Challenge: CrownJewel-1

Platform: HackTheBox

Category: Sherlock

Difficulty: Very Easy

Tools Used: Hayabusa, Timeline Explorer, EvtxECmd, MFTECmd, Event Viewer

Summary: This sherlock involved analysing Windows event logs and the MFT file to determine if a threat actor dumped the NTDS.dit database on the domain controller (DC). Once again, I found that this sherlock being labelled as very easy is not accurate whatsoever. I am in no way an expert, however, I found this relatively difficult, even though I have a lot of experience analysing event logs and the MFT. Regardless, this was an enjoyable challenge, it was interesting to learn about how threat actors abuse Volume Shadow Copies to dump the NTDS.dit database.

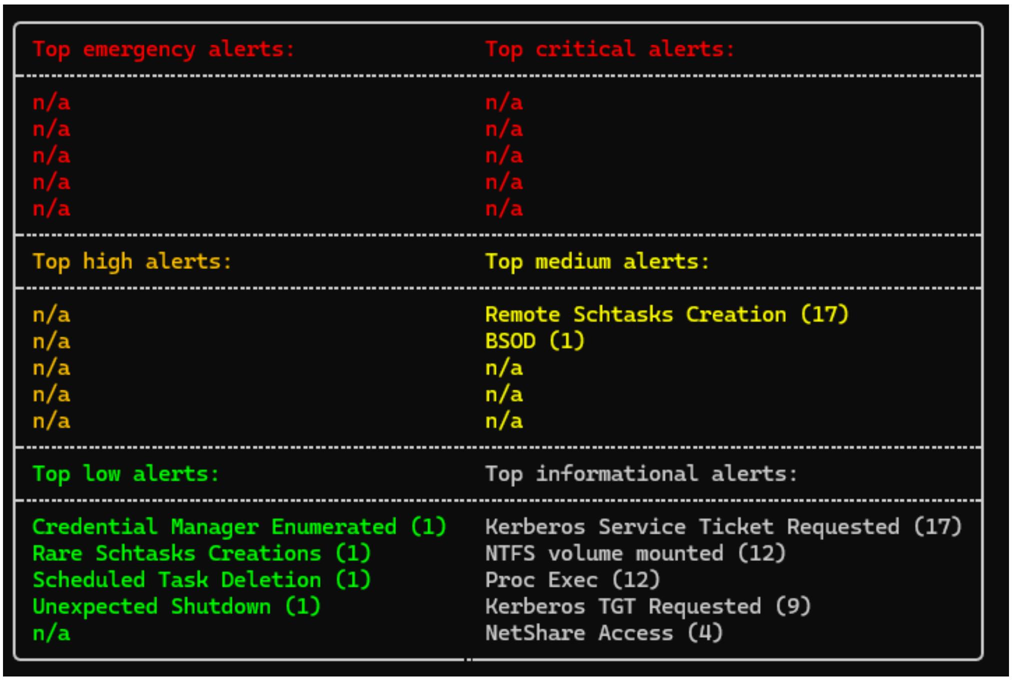
Scenario: Forela's domain controller is under attack. The Domain Administrator account is believed to be compromised, and it is suspected that the threat actor dumped the NTDS.dit database on the DC. We just received an alert of vssadmin being used on the DC, since this is not part of the routine schedule we have good reason to believe that the attacker abused this LOLBIN utility to get the Domain environment's crown jewel. Perform some analysis on provided artifacts for a quick triage and if possible kick the attacker as early as possible.

Attackers can abuse the vssadmin utility to create volume shadow snapshots and then extract sensitive files like NTDS.dit to bypass security mechanisms. Identify the time when the Volume Shadow Copy service entered a running state.

Vssadmin is a legitimate Windows utility that controls volume shadow copies. It is often abused by threat actors to create volume shadow copies for which they use to extract key files like the NTDS.dit database. The NTDS.dit file is a crucial database for Active Directory Domain Services (AD DS) on domain controllers that stores all directory data, including user accounts, group memberships, and more. Crucially, this database also stores hashed passwords for all users in a domain.

Let's start by analysing the given event logs (SYSTEM.evtx, SECURITY.evtx, and Microsoft-Windows-NTFS.evtx). I am going to use a tool called Hayabusa to get a high-level overview of the logs. For context, Hayabusa is a Windows event log threat hunting tool created by the Yamato Security group. You can download it here, and execute the following command:

- `./hayabusa-3.3.0-win-x64.exe csv-timeline -d . -o crownjewel_out.csv`
  - -d points to a directory containing multiple .evtx files, in this case it's the current directory.
  - -o specifies the output filename.



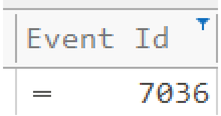
```
Top emergency alerts:                Top critical alerts:
----------------------------------------------------------------------------
n/a                                  n/a
n/a                                  n/a
n/a                                  n/a
n/a                                  n/a
n/a                                  n/a
----------------------------------------------------------------------------
Top high alerts:                     Top medium alerts:
----------------------------------------------------------------------------
n/a                                  Remote Schtasks Creation (17)
n/a                                  BSOD (1)
n/a                                  n/a
n/a                                  n/a
n/a                                  n/a
----------------------------------------------------------------------------
Top low alerts:                      Top informational alerts:
----------------------------------------------------------------------------
Credential Manager Enumerated (1)    Kerberos Service Ticket Requested (17)
Rare Schtasks Creations (1)          NTFS volume mounted (12)
Scheduled Task Deletion (1)          Proc Exec (12)
Unexpected Shutdown (1)              Kerberos TGT Requested (9)
n/a                                  NetShare Access (4)
```

You can open the output using a tool like Timeline Explorer. Unfortunately, after some time investigating the output, I found no results of interest.

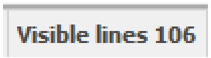
Alternatively, let's parse all three event logs using EvtxECmd, and see if we can find anything related to vssadmin:

- `.\EvtxECmd.exe -d . --csv . --csvf event_logs_out.csv`
    - -d points to a directory containing multiple .evtx files, in this case it's the current directory.
    - --csv tells EvtxECmd to output it in csv format.
    - --csvf specifies the output filename.

After extracting all the event IDs, I was able to identify an interesting log with event ID 7036. Event ID 7036 is logged whenever a Windows service changes state, such as starting, stopping, or restarting. If you filter for this event ID, you will see 106 results:

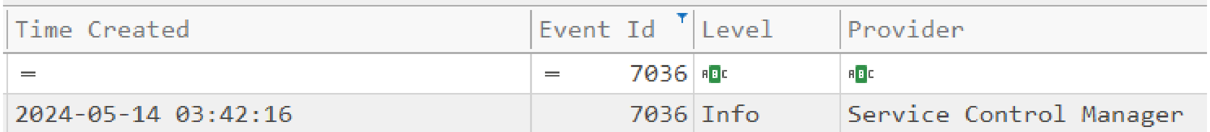


Event Id
= 7036

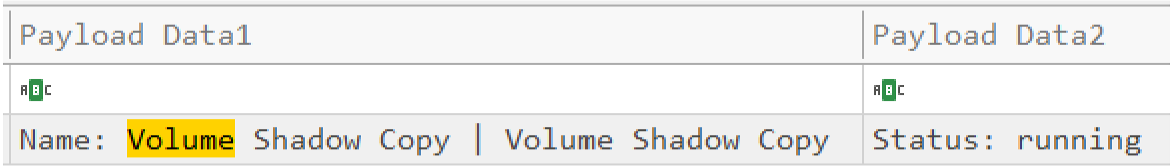


Visible lines 106

To cut this down further, I filtered for the keyword "volume" and found this:



Time Created	Event Id	Level	Provider
=	= 7036	ABC	ABC
2024-05-14 03:42:16	7036	Info	Service Control Manager



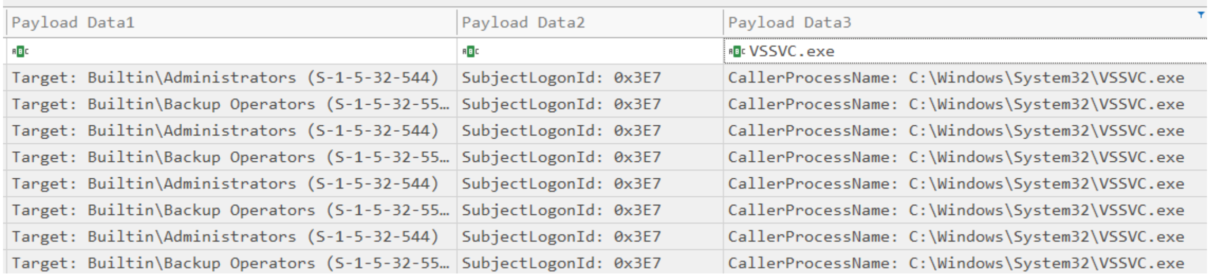
Payload Data1	Payload Data2
Name: Volume Shadow Copy | Volume Shadow Copy	Status: running

On the 14th of May 2024, at 03:42:16 the Volume Shadow Copy service started.

Answer: 2024-05-14 03:42:16

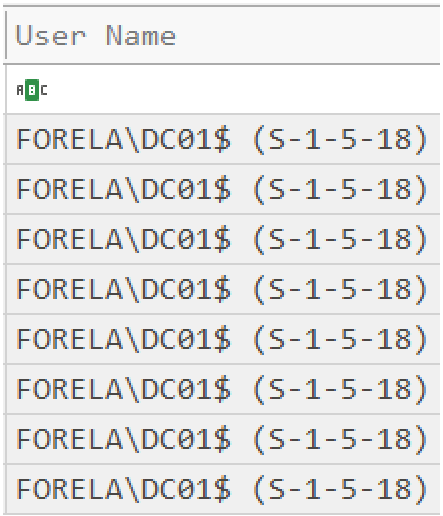
When a volume shadow snapshot is created, the Volume shadow copy service validates the privileges using the Machine account and enumerates User groups. Find the two user groups the volume shadow copy process queries and the machine account that did it.

When the Volume Shadow Copy Service (VSS) creates a snapshot, it performs security validation using the machine account and queries group memberships. Event ID 4799 is logged every time a security-enabled local group membership is enumerated. If you filter for this event ID, we can see that the Volume Shadow Copy Service (VSSVC.EXE) queried two user groups:



Payload Data1	Payload Data2	Payload Data3
		VSSVC.exe
Target: Builtin\Administrators (S-1-5-32-544)	SubjectLogonId: 0x3E7	CallerProcessName: C:\Windows\System32\VSSVC.exe
Target: Builtin\Backup Operators (S-1-5-32-55…	SubjectLogonId: 0x3E7	CallerProcessName: C:\Windows\System32\VSSVC.exe
Target: Builtin\Administrators (S-1-5-32-544)	SubjectLogonId: 0x3E7	CallerProcessName: C:\Windows\System32\VSSVC.exe
Target: Builtin\Backup Operators (S-1-5-32-55…	SubjectLogonId: 0x3E7	CallerProcessName: C:\Windows\System32\VSSVC.exe
Target: Builtin\Administrators (S-1-5-32-544)	SubjectLogonId: 0x3E7	CallerProcessName: C:\Windows\System32\VSSVC.exe
Target: Builtin\Backup Operators (S-1-5-32-55…	SubjectLogonId: 0x3E7	CallerProcessName: C:\Windows\System32\VSSVC.exe
Target: Builtin\Administrators (S-1-5-32-544)	SubjectLogonId: 0x3E7	CallerProcessName: C:\Windows\System32\VSSVC.exe
Target: Builtin\Backup Operators (S-1-5-32-55…	SubjectLogonId: 0x3E7	CallerProcessName: C:\Windows\System32\VSSVC.exe

The queried user groups were “Administrators” and “Backup Operators”. The machine account that did this was “DC01$”:

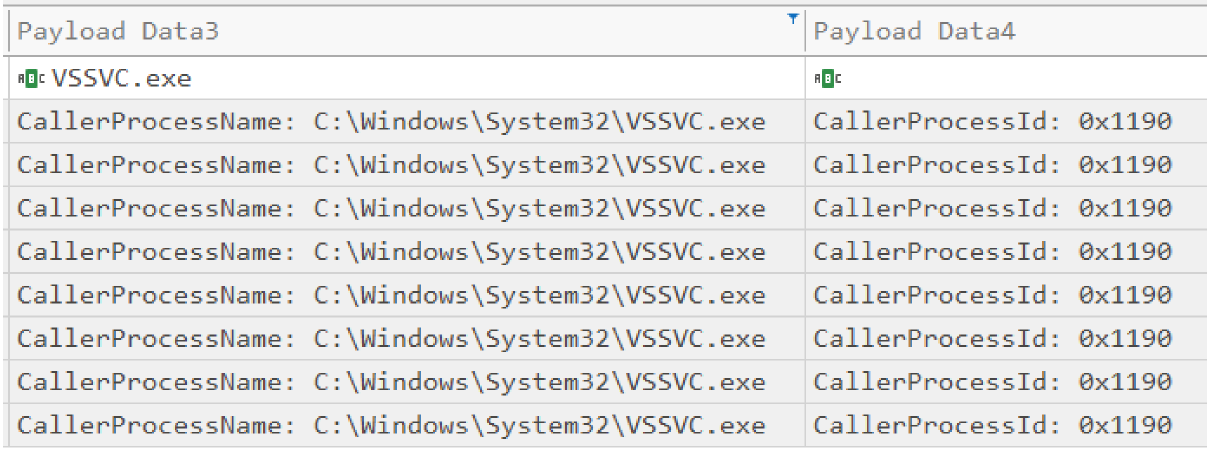


User Name
FORELA\DC01$ (S-1-5-18)
FORELA\DC01$ (S-1-5-18)
FORELA\DC01$ (S-1-5-18)
FORELA\DC01$ (S-1-5-18)
FORELA\DC01$ (S-1-5-18)
FORELA\DC01$ (S-1-5-18)
FORELA\DC01$ (S-1-5-18)
FORELA\DC01$ (S-1-5-18)

Answer: Administrators, Backup Operators, DC01$

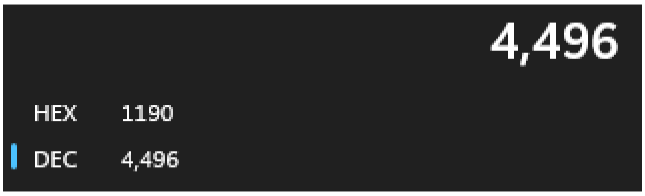
Identify the Process ID (in Decimal) of the volume shadow copy service process.

Within the Payload Data4 column, you can find the PID of the VSSVC.exe process:



Payload Data3	Payload Data4
VSSVC.exe	
CallerProcessName: C:\Windows\System32\VSSVC.exe	CallerProcessId: 0x1190
CallerProcessName: C:\Windows\System32\VSSVC.exe	CallerProcessId: 0x1190
CallerProcessName: C:\Windows\System32\VSSVC.exe	CallerProcessId: 0x1190
CallerProcessName: C:\Windows\System32\VSSVC.exe	CallerProcessId: 0x1190
CallerProcessName: C:\Windows\System32\VSSVC.exe	CallerProcessId: 0x1190
CallerProcessName: C:\Windows\System32\VSSVC.exe	CallerProcessId: 0x1190
CallerProcessName: C:\Windows\System32\VSSVC.exe	CallerProcessId: 0x1190
CallerProcessName: C:\Windows\System32\VSSVC.exe	CallerProcessId: 0x1190

0x1190 is hex for 4496:



4,496

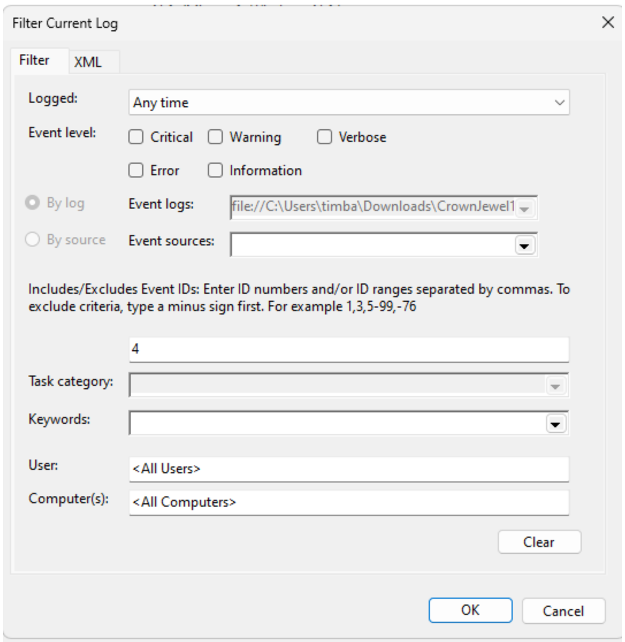
HEX 1190

DEC 4,496

Answer: 4496

Find the assigned Volume ID/GUID value to the Shadow copy snapshot when it was mounted.

To find the assigned Volume ID/GUID value to the shadow copy when it was mounted, we can filter for event ID 4, 9, 10, 300 and 303. This will show NTFS volumes mount and dismount events. In this case, I am going to use Event Viewer to filter for event ID 4 (mount events):



Logged: Any time

Event level: □ Critical □ Warning □ Verbose □ Error □ Information

● By log Event logs: file://C:\Users\timba\Downloads\CrownJewel1

○ By source Event sources:

Includes/Excludes Event IDs: Enter ID numbers and/or ID ranges separated by commas. To exclude criteria, type a minus sign first. For example 1,3,5-99,-76

4

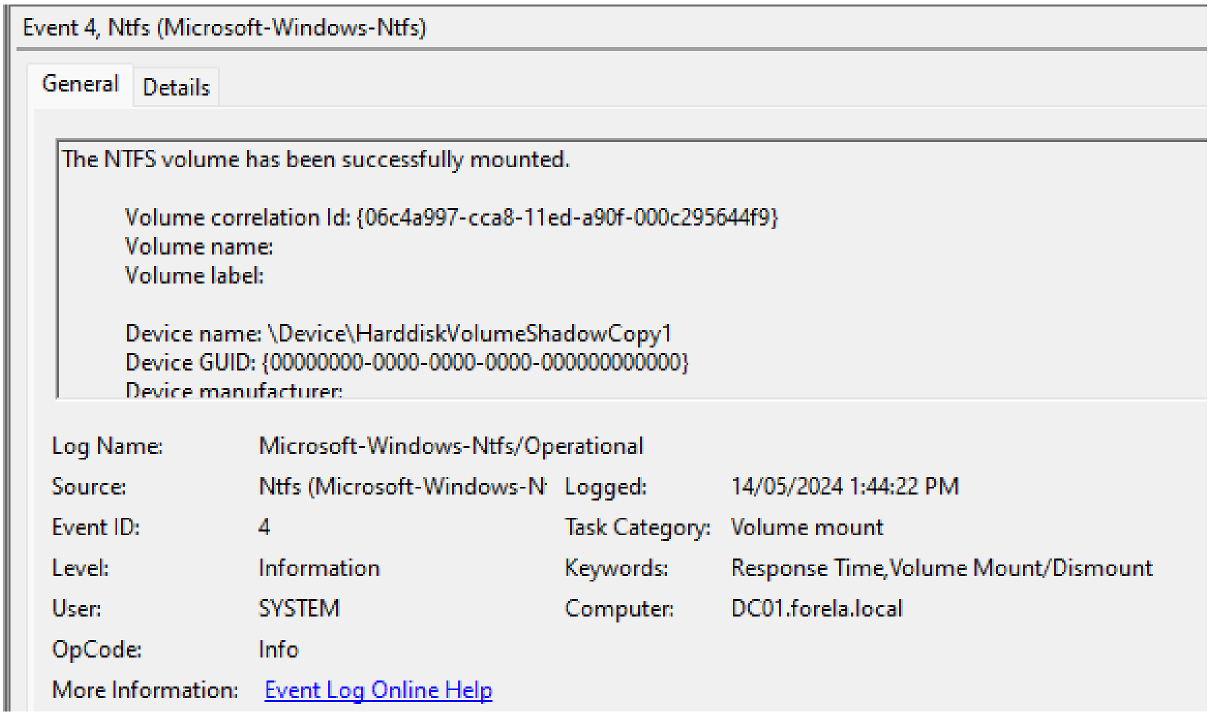
Task category:

Keywords:

User: <All Users>

Computer(s): <All Computers>

If you look through the logs, we can find something interesting:



The NTFS volume has been successfully mounted.

Volume correlation Id: {06c4a997-cca8-11ed-a90f-000c295644f9}
Volume name:
Volume label:

Device name: \Device\HarddiskVolumeShadowCopy1
Device GUID: {00000000-0000-0000-0000-000000000000}
Device manufacturer:

Log Name: Microsoft-Windows-Ntfs/Operational
Source: Ntfs (Microsoft-Windows-N
Logged: 14/05/2024 1:44:22 PM
Event ID: 4
Task Category: Volume mount
Level: Information
Keywords: Response Time,Volume Mount/Dismount
User: SYSTEM
Computer: DC01.forela.local
OpCode: Info
More Information: Event Log Online Help

The above log indicates that a volume shadow copy snapshot was mounted by the NTFS file system. What gives this off is the device name, which shows that the volume mounted is a VSS snapshot, specifically ShadowCopy1. The Volume ID/GUID assigned to the shadow copy is found in the volume correlation ID field.

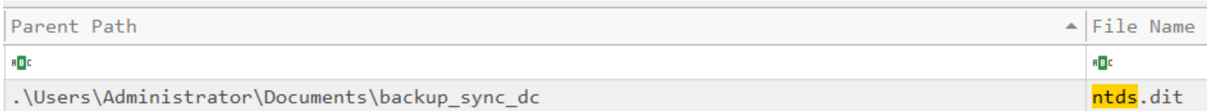
Answer: {06c4a997-cca8-11ed-a90f-000c295644f9}

Identify the full path of the dumped NTDS database on disk.

To find the full path of the dumped NTDS.dit database we need to parse the MFT file. The Master File Table (MFT) serves as a database that tracks all files and directories on the system. Each file or directory on the disk has a corresponding MFT record, which acts as a detailed metadata repository for that object. To parse the MFT, we can use a tool called MFTECmd:

- .\MFTECmd.exe -f ".\` $MFT" --csv . --csvf mft_out.csv
  - -f points to the MFT file.
  - --csv tells MFTECmd to output it in csv format.
  - --csvf specifies the output filename.

If you filter for the keyword "ntds" in the parsed MFT output, you can see that a file called ntds.dit was created on the 14th of May 2024, at 03:44:22 within the \Users\Administrator\Documents\backup_sync_dc directory:

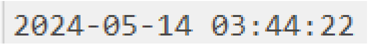


Parent Path	File Name
.\Users\Administrator\Documents\backup_sync_dc	ntds.dit

Answer: C:\Users\Administrator\Documents\backup_sync_Dc\Ntds.dit

When was newly dumped ntds.dit created on disk?

We determined when the NTDS.dit file was created on disk in the previous question:

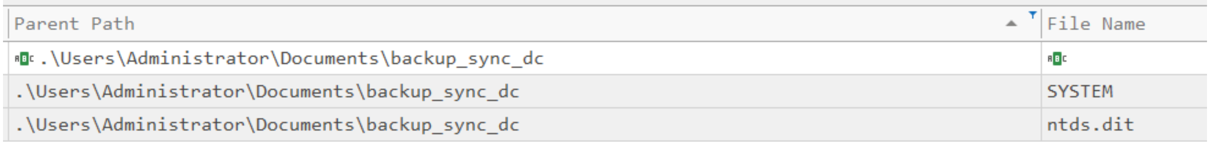


2024-05-14 03:44:22

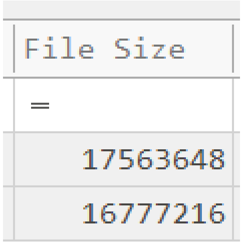
Answer: 2024-05-14 03:44:22

A registry hive was also dumped alongside the NTDS database. Which registry hive was dumped and what is its file size in bytes?

Seeing as the ntds.dit file was dumped on the 14th of May 2024, at 03:44:22 within the \Users\Administrator\Documents\backup_sync_dc directory, we can use this information to filter down the results:



Parent Path	File Name
.\Users\Administrator\Documents\backup_sync_dc	
.\Users\Administrator\Documents\backup_sync_dc	SYSTEM
.\Users\Administrator\Documents\backup_sync_dc	ntds.dit



File Size
=
17563648
16777216

We can see only two files were dumped to this directory, one being the ntds.dit database discovered earlier and the other being the SYSTEM registry hive.

Answer: SYSTEM, 17563648