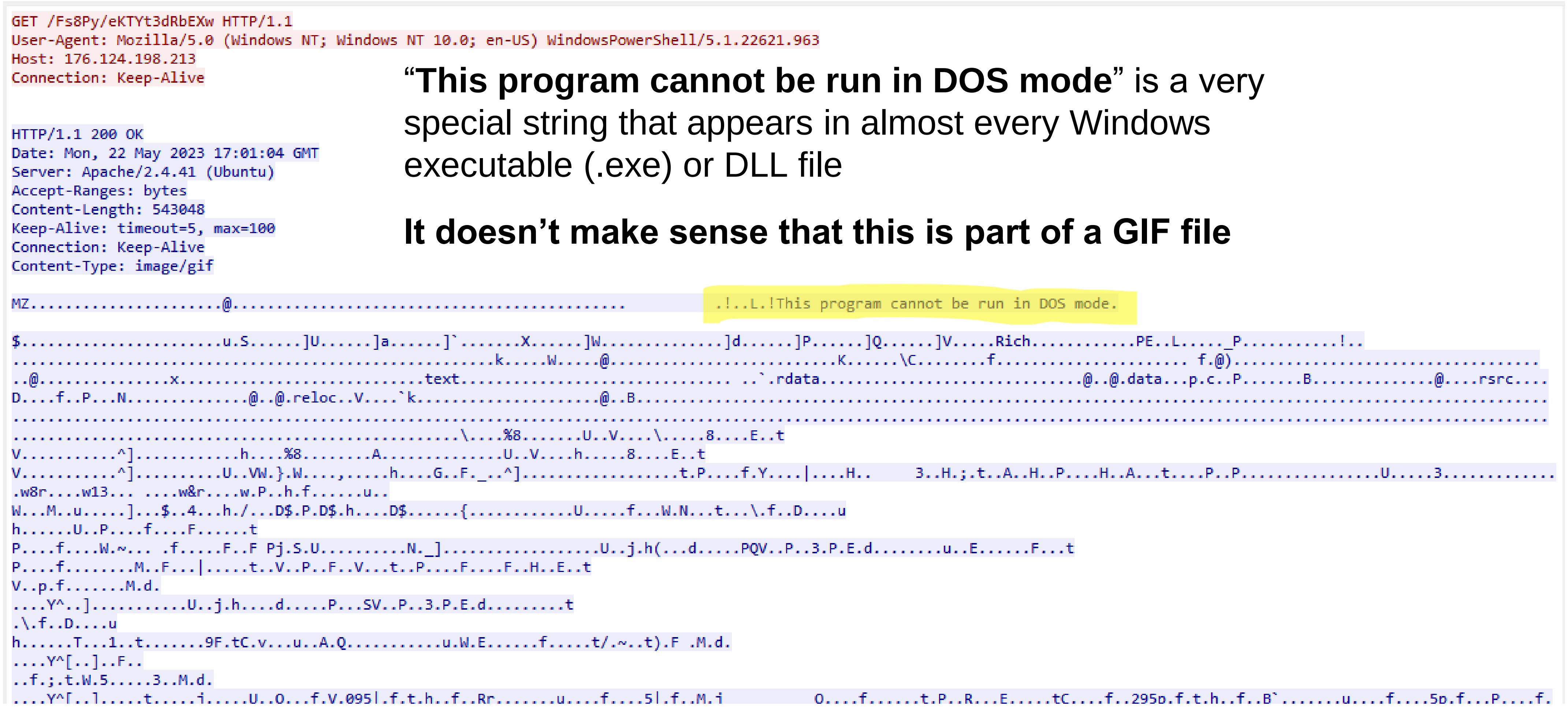
There is something *very* wrong here…

What is in this malicious HTTP request?



```
41.067893    10.13.13.99        HTTP    HTTP/1.1 200 OK  (image/gif)
```

Right click → Follow TCP Stream

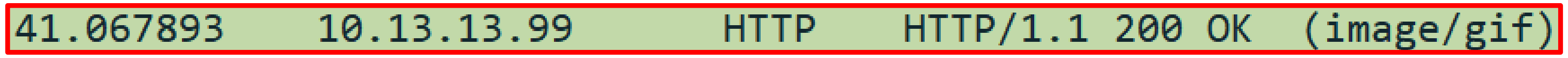


```
GET /Fs8Py/eKTYt3dRbEXw HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT; Windows NT 10.0; en-US) WindowsPowerShell/5.1.22621.963
Host: 176.124.198.213
Connection: Keep-Alive


HTTP/1.1 200 OK
Date: Mon, 22 May 2023 17:01:04 GMT
Server: Apache/2.4.41 (Ubuntu)
Accept-Ranges: bytes
Content-Length: 543048
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: image/gif

MZ.......................@.........................................    .!..L.!This program cannot be run in DOS mode.

$.......................u.S......]U......]a......]`.......X......]W..............]d......]P......]Q......]V.....Rich............PE..L....._P...........!..
.......................................................k.....W.....@........................K......\C........f....................... f.@)................................
..@...............x............................text............................. ..`.rdata.............................@..@.data...p.c..P.......B.............@....rsrc....
D....f..P...N..............@..@.reloc..V....`k.....................@..B................................................................................................
...........................................................................................................................................................................
..................................................\....%8.......U..V....\.....8....E..t
V...........^]............h....%8........A...............U..V....h.....8....E..t
V...........^]..........U..VW.}.W....,.....h....G..F._..^]...................t.P....f.Y....|....H..     3..H.;.t..A..H..P....H..A...t....P..P................U.....3.............
.w8r....w13...  ....w&r....w.P..h.f......u..
W...M..u.....]...$..4...h./...D$.P.D$.h....D$......{............U.....f...W.N...t...\.f..D....u
h......U..P....f....F......t
P....f....W.~...  .f.....F..F Pj.S.U..........N._]..................U..j.h(...d.....PQV..P..3.P.E.d........u..E......F...t
P....f.........M..F...|.....t..V..P..F..V...t..P....F....F..H..E..t
V..p.f.......M.d.
....Y^..]...........U..j.h....d.....P...SV..P..3.P.E.d.........t
.\.f..D....u
h......T...1..t.......9F.tC.v...u..A.Q...........u.W.E......f.....t/.~..t).F .M.d.
....Y^[..]..F..
..f.;.t.W.5.....3..M.d.
....Y^[..].....t.....i.....U..O...f.V.095|.f.t.h..f..Rr.......u....f....5|.f..M.i          O....f......t.P..R...E.....tC....f..295p.f.t.h..f..B`.......u....f....5p.f...P....f.
```

“**This program cannot be run in DOS mode**” is a very special string that appears in almost every Windows executable (.exe) or DLL file

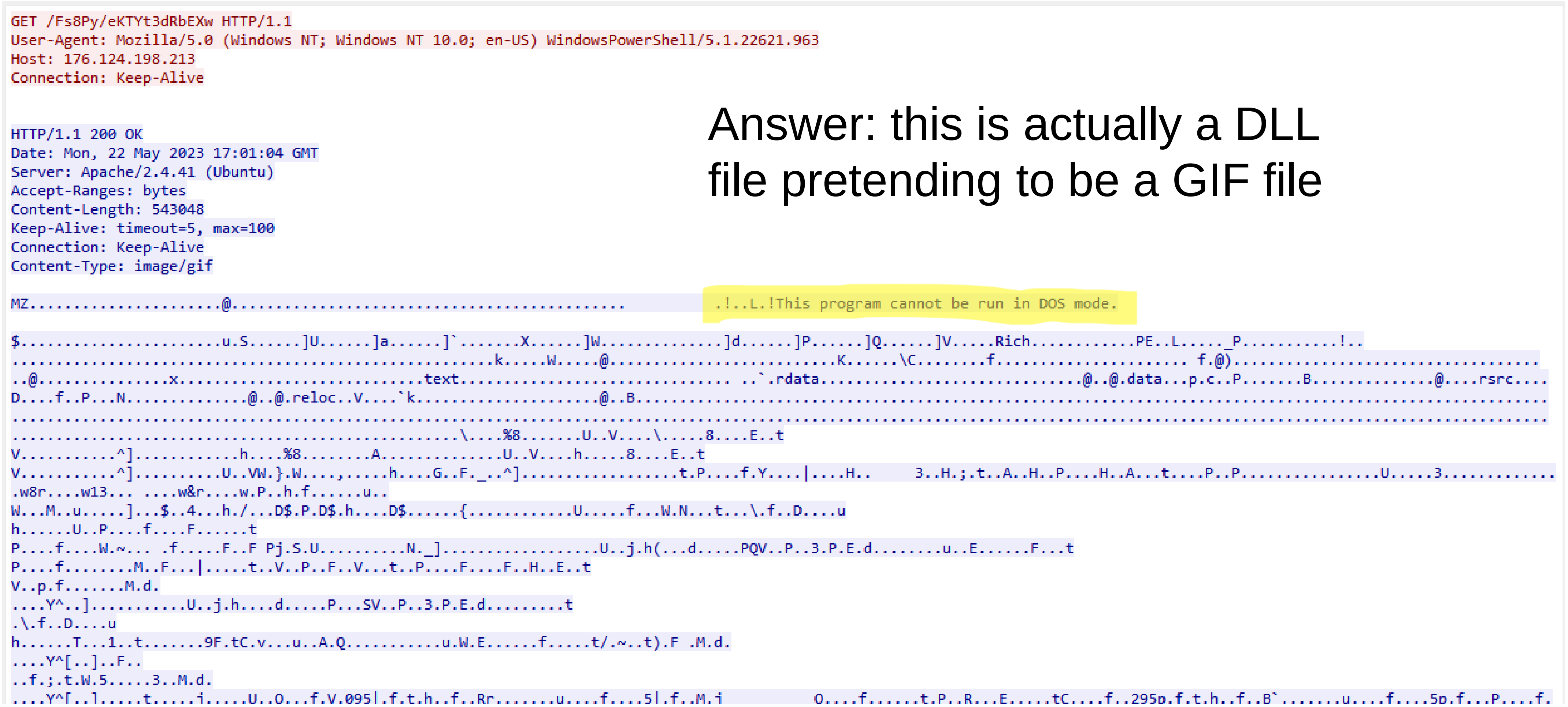
It doesn’t make sense that this is part of a GIF file

What is in this malicious HTTP request?



```
41.067893    10.13.13.99        HTTP    HTTP/1.1 200 OK   (image/gif)
```

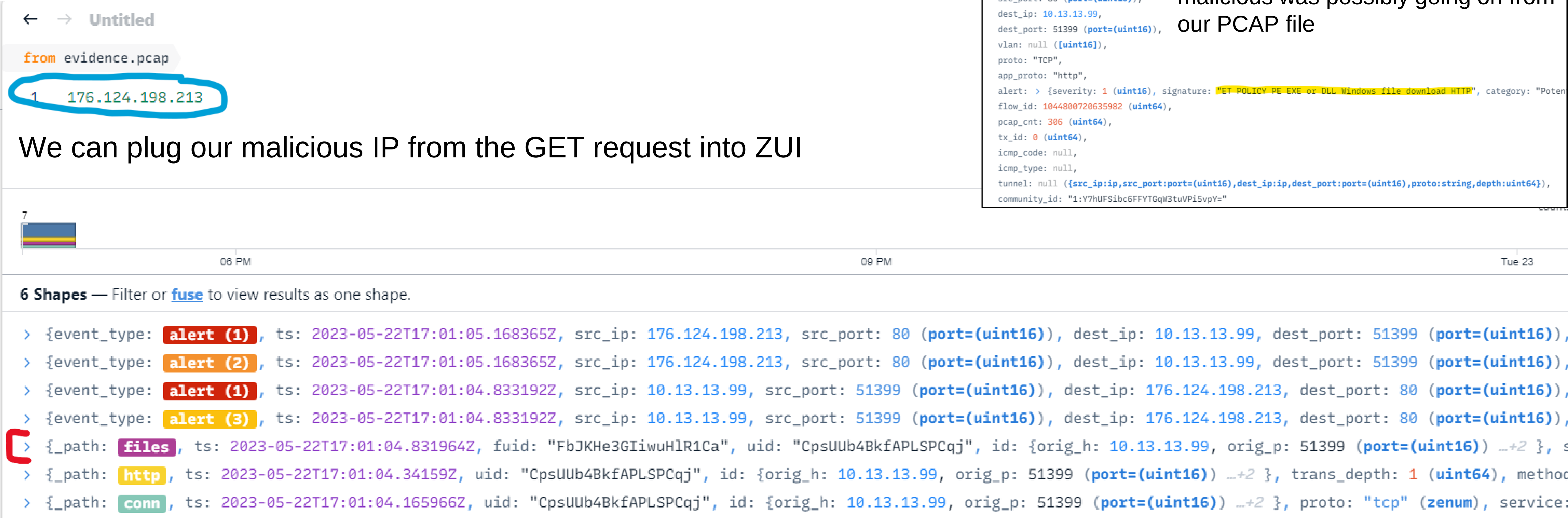
Right click → Follow TCP Stream



```
GET /Fs8Py/eKTYt3dRbEXw HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT; Windows NT 10.0; en-US) WindowsPowerShell/5.1.22621.963
Host: 176.124.198.213
Connection: Keep-Alive


HTTP/1.1 200 OK
Date: Mon, 22 May 2023 17:01:04 GMT
Server: Apache/2.4.41 (Ubuntu)
Accept-Ranges: bytes
Content-Length: 543048
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: image/gif

MZ.......................@..........................................    .!..L.!This program cannot be run in DOS mode.

$.......................u.S......]U......]a......]`.......X......]W..............]d......]P......]Q......]V.....Rich............PE..L....._P...........!..
......................................................k.....W.....@.........................K......\C........f.......................  f.@)................................
..@..............x...........................text.............................  ..`.rdata.............................@..@.data...p.c..P.......B.............@....rsrc....
D....f..P...N..............@..@.reloc..V....`k.....................@..B.................................................................................................
.....................................................................................................................................................................
...................................................\....%8.......U..V....\.....8....E..t
V...........^].............h....%8........A...............U..V....h.....8....E..t
V...........^]..........U..VW.}.W....,.....h....G..F._..^]..................t.P....f.Y....|....H..     3..H.;.t..A..H..P....H..A...t....P..P................U.....3............
.w8r....w13...  ....w&r....w.P..h.f......u..
W...M..u.....]...$..4...h./...D$.P.D$.h....D$......{............U.....f...W.N...t...\.f..D....u
h......U..P....f....F......t
P....f....W.~...  .f.....F..F Pj.S.U..........N._]..................U..j.h(...d.....PQV..P..3.P.E.d........u..E......F...t
P....f.........M..F...|.....t..V..P..F..V...t..P....F....F..H..E..t
V..p.f.......M.d.
....Y^..]...........U..j.h....d.....P...SV..P..3.P.E.d.........t
.\.f..D....u
h......T...1..t.......9F.tC.v...u..A.Q...........u.W.E......f.....t/.~..t).F .M.d.
....Y^[..]..F..
..f.;.t.W.5.....3..M.d.
....Y^[..].....t.....i.....U..O...f.V.095|.f.t.h..f..Rr.......u....f....5|.f..M.i          O....f......t.P..R...E.....tC....f..295p.f.t.h..f..B`.......u....f....5p.f...P....f.
```

Answer: this is actually a DLL file pretending to be a GIF file

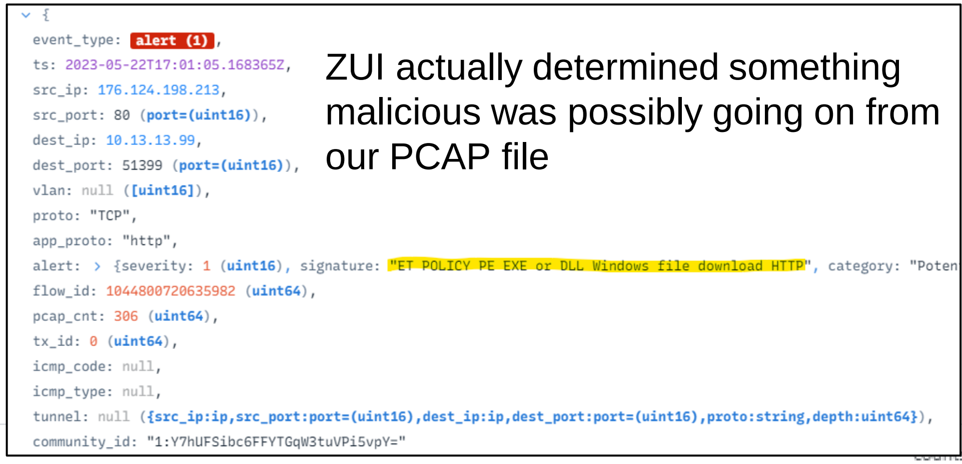
What is this malicious DLL file?



```
←  →  Untitled

from evidence.pcap

1    176.124.198.213
```

We can plug our malicious IP from the GET request into ZUI

```
7

        06 PM                                  09 PM                                  Tue 23

6 Shapes — Filter or fuse to view results as one shape.

> {event_type: alert (1), ts: 2023-05-22T17:01:05.168365Z, src_ip: 176.124.198.213, src_port: 80 (port=(uint16)), dest_ip: 10.13.13.99, dest_port: 51399 (port=(uint16)),
> {event_type: alert (2), ts: 2023-05-22T17:01:05.168365Z, src_ip: 176.124.198.213, src_port: 80 (port=(uint16)), dest_ip: 10.13.13.99, dest_port: 51399 (port=(uint16)),
> {event_type: alert (1), ts: 2023-05-22T17:01:04.833192Z, src_ip: 10.13.13.99, src_port: 51399 (port=(uint16)), dest_ip: 176.124.198.213, dest_port: 80 (port=(uint16)),
> {event_type: alert (3), ts: 2023-05-22T17:01:04.833192Z, src_ip: 10.13.13.99, src_port: 51399 (port=(uint16)), dest_ip: 176.124.198.213, dest_port: 80 (port=(uint16)),
> {_path: files, ts: 2023-05-22T17:01:04.831964Z, fuid: "FbJKHe3GIiwuHlR1Ca", uid: "CpsUUb4BkfAPLSPCqj", id: {orig_h: 10.13.13.99, orig_p: 51399 (port=(uint16)) …+2 }, s
> {_path: http, ts: 2023-05-22T17:01:04.34159Z, uid: "CpsUUb4BkfAPLSPCqj", id: {orig_h: 10.13.13.99, orig_p: 51399 (port=(uint16)) …+2 }, trans_depth: 1 (uint64), method
> {_path: conn, ts: 2023-05-22T17:01:04.165966Z, uid: "CpsUUb4BkfAPLSPCqj", id: {orig_h: 10.13.13.99, orig_p: 51399 (port=(uint16)) …+2 }, proto: "tcp" (zenum), service:
```

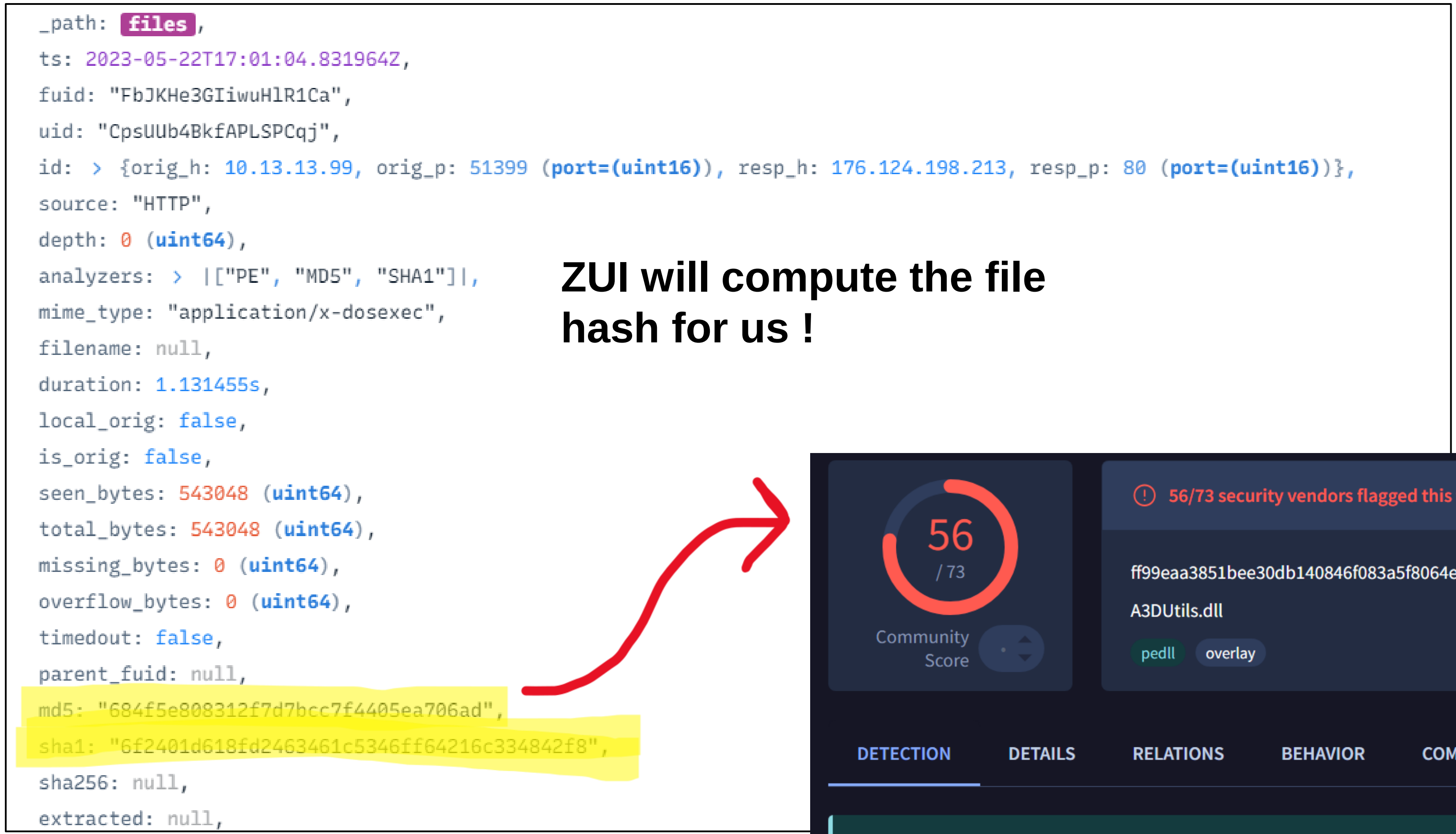


```
⌄ {
  event_type: alert (1),
  ts: 2023-05-22T17:01:05.168365Z,
  src_ip: 176.124.198.213,
  src_port: 80 (port=(uint16)),
  dest_ip: 10.13.13.99,
  dest_port: 51399 (port=(uint16)),
  vlan: null ([uint16]),
  proto: "TCP",
  app_proto: "http",
  alert: > {severity: 1 (uint16), signature: "ET POLICY PE EXE or DLL Windows file download HTTP", category: "Poten
  flow_id: 1044800720635982 (uint64),
  pcap_cnt: 306 (uint64),
  tx_id: 0 (uint64),
  icmp_code: null,
  icmp_type: null,
  tunnel: null ({src_ip:ip,src_port:port=(uint16),dest_ip:ip,dest_port:port=(uint16),proto:string,depth:uint64}),
  community_id: "1:Y7hUFSibc6FFYTGqW3tuVPi5vpY="
```

ZUI actually determined something malicious was possibly going on from our PCAP file

We can expand the “files” info to see information about this file that was downloaded

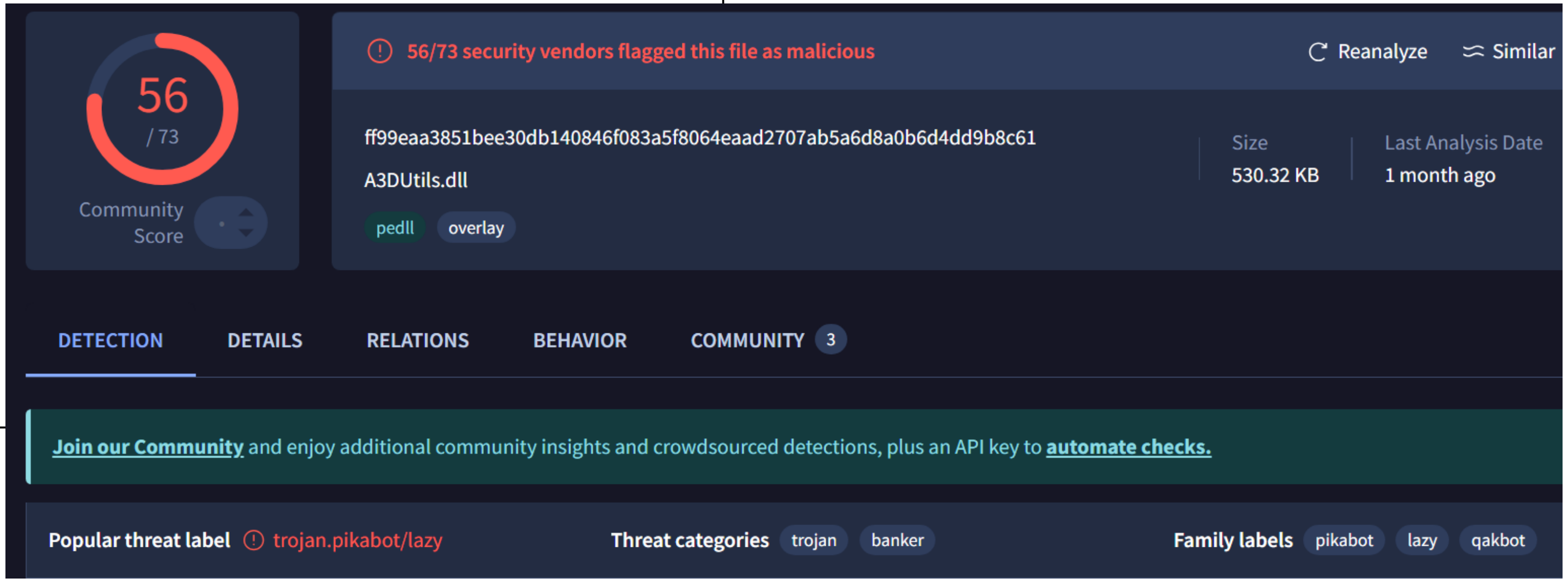
What is this malicious DLL file?



```
_path: files,
ts: 2023-05-22T17:01:04.831964Z,
fuid: "FbJKHe3GIiwuHlR1Ca",
uid: "CpsUUb4BkfAPLSPCqj",
id: > {orig_h: 10.13.13.99, orig_p: 51399 (port=(uint16)), resp_h: 176.124.198.213, resp_p: 80 (port=(uint16))},
source: "HTTP",
depth: 0 (uint64),
analyzers: > |["PE", "MD5", "SHA1"]|,
mime_type: "application/x-dosexec",
filename: null,
duration: 1.131455s,
local_orig: false,
is_orig: false,
seen_bytes: 543048 (uint64),
total_bytes: 543048 (uint64),
missing_bytes: 0 (uint64),
overflow_bytes: 0 (uint64),
timedout: false,
parent_fuid: null,
md5: "684f5e808312f7d7bcc7f4405ea706ad",
sha1: "6f2401d618fd2463461c5346ff64216c334842f8",
sha256: null,
extracted: null,
```

ZUI will compute the file hash for us !

By plugging it into VirusTotal, we can see that this DLL file-pretending-to-be-a-gif is a known malicious file!



56 / 73

Community Score

56/73 security vendors flagged this file as malicious

Reanalyze | Similar

ff99eaa3851bee30db140846f083a5f8064eaad2707ab5a6d8a0b6d4dd9b8c61

A3DUtils.dll

pedll | overlay

Size: 530.32 KB

Last Analysis Date: 1 month ago

DETECTION | DETAILS | RELATIONS | BEHAVIOR | COMMUNITY 3

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: trojan.pikabot/lazy

Threat categories: trojan, banker

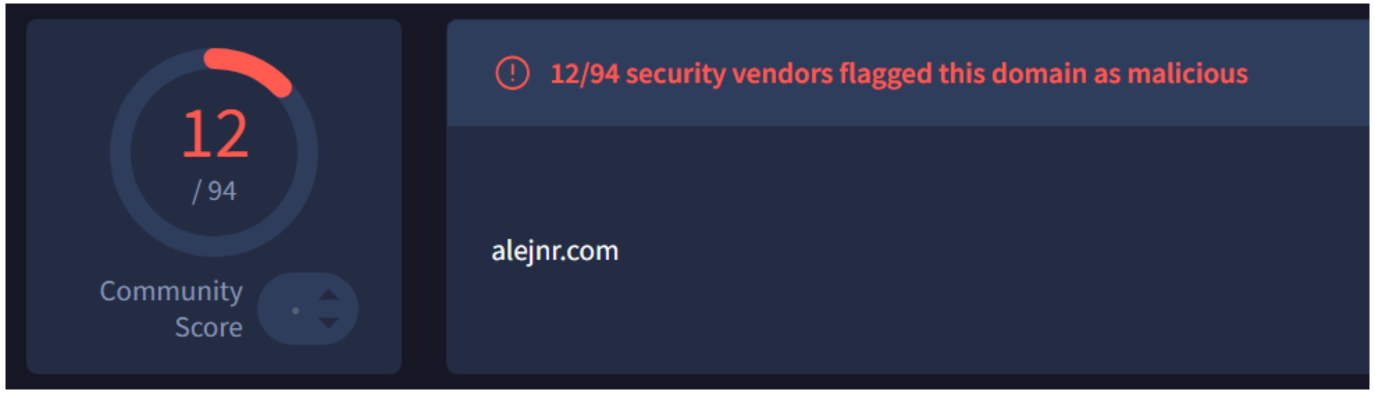
Family labels: pikabot, lazy, qakbot

The name of this malware is "PIKABOT"

VT seemed to redirect to the SHA256 hash value

What is this malicious DLL file?

This domain had been flagged as malicious in the past

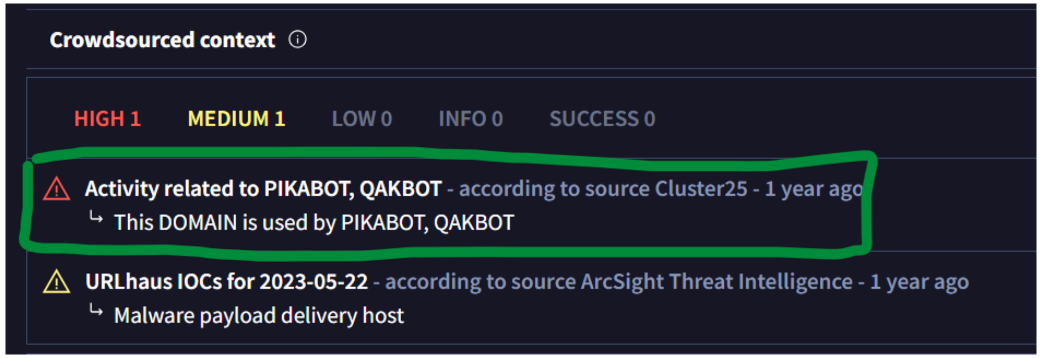


12 / 94 Community Score

12/94 security vendors flagged this domain as malicious

alejnr.com

VirusTotal also provides context as to why it was flagged as malicious



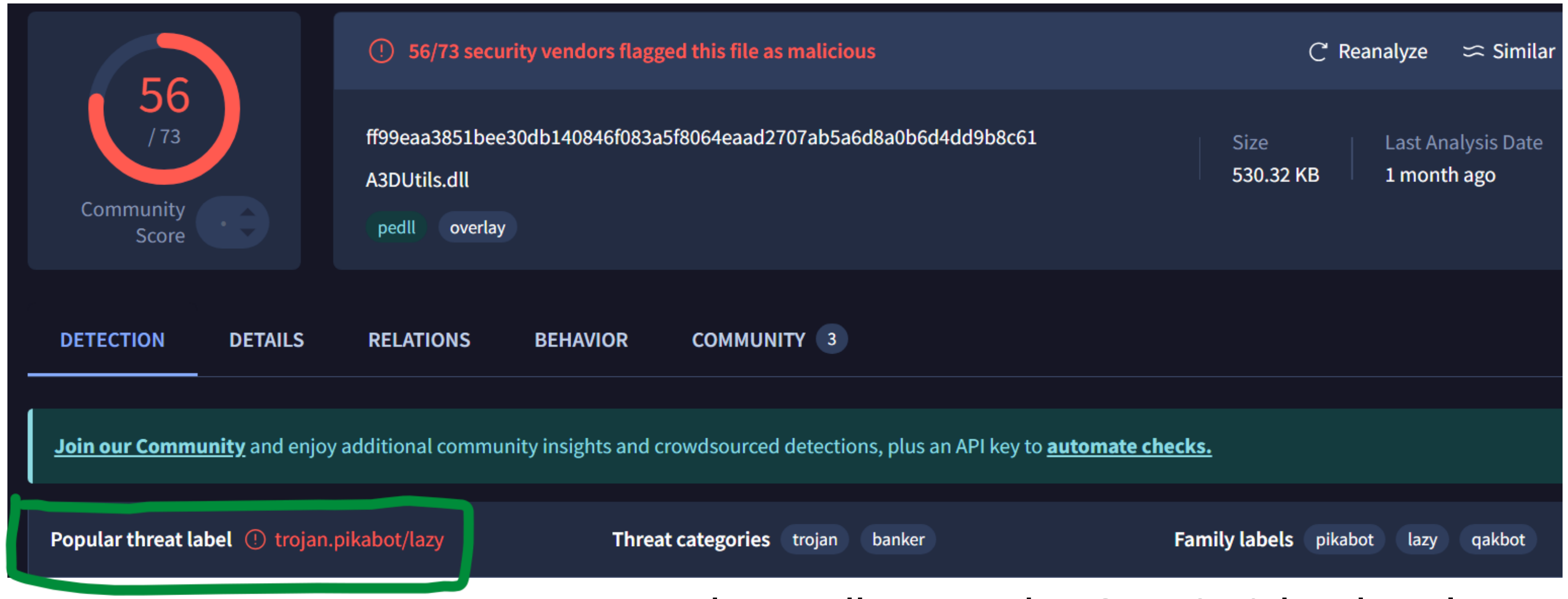
Crowdsourced context

HIGH 1 MEDIUM 1 LOW 0 INFO 0 SUCCESS 0

Activity related to PIKABOT, QAKBOT - according to source Cluster25 - 1 year ago
↳ This DOMAIN is used by PIKABOT, QAKBOT

URLhaus IOCs for 2023-05-22 - according to source ArcSight Threat Intelligence - 1 year ago
↳ Malware payload delivery host

We saw “PIKABOT” earlier when we were investigating **alejnr.com**



56 / 73 Community Score

56/73 security vendors flagged this file as malicious

Reanalyze Similar

ff99eaa3851bee30db140846f083a5f8064eaad2707ab5a6d8a0b6d4dd9b8c61

A3DUtils.dll

pedll overlay

Size	Last Analysis Date
530.32 KB	1 month ago

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 3

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label trojan.pikabot/lazy

Threat categories trojan banker

Family labels pikabot lazy qakbot

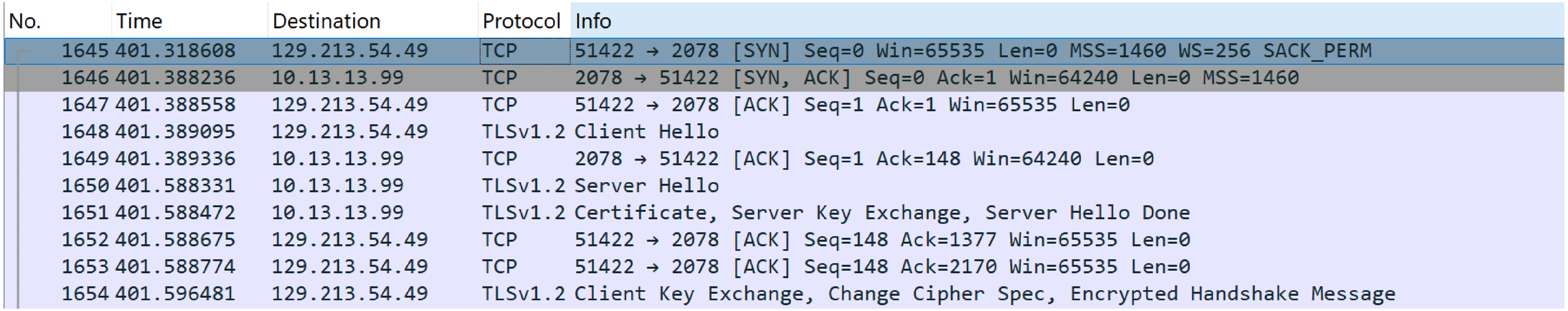
The name of this malware is “PIKABOT”

VT seemed to redirect to the SHA256 hash value

What happens after the malware was downloaded?

Let’s see if we can find evidence of **command and control** from this cyber attack

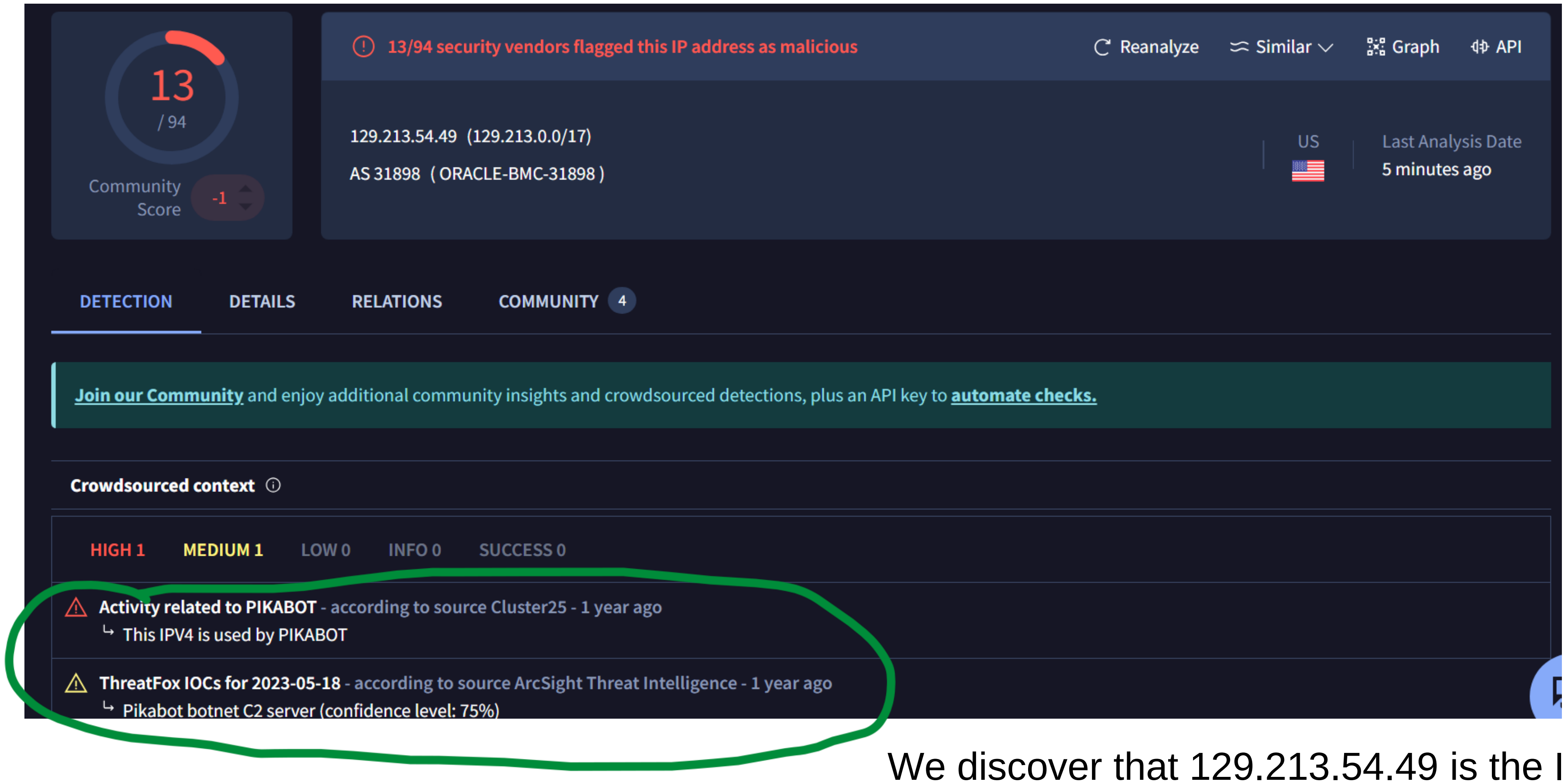
We can start plugging in IP addresses into VirusTotal to see if we get a hit



No.	Time	Destination	Protocol	Info
1645	401.318608	129.213.54.49	TCP	51422 → 2078 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
1646	401.388236	10.13.13.99	TCP	2078 → 51422 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
1647	401.388558	129.213.54.49	TCP	51422 → 2078 [ACK] Seq=1 Ack=1 Win=65535 Len=0
1648	401.389095	129.213.54.49	TLSv1.2	Client Hello
1649	401.389336	10.13.13.99	TCP	2078 → 51422 [ACK] Seq=1 Ack=148 Win=64240 Len=0
1650	401.588331	10.13.13.99	TLSv1.2	Server Hello
1651	401.588472	10.13.13.99	TLSv1.2	Certificate, Server Key Exchange, Server Hello Done
1652	401.588675	129.213.54.49	TCP	51422 → 2078 [ACK] Seq=148 Ack=1377 Win=65535 Len=0
1653	401.588774	129.213.54.49	TCP	51422 → 2078 [ACK] Seq=148 Ack=2170 Win=65535 Len=0
1654	401.596481	129.213.54.49	TLSv1.2	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message

What happens after the malware was downloaded?

Let's see if we can find evidence of **command and control** from this cyber attack



13/94 security vendors flagged this IP address as malicious

129.213.54.49 (129.213.0.0/17)
AS 31898 (ORACLE-BMC-31898)

Last Analysis Date
5 minutes ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Crowdsourced context

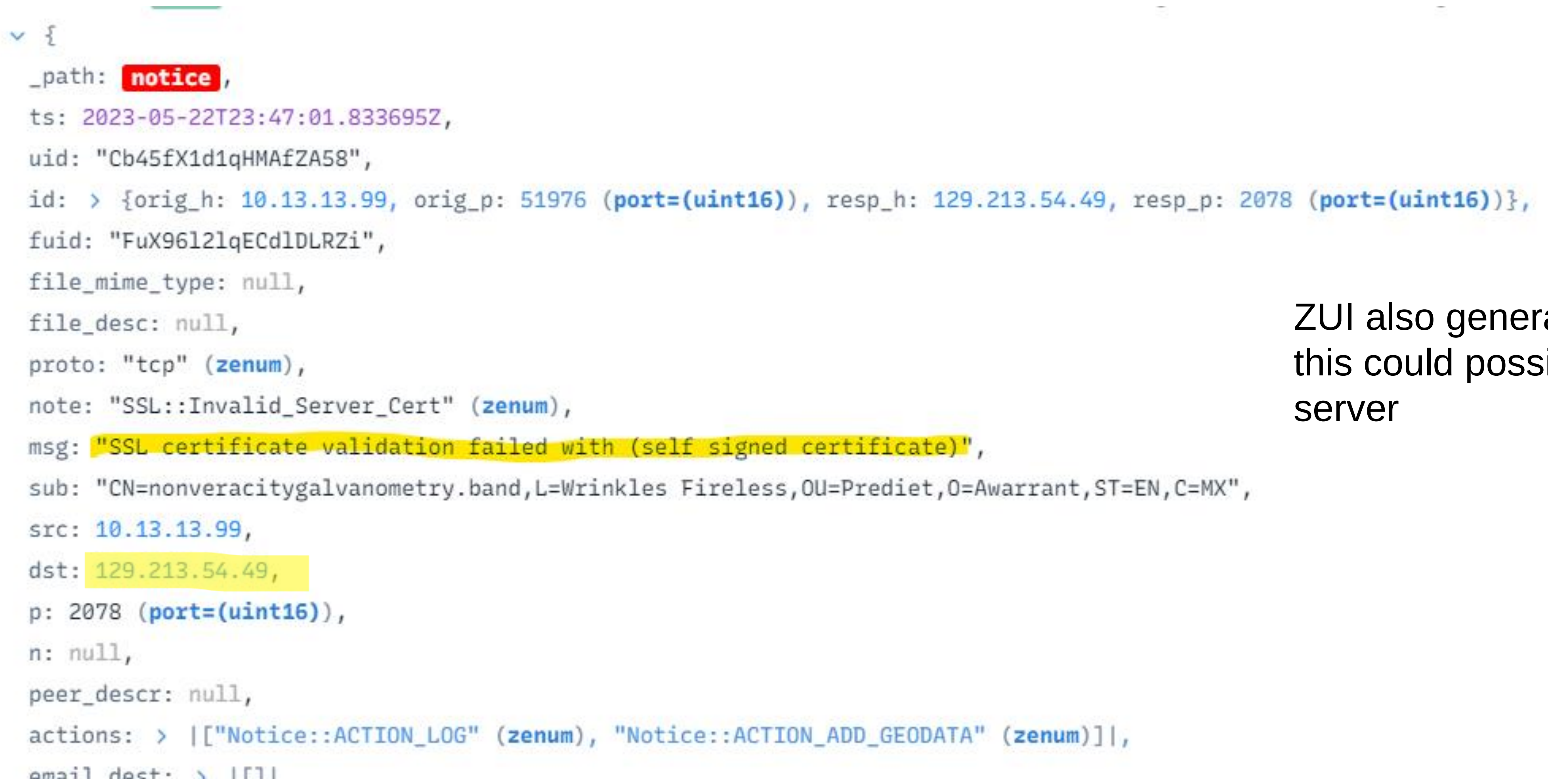
Activity related to PIKABOT - according to source Cluster25 - 1 year ago
↳ This IPV4 is used by PIKABOT

ThreatFox IOCs for 2023-05-18 - according to source ArcSight Threat Intelligence - 1 year ago
↳ Pikabot botnet C2 server (confidence level: 75%)

We discover that 129.213.54.49 is the IP address for the PIKABOT C2 server

What happens after the malware was downloaded?

Let’s see if we can find evidence of **command and control** from this cyber attack

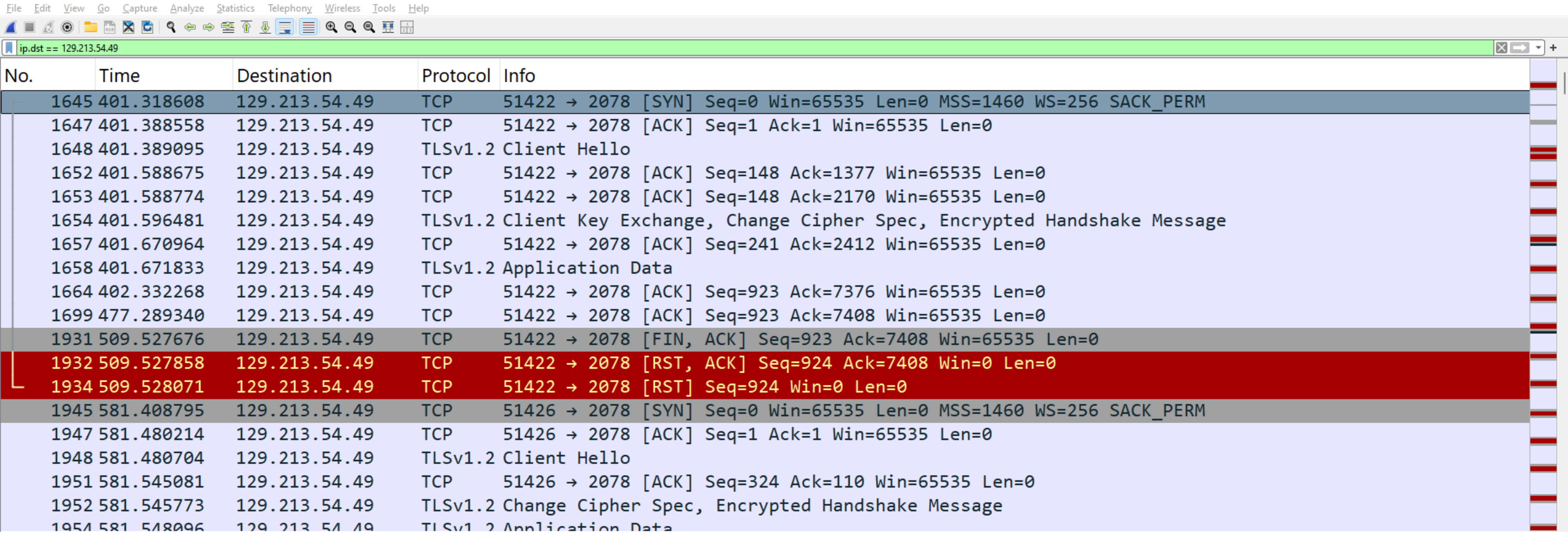


```
{
  _path: notice,
  ts: 2023-05-22T23:47:01.833695Z,
  uid: "Cb45fX1d1qHMAfZA58",
  id: > {orig_h: 10.13.13.99, orig_p: 51976 (port=(uint16)), resp_h: 129.213.54.49, resp_p: 2078 (port=(uint16))},
  fuid: "FuX96l2lqECdlDLRZi",
  file_mime_type: null,
  file_desc: null,
  proto: "tcp" (zenum),
  note: "SSL::Invalid_Server_Cert" (zenum),
  msg: "SSL certificate validation failed with (self signed certificate)",
  sub: "CN=nonveracitygalvanometry.band,L=Wrinkles Fireless,OU=Prediet,O=Awarrant,ST=EN,C=MX",
  src: 10.13.13.99,
  dst: 129.213.54.49,
  p: 2078 (port=(uint16)),
  n: null,
  peer_descr: null,
  actions: > |["Notice::ACTION_LOG" (zenum), "Notice::ACTION_ADD_GEODATA" (zenum)]|,
```

ZUI also generated a notice that this could possibly be a C2 server

What damage was done?

A good amount of Data was exchanged with the C2 server



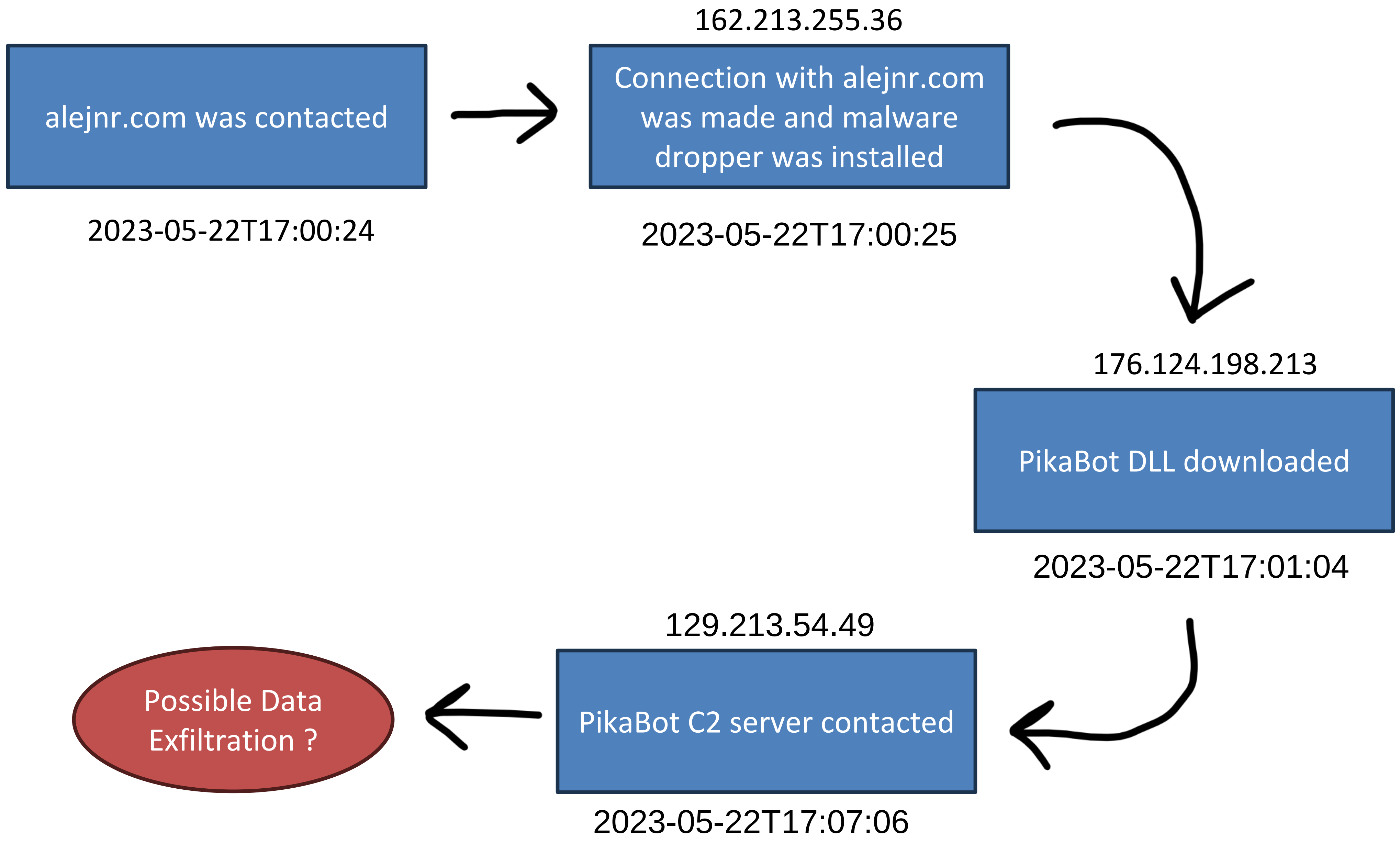
ip.dst == 129.213.54.49

No.	Time	Destination	Protocol	Info
1645	401.318608	129.213.54.49	TCP	51422 → 2078 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
1647	401.388558	129.213.54.49	TCP	51422 → 2078 [ACK] Seq=1 Ack=1 Win=65535 Len=0
1648	401.389095	129.213.54.49	TLSv1.2	Client Hello
1652	401.588675	129.213.54.49	TCP	51422 → 2078 [ACK] Seq=148 Ack=1377 Win=65535 Len=0
1653	401.588774	129.213.54.49	TCP	51422 → 2078 [ACK] Seq=148 Ack=2170 Win=65535 Len=0
1654	401.596481	129.213.54.49	TLSv1.2	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
1657	401.670964	129.213.54.49	TCP	51422 → 2078 [ACK] Seq=241 Ack=2412 Win=65535 Len=0
1658	401.671833	129.213.54.49	TLSv1.2	Application Data
1664	402.332268	129.213.54.49	TCP	51422 → 2078 [ACK] Seq=923 Ack=7376 Win=65535 Len=0
1699	477.289340	129.213.54.49	TCP	51422 → 2078 [ACK] Seq=923 Ack=7408 Win=65535 Len=0
1931	509.527676	129.213.54.49	TCP	51422 → 2078 [FIN, ACK] Seq=923 Ack=7408 Win=65535 Len=0
1932	509.527858	129.213.54.49	TCP	51422 → 2078 [RST, ACK] Seq=924 Ack=7408 Win=0 Len=0
1934	509.528071	129.213.54.49	TCP	51422 → 2078 [RST] Seq=924 Win=0 Len=0
1945	581.408795	129.213.54.49	TCP	51426 → 2078 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
1947	581.480214	129.213.54.49	TCP	51426 → 2078 [ACK] Seq=1 Ack=1 Win=65535 Len=0
1948	581.480704	129.213.54.49	TLSv1.2	Client Hello
1951	581.545081	129.213.54.49	TCP	51426 → 2078 [ACK] Seq=324 Ack=110 Win=65535 Len=0
1952	581.545773	129.213.54.49	TLSv1.2	Change Cipher Spec, Encrypted Handshake Message
1954	581.548096	129.213.54.49	TLSv1.2	Application Data

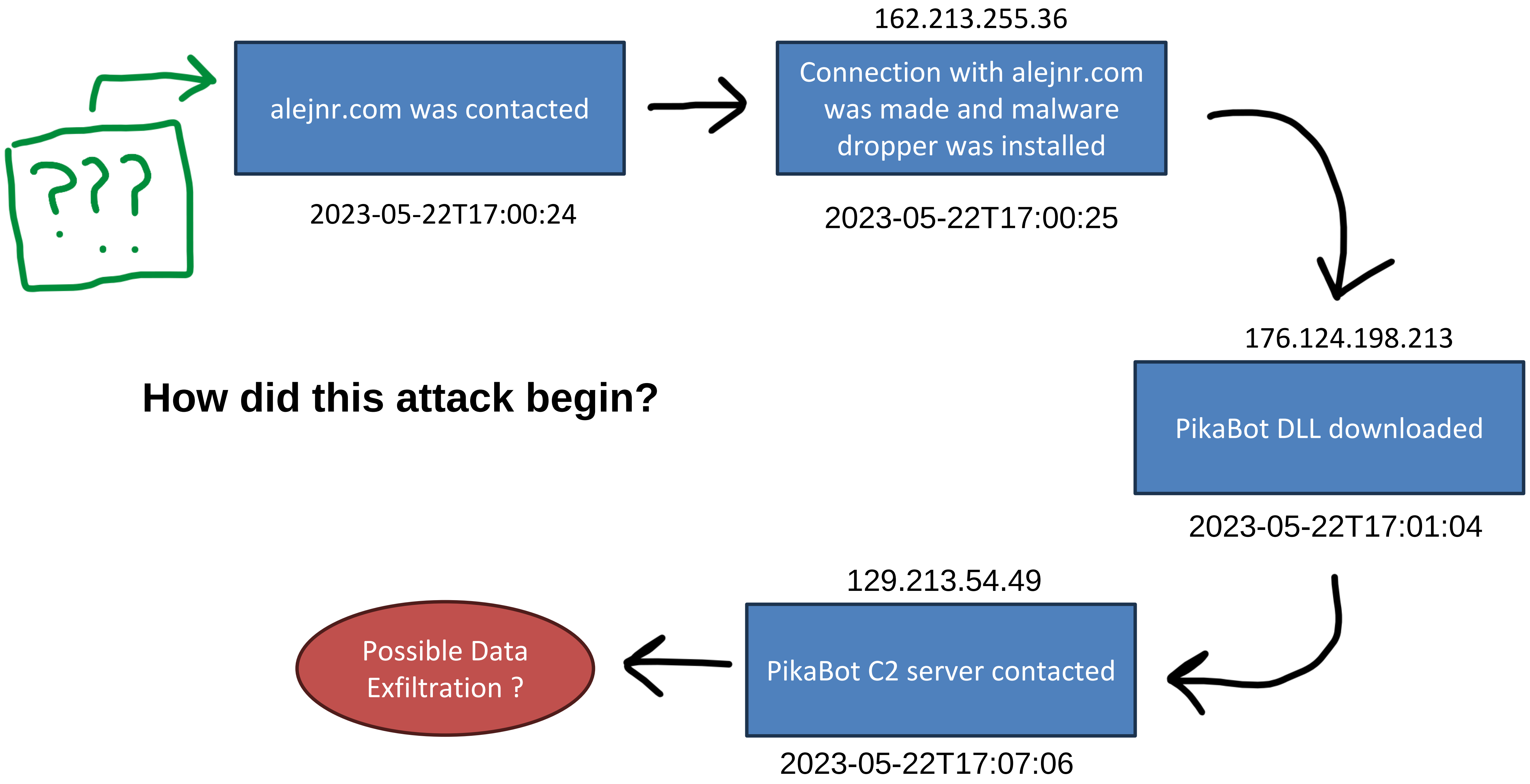
Since this all HTTPS encrypted traffic, we cannot see the content of these messages

However, this is evidence that it is possible that data may have been exfiltrated

Timeline


162.213.255.36
alejnr.com was contacted
Connection with alejnr.com was made and malware dropper was installed
2023-05-22T17:00:24
2023-05-22T17:00:25
176.124.198.213
PikaBot DLL downloaded
2023-05-22T17:01:04
129.213.54.49
Possible Data Exfiltration ?
PikaBot C2 server contacted
2023-05-22T17:07:06

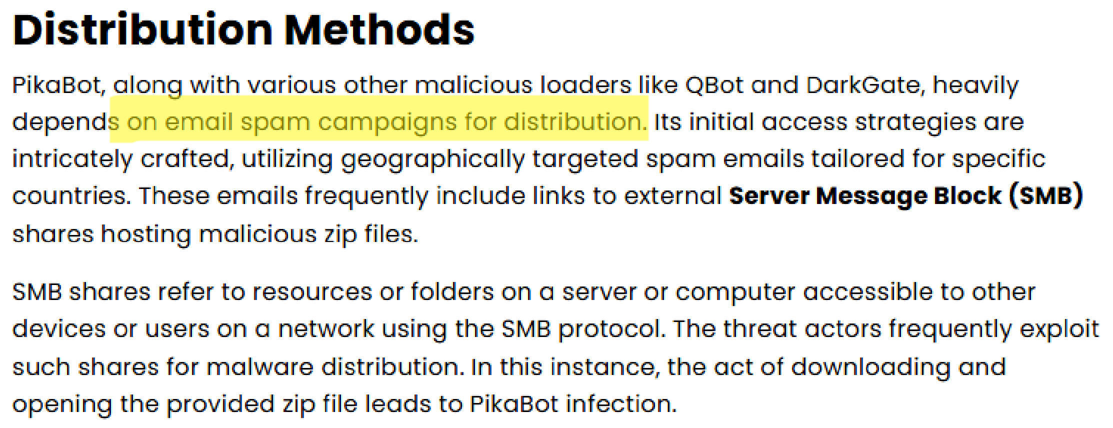
Timeline



How did this attack begin?

Finding Root cause

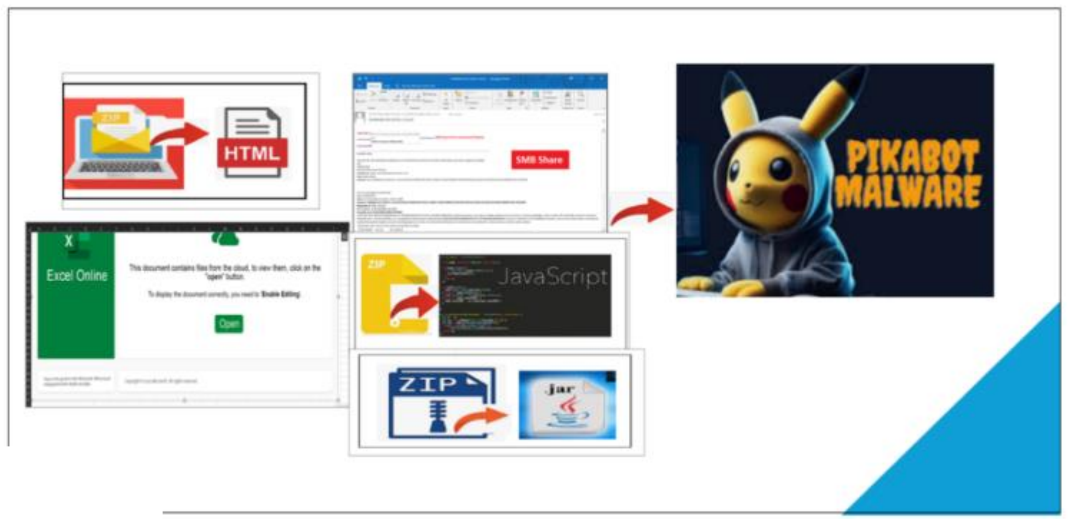
PIKABOT is a well-documented malware. We can do a little bit of investigation ourselves to figure out what happened

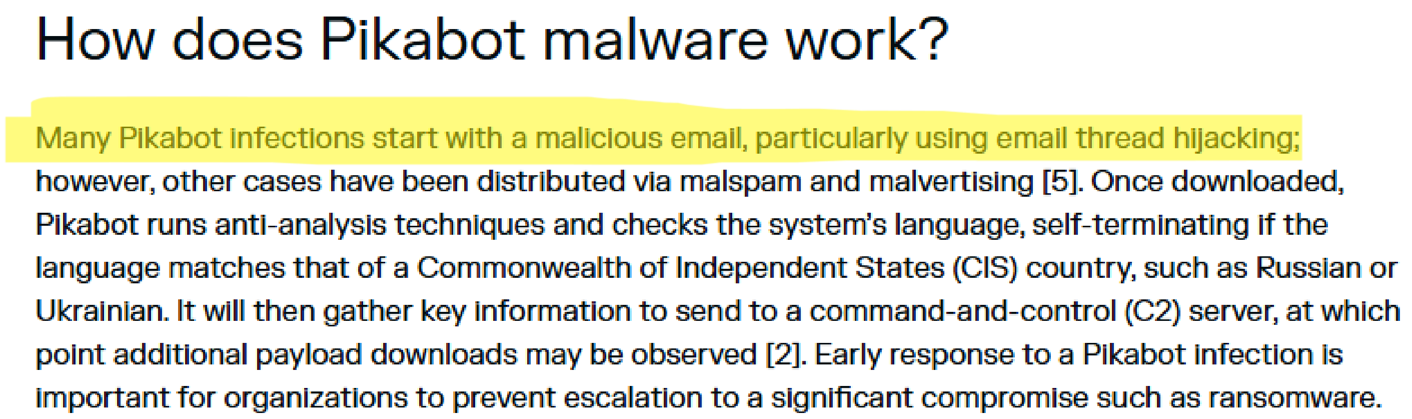


Distribution Methods

PikaBot, along with various other malicious loaders like QBot and DarkGate, heavily depends on email spam campaigns for distribution. Its initial access strategies are intricately crafted, utilizing geographically targeted spam emails tailored for specific countries. These emails frequently include links to external **Server Message Block (SMB)** shares hosting malicious zip files.

SMB shares refer to resources or folders on a server or computer accessible to other devices or users on a network using the SMB protocol. The threat actors frequently exploit such shares for malware distribution. In this instance, the act of downloading and opening the provided zip file leads to PikaBot infection.



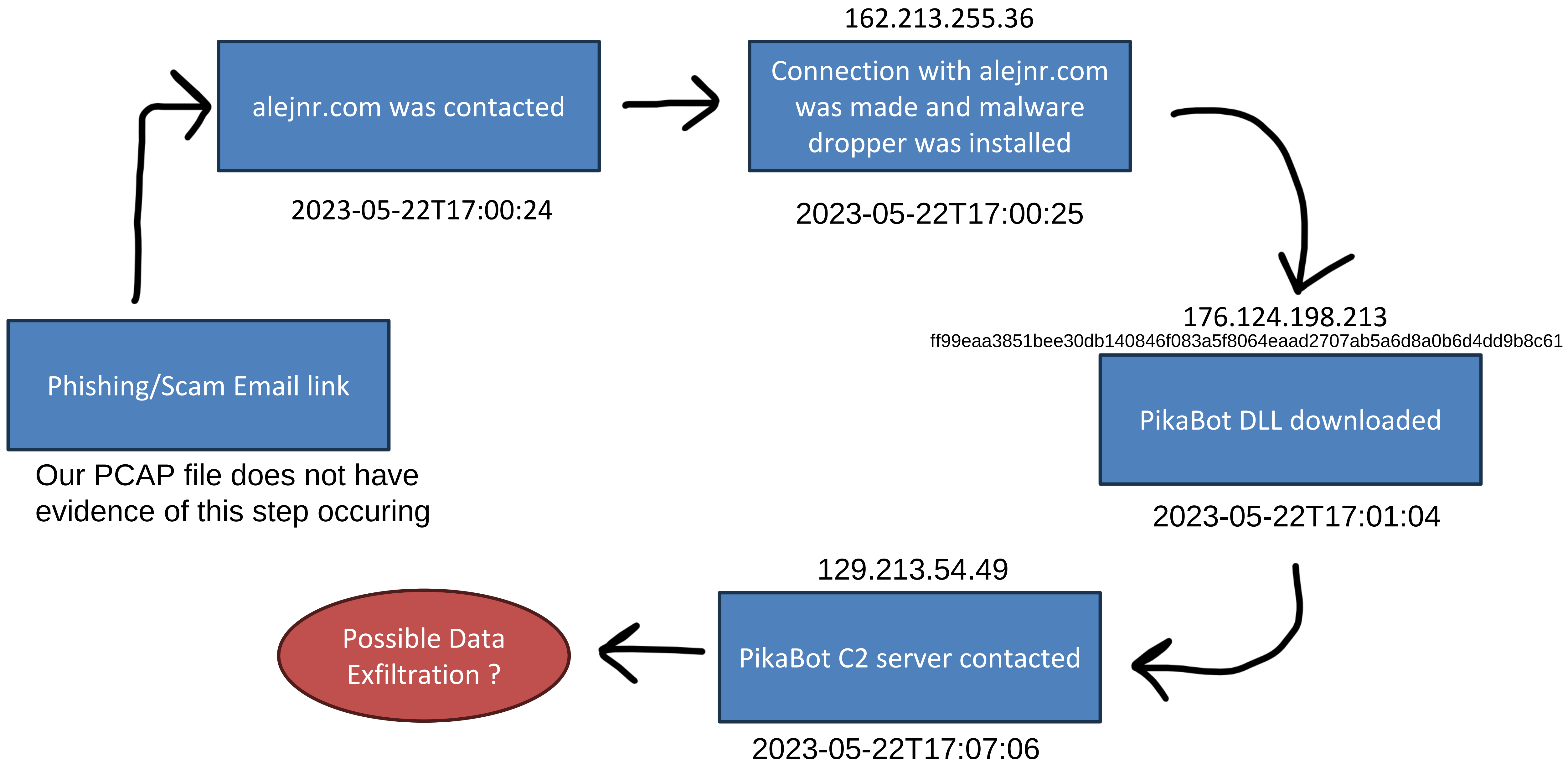


How does Pikabot malware work?

Many Pikabot infections start with a malicious email, particularly using email thread hijacking; however, other cases have been distributed via malspam and malvertising [5]. Once downloaded, Pikabot runs anti-analysis techniques and checks the system's language, self-terminating if the language matches that of a Commonwealth of Independent States (CIS) country, such as Russian or Ukrainian. It will then gather key information to send to a command-and-control (C2) server, at which point additional payload downloads may be observed [2]. Early response to a Pikabot infection is important for organizations to prevent escalation to a significant compromise such as ransomware.

Full IOC: https://github.com/pan-unit42/tweets/blob/master/2023-05-22-IOCs-for-Pikabot-infection-with-Cobalt-Strike.txt

Timeline


Phishing/Scam Email link
Our PCAP file does not have evidence of this step occuring
alejnr.com was contacted
2023-05-22T17:00:24
162.213.255.36
Connection with alejnr.com was made and malware dropper was installed
2023-05-22T17:00:25
176.124.198.213
ff99eaa3851bee30db140846f083a5f8064eaad2707ab5a6d8a0b6d4dd9b8c61
PikaBot DLL downloaded
2023-05-22T17:01:04
129.213.54.49
PikaBot C2 server contacted
2023-05-22T17:07:06
Possible Data Exfiltration ?