Cyber And Ramen

Saturday, August 10, 2019

Malware Traffic Analysis Exercise (July 2019)

It had been a while since the last time I completed one of Brad's exercises. I felt I might be a little rusty with Wireshark and Scapy, so I decided to lookup the latest blog entry.

The exercise for this post can be found at: https://www.malware-traffic-analysis.net/2019/07/19/index.html

Our Mission:

Review the pcap and alerts to answer the following questions:

- What is the IP address, MAC address, and host name of the infected Windows host?
- What is the Windows user account name for the infected Windows host?
- Based on the alerts what is the name of the campaign that delivered the malware?
- Based on the alerts, what is the final malware that infected the Windows host?
- What are the two IP addresses used in the actual infection traffic?
- What type of animal is in the desktop background of the infected Windows host?

Before we start answering the above questions, I like to utilize not only Wireshark, but Scapy as well for parsing PCAP files. To execute the Scapy commands, I will use Jupyter Notebook. In my opinion, the notebooks make this kind of hunting (we are really just looking through a PCAP) cleaner and more readable as opposed to the terminal.

Search This Blog

Search

Pages

Resume & Links to my Other Blog

About Me

msec1203

View my complete profile

Report Abuse

Blog Archive

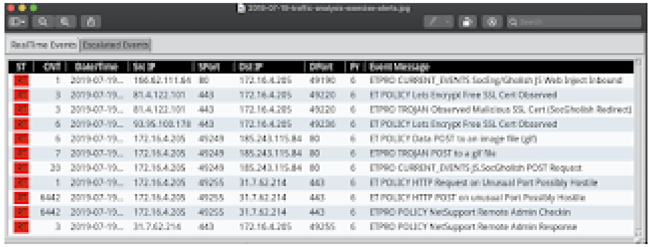
▼ 2019 (6)

► September (1)

▼ August (2)

Enough talking, let's get to it!

Since this exercise includes alerts files, that is where I will begin. The alerts will give us an idea of what type of traffic we need to search for. This way we aren't digging through SMB traffic when the workstation was compromised by a web exploit.



ST	CNT	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	1	2019-07-19...	166.62.111.64	80	172.16.4.205	49190	6	ETPRO CURRENT_EVENTS SocGholish JS Web Inject Inbound
RT	3	2019-07-19...	81.4.122.101	443	172.16.4.205	49220	6	ET POLICY Lets Encrypt Free SSL Cert Observed
RT	3	2019-07-19...	81.4.122.101	443	172.16.4.205	49220	6	ETPRO TROJAN Observed Malicious SSL Cert (SocGholish Redirect)
RT	6	2019-07-19...	93.95.100.178	443	172.16.4.205	49236	6	ET POLICY Lets Encrypt Free SSL Cert Observed
RT	6	2019-07-19...	172.16.4.205	49249	185.243.115.84	80	6	ET POLICY Data POST to an image file (gif)
RT	7	2019-07-19...	172.16.4.205	49249	185.243.115.84	80	6	ETPRO TROJAN POST to a gif file
RT	20	2019-07-19...	172.16.4.205	49249	185.243.115.84	80	6	ETPRO CURRENT_EVENTS JS.SocGholish POST Request
RT	1	2019-07-19...	172.16.4.205	49255	31.7.62.214	443	6	ET POLICY HTTP Request on Unusual Port Possibly Hostile
RT	6442	2019-07-19...	172.16.4.205	49255	31.7.62.214	443	6	ET POLICY HTTP POST on unusual Port Possibly Hostile
RT	6442	2019-07-19...	172.16.4.205	49255	31.7.62.214	443	6	ETPRO POLICY NetSupport Remote Admin Checkin
RT	3	2019-07-19...	31.7.62.214	443	172.16.4.205	49255	6	ETPRO POLICY NetSupport Remote Admin Response

Figure 1

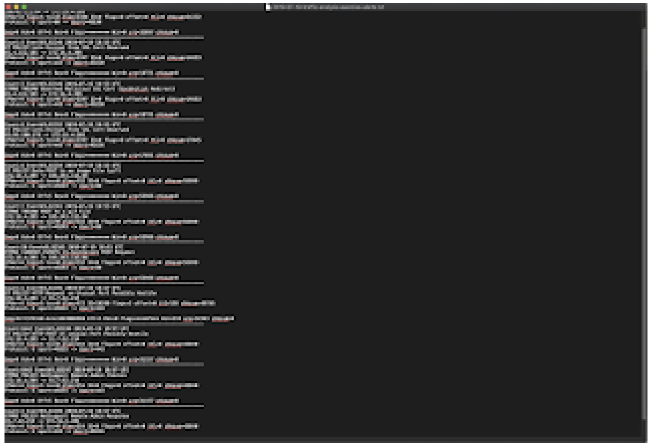


Figure 2

In the above two alert files we can clearly see events related to SocGholish, POST requests to an image file, as well as NetSupport Remote Admin Checkin/Response messages. The latter two events should jump out to an analyst as possible command and control traffic, and kick incident response into gear.

Before we start looking at possible malware infections, let's head back up to our questions and

Quick Network Analysis of Excel Document Utilizing...

Malware Traffic Analysis Exercise (July 2019)

▶ July (1)

▶ June (1)

▶ May (1)

answer the first two.

Filtering for NetBios Name Service traffic, we can answer our first question pretty easily. See Figure 3.

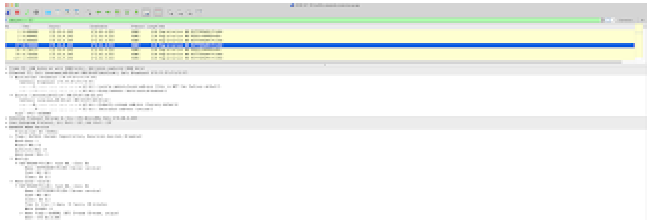


Figure 3

We now have a victim IP address of 172.16.4[.]205, a MAC address of 00:59:07:b0:63:a4, and host name of Rotterdam-PC.

Filtering through Kerberos traffic, we can grab the victim account name and answer question #2.

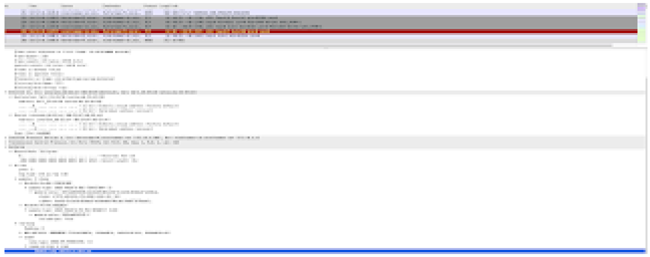


Figure 4

We know that our alerts files identified the infection as SocGholish as the campaign that is responsible for the infection. For now we will answer that question and come back if needed.

The remaining questions involve a bit more than just filtering on different ports in Wireshark and will require us to get our hands dirty. There are a number of ways one could go about filtering out useless traffic, this just happens to be my way.

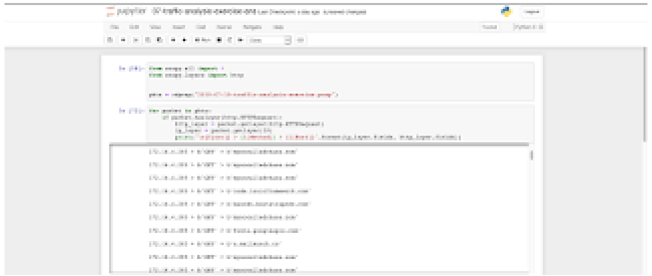


Figure 5

In the above image, we simply read in the PCAP using Scapy then use a couple loop statements to reveal HTTP requests from our victim to outside the network. This can easily be done in Wireshark, but then you wouldn't have the fun of using python to read a PCAP.

The first thing we notice looking at the above output is there are a large number of requests to mysocalledchaos[.]com. What are Scapy script doesn't give us is the length of these connections and number of packets in order.

Let's take a quick look at the "Conversations" in Wireshark.

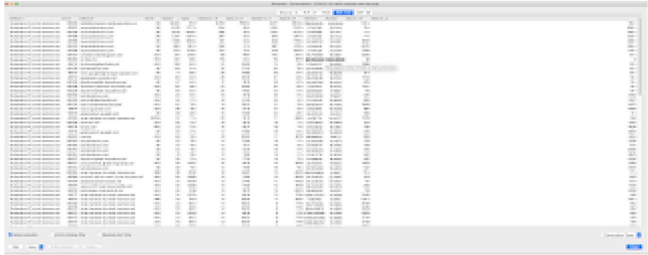


Figure 6

Since we have resolved our IP addresses to host names, we can easily tell who our victim may be communicating with. Interestingly, our Scapy script had a large number of connections to mysocalledchaos[.]com. But filtering by the most packets, we see a new domain appear: b5689023[.]green.mattingsolutions.co, IP address: 185[.]243.115.84.

Going down the line, traffic to 31.7[.]62.214 has a suspiciously high relative start and duration. This could be command and control traffic, but we need to investigate further. For now, we can keep the two IP addresses above in our pocket.

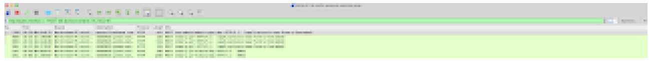


Figure 7

Filtering out our possible C2 traffic and only looking for POST requests as the alerts told us, we can quickly identify the requests for the malicious GIF files. A quick Google search on "empty.gif" brings up a number of app.any.run and hybrid-analysis reports for review.

For the final malware that infected the host question, we need to trot back to the 31.*.*.* address we saw the high duration time for.

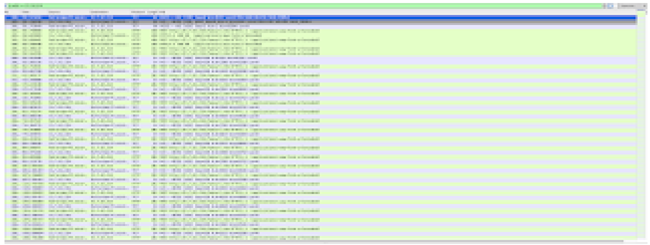


Figure 8

Hopefully the above traffic in Figure 8 looks alarming to you and just looking at the times of the packets we can confirm this is indeed C2 traffic. After some searching around, the NetSupport RAT is a well known trojan that utilizes fake update pages to infect victims.

While there were other IP addresses that could have required more digging, we can answer the second to last question. Actual infection traffic belongs to addresses 185[.]243.115.84 & 31.7[.]62.214.

The last question took me a while to answer, and really showed how rusty I was working Wireshark. The POST requests for the image files in Figure 7 can be exported easily in Wireshark.

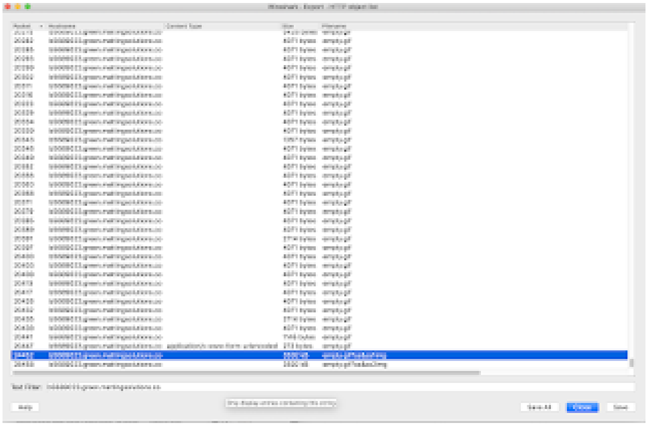


Figure 9

Saving the file to our computer and opening it leads us to answer our last question. Figure 10 depicts the image on the victim's desktop and concludes the exercise.

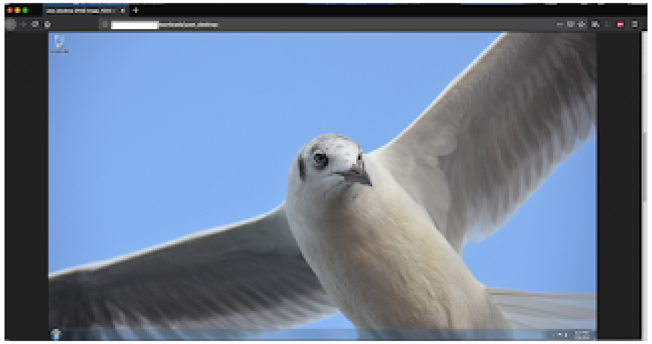


Figure 10

That is it for now. I hope to be more active on here and share some of the research I am conducting in my home lab. I am focusing on intrusion detection, both on the network and endpoint.

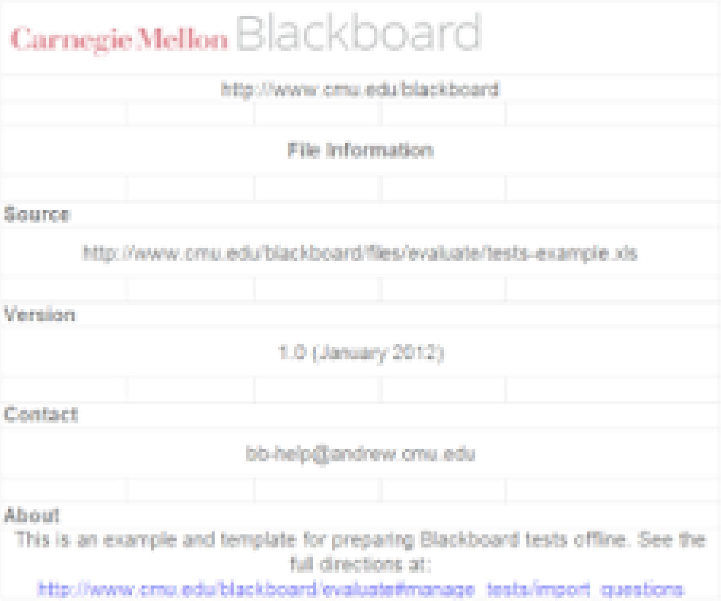
Until the next time!

at August 10, 2019

Newer Post Home Older Post

Analysis of RTF document from Cyber Comm Drop

18 Sep 2019 About a week ago, U.S. Cyber Command made public a number of malware samples purported to be from North Korea via Twitter. It...



Carnegie Mellon Blackboard

http://www.cmu.edu/blackboard

File Information

Source

http://www.cmu.edu/blackboard/files/evaluate/tests-example.xls

Version

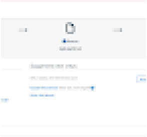
1.0 (January 2012)

Contact

bb-help@andrew.cmu.edu

About

This is an example and template for preparing Blackboard tests offline. See the full directions at:
http://www.cmu.edu/blackboard/evaluate#manage_tests/import_questions



Defeating The Empire With The Basics: Detecting Powershell Empire

19 May 2019 Introduction   Powershell Empire is a household name for penetration testers, red team members, and even your favorite APT g...

Quick Network Analysis of Excel Document Utilizing WS-Discovery Protocol

I thoroughly enjoy analyzing how malware slithers through endpoints using new/novel techniques to evade detection. Searching through netwo...

Awesome Inc. theme. Powered by Blogger.