Decrypting AzoRult traffic for fun and profit

mariohenkel.medium.com/decrypting-azorult-traffic-for-fun-and-profit-9f28d8638b05

Mario Henkel

February 6, 2021

Mario Henkel

Feb 6, 2021

.

5 min read

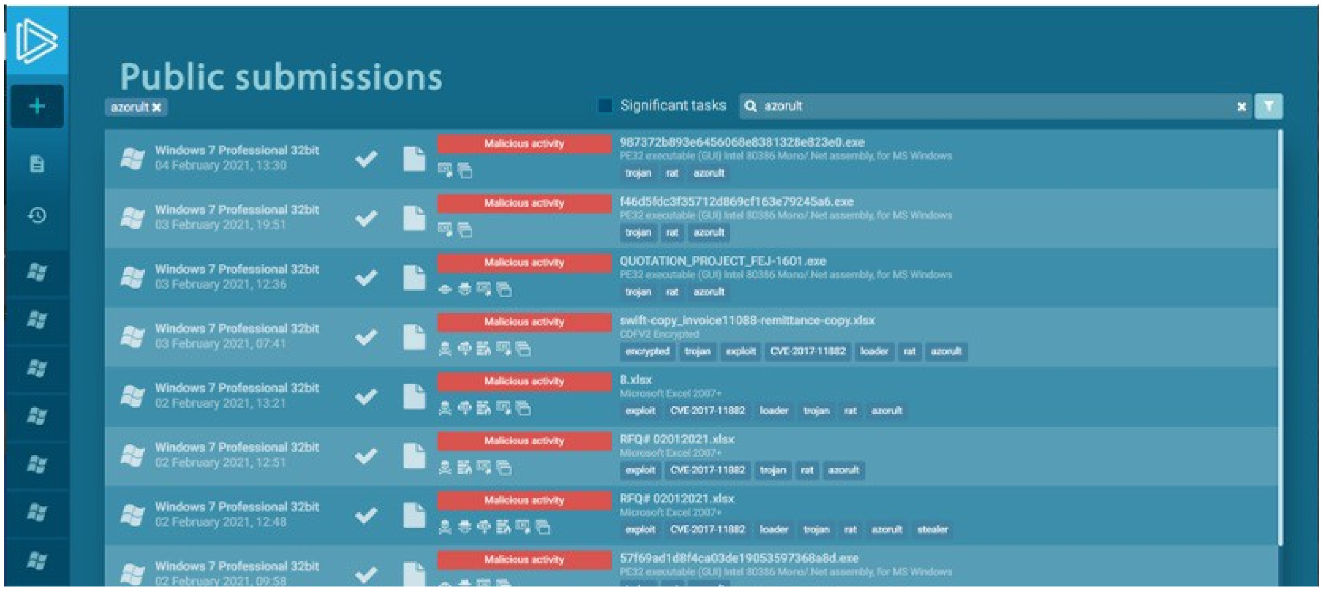
There will be times in your career when you will be presented with a traffic capture and get the task to determine what happened and if any data was stolen.

In this post, I will show you how you can squeeze all those juicy information from a PCAP traffic capture from an Azorult infection.

At the end, you will be able to answer which data has been stolen so you can act accordingly. Let's start!

Getting sample data

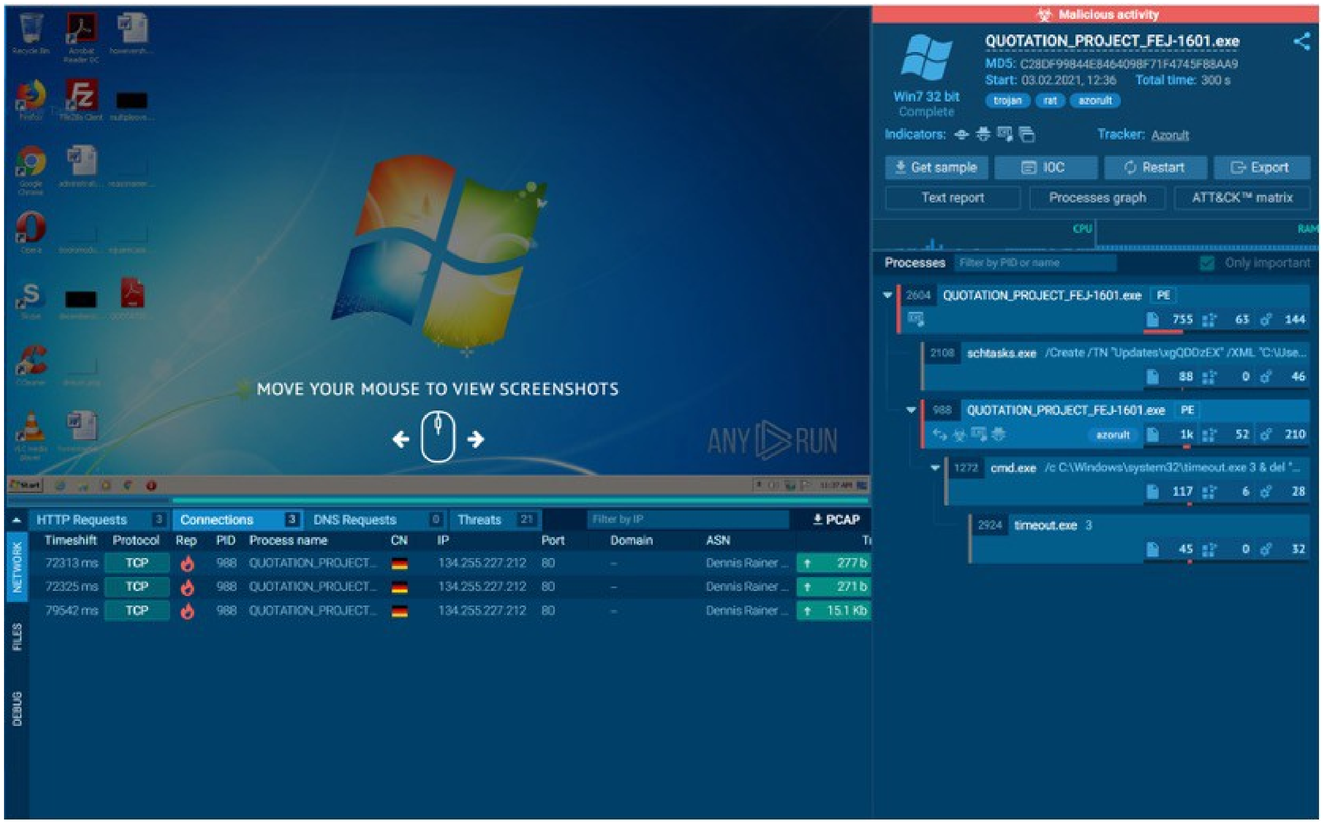
Head over to https://any.run and search for "Azorult" in public submissions or use the PCAP you already got



Windows 7 Professional 32bit 04 February 2021, 13:30	✓	Malicious activity	987372b893e6456068e8381328e823e0.exe PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows	trojan rat azorult
Windows 7 Professional 32bit 03 February 2021, 19:51	✓	Malicious activity	f46d5fdc3f35712d869cf163e79245a6.exe PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows	trojan rat azorult
Windows 7 Professional 32bit 03 February 2021, 12:36	✓	Malicious activity	QUOTATION_PROJECT_FEJ-1601.exe PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows	trojan rat azorult
Windows 7 Professional 32bit 03 February 2021, 07:41	✓	Malicious activity	swift-copy_invoice11088-remittance-copy.xlsx CDFV2 Encrypted	encrypted trojan exploit CVE-2017-11882 loader rat azorult
Windows 7 Professional 32bit 02 February 2021, 13:21	✓	Malicious activity	8.xlsx Microsoft Excel 2007+	exploit CVE-2017-11882 loader trojan rat azorult
Windows 7 Professional 32bit 02 February 2021, 12:51	✓	Malicious activity	RFQ# 02012021.xlsx Microsoft Excel 2007+	exploit CVE-2017-11882 trojan rat azorult
Windows 7 Professional 32bit 02 February 2021, 12:48	✓	Malicious activity	RFQ# 02012021.xlsx Microsoft Excel 2007+	exploit CVE-2017-11882 loader trojan rat azorult stealer
Windows 7 Professional 32bit 02 February 2021, 09:58	✓	Malicious activity	57f69ad1d8f4ca03de19053597368a8d.exe PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows	

Most likely you will find a lot of samples

You will find a lot of samples without actual network traffic since the command and control server was already offline when any.run analyzed the sample. Have a look at samples which show POST requests



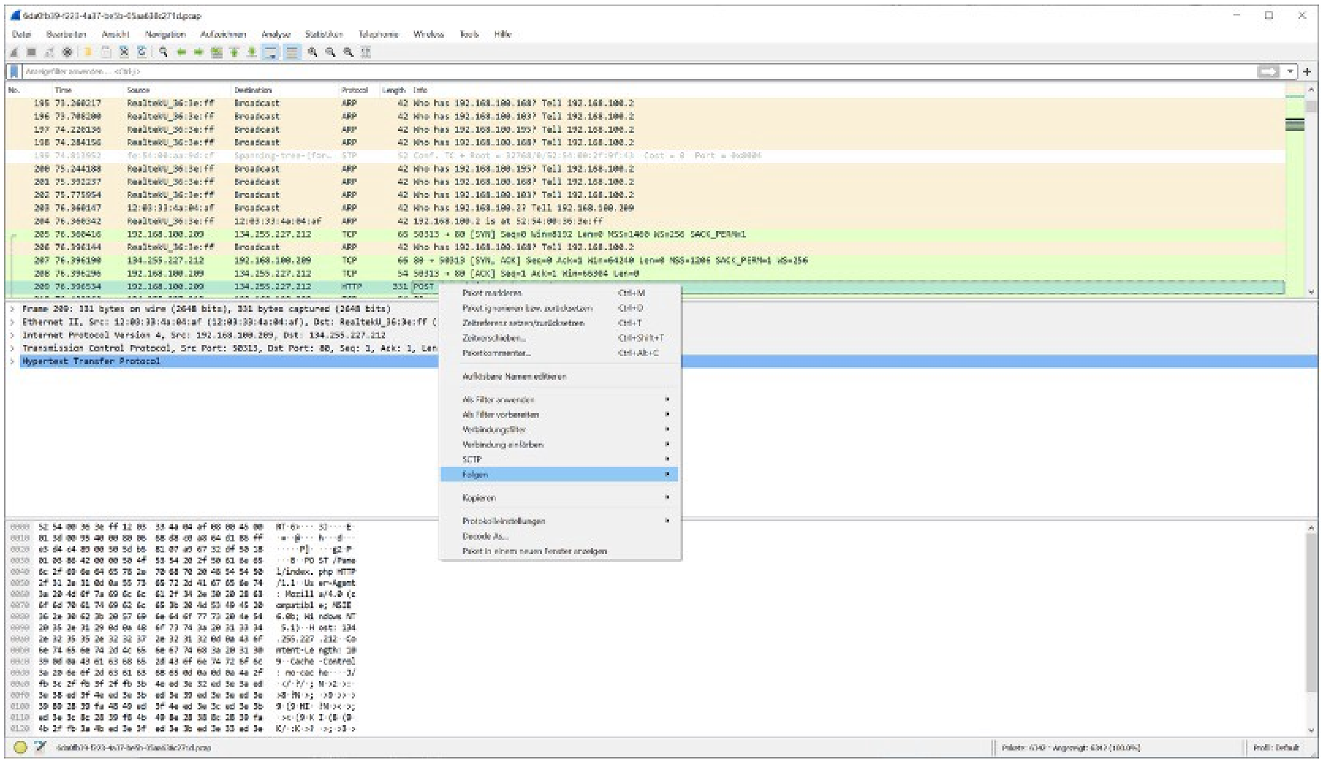
Timeshift	Protocol	Rep	PID	Process name	CN	IP	Port	Domain	ASN	Tr
72313 ms	TCP		988	QUOTATION_PROJECT...		134.255.227.212	80	–	Dennis Rainer ...	277 b
72325 ms	TCP		988	QUOTATION_PROJECT...		134.255.227.212	80	–	Dennis Rainer ...	271 b
79542 ms	TCP		988	QUOTATION_PROJECT...		134.255.227.212	80	–	Dennis Rainer ...	15.1 Kb

QUOTATION_PROJECT_FEJ-1601.exe
MD5: C28DF99844E8464098F71F4745F88AA9
Start: 03.02.2021, 12:36 Total time: 300 s
trojan rat azorult

```
2604 QUOTATION_PROJECT_FEJ-1601.exe  PE                         755   63   144
  2108 schtasks.exe /Create /TN "Updates\xgQDDzEX" /XML "C:\Use...   88    0    46
  988 QUOTATION_PROJECT_FEJ-1601.exe  PE        azorult           1k   52   210
    1272 cmd.exe /c C:\Windows\system32\timeout.exe 3 & del "...   117    6    28
      2924 timeout.exe 3                                           45    0    32
```

A good candidate for further investigation since you can see multiple POST requests

Once you found an appropriate sample, download the PCAP to your machine and open it in Wireshark.

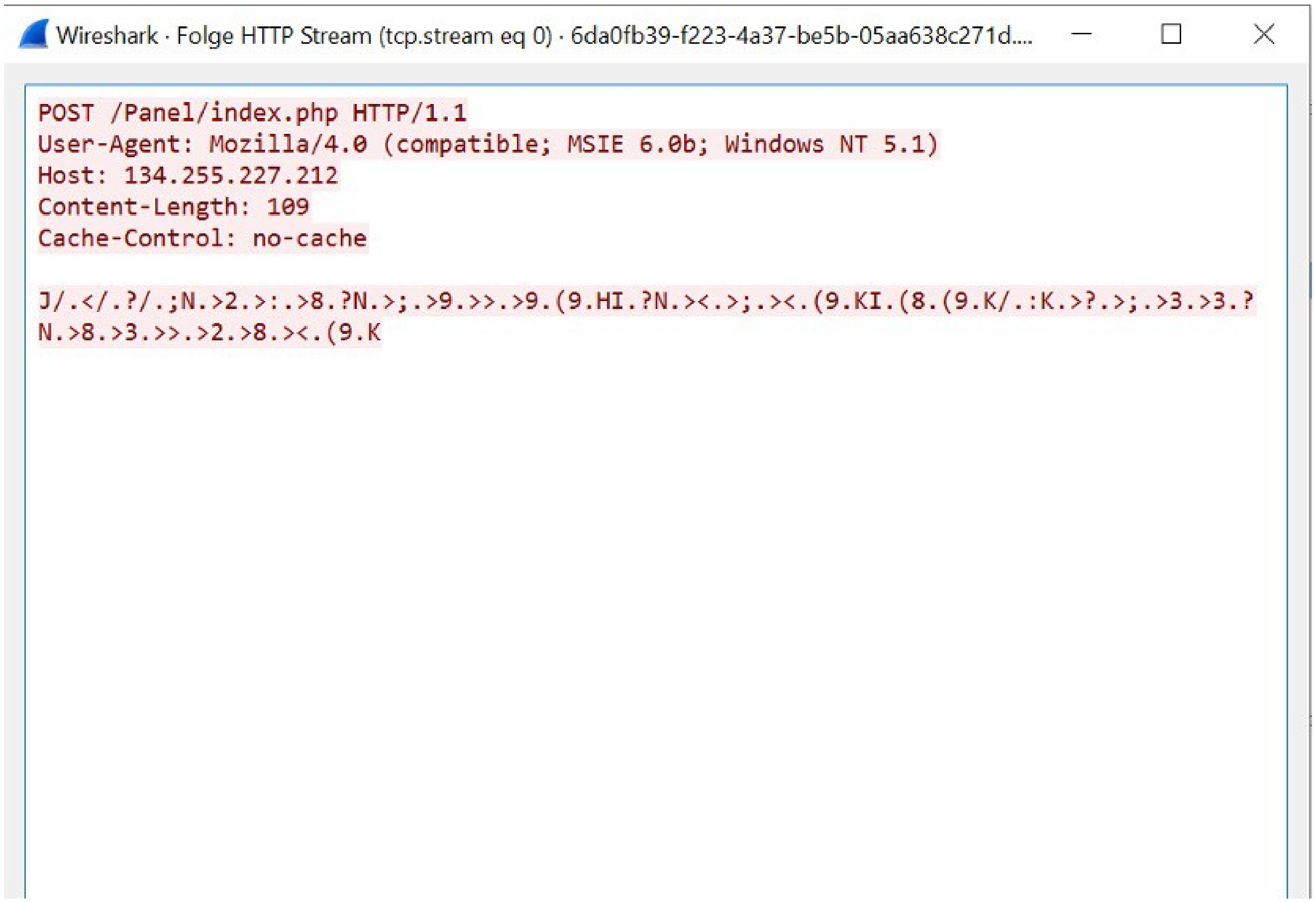


```
No.   Time        Source              Destination          Protocol Length Info
  195 73.260217   RealtekU_36:3e:ff   Broadcast            ARP      42 Who has 192.168.100.168? Tell 192.168.100.2
  196 73.708200   RealtekU_36:3e:ff   Broadcast            ARP      42 Who has 192.168.100.103? Tell 192.168.100.2
  197 74.220136   RealtekU_36:3e:ff   Broadcast            ARP      42 Who has 192.168.100.155? Tell 192.168.100.2
  198 74.284156   RealtekU_36:3e:ff   Broadcast            ARP      42 Who has 192.168.100.168? Tell 192.168.100.2
  199 74.813952   fe:54:00:aa:9d:cf   Spanning-tree-(for…  STP      52 Conf. TC + Root = 32768/0/52:54:00:2f:9f:43  Cost = 0  Port = 0x8004
  200 75.244188   RealtekU_36:3e:ff   Broadcast            ARP      42 Who has 192.168.100.155? Tell 192.168.100.2
  201 75.332237   RealtekU_36:3e:ff   Broadcast            ARP      42 Who has 192.168.100.168? Tell 192.168.100.2
  202 75.775954   RealtekU_36:3e:ff   Broadcast            ARP      42 Who has 192.168.100.103? Tell 192.168.100.2
  203 76.360147   12:03:33:4a:04:af   Broadcast            ARP      42 Who has 192.168.100.2? Tell 192.168.100.209
  204 76.360342   RealtekU_36:3e:ff   12:03:33:4a:04:af    ARP      42 192.168.100.2 is at 52:54:00:36:3e:ff
  205 76.360416   192.168.100.209     134.255.227.212      TCP      66 50313 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
  206 76.396144   RealtekU_36:3e:ff   Broadcast            ARP      42 Who has 192.168.100.168? Tell 192.168.100.2
  207 76.396190   134.255.227.212     192.168.100.209      TCP      66 80 → 50313 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1286 SACK_PERM=1 WS=256
  208 76.396296   192.168.100.209     134.255.227.212      TCP      54 50313 → 80 [ACK] Seq=1 Ack=1 Win=66304 Len=0
  209 76.396534   192.168.100.209     134.255.227.212      HTTP    331 POST

> Frame 209: 331 bytes on wire (2648 bits), 331 bytes captured (2648 bits)
> Ethernet II, Src: 12:03:33:4a:04:af (12:03:33:4a:04:af), Dst: RealtekU_36:3e:ff (
> Internet Protocol Version 4, Src: 192.168.100.209, Dst: 134.255.227.212
> Transmission Control Protocol, Src Port: 50313, Dst Port: 80, Seq: 1, Ack: 1, Len
> Hypertext Transfer Protocol

0000  52 54 00 36 3e ff 12 03  33 4a 04 af 08 00 45 00   RT·6>··· 3J····E·
0010  01 3d 00 95 40 00 80 06  68 d8 c0 a8 64 d1 86 ff   ·=··@··· h···d···
0020  e3 d4 c4 85 00 50 5d b6  81 07 a9 67 32 df 50 18   ·····P]· ···g2·P·
0030  01 00 86 42 00 00 50 4f  53 54 20 2f 50 61 6e 65   ···B··PO ST /Pane
0040  6c 2f 69 6e 64 65 78 2e  70 68 70 20 48 54 54 50   l/index. php HTTP
0050  2f 31 2e 31 0d 0a 55 73  65 72 2d 41 67 65 6e 74   /1.1··Us er-Agent
0060  3a 20 4d 6f 7a 69 6c 6c  61 2f 34 2e 30 20 28 63   : Mozill a/4.0 (c
0070  6f 6d 70 61 74 69 62 6c  65 3b 20 4d 53 49 45 20   ompatibl e; MSIE 
0080  36 2e 30 62 3b 20 57 69  6e 64 6f 77 73 20 4e 54   6.0b; Wi ndows NT
0090  20 35 2e 31 29 0d 0a 48  6f 73 74 3a 20 31 33 34    5.1)··H ost: 134
00a0  2e 32 35 35 2e 32 32 37  2e 32 31 32 0d 0a 43 6f   .255.227 .212··Co
00b0  6e 74 65 6e 74 2d 4c 65  6e 67 74 68 3a 20 31 30   ntent-Le ngth: 10
00c0  39 0d 0a 43 61 63 68 65  2d 43 6f 6e 74 72 6f 6c   9··Cache -Control
00d0  3a 20 6e 6f 2d 63 61 63  68 65 0d 0a 0d 0a 4a 2f   : no-cac he····J/
00e0  fb 3c 2f fb 3f 2f fb 3b  4e ed 3e 32 ed 3e 3a ed   ·</·?/·; N·>2·>:·
00f0  3e 38 ed 3f 4e ed 3e 3b  ed 3e 39 ed 3e 3e ed 3e   >8·?N·>; ·>9·>>·>
0100  39 ed 28 39 fa 48 49 ed  3f 4e ed 3e 3c ed 3e 3b   9·(9·HI· ?N·><·>;
0110  ed 3e 3c 8c 28 39 fb 4b  49 8a 28 38 8c 28 39 fa   ·><·(9·K I·(8·(9·
0120  4b 2f fb 3a 4b ed 3e 3f  ed 3e 3b ed 3e 33 ed 3e   K/·:K·>? ·>;·>3·>
```

Paket markieren Ctrl+M
Paket ignorieren bzw. zurücksetzen Ctrl+D
Zeitreferenz setzen/zurücksetzen Ctrl+T
Zeitverschieben… Ctrl+Shift+T
Paketkommentar… Ctrl+Alt+C
Auflösbare Namen editieren
Als Filter anwenden
Als Filter vorbereiten
Verbindungsfilter
Verbindung einfärben
SCTP
Folgen
Kopieren
Protokolleinstellungen
Decode As…
Paket in einem neuen Fenster anzeigen

Using Wireshark to follow HTTP streams

You then have to be on the lookout for HTTP POST requests. If you want to see the content of the request, you can right click the appropriate row and click on “Follow” and “HTTP stream”

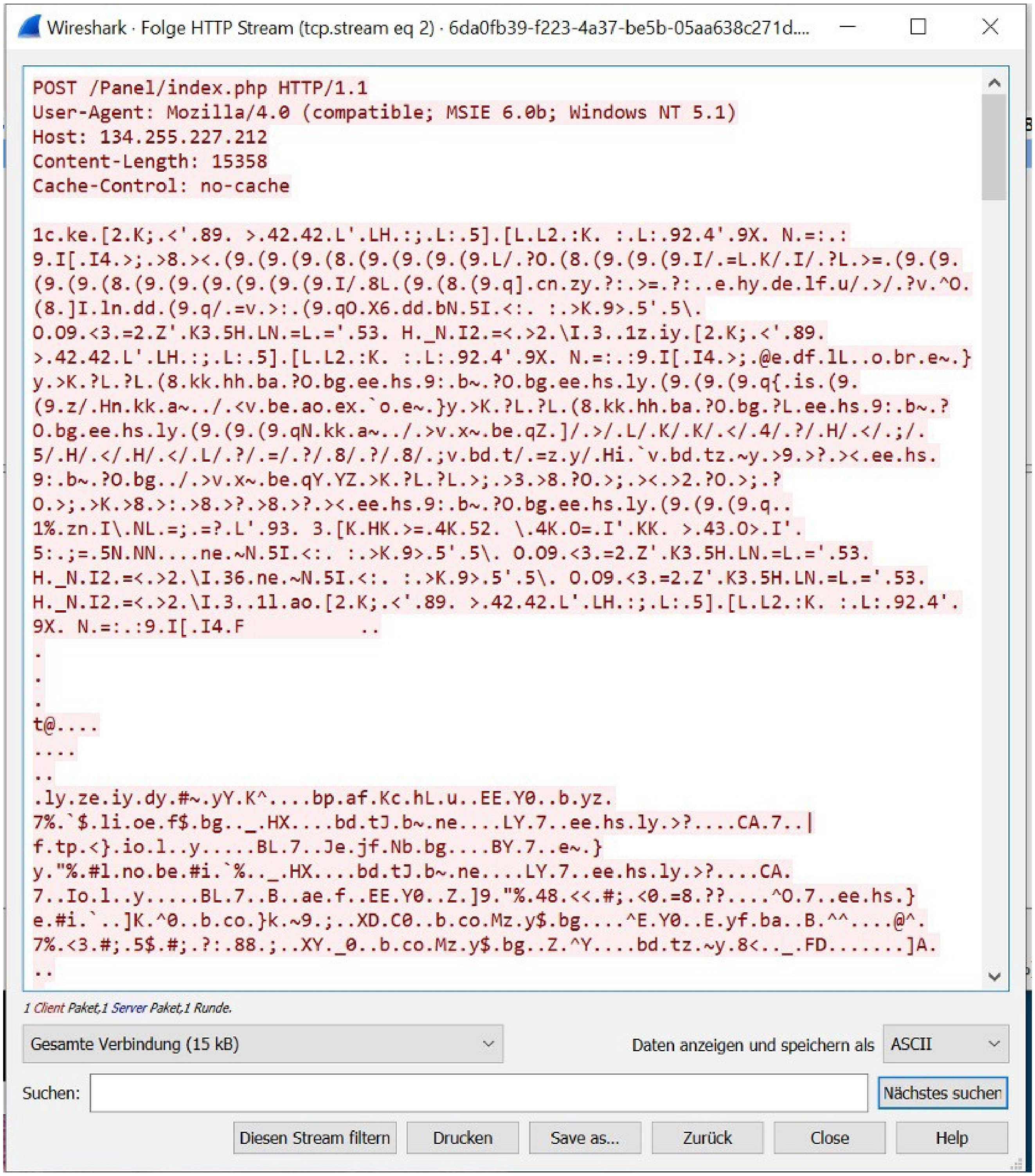


Wireshark · Folge HTTP Stream (tcp.stream eq 0) · 6da0fb39-f223-4a37-be5b-05aa638c271d....

```
POST /Panel/index.php HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0b; Windows NT 5.1)
Host: 134.255.227.212
Content-Length: 109
Cache-Control: no-cache

J/.</.?/.;N.>2.>:.>8.?N.>;.>9.>>.>9.(9.HI.?N.><.>;.><.(9.KI.(8.(9.K/.:K.>?.>;.>3.>3.?
N.>8.>3.>>.>2.>8.><.(9.K
```

The “Check-in” which does not contain any interesting info

You will notice multiple POST requests while the first is pretty small and functions as a check-in to the Command and Control server. Skip this one since this does not contain any valuable data for us.



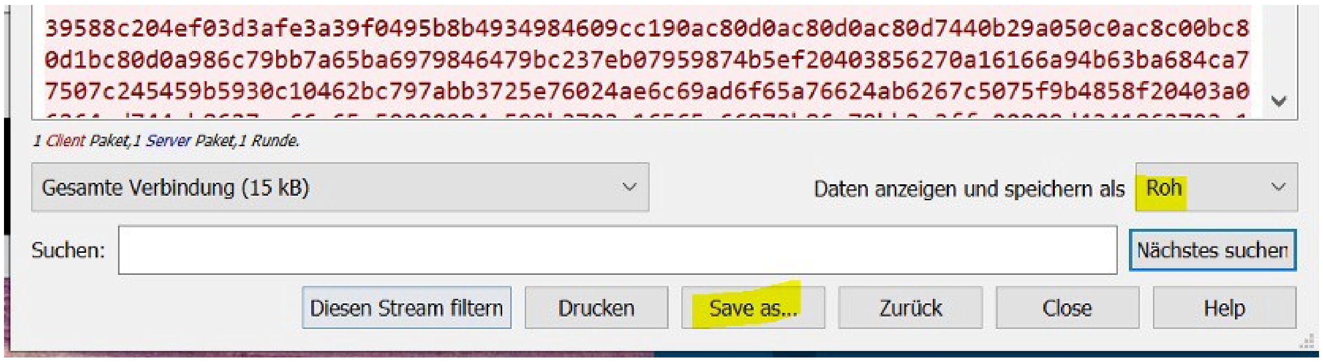
Wireshark · Folge HTTP Stream (tcp.stream eq 2) · 6da0fb39-f223-4a37-be5b-05aa638c271d....

```
POST /Panel/index.php HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0b; Windows NT 5.1)
Host: 134.255.227.212
Content-Length: 15358
Cache-Control: no-cache

1c.ke.[2.K;.<'.89. >.42.42.L'.LH.:;.L:.5].[L.L2.:K. :.L:.92.4'.9X. N.=:.:
9.I[.I4.>;.>8.><.(9.(9.(9.(8.(9.(9.(9.(9.L/.?O.(8.(9.(9.(9.I/.=L.K/.I/.?L.>=.(9.(9.
(9.(9.(8.(9.(9.(9.(9.(9.(9.I/.8L.(9.(8.(9.q].cn.zy.?:.>=.?:..e.hy.de.1f.u/.>/.?v.^O.
(8.]I.ln.dd.(9.q/.=v.>:.(9.qO.X6.dd.bN.5I.<:. :.>K.9>.5'.5\.
O.O9.<3.=2.Z'.K3.5H.LN.=L.='.53. H._N.I2.=<.>2.\I.3..1z.iy.[2.K;.<'.89.
>.42.42.L'.LH.:;.L:.5].[L.L2.:K. :.L:.92.4'.9X. N.=:.:9.I[.I4.>;.@e.df.1L..o.br.e~.}
y.>K.?L.?L.(8.kk.hh.ba.?O.bg.ee.hs.9:.b~.?O.bg.ee.hs.ly.(9.(9.(9.q{.is.(9.
(9.z/.Hn.kk.a~../.<v.be.ao.ex.`o.e~.}y.>K.?L.?L.(8.kk.hh.ba.?O.bg.?L.ee.hs.9:.b~.?
O.bg.ee.hs.ly.(9.(9.(9.qN.kk.a~../.>v.x~.be.qZ.]/.>/.L/.K/.K/.</.4/.?/.H/.</.;/.
5/.H/.</.H/.</.L/.?/.=/.?/.8/.?/.8/.;v.bd.t/.=z.y/.Hi.`v.bd.tz.~y.>9.>?.><.ee.hs.
9:.b~.?O.bg../.>v.x~.be.qY.YZ.>K.?L.?L.>;.>3.>8.?O.>;.><.>2.?O.>;.?
O.>;.>K.>8.>:.>8.>?.>8.>?.><.ee.hs.9:.b~.?O.bg.ee.hs.ly.(9.(9.(9.q..
1%.zn.I\.NL.=;.=?.L'.93. 3.[K.HK.>=.4K.52. \.4K.O=.I'.KK. >.43.O>.I'.
5:.;=.5N.NN....ne.~N.5I.<:. :.>K.9>.5'.5\. O.O9.<3.=2.Z'.K3.5H.LN.=L.='.53.
H._N.I2.=<.>2.\I.36.ne.~N.5I.<:. :.>K.9>.5'.5\. O.O9.<3.=2.Z'.K3.5H.LN.=L.='.53.
H._N.I2.=<.>2.\I.3..1l.ao.[2.K;.<'.89. >.42.42.L'.LH.:;.L:.5].[L.L2.:K. :.L:.92.4'.
9X. N.=:.:9.I[.I4.F          ..
.
.
.
t@....
....
..
.ly.ze.iy.dy.#~.yY.K^....bp.af.Kc.hL.u..EE.Y0..b.yz.
7%.`$.li.oe.f$.bg.._.HX....bd.tJ.b~.ne....LY.7..ee.hs.ly.>?....CA.7..|
f.tp.<}.io.l..y.....BL.7..Je.jf.Nb.bg....BY.7..e~.}
y."%.#l.no.be.#i.`%.._.HX....bd.tJ.b~.ne....LY.7..ee.hs.ly.>?....CA.
7..Io.l..y.....BL.7..B..ae.f..EE.Y0..Z.]9."%.48.<<.#;.<0.=8.??....^O.7..ee.hs.}
e.#i.`..]K.^0..b.co.}k.~9.;..XD.C0..b.co.Mz.y$.bg....^E.Y0..E.yf.ba..B.^^....@^.
7%.<3.#;.5$.#;.?:.88.;..XY._0..b.co.Mz.y$.bg..Z.^Y....bd.tz.~y.8<.._.FD.......]A.
..
```

1 Client Paket,1 Server Paket,1 Runde.

Size matters! The biggest POST request in the PCAP is our target

One man’s trash is another man’s treasure! Looks like trash but is actually the stolen data getting exfiltrated! That’s the request we are interested in! Notice, that this request contains much more data!



```
39588c204ef03d3afe3a39f0495b8b4934984609cc190ac80d0ac80d0ac80d7440b29a050c0ac8c00bc8
0d1bc80d0a986c79bb7a65ba6979846479bc237eb07959874b5ef20403856270a16166a94b63ba684ca7
7507c245459b5930c10462bc797abb3725e76024ae6c69ad6f65a76624ab6267c5075f9b4858f20403a0
```

1 Client Paket,1 Server Paket,1 Runde.

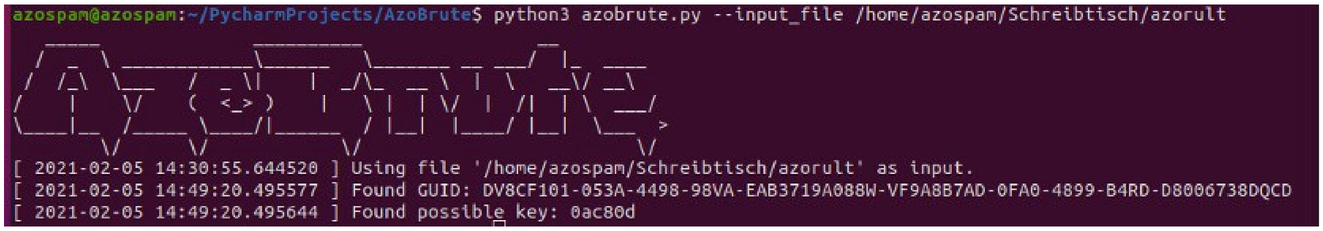
Gesamte Verbindung (15 kB)

Daten anzeigen und speichern als: Roh

Suchen:

Change view and save

Change the view to “Raw” and save the output to disk so we can further process it.



```
azospam@azospam:~/PycharmProjects/AzoBrute$ python3 azobrute.py --input_file /home/azospam/Schreibtisch/azorult
[ 2021-02-05 14:30:55.644520 ] Using file '/home/azospam/Schreibtisch/azorult' as input.
[ 2021-02-05 14:49:20.495577 ] Found GUID: DV8CF101-053A-4498-98VA-EAB3719A088W-VF9A8B7AD-0FA0-4899-B4RD-D8006738DQCD
[ 2021-02-05 14:49:20.495644 ] Found possible key: 0ac80d
```

You might need some patience

Now comes the fun part! As you might have noticed, the POST request data is encrypted in some way. Turns out, it is just XORed with a 3 byte key which unfortunately is not the same for all variants. What now? Make “some” educated guesses?

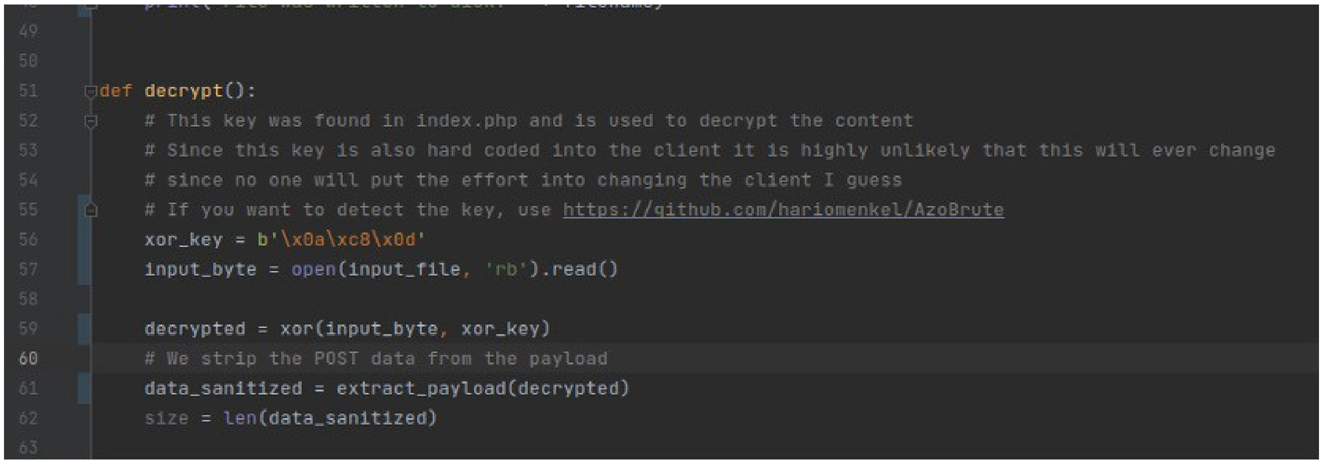
Fear not, I created a tool which first tries to decrypt it with keys I found in the wild and if this is not successful, it will start to brute force the key. This is possible with the help of a known plaintext attack since I learned through manually reversing AzoRult that the plaintext stolen data contains strings like “<info” which we can look for after every decryption try.

You can get it here: https://GitHub.com/hariomenkel/AzoBrute

Once downloaded, let it run against the extracted POST request and hopefully, you’ll receive the key.

Please consider creating an issue at the AzoBrute GitHub repository with your key so I can add it to the list of keys which are tried before trying brute force. Sharing is caring!

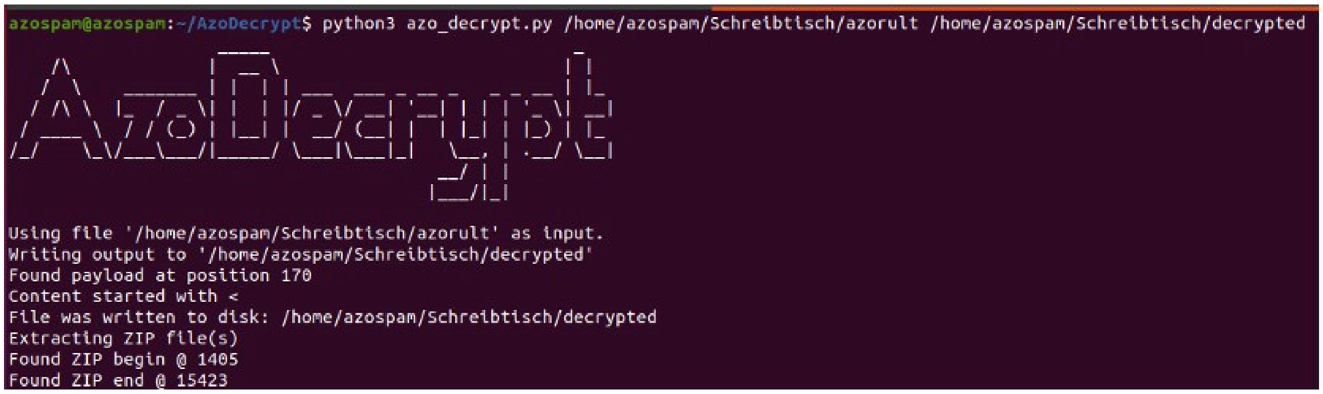
Once you have the key, copy it—you will need it for another tool



```python
def decrypt():
    # This key was found in index.php and is used to decrypt the content
    # Since this key is also hard coded into the client it is highly unlikely that this will ever change
    # since no one will put the effort into changing the client I guess
    # If you want to detect the key, use https://github.com/hariomenkel/AzoBrute
    xor_key = b'\x0a\xc8\x0d'
    input_byte = open(input_file, 'rb').read()

    decrypted = xor(input_byte, xor_key)
    # We strip the POST data from the payload
    data_sanitized = extract_payload(decrypted)
    size = len(data_sanitized)
```

Change the value of “xor_key”

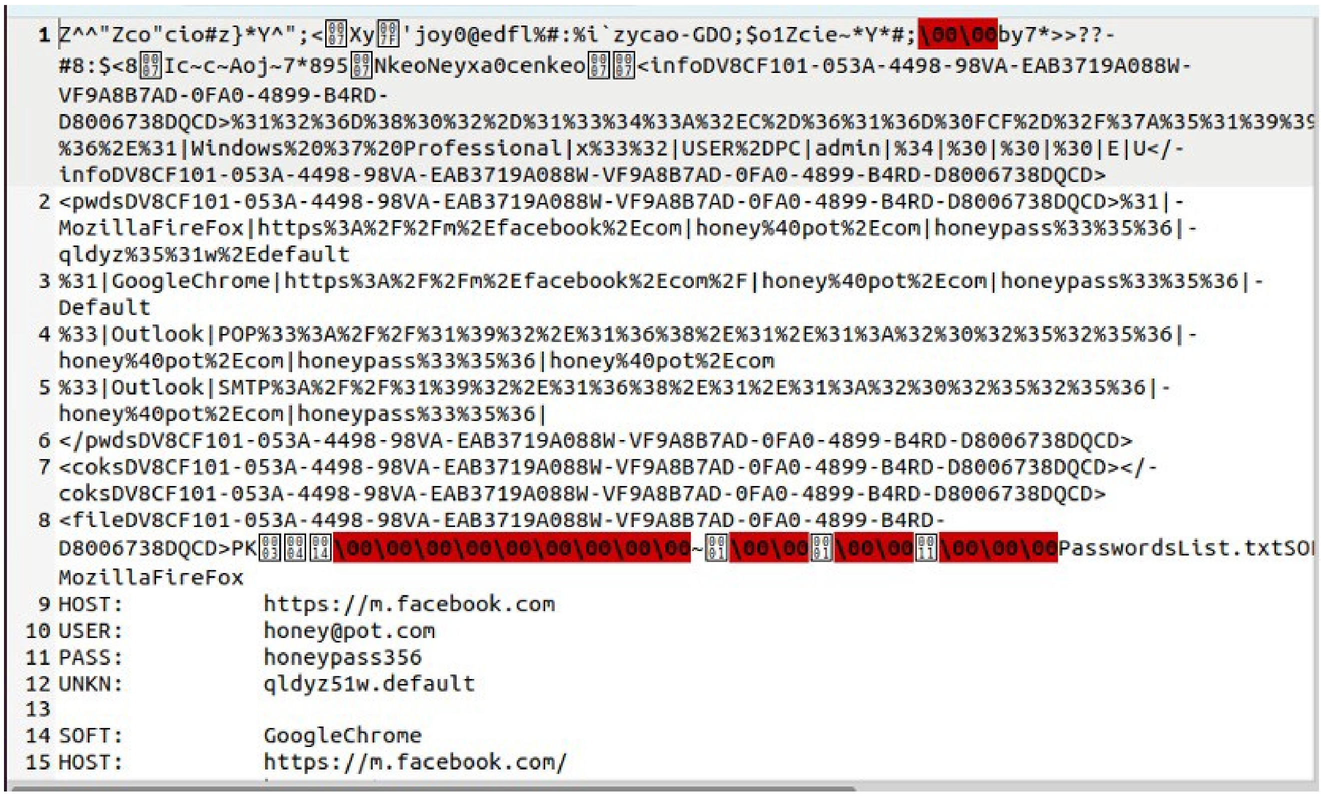
Now when you have the key, you might want to extract the stolen data don’t you? To do so, get https://GitHub.com/hariomenkel/AzoDecrypt and open it in an editor to change the value of “xor_key” to the value you found earlier. Don’t forget to add \x before every byte so the format is the same as before. When you are finished, save it and run that bad boy!



```
azospam@azospam:~/AzoDecrypt$ python3 azo_decrypt.py /home/azospam/Schreibtisch/azorult /home/azospam/Schreibtisch/decrypted
                    _____                             _   
     /\            |  __ \                           | |  
    /  \    _______| |  | | ___  ___ _ __ _   _ _ __ | |_ 
   / /\ \  |_  / _ \ |  | |/ _ \/ __| '__| | | | '_ \| __|
  / ____ \  / / (_) | |__| |  __/ (__| |  | |_| | |_) | |_ 
 /_/    \_\/___\___/|_____/ \___|\___|_|   \__, | .__/ \__|
                                            __/ | |        
                                           |___/|_|        

Using file '/home/azospam/Schreibtisch/azorult' as input.
Writing output to '/home/azospam/Schreibtisch/decrypted'
Found payload at position 170
Content started with <
File was written to disk: /home/azospam/Schreibtisch/decrypted
Extracting ZIP file(s)
Found ZIP begin @ 1405
Found ZIP end @ 15423
```

Decrypt that bad boy

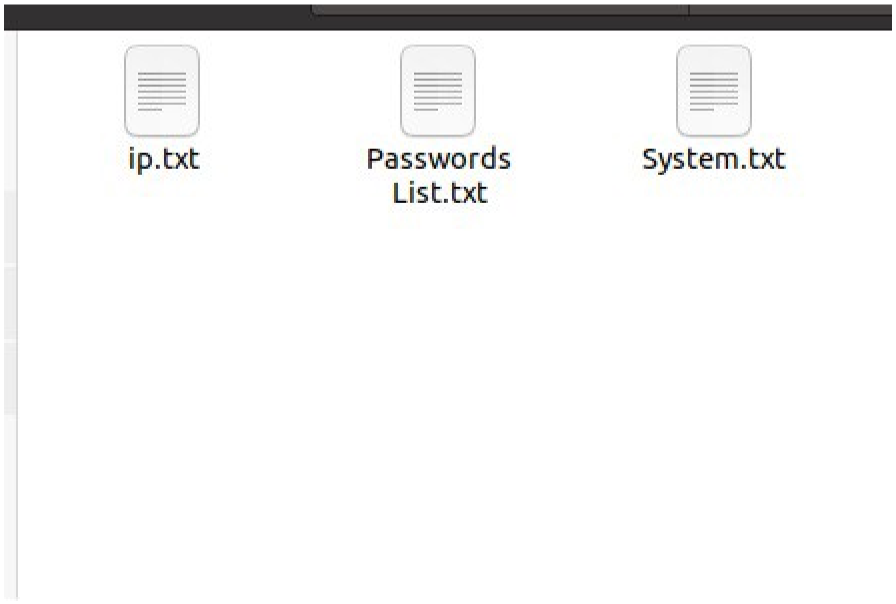
As you can see, the tool was able to use your key to find the payload, extract it and also squeeze out a ZIP archive containing all stolen credentials, cookies, system infos etc.!



```
Z^^"Zco"cio#z}*Y^";<Xy'joy0@edfl%#:%i`zycao-GDO;$o1Zcie~*Y*#;\00\00by7*>>??-
#8:$<8Ic~c~Aoj~7*895NkeoNeyxa0cenkeo<infoDV8CF101-053A-4498-98VA-EAB3719A088W-
VF9A8B7AD-0FA0-4899-B4RD-
D8006738DQCD>%31%32%36D%38%30%32%2D%31%33%34%33A%32EC%2D%36%31%36D%30FCF%2D%32F%37A%35%31%39%3
%36%2E%31|Windows%20%37%20Professional|x%33%32|USER%2DPC|admin|%34|%30|%30|%30|E|U</-
infoDV8CF101-053A-4498-98VA-EAB3719A088W-VF9A8B7AD-0FA0-4899-B4RD-D8006738DQCD>
<pwdsDV8CF101-053A-4498-98VA-EAB3719A088W-VF9A8B7AD-0FA0-4899-B4RD-D8006738DQCD>%31|-
MozillaFireFox|https%3A%2F%2Fm%2Efacebook%2Ecom|honey%40pot%2Ecom|honeypass%33%35%36|-
qldyz%35%31w%2Edefault
%31|GoogleChrome|https%3A%2F%2Fm%2Efacebook%2Ecom%2F|honey%40pot%2Ecom|honeypass%33%35%36|-
Default
%33|Outlook|POP%33%3A%2F%2F%31%39%32%2E%31%36%38%2E%31%2E%31%3A%32%30%32%35%32%35%36|-
honey%40pot%2Ecom|honeypass%33%35%36|honey%40pot%2Ecom
%33|Outlook|SMTP%3A%2F%2F%31%39%32%2E%31%36%38%2E%31%2E%31%3A%32%30%32%35%32%35%36|-
honey%40pot%2Ecom|honeypass%33%35%36|
</pwdsDV8CF101-053A-4498-98VA-EAB3719A088W-VF9A8B7AD-0FA0-4899-B4RD-D8006738DQCD>
<coksDV8CF101-053A-4498-98VA-EAB3719A088W-VF9A8B7AD-0FA0-4899-B4RD-D8006738DQCD></-
coksDV8CF101-053A-4498-98VA-EAB3719A088W-VF9A8B7AD-0FA0-4899-B4RD-D8006738DQCD>
<fileDV8CF101-053A-4498-98VA-EAB3719A088W-VF9A8B7AD-0FA0-4899-B4RD-
D8006738DQCD>PK\00\00\00\00\00\00\00\00\00~\00\00\00\00\00\00\00PasswordsList.txtSO
MozillaFireFox
HOST:           https://m.facebook.com
USER:           honey@pot.com
PASS:           honeypass356
UNKN:           qldyz51w.default

SOFT:           GoogleChrome
HOST:           https://m.facebook.com/
```

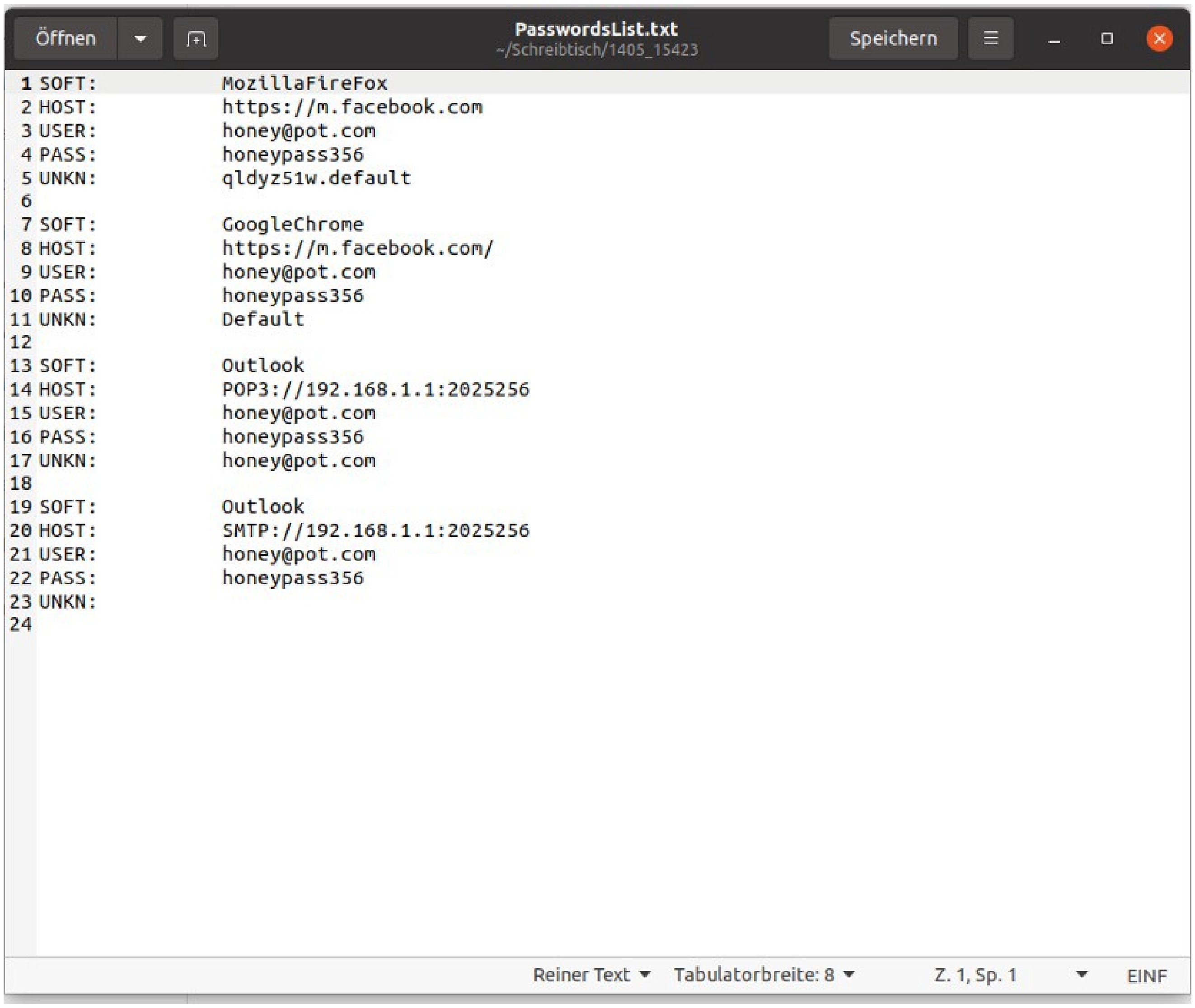
Beauty lies in the eye of the beholder

This is the POST request which has been XORed which already shows some information



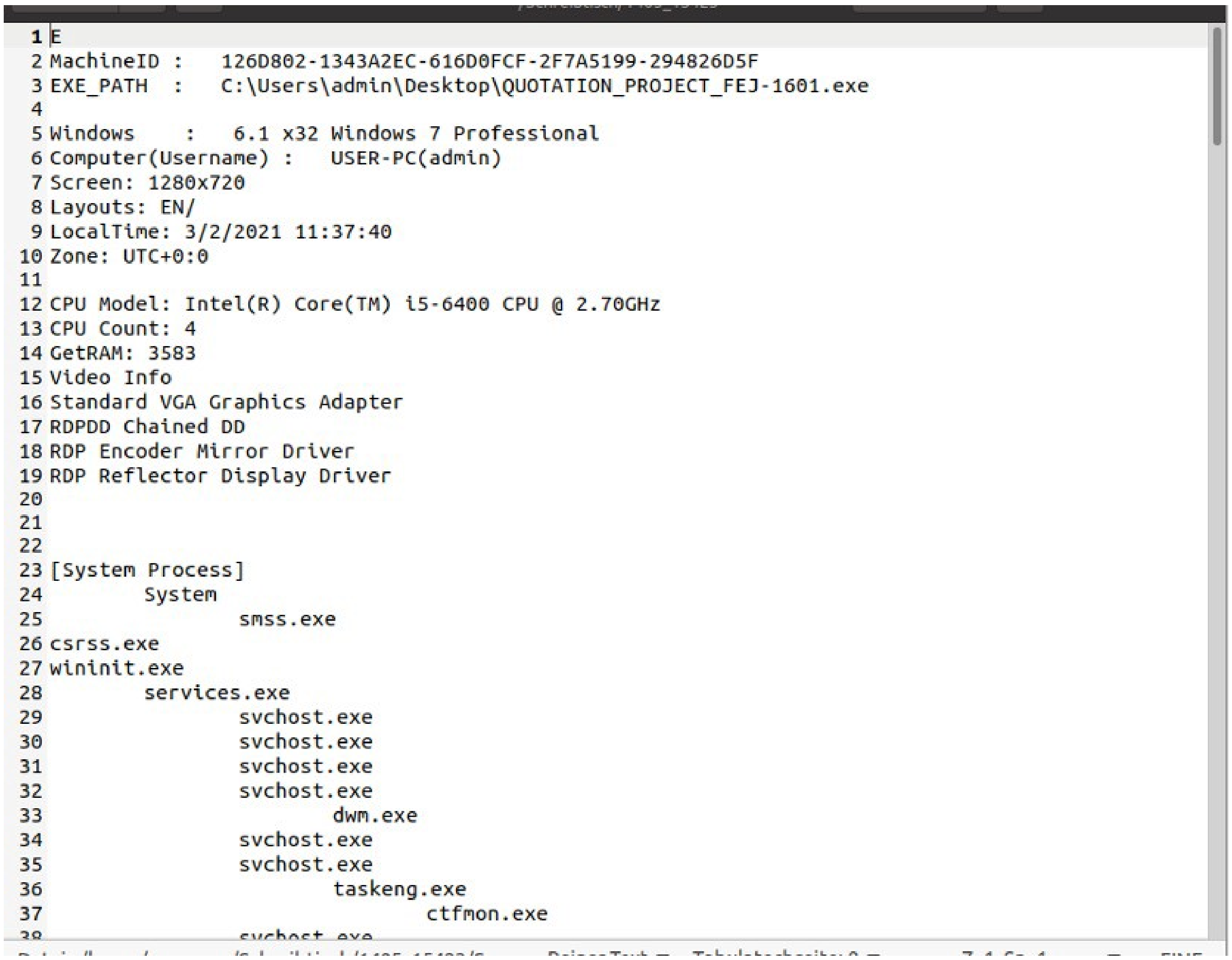
Content of ZIP file

The really interesting data is in the ZIP file. You might want to have a look at it!


PasswordsList.txt
~/Schreibtisch/1405_15423
Öffnen
Speichern

```
SOFT:           MozillaFireFox
HOST:           https://m.facebook.com
USER:           honey@pot.com
PASS:           honeypass356
UNKN:           qldyz51w.default

SOFT:           GoogleChrome
HOST:           https://m.facebook.com/
USER:           honey@pot.com
PASS:           honeypass356
UNKN:           Default

SOFT:           Outlook
HOST:           POP3://192.168.1.1:2025256
USER:           honey@pot.com
PASS:           honeypass356
UNKN:           honey@pot.com

SOFT:           Outlook
HOST:           SMTP://192.168.1.1:2025256
USER:           honey@pot.com
PASS:           honeypass356
UNKN:
```

PasswordsList.txt
Here you can see the Fake credentials any.run was hosting while running the sample



```
E
MachineID :   126D802-1343A2EC-616D0FCF-2F7A5199-294826D5F
EXE_PATH  :   C:\Users\admin\Desktop\QUOTATION_PROJECT_FEJ-1601.exe

Windows    :   6.1 x32 Windows 7 Professional
Computer(Username) :   USER-PC(admin)
Screen: 1280x720
Layouts: EN/
LocalTime: 3/2/2021 11:37:40
Zone: UTC+0:0

CPU Model: Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz
CPU Count: 4
GetRAM: 3583
Video Info
Standard VGA Graphics Adapter
RDPDD Chained DD
RDP Encoder Mirror Driver
RDP Reflector Display Driver



[System Process]
        System
                smss.exe
csrss.exe
wininit.exe
        services.exe
                svchost.exe
                svchost.exe
                svchost.exe
                svchost.exe
                        dwm.exe
                svchost.exe
                svchost.exe
                        taskeng.exe
                                ctfmon.exe
```

Systems.txt

The attacker is also able to gain a lot of information from your system through system.txt

What next?

Now that you have the correct XOR key and GUID you can use another tool I created to annoy the attacker through flooding his/her AzoRult panel with real looking fake data which makes it hard to distinguish between real and fake victims and ultimately may stop the attacker from selling the data due to its bad quality.

You can get it here: https://GitHub.com/hariomenkel/AzoSpam