



HACKTHEBOX

CampFire-1 Writeup



Prepared by: Cyberjunkie & Sebh24

Machine Author(s): Cyberjunkie

Difficulty: **Very Easy**

Scenario

Alonzo Spotted Weird files on his computer and informed the newly assembled SOC Team ASAP. Assessing the situation it is believed Kerberoasting attack may have occurred in the network. It is your Job to confirm the findings by analyzing the provided evidence. You are given

- 1-Security Logs from the Domain Controller
- 2- PowerShell-Operational Logs from the affected workstation
- 3- Prefetch Files from affected workstation

Artefacts provided

1- CampFire1.zip , sha1 : 089B7C3241C43F74E61C83488910F7520C71CAF3



Skills Learnt

- Active Directory Attacks Anlysis
- Timeline creation
- Kerberoasting Analysis

- Windows Event log analysis
- Windows Forensics
- Chaining of multiple Artefacts

Initial Analysis :

Lets see the data we are provided.

 Domain Controller	5/21/2024 10:03 AM	File folder
 Workstation	5/21/2024 10:03 AM	File folder

Upon taking a look, we find that we have security logs from the domain controller, and PowerShell logs and prefetch files from the workstation.

What are prefetch files?

Prefetch speeds up the loading of a specific application resource, allowing you to open your most used application faster. Prefetching enables a browser to fetch the resources required to view content that will be accessed later. Prefetch files will disclose whether the individual installed and ran a particular program; tracking such information is critical during the digital forensic analysis process. This way, we can determine which executable was executed and when. Prefetch also records the loaded files' information, which tells us which files and paths it interacted with during its execution.

We will use PeCmd by Eric Zimmerman to parse the prefetch files and Event Viewer to go through the event logs.

Let's parse the prefetch to make it analysis-ready.

```
CSV file output will be saved to .\analysis_timeline.csv
PS F:\Forensics Tools\KAPE\Modules\bin> .\PECmd.exe -d "H:\Project-Sherlock\Camp Fire 1\Triage\Workstation\2024-05-21T033012-triage_asset\C\Windows\prefetch" --csv . --csvf analysis.csv
```

We use PeCmd to parse the provided prefetch files and save it in the current directory with the name `analysis.csv`.

- `PECmd.exe -d "path-to-prefetch-files" --csv . --csvf outputfilename.csv`

We will analyze the CSV file with Timeline Explorer in the later part of our analysis.

Note : We have prepared a [mini blog](#) for our audience where we go over kerberoasting attack and this exact scenario in detail on how to detect and analyze this attack's activity

Analysis

Q1 Analyzing Domain Controller Security Logs, can you find the timestamp when the kerberoasting activity occurred?

Hint: In Security Logs, Filter for Event ID 4769. Now Look for any event where the service name is NOT(krbtgt or ends with \$ (For e.g DC01\$)). The ticket type should be 0x17 which is for RC4 type encryption. The failure code should be 0x0. The event that matches all above conditions is the event detailing information about the kerberoasting attack activity.

Let's start off by opening the security log file. Initially, we see lots of events (these are still very boiled down for this very easy investigation; in real life, there would be thousands or millions).

SECURITY-DC_1 Number of events: 293				
Level	Date and Time	Source	Event ID	Task Category
Information	5/21/2024 8:21:25 AM	Microsoft Windows security auditi...	4771	Kerberos Authentication Service
Information	5/21/2024 8:21:25 AM	Microsoft Windows security auditi...	4771	Kerberos Authentication Service
Information	5/21/2024 8:21:19 AM	Microsoft Windows security auditi...	4771	Kerberos Authentication Service
Information	5/21/2024 8:21:19 AM	Microsoft Windows security auditi...	4771	Kerberos Authentication Service
Information	5/21/2024 8:21:16 AM	Microsoft Windows security auditi...	5140	File Share
Information	5/21/2024 8:21:15 AM	Microsoft Windows security auditi...	5140	File Share
Information	5/21/2024 8:21:13 AM	Microsoft Windows security auditi...	4771	Kerberos Authentication Service
Information	5/21/2024 8:21:13 AM	Microsoft Windows security auditi...	4771	Kerberos Authentication Service
Information	5/21/2024 8:21:07 AM	Microsoft Windows security auditi...	4771	Kerberos Authentication Service
Information	5/21/2024 8:21:07 AM	Microsoft Windows security auditi...	4771	Kerberos Authentication Service
Information	5/21/2024 8:21:01 AM	Microsoft Windows security auditi...	4771	Kerberos Authentication Service
Information	5/21/2024 8:21:01 AM	Microsoft Windows security auditi...	4771	Kerberos Authentication Service
Information	5/21/2024 8:21:01 AM	Microsoft Windows security auditi...	4771	Kerberos Authentication Service
Information	5/21/2024 8:21:01 AM	Microsoft Windows security auditi...	4771	Kerberos Authentication Service
Information	5/21/2024 8:20:55 AM	Microsoft Windows security auditi...	4771	Kerberos Authentication Service
Event 4771, Microsoft Windows security auditing.				

According to the hint provided in the question, we want to look for events where the encryption type is 0x17, the service name does not end with \$, and is neither krbtgt. Let's first start by filtering for event ID 4769.

We still get lots of events.

SECURITY-DC_1

Number of events: 293

Filtered: Log: file://H:\Project-Sherlock\Camp Fire 1\Triage\Domain Controller\SECURITY-DC.evtx; Source: ; Event ID: 4769. Number of events: 16

Level	Date and Time	Source	Event ID	Task Category
Information	5/21/2024 8:20:24 AM	Microsoft Windows security auditi...	4769	Kerberos Service Ticket Operations
Information	5/21/2024 8:18:51 AM	Microsoft Windows security auditi...	4769	Kerberos Service Ticket Operations
Information	5/21/2024 8:18:09 AM	Microsoft Windows security auditi...	4769	Kerberos Service Ticket Operations
Information	5/21/2024 8:15:12 AM	Microsoft Windows security auditi...	4769	Kerberos Service Ticket Operations
Information	5/21/2024 8:13:02 AM	Microsoft Windows security auditi...	4769	Kerberos Service Ticket Operations
Information	5/21/2024 8:12:05 AM	Microsoft Windows security auditi...	4769	Kerberos Service Ticket Operations
Information	5/21/2024 8:12:05 AM	Microsoft Windows security auditi...	4769	Kerberos Service Ticket Operations
Information	5/21/2024 8:06:15 AM	Microsoft Windows security auditi...	4769	Kerberos Service Ticket Operations
Information	5/21/2024 8:06:15 AM	Microsoft Windows security auditi...	4769	Kerberos Service Ticket Operations
Information	5/21/2024 8:05:54 AM	Microsoft Windows security auditi...	4769	Kerberos Service Ticket Operations
Information	5/21/2024 8:05:54 AM	Microsoft Windows security auditi...	4769	Kerberos Service Ticket Operations
Information	5/21/2024 8:05:54 AM	Microsoft Windows security auditi...	4769	Kerberos Service Ticket Operations
Information	5/21/2024 8:05:54 AM	Microsoft Windows security auditi...	4769	Kerberos Service Ticket Operations

Event 4769, Microsoft Windows security auditing.

General

Details

A Kerberos service ticket was requested.

Account Information:

Account Name:

DC01\$@FORELA.LOCAL

Account Domain:

FORELA.LOCAL

Logon GUID:

{8c14bbd3-6291-06a0-ae34-e9ac6b84defe}

Service Information:

Service Name:

DC01\$

Service ID:

S-1-5-21-3239415629-1862073780-2394361899-1000

Network Information:

Client Address:

::1

Client Port:

0

Additional Information:

Ticket Options:

0x40800000

Ticket Encryption Type:

0x12

Failure Code:

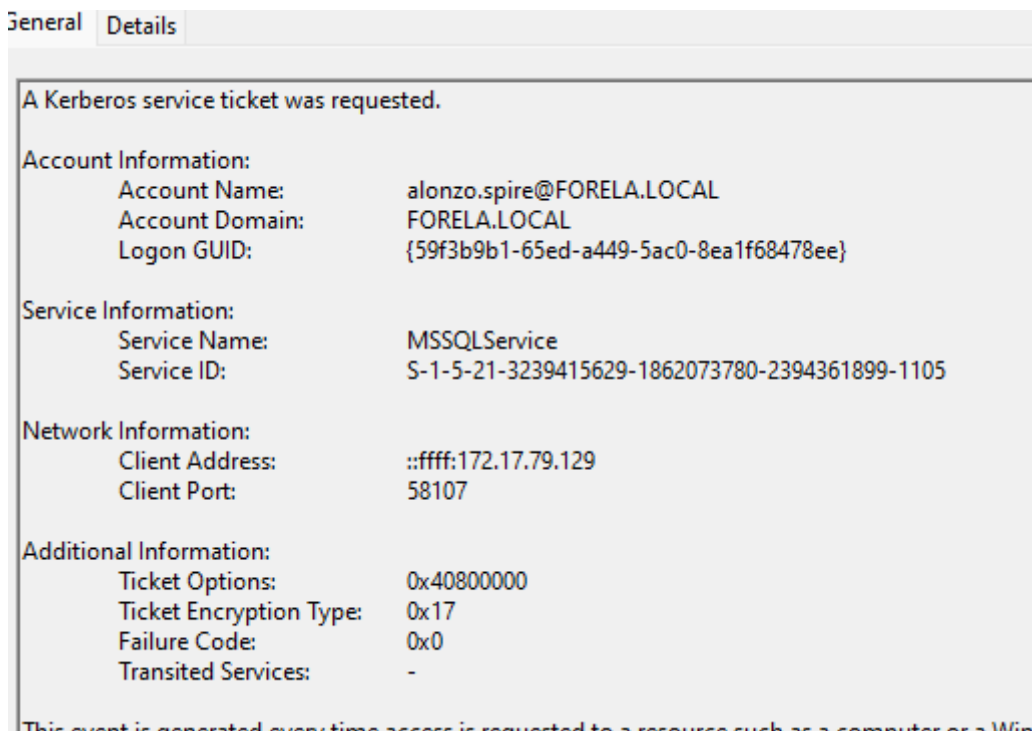
0x0

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

We can search for the keyword 0x17. We stumble upon an event that matches our criteria. The timestamp of this event is our answer.



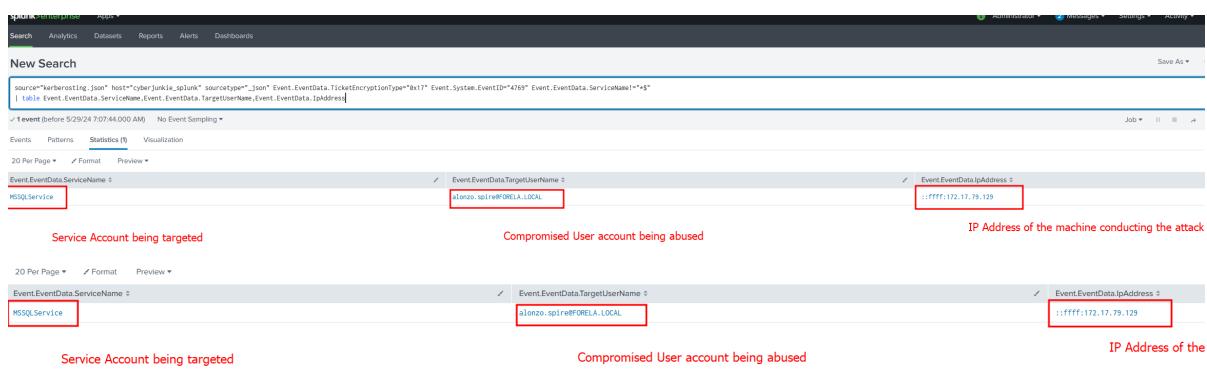
Let's also go over an SIEM example and how this activity would look in SIEM. We will showcase Splunk.

The query that would filter for kerberoasting activity will be something like this :

- ```
Event.EventData.TicketEncryptionType="0x17" Event.System.EventID="4769"
Event.EventData.ServiceName!="*$"
```

To display the fields from the filtered event, we can use the "table" command in SPL. For example, if we want to display the username conducting the attack, the service which is being attacked, and the source IP address from where it is originating, our SPL query becomes:

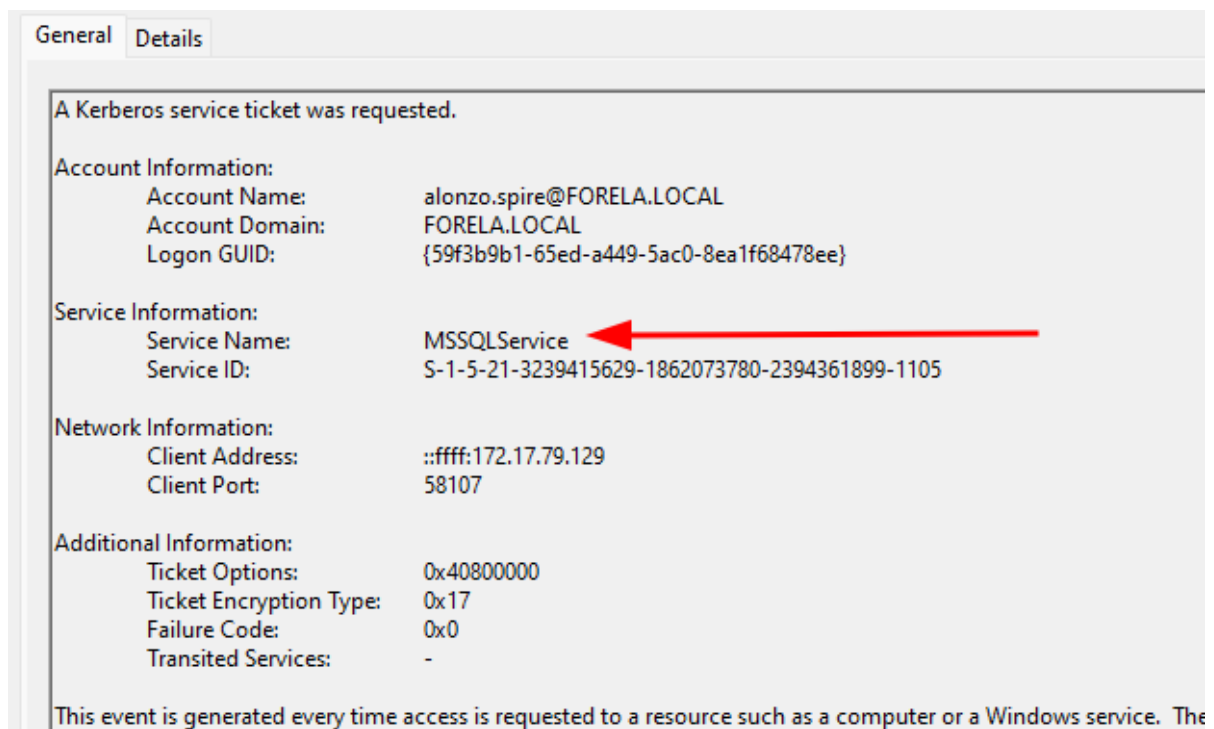
- ```
Event.EventData.TicketEncryptionType="0x17" Event.System.EventID="4769"
Event.EventData.ServiceName!="*$" | table Event.EventData.ServiceName,
Event.EventData.TargetUserName, Event.EventData.IpAddress
```



Answer: 2024-05-21 03:18:09

Q2 What is the Service Name that was targeted?

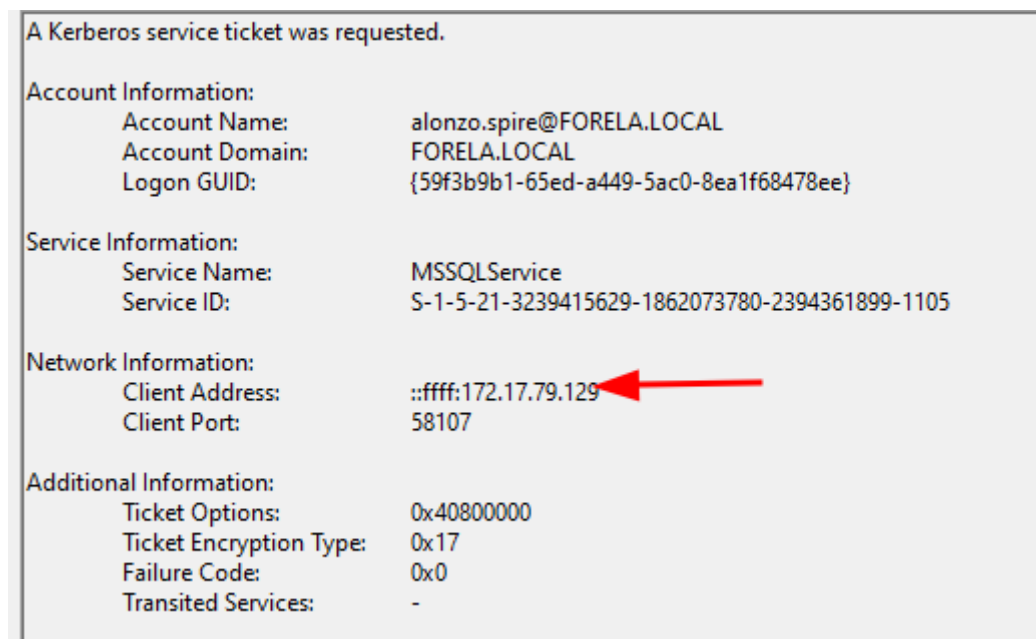
Hint: Look for service name in the Relevant Event identified



Answer: MSSQLService

Q3 It is really important to identify the Workstation from which this activity occurred. What is the IP Address of the workstation?

Hint: Look for the Client IP Address in the Relevant Event identified



Answer: 172.17.79.129

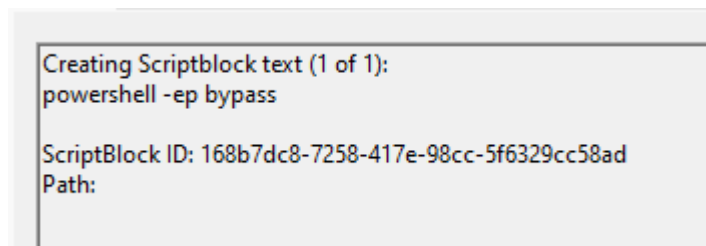
Q4 Now that we have identified the workstation, a triage including PowerShell logs and Prefetch files are provided to you for some deeper insights so we can understand how this activity occurred on the endpoint. What is name of the file used to Enumerate Active directory objects and possibly find Kerberoastable accounts in the network?

Hint: Use PowerShell logs and filter for event ID 4104. We can see all the contents of the script executed and its name as well.

Now let's pivot down to our PowerShell logs. We do have a timeline (the identified event from DC Security logs) which we will use here to track back the activities. Let's filter for Event ID 4104 here.

Powershell-Operational_2 Number of events: 42				
Filtered: Log: file://H:\Project-Sherlock\Camp Fire 1\Triage\Workstation\Powershell-Operational.evtx; Source: ; E				
Level	Date and Time	Source	Event ID	Task Category
Warning	5/21/2024 8:16:32 AM	PowerShell (...)	4104	Execute a Remote Command
Warning	5/21/2024 8:16:32 AM	PowerShell (...)	4104	Execute a Remote Command
Warning	5/21/2024 8:16:32 AM	PowerShell (...)	4104	Execute a Remote Command
Warning	5/21/2024 8:16:32 AM	PowerShell (...)	4104	Execute a Remote Command
Warning	5/21/2024 8:16:32 AM	PowerShell (...)	4104	Execute a Remote Command
Warning	5/21/2024 8:16:32 AM	PowerShell (...)	4104	Execute a Remote Command
Warning	5/21/2024 8:16:32 AM	PowerShell (...)	4104	Execute a Remote Command
Warning	5/21/2024 8:16:32 AM	PowerShell (...)	4104	Execute a Remote Command
Warning	5/21/2024 8:16:32 AM	PowerShell (...)	4104	Execute a Remote Command
Warning	5/21/2024 8:16:32 AM	PowerShell (...)	4104	Execute a Remote Command
Warning	5/21/2024 8:16:32 AM	PowerShell (...)	4104	Execute a Remote Command
Warning	5/21/2024 8:16:32 AM	PowerShell (...)	4104	Execute a Remote Command
Warning	5/21/2024 8:16:29 AM	PowerShell (...)	4104	Execute a Remote Command

If we see the timestamps, it is 8:16 which is in my local time. Converting this to UTC, the time is 03:16 which is just 2 minutes before our Kerberoasting activity. Looking at the first event, we see a PowerShell script execution bypass being performed.



Attackers bypass PowerShell script execution policies to run malicious scripts without restrictions. Reasons include:

- **Execution Policy Restrictions:** PowerShell has several execution policies (e.g., Restricted, AllSigned, RemoteSigned) that can prevent unauthorized scripts from running.
- **Evasion:** By bypassing execution policies, attackers can evade detection mechanisms that rely on these policies to block malicious activity.
- **Flexibility:** Bypassing execution policies allows attackers to use powerful offensive PowerShell scripts and tools, such as PowerView and Invoke-Mimikatz, which are crucial for enumeration and credential dumping.

The follow-up events occurred all at the same time, which could be part of a single script as PowerShell ScriptBlock records the full script being executed.

```
Creating Scriptblock text (1 of 20):
#requires -version 2

<#
    PowerShell File: PowerView.ps1
    Author: Will Schroeder (@harmj0y)
    License: BSD 3-Clause
    Required Dependencies: None
    Optional Dependencies: None
#>

#####
#
# PSReflect code for Windows API access
# Author: @mattifestation
# https://raw.githubusercontent.com/mattifestation/PSReflect/master/PSReflect.psm1
#
#####

function New-InMemoryModule
{
    <#
        .SYNOPSIS
```

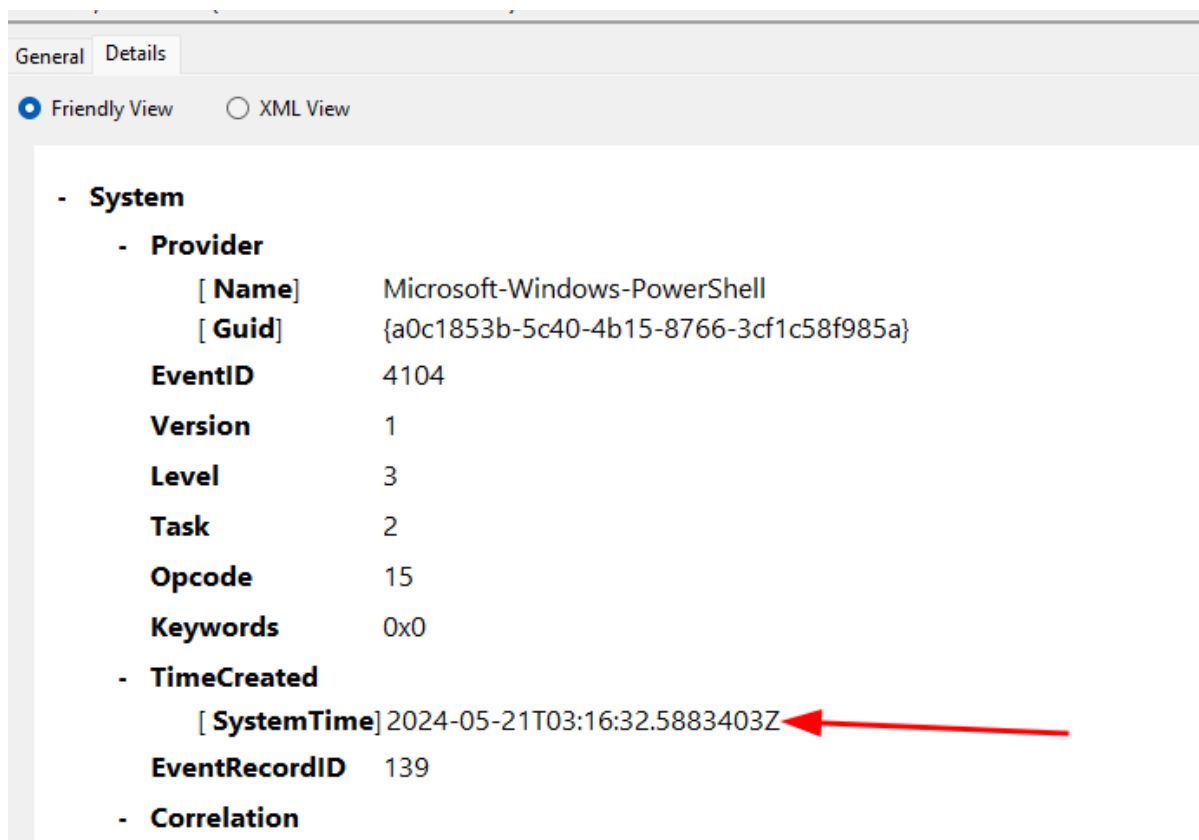
We find evidence that this is the powerview.ps1 script. **PowerView** is a PowerShell tool designed for network and Active Directory enumeration. Part of the PowerSploit framework, it is often used by penetration testers and attackers for:

- **AD Enumeration:** Discovering information about the domain, users, groups, computers, and more.
- **Finding Privileged Accounts:** Identifying accounts with high privileges that might be targeted for attacks.
- **Mapping AD Relationships:** Understanding trust relationships, group memberships, and user-to-computer associations.

Answer: powerview.ps1

Q5 When was this script executed?

Hint: Look at the first event in event id 4104 list where script blocks were recorded.



Answer: 2024-05-21 03:16:32

Q6 What is the full path of the tool used to perform the actual kerberoasting attack?

Hint: Parse the prefetch files using the PEcmd Tool by Eric Zimmerman. The syntax is Pecmd.exe -d "Path of prefetchArtifacts" --csv . --csvf result.csv. This command will create a CSV called result.csv in your current directory from where you are executing the CLI Tool. Open the CSV file in the Timeline Explorer tool (Another Eric Zimmerman tool), then look for any executables executed around the timeline we have established so far. A certain tool name will catch your eye. Then to get the path, go to the Files Loaded column and double-click the value to get a list of files interacted by the executable. It will also include its path.

We already followed the hint and created the CSV file. Let's open it in Timeline Explorer and follow the breadcrumbs in the hint.

Timeline Explorer is a forensic tool developed by Eric Zimmerman. It is used for visualizing and analyzing timelines from various data sources, particularly useful for incident response and digital forensics investigations. Timeline Explorer can ingest CSV files and allows investigators to filter, sort, and search through large datasets efficiently. Key features include:

- **Customizable Views:** Users can configure columns, sorting, and filtering to tailor the data presentation to their needs.
- **Data Integration:** It supports integrating data from multiple sources, helping to create a comprehensive timeline of events.
- **Efficiency:** It handles large datasets effectively, making it a powerful tool for forensic analysis.

We should look for any execution around the timeline we established so far. Let's filter for the date of the incident to reduce the noise.

We add the filter for the "Last Run" field.

Last Run	Previous Run
=	
2024-05-21 03:12:05	
2024-05-21 03:15:43	
2024-05-21 03:12:56	
2024-05-21 03:15:43	
2024-05-21 03:12:58	
2024-05-21 03:10:44	
2024-05-21 03:10:41	
2024-05-21 03:18:08	
2024-05-21 03:12:05	
2024-05-21 03:28:21	
2024-05-21 03:12:44	

Values Date Filters

Is Same Day
5/21/2024

Clear Filter
Close

We still have lots of results. A trick we can use is to only look for entries which have the last run entry filled and previous run entries empty. These are the executables that were run on the system for the first time.

We spot an executable named after a well-known active directory offensive tool.

Executable Name	Run C...	Si
...	=	=
CTFMON.EXE	1	
ELEVATION_SERVICE.EXE	1	
FILESYNCCONFIG.EXE	1	
IDENTITY_HELPER.EXE	1	
MICROSOFT.SHAREPOINT.EXE	1	
MICROSOFTEDGE_X64_125.0.2535.	1	
MSCORSVW.EXE	1	
RUBEUS.EXE	1	
RUNDLL32.EXE	1	
RUNDLL32.EXE	1	
RUNTIMEBROKER.EXE	1	
RUNTIMEBROKER.EXE	1	
SECURITYHEALTHSERVICE.EXE	1	
SETUP.EXE	1	

We can also look for files on the basis of their last run time.


Executable Name	Run C...	Size	Last Run
...	=	=	=
RUBEUS.EXE	1	86612	2024-05-21 03:18:08


The Rubeus tool was executed just 1 second before our DC logged the malicious event.

Rubeus is a post-exploitation tool used for Kerberos-related attacks in Active Directory environments. Developed as a part of the offensive toolkit, Rubeus provides capabilities to:

- **Kerberoasting:** Extract service tickets (TGS tickets) that can be cracked offline to retrieve service account passwords.
- **Pass-the-Ticket (PTT):** Inject and use Kerberos tickets directly into a user's session.
- **Ticket Renewal and Overpass-the-Hash:** Request new tickets using valid Kerberos ticket-granting ticket (TGT) or hash values.
- **S4U:** Perform Service-for-User attacks, requesting tickets on behalf of other users.

To get the full path of the file, go to the files loaded and double-click to see all files loaded by this tool at execution.

...	Files Loaded
	Rubeus
<input type="checkbox"/>	\VOLUME{01d951602330db46-52233816}\WINDOWS\SY


```
VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\NTDLL.DLL,  
VOLUME{01d951602330db46-52233816}\USERS\ALONZO.SPIRE\DOWNLOADS\RUBEUS.EXE,   
VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\MSCOREE.DLL,  
VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\KERNEL32.DLL,  
VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\KERNELBASE.DLL,  
VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\LOCALE.NLS,  
VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\APPHELP.DLL,  
VOLUME{01d951602330db46-52233816}\WINDOWS\APPPATCH\SYSMAIN.SDB,  
VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\ADVAPI32.DLL,  
VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\MSVCRT.DLL,
```

Answer: c:\Users\Alonzo.spire\Downloads\Rubeus.exe

Q7 When was The tool executed to dump credentials?

Hint: Look for the "Last Run" Value in the PEcmd output.

We already discussed this.

...	Executable Name	Run C...	Size	Last Run
	Rube	=	=			=
...	RUBEUS.EXE	1	86612	2024-05-21 03:18:08

Answer: 2024-05-21 03:18:08