UBNETDEF Network Activity Report

Date: 2020-5-4

EXECUTIVE SUMMARY

On March 19th 2020 at 21:24:36 UTC, UBNETDEF detected a Trickbot virus running on a system in the mondogreek computer network. The system was currently being used by Otis Witherspoon when the incident occurred. The system's hostname is "LAPTOP-I9ZPI04" and the IP address is 10.3.11[.]194.

UBNETDEF determined that the Trickbot virus reached the final stage of the cyber kill chain, Actions on Objectives. This means the network activity from this system extremely severe, and it is a large threat on the network. The packet capture information had the Delivery of three total malicious files. One of the files was installed and remotely commanded. The first file, named Yas20.exe, has a sha256 hash of *02db3c6b9aff73bf8a11c41107c836b6c800c919c5d3d1304f336aee03f79f4c* and triggered 62/72 engines through Virustotal. This is the file that was able to execute and extract data.

UBNETDEF could not determine on the other two file's executions. The next two files are named "imgpaper.png" and "cursor.png". The hash of the "imgpaper.png" file is *8aa9c596dd3eb7560bc7416ba181e858f1174fcbcb5432050f3f9a663ed1ffa2*, which triggered 37/72 engines on Virustotal. This file was not seen executing on the network. The hash of the third file is *68798ccf8e2a5f9682a4e011bec288ad9b3f900244f82c6ae5e8ca538725f92e* and triggerd 40/72 engines on Virustotal. This file was also not seen executing on the network. The information listed in Virustotal about connections was used to scan the packet capture, and no connections or actions were made that would be detectable in the packet capture.

The Trickbot first was delivered to the system by HTTP, and after executing, reached out to an external server and started exfiltrating login credentials from the system. There were no credentials from the system itself, and were all from browser data and saved logins.

To solve this problem, defense in depth must be implemented for the future. The infected host must be disconnected. Firewall rules to block malicious connections should be put in place. In addition, implementing Intrusion Detection Systems will be incredibly helpful. To detect malicious traffic and intrusions, these systems will create a strong defense.

CONTENTS

Executive Summary..1

Technical Analysis...2

Recommended Clean Up and Mitigation Strategies...2

Contributing Analysts...2

Appendix: Analysis Cheat Sheet..4

TECHNICAL ANALYSIS

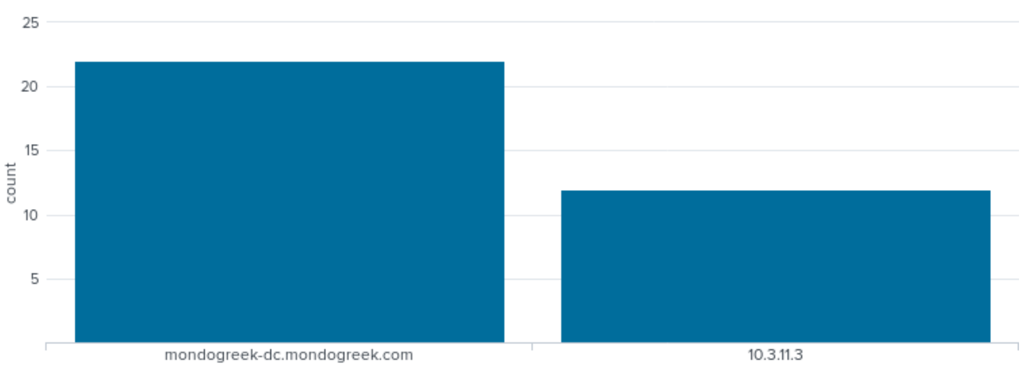
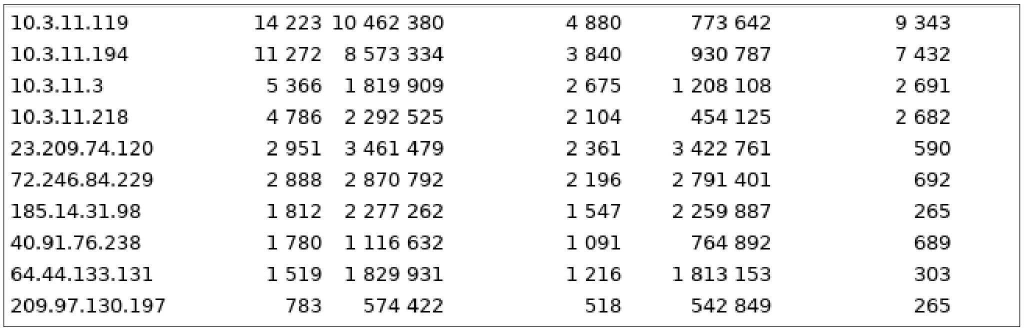


Figure 1. Splunk DNS Search

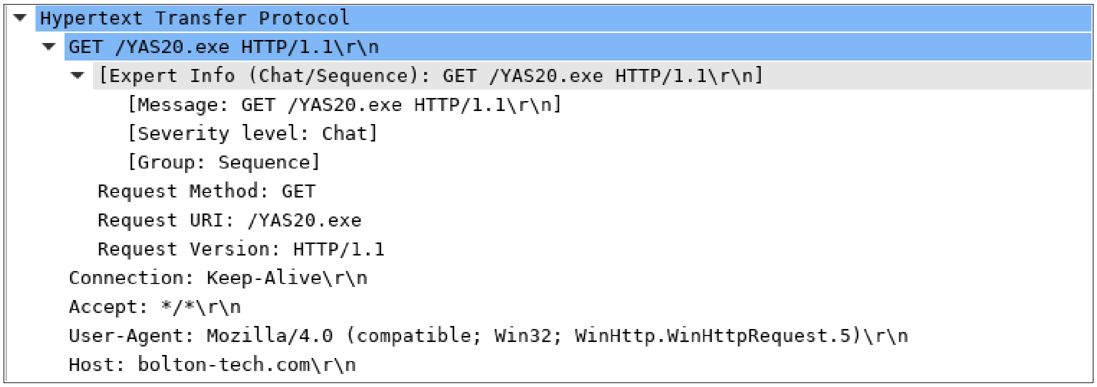
UBNETDEF was given a network capture file from March 11th 2020, recording from 21:22:34 UTC to 22:06:40 UTC. There was a total of 30,000 packets in the 21mb file. UBNETDEF used the tools Wireshark, Splunk, Zeek, Snort, and online references like Virustotal to help identify malware. The data generated by Zeek reading the file was loaded into Splunk. Searching the DNS resolutions show the domain controller's hostname and associated ip address, mondogreek-dc.mondogreek[.]com and 10.3.11[.]3 (see Figure 1). Using wireshark to check the endpoints and packet counts, there are three notable IPs at the top of the packet counts. Two are computers in the network with the two highest packet counts (10.3.11[.]119 and 10.3.11[.]194), and then the domain controller with substantially less packets (see Figure 2). One of the two higher packet count computers most likely have malware installed, if not both.



10.3.11.119	14 223	10 462 380	4 880	773 642	9 343
10.3.11.194	11 272	8 573 334	3 840	930 787	7 432
10.3.11.3	5 366	1 819 909	2 675	1 208 108	2 691
10.3.11.218	4 786	2 292 525	2 104	454 125	2 682
23.209.74.120	2 951	3 461 479	2 361	3 422 761	590
72.246.84.229	2 888	2 870 792	2 196	2 791 401	692
185.14.31.98	1 812	2 277 262	1 547	2 259 887	265
40.91.76.238	1 780	1 116 632	1 091	764 892	689
64.44.133.131	1 519	1 829 931	1 216	1 813 153	303
209.97.130.197	783	574 422	518	542 849	265

Figure 2. Wireshark Endpoints

Investigating the first IP listed with 14223 packets, the ip address 10.3.11[.]119 is one of the work computers on the network. The computer's ID is "LAPTOP-SQSA420", and was logged in by the user "dorian.neff". None of the network traffic done was determined to be malicious, and is not considered a threat. The second highest traffic endpoint with 11272 packets sent was then investigated. The ip address 10.3.11[.]194 is associated with a work computer with the ID "LAPTOP-I9ZPI04", and the user logged in is "otis.witherspoon". The network communications were done with a useragent that is not expected when connecting using a web browser (see Figure 3).



```
▼ Hypertext Transfer Protocol
  ▼ GET /YAS20.exe HTTP/1.1\r\n
    ▼ [Expert Info (Chat/Sequence): GET /YAS20.exe HTTP/1.1\r\n]
        [Message: GET /YAS20.exe HTTP/1.1\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Method: GET
      Request URI: /YAS20.exe
      Request Version: HTTP/1.1
    Connection: Keep-Alive\r\n
    Accept: */*\r\n
    User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)\r\n
    Host: bolton-tech.com\r\n
```

Figure 3. Suspicious User Agent

The user agent here is associated with WinHttp, an official Microsoft produced bot. This http request was the first one from the computer, and is not the only one using WinHttp. This request downloaded a PE EXE file named YAS20.exe, determined as malicious through using the hash of the file through Virustotal. The hash triggered 62/72 engines, and is regarded as very dangerous (see Figure 4). After first downloading this file, two more files were downloaded. They were named imgpaper.png and cursor.png. These two files were malware disguised as .png files, and triggered in Virustotal (see Figure 4).

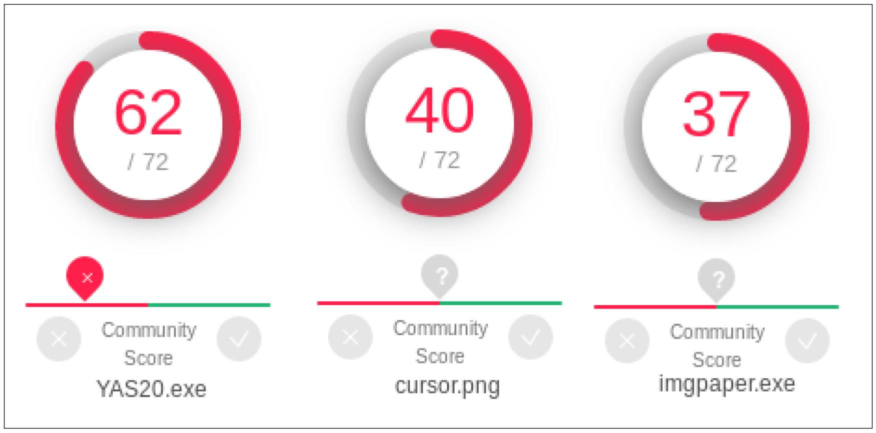
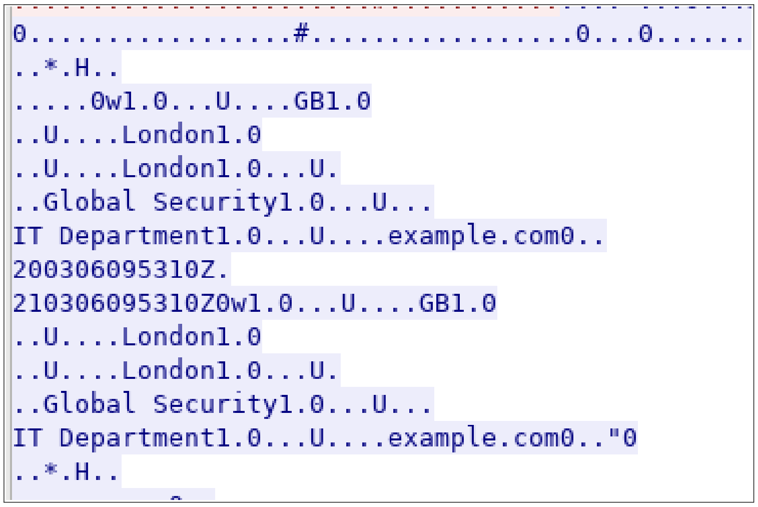


Figure 4. Virustotal Ratings of Files

After the download of these files, the computer made post requests to a server at 203.176.135[.]102. The TCP stream from wireshark shows that the virus was exfiltrating private data. The virus exfiltrated the facebook account and password of Otis Witherspoon, the firefox account, the current running processes on the computer, the system specs, and the networking information of the computer. The virus also attempted to exfiltrate more data from the browser about login information and credentials, but it did not do this successfully.

The virus also used ddm-dfm traffic (distributed file management) to exfiltrate more data. The nature of the information exfiltrated could not be determined, and the total extracted data was about 2.1mb. Basic login information was most likely what was taken through ddm-dfm traffic. The connections to do this had malicious SSL Certificates (see Figure 5).



```
0.................#.................0...0......
..*.H..
.....0w1.0...U....GB1.0
..U....London1.0
..U....London1.0...U.
..Global Security1.0...U...
IT Department1.0...U....example.com0..
200306095310Z.
210306095310Z0w1.0...U....GB1.0
..U....London1.0
..U....London1.0...U.
..Global Security1.0...U...
IT Department1.0...U....example.com0.."0
..*.H..
```

Figure 5. ddm-dfm Certificate

UBNETDEF analyzed all the information avaliable and determined that the virus was a trickbot that reached the final stage of the cyber kill chain, Actions on Objectives. The virus was first deployed in the network at 21:24:36 UTC, when YAS20.exe was first downloaded onto the system. A few moments later, the virus employed the exploitation phase, running the malicious code on the system. The installation phase then followed, giving the malware permanence on the system. Command and Control then engaged, when it first reached out to the remote computer controlling it. It then executed Actions on Objectives, stealing login credentials from the system the virus was installed on.

RECOMMENDED CLEAN UP AND MITIGATION STRATEGIES

UBNETDEF has a set of recommended activities to clean up this incident, and to prevent intrusions like this in the future. First to clean up the incident, disconnect the infected computer from the network. This will prevent the spread of the virus. On a different computer, change the login to the main email address used on the computer. The malware exfiltrated login data, and you have to secure the main email address before the hackers do, so you can then gain control of the other accounts. Also, ensure there are strong passwords to company accounts, as some of the passwords were very weak passwords. After gaining control of accounts again, the clean up on the system can continue. If there is any data that must be exfiltrated, ensure that the last modified date is no later than 21:24:36 UTC on March 19th. If the date is later than that, the file is infected for sure. The malware isn't targeted, so it most likely didn't touch any files. Checking the modified date is a very simple and quick check to make sure. The last modified date is not a check to ensure the file is not infected, it is a check to make sure the file isn't blatantly infected. Files that look unmodified might still be infected, so only exfiltrate exactly what you need. After exfiltrating all the necessary data, wipe the hard drive completely, and reinstall the Windows 10 operating system. Ensure that on the newly installed operating system, Windows Defender is running properly.

Windows Defender recognizes all of these malicious files in this incident, and will mitigate similar attacks.

UBNETDEF recommends the implementation of an Intrusion Detection System (IDS). There are 3 types of IDS that can be implemented. There are network IDS like Snort and Zeek. A network IDS can watch the entire network and notify you of anything on the network. There are host IDS like OSSEC, which watch over a specific host. This will be a good idea to implement onto extremely important systems, such as a domain controller, to mitigate attacks on critical systems. There is a third type, which instead of watching network traffic, analyzes log files, such as ManageEngine EventLog Analyzer. This isn't the ideal system, but still is an option to be aware of. UBNETDEF recommends using Snort to watch the entire network, and a host IDS like OSSEC on critical systems like the domain controller. Setting up Snort with the Community ruleset and the Emerging Threats ruleset will provide an effective system for identifying network intrusions quickly, while OSSEC will defend the systems that your infrastructure is focused around. With these systems implemented, similar attacks in the future will be mitigated.

To help defend the network, setting up firewall rules will also be important. There is a set of IPs listed below. Each IP will need to be implemented as it's own rule, using the rule template listed below. Use the rule template for your firewall setup, or if you have a custom firewall application, block all traffic from these IPs. These IPs listed are malicious ones involved with this incident, and all those listed from Virustotal based off other incidents.

IP List:

50.87.248.17

51.254.164.244

185.14.31.98

64.44.133.131

170.84.78.224

91.235.129.144

181.113.28.146

40.81.188.85

Firewall Rules Templates:

IPTABLES: iptables -A INPUT -s <IP> -j DROP

FirewallD: firewall-cmd --add-rich-rule='rule family=ipv4 source address=<ip> reject' –permanent

*After all of the FirewallD rules, you must enter "firewall-cmd –reload"

UBNETDEF determines these actions to be productive in mitigating incidents through this attack vector again in the future.

CONTRIBUTING ANALYSTS

Lead Analyst: Sean Manly
Contributing Analysts: UBNETDEF

APPENDIX: ANALYSIS CHEAT SHEET

- Generate Snort Alerts - sudo snort -A full -r <name>.pcap -c /etc/snort/snort.conf -U -l ~/<Directory>
- Read the Alerts – (in Directory) cat alerts
- Use zeek to read the files - zeek -r <filename>