Global Information Assurance Certification Paper

The All-Seeing Eye of Sauron: A PowerShell tool for data collection and threat hunting

GIAC (GDAT) Gold Certification

Author: Timothy Hoffman, timothy.hoffman.83@gmail.com
Advisor: *Clay Risenhoover*

Accepted: *August 21, 2020*

Abstract

The cost of a data breach directly relates to the time it takes to detect, contain, and eradicate it. According to a study by the Ponemon Institute, the average time to identify a breach in 2019 was 206 days (Ponemon Institute, 2019). Reducing this timeframe is paramount to reducing the overall timeline of removing a breach, and the costs associated with it. With ever-evolving adversaries creating new ways of compromising organizations, preventive security measures are essential, but not enough. Organizations should not assume they will be compromised, but instead that they already have been. Finding and removing these already existing breaches can be difficult. To find existing breaches, organizations need to conduct threat hunting, which seeks to uncover the presence of an attacker in an environment not previously discovered by existing detection technologies (Gunter & Seitz, 2018). This paper looks at the PowerShell tool, Eye of Sauron, which can be used for threat hunting by identifying indicators of compromise (IOCs), as well as anomaly detection using data stacking in a Windows environment. Its' capability to detect the presence of IOCs is tested in two scenarios, first in a simulated attack, and second after the introduction of malware.

Author Name, email@addresstimothy.hoffman.83@gmail.com

1. Introduction

The time it takes to detect, contain, and eradicate an incident directly impacts its financial cost. According to a study conducted by IBM and the Ponemon Institute, the average time to identify a breach in 2019 was 206 days, with another 73 days to contain it, for a total of 279 days from detection to eradication, with an average cost of $3.92 million (Ponemon Institute, 2019). This is a 4.9% increase from 2018, which had an average lifecycle of 266 days. Meaning, that although defensive capabilities and technology are improving, so are the adversaries' tactics and procedures.

Recent activities show evidence of an ever-evolving adversary. In a study covering the timeframe between October 2018 and September 2019, Mandiant, a leading provider of endpoint security products, researchers studied tens of thousands of malware samples, which consisted of 186 unique families. Of these families, the research team had never seen 41% before. Additionally, 46% of the samples functioned as backdoors, and 70% were introduced as portable executables (FireEye, 2020). Even if malware avoids initial detection, it needs to create changes on the system to establish persistence and often leaves files and other evidence behind. If analysts know what to look for, these files and alterations can be detected, and help identify the malicious activity.

To aid organizations in identifying, assessing, monitoring, and responding to cyber threats, they can participate in cyber threat information sharing. Threat information is "any information that might help an organization protect itself against a threat or detect the activities of an actor (Johnson, Badger, Waltermire, Snyder, & Skorupka, 2016). Threat information includes, but is not limited to, tactics, techniques, and procedures (TTPs), security alerts, threat intelligence reports, and indicators of compromise (IOCs). Sharing threat information has several benefits, and better prepares an organization to identify and protect against the attacks others have already experienced.

IOCs, which are forensic artifacts, indicating the potential presence of a compromise, or at a minimum, suspicious activity (Mertens, 2018) provide the groundwork for more advanced threat hunting. They can include any number of indicators, such as: IP Addresses, ports, DNS queries, services, processes, files, startup entries, and more. By sharing these IOCs in threat information sharing platforms,

organizations can query their networks to look for potential compromises that have happened in the past or are presently occurring. Allowing these compromises to be identified, contained, and eradicated, will help decrease the average timeframe and cost of a breach.

There are thousands of indicators available in databases and IOC feeds, but they are not all created equally, and some are more valuable than others when combating an adversary (Bianco, 2013). Per David Bianco's Pyramid of Pain, indicators such as hash values, IP addresses, and Domain names are easily changed by adversaries, making them less valuable than others. Tools and TTPs are more challenging to alter but also harder to identify. The higher up the pyramid defenders can operate, the more difficult it is for an adversary to avoid detection. Once at the TTP level, an adversary must either learn new behaviors and reinvent themselves or give up; this is the level at which organizations should strive to achieve (Bianco, 2013).

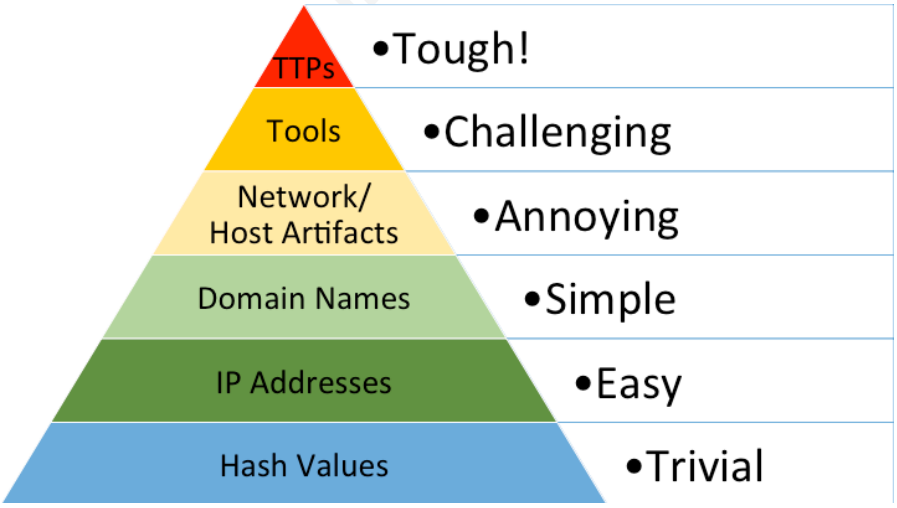


Figure 1 David Bianco's Pyramid of Pain (Bianco, 2013)

FireEye has released reports on several advanced persistent threat (APT) groups, highlighting their observed indicators to include file hashes and IPs as well as tools and TTPs. The most recent report on APT 41- Double Dragon, listed 71 easily detectable IOCs and numerous insights into the TTPs they use (FireEye, 2019). As an organization's threat hunting matures, its focus shifts from the bottom of the pyramid to the top. But failing to identify the low-level indicators will make it is impossible to operate at higher levels.

Author Name, email@addresstimothy.hoffman.83@gmail.com

In a threat hunting report by the SANS Institute, four levels of the threat hunting pyramid are identified. IOCs make up the first two levels, first as general IOCs, as previously discussed, and second as curated IOCs, which are indicators tailored to an environment.

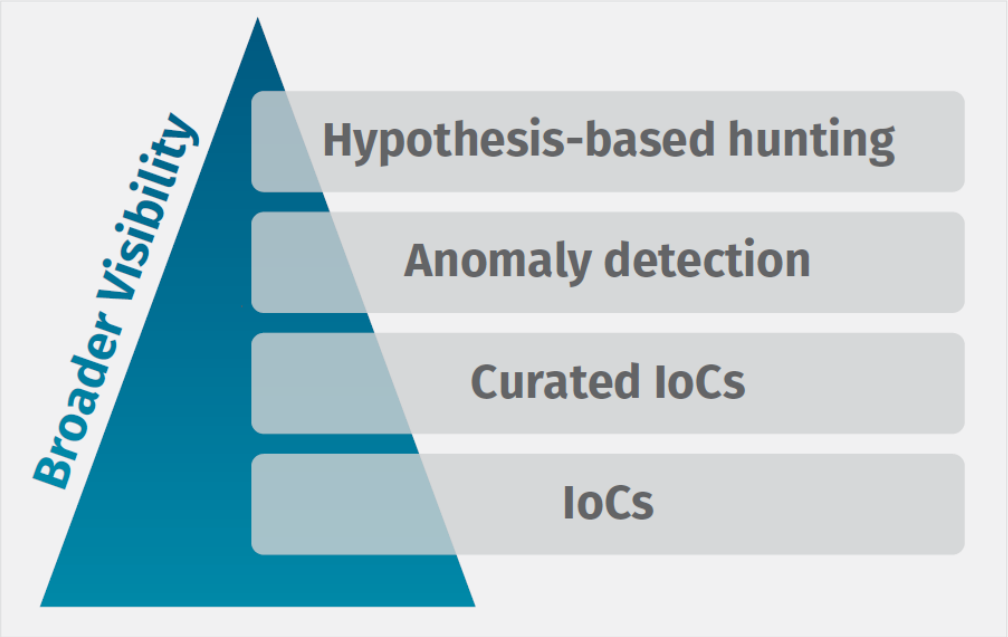


Figure 2 SANS Threat Hunting Maturity Pyramid (Fuchs, 2020)

The third level of the pyramid, anomaly detection, relies on identifying unusual activity in an environment. But for an organization to identify anomalies, they must first be able to determine what is normal and have a baseline configuration. A baseline does not merely include software, but also services, startup programs, running processes, and much more. Detecting a new application on an individual workstation may be a cause for concern, and likewise, discovering a new startup entry or service should also warrant further investigation. Anomalies do not guarantee the presence of an adversary. But anything outside of the baseline needs to be taken into consideration and analyzed appropriately.

Collecting data necessary to determine what is normal and what is unusual in an environment can be a challenge. OSQuery is one option, which, according to their website, "is a framework which exposes an operating system as a high-performance

database (OSQuery, n.d.)." It is a powerful tool, which can provide valuable insight into the environment. However, to unleash the true potential of OSQuery across an enterprise, it needs to be combined with other systems such as a Security Information and Event Management (SIEM) system, and translating the data into actionable intelligence can be complicated (Picotte, 2018).

Another option, The Eye of Sauron, is a PowerShell based tool, designed to provide insight into a Windows environment to identify anomalies and search for IOCs. It is simple to use and easily customizable, allowing organizations to tune it for their needs and look for the data that is most helpful for them. It works by collecting several data points from clients and analyzing the results collectively. The tool can be used in a case by case basis, or run daily, creating a history of the data to identify new and unusual activity better. This research will evaluate the effectiveness of the tool to identify changes made during a simulated attack campaign, as well as after the introduction of unknown malware samples within a lab environment.

2. Research Method

This research will test the effectiveness of the tools' ability to shine a light on potential alterations, which may indicate the presence of malware or an ongoing or completed attack campaign. The simulated attack will consist of initial execution, establishing persistence, lateral movement, and command and control elements. Upon completion of the entire attack process, the tool will run, and the post-attack results will be compared to pre-attack results to identify any indications that something malicious took place.

The lab environment consists of the following:

- Three Windows 10 client workstations
- One Windows Server 2019 (Domain controller and file server)
- One Kali Linux machine with Metasploit

The focus of this research is to identify potential IOCs, not to bypass security controls. Therefore, controls such as Windows Defender and Windows Firewall are disabled for the duration of the test. Initial compromise happens under a domain

Author Name, email@addresstimothy.hoffman.83@gmail.com

administrator account, simulating an administrator downloading and installing a portable executable file from a compromised website.

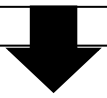
A second test with unknown, active malware will provide a blind test, as the behavior of the malware samples are unknown.

2.1. Simulated Attack

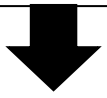
The simulated attack conducted during this project relates to the following techniques from the MITRE ATT&CK Framework and follows the steps in figure 3. MITRE ATT&CK is a knowledge base of adversary tactics and techniques based on real-world observations (MITRE, 2020). According to a study by FireEye, the majority of the methods used below are in the top five for their respective categories, indicating adversaries are using them often (FireEye, 2020).

The final stage is not related to the attack itself. Instead, using the tool to identify previously undetected compromises by searching for the IOCs already discovered. The ability to search for previous incidents is the benefit of maintaining a history of results, allowing newly identified indicators to be examined through past results, potentially finding previously undetected events.

Initial Compromise on WIN101
MITRE Phase: Execution
MITRE ID: T1204 – User Execution
Event: Run executable from compromised web site by administrator, installing malware on system.
Potential artifacts: new Executables, Processes



Persistence on WIN101
MITRE Phase: Persistence
MITRE ID: T1136 – Create Account
Event: Create a new local account and add to the local admins group. Additionally, create a new domain admin account to use for lateral movement.
Potential artifacts: local accounts and Administrator group members



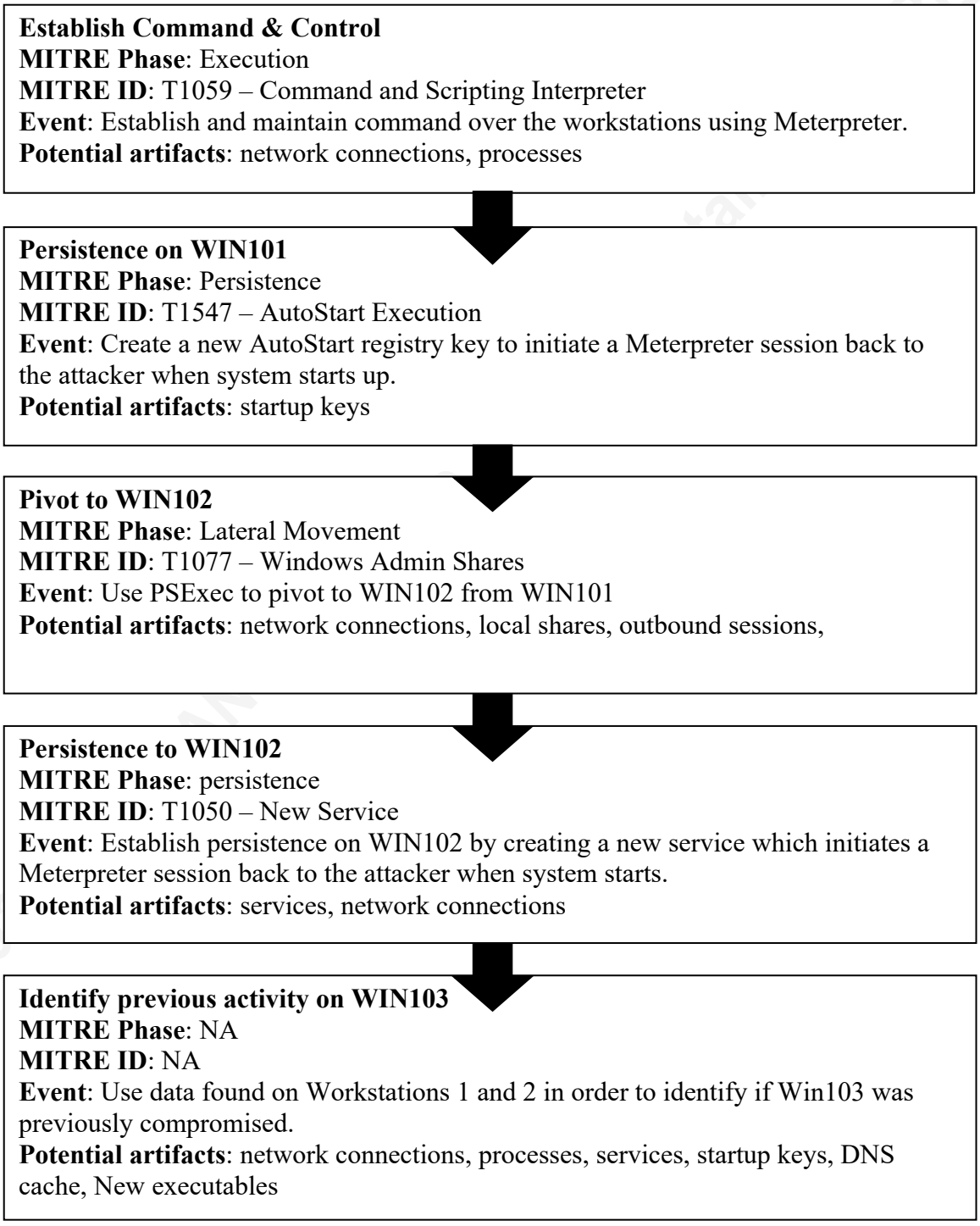


Figure 3 Attack Phases linked to MITR ATT&CK and detection categories of tool

2.2. Malware Samples

The second portion of the experiment introduces live malware into the environment on a single workstation. Afterward, indicators identifying the presence of

malware are detected using the tool. The malware samples used are from http://www.tekdefense.com, and appropriate measures are taken to ensure the malware does not leave the testing environment. This test simulates a drive-by download; the behavior of the malware is unknown at the beginning of the test.

2.3. Overview of The Eye of Sauron Tool

The Eye of Sauron is a tool created by Adam Clark, David Betteridge, and myself, consisting of three PowerShell scripts designed to collect and analyze data from Windows systems. There are two primary ways of using the tool. The first approach is to have it run daily, creating a record to compare recent findings to previous results to search for anomalies by identifying new activity. The second option is to deploy it in a one-time capacity to collect data from across an enterprise to aid in containing and eradicating a specific incident by searching for IOCs and using data stacking to identify anomalies. This research will attempt to identify changes by comparing previous results, as well as data stacking, to identify one-offs in the environment. The tool is available at https://github.com/Anubis876/EyeOfSauron.

The data sets collected by the tool fall into the classes in the below list. These classes are each saved in individual CSV files, and later summarized for enhanced querying. The capabilities of PowerShell enable the queries to be adjusted as necessary to meet the needs of the organization and collect only relevant data. Not all categories are designed for threat hunting and are instead intended more for compliance checks, such as installed Windows updates.

- Members of the Local Admin Group
- Local Accounts on system
- DNS cache
- Netstat details
- New Executables in the last 24hrs
- New Files in the previous 24hrs
- Current out Bound sessions
- Printers
- Running processes
- Installed applications
- Services
- Available Shares
- Startup entries in the registry
- Scheduled tasks
- Installed Windows updates
- Master (example in figure 4)

PSComputerName	Date	Username	IP	IPv6	Make	Model	SerialNumber	MAC	HddSizeGB	HddFreeGB
WIN101	7/7/2020	WIN101$	192.168.2.11	True	VMware, Inc.	VMware7,1	VMware-56 4...	00:0C...	59.39	40.63
WIN102	7/7/2020	WIN102$	192.168.2.12	True	VMware, Inc.	VMware7,1	VMware-56 4...	00:0C...	59.39	40.48
WIN103	7/7/2020	WIN103$	192.168.2.13	True	VMware, Inc.	VMware7,1	VMware-56 4...	00:0C...	59.39	38.67

OnDomain	Domain	LastBootTime	RebootRequired	Agent	ENS	DATNum	SCCM	LoggedOnUser	OperatingSystem	OSArchitecture	OSBuild	OSInstallDate
True	Lab.com	7/7/2020 5:47...	True	False	False	N/A	False	LAB\Administrator	Microsoft Windows 10...	64-bit	10.0.19041	6/25/2020 4:...
True	Lab.com	6/27/2020 1:3...	True	False	False	N/A	False	LAB\administrator	Microsoft Windows 10...	64-bit	10.0.19041	6/25/2020 4:...
True	Lab.com	7/7/2020 4:55...	True	False	False	N/A	False	LAB\Administrator	Microsoft Windows 10...	64-bit	10.0.19041	6/25/2020 4:...

Figure 4 Example Master CSV from the collection script

2.3.1. Collection Script

This script runs on each endpoint to collect the data points mentioned above. It uses windows management instrumentation (WMI) and registry queries to gather the majority of data and saves it to a specified share. Ideally, this script runs daily using a scheduled task or other means of automation, to provide a history of results for use in finding new additions. The Variables.txt file shares variables between all three scripts, preventing the need to edit them individually and ensuring consistency.

2.3.2. Summarization Script

The summarization script is run after all of the systems have run the collection script. This script has three functions; first, it summarizes the results gathered by the collection script and combines them into Summary CSV files. The second function is to compare the most recent results with the most recent previous results, exporting all new findings to an excel spreadsheet for easy analysis. The third function is to conduct maintenance, which deletes the individual system data and the summarized data according to the specified timeframes in the variables.txt file.

The longer the summary files are maintained, the better the capability to search for past findings and to find previously missed indicators. This history can also aid in identifying normal behavior, such as running processes, or established network connections. The Eye can combine results from multiple days to identify unusual activity, such as showing all processes that have run in the last seven days, or new local accounts detected during the previous 30 days.

2.3.3. User Interface Script

The user interface script is The Eye and is used to query the collected data for further analysis using a series of menus and pre-defined queries. The menu and query method allows people with little to no knowledge of PowerShell scripting to use the tool effectively. In contrast, those with PowerShell scripting knowledge can easily add additional queries to get their desired

results. Figure 5 shows the main menu of this script, and the local administrator accounts sub-menu, which is accessed by entering "1" from the main menu.

Figure 5 Eye of Sauron main menu and local administrator account menu

With the data saved in CSV files, there are numerous other options for accessing the data as well, such as additional PowerShell scripts or in a program like Microsoft Excel. However, depending on the amount of data to process, some queries can take a considerable amount of time to process. For example, searching for all network connections from the last 30 days in an environment of 2,400 workstations can take over an hour, while limiting the query to the most recent results takes minutes. The time is also significantly impacted by the available resources of the system performing the queries. By default, results are presented in an interactive grid view table. These results can be further filtered and sorted or copied to another application like Excel.

3. Detection Tests

Two tests were conducted to test the effectiveness of the tools' ability to detect IOCs and other anomalies. First, a simulated attack is performed against the three Windows 10 workstations using Metasploit and the steps in figure three. The second test introduces live malware into the environment and attempts to identify the alterations it creates. After each test, the Eye of Sauron is used to identify changes that occurred, and the workstations are returned to their original configuration using VMware snapshots.

3.1. Simulated Attack

This test follows the steps indicated in section 2.1. First, using msfvenom, an executable file is created using the below command. When run, this executable establishes a connection to a Metasploit listener on the attackers' machine.

msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.2.100 -f exe > /var/www/html/fun.exe

Using a Meterpreter shell, a local account is created using net user, and then added to the local administrators' group with the net group command. Additionally, a domain account is created and added to the domain administrators group using similar commands. Persistence is created using the *post/windows/manage/persistence_exe* Metasploit module configured as a user-based startup entry in the registry.

Using the created domain administrator account, the attack pivots to the second workstation using the PSEXEC module in Metasploit. Next, establishing persistence using the same module as before, but as a new service instead of a startup key. Lastly, creating a network share with staged malware for use in future attacks.

3.1.1. Analysis of Results

The following categories provided useful insight into the attack:

Processes. As seen in figure 6, when looking for new processes, there is a unique process called fun.exe, which is the name of the executable used to launch the malware. While knowing the name of the malware makes identifying it easier, not knowing that fun.exe was malware should still raise a red flag. It's an unusual name for a process and is running from a user's downloads directory, with the parent process browser_broker, which is part of the Edge browser.

PSComputerName	ProcessName	ParentProcessName	executablePath
WIN101	jusched.exe		C:\Program Files (x86)\Common Files\Java\Java Update\jusched.exe
WIN101	fun.exe	explorer	C:\Users\administrator\Downloads\fun.exe
WIN101	audiodg.exe	svchost	C:\Windows\system32\AUDIODG.EXE
WIN102	MoUsoCoreWorker.exe	svchost	C:\Windows\System32\mousocoreworker.exe
WIN102	powershell.exe		C:\Windows\syswow64\WindowsPowerShell\v1.0\powershell.exe
WIN103	csrss.exe	svchost	
WIN103	winlogon.exe	svchost	C:\Windows\system32\winlogon.exe

Figure 6 New Processes

The new processes also show PowerShell running on WIN102. Having PowerShell running is far from an anomaly, but looking at the processes running on WIN102 provides additional insight, to include the command line used by the process. These results, as seen in figure 7, show that the PowerShell process is far more interesting than initially thought—the payload consists of an encoded command, which in most environments would be highly unusual and warrant further investigation.

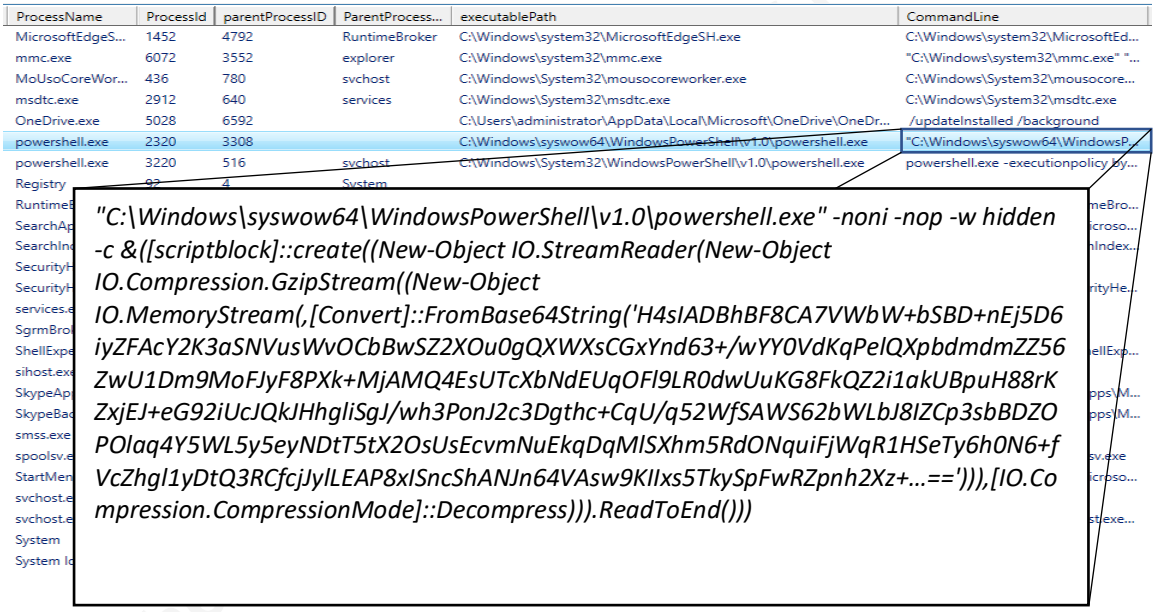


Figure 7 Processes running on WIN102 with a section of encrypted PowerShell command

Local account and administrator group. Detecting new local accounts is always a reason for concern, even more so if it is a member of the local administrators' group. The attack created two accounts, one a local account, and the second a domain account. The domain account will not be discovered by this tool as it only queries the local system.

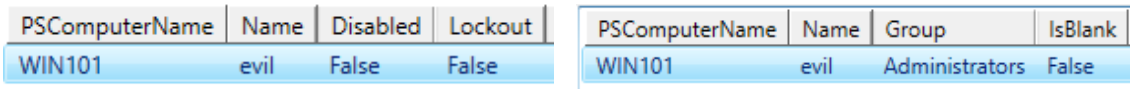


Figure 8 New Local Accounts and new members of the local administrators' group

An alternative option to looking for new accounts is to use data stacking to count the occurrence of every unique account, looking for the anomalies among them (Fuchs, 2020). This summary, in figure 9, shows five accounts existing on three workstations, with the *evil* account on only one system. Running a separate query shows which workstation has the "evil" user.

Count	Name	Disabled	Lockout
3	Guest	True	False
3	WDAGUtilityAccount	True	False
3	DefaultAccount	True	False
3	Admin	False	False
3	Administrator	True	False
1	evil	False	False

Figure 9 Summary of local accounts found in the environment

Services. New services created in the last 24 hours reveals one result, seen in figure 10, with an unusual name. Looking further into the service shows that it is running an executable from the \Windows\TEMP directory. No legitimate service would run from this directory or have a name like this, and it is a common location for malware to install because the logged-in user will have read-write access to the folder regardless of whether they are an administrator or not.

PSCom...	Date	DisplayName	Name	PathName	ProcessId	StartMode	State	Status
WIN102	7/7/2020	IKTrlbLG	gAgHgHXSuaUNfzW	cmd /c "C:\Windows\TEMP\default.exe"	0	Auto	Stopped	OK

Figure 10 Service created on WIN102 for persistence

Startup Items. When not related to the deployment of a new application, new startup entries are rare in an enterprise environment. Limit the creation to an individual or a small number of systems, and it becomes even more suspicious. Looking at the new startup entry on WIN101 in figure 11 shows it has an unusual name and runs from the user's temp directory. Additionally, the executable has the same name as the previously discovered service.

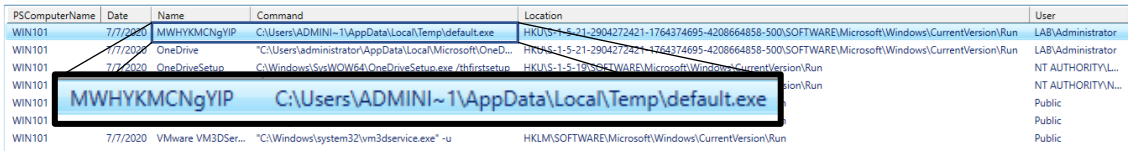


Figure 11 Startup entries on WIN101

When using data stacking to look at all startup entries discovered in the latest results in figure 12, two additional items with unusual names and only single instances are detected. They are configured to run from the same location but have different executable names.

Count	Name	Command	Location
3	OneDriveSetup	C:\Windows\SysWOW64\OneDriveSetup.exe /thfirstsetup	HKU\S-1-5-20\SOFTWARE\Microsoft\Windows\CurrentVersi...
3	SecurityHealth	%windir%\system32\SecurityHealthSystray.exe	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
3	VMware VM3DService...	"C:\Windows\system32\vm3dservice.exe" -u	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
3	VMware User Process	"C:\Program Files\VMware\VMware Tools\vmtoolsd.exe" -...	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
3	OneDrive	"C:\Users\administrator\AppData\Local\Microsoft\OneDri...	HKU\S-1-5-21-2904272421-1764374695-4208664858-500\S...
3	OneDriveSetup	C:\Windows\SysWOW64\OneDriveSetup.exe /thfirstsetup	HKU\S-1-5-19\SOFTWARE\Microsoft\Windows\CurrentVersi...
1	qduTUYCZhya	C:\Users\ADMINI~1\AppData\Local\Temp\evil.exe	HKU\.DEFAULT\SOFTWARE\Microsoft\Windows\CurrentVersi...
1	qduTUYCZhya	C:\Users\ADMINI~1\AppData\Local\Temp\evil.exe	HKU\S-1-5-18\SOFTWARE\Microsoft\Windows\CurrentVersi...
1	AdobeGCInvoker-1.0	"C:\Program Files (x86)\Common Files\Adobe\AdobeGCCli...	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
1	MWHYKMCNgYIP	C:\Users\ADMINI~1\AppData\Local\Temp\default.exe	HKU\S-1-5-21-2904272421-1764374695-4208664858-500\S...
1	AdobeAAMUpdater-1.0	"C:\Program Files (x86)\Common Files\Adobe\OOBE\PDA...	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Figure 12 All startup items discovered

A third query to find workstations with the "qduTUYZhya" startup item reveals these two additional startup entries are on WIN103, indicating that the workstation was previously compromised.

PSComputerName	Date	Name	Command	Location	User
WIN103	7/7/2020	qduTUYCZhya	C:\Users\ADMINI~1\AppData\Local\Temp\evil.exe	HKU\.DEFAULT\SOFTW...	.DEFAULT
WIN103	7/7/2020	qduTUYCZhya	C:\Users\ADMINI~1\AppData\Local\Temp\evil.exe	HKU\S-1-5-18\SOFTW...	NT AUTHORITY\SYSTEM

Figure 13 Query for "qduTUYCZhya" startup item

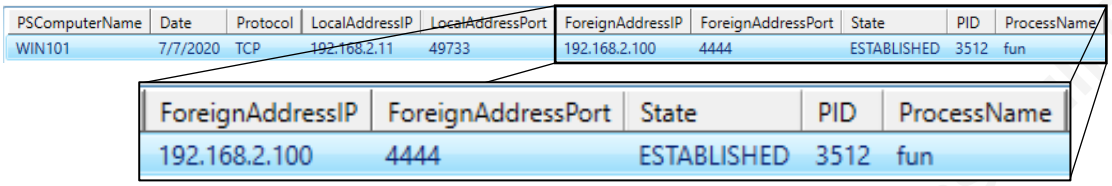
Executable Files. Having found startup items and services with unusual characteristics, and suspicious files, a query is run to see executable files discovered in the last 30 days. These results identify fun.exe was found on two systems, while evil and default were each located on one. It also shows that evil and default are the same file, based on the md5 hash. Further analysis of the files would be needed to verify if they are malicious.

Count	Name	Directory	Filehash
2	fun.exe		D9FC0C17D8405C923DB27709154BA1DF34F3960B
1	evil.exe		C744B4DC934587750442ED3D54316A43C3969336
1	default.exe		C744B4DC934587750442ED3D54316A43C3969336

Figure 14 Detected executables in the last 30 days

Network Stats. Querying network connections identifies some unusual network behavior as well. This traffic was made more notable by using the Meterpreter default port 4444, using ports 80, 443, or other common ports instead would make identifying the network connection by port or IP considerably more difficult due to blending in with regular traffic. To better simulate this, the connections are queried by the process name establishing the connection, showing the fun.exe process having an outbound connection.

Author Name, email@addresstimothy.hoffman.83@gmail.com



PSComputerName	Date	Protocol	LocalAddressIP	LocalAddressPort	ForeignAddressIP	ForeignAddressPort	State	PID	ProcessName
WIN101	7/7/2020	TCP	192.168.2.11	49733	192.168.2.100	4444	ESTABLISHED	3512	fun

ForeignAddressIP	ForeignAddressPort	State	PID	ProcessName
192.168.2.100	4444	ESTABLISHED	3512	fun

Figure 15 "fun" process established outbound connection

A second query based on the foreign IP of the first query (seen in figure 15) reveals additional connections established to it. The results, seen in figure 16, show all three workstations have had connections established to this IP. However, based on the date, WIN103 has not communicated within the last 24 hours. Additionally, there is an SMB connection (local port 445) from the external host to WIN102, demanding further investigation.

PSCom...	Date	..	LocalAddressIP	LocalAddressPort	ForeignAddressIP	ForeignAddressPort	State	PID	ProcessName
WIN103	6/28/2020	...	192.168.2.13	49890	192.168.2.100	4444	ESTA...	5736	fun
WIN103	7/6/2020	...	192.168.2.13	49890	192.168.2.100	4444	ESTA...	5736	fun
WIN101	7/7/2020	...	192.168.2.11	49733	192.168.2.100	4444	ESTA...	3512	fun
WIN102	7/7/2020	...	192.168.2.12	445	192.168.2.100	38972	ESTA...	4	System
WIN102	7/7/2020	...	192.168.2.12	445	192.168.2.100	38980	ESTA...	4	System
WIN102	7/7/2020	...	192.168.2.12	54534	192.168.2.100	4444	ESTA...	2320	powershell

Figure 16 All network connections to 192.168.2.100

Local shares. The identification of an SMB connection to WIN102 warrants a query to identify what local shares are on the workstation. The results show an unauthorized share with the path *c:\share*, which, when accessed, contains the default.exe executable.

PSComputerName	Date	Name	Path	Descrition
WIN102	7/7/2020	ADMIN$	C:\Windows	
WIN102	7/7/2020	C$	C:\	
WIN102	7/7/2020	IPC$		
WIN102	7/7/2020	share	c:\share	

Figure 17 query of shares on WIN102

3.1.2. Summary of results

In this experiment, the tool was able to find all changes to the local systems, and provide proof of a compromise. The one item it is unable to detect is the creation of a domain account, which is by design. However, several situations could affect these results. The tool is capturing a moment in time, so deleting files, stopping processes, and killing network connections would prevent the tool from detecting them. The creation of startup keys in the registry and services for persistence would be much harder to hide but could be named less conspicuously to avoid

detection. However, a close analysis of new items would still produce a high probability of discovering them.

Another factor that can severely hamper the tools' ability to identify anomalies is configuration management. Poor configuration management makes identifying unauthorized changes and abnormalities considerably harder and results in more false positives. In contrast, strict configuration management makes these anomalies even more apparent. Apart from threat hunting, this tool can also help audit and enforce configuration management.

This tool does not provide a clear picture or roadmap to what happened in its entirety, only that something happened. Further investigation into these machines and analysis of the files is required to obtain an in-depth understanding of everything that happened. The findings do provide critical items such as file names, hashes, and external IPs, to narrow the search of logs and other security tools.

3.2. Introduction of Malware

For this portion of the experiment, random malware is obtained from tekdefense.com and installed on a workstation. The malz6.zip and tekdefense.7Z are downloaded and run from http://www.tekdefense.com/downloads/malware-samples/. After installing the malware, the workstation is restarted, and the Eye of Sauron is used to collect and analyze modifications created by the malware.

3.2.1. Analysis of Results

Using the Eye of Sauron, the results of the test are analyzed. Unlike the previous experiment, the actions of the malware are unknown, providing a more likely real-world scenario. The analysis of these results uncovered the following:

New Local Accounts. There is a new local account on the system with the username TekDefense. This account is not a member of the local administrators' group, and further research into logs is necessary to identify additional information on it.

PSComputerName	Name	Disabled	Lockout
WIN101	TekDefense	False	False

Figure 18 New local account after installation of malware

New Processes. There are several new processes, shown in figure 19, on the system which warrant further investigation. Two, in particular, stick out as malicious immediately due to attempting to imitate legitimate processes with slightly changed names (lssas.exe and svchsot.exe instead of lsass.exe and svchost.exe). The other three have unusual names and paths, indicating they are most likely malicious as well.

PSComputerName	ProcessName	ParentProcessName	executablePath
WIN101	lkpfye.exe	services	C:\Windows\lkpfye.exe
WIN101	lssas.exe	explorer	C:\Windows\SysWOW64\lssas.exe
WIN101	svchsot.exe		C:\Windows\XXXXXX2063534F\svchsot.exe
WIN101	xm.exe	explorer	C:\ProgramData\Microsoft\Windows\Start Menu\
WIN101	mymqmy.exe	services	C:\Windows\mymqmy.exe

Figure 19 New processes on WIN101 after installing malware

New executables. Further investigation into the executables responsible for the new services provides additional details to include the md5 hash of the file. The results also contain the last write time (removed from figure 20), which can give some indication on when the incident occurred. Identifying if these files are malicious can be accomplished by searching for them on a site like Virus Total.

Executable Files on win101 - Latest

Filter

Add criteria

PSComputerName	Date	Name	Directory	FileHash
WIN101	7/21/2020	7z.exe	C:\Program Files (x86)\7-Zip	E8DCDDB302F01D51DA3BCBFA6707D025A896AA57
WIN101	7/21/2020	7zFM.exe	C:\Program Files (x86)\7-Zip	DEC2A6DC8E8CD77F7770330411C2B0AF9AA20C6C
WIN101	7/21/2020	7zG.exe	C:\Program Files (x86)\7-Zip	E71D982B30CBB40CA90426B48BB9B32766339 2B6
WIN101	7/21/2020	DesktopLayer.exe	C:\Program Files (x86)\Microsoft	D03FA4A202AFC929725F969471A8C1D5943FD12A
WIN101	7/21/2020	fgdump.exe	C:\Users\administrator\AppData\Local\Temp	72C99FC933A165D3F9DD050EFEC8EC370EB967E0
WIN101	7/21/2020	FileCoAuth.exe	C:\Users\administrator\AppData\Local\Microsoft\OneDrive\20.114.0607.0002	FB9551544C75C1C66B19D43BC3A61E089FB1FBFF
WIN101	7/21/2020	FileSyncConfig.exe	C:\Users\administrator\AppData\Local\Microsoft\OneDrive\20.114.0607.0002	4B07BFBE2C02C198FFF4F9DCD31AEA021F86A68B
WIN101	7/21/2020	FileSyncHelper.exe	C:\Users\administrator\AppData\Local\Microsoft\OneDrive\20.114.0607.0002	F9EF20B86601492C0091992F32A2F7D7AB8D64C9
WIN101	7/21/2020	lkpfye.exe	C:\Windows	F09D773D50F87C47500305E79BC1E6FCF503EBD6
WIN101	7/21/2020	lssas.exe	C:\Windows\SysWOW64	91690D293BC2ECF4C77F46954D93B990269D8C0F
WIN101	7/21/2020	mymqmy.exe	C:\Windows	076D233EF06971A64F9B009C03627A491444A422
WIN101	7/21/2020	nc.exe	C:\Users\administrator\AppData\Local\Temp	032815CB1942F08B697C381C38037C5CC253B0B2
WIN101	7/21/2020	OneDriveSetup.exe	C:\Users\administrator\AppData\Local\Microsoft\OneDrive\20.114.0607.0002	E63AA06AAAB5404F870D3A34C3F8DE707EADA013
WIN101	7/21/2020	OneDriveUpdaterService.exe	C:\Users\administrator\AppData\Local\Microsoft\OneDrive\20.114.0607.0002	6EB3B11AF34301C50051E70F37044223653 39D8D
WIN101	7/21/2020	smssSrv.exe	C:\Users\administrator\Downloads\MALZ6	D03FA4A202AFC929725F969471A8C1D5943FD12A
WIN101	7/21/2020	svchsot.exe	C:\Windows\XXXXXX2063534F	D02D2D83B9887BFC12E3A3E47E6A700E68406E6B
WIN101	7/21/2020	syringe.exe	C:\Users\administrator\AppData\Local\Temp	032815CB1942F08B697C381C38037C5CC253B0B2
WIN101	7/21/2020	Uninstall.exe	C:\Program Files (x86)\7-Zip	D7793605D6B7ABD16A735089DAE0A8DE0EB7079F

Figure 20 new executable files after malware is installed. Highlighted items are suspicious.

The hash of lssas.exe in VirusTotal verifies the file is malicious. It also provides additional behaviors that can be searched for and correlated to the event, such as IPs, domain names, and other file names. A second example is the lkpfye.exe file, which VirusTotal says communicates with ilo.brenz.pl at IP 148.81.111.121, shown in figure 21. Searching for these

entries in the netstat and DNS portions of the tool identifies similar behavior, which will be covered shortly.

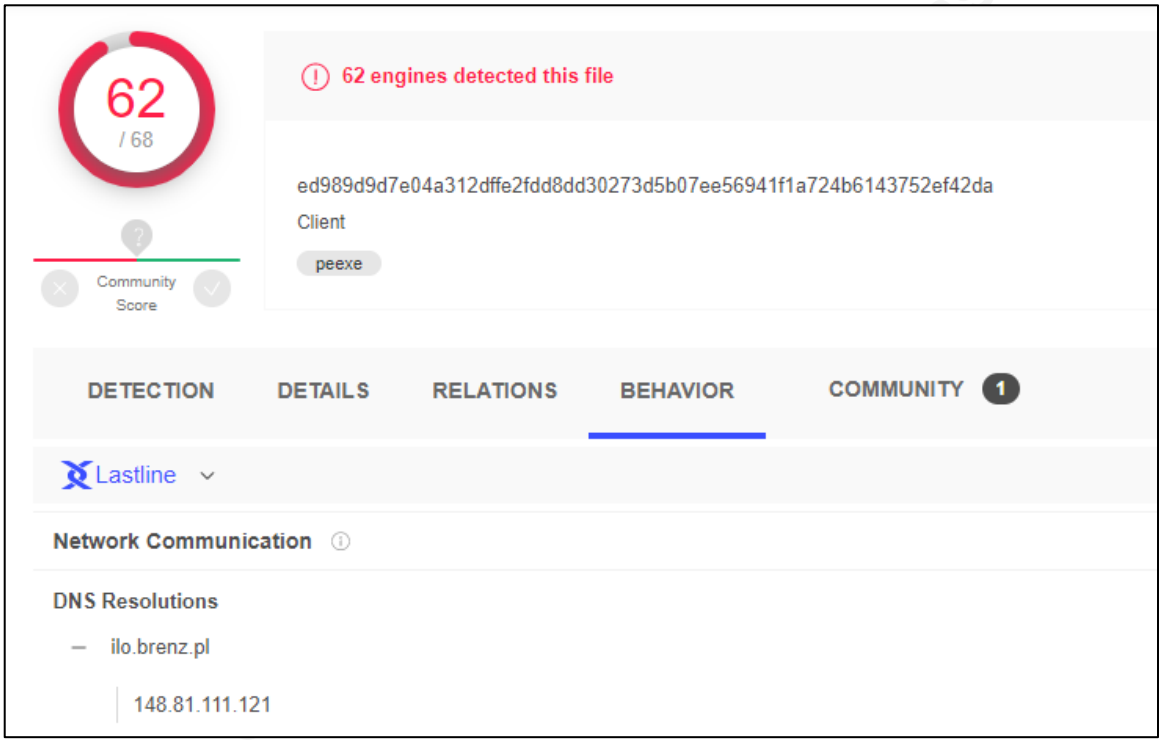


Figure 21 lkpfye.exe results from VirusTotal, showing a domain name and IP it establishes connections with.

Startup entries. Searching for new startup entries returns two results in figure 22, both look suspicious due to random names. The first starts the lssas.exe process, which has already been identified as malicious while the second entry is for rqxm.exe. Previous queries have not defined this file, and further investigation on the workstation reveals the file was created on May 21, 2015. The creation date explains why it was not discovered as a new file because the collection script queries by creation date and not modified date, indicating this file was not created but moved from a different location. Researching the hash of this file in VirusTotal verifies it is malicious and resolves the same ilo.brenz.pl DNS entries, meaning it is most likely related to the lkpfye executable.

PSComputerName	Name	Command	Location
WIN101	denwz	C:\Windows\SysWOW64\lssas.exe	HKU\S-1-5-21-2904272421-1764374695-420
WIN101	rqxm	rqxm.exe	Common Startup

Figure 22 new startup entries after installing malware

Services. There are three new services which are suspicious, shown in figure 23. Two of these services are using executables previously identified as questionable, while the third, rgvrixx, runs "*svchost.exe -k netsvcs*" and does not appear to belong. Further research on the host would be required to determine additional information on this service.

PSComputerName	Date	DisplayName	Name	PathName
WIN101	7/21/2020	rgevrixx	fastuserswitchingcompatibility	C:\Windows\System32\svchost.exe -k netsvcs
WIN101	7/21/2020	Hello Mask,this is the demo	Hello Mask,this is the demo	C:\Users\administrator\Downloads\MALZ6\smss.exe
WIN101	7/21/2020	Mniopqr Tuvwxyab Defghijk Mnop	Mniopqr Tuvwxyab Def	C:\Windows\mymqmy.exe

Figure 23 Suspicious services after installing malware

Network connections. The results from the network connections in figure 24 show three entries of concern, one each from lkpfye.exe, lssas.exe, and svchsot.exe. Lkpfye has established a connection with 148.81.111.121, which VirusTotal has listed, but the other two IPs were not mentioned.

Protocol	LocalAddressIP	LocalAddressPort	ForeignAddressIP	ForeignAddressPort	State	PID	ProcessName
TCP	0.0.0.0	49671	0.0.0.0	0	LISTENING	1852	spoolsv
TCP	192.168.1.44	139	0.0.0.0	0	LISTENING	4	System
TCP	192.168.1.44	51752	148.81.111.121	80	ESTABLISHED	1292	lkpfye
TCP	192.168.1.44	49786	192.186.157.43	6667	ESTABLISHED	6000	lssas
TCP	192.168.1.44	51117	192.229.221.185	443	ESTABLISHED	3144	OneDrive
TCP	192.168.1.44	49755	23.201.210.161	443	ESTABLISHED	3516	svchost
TCP	192.168.1.44	49757	23.201.211.174	80	ESTABLISHED	3516	svchost
TCP	192.168.1.44	49784	40.90.137.124	443	ESTABLISHED	3144	OneDrive
TCP	192.168.1.44	49732	52.114.128.73	443	ESTABLISHED	2036	OfficeClickToRun
TCP	192.168.1.44	49753	52.114.128.9	443	TIME_WAIT	0	Idle
TCP	192.168.1.44	52986	58.252.3.169	6380	SYN_SENT	5204	svchsot
TCP	192.168.2.11	139	0.0.0.0	0	LISTENING	4	System

Figure 24 Network connections after installing malware

DNS queries. DNS queries shown in figure 25 returned two suspicious entries as well. Both of these entries resolve to the same IP and are listed in VirusTotal again for the lkpfye file.

PSComputerName	Entry	DataLength	Data
WIN101	cghj3322org.eicp.net	4	0.0.0.0
WIN101	ant.trenz.pl	4	148.81.111.121
WIN101	ilo.brenz.pl	4	148.81.111.121

Figure 25 Suspicious DNS cached entries after installing malware

Further research into the ant.trenz.pl domain on threatstop.com in figure 26 indicates it is a high threat domain name and used maliciously, verifying the data obtained from VirusTotal.

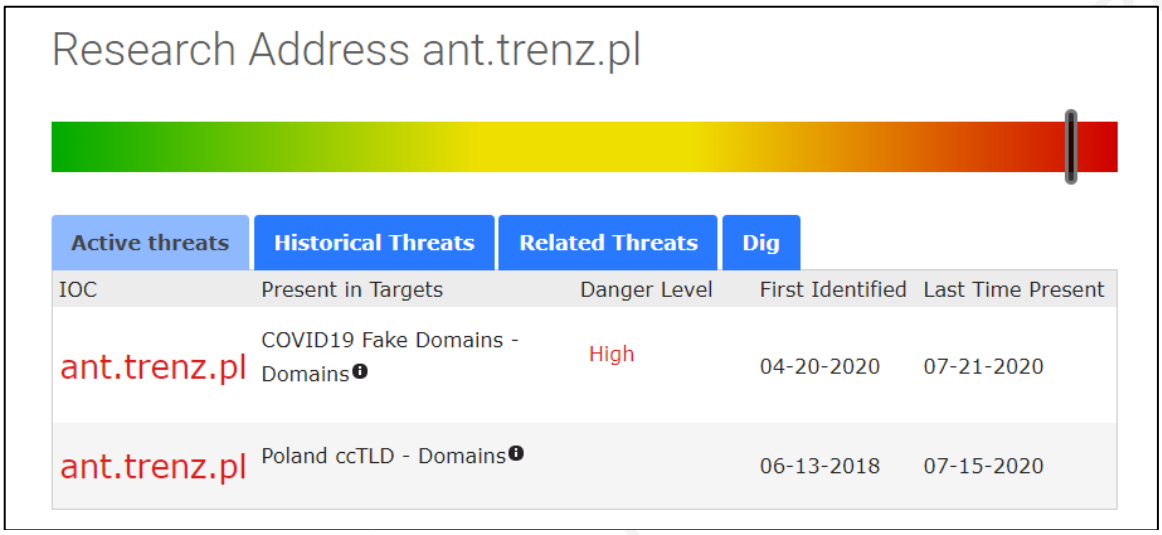


Figure 26 ThreatStop.com analysis of ant.trenz.pl domain

3.2.2. Summary of results

In this test, the tool discovered multiple indicators that warrant further investigation from numerous categories. Correlating the data from multiple queries provides a clearer picture and increases the suspicion of malicious activities. Individual indicators can be overlooked by an analyst or missed entirely by the tool, such as the rqxm.exe file not being detected as a new file. Collectively, they provide strong evidence of malware.

These results clearly show suspicious activities occurred, but once again is not enough to put all of the pieces together. Further analysis of logs and network data would be required. The tool has provided processes, file names and hashes, IPs, and DNS entries, for further investigation.

4. Analysis

In each scenario, the tool identified anomalies that indicated potential malicious activity. The tool is not designed to, nor does it, provide an in-depth analysis of these findings. Instead, it attempts to shine a light on new or suspicious items and provide a means to further search an environment for a specific IOC, such as a file hash. Multiple times during this research, the tool was modified to provide refined results, advertising its flexibility, and a defender's ability to alter it to their needs.

In a small environment of only three systems, it can be hard to identify anomalies. When used in organizations with hundreds of workstations, the results become clearer to read as long as

configuration control is in place. In an environment with no configuration control, the usefulness of the tool decreases drastically, as its ability to identify what doesn't belong requires having a common baseline.

4.1. Further enhancements

Further enhancements to the tool are necessary, such as the ability to pull IOCs from a database and search for them. In this experiment, finding anomalies and new items were more fruitful than looking for individual indicators would have been. Continued refinement of the tool is necessary, as well as more advanced tests to better determine its capabilities. The performance of queries is also a concern. Even in a small environment, some queries began to take some time; for example, thirty-plus minutes to return a query for new files created in the last 24 hours. In a large environment, running a query like this is not feasible due to the time it takes for results. Using a database instead of CSV files for storage is one potential possibility for increased performance and capabilities.

5. Conclusion

Monitoring client workstations is vital, as they are the gateway into an organization. For large environments collecting logs and processing them through a SIEM can be a massive undertaking requiring an enormous amount of storage and processing power. "A Fortune 500 enterprise can generate ten terabytes of plain-text log data per month" (Constantine, 2018). Due to the large amount of data required to be collected, often client logs are not included. Instead, defenders rely on what the network and servers provide, potentially creating a blind spot, where a careful adversary could navigate and collect data for a significant amount of time without being detected, as shown by the average time it takes to identify a breach.

For organizations that have this blind spot, the Eye of Sauron provides some help by automating the collection and providing analysis of client data. Without a tool like this, "security teams have to find a way to go into each endpoint and gather data manually or buy a third-party tool to do it" (Mello, 2019). Although not as powerful as OSQuery, it can be quickly and easily implemented, and provide immediate insight into an environment. It has proven useful in discovering anomalies, IOCs, and changes in general during this experiment. It is one more tool for Cyber Defenders to use as necessary to defend their networks better.

Author Name, email@addresstimothy.hoffman.83@gmail.com

References

Bianco, D. (2013, March 1). The Pyramid of Pain. Retrieved July 24, 2020, from https://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html

Constantine, C. (2018, December 18). Standards and Best Practices for SIEM Logging. Retrieved July 22, 2020, from https://cybersecurity.att.com/blogs/security-essentials/what-kind-of-logs-for-effective-siem-implementation

FireEye. (2019). *Double Dragon, APT41, a dual espionage and cyber crime operation* (Tech). FireEye

FireEye. (2020). *M-TRENDS 2020* (Rep.). Retrieved July 19, 2020, from https://content.fireeye.com/m-trends/rpt-m-trends-2020

Fuchs, M. (2020). *Is Your Threat Hunting Working? A New SANS Survey for 2020* (Rep.). SANS.

Gunter, D., & Seitz, M. (2018, November 29). *A Practical Model for Conducting Cyber Threat Hunting* (Tech.). Retrieved July 26, 2020, from https://www.sans.org/reading-room/whitepapers/threathunting/practical-model-conducting-cyber-threat-hunting-38710

Johnson, C., Badger, L., Waltermire, D., Snyder, J., & Skorupka, C. (2016). *Guide to Cyber Threat Information Sharing* (800-150) (United States, National Institute of Standards and Technology). NIST.

Mello, J. P. (2019, January 22). How osquery can lift your security team's game. Retrieved from https://techbeacon.com/security/how-osquery-can-lift-your-security-teams-game

Mertens, X. (2018, April 24). The real value of an IOC? Retrieved July 14, 2020, from https://isc.sans.edu/forums/diary/The+real+value+of+an+IOC/23585/

MITRE. (2020). MITRE ATT&CK®. Retrieved July 24, 2020, from https://attack.mitre.org/

Osquery. (n.d.). Welcome to osquery. Retrieved July 15, 2020, from https://osquery.readthedocs.io/en/stable/

Picotte, A. (2018, July 12). Intro to Osquery: Frequently Asked Questions for Beginners. Retrieved July 15, 2020, from https://www.uptycs.com/blog/intro-to-osquery-frequently-asked-questions-for-beginners

Ponemon Institute LLC. (2019). *Cost of a Data Breach Report* (Tech.). IBM Security.

Author Name, email@addresstimothy.hoffman.83@gmail.com