

Terminal--Snort Challenge (Part 2)

Solution (examining the traffic in Wireshark)

When you look at the packet capture from the Snort sensor, you should immediately notice that it is all DNS, and it is all UDP. Also, every packet has port 53 in either the source or destination.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.126.0.200	249.13.220.86	DNS	95	Standard query 0x9e43 TXT 77616E61
2	0.010190	249.13.220.86	10.126.0.200	DNS	159	Standard query response 0x9e43 TX
3	0.020407	10.126.0.72	210.219.34.1	DNS	95	Standard query 0x8b7a TXT 77616E61
4	0.030583	210.219.34.1	10.126.0.72	DNS	159	Standard query response 0x8b7a TX
5	0.040781	10.126.0.222	172.217.7.233	DNS	64	Standard query 0x6d41 TXT semes.b

> Frame 2: 159 bytes on wire (1272 bits), 159 bytes captured (1272 bits)
> Internet Protocol Version 4, Src: 249.13.220.86, Dst: 10.126.0.200
> User Datagram Protocol, Src Port: 53, Dst Port: 43606
> Domain Name System (response)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.126.0.200	249.13.220.86	DNS	95	Standard query 0x9e43 TXT 77616E61
2	0.010190	249.13.220.86	10.126.0.200	DNS	159	Standard query response 0x9e43 TX
3	0.020407	10.126.0.72	210.219.34.1	DNS	95	Standard query 0x8b7a TXT 77616E61
4	0.030583	210.219.34.1	10.126.0.72	DNS	159	Standard query response 0x8b7a TX
5	0.040781	10.126.0.222	172.217.7.233	DNS	64	Standard query 0x6d41 TXT semes.b

> Frame 1: 95 bytes on wire (760 bits), 95 bytes captured (760 bits)
> Internet Protocol Version 4, Src: 10.126.0.200, Dst: 249.13.220.86
> User Datagram Protocol, Src Port: 43606, Dst Port: 53
> Domain Name System (query)

We can't just block all DNS, though. None of Santa's users would be able to connect to the Internet if we did that. We will have to fine tune our filter somewhat. We can quickly see that what appears to be the evil traffic all has a long hex string prepended to the domain name. We see things like
[long hex string].ugrber.com
[long hex string].rgeubr.net
[long hex string].ugrber.org

The domain names obviously change. If you look at the IP addresses, you will see that the IP address of the server changes as well.

That long text string seems to be in every packet. If we look at the first packet in the capture in detail, we see something interesting.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.126.0.200	249.13.220.86	DNS	95	Standard query 0x9e43 TXT 77616E61636F6F606052E6D06E2E707331.ugrber.org
2	0.010190	249.13.220.86	10.126.0.200	DNS	159	Standard query response 0x9e43 TXT 77616E61636F6F606052E6D06E2E707331.ugrber.org TXT
3	0.020407	10.126.0.72	210.219.34.1	DNS	95	Standard query 0x8b7a TXT 77616E61636F6F606052E6D06E2E707331.rgeubr.net
4	0.030583	210.219.34.1	10.126.0.72	DNS	159	Standard query response 0x8b7a TXT 77616E61636F6F606052E6D06E2E707331.rgeubr.net TXT
5	0.040781	10.126.0.222	172.217.7.233	DNS	64	Standard query 0x6d41 TXT sees.blogspot.com
6	0.051000	172.217.7.233	10.126.0.222	DNS	123	Standard query response 0x6d41 TXT sees.blogspot.com TXT
7	0.061221	10.126.0.43	90.138.219.232	DNS	83	Standard query 0xb2ed TXT uncarnivoraciousness.birchtree.yahoo.com
8	0.071000	90.138.219.232	10.126.0.43	DNS	145	Standard query response 0xb2ed TXT uncarnivoraciousness.birchtree.yahoo.com TXT

Where the periods would be in the domain name, there are numbers. We can use that to get an idea of

how many digits are in each section. For example, in this query, the name is 77616E6E61636F6F6B69652E6D696E2E707331.ugrber.org

Domain Name System (query)		
Transaction ID: 0x9e43		
> Flags: 0x0100 Standard query		
Questions: 1		
Answer RRs: 0		
Authority RRs: 0		
Additional RRs: 0		
Queries		
> 77616E6E61636F6F6B69652E6D696E2E707331.ugrber.org: type TXT, class IN		
[Response In: 2]		

0000	45 00 00 5f 00 01 00 00	40 11 99 e3 0a 7e 00 c8	E @
0010	f9 0d dc 56 aa 56 00 35	00 4b 64 c6 9e 43 01 00	. . . V . V . 5 . Kd . . C . .
0020	00 01 00 00 00 00 00 00	26 37 37 36 31 36 45 36 & 77616E6
0030	45 36 31 36 33 36 46 36	46 36 42 36 39 36 35 32	E61636F6 F6B69652
0040	45 36 44 36 39 36 45 32	45 37 30 37 33 33 31 06	E6D696E2 E707331.
0050	75 67 72 62 65 72 03 6f	72 67 00 00 10 00 01	ugrber.o rg

There are 0x26 characters in the hex section, 0x06 in the next (ugrber) and 0x03 in the last (net).

Here is a response. Again, the hex section of the address has 0x26 characters.

Domain Name System (response)		
Transaction ID: 0x8b7a		
> Flags: 0x8400 Standard query response, No error		
Questions: 1		
Answer RRs: 1		
Authority RRs: 0		
Additional RRs: 0		
Queries		
> 77616E6E61636F6F6B69652E6D696E2E707331.rgeubr.net: type TXT, class IN		
Answers		

0000	45 00 00 9f 00 01 00 00	40 11 7a ab d2 db 22 01	E @ . z
0010	0a 7e 00 48 00 35 70 ae	00 8b bf bd 8b 7a 84 00	. . . H . 5p z . .
0020	00 01 00 00 00 00 00 00	26 37 37 36 31 36 45 36 & 77616E6
0030	45 36 31 36 33 36 46 36	46 36 42 36 39 36 35 32	E61636F6 F6B69652
0040	45 36 44 36 39 36 45 32	45 37 30 37 33 33 31 06	E6D696E2 E707331.
0050	72 67 65 75 62 72 03 6e	65 74 00 00 10 00 01 26	rgeubr.n et &
0060	37 37 36 31 36 45 36 45	36 31 36 33 36 46 36 46	77616E6E 61636F6F
0070	36 42 36 39 36 35 32 45	36 44 36 39 36 45 32 45	6B69652E 6D696E2E
0080	37 30 37 33 33 31 06 72	67 65 75 62 72 03 6e 65	707331.r geubr.ne
0090	74 00 00 10 00 01 00 00	02 58 00 03 02 36 34	t X . . . 64

There is some variation in the format. Here we see that there are two digits at the beginning before the long hex string. The hex string is still 0x26 characters long, however.

No.	Time	Source	Destination	Protocol	Length	Info
88	0.887598	210.219.34.1	10.126.0.72	DNS	417	Standard query
89	0.897826	10.126.0.200	249.13.220.86	DNS	98	Standard query
90	0.907952	249.13.220.86	10.126.0.200	DNS	417	Standard query

> Frame 88: 417 bytes on wire (3336 bits), 417 bytes captured (3336 bits)

> Internet Protocol Version 4, Src: 210.219.34.1, Dst: 10.126.0.72

> User Datagram Protocol, Src Port: 53, Dst Port: 65209

▼ Domain Name System (response)

Transaction ID: 0xf8f0

> Flags: 0x8400 Standard query response, No error

Questions: 1

Answer RRs: 1

Authority RRs: 0

Additional RRs: 0

▼ Queries

▼ 13.77616E6E61636F6F6B69652E6D696E2E707331.rgeubr.net: type TXT, class IN

Name: 13.77616E6E61636F6F6B69652E6D696E2E707331.rgeubr.net

[Name Length: 52]

[Label Count: 4]

Type: TXT (Text strings) (16)

Class: IN (0x0001)

> Answers

[Request In: 87]

If we can write something that finds long strings of hex, we are half the way there. This is a simple task for regular expressions.

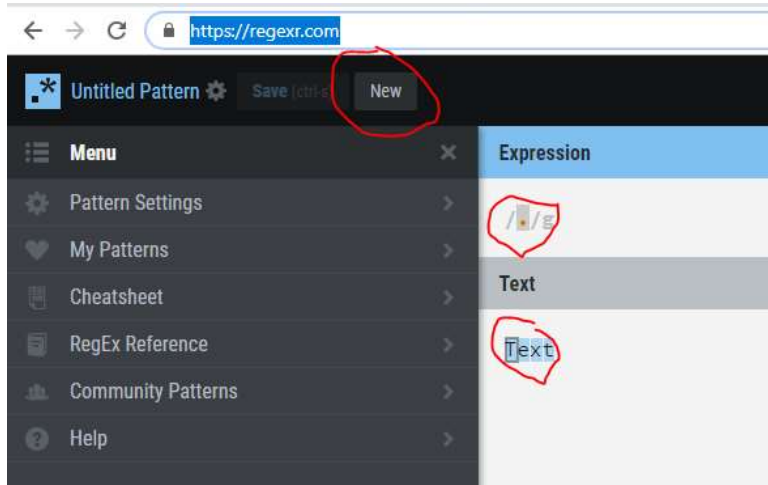
First, read about character classes. We want to make one that alerts on hex digits. Not only that, the malware does not appear to use lower case letters. You need a short expression that will alert on one hex digit, comprised of either numbers or the letters A through F.

<https://www.regular-expressions.info/charclass.html>

Next, we need to alert on a long string of hex instead of one character. The article below talks about “limiting” the number of matches, which is not quite what we want. Instead of matching something like one to four characters {1,4} we want to match on a big number {big number}. The number should not be so big that we miss packets, however.

<https://www.regular-expressions.info/repeat.html>

Finally, you can [go to this site](https://regexr.com) and test your regex if you like.



Click on New to clear the page, put your regex in Expression, copy data from the packet into Text, and see what happens. Test some that should not match (www.freddeadbeef.com or something) to make sure you do not have false positives.

Hand in

- 1) What is the regular expression you will use to detect the evil traffic?