Beyond the surface: Exploring the depths of SRUM for incident response

SANS DFIR Summit Europe 2023

WITH secure

Who are we?

Catarina de Faria Cristas (@c_defaria)

- **Incident Response consultant** at WithSecure in Helsinki 🇫🇮
- Former **security researcher** and **malware analyst** at F-Secure / WithSecure

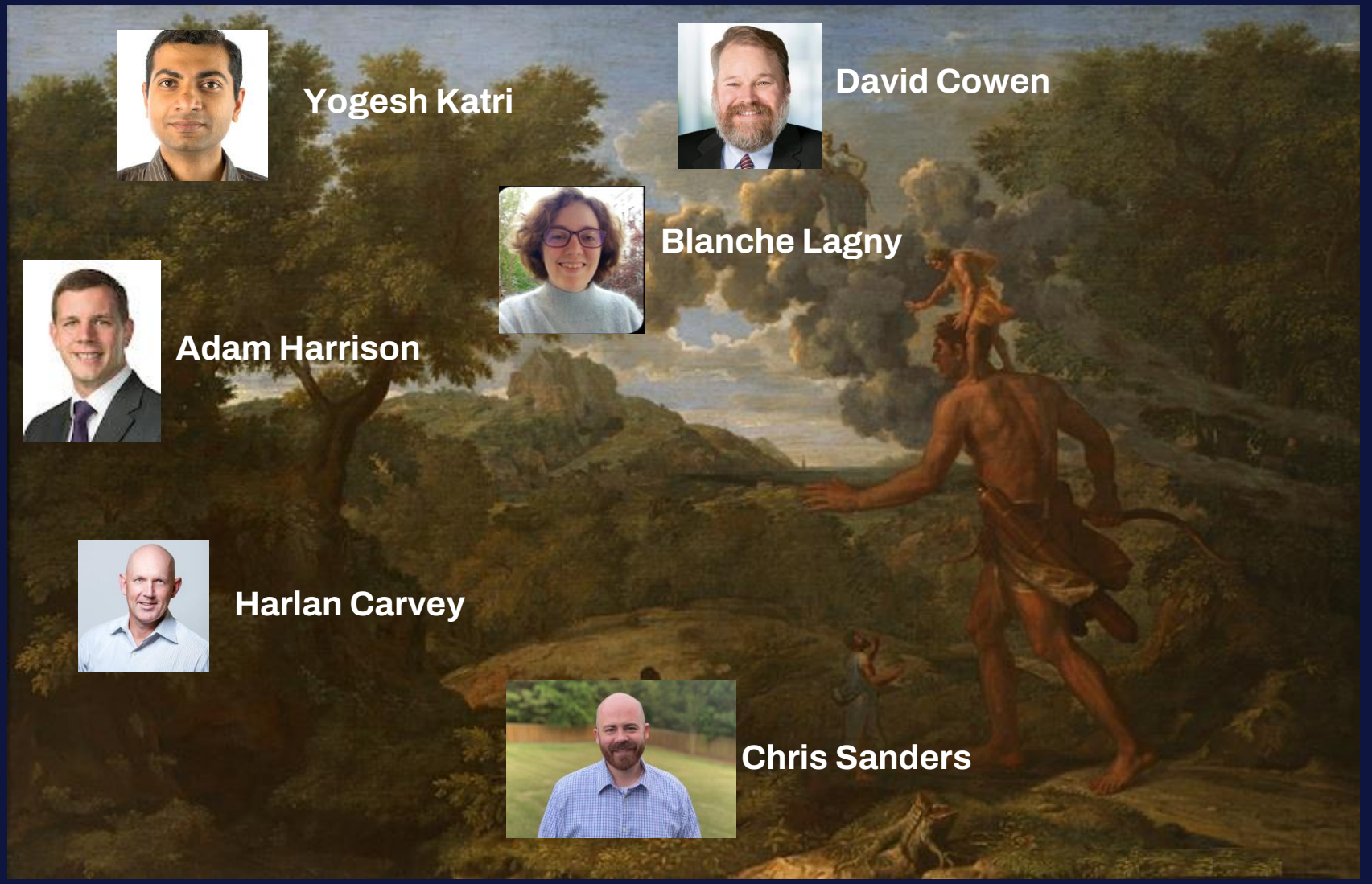
Lucas Echard

- **Incident Response consultant** at WithSecure in Helsinki 🇫🇮
- Former **EDR** and **malware analyst** at F-Secure / WithSecure

Diego Fuschini (@FuschiniDiego)

- **Senior Incident Response consultant** at WithSecure in London 🇬🇧

Standing on the Shoulders of Giants



Blind Orion Searching for the Rising Sun 1658

Premise of the talk

- **How we investigated the inner-workings of SRUM?**

Experimentation

Observation

In-depth analysis

Forensic insights

- **Experimentation** was done on the **main versions** of Windows Desktop and Server.
- **Observation** and **in-depth analysis** were performed on a **Windows 10 (2022)**, where the **DLLs** were **last modified in 2022**.

Introduction to SRUM

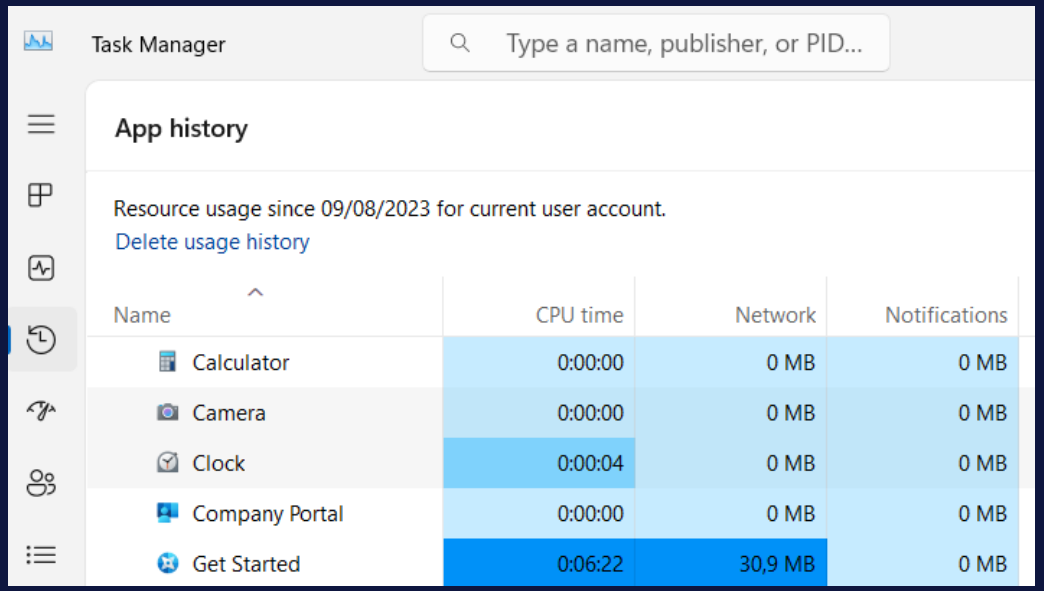
Overview

Introduction to SRUM

- First described in **Yogesh Khatri**'s paper "***Forensic implications of System Resource Usage Monitor (SRUM) data in Windows 8***" (March 2015) – Presented at **SANS DFIR Summit in July 2015**.

- Related to the **Diagnostic Policy Service (DPS)**.
- Can track **programs**, **services**, **windows apps** and **network connectivity**.
- Multiple **extensions** defined in the registry, each associated with a **DLL**, are used to **retrieve the data**.

- Can be used as a **forensic artifact to prove execution** and **data exfiltration**.



Resource usage in the Task Manager on Windows 11

SRUM on Windows

Introduction to SRUM

Experimentation

- Available starting **Windows 8** and on certain versions of **Windows Server 2019** and **onwards**
  - **Disk:**
    - SRUM database located at `C:\Windows\System32\sru\SRUDB.dat`
    - **Extensible Storage Engine (ESE)**
  - **Registry:**
    - `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SRUM\(Parameters|Extensions|Telemetry)`

	Artifact	SRUM
Desktop	Windows 11	Yes
	Windows 10	Yes
	Windows 8.1	Yes
	Windows 8	Yes

Presence of the SRUM database on Windows Desktop

	Artifact	SRUM
Server	Windows Server 2022 version 21H2 build 20348.587	Yes
	Windows Server 2019 version 1809 build 17763.4645	No
	Windows Server 2019 version 1809 build 17763.107	Yes

Presence of the SRUM database on Windows Server

SRUM on Windows

Introduction to SRUM

Experimentation

Table Name	Informal Name	Windows 10	Windows 11	Server 2019	Server 2022
MSysLocales	N/A	Yes	Yes	Yes	Yes
MSysObjects	N/A	Yes	Yes	Yes	Yes
MSysObjectsShadow	N/A	Yes	Yes	Yes	Yes
MSysObjids	N/A	Yes	Yes	Yes	Yes
SruCheckpointTable	N/A	Yes	Yes	Yes	Yes
SruDbIdMapTable	N/A	Yes	Yes	Yes	Yes
{5C8CF1C7-7257-4F13-B223-970EF5939312}	App Timeline Provider	Yes	Yes	No	No
{7ACBBAA3-D029-4BE4-9A7A-0885927F1D8F}	vfuprov	Yes	Yes	No	No
{973F5D5C-1D90-4944-BE8E-24B94231A174}	Windows Network Data Usage Monitor	Yes	Yes	No	No
{D10CA2FE-6F6C-4F6D-848E-B2E99266FA86}	WPN SRUM Provider (Push Notification Data)	Yes	Yes	No	No
{D10CA2FE-6F6C-4F6D-848E-B2E99266FA89}	Application Resource Usage Provider	Yes	Yes	Yes	Yes
{DD6636C4-8929-4683-974E-22C046A43763}	Windows Network Connectivity Usage Monitor	Yes	Yes	Yes	Yes
{FEE4E14F-02A9-4550-B5CE-5FA2DA202E37}	Energy Usage Provider	Yes	Yes	No	No
{FEE4E14F-02A9-4550-B5CE-5FA2DA202E37}LT	Energy Usage Provider Long Term	Yes	Yes	No	No
{DA73FB89-2BEA-4DDC-86B8-6E048C6DA477}	Energy Estimation Provider	No	Yes	No	No
{B6D82AF1-F780-4E17-8077-6CB9AD8A6FC4}	Tagged Energy Provider	No	Yes	No	No
{EEE2F477-0659-5C47-EF03-6D6BEFD441B3}	SDP Network Provider	No	No	Yes	Yes
{DC3D3B50-BB90-5066-FA4E-A5F90DD8B677}	SDP Cpu Provider	No	No	Yes	Yes
{841A7317-3805-518B-C2EA-AD224CB4AF84}	SDP Physical Disk Provider	No	No	Yes	Yes
{17F4D97B-F26A-5E79-3A82-90040A47D13D}	SDP Volume Provider	No	No	Yes	Yes

Presence of the SRUM database tables on Windows Desktop and Server

Database in the registry

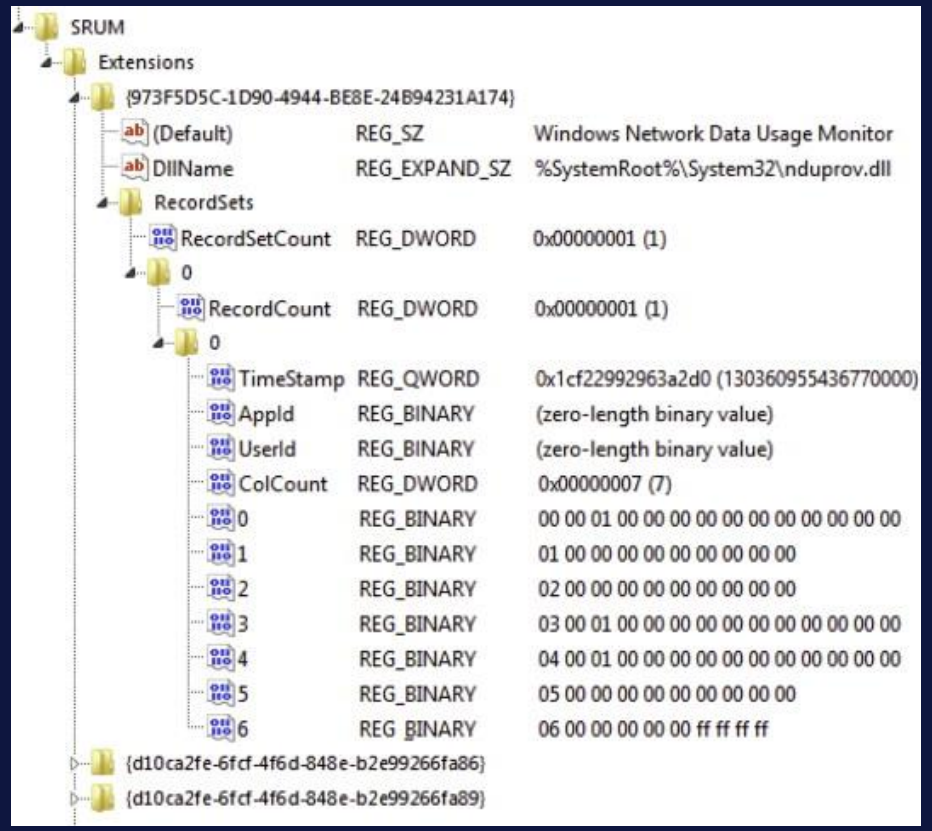
Introduction to SRUM

Observation

When the SRUM was first implemented on Windows, the **registry** would store the **data temporarily** before it was moved to **SRUDB.dat**.

This is no longer the case in recent versions of Windows!

On **Windows desktop**, **it seems** that after **Windows 10 1607 (Redstone)**, the information stays in **memory**.



The content of the SRUM database in the registry on Windows 8

Monitoring & Analysing SRUM

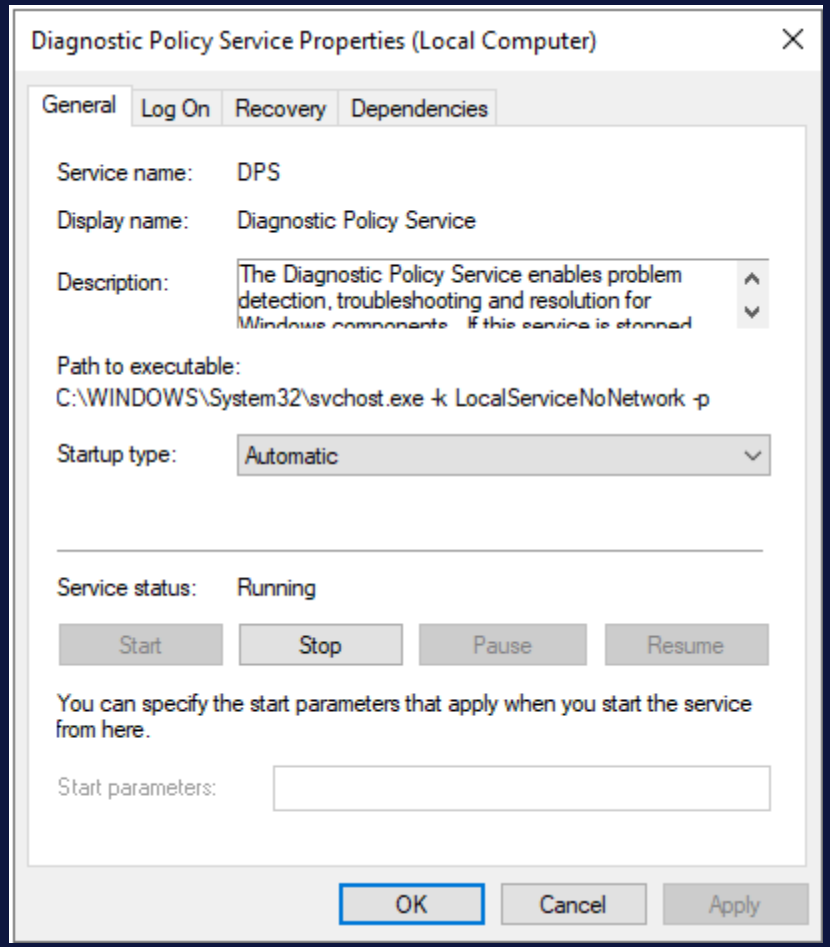
DPS and SRUM

Monitoring & Analysing SRUM

In-depth analysis

- **Diagnostic Policy Service** (`C:\Windows\System32\dps.dll`)
  - **Windows Diagnostic Infrastructure (WDI)**
  - **Diagnostic modules**
    - `HKLM\System\CurrentControlSet\Control\WDI\DiagnosticModules`

srumsvc.dll, the **System Resource Usage Monitor Service DLL**



Diagnostic Policy Service (DPS) properties

Monitoring the interactions

Monitoring & Analysing SRUM

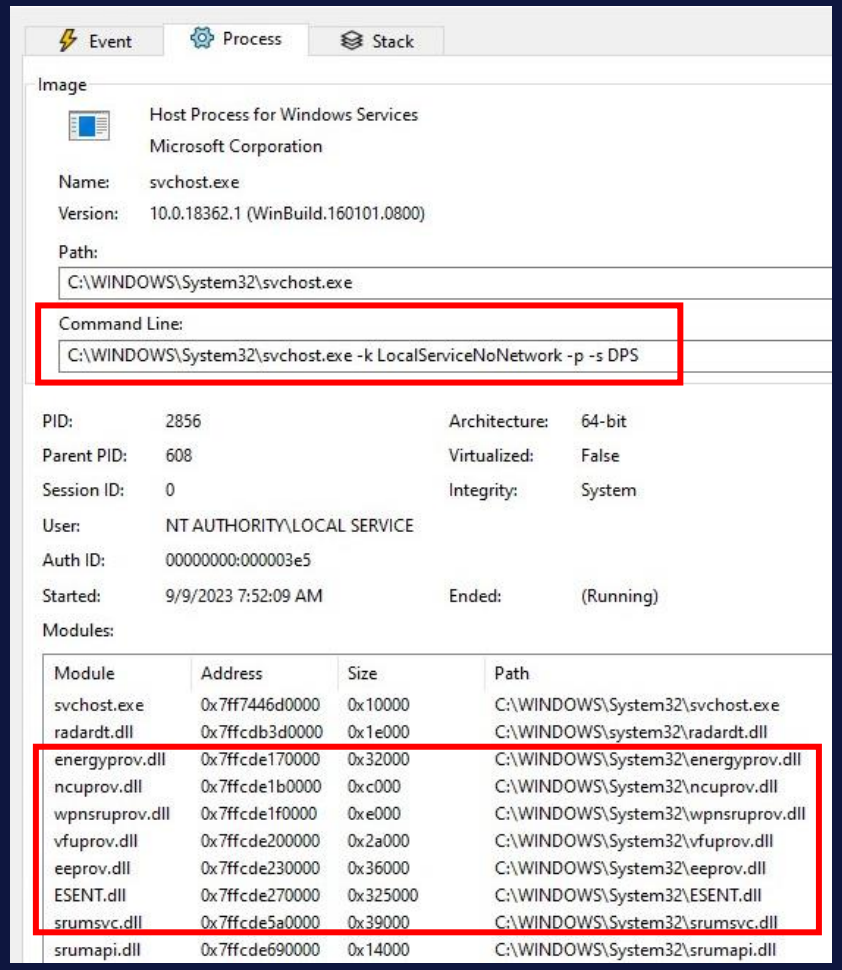
Observation

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time of Day	Process Name	PID	Operation	Path
8:19:13.3526022 AM	svchost.exe	2856	QueryInformationVolume	C:\Windows\System32\sru\SRUDB.dat
8:19:13.3526178 AM	svchost.exe	2856	QueryAllInformationFile	C:\Windows\System32\sru\SRUDB.dat
8:19:13.3527467 AM	svchost.exe	2856	WriteFile	C:\Windows\System32\sru\SRUDB.dat
8:19:13.3548447 AM	svchost.exe	2856	WriteFile	C:\Windows\System32\sru\SRUDB.dat
8:19:13.3553028 AM	svchost.exe	2856	CloseFile	C:\Windows\System32\sru\SRUDB.dat
8:19:13.3561299 AM	svchost.exe	2856	CreateFile	C:\Windows\System32\sru\SRUDB.dat
8:19:13.3562667 AM	svchost.exe	2856	QueryInformationVolume	C:\Windows\System32\sru\SRUDB.dat
8:19:13.3562933 AM	svchost.exe	2856	QueryAllInformationFile	C:\Windows\System32\sru\SRUDB.dat
8:19:13.3565886 AM	svchost.exe	2856	CreateFileMapping	C:\Windows\System32\sru\SRUDB.dat
8:19:13.3566017 AM	svchost.exe	2856	QueryStandardInformationFile	C:\Windows\System32\sru\SRUDB.dat
8:19:13.3566296 AM	svchost.exe	2856	CreateFileMapping	C:\Windows\System32\sru\SRUDB.dat
8:19:13.4443049 AM	svchost.exe	2856	ReadFile	C:\Windows\System32\sru\SRUDB.dat
8:19:13.4443570 AM	svchost.exe	2856	ReadFile	C:\Windows\System32\sru\SRUDB.dat

Procmon events where the Path contains SRUDB.dat



Properties of the svchost process that interacts with SRUDB.dat

Monitoring the interactions

Monitoring & Analysing SRUM

Observation

Event | Process | Stack

Frame	Module	Location	Address	Path
K 0	FLTMGR.SYS	FltDecodeParameters + 0x210c	0xfffff8034b9e64cc	C:\Windows\System32\drivers\FLTMGR.SYS
K 1	FLTMGR.SYS	FltDecodeParameters + 0x1bba	0xfffff8034b9e5f7a	C:\Windows\System32\drivers\FLTMGR.SYS
K 2	FLTMGR.SYS	FltDecodeParameters + 0xc61	0xfffff8034b9e5021	C:\Windows\System32\drivers\FLTMGR.SYS
K 3	FLTMGR.SYS	FltDecodeParameters + 0x66b	0xfffff8034b9e4a2b	C:\Windows\System32\drivers\FLTMGR.SYS
K 4	ntoskrnl.exe	IofCallDriver + 0x55	0xfffff8034d010665	C:\Windows\system32\ntoskrnl.exe
K 5	ntoskrnl.exe	NtDeviceIoControlFile + 0x108c	0xfffff8034d400fec	C:\Windows\system32\ntoskrnl.exe
K 6	ntoskrnl.exe	NtCopyFileChunk + 0xf79	0xfffff8034d3ce0d9	C:\Windows\system32\ntoskrnl.exe
K 7	ntoskrnl.exe	NtWriteFile + 0x996	0xfffff8034d468066	C:\Windows\system32\ntoskrnl.exe
K 8	ntoskrnl.exe	_setjmpex + 0x8305	0xfffff8034d2105f5	C:\Windows\system32\ntoskrnl.exe
U 9	ntdll.dll	NtWriteFile + 0x14	0x7ffb76cecf54	C:\Windows\SYSTEM32\ntdll.dll
U 10	KERNELBASE.dll	WriteFile + 0xfa	0x7ffb746253aa	C:\Windows\System32\KERNELBASE.dll
U 11	ESENT.dll	JetSetColumn + 0xd0c7	0x7ffb649015d7	C:\Windows\system32\ESENT.dll
U 12	ESENT.dll	JetSetColumn + 0xce8b	0x7ffb6490139b	C:\Windows\system32\ESENT.dll
U 13	ESENT.dll	JetOpenDatabaseW + 0x1222e	0x7ffb648d038e	C:\Windows\system32\ESENT.dll
U 14	ESENT.dll	JetSetColumn + 0xdf26	0x7ffb64902436	C:\Windows\system32\ESENT.dll
U 15	ESENT.dll	JetSetColumn + 0xdfd0	0x7ffb649024e0	C:\Windows\system32\ESENT.dll
U 16	ESENT.dll	JetSetColumn + 0x14a6e	0x7ffb64908f7e	C:\Windows\system32\ESENT.dll
U 17	ESENT.dll	JetSetColumn + 0x14860	0x7ffb64908d70	C:\Windows\system32\ESENT.dll
U 18	ESENT.dll	JetSetColumn + 0x146d8	0x7ffb64908be8	C:\Windows\system32\ESENT.dll
U 19	ESENT.dll	JetSetColumn + 0x16538	0x7ffb6490aa48	C:\Windows\system32\ESENT.dll
U 20	ESENT.dll	JetBeginSessionA + 0x9c15	0x7ffb648adaa5	C:\Windows\system32\ESENT.dll
U 21	ESENT.dll	JetBeginSessionA + 0xa0c2	0x7ffb648adf52	C:\Windows\system32\ESENT.dll
U 22	ESENT.dll	JetSetColumn + 0x1da34	0x7ffb64911f44	C:\Windows\system32\ESENT.dll
U 23	ESENT.dll	JetAttachDatabase2W + 0x360e	0x7ffb6489e3ee	C:\Windows\system32\ESENT.dll
U 24	ESENT.dll	JetInit4W + 0x7708	0x7ffb649263f8	C:\Windows\system32\ESENT.dll
U 25	ESENT.dll	JetCreateInstance2W + 0x235b	0x7ffb64928b1b	C:\Windows\system32\ESENT.dll
U 26	ESENT.dll	JetTerm + 0x68	0x7ffb64928d58	C:\Windows\system32\ESENT.dll
U 27	srumsvc.dll	srumsvc.dll + 0x3de0	0x7ffb64d43de0	C:\Windows\System32\srumsvc.dll
U 28	srumsvc.dll	srumsvc.dll + 0x3e77	0x7ffb64d43e77	C:\Windows\System32\srumsvc.dll
U 29	ntdll.dll	RtlSetCriticalSectionSpinCount + 0xf9	0x7ffb76cc16e9	C:\Windows\SYSTEM32\ntdll.dll
U 30	ntdll.dll	TpReleaseCleanupGroupMembers + 0xada	0x7ffb76ca31ba	C:\Windows\SYSTEM32\ntdll.dll
U 31	KERNEL32.DLL	BaseThreadInitThunk + 0x14	0x7ffb753e7344	C:\Windows\System32\KERNEL32.DLL
U 32	ntdll.dll	RtlUserThreadStart + 0x21	0x7ffb76ca26b1	C:\Windows\SYSTEM32\ntdll.dll

Call stack related to a WriteFile event on the SRUDB.dat in Procmon

WITH secure

Monitoring the interactions

Monitoring & Analysing SRUM

Observation

Summary | 46 of 10,574 calls | 99% filtered out | 4.59 MB used | svchost.exe

#	Time of Day	Thread	Module	API
29	8:10:00.002 AM	4	srumsvc.dll	memset (0x0000005b425ff400, 0, 512)
79	8:10:00.002 AM	3	srumsvc.dll	memset (0x0000005b426ff000, 0, 512)
325	8:10:00.033 AM	4	srumsvc.dll	_vsnwprintf (0x0000005b425ff400, 255, "Update of T1 store happening at time: %02d:%02d:%02dwith entry timestam...
382	8:10:00.033 AM	3	srumsvc.dll	_vsnwprintf (0x0000005b426ff000, 255, "Update of T1 store happening at time: %02d:%02d:%02dwith entry timestam...
383	8:10:00.033 AM	3	srumsvc.dll	memset (0x0000005b426ff000, 0, 512)
429	8:10:00.048 AM	3	srumsvc.dll	_vsnwprintf (0x0000005b426ff000, 255, "Update of T1 store happening at time: %02d:%02d:%02dwith entry timestam...
458	8:10:00.064 AM	3	srumsvc.dll	memset (0x0000005b426feff0, 0, 512)
530	8:10:00.064 AM	3	srumsvc.dll	_vsnwprintf (0x0000005b426feff0, 255, "Entry inserted/merged to T1 store at time: %02d:%02d:%02dwith entry timest...
531	8:10:00.064 AM	3	srumsvc.dll	memset (0x0000005b426ff000, 0, 512)
571	8:10:00.064 AM	3	srumsvc.dll	_vsnwprintf (0x0000005b426ff000, 255, "Update of T1 store happening at time: %02d:%02d:%02dwith entry timestam...
1324	8:10:00.142 AM	2	srumsvc.dll	memset (0x0000005b41eff370, 0, 512)
1378	8:10:00.158 AM	2	srumsvc.dll	_vsnwprintf (0x0000005b41eff370, 255, "Update of T1 store happening at time: %02d:%02d:%02dwith entry timestam...
1379	8:10:00.158 AM	2	srumsvc.dll	memset (0x0000005b41eff360, 0, 512)
1397	8:10:00.158 AM	2	srumsvc.dll	_vsnwprintf (0x0000005b41eff360, 255, "Entry inserted/merged to T1 store at time: %02d:%02d:%02dwith entry timest...
4860	8:10:00.408 AM	4	srumsvc.dll	memset (0x0000005b425ff400, 0, 512)
4926	8:10:00.408 AM	4	srumsvc.dll	_vsnwprintf (0x0000005b425ff400, 255, "Update of T1 store happening at time: %02d:%02d:%02dwith entry t estam...
4927	8:10:00.408 AM	4	srumsvc.dll	memset (0x0000005b425ff3f0, 0, 512)

API monitor output, where the module is srumsvc.dll, and showing message logs


```
"Entry inserted/merged to T1 store at time: 15:11:00\nwith entry timestamp: 15:11:00"
```

WITH secure

Event Tracing for Windows (ETW)

Monitoring & Analysing SRUM

Observation

- **SRUM components** and **extensions** write **ETW events** using the `EventWriteTransfer` API.

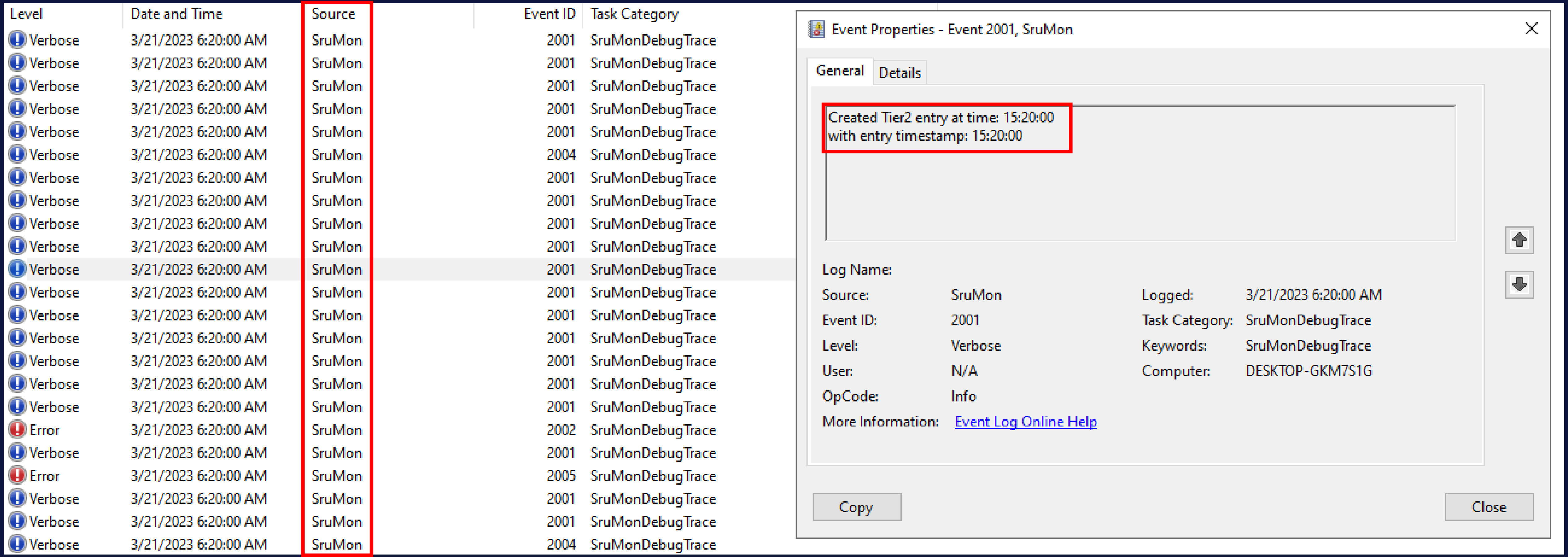
WINEVT Channel	DLL
Microsoft-Windows-SruMon	srumapi.dll/srumsvc.dll
Microsoft-Windows-SruTelemetry	energyprov.dll
Microsoft-Windows-Energy-Estimation-Engine	eeprov.dll
Microsoft-Windows-AppSruProv	appsruprov.dll
Microsoft-Windows-Ndu	nduprov.dll
…	…

- **logman** was used to create a **tracing session** to monitor all the **providers** related to **SRUM**.

Event Tracing for Windows (ETW)

Monitoring & Analysing SRUM

Observation



Level	Date and Time	Source	Event ID	Task Category
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2004	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Error	3/21/2023 6:20:00 AM	SruMon	2002	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Error	3/21/2023 6:20:00 AM	SruMon	2005	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2001	SruMonDebugTrace
Verbose	3/21/2023 6:20:00 AM	SruMon	2004	SruMonDebugTrace

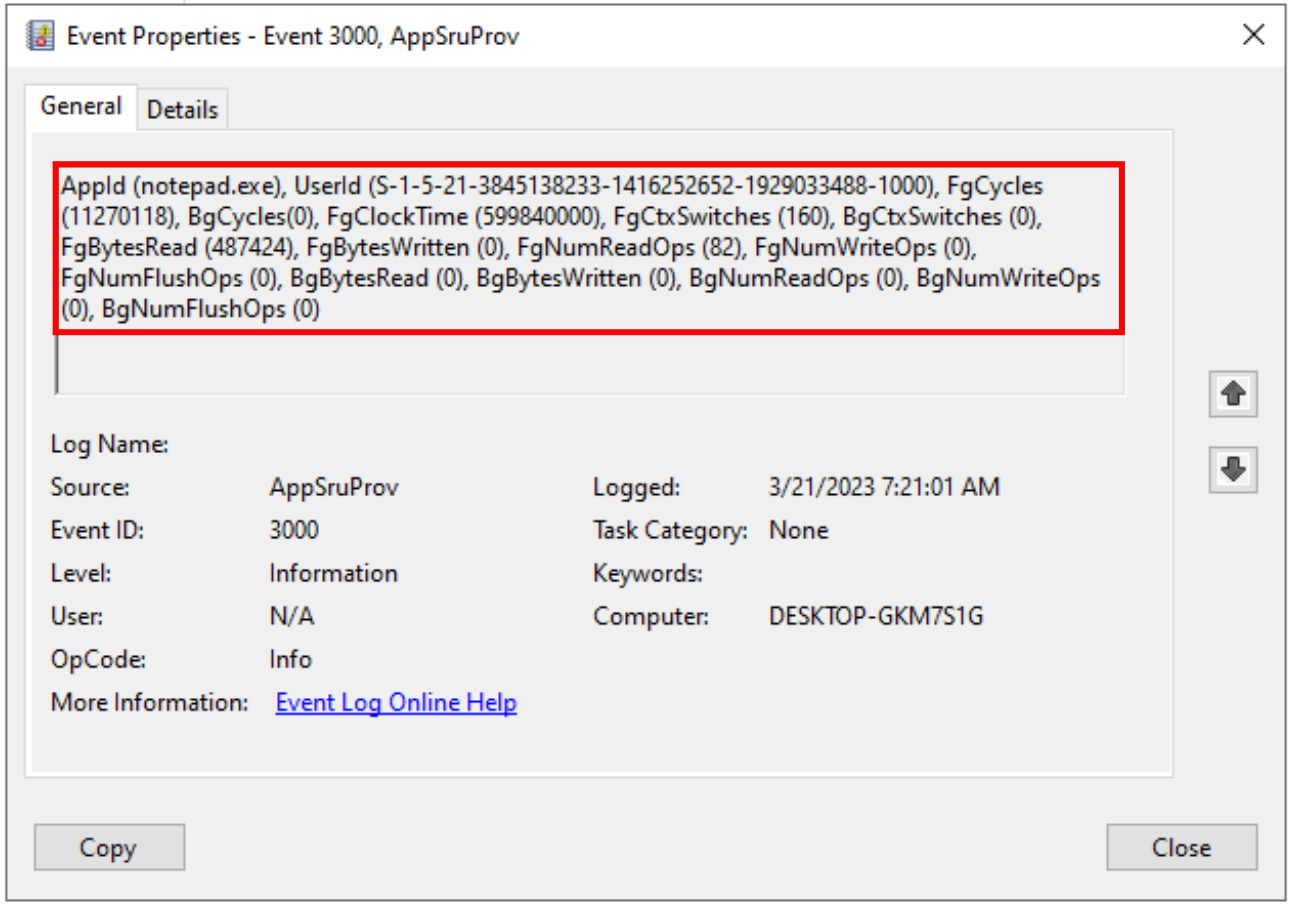
Windows Event Logs related to the Microsoft-Windows-SruMon channel

Event Tracing for Windows (ETW)

Monitoring & Analysing SRUM

Observation

Level	Date and Time	Source	Event ID	Task Category
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None
Information	3/21/2023 7:21:01 AM	AppSruProv	3000	None



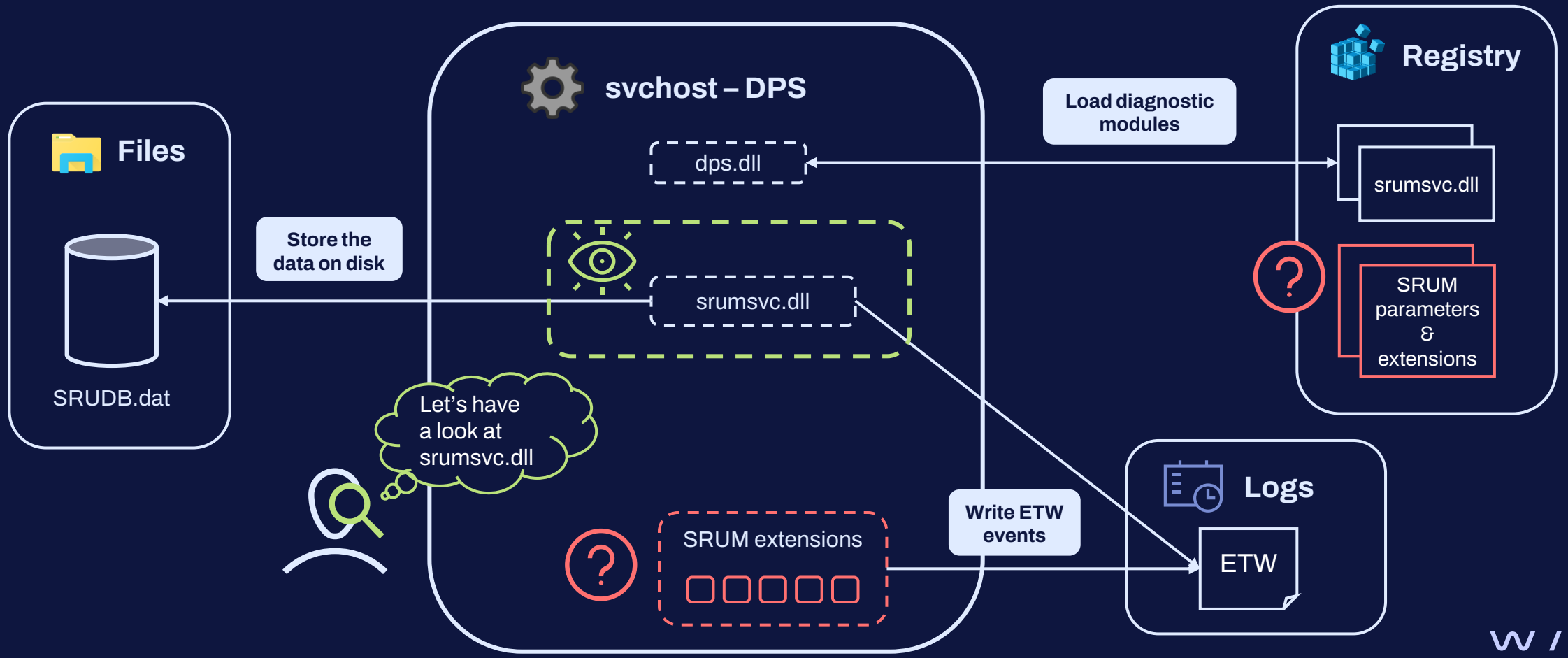
Windows Event Logs related to the Microsoft-Windows-AppSruProv channel

Main components

Monitoring & Analysing SRUM

Observation

- **Summary** of the findings up until this point:



Focus on srumsvc.dll

Monitoring & Analysing SRUM

In-depth analysis

- **srumsvc.dll handles all the components related to the SRUM** so the data from each extension is stored in the database.
- The SRUM service DLL has **multiple roles**:

WITH secure

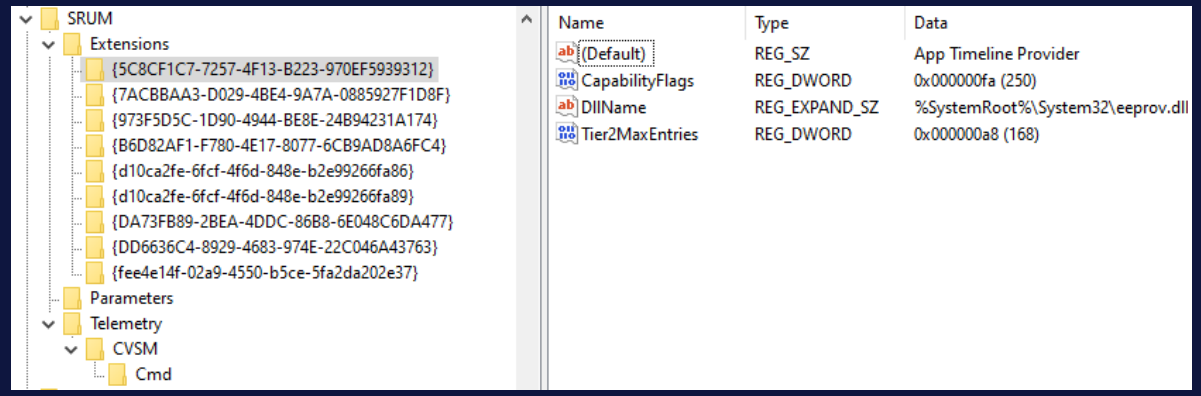
Focus on srumsvc.dll

Monitoring & Analysing SRUM

In-depth analysis

Initialization

- **Registry**
  - Retrieves **global parameters**
  - Gets the information about the **extensions, including the extension/provider IDs**



Registry Editor for `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SRUM\`

- **Callback functions**
  - Specific to each extension/provider
  - Registered into a structure so they can be directly called by srumsvc.dll
    - `SruInitializeProvider`
    - `SruUninitializeProvider`
    - `QueryColumnInfo`
    - `QueryStats`
    - …

WITH secure

Focus on srumsvc.dll

Monitoring & Analysing SRUM

In-depth analysis

Querying

- Retrieves data from each extension/provider using their **`QueryStats` callback** function
- Merges the data using the **`MergeStats` callback** function to **prevent duplicated records**.

Storing

- 2 types of storage

Tier1 store
In-memory

Tier2 store
SRUDB.dat

- Periodic updates

Tier1Period
(default: 60s) → Update **Tier1 store**

Tier2Period
(default: 1h) → Persist the **Tier1 store** to the **Tier2 store**

Focus on srumsvc.dll

Monitoring & Analysing SRUM

Experimentation

In-depth analysis

Uninitialization

- The **Tier2 store** is updated with the data from the **Tier1 store**.
- The providers are **uninitialized** by calling their `SruUninitializeProvider` **callback function**.

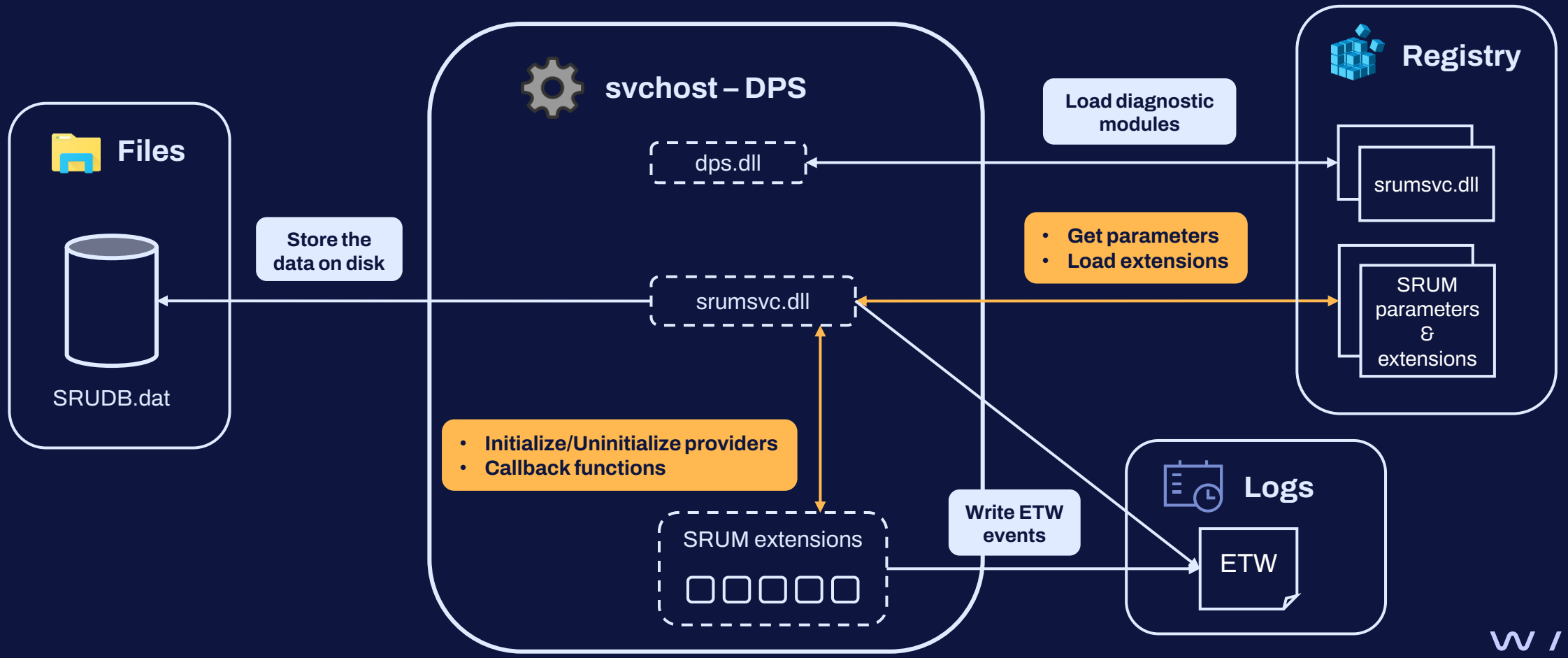
Tests	Result
DPS: **Stop DPS service**	The SRUM database is updated as part of the **uninitialization** routine
Power off: **Power off the machine**	

Main components

Monitoring & Analysing SRUM

In-depth analysis

- **Updated version** based on the findings about **srumsvc.dll**



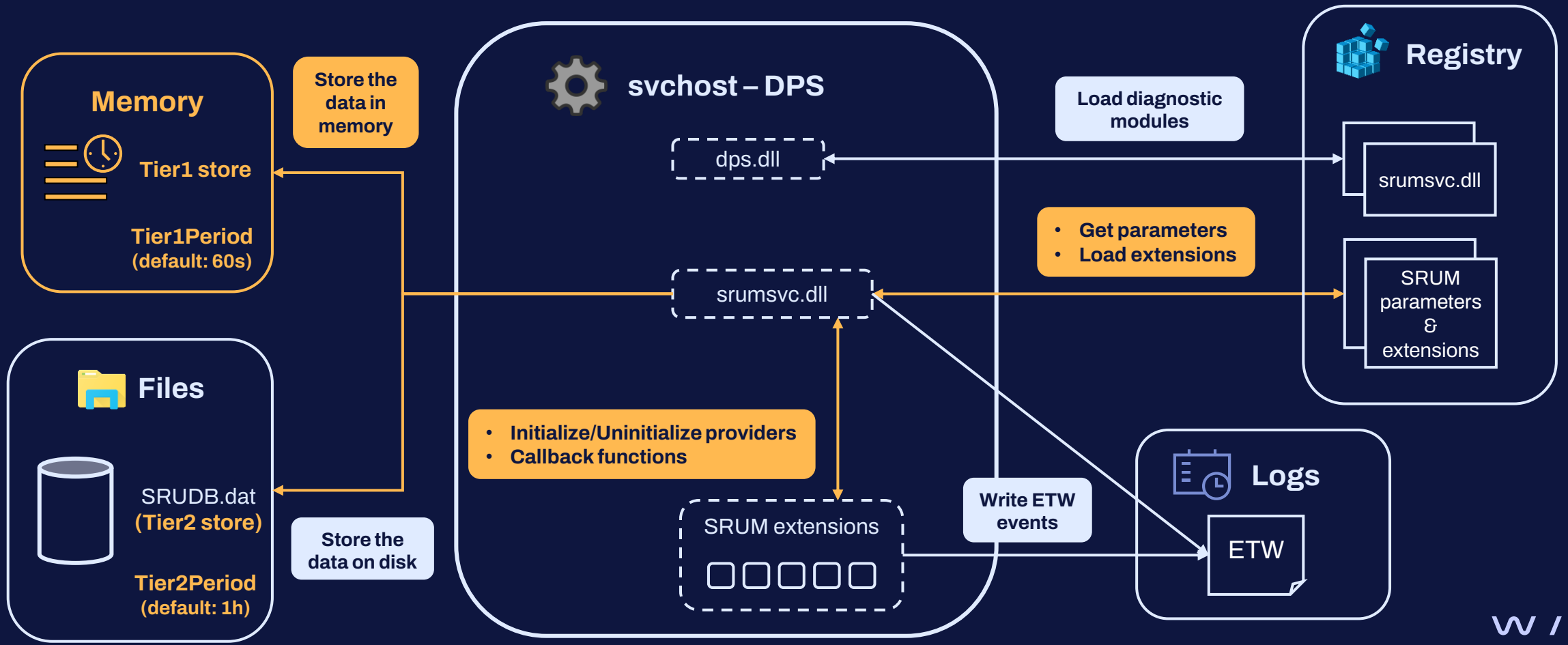
WITH secure

Main components

Monitoring & Analysing SRUM

In-depth analysis

- **Updated version** based on the findings about **srumsvc.dll**



Deep dive into SRUM extensions

Monitoring & Analysing SRUM

nduprov.dll

- **Network Data Usage Provider**
- Evidence of **network activity** and possibly **data exfiltration**

eeprov.dll

- **Energy Estimator Provider**
- Populates 3 tables **Tagged Energy Provider**, **Energy Estimation Provider**, **App Timeline Provider**
- Evidence of **execution**

Focus on nduprov.dll

Monitoring & Analysing SRUM

Observation

- Table name: {973F5D5C-1D90-4944-BE8E-24B94231A174}
- Most interesting columns:
  - **BytesSent**
  - **BytesRecvd**
  - **InterfaceLuid**

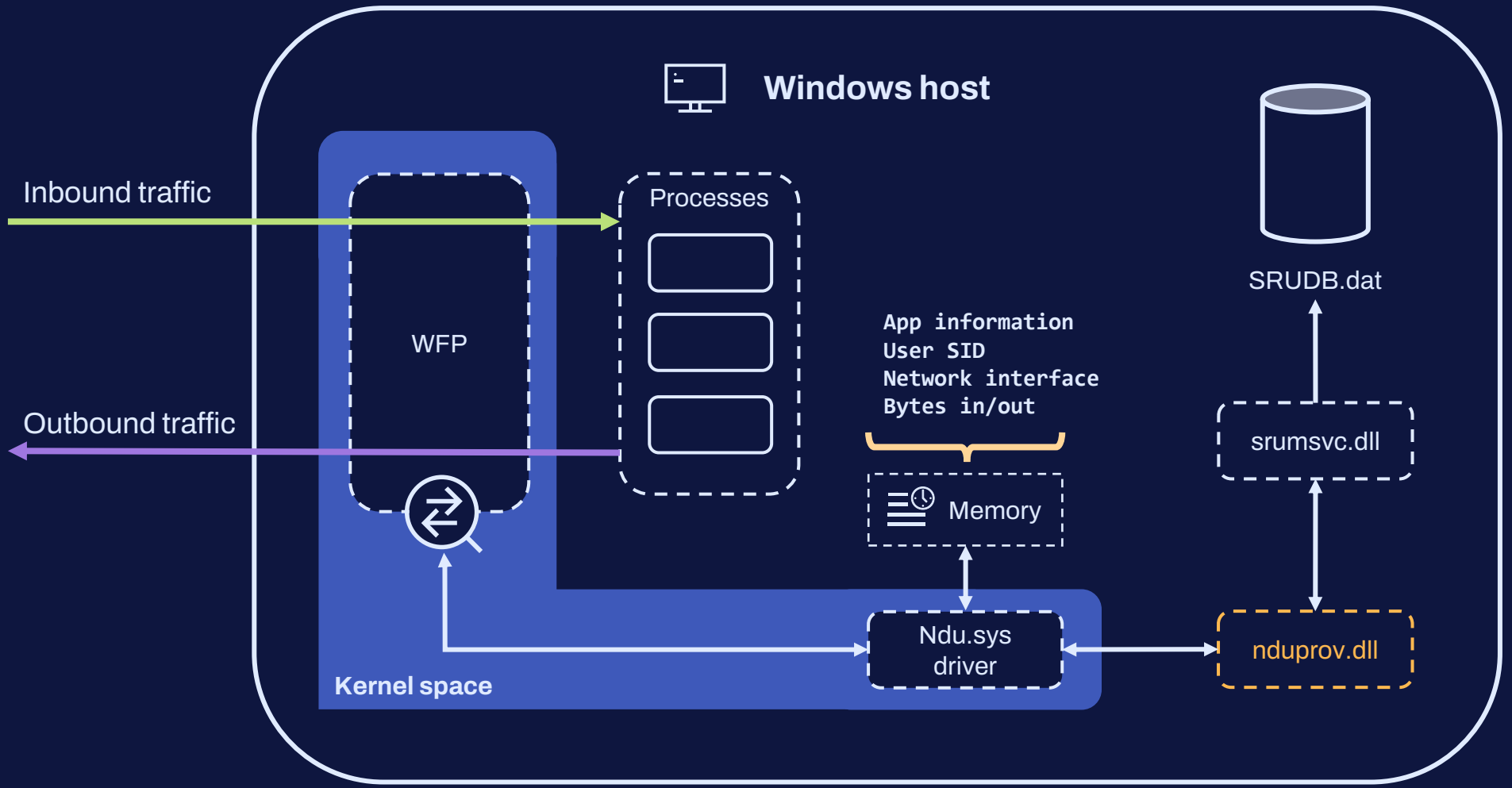
AutoIncId	TimeStamp	AppId	UserId	InterfaceLuid	L2ProfileId	L2ProfileFlags	BytesSent	BytesRecvd
55774	9/16/2023 3:30:30 PM	587	4	1689399632855040	0	0	8204	16714
55775	9/16/2023 3:30:30 PM	202	125	1689399632855040	0	0	35359	41786
55776	9/16/2023 3:30:30 PM	58	10	1689399632855040	0	0	3883	8914

The content of the Network Data Usage table in ESEDatabaseView

Focus on nduprov.dll

Monitoring & Analysing SRUM

In-depth analysis



WFP: Windows Filtering Platform

WITH secure

Focus on nduprov.dll

Monitoring & Analysing SRUM

In-depth analysis

- The **Network Data Usage driver** continuously monitors the **processes** by relying on the **WFP**.
- **All the processes are monitored**, **no exception found**.
- The **bytes in/out** calculation is based on the size of the **frames** (from the **Layer 2** of the **OSI model**).
- It does not provide any information on the exfiltrated files, only **an estimate of network usage (volume)**.

Focus on nduprov.dll

Monitoring & Analysing SRUM

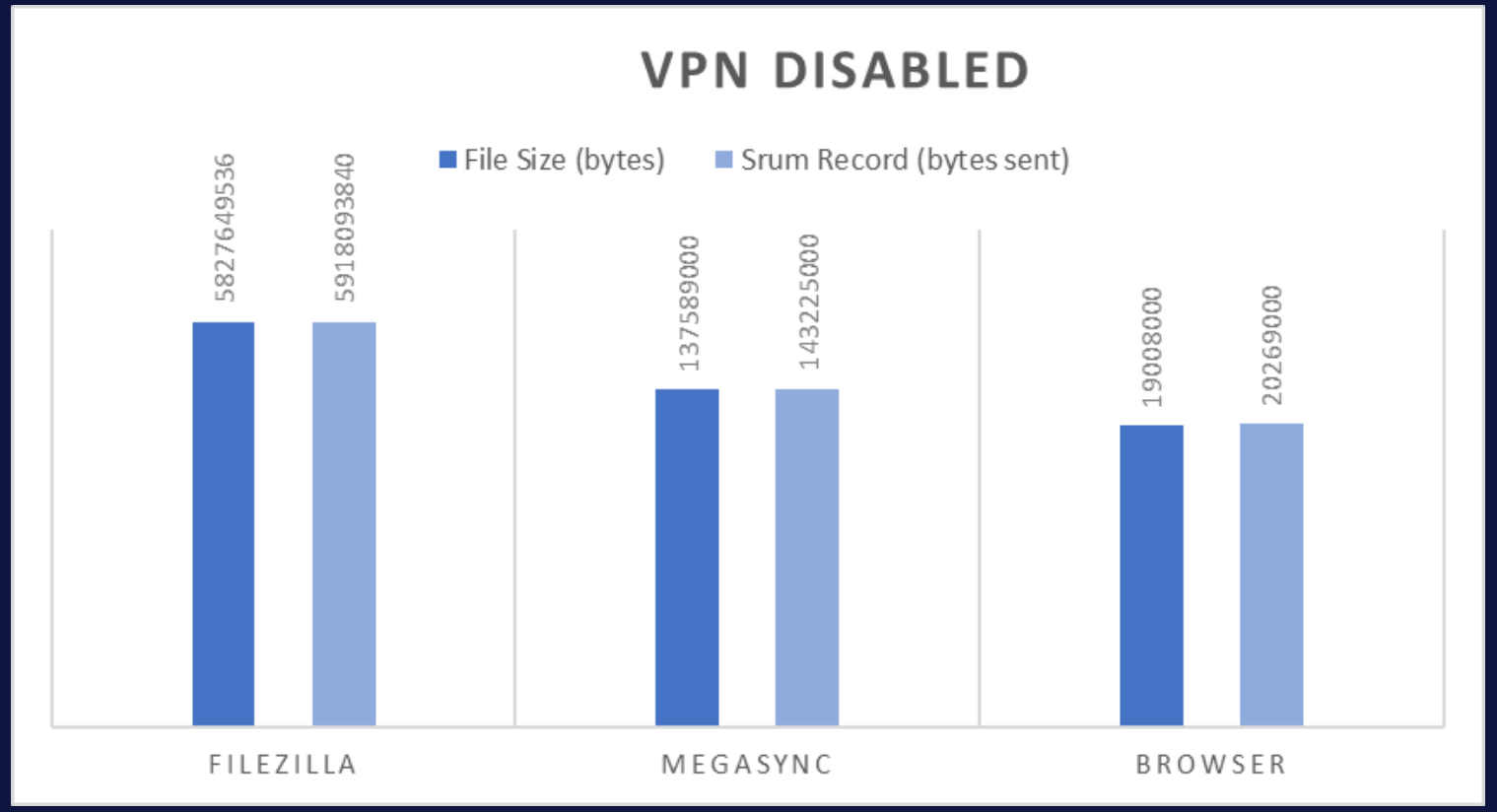
Experimentation

When is evidence recorded?

Tests	Result
Network **Network tests** were performed	• **Continuous monitoring** because of **Ndu.sys** • **Bytes in/out within the expected range** • E.g., For a file size transfer of 77.989.497 bytes, SRUM registered 79.414.089 bytes
	• **VPN** has an impact in the processes that will appear in the **Network Data Usage table**

Focus on nduprov.dll

Monitoring & Analysing SRUM



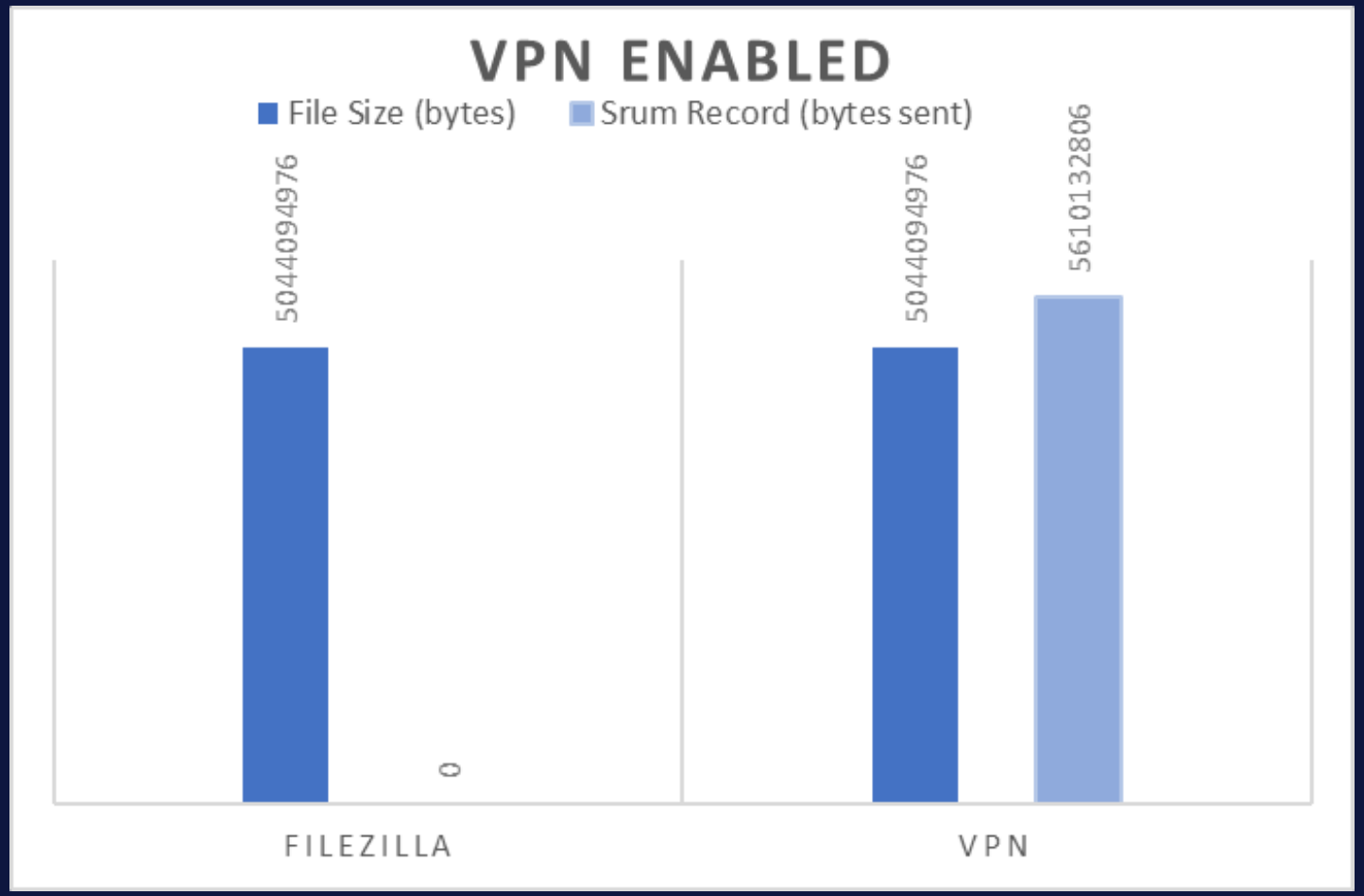
Experimentation

WITH secure

Focus on nduprov.dll

Monitoring & Analysing SRUM

Experimentation



Unsure at this point about this behaviour

Focus on App Timeline Provider

Monitoring & Analysing SRUM

Observation

- Table name: {5C8CF1C7-7257-4F13-B223-970EF5939312}
- 44 columns
- Some of the most interesting columns:
  - **EndTime** seems to be the timestamp of the approximate end of execution.
  - **DurationMS** is the total duration of execution (in milliseconds).

AutoIncId	TimeStamp	AppId	UserId	Flags	EndTime	DurationMS	SpanMS	TimelineEnd	InFocusTimeline	UserInputTimeline	...	KeyboardInputS	MouseInputS
516105	8/24/2023 12:50:30 PM	750	125	17563650	133373550300098449	120000	120000	413301130	3038287259199220266	3038287259199220266	...	707406378	707406378
516106	8/24/2023 12:50:30 PM	23	8	17563650	133373550300098449	120000	120000	413301130	3038287259199220266	3038287259199220266	...	707406378	707406378
516107	8/24/2023 12:50:30 PM	108	125	17563650	133373550300098449	60000	90000	413301126	3038287259199220266	3038287259199220266	...	707406378	707406378

The content of the App Timeline Provider table in ESEDatabaseView

Focus on App Timeline Provider

Monitoring & Analysing SRUM

In-depth analysis

- **Initialization**:
  - **Create** an **energy tracker** i.e., `NtPowerInformation(EnergyTrackerCreate,…)`
  - If successful, then the provider populates a srumsvc structure with the **callback functions**

- Callback function **`QueryStatsEx`** called every **Tier1Period**, i.e., 60 seconds by default
  - **Query** the **energy tracker** i.e., `NtPowerInformation(EnergyTrackerQuery,…)`

```
__kernel_entry NTSYSCALLAPI NTSTATUS NtPowerInformation(
  [in]           POWER_INFORMATION_LEVEL InformationLevel,
  [in, optional] PVOID                   InputBuffer,
  [in]           ULONG                   InputBufferLength,
  [out, optional] PVOID                  OutputBuffer,
  [in]           ULONG                   OutputBufferLength
);
```

The function prototype of NtPowerInformation

`EnergyTrackerCreate`
Indicates that the energy tracker is created.

`EnergyTrackerQuery`
Indicates that the energy tracker is queried.

POWER_INFORMATION_LEVEL constants related to the Energy Tracker

Focus on App Timeline Provider

Monitoring & Analysing SRUM

In-depth analysis

- The result of the **query** to the **energy tracker** is an **_ENERGY_TRACKER_DATA** structure.

_ENERGY_TRACKER_DATA	
…	…
0x10	int SpanMS
0x14	int DurationMS
…	
0x20	int TimeMS_4096
0x40	int EndTime
…	…

→ … →

_PROCESS_EXTENDED_ENERGY_VALUES	
0x00	_PROCESS_ENERGY_VALUES Base
0x110	_PROCESS_ENERGY_VALUES_EXTENSION Extension

↓

AutoIncId	TimeStamp	AppId	UserId	Flags	EndTime	DurationMS	SpanMS	TimelineEnd	InFocusTimeline	UserInputTimeline	…	KeyboardInputS	MouseInputS
516105	8/24/2023 12:50:30 PM	750	125	17563650	133373550300098449	120000	120000	413301130	3038287259199220266	3038287259199220266	…	707406378	707406378
516106	8/24/2023 12:50:30 PM	23	8	17563650	133373550300098449	120000	120000	413301130	3038287259199220266	3038287259199220266	…	707406378	707406378
516107	8/24/2023 12:50:30 PM	108	125	17563650	133373550300098449	60000	90000	413301126	3038287259199220266	3038287259199220266	…	707406378	707406378

Present in table

Focus on App Timeline Provider

Monitoring & Analysing SRUM

Observation

- App Timeline Provider generates **specific app names** that are formatted as follows:

AppX package name	!	AppX package ID (PRAID)	!	Executable name	!	Compilation/Linking Timestamp	!	PE checksum	!

  - `Microsoft.YourPhone_1.23072.153.0_x64__8wekyb3d8bbwe!runtimebroker07f4358a809ac99a64a67c1!RuntimeBroker.exe!2018/08/18:19:29:32!1d738!`
  - `Microsoft.WindowsStore_22212.1401.8.0_x64__8wekyb3d8bbwe!App!WinStore.App.exe!2023/02/10:07:46:13!0!`
  - `!!msedge.exe!2023/09/11:23:06:04!3f0639!`

- These 5 values can help to identify the **exact binary** that was executed

Focus on App Timeline Provider

Monitoring & Analysing SRUM

In-depth analysis

- **Identical values** in the **App Timeline Provider table**

CpuTimeline	DiskTimeline	NetworkTimeline	MBBTimeline	InFocusS	PSMForegroundS
3661335631	3038287259199220266	3038287259199220266	3038287259199220266	707406378	707406378
2147483713	3038287259199220266	3038287259199220266	3038287259199220266	707406378	707406378
1099511627773	3038287259199220266	3038287259199220266	3038287259199220266	707406378	707406378
2199023255551	3038287259199220266	3038287259199220266	3038287259199220266	707406378	707406378
3038287259199220266	3038287259199220266	3038287259199220266	3038287259199220266	40	707406378
536887301	3038287259199220266	3038287259199220266	3038287259199220266	707406378	707406378

- In **ESE databases**, if the column value is **NULL**, then a **default value** is used

Type of value	Default value (if none specified)
1 byte	42 (0x2a)
2 bytes	10794 (0x2a2a)
4 bytes	707406378 (0x2a2a2a2a)
8 bytes	3038287259199220266 (0x2a2a2a2a2a2a2a2a)
DateTime	December 30 1899 12:00:00 AM
Float	0.0

The remaining column types (**text** and **binary**) cannot be set to **NULL**

WITH secure

Focus on App Timeline Provider

Monitoring & Analysing SRUM

Experimentation

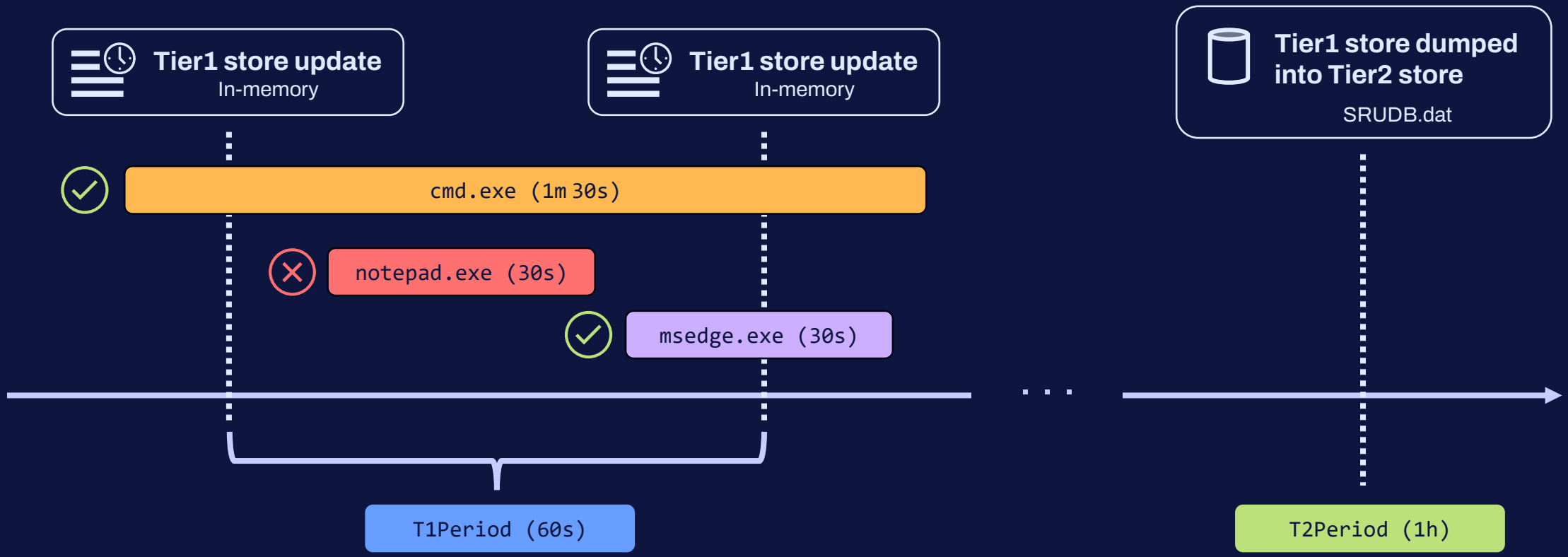
When is evidence recorded?

Tests	Result
 Application Execution of **GUI** and **command line applications**	• If the data **reaches** the **Tier1 store**, then it will be registered in the **Tier2 store**, i.e., **SRUDB.dat** i.e., the process needs to be **running** when Tier1 store is updated every Tier1Period (60s).

Focus on App Timeline Provider

Monitoring & Analysing SRUM

Experimentation



How long is the evidence stored?

Monitoring & Analysing SRUM

In-depth analysis

- **By default**, the **retention period** is **60 days** and can be calculated as follows (in seconds):

```
Tier2Period * Tier2MaxEntries
```

where, by default,

```
Tier2Period = 3600
Tier2MaxEntries = 1440
```

- On **Windows Server**, many extensions define their own **`Tier2MaxEntries`** value, which is usually equal to **9000.** The retention period becomes **375 days**.

- For **long term** tables, i.e., ending in **"}LT"**, it is **1820 days ~ 5 years**:

```
Tier2LongTermPeriod * Tier2LongTermMaxEntries
```

where, by default,

```
Tier2LongTermPeriod = 604800 (7 days)
Tier2LongTermMaxEntries = 260
```

Summary - Forensic Insights

Monitoring & Analysing SRUM

Forensic insights

How can SRUM be used as a forensic artefact?	It can be used as a forensic artefact to prove **execution** and **data exfiltration**.
On which Windows versions is SRUM available?	**On Windows Desktop, it is available since Windows 8**, but on **Windows Server**, it is only available on **specific versions**.
Where is the SRUM data stored?	There are 2 types of storage: **Tier1 store (in-memory)**, **Tier2 store on disk (`C:\Windows\System32\sru\SRUDB.dat`)**. The registry **no longer** acts as a temporary storage for the database records.
When is the SRUM data written to the database?	The Tier1 store (in-memory) is dumped into the Tier2 store (SRUDB.dat): • every **Tier2Period** (1h), • when the machine is **shutdown,** • and if **DPS** is **stopped**.
Are the SRUM event logs available on Windows?	Windows has **WINEVT channels** used to receive the logs from SRUM components and extensions. However, they are **not enabled by default**.

Summary - Forensic Insights

Monitoring & Analysing SRUM

Forensic insights

Is there a scenario where an executable does not appear in the App Timeline Provider table?	A process needs to be **running** when the **Tier1 store** gets updated, every **Tier1Period** (60 s). Otherwise, there won't be a record of the execution of that program.
Is there a scenario where the network traffic of a process is not monitored?	No exceptions were found, the network traffic is **continuously monitored** because of the **Ndu.sys driver**.
What do the bytes in/out represent?	The **bytes in/out** include the **frames** (from the **layer 2** of the **OSI model**).
What if a VPN is enabled on the system?	If the network traffic goes through a **VPN**, then **all the bytes in/out will be associated with the VPN process/service**.
How long is the evidence stored in the database?	On Windows Desktop and Server, it is by default **60 days**. On Windows Server, we usually see **375 days** since the **Tier2MaxEntries** is specified in the registry.

Next steps

Chainsaw

Next steps

- We encountered **errors** when parsing the SRUM database with tools like SrumEcmd, SrumMonkey
- A **new version** of **Chainsaw**, an **open-source tool** developed by **WithSecure**, will be published to:
  - Dump the **raw content** of **ESE databases**
  - **Analyse the SRUM database** and provide **insights**

By WithSecure Countercept (@FranticTyping, @AlexKornitzer)

```
[+] ESE database file loaded from "/home/cat/Documents/GitHub/DFIRArtifactMuseum/Windows/SRUM/Win10/APTSimulatorVM/Clean/SRUDB.dat"
[+] Parsing the ESE database...
[+] Parsing the SOFTWARE registry hive ...
[+] Analysing the SRUM database (all tables)
[+] Details about the tables related to the SRUM extensions:
```

Table GUID	Table Name	DLL Path	Timeframe of the data	Expected Retention Time
{D10CA2FE-6FCF-4F6D-848E-B2E99266FA89}	Application Resource Usage Provider	%SystemRoot%\\System32\\appsruprov.dll	2022-03-10 16:34:59 UTC 2022-03-10 21:10:00 UTC	60 days
{D10CA2FE-6FCF-4F6D-848E-B2E99266FA86}	WPN SRUM Provider	%SystemRoot%\\System32\\wpnsruprov.dll	2022-03-10 20:09:00 UTC 2022-03-10 21:09:00 UTC	60 days
{973F5D5C-1D90-4944-BE8E-24B94231A174}	Windows Network Data Usage Monitor	%SystemRoot%\\System32\\nduprov.dll	2022-03-10 16:34:59 UTC 2022-03-10 21:10:00 UTC	60 days
{DD6636C4-8929-4683-974E-22C046A43763}	Windows Network Connectivity Usage Monitor	%SystemRoot%\\System32\\ncuprov.dll	2022-03-10 16:34:59 UTC 2022-03-10 21:10:00 UTC	60 days
{FEE4E14F-02A9-4550-B5CE-5FA2DA202E37}LT	Energy Usage Provider (Long Term)	%SystemRoot%\\System32\\energyprov.dll	No records	1820 days
{B6D82AF1-F780-4E17-8077-6CB9AD8A6FC4}	Tagged Energy Provider	%SystemRoot%\\System32\\eeprov.dll	No records	3 days
{FEE4E14F-02A9-4550-B5CE-5FA2DA202E37}	Energy Usage Provider	%SystemRoot%\\System32\\energyprov.dll	No records	60 days
{7ACBBAA3-D029-4BE4-9A7A-0885927F1D8F}	vfuprov	%SystemRoot%\\System32\\vfuprov.dll	2022-03-10 20:09:00 UTC 2022-03-10 21:10:00 UTC	60 days
{5C8CF1C7-7257-4F13-B223-970EF5939312}	App Timeline Provider	%SystemRoot%\\System32\\eeprov.dll	2022-03-10 16:34:59 UTC 2022-03-10 21:10:00 UTC	7 days
{DA73FB89-2BEA-4DDC-86B8-6E048C6DA477}	Energy Estimation Provider	%SystemRoot%\\System32\\eeprov.dll	No records	7 days

```
[+] SRUM database parsed successfully
[+] Saving output to "/home/cat/Documents/GitHub/analyse_srum.json"
[+] Saved output to "/home/cat/Documents/GitHub/analyse_srum.json"
```

Future work

Next steps

- Publish a **paper/article** containing more details about the research
- **Continue** the research about:
  - the analysis of **eeprov.dll**
  - understanding how a **VPN** impacts the monitoring of the **network data usage**
  - **additional components and extensions**
    - Application Resource Usage (**appsruprov.dll**)
    - Network Connectivity Usage (**ncuprov.dll**)
  - the **differences** between the **SRUM** on **Windows Desktop** and on **Windows Server**
- **Investigate** other **forensic artefacts** that prove **execution** on Windows

References

- https://www.sciencedirect.com/science/article/abs/pii/S1742287615000031 - Forensic implications of System Resource Usage Monitor (SRUM) data in Windows 8, Yogesh Khatri March 2015
- https://blog.1234n6.com/2019/01/some-testing-of-srum-on-windows-server.html
- https://blog.1234n6.com/2019/01/testing-of-srum-on-windows-server-2019.html
- https://www.hecfblog.com/2019/01/daily-blog-595-solution-saturday-11219.html
- https://winbindex.m417z.com/
- https://velociraptor.velocidex.com/digging-into-the-system-resource-usage-monitor-srum-afbadb1a375
- https://www.magnetforensics.com/blog/srum-forensic-analysis-of-windows-system-resource-utilization-monitor/?utm_source=Google&utm_medium=Search&utm_campaign=2023_Resource_Centre&gad=1&gclid=EAIaIQobChMIsLngm8LIgQMVWRV7Ch2ODAYgEAAYASAAEgKOofD_BwE
- https://www.vergiliusproject.com/
- https://github.com/libyal/esedb-kb/blob/main/documentation/System%20Resource%20Usage%20Monitor%20(SRUM).asciidoc
- https://aboutdfir.com/app-timeline-provider-srum-database/

WITH®

secure