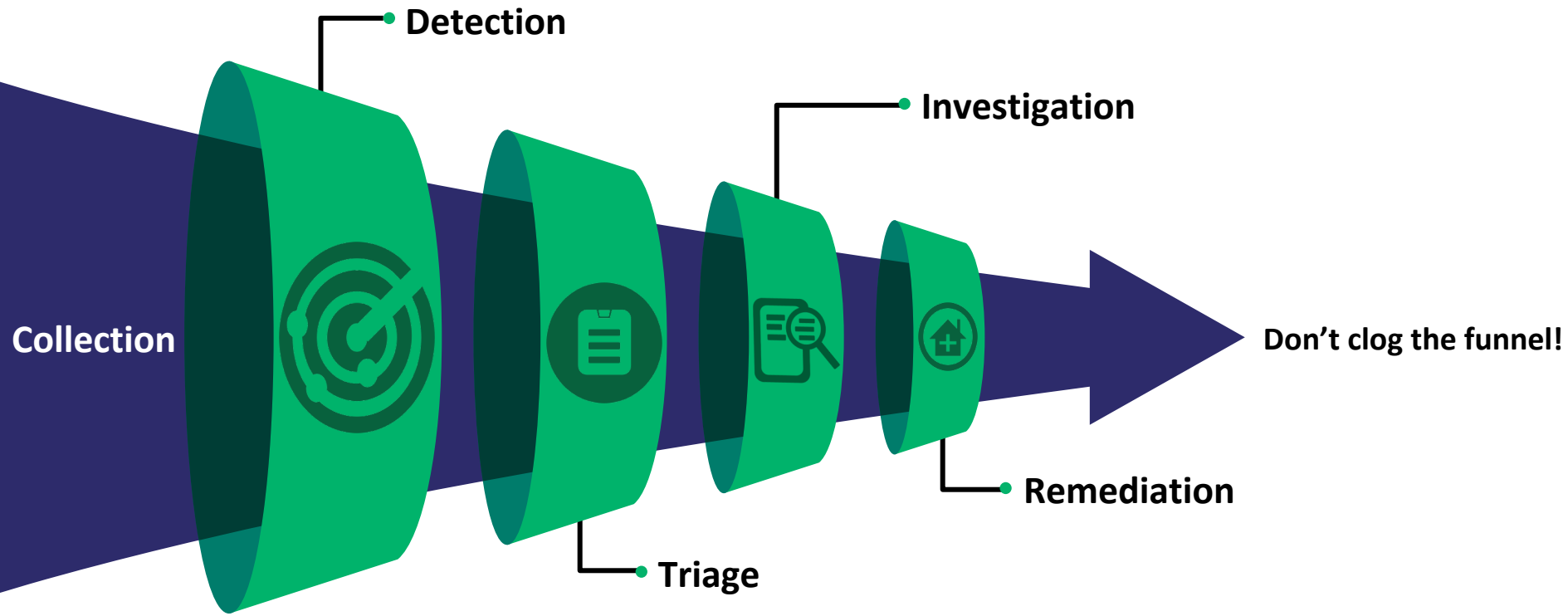
Abstraction Workshop

SPECTER OPS

Lab Infrastructure

Labs:
https://jaredcatkinson.github.io/abstraction-workshop/

Funnel of Fidelity

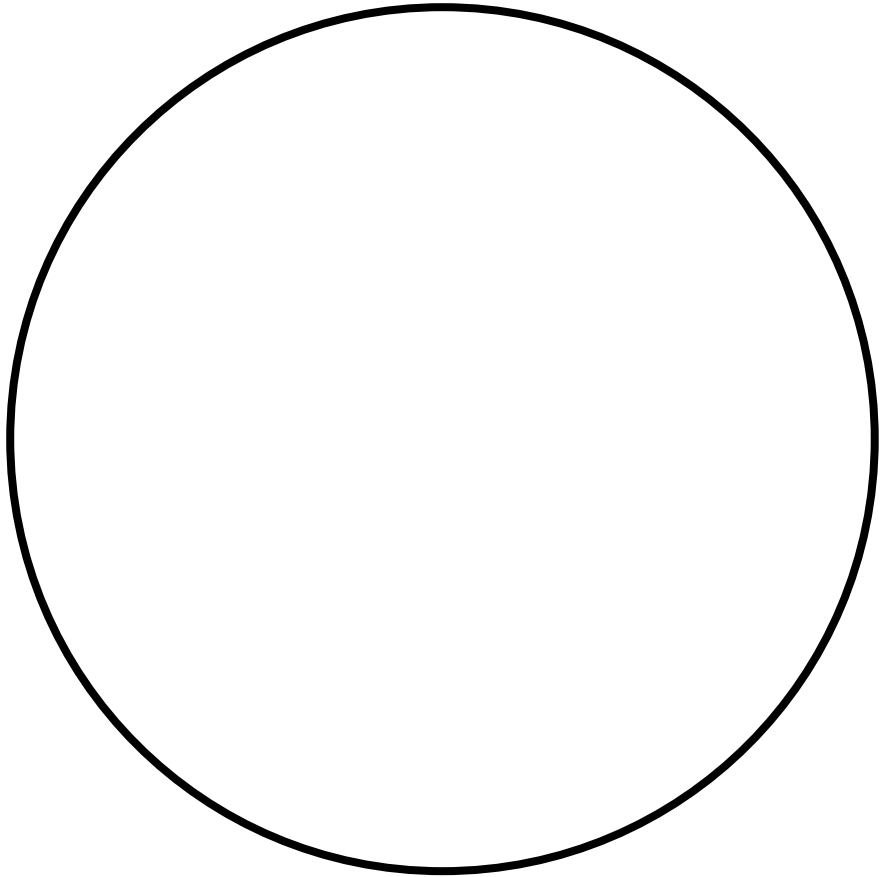


Capability Abstraction

- When Detection Engineers create detections, they traditionally haven't had a method of determining quality of coverage
- Focuses on identifying what technology is abstracted by a tool
- Once the abstraction layers are identified, the detection engineer can test the technique at each layer
- Higher layers (tools) are more superficial
- Lower layers (network protocols) are more comprehensive
- Typically attackers can execute an attack at several different layers

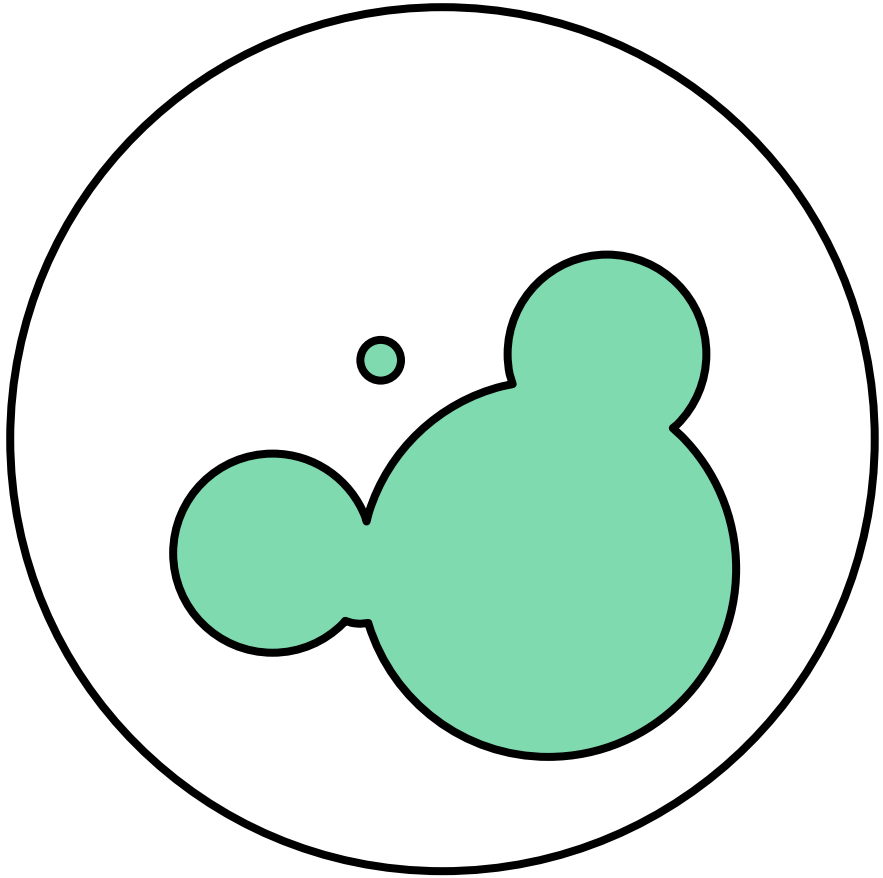
Population

- A set of similar items or events which is of interest for some question or experiment.
- Example
  - All people tested for a disease
  - All telemetry collected from enterprise assets



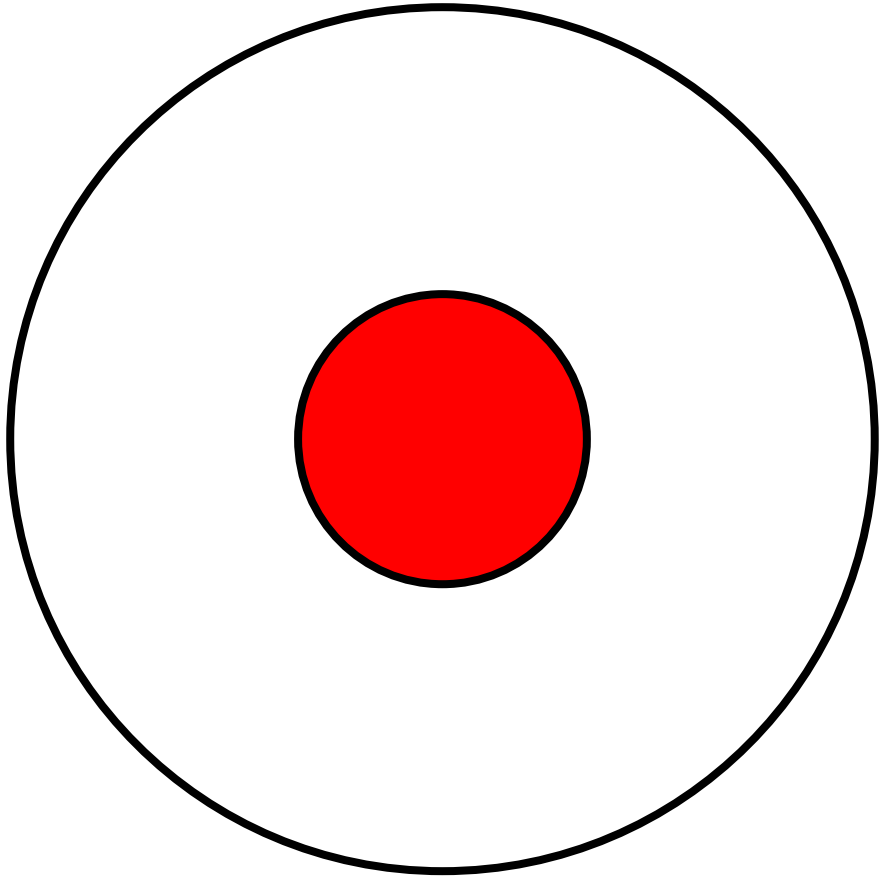
Classification Rule

- A procedure by which the elements of the population are each predicted to belong to one of the classes
  - In Binary Classification there are only two classes (positive and negative)
- Examples
  - People who test positive for the disease
  - Events that are classified as malicious services



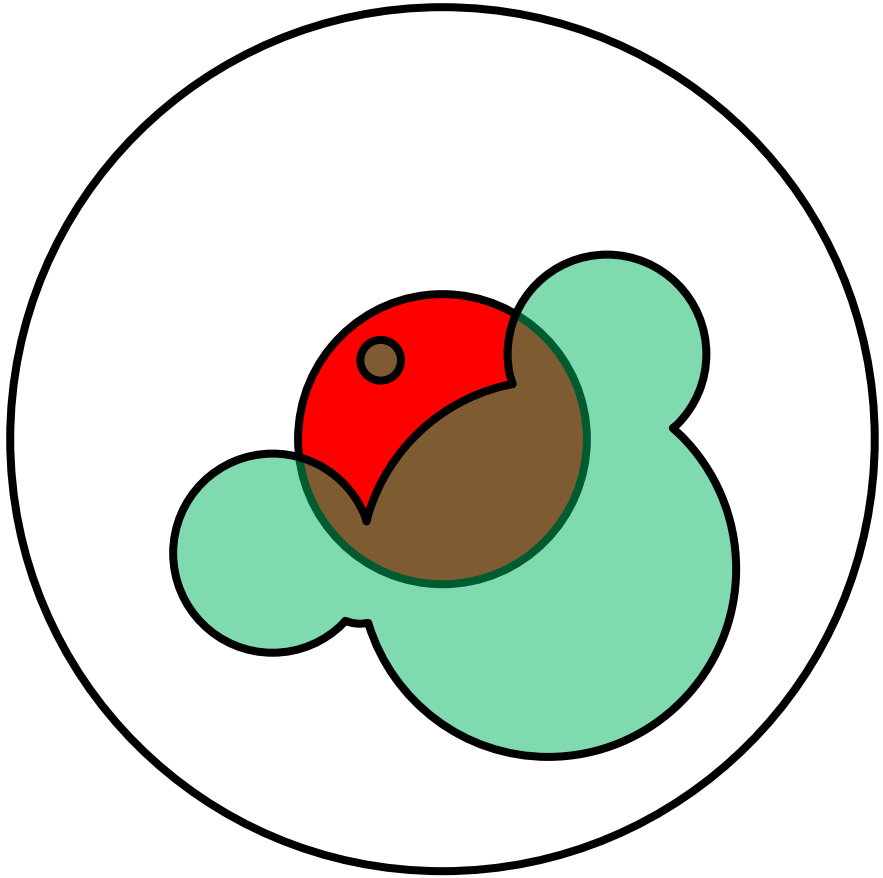
Condition Positive

- Represents the real number of positive events in the collected data
- Examples
  - The objective sub population of sick people (positive for disease)
  - The objective sub population of malicious events



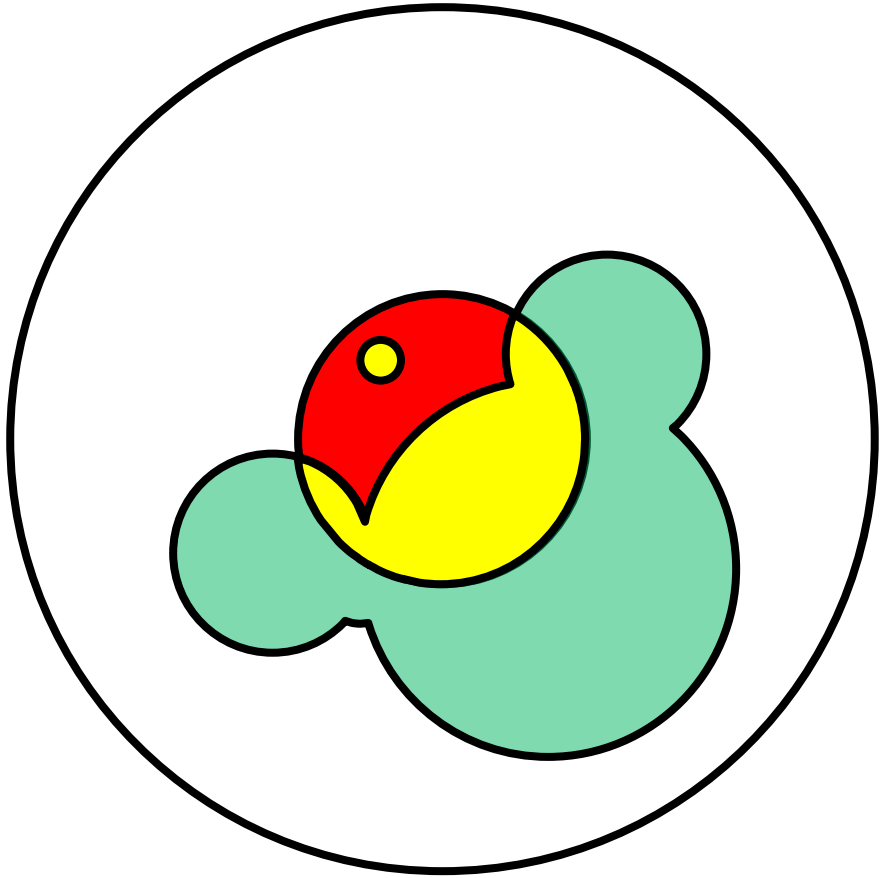
Condition Positive

- Represents the real number of positive events in the collected data
- Examples
  - The objective sub population of sick people (positive for disease)
  - The objective sub population of malicious events
- Two possible predictions:
  - True Positive
  - False Negative



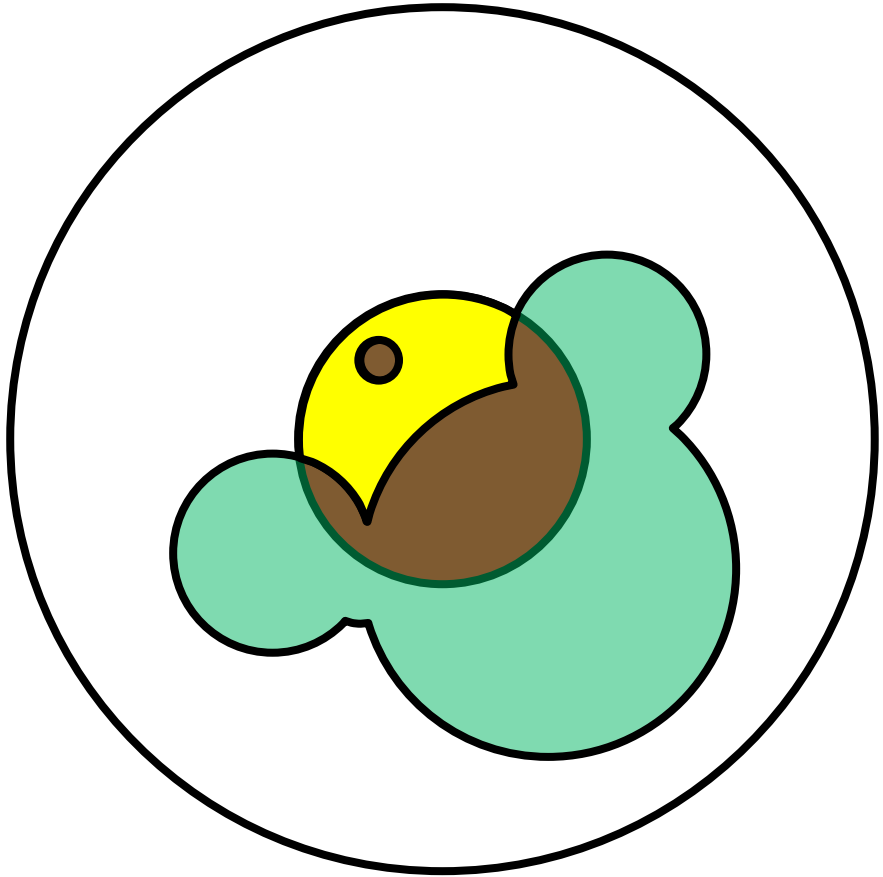
True Positives (TP)

- Positive events that are classified as positive
- Examples
  - Sick people correctly identified as sick
  - Malicious services correctly identified as malicious



False Negatives (FN)

- Positive events that are classified as negative
- Examples
  - Sick people incorrectly identified as healthy
  - Malicious services incorrectly identified as benign
- Problem
  - False Negatives fail silently
  - Risk is implicitly assumed



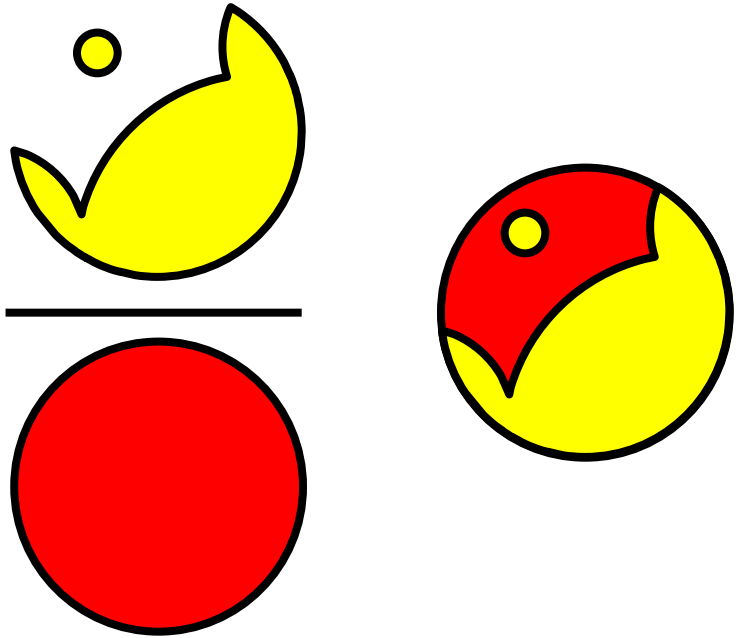
Sensitivity

- Measures the ability of the test to correctly detect events that meet the condition
- How likely are you to identify sick people as being sick?

$$\text{sensitivity} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}}$$

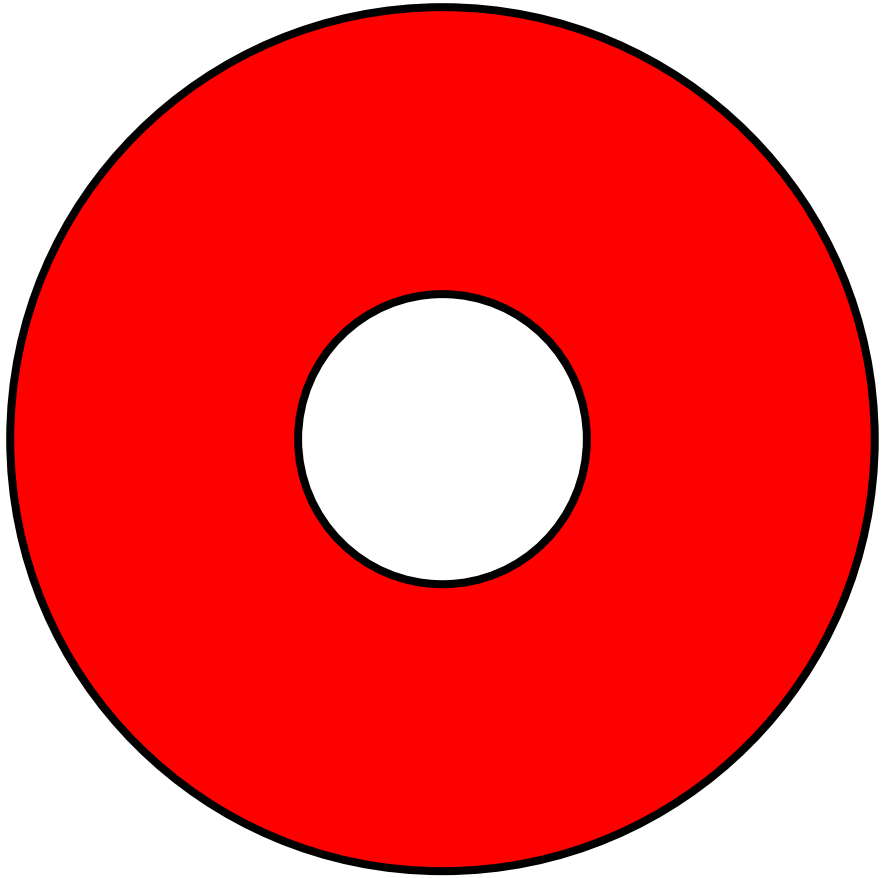
$$= \frac{\text{number of true positives}}{\text{total number of sick individuals in population}}$$

$$= \text{probability of a positive test given that the patient has the disease}$$



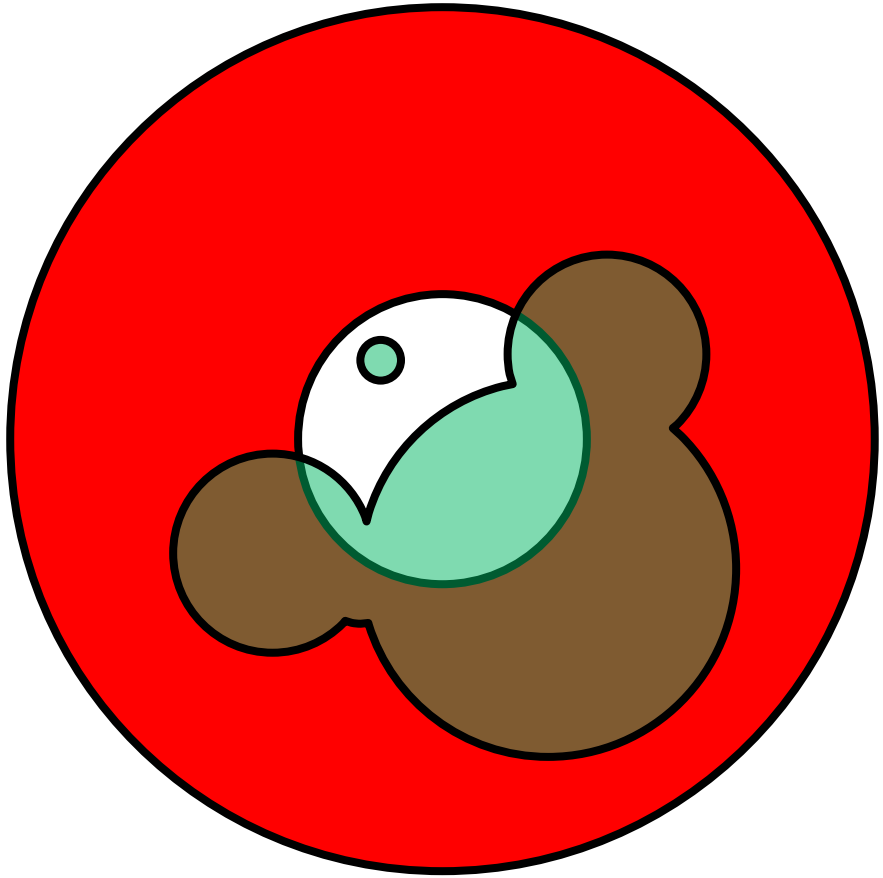
Condition Negative

- Represents the real number of negative events in the collected data.
- Examples
  - The objective sub population of healthy people (negative for disease)
  - The objective sub population of benign events



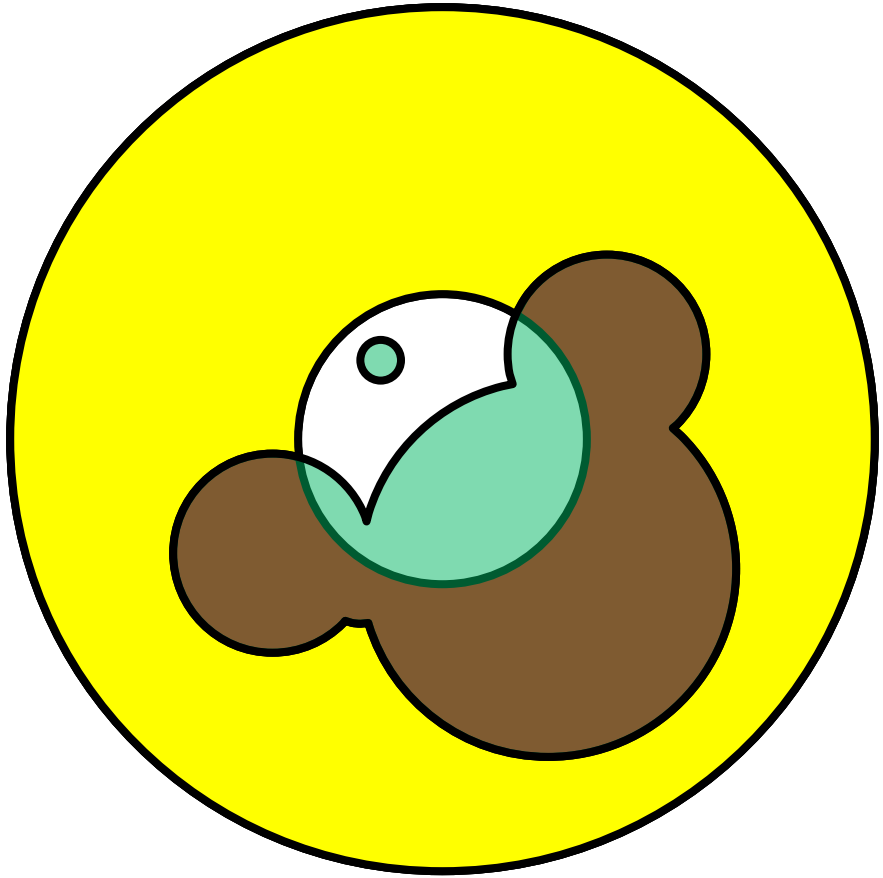
Condition Negative

- Represents the real number of negative events in the collected data.
- Examples
  - The objective sub population of healthy people (negative for disease)
  - The objective sub population of benign events
- Two possible predictions:
  - True Negative
  - False Positive



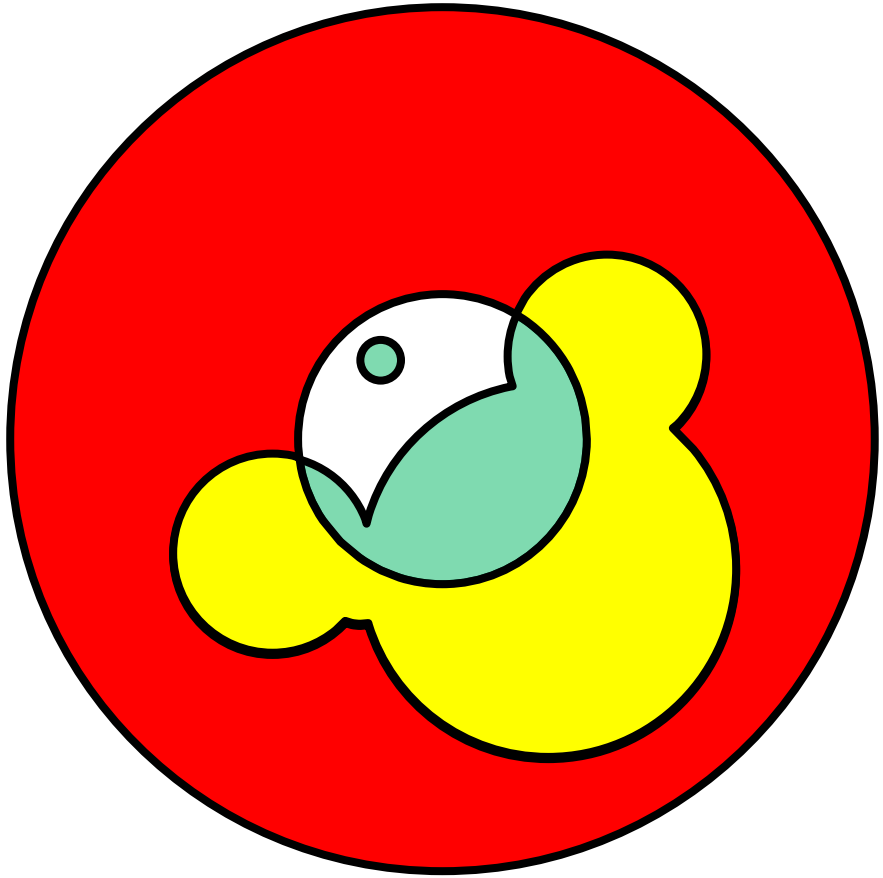
True Negatives (TN)

- Negative events that are classified as negative
- Examples
  - A healthy person correctly identified as healthy
  - A benign service correctly identified as benign



False Positives (FP)

- Negative events that are classified as positive
- Examples
  - Healthy people incorrectly identified as sick
  - Benign services incorrectly identified as malicious
- Problem
  - False Positives fail noisily
  - Triage and Investigation resources may be diverted from actual malicious activity



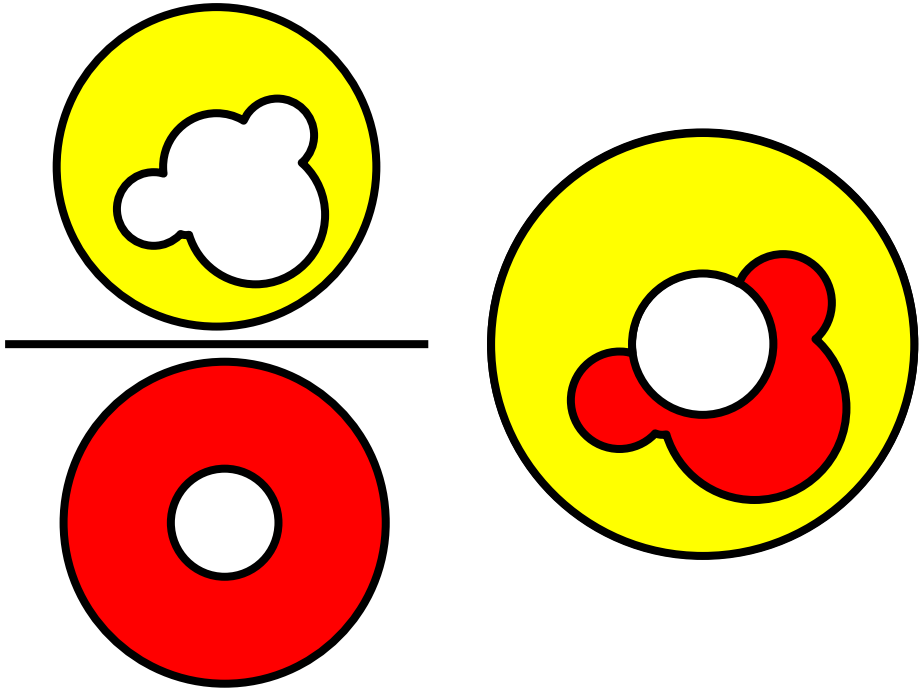
Specificity

- Measures the ability of the test to correctly reject events that do not meet the condition
- How likely are you to identify

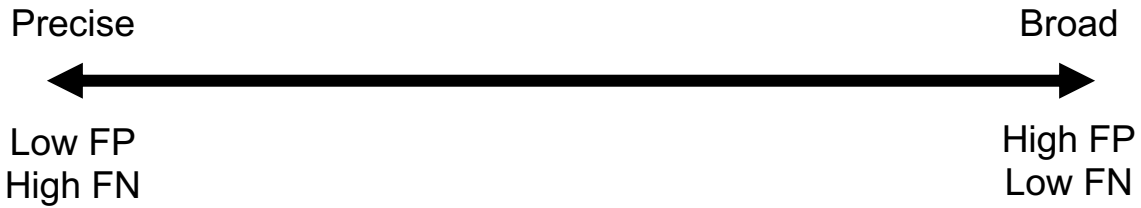
$$\text{specificity} = \frac{\text{number of true negatives}}{\text{number of true negatives} + \text{number of false positives}}$$

$$= \frac{\text{number of true negatives}}{\text{total number of well individuals in population}}$$

$$= \text{probability of a negative test given that the patient is well}$$



Detection Spectrum



Abstraction Questions

1. **Technique:** What Technique are we interested in?
2. **Tools:** What Tool(s) do we know perform this technique?
3. **Functions:** What API Function enables the technique?
4. **Files:** Does this technique require the creation or access of any files?
5. **Registry:** Does this technique require configuration from the registry?
6. **Network:** Does this technique require network activity to occur?

Technique

- Hunts or Detection Engineering efforts require a goal
- What technique are we hunting (creating a detection) for?

- New Service
  - Services are long running applications that are typically transparent to the user
  - Can be set to execute upon boot up
  - Often run in an elevated user context
  - Attackers can leverage services for Persistence (starting at boot up) or Privilege Escalation (execute code in an elevated context)

Specific Service Name – Google Update

Tree: 48d95f027c ▾ **sigma** / rules / windows / builtin / **win_apt_apt29_tor.yml** Find file | Copy path

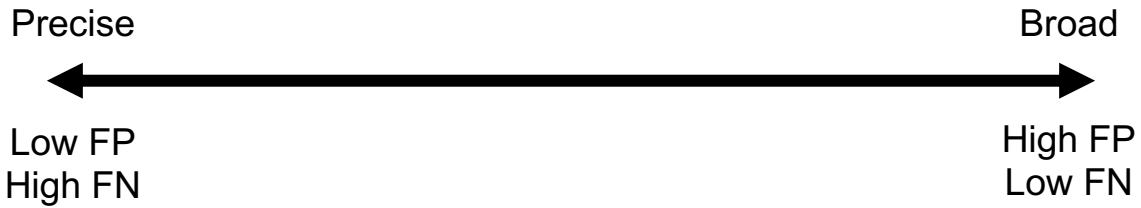
thomaspatzke Merge branch 'master' into oscd d7bd90c on Feb 3

1 contributor

Executable File | 38 lines (38 sloc) | 1.12 KB Raw | Blame | History

```
1  action: global
2  title: APT29 Google Update Service Install
3  id: c069f460-2b87-4010-8dcf-e45bab362624
4  description: This method detects malicious services mentioned in APT29 report by FireEye. The legitimate path for the Google up
5      so the service names and executable locations used by APT29 are specific enough to be detected in log files.
6  references:
7      - https://www.fireeye.com/blog/threat-research/2017/03/apt29_domain_frontin.html
8  tags:
9      - attack.persistence
10     - attack.g0016
11     - attack.t1050
12 date: 2017/11/01
13 author: Thomas Patzke
14 logsource:
15     product: windows
16     service: system
17 detection:
18     service_install:
19         EventID: 7045
20         ServiceName: 'Google Update'
21     timeframe: 5m
22     condition: service_install | near process
23 falsepositives:
24     - Unknown
25 level: high
26 ---
27 logsource:
28     category: process_creation
29     product: windows
30 detection:
31     process:
32         Image:
33             - 'C:\Program Files(x86)\Google\GoogleService.exe'
34             - 'C:\Program Files(x86)\Google\GoogleUpdate.exe'
35 fields:
36     - ComputerName
37     - User
38     - CommandLine
```

Detection Spectrum



Specific Service Name – WerFaultSvc

Tree: aa8a0f5e1f ▾ sigma / rules / windows / builtin / win_apt_turla_service_png.yml Find file Copy path

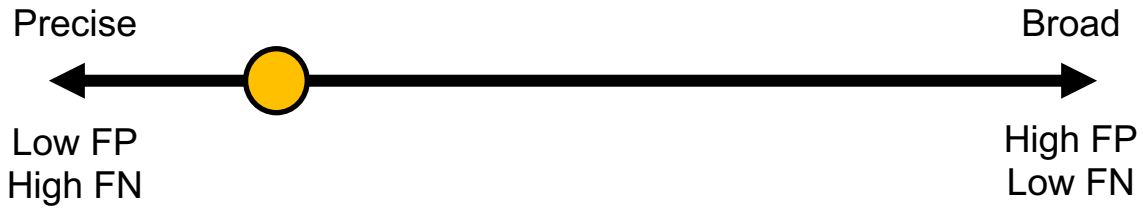
Florian Roth refactor: moved rues from 'apt' folder in respective folders 03ecb3b on Feb 1

0 contributors

22 lines (22 sloc) | 632 Bytes Raw Blame History

```yaml
title: Turla PNG Dropper Service
id: 1228f8e2-7e79-4dea-b0ad-c91f1d5016c1
description: This method detects malicious services mentioned in Turla PNG dropper report by NCC Group in November 2018
references:
    - https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2018/november/turla-png-dropper-is-back/
author: Florian Roth
date: 2018/11/23
tags:
    - attack.persistence
    - attack.g0010
    - attack.t1050
logsource:
    product: windows
    service: system
detection:
    selection:
        EventID: 7045
        ServiceName: 'WerFaultSvc'
    condition: selection
falsepositives:
    - unlikely
level: critical
```

Detection Spectrum



Specific Service Name – NtsSrv

Tree: aa8a0f5e1f ▾ **sigma** / rules / windows / builtin / **win_apt_stonedrill.yml** Find file | Copy path

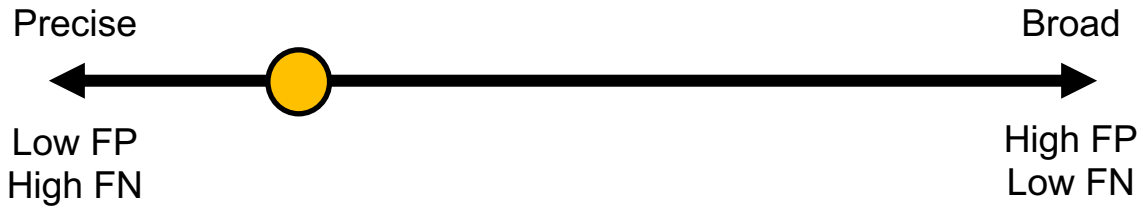
Florian Roth refactor: moved rues from 'apt' folder in respective folders 03ecb3b on Feb 1

0 contributors

Executable File | 23 lines (23 sloc) | 675 Bytes Raw | Blame | History

```
1  title: StoneDrill Service Install
2  id: 9e987c6c-4c1e-40d8-bd85-dd26fba8fdd6
3  description: This method detects a service install of the malicious Microsoft Network Realtime Inspection Service service descr
4  author: Florian Roth
5  date: 2017/03/07
6  references:
7      - https://securelist.com/blog/research/77725/from-shamoon-to-stonedrill/
8  tags:
9      - attack.persistence
10     - attack.g0064
11     - attack.t1050
12 logsource:
13     product: windows
14     service: system
15 detection:
16     selection:
17         EventID: 7045
18         ServiceName: NtsSrv
19         ServiceFileName: '* LocalService'
20     condition: selection
21 falsepositives:
22     - Unlikely
23 level: high
```

Detection Spectrum



Specific Service Name – Carbon Paper

Tree: aa8a0f5e1f ▾ **sigma** / rules / windows / builtin / **win_apt_carbonpaper_turla.yml** Find file | Copy path

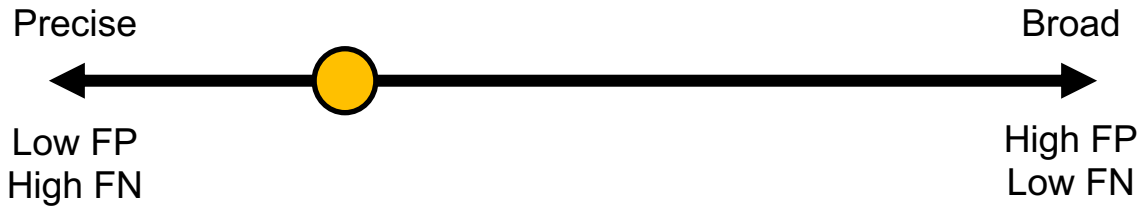
Florian Roth refactor: moved rues from 'apt' folder in respective folders 03ecb3b on Feb 1

0 contributors

Executable File | 25 lines (25 sloc) | 669 Bytes Raw | Blame | History

```
1  title: Turla Service Install
2  id: 1df8b3da-b0ac-4d8a-b7c7-6cb7c24160e4
3  description: This method detects a service install of malicious services mentioned in Carbon Paper - Turla report by ESET
4  references:
5      - https://www.welivesecurity.com/2017/03/30/carbon-paper-peering-turlas-second-stage-backdoor/
6  tags:
7      - attack.persistence
8      - attack.g0010
9      - attack.t1050
10 date: 2017/03/31
11 author: Florian Roth
12 logsource:
13     product: windows
14     service: system
15 detection:
16     selection:
17         EventID: 7045
18         ServiceName:
19             - 'srservice'
20             - 'ipvpn'
21             - 'hkmsvc'
22     condition: selection
23 falsepositives:
24     - Unknown
25 level: high
```

Detection Spectrum



Specific Service Name

Tree: 48d95f027c ▾ **sigma** / rules / windows / builtin / **win_mal_service_installs.yml** Find file | Copy path

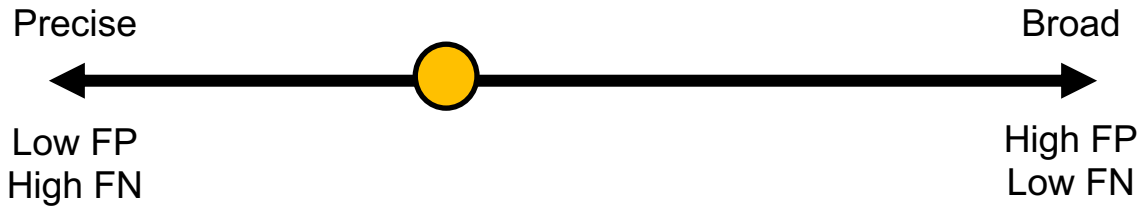
thomaspatzke Fixes 01d6c3b on Feb 16

2 contributors

29 lines (29 sloc) | 869 Bytes Raw | Blame | History

```yaml
title: Malicious Service Installations
id: 5a105d34-05fc-401e-8553-272b45c1522d
description: Detects known malicious service installs that only appear in cases of lateral movement, credential dumping and oth
author: Florian Roth, Daniil Yugoslavskiy, oscd.community (update)
date: 2017/03/27
modified: 2019/11/01
tags:
    - attack.persistence
    - attack.privilege_escalation
    - attack.t1003
    - attack.t1035
    - attack.t1050
    - car.2013-09-005
logsource:
    product: windows
    service: system
detection:
    selection:
        EventID: 7045
    malsvc_paexec:
        ServiceFileName|contains: '\PAExec'
    malsvc_wannacry:
        ServiceName: 'mssecsvc2.0'
    malsvc_persistence:
        ServiceFileName|contains: 'net user'
    condition: selection and 1 of malsvc_*
falsepositives:
    - Penetration testing
level: critical
```

Detection Spectrum



Service Creation w/ SC.exe or PowerShell

Tree: 48d95f027c ▾ **sigma** / rules / windows / process_creation / **win_new_service_creation.yml** Find file | Copy path

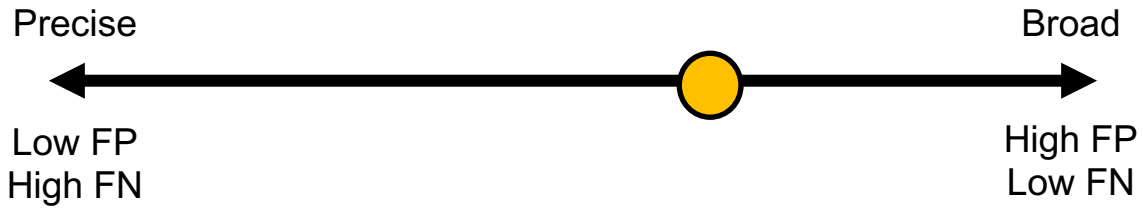
thomaspatzke Rule fixes 373424f on Feb 20

2 contributors

28 lines (28 sloc) | 846 Bytes Raw | Blame | History

```yaml
title: New Service Creation
id: 7fe71fc9-de3b-432a-8d57-8c809efc10ab
status: experimental
description: Detects creation if a new service
author: Timur Zinniatullin, Daniil Yugoslavskiy, oscd.community
date: 2019/10/21
modified: 2019/11/04
tags:
    - attack.persistence
    - attack.privilege_escalation
    - attack.t1050
references:
    - https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1050/T1050.yaml
logsource:
    category: process_creation
    product: windows
detection:
    selection:
      - Image|endswith: '\sc.exe'
        CommandLine|contains|all:
            - 'create'
            - 'binpath'
      - Image|endswith: '\powershell.exe'
        CommandLine|contains: 'new-service'
    condition: selection
falsepositives:
    - Legitimate administrator or user creates a service for legitimate reason
level: low
```

Detection Spectrum



Rare Service Creation via Windows Events

Tree: c10332b06c ▾ **sigma** / rules / windows / builtin / **win_rare_service_installs.yml** Find file | Copy path

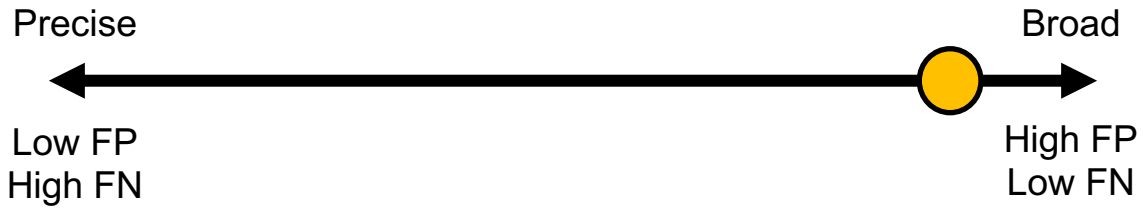
Florian Roth fix: fixed missing date fields in remaining files e79e99c on Jan 30

2 contributors

23 lines (23 sloc) | 666 Bytes Raw | Blame | History

```
1  title: Rare Service Installs
2  id: 66bfef30-22a5-4fcd-ad44-8d81e60922ae
3  description: Detects rare service installs that only appear a few times per time frame and could reveal password dumpers, backd
4  status: experimental
5  author: Florian Roth
6  date: 2017/03/08
7  tags:
8      - attack.persistence
9      - attack.privilege_escalation
10     - attack.t1050
11     - car.2013-09-005
12 logsource:
13     product: windows
14     service: system
15 detection:
16     selection:
17         EventID: 7045
18     timeframe: 7d
19     condition: selection | count() by ServiceFileName < 5
20 falsepositives:
21     - Software installation
22     - Software updates
23 level: low
```

Detection Spectrum



T1050 - Service Creation	
Tools	

Mitre ATT&CK

- Good starting point
- Contains relevant data sources

ID: T1050

Tactic: Persistence, Privilege Escalation

Platform: Windows

Permissions Required: Administrator, SYSTEM

Effective Permissions: SYSTEM

Data Sources: Windows Registry, Process monitoring, Process command-line parameters, Windows event logs

CAPEC ID: CAPEC-550

Contributors: Pedro Harrison

Version: 1.0

Created: 31 May 2017

Last Modified: 18 July 2019

Mitre ATT&CK

Detection

Monitor service creation through changes in the Registry and common utilities using command-line invocation. Creation of new services may generate an alterable event (ex: Event ID 4697 and/or 7045 [75] [76]). New, benign services may be created during installation of new software. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as network connections made for Command and Control, learning details about the environment through Discovery, and Lateral Movement.

Tools such as Sysinternals Autoruns may also be used to detect system changes that could be attempts at persistence. [77] Look for changes to services that do not correlate with known software, patch cycles, etc. Suspicious program execution through services may show up as outlier processes that have not been seen before when compared against historical data.

Monitor processes and command-line arguments for actions that could create services. Remote access tools with built-in features may interact directly with the Windows API to perform these functions outside of typical system utilities. Services may also be created through Windows system management tools such as Windows Management Instrumentation and PowerShell, so additional logging may need to be configured to gather the appropriate data.

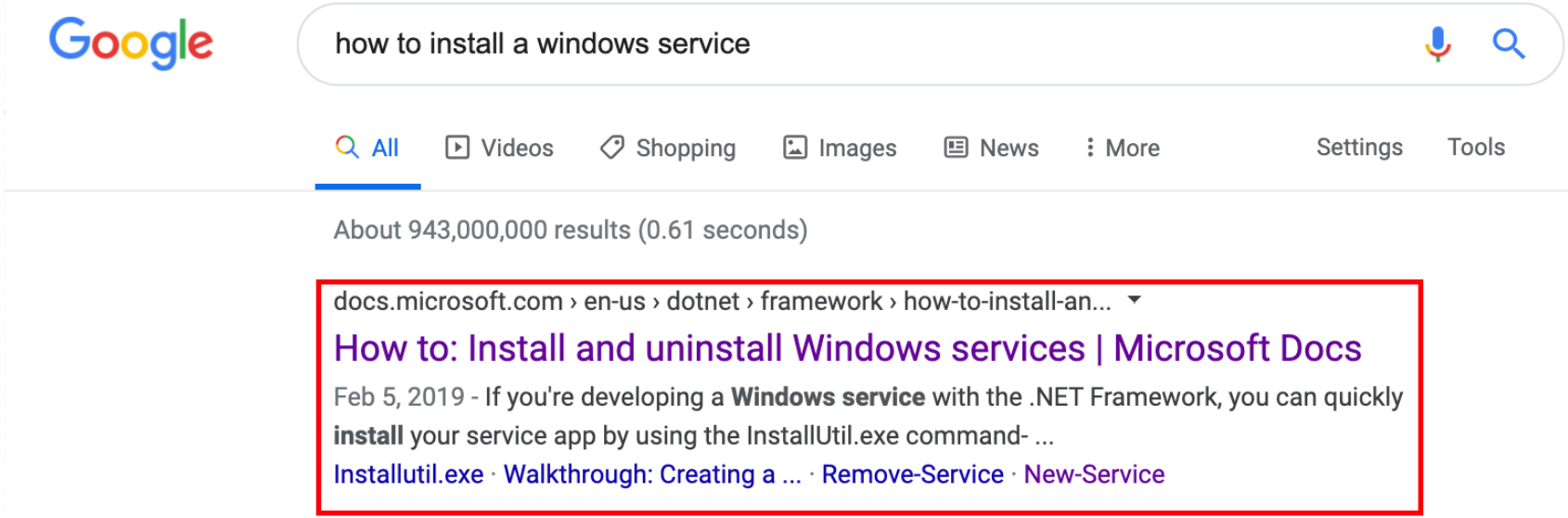
Questions

- What are common utilities to install a service?
- Why would we monitor the registry for service creations?

Tools

What Tool(s) do we know perform this technique?

Tools


Google
how to install a windows service
All
Videos
Shopping
Images
News
More
Settings
Tools
About 943,000,000 results (0.61 seconds)
docs.microsoft.com › en-us › dotnet › framework › how-to-install-an...
How to: Install and uninstall Windows services | Microsoft Docs
Feb 5, 2019 - If you're developing a Windows service with the .NET Framework, you can quickly install your service app by using the InstallUtil.exe command- ...
Installutil.exe · Walkthrough: Creating a ... · Remove-Service · New-Service

Tools

Install your service manually using PowerShell

1. From the **Start** menu, select the **Windows PowerShell** directory, then select **Windows PowerShell**.
2. Access the directory where your project's compiled executable file is located.
3. Run the **New-Service** cmdlet with the with your project's output and a service name as parameters:

PowerShell | Copy

```
New-Service -Name "YourServiceName" -BinaryPathName <yourproject>.exe
```

Tools

T1050 - Service Creation	
Tools	New-Service

Service Creation w/ SC.exe or PowerShell

Tree: 48d95f027c ▾ **sigma** / rules / windows / process_creation / **win_new_service_creation.yml** Find file | Copy path

thomaspatzke Rule fixes 373424f on Feb 20

2 contributors

28 lines (28 sloc) | 846 Bytes Raw | Blame | History

```yaml
title: New Service Creation
id: 7fe71fc9-de3b-432a-8d57-8c809efc10ab
status: experimental
description: Detects creation if a new service
author: Timur Zinniatullin, Daniil Yugoslavskiy, oscd.community
date: 2019/10/21
modified: 2019/11/04
tags:
    - attack.persistence
    - attack.privilege_escalation
    - attack.t1050
references:
    - https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1050/T1050.yaml
logsource:
    category: process_creation
    product: windows
detection:
    selection:
      - Image|endswith: '\sc.exe'
        CommandLine|contains|all:
            - 'create'
            - 'binpath'
      - Image|endswith: '\powershell.exe'
        CommandLine|contains: 'new-service'
    condition: selection
falsepositives:
    - Legitimate administrator or user creates a service for legitimate reason
level: low
```

Questions

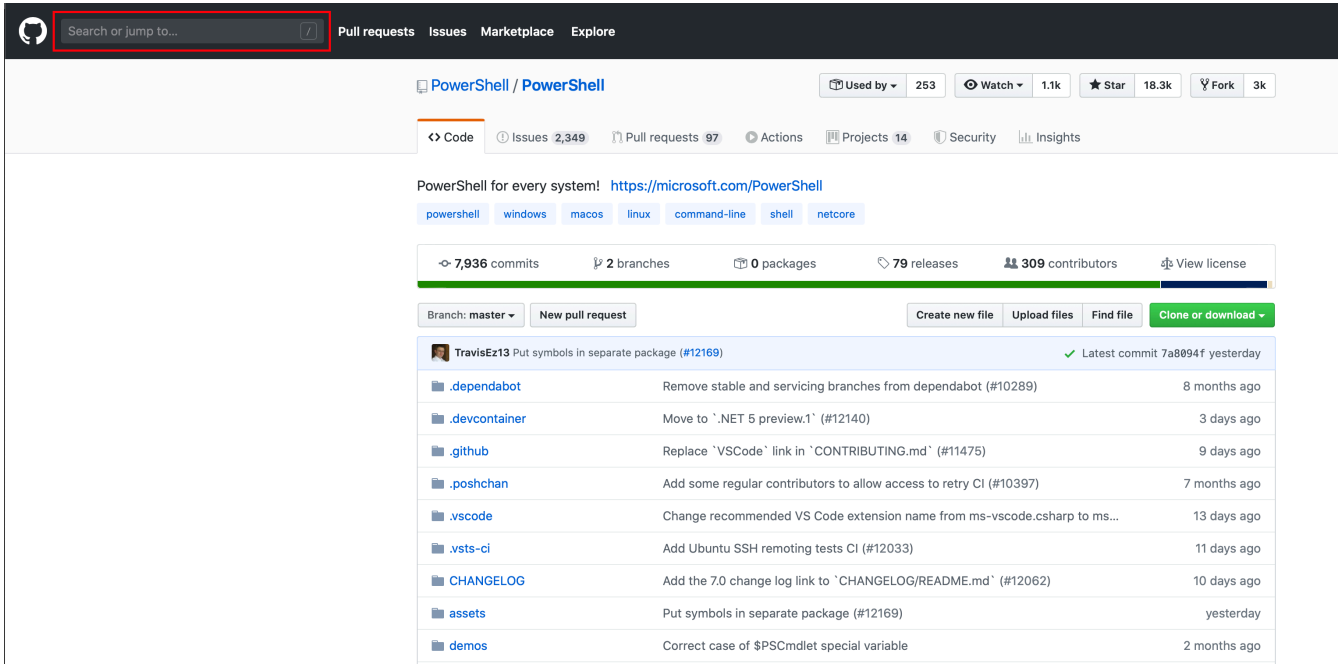
- What are common utilities to install a service?
  - New-Service is one of them, but are there more?
- Why would we monitor the registry for service creations?
  - Unknown

Functions
Static Analysis – Source Code Review

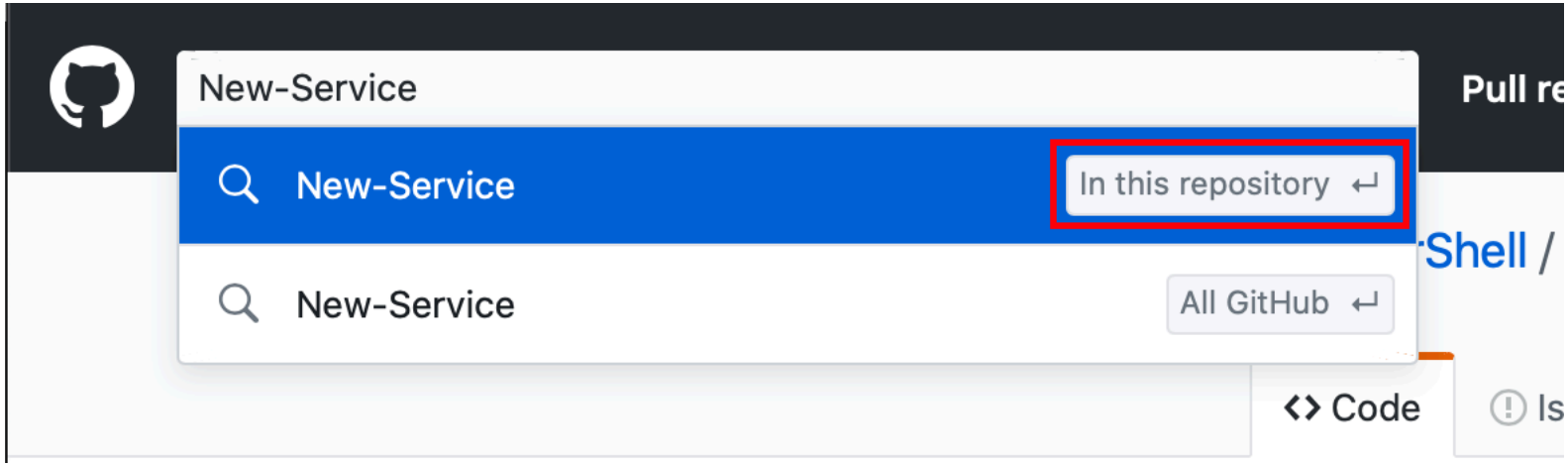
Functions

- Are there managed (ex. .NET) APIs that we should consider?
- What unmanaged function(s) is/are used by the tool(s) that we previously identified?
- What DLL(s) export these functions?
- Are there alternative functions that can be called?
- Are there any native, undocumented, or underlying functions that are relied upon?

Functions


Search or jump to...
Pull requests
Issues
Marketplace
Explore
PowerShell / PowerShell
Used by 253
Watch 1.1k
Star 18.3k
Fork 3k
Code
Issues 2,349
Pull requests 97
Actions
Projects 14
Security
Insights
PowerShell for every system! https://microsoft.com/PowerShell
powershell
windows
macos
linux
command-line
shell
netcore
7,936 commits
2 branches
0 packages
79 releases
309 contributors
View license
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
TravisEz13 Put symbols in separate package (#12169)
Latest commit 7a8094f yesterday
.dependabot Remove stable and servicing branches from dependabot (#10289) 8 months ago
.devcontainer Move to `.NET 5 preview.1` (#12140) 3 days ago
.github Replace `VSCode` link in `CONTRIBUTING.md` (#11475) 9 days ago
.poshchan Add some regular contributors to allow access to retry CI (#10397) 7 months ago
.vscode Change recommended VS Code extension name from ms-vscode.csharp to ms... 13 days ago
.vsts-ci Add Ubuntu SSH remoting tests CI (#12033) 11 days ago
CHANGELOG Add the 7.0 change log link to `CHANGELOG/README.md` (#12062) 10 days ago
assets Put symbols in separate package (#12169) yesterday
demos Correct case of $PSCmdlet special variable 2 months ago

Functions


New-Service
New-Service
In this repository
New-Service
All GitHub
Pull re
Shell /
Code

Functions

src/Microsoft.PowerShell.Commands.Management/commands/management/Service.cs

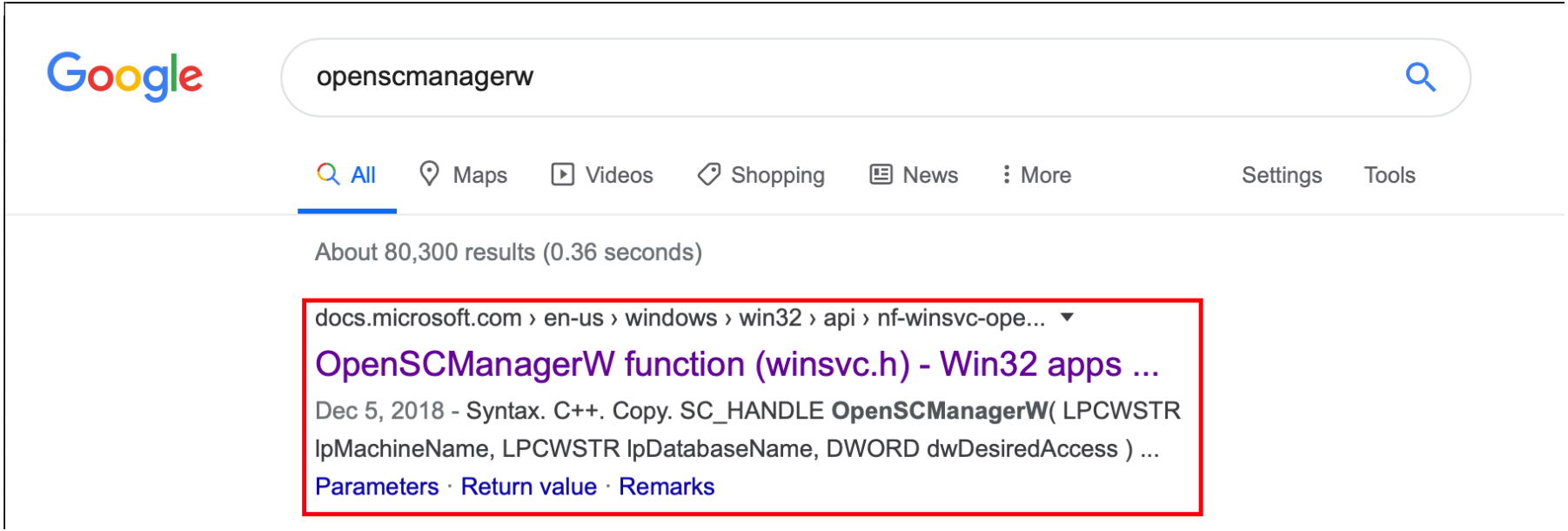
```
1994        /// <summary>
1995        /// This class implements the New-Service command.
1996        /// </summary>
1997        [Cmdlet(VerbsCommon.New, "Service", SupportsShouldProcess = true, HelpUri =
    "https://go.microsoft.com/fwlink/?LinkID=2096905")]
    …
2167                        WriteNonTerminatingError(StartupType.ToString(), "New-Service",
    Name,
2168                            new ArgumentException(), "CouldNotNewService",
```

C# Showing the top two matches Last indexed on Feb 18

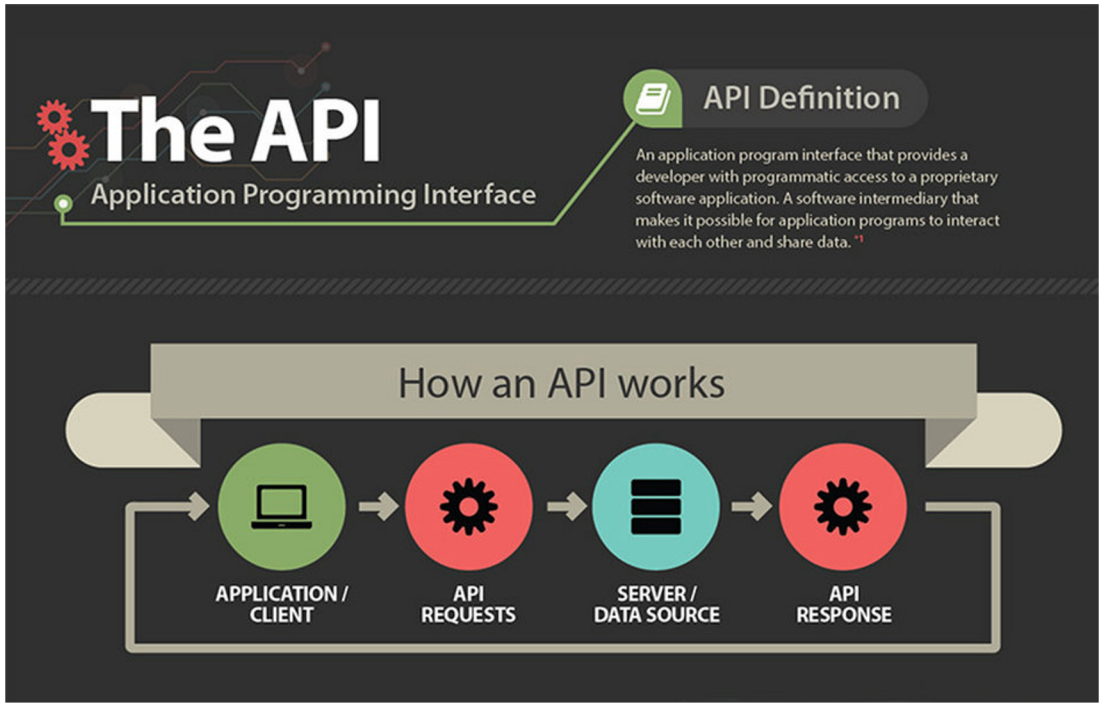
Functions

```
2120        /// Create the service.
2121        /// </summary>
2122        [ArchitectureSensitive]
2123        protected override void BeginProcessing()
2124        {
2125            ServiceController service = null;
2126            Diagnostics.Assert(!string.IsNullOrEmpty(Name),
2127                "null ServiceName");
2128            Diagnostics.Assert(!string.IsNullOrEmpty(BinaryPathName),
2129                "null BinaryPathName");
2130
2131            // confirm the operation first
2132            // this is always false if WhatIf is set
2133            if (!ShouldProcessServiceOperation(DisplayName ?? string.Empty, Name))
2134            {
2135                return;
2136            }
2137
2138            // Connect to the service controller
2139            NakedWin32Handle hScManager = IntPtr.Zero;
2140            NakedWin32Handle hService = IntPtr.Zero;
2141            IntPtr password = IntPtr.Zero;
2142            IntPtr delayedAutoStartInfoBuffer = IntPtr.Zero;
2143            try
2144            {
2145                hScManager = NativeMethods.OpenSCManagerW(
2146                    null,
2147                    null,
2148                    NativeMethods.SC_MANAGER_CONNECT | NativeMethods.SC_MANAGER_CREATE_SERVICE
2149                    );
```

Functions


Google
openscmanagerw
All
Maps
Videos
Shopping
News
More
Settings
Tools
About 80,300 results (0.36 seconds)
docs.microsoft.com › en-us › windows › win32 › api › nf-winsvc-ope...
OpenSCManagerW function (winsvc.h) - Win32 apps ...
Dec 5, 2018 - Syntax. C++. Copy. SC_HANDLE OpenSCManagerW(LPCWSTR lpMachineName, LPCWSTR lpDatabaseName, DWORD dwDesiredAccess) ...
Parameters · Return value · Remarks

Application Programming Interface (API)



Source: Sleter (2014), What's an API and Why Do You Need One?,https://www.govtech.com/applications/Whats-an-API-and-Why-Do-You-Need-One.html

Windows API

- User-System programing interface to the Windows OS family
  - Formerly Win32 API for 32bit Windows
  - Originally consisted of C-style functions
- Newer API development leverages Component Object Model (COM)
  - COM consists of two fundamental principles
    - Clients communicate with objects
      - COM server objects
    - Component implementation is loaded dynamically
- The Windows API index is extensive and acts as reference content for server and desktop applications

Source: Windows API (2018), https://docs.microsoft.com/en-us/windows/desktop/apiindex/windows-api-list

Windows API

- Windows API Sets
  - API contracts providing separation between API contract and host (DLL) implementation
    - Subsets of Win API
    - Reduces the number of loaded DLLs in a processes
  - API sets rely on the OS support in the loader library
    - Introduces namespace redirection in the library binding process
  - Depending on the set name (import context), the library loader will perform runtime redirection to call the correct host binary that contains the correct implementation of the API Set

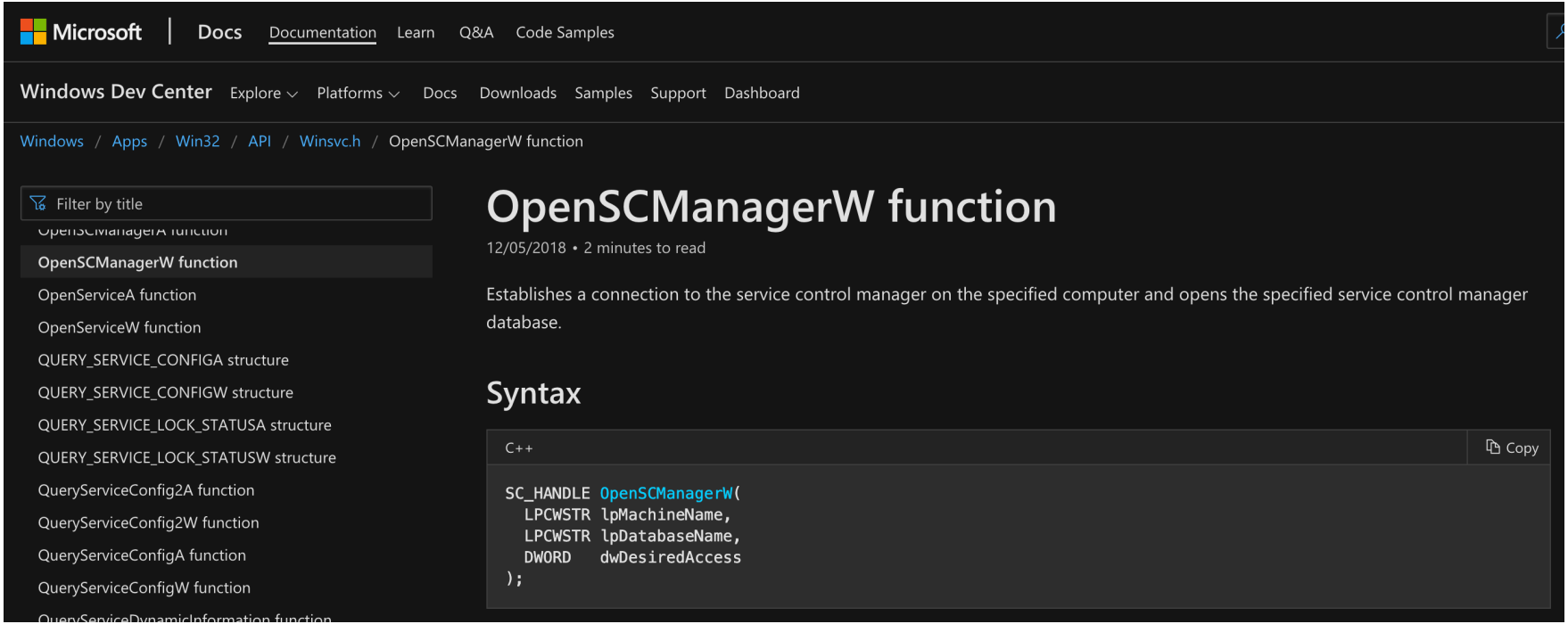
Source: Windows API (2018), https://docs.microsoft.com/en-us/windows/desktop/apiindex/windows-api-list

OpenSCManagerA vs. OpenSCManagerW

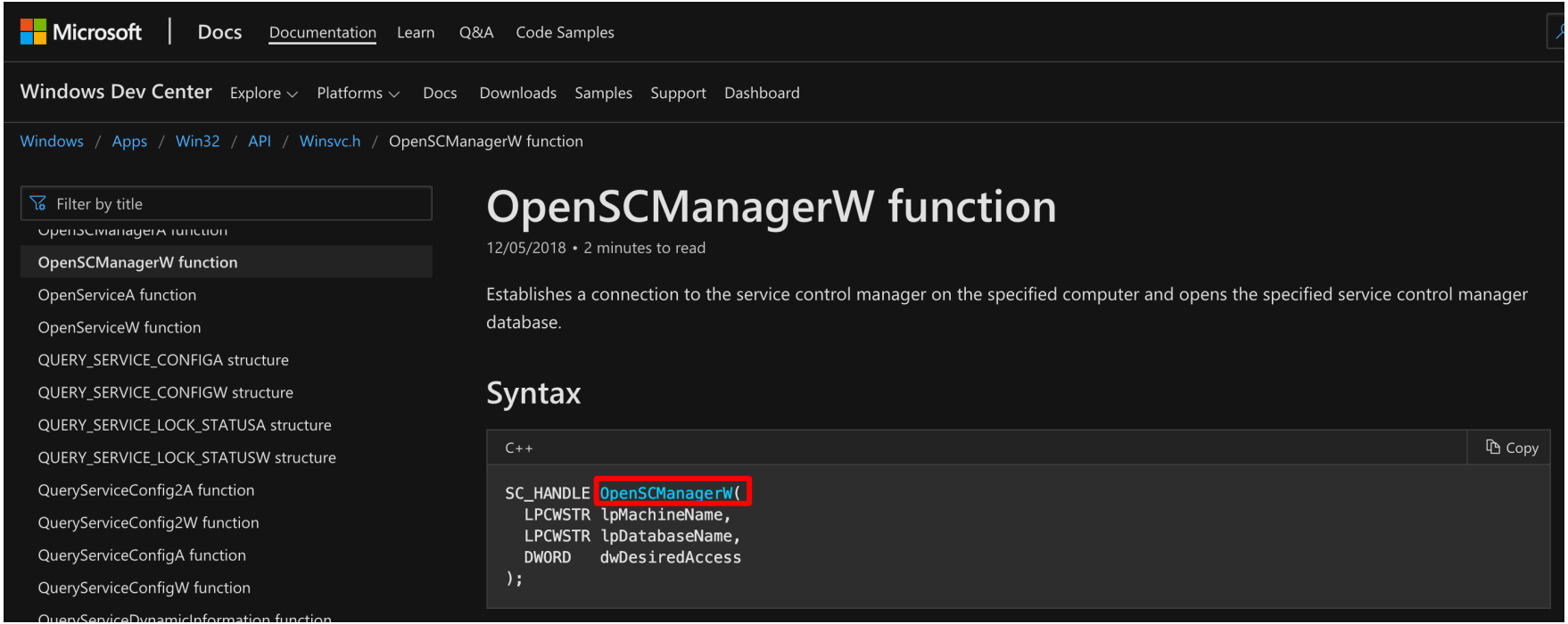
- Functions that end in A use ASCII strings
- Functions that end in W use Wide (Unicode) strings

- When Windows was initially created, they used ASCII character encoding
- With its international popularity Windows required a broader encoding scheme
- On modern versions of Windows, any API that takes a string argument has both an ASCII and Wide (Unicode) version

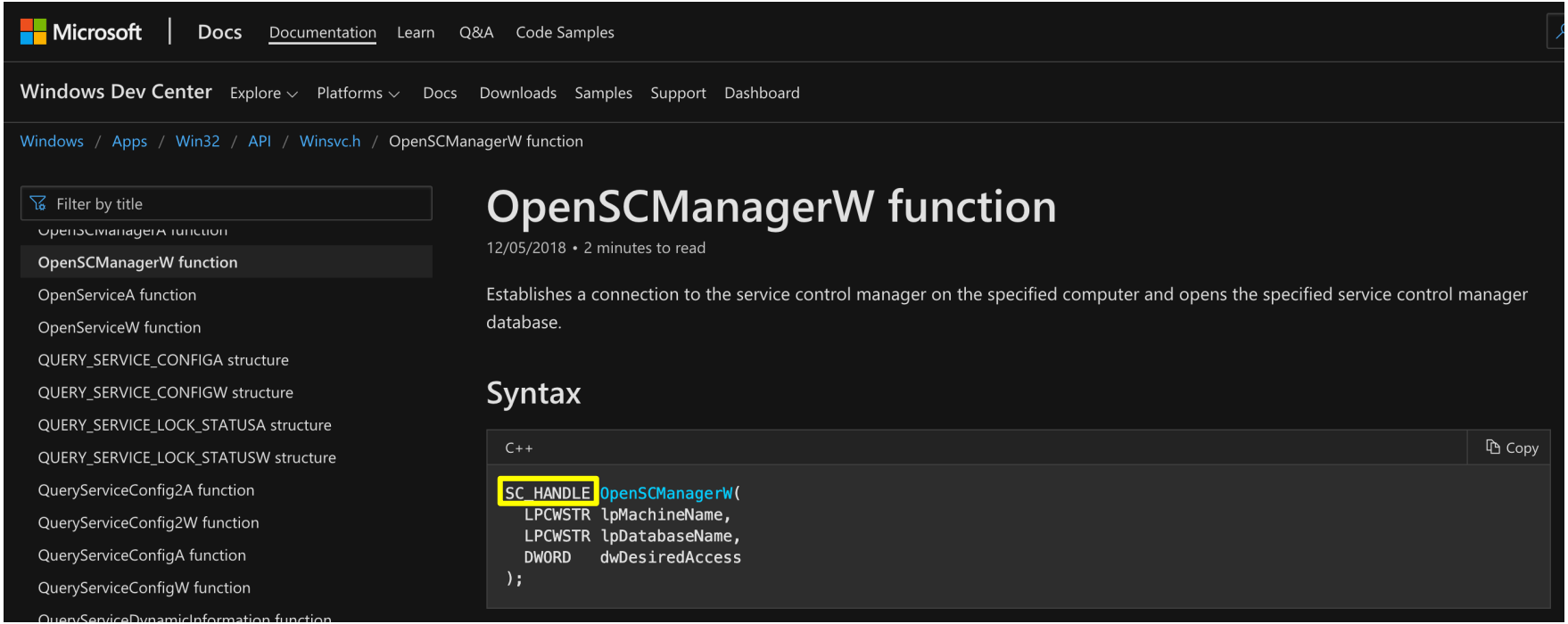
Function Overview



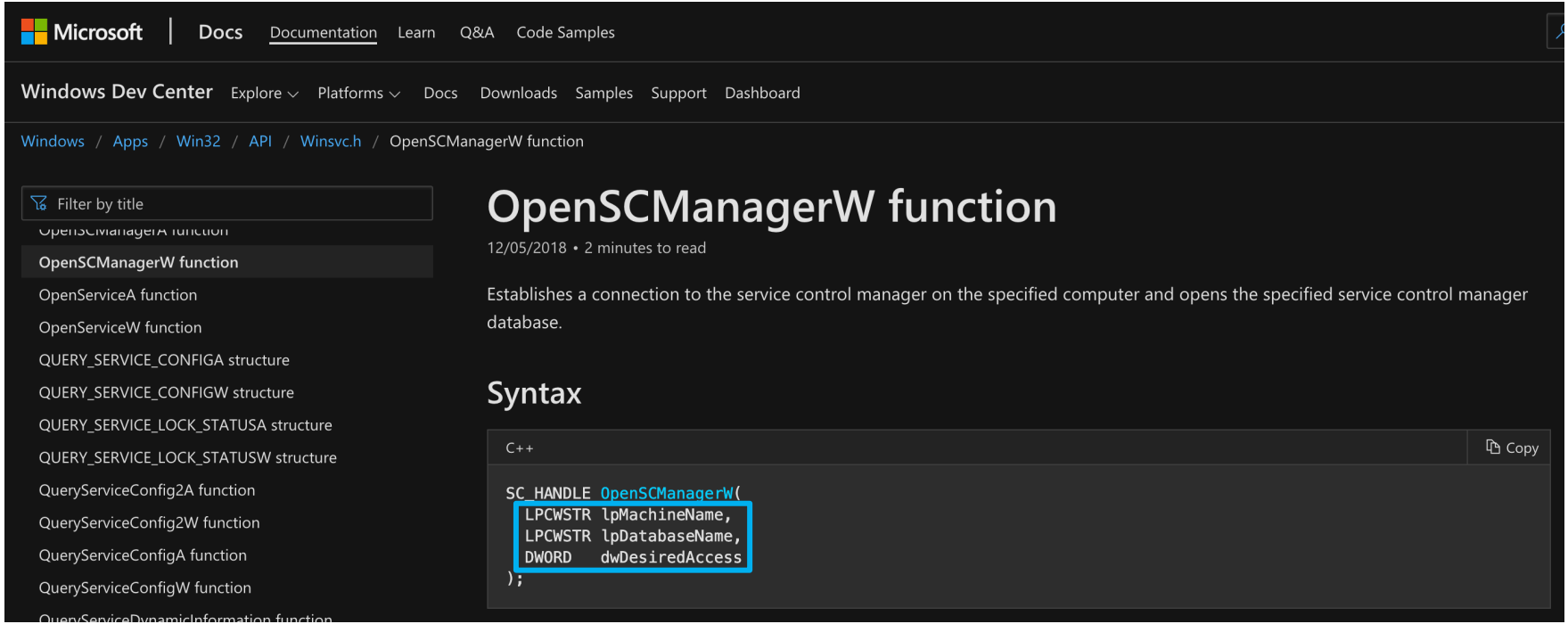
Function Overview – Function Name



Function Overview – Output Type



Function Overview – Parameters



Parameters

Parameters

`lpMachineName`

The name of the target computer. If the pointer is NULL or points to an empty string, the function connects to the service control manager on the local computer.

`lpDatabaseName`

The name of the service control manager database. This parameter should be set to SERVICES_ACTIVE_DATABASE. If it is NULL, the SERVICES_ACTIVE_DATABASE database is opened by default.

`dwDesiredAccess`

The access to the service control manager. For a list of access rights, see Service Security and Access Rights.

Before granting the requested access rights, the system checks the access token of the calling process against the discretionary access-control list of the security descriptor associated with the service control manager.

The SC_MANAGER_CONNECT access right is implicitly specified by calling this function.

https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-openscmanagerw#parameters

Parameters

Parameters

`lpMachineName`

The name of the target computer. If the pointer is NULL or points to an empty string, the function connects to the service control manager on the local computer.

`lpDatabaseName`

The name of the service control manager database. This parameter should be set to SERVICES_ACTIVE_DATABASE. If it is NULL, the SERVICES_ACTIVE_DATABASE database is opened by default.

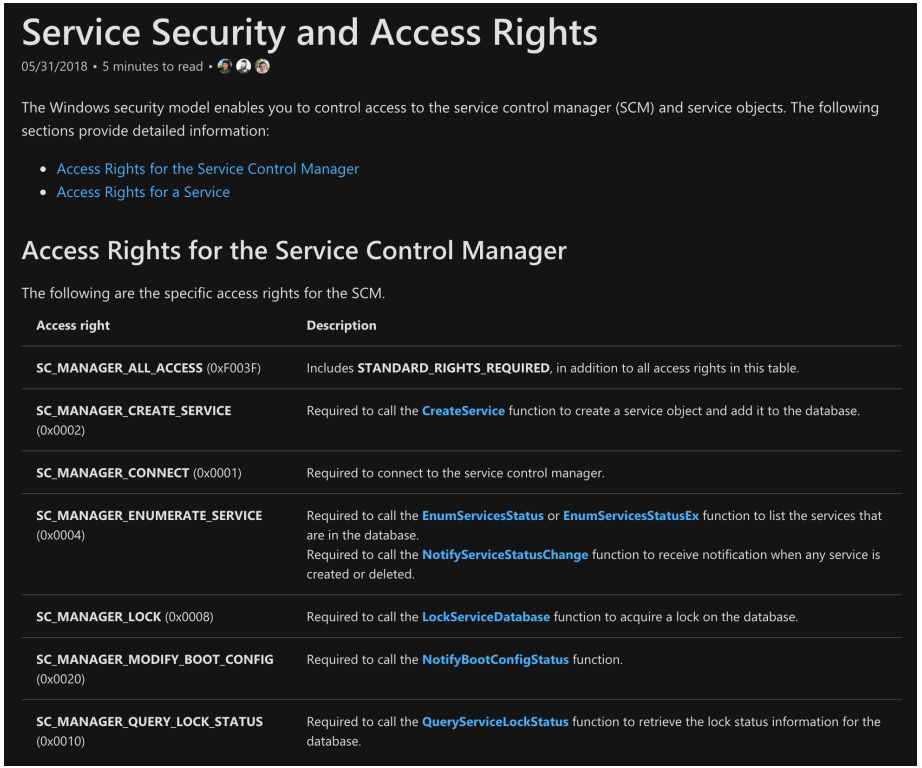
`dwDesiredAccess`

The access to the service control manager. For a list of access rights, see Service Security and Access Rights.

Before granting the requested access rights, the system checks the access token of the calling process against the discretionary access-control list of the security descriptor associated with the service control manager.

The SC_MANAGER_CONNECT access right is implicitly specified by calling this function.

Service Security and Access Rights



Service Security and Access Rights

05/31/2018 • 5 minutes to read •

The Windows security model enables you to control access to the service control manager (SCM) and service objects. The following sections provide detailed information:

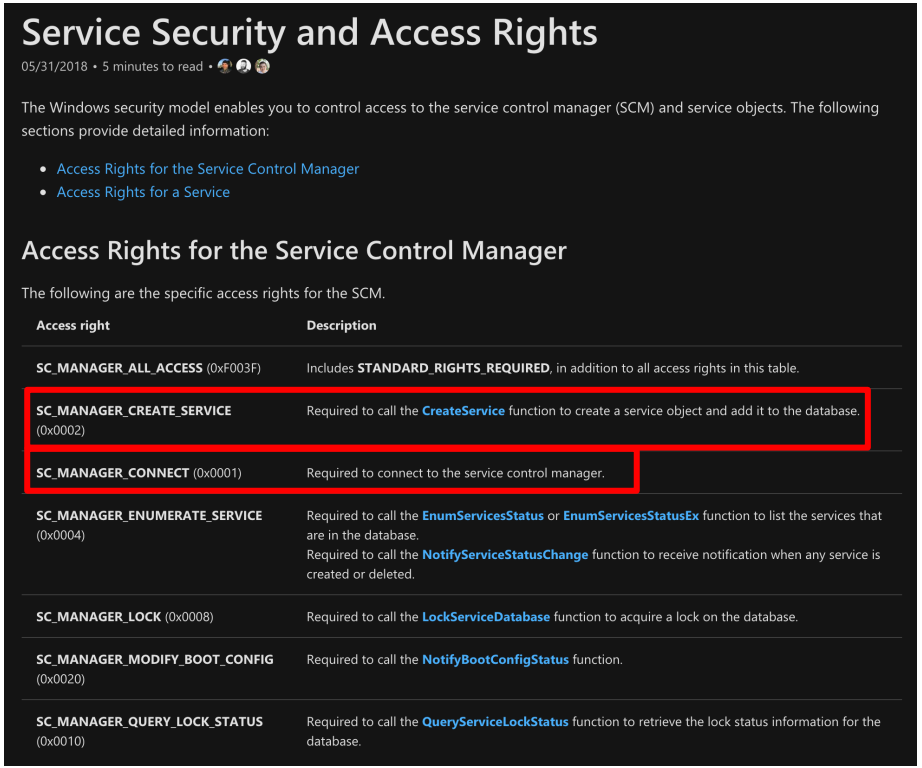
- Access Rights for the Service Control Manager
- Access Rights for a Service

Access Rights for the Service Control Manager

The following are the specific access rights for the SCM.

Access right	Description
SC_MANAGER_ALL_ACCESS (0xF003F)	Includes **STANDARD_RIGHTS_REQUIRED**, in addition to all access rights in this table.
SC_MANAGER_CREATE_SERVICE (0x0002)	Required to call the **CreateService** function to create a service object and add it to the database.
SC_MANAGER_CONNECT (0x0001)	Required to connect to the service control manager.
SC_MANAGER_ENUMERATE_SERVICE (0x0004)	Required to call the **EnumServicesStatus** or **EnumServicesStatusEx** function to list the services that are in the database. Required to call the **NotifyServiceStatusChange** function to receive notification when any service is created or deleted.
SC_MANAGER_LOCK (0x0008)	Required to call the **LockServiceDatabase** function to acquire a lock on the database.
SC_MANAGER_MODIFY_BOOT_CONFIG (0x0020)	Required to call the **NotifyBootConfigStatus** function.
SC_MANAGER_QUERY_LOCK_STATUS (0x0010)	Required to call the **QueryServiceLockStatus** function to retrieve the lock status information for the database.

Service Security and Access Rights



Service Security and Access Rights

05/31/2018 • 5 minutes to read •

The Windows security model enables you to control access to the service control manager (SCM) and service objects. The following sections provide detailed information:

- Access Rights for the Service Control Manager
- Access Rights for a Service

Access Rights for the Service Control Manager

The following are the specific access rights for the SCM.

Access right	Description
SC_MANAGER_ALL_ACCESS (0xF003F)	Includes **STANDARD_RIGHTS_REQUIRED**, in addition to all access rights in this table.
SC_MANAGER_CREATE_SERVICE (0x0002)	Required to call the **CreateService** function to create a service object and add it to the database.
SC_MANAGER_CONNECT (0x0001)	Required to connect to the service control manager.
SC_MANAGER_ENUMERATE_SERVICE (0x0004)	Required to call the **EnumServicesStatus** or **EnumServicesStatusEx** function to list the services that are in the database. Required to call the **NotifyServiceStatusChange** function to receive notification when any service is created or deleted.
SC_MANAGER_LOCK (0x0008)	Required to call the **LockServiceDatabase** function to acquire a lock on the database.
SC_MANAGER_MODIFY_BOOT_CONFIG (0x0020)	Required to call the **NotifyBootConfigStatus** function.
SC_MANAGER_QUERY_LOCK_STATUS (0x0010)	Required to call the **QueryServiceLockStatus** function to retrieve the lock status information for the database.

Return Value

Return value

If the function succeeds, the return value is a handle to the specified service control manager database.

If the function fails, the return value is NULL. To get extended error information, call GetLastError.

The following error codes can be set by the SCM. Other error codes can be set by the registry functions that are called by the SCM.

Return code	Description
ERROR_ACCESS_DENIED	The requested access was denied.
ERROR_DATABASE_DOES_NOT_EXIST	The specified database does not exist.

https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-openscmanagerw#return-value

Remarks

Remarks

When a process uses the **OpenSCManager** function to open a handle to a service control manager database, the system performs a security check before granting the requested access. For more information, see Service Security and Access Rights.

If the current user does not have proper access when connecting to a service on another computer, the **OpenSCManager** function call fails. To connect to a service remotely, call the LogonUser function with LOGON32_LOGON_NEW_CREDENTIALS and then call ImpersonateLoggedOnUser before calling **OpenSCManager**. For more information about connecting to services remotely, see Services and RPC/TCP.

Only processes with Administrator privileges are able to open a database handle that can be used by the CreateService function.

The returned handle is only valid for the process that called the **OpenSCManager** function. It can be closed by calling the CloseServiceHandle function.

Examples

For an example, see Changing a Service's Configuration.

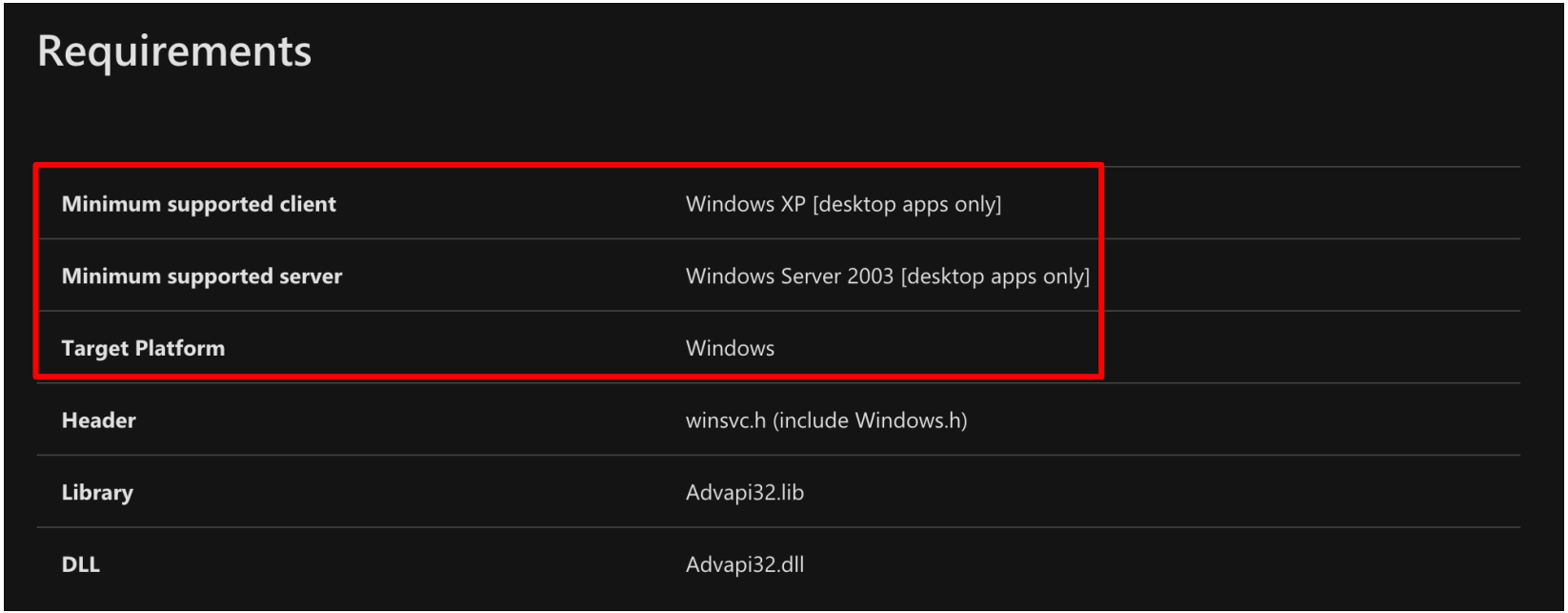
Requirements

Requirements

Minimum supported client	Windows XP [desktop apps only]
Minimum supported server	Windows Server 2003 [desktop apps only]
Target Platform	Windows
Header	winsvc.h (include Windows.h)
Library	Advapi32.lib
DLL	Advapi32.dll

https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-openscmanagerw#requirements

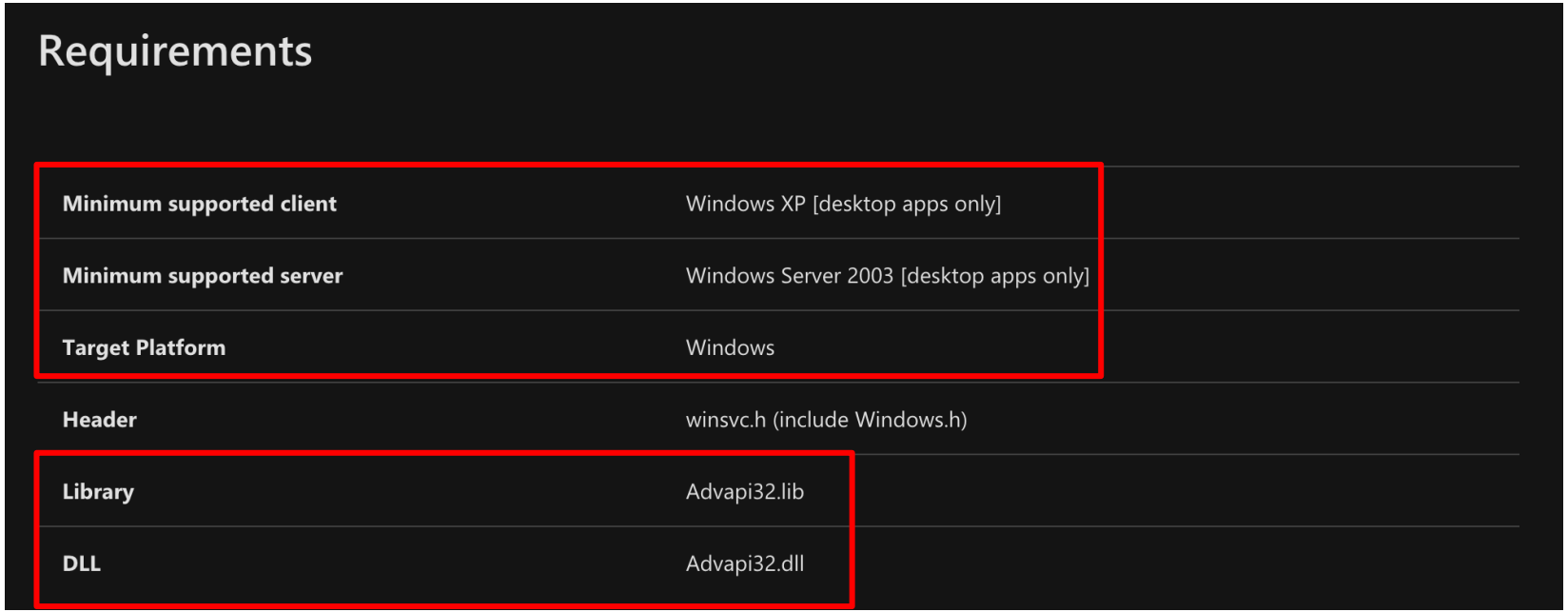
Requirements



Requirements

Minimum supported client	Windows XP [desktop apps only]
Minimum supported server	Windows Server 2003 [desktop apps only]
Target Platform	Windows
Header	winsvc.h (include Windows.h)
Library	Advapi32.lib
DLL	Advapi32.dll

Requirements



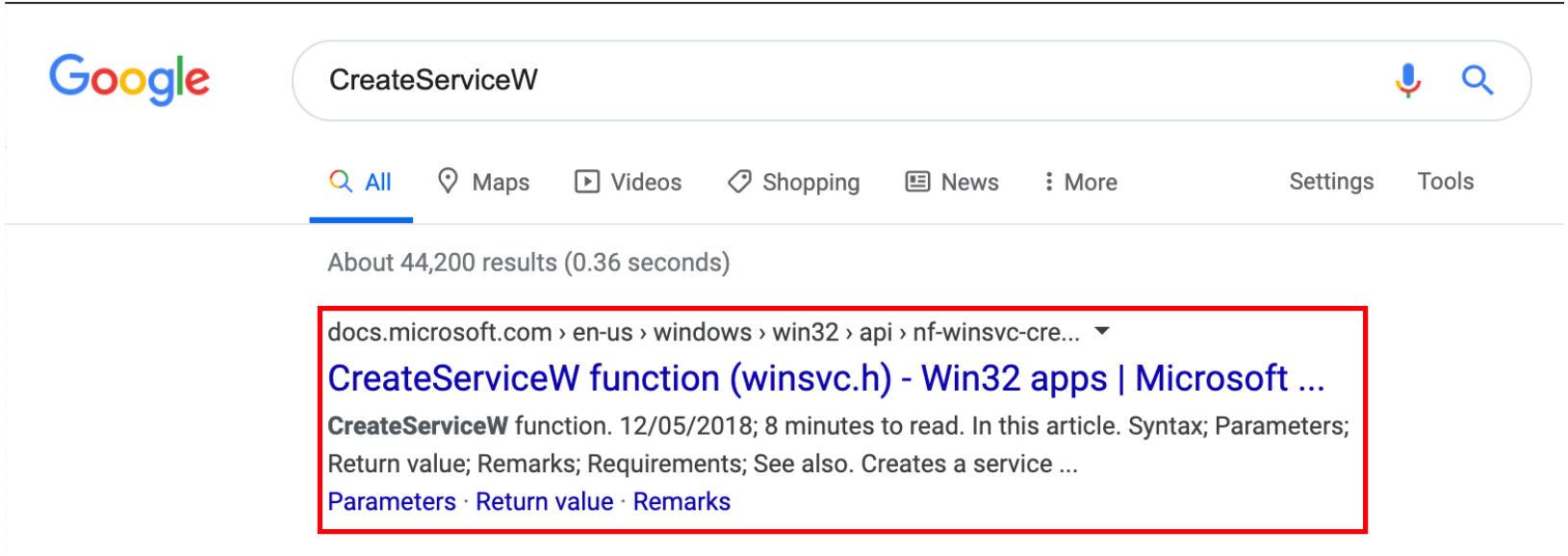
Requirements

Minimum supported client	Windows XP [desktop apps only]
Minimum supported server	Windows Server 2003 [desktop apps only]
Target Platform	Windows
Header	winsvc.h (include Windows.h)
Library	Advapi32.lib
DLL	Advapi32.dll

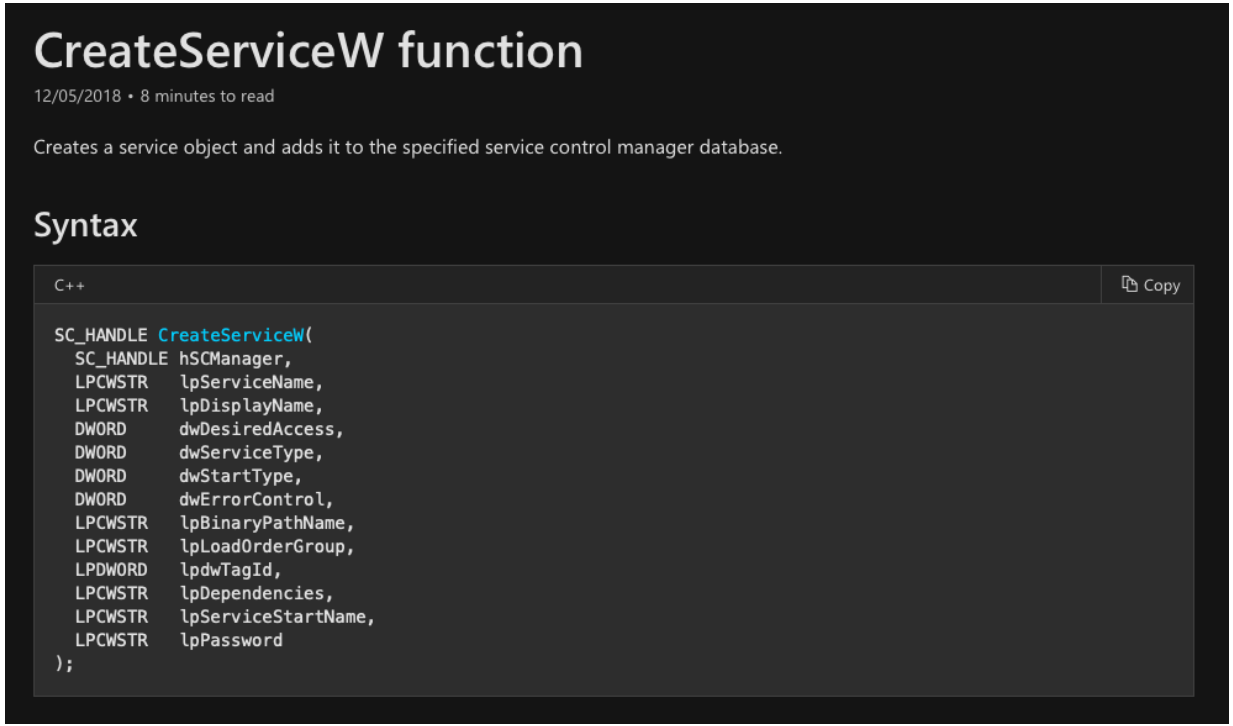
Functions

```
2211            // Create the service
2212            hService = NativeMethods.CreateServiceW(
2213                hScManager,
2214                Name,
2215                DisplayName,
2216                NativeMethods.SERVICE_CHANGE_CONFIG | NativeMethods.WRITE_DAC | NativeMethods.WRITE_OWNER,
2217                NativeMethods.SERVICE_WIN32_OWN_PROCESS,
2218                dwStartType,
2219                NativeMethods.SERVICE_ERROR_NORMAL,
2220                BinaryPathName,
2221                null,
2222                null,
2223                lpDependencies,
2224                username,
2225                password
2226                );
```

Functions


Google
CreateServiceW
All
Maps
Videos
Shopping
News
More
Settings
Tools
About 44,200 results (0.36 seconds)
docs.microsoft.com › en-us › windows › win32 › api › nf-winsvc-cre...
CreateServiceW function (winsvc.h) - Win32 apps | Microsoft ...
CreateServiceW function. 12/05/2018; 8 minutes to read. In this article. Syntax; Parameters; Return value; Remarks; Requirements; See also. Creates a service ...
Parameters · Return value · Remarks

Functions



CreateServiceW function

12/05/2018 • 8 minutes to read

Creates a service object and adds it to the specified service control manager database.

Syntax

C++ Copy

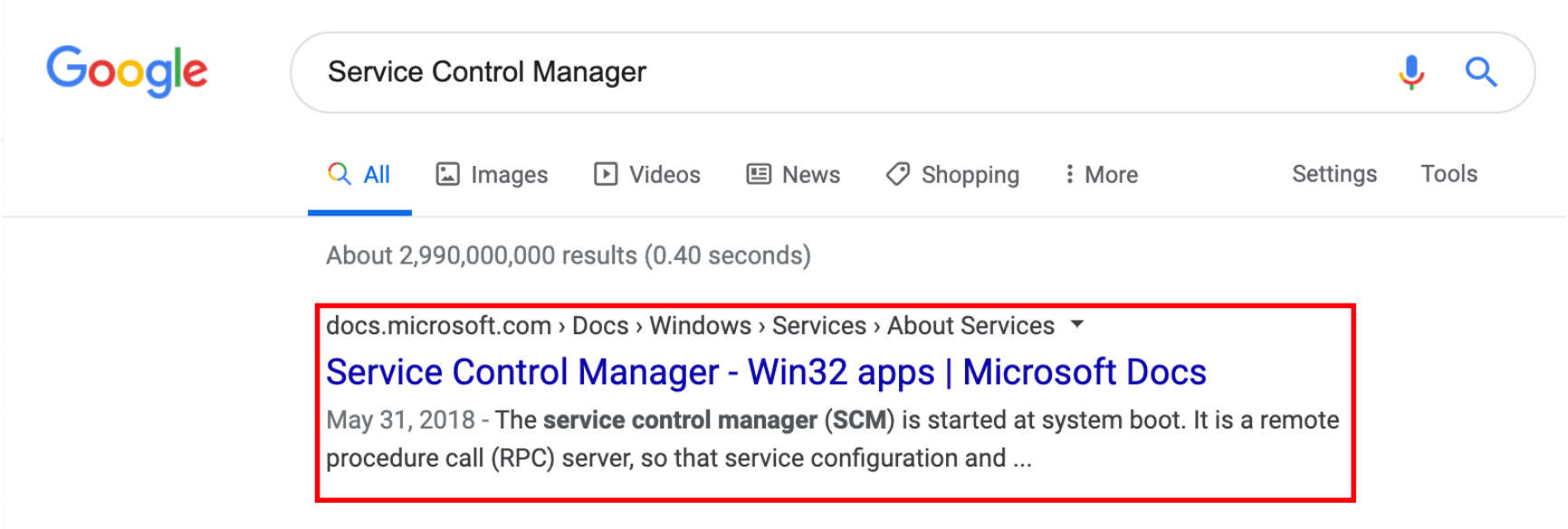
```
SC_HANDLE CreateServiceW(
  SC_HANDLE hSCManager,
  LPCWSTR   lpServiceName,
  LPCWSTR   lpDisplayName,
  DWORD     dwDesiredAccess,
  DWORD     dwServiceType,
  DWORD     dwStartType,
  DWORD     dwErrorControl,
  LPCWSTR   lpBinaryPathName,
  LPCWSTR   lpLoadOrderGroup,
  LPDWORD   lpdwTagId,
  LPCWSTR   lpDependencies,
  LPCWSTR   lpServiceStartName,
  LPCWSTR   lpPassword
);
```

T1050 - Service Creation	
Tools	New-Service
Windows API	OpenSCManager/CreateService

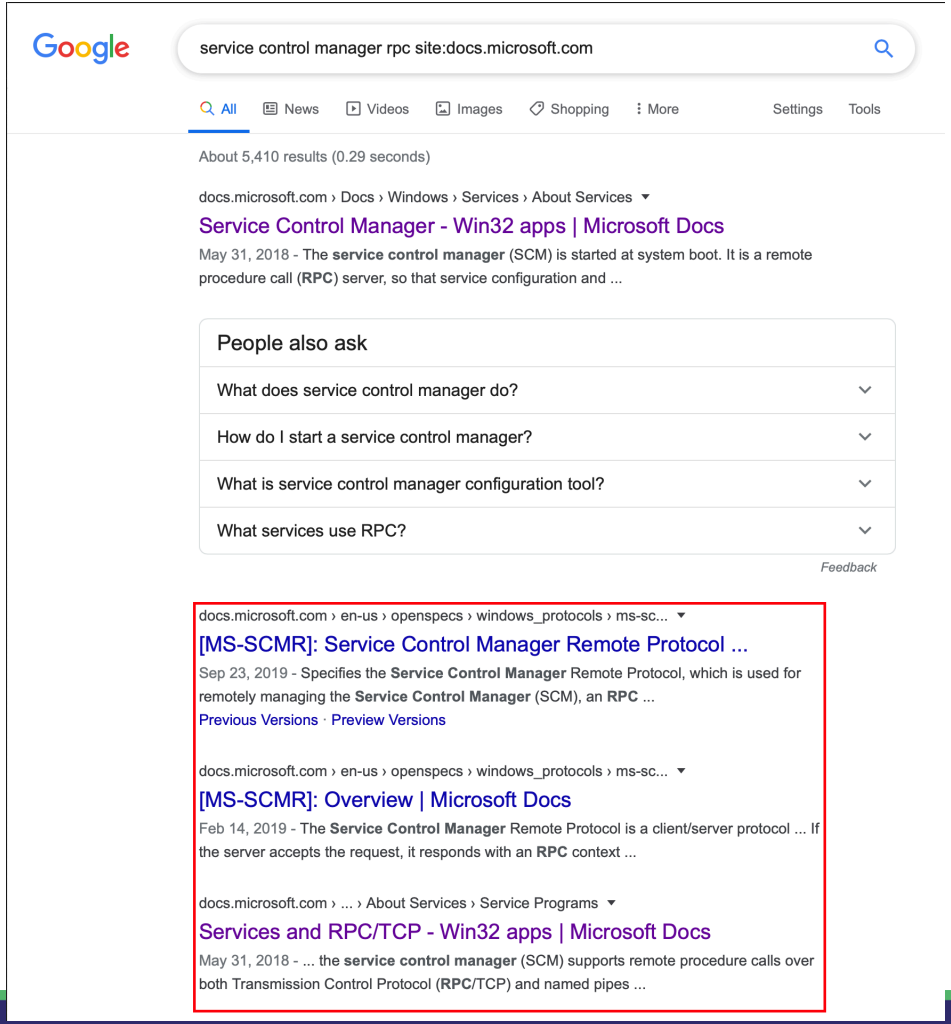
Questions

- What are common utilities to install a service?
  - New-Service is one of them, but are there more?
- Why would we monitor the registry for service creations?
  - Unknown
- What is the Service Control Manager?
  - Unknown
- What is the Service Control Database?
  - Unknown

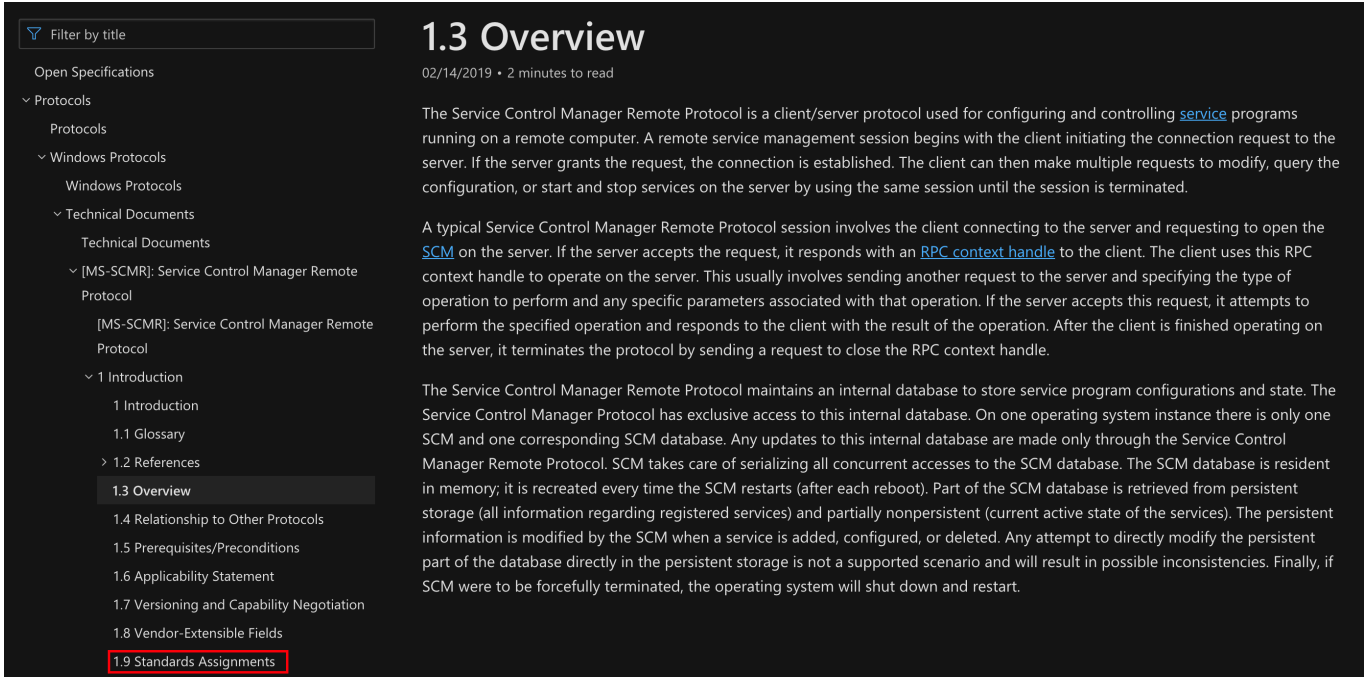
Documentation



Documentation



Documentation



Documentation

1.9 Standards Assignments

02/14/2019 • 2 minutes to read

The Service Control Manager Remote Protocol has no standards assignments, only private assignments made by Microsoft using allocation procedures specified in other protocols.

Microsoft has allocated to this protocol an RPC interface universally unique identifier (UUID) (using the procedure specified in [C706]) and a named pipe (as specified in [MS-SMB]). The assignments are as follows.

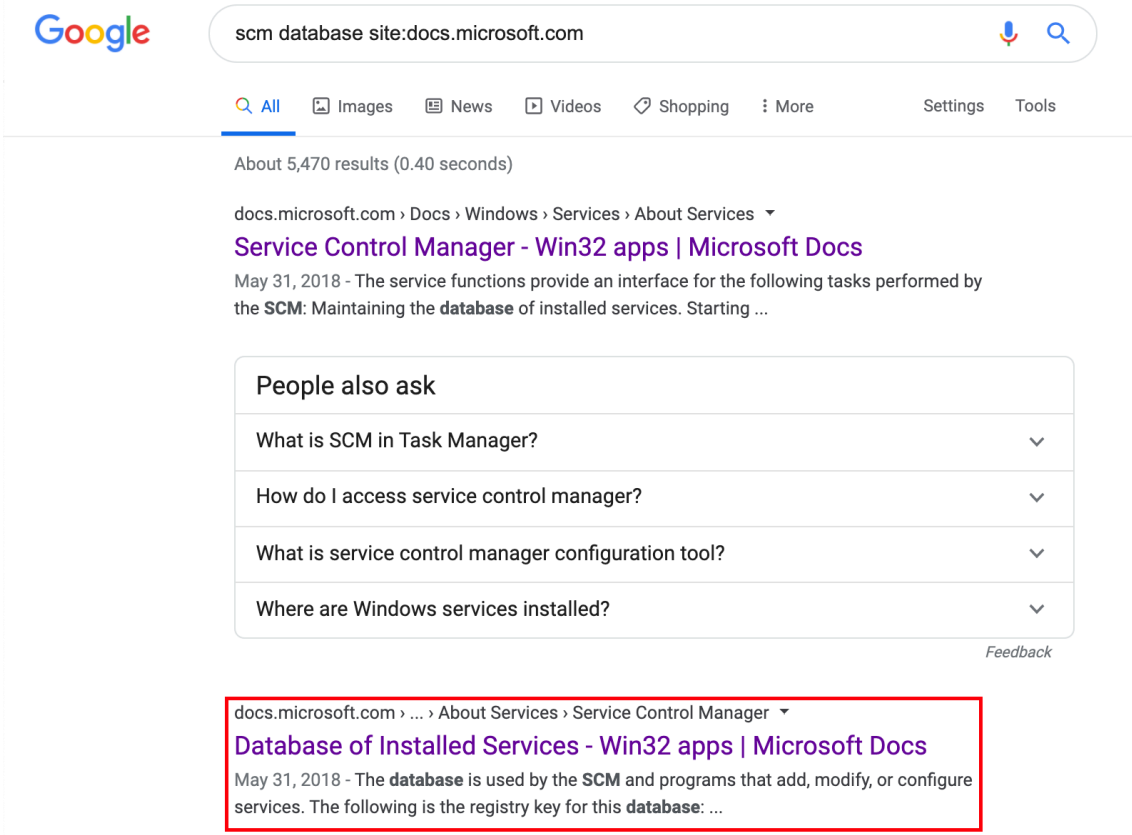
Parameter	Value
RPC interface UUID	{367ABB81-9844-35F1-AD32-98F038001003}
Named pipe	\PIPE\svcctl

T1050 - Service Creation	
Tools	New-Service
Windows API	OpenSCManager/CreateService
RPC Interface/Named Pipe	367ABB81-9844-35F1-AD32-98F038001003
	\\PIPE\svcctl

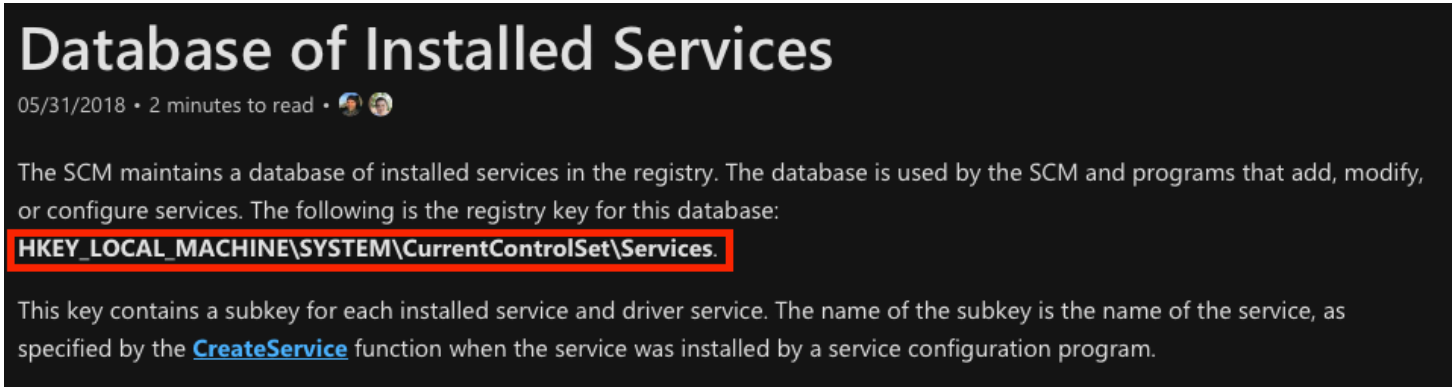
Questions

- What are common utilities to install a service?
  - New-Service is one of them, but are there more?
- Why would we monitor the registry for service creations?
  - Unknown
- What is the Service Control Manager?
  - The SCM is a service that supports RPC/TCP and RPC/NP for remote service management.
- What is the Service Control Database?
  - Unknown

Documentation



Documentation



Database of Installed Services

05/31/2018 • 2 minutes to read •

The SCM maintains a database of installed services in the registry. The database is used by the SCM and programs that add, modify, or configure services. The following is the registry key for this database:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services.

This key contains a subkey for each installed service and driver service. The name of the subkey is the name of the service, as specified by the **CreateService** function when the service was installed by a service configuration program.

T1050 - Service Creation	
Tools	New-Service
Windows API	OpenSCManager/CreateService
RPC Interface/Named Pipe	367ABB81-9844-35F1-AD32-98F038001003 [MS-SCMR]
	\\PIPE\svcctl
Registry Service Database	HKLM\SYSTEM\CurrentControlSet\Services

Questions

- What are common utilities to install a service?
  - New-Service is one of them, but are there more?
- Why would we monitor the registry for service creations?
  - The SCM maintains a database of installed services in the registry.
- What is the Service Control Manager?
  - The SCM is a service that supports RPC/TCP and RPC/NP for remote service management.
- What is the Service Control Database?
  - The SC Database is a database of installed services that resides in the registry.

Network

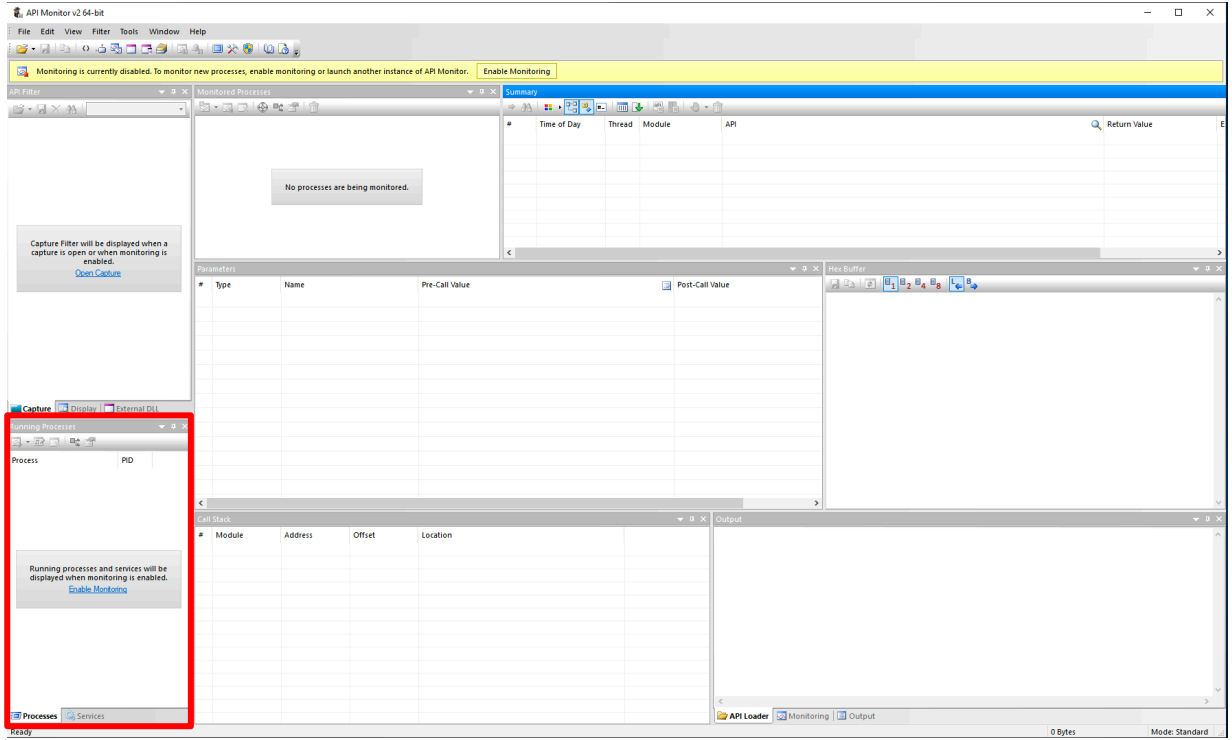
- Can this technique be performed over the network?
- Does this technique require network connectivity?
- What port, protocol, etc. is used?
- What specific details about the protocol can be used to differentiate this activity from other possibly benign activity?

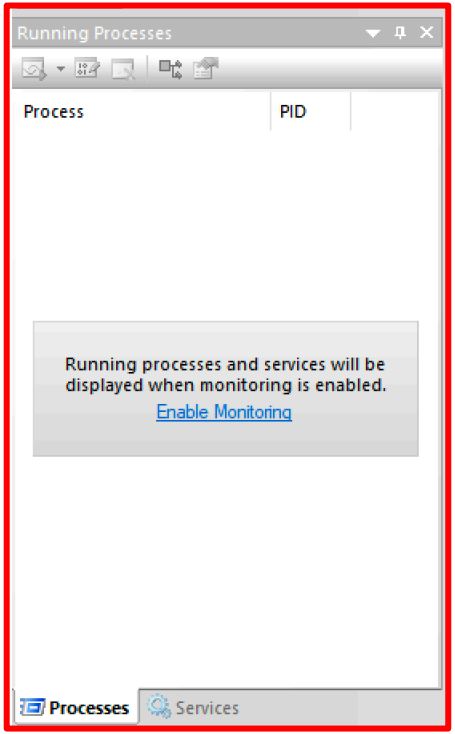
Verification – Walking Through the Map

T1050 - Service Creation	
Tools	New-Service
Windows API	OpenSCManager/CreateService
RPC Interface/Named Pipe	367ABB81-9844-35F1-AD32-98F038001003 [MS-SCMR]
	\\PIPE\svcctl
Registry Service Database	HKLM\SYSTEM\CurrentControlSet\Services

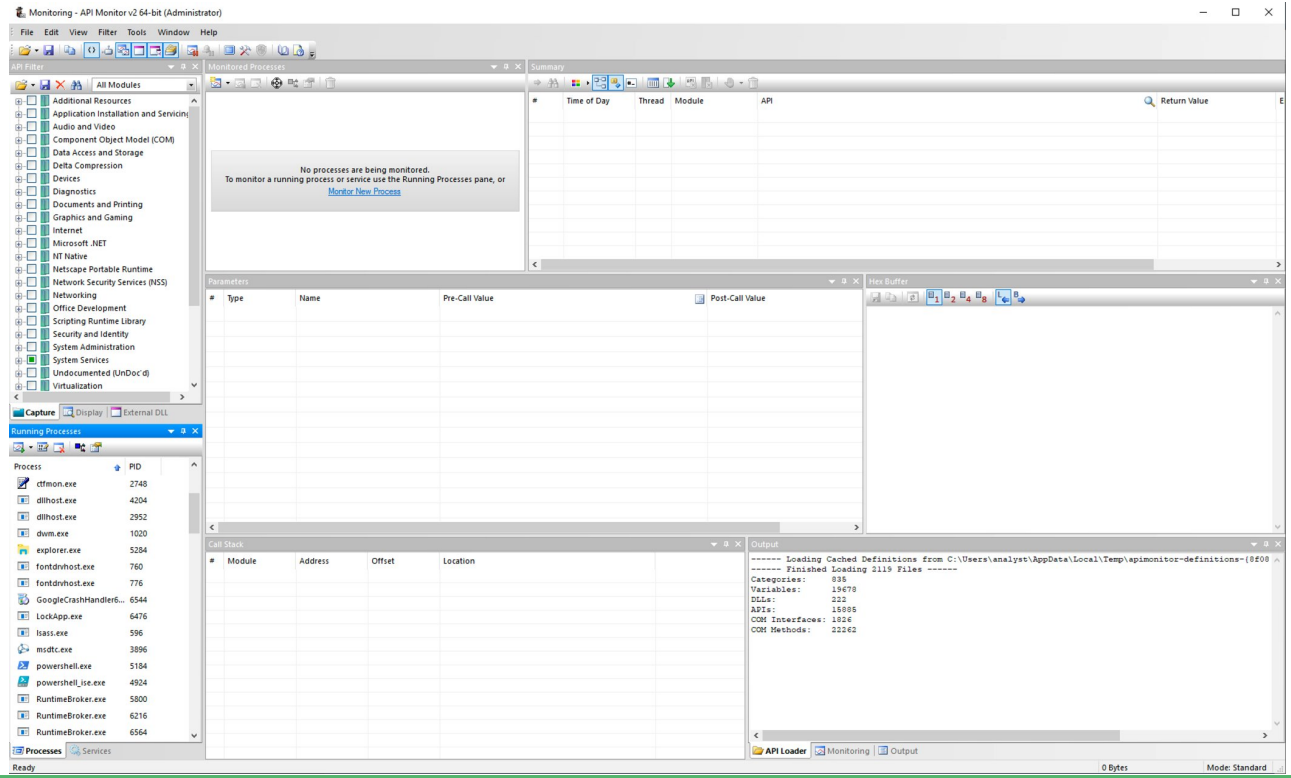
Functions
Dynamic Analysis - API Monitor

Enable Monitoring

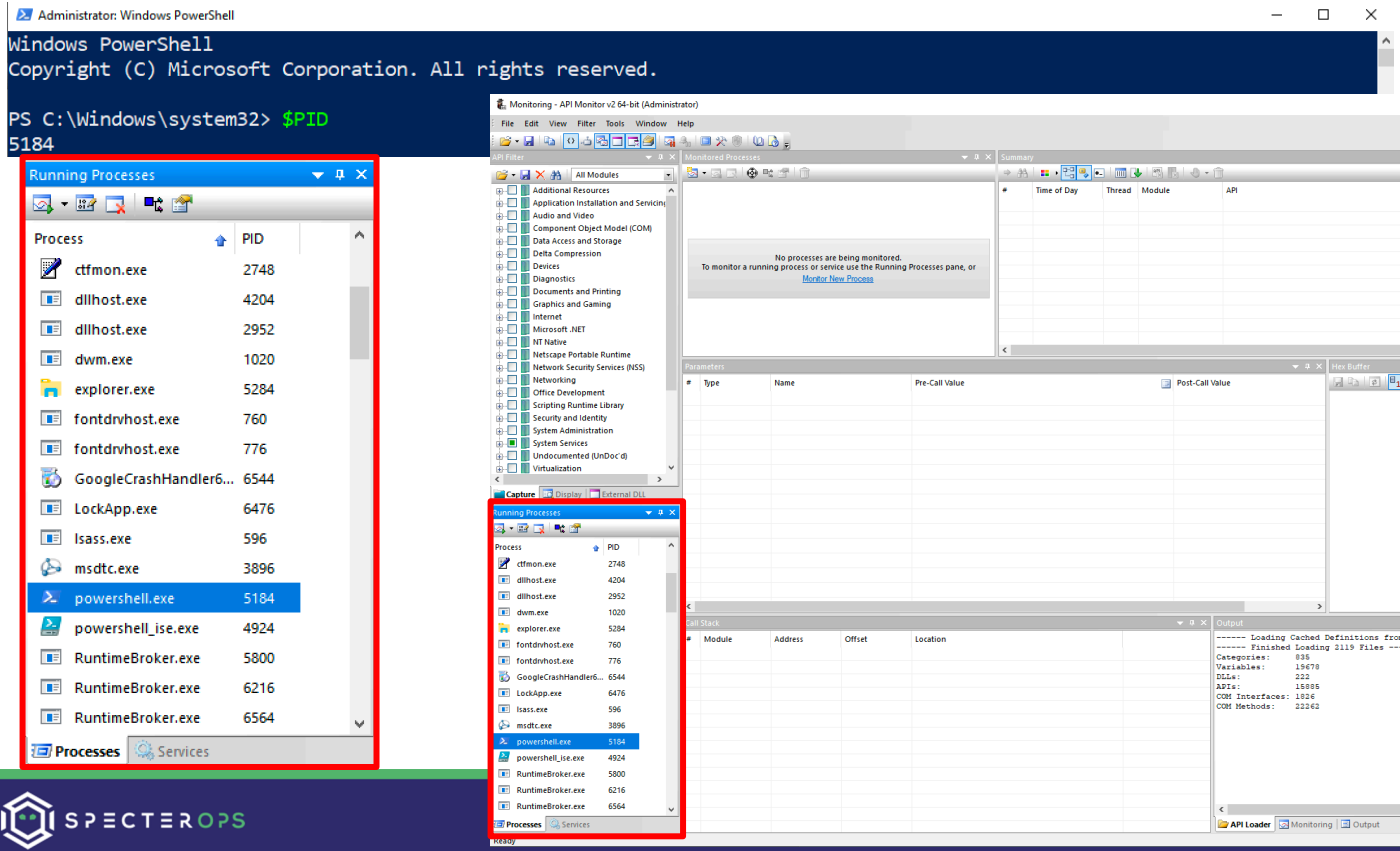




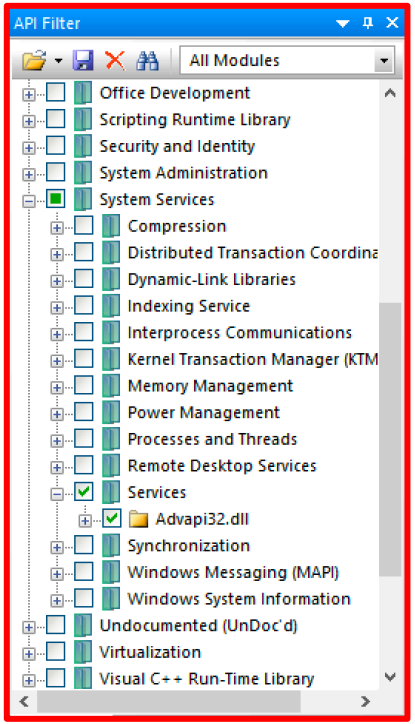
Monitoring Enabled

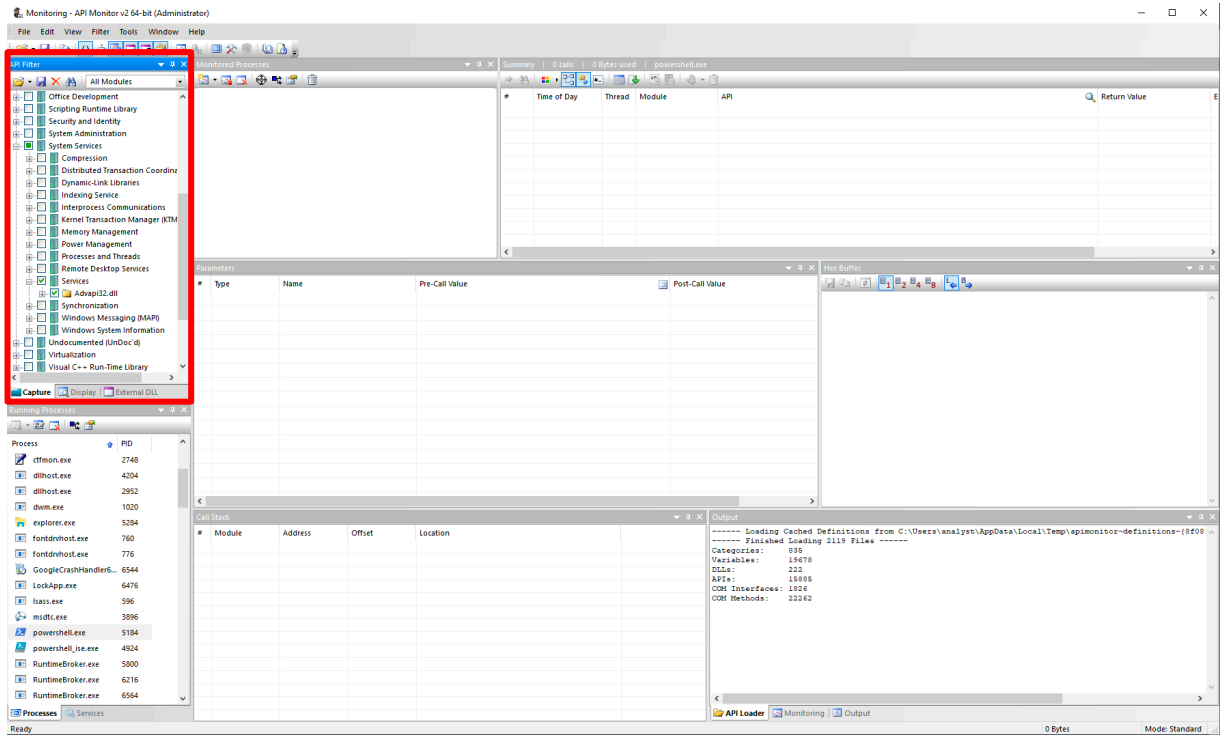


Find the Correct PowerShell Process

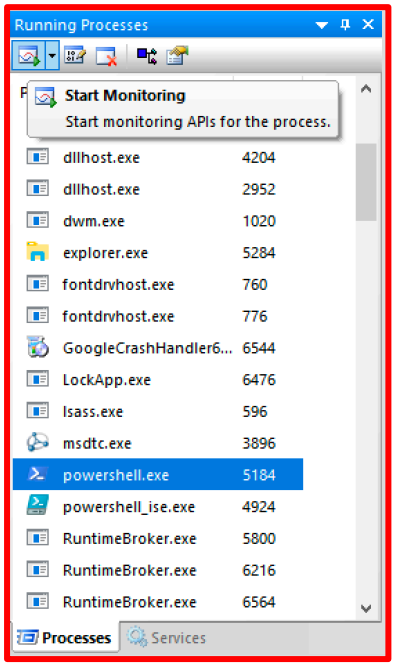


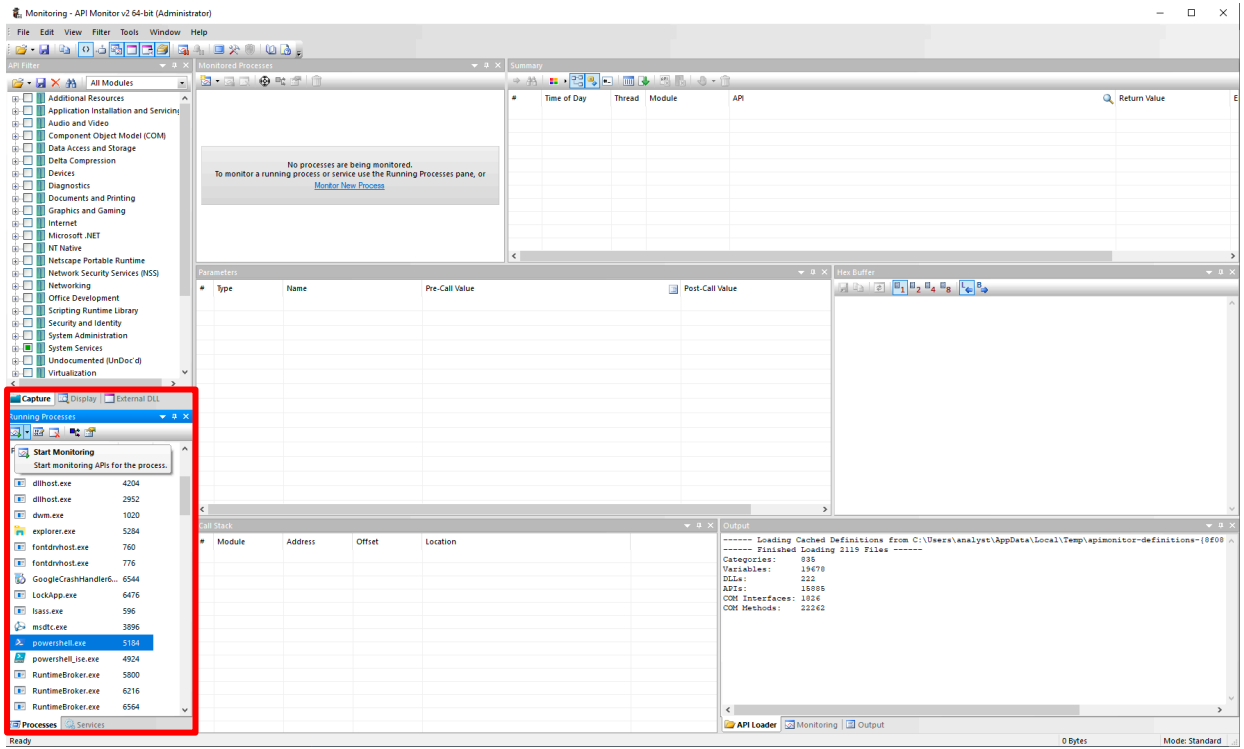
Select Services Related API Functions



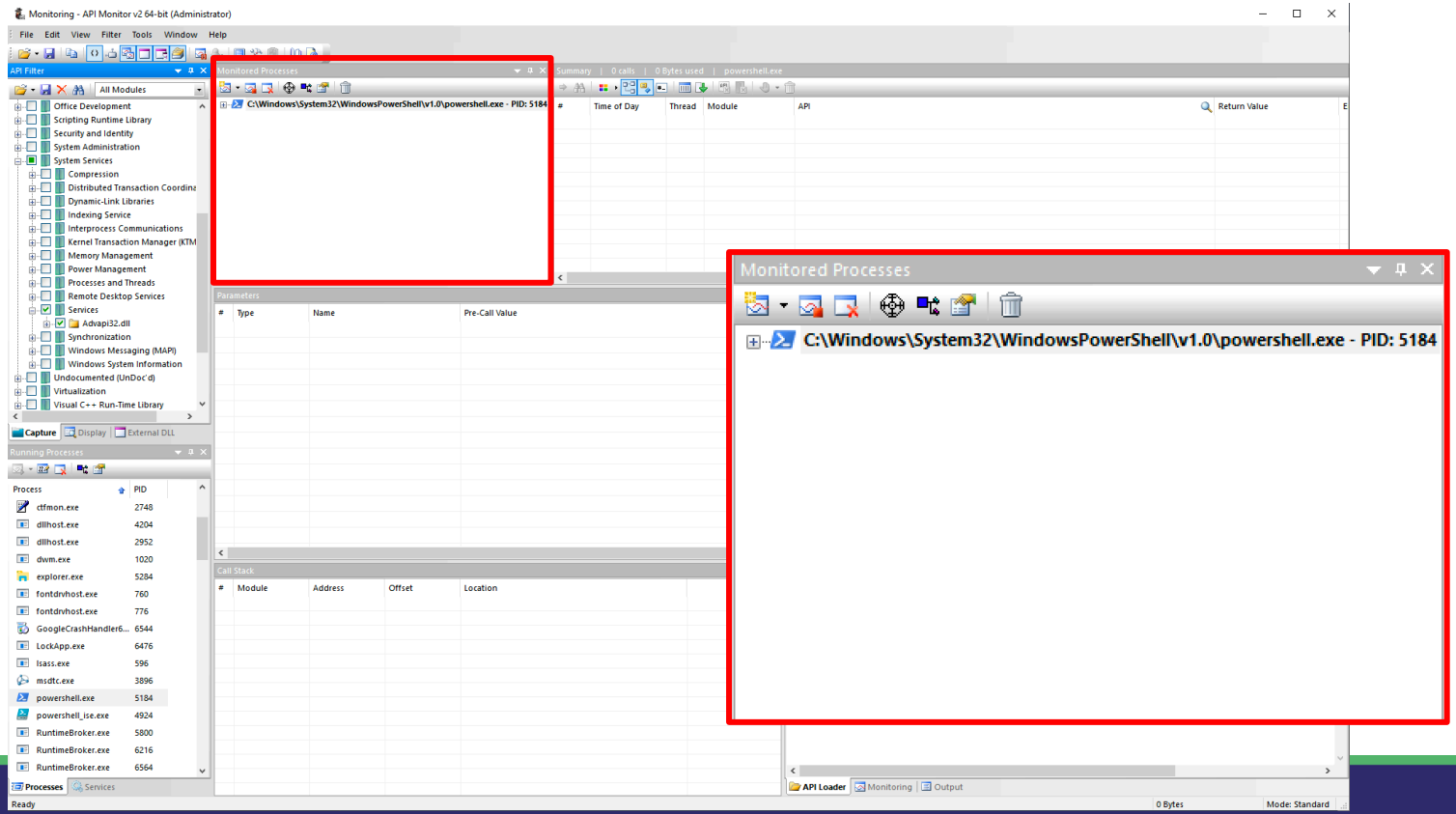


Start Monitoring PowerShell

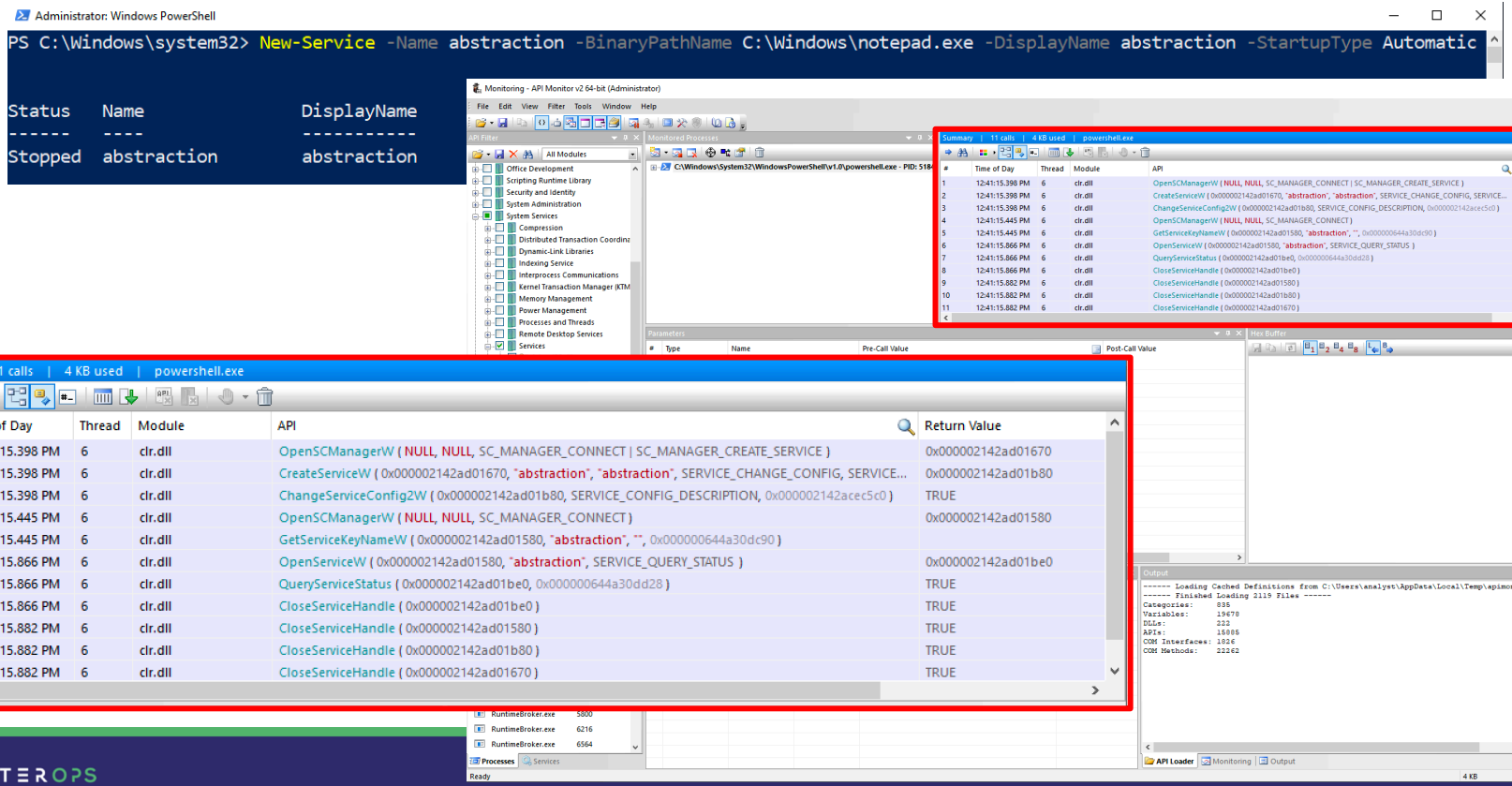




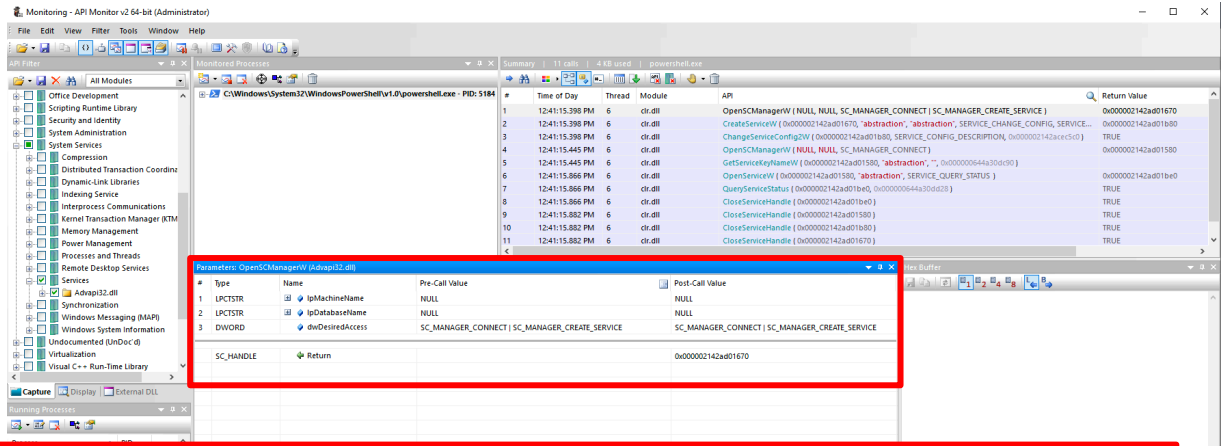
Monitoring Started



Executing New-Service



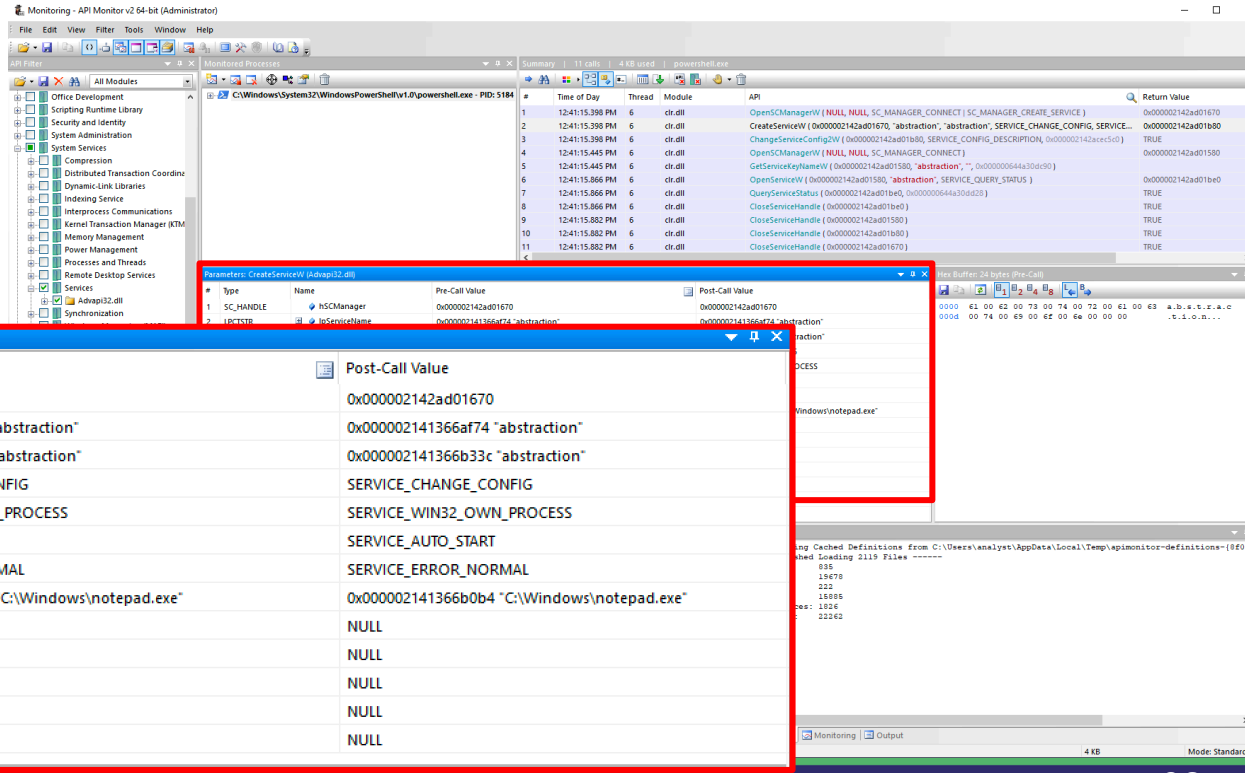
Calling OpenSCManagerW



Parameters: OpenSCManagerW (Advapi32.dll)

#	Type	Name	Pre-Call Value	Post-Call Value
1	LPCTSTR	lpMachineName	NULL	NULL
2	LPCTSTR	lpDatabaseName	NULL	NULL
3	DWORD	dwDesiredAccess	SC_MANAGER_CONNECT | SC_MANAGER_CREATE_SERVICE	SC_MANAGER_CONNECT | SC_MANAGER_CREATE_SERVICE
	SC_HANDLE	Return		0x000002142ad01670

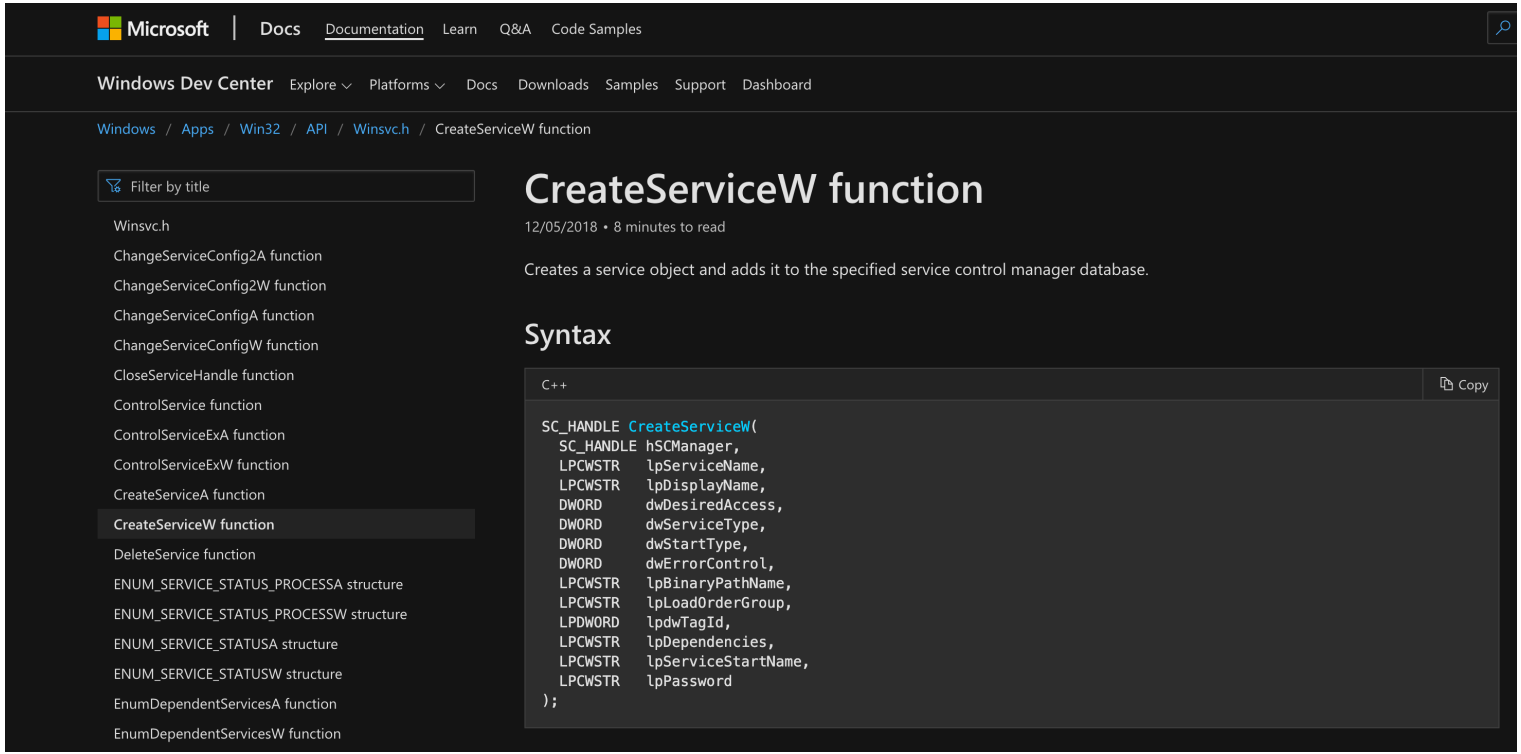
Calling CreateServiceW



#	Type	Name	Pre-Call Value	Post-Call Value
1	SC_HANDLE	hSCManager	0x000002142ad01670	0x000002142ad01670
2	LPCTSTR	lpServiceName	0x000002141366af74 "abstraction"	0x000002141366af74 "abstraction"
3	LPCTSTR	lpDisplayName	0x000002141366b33c "abstraction"	0x000002141366b33c "abstraction"
4	DWORD	dwDesiredAccess	SERVICE_CHANGE_CONFIG	SERVICE_CHANGE_CONFIG
5	DWORD	dwServiceType	SERVICE_WIN32_OWN_PROCESS	SERVICE_WIN32_OWN_PROCESS
6	DWORD	dwStartType	SERVICE_AUTO_START	SERVICE_AUTO_START
7	DWORD	dwErrorControl	SERVICE_ERROR_NORMAL	SERVICE_ERROR_NORMAL
8	LPCTSTR	lpBinaryPathName	0x000002141366b0b4 "C:\Windows\notepad.exe"	0x000002141366b0b4 "C:\Windows\notepad.exe"
9	LPCTSTR	lpLoadOrderGroup	NULL	NULL
10	LPDWORD	lpdwTagId	NULL	NULL
11	LPCTSTR	lpDependencies	NULL	NULL
12	LPCTSTR	lpServiceStartName	NULL	NULL
13	LPCTSTR	lpPassword	NULL	NULL

T1050 - Service Creation	
Tools	New-Service
Windows API	OpenSCManager/CreateService
RPC Interface/Named Pipe	367ABB81-9844-35F1-AD32-98F038001003 [MS-SCMR]
	\\PIPE\svcctl
Registry Service Database	HKLM\SYSTEM\CurrentControlSet\Services

Visit Function Documentation



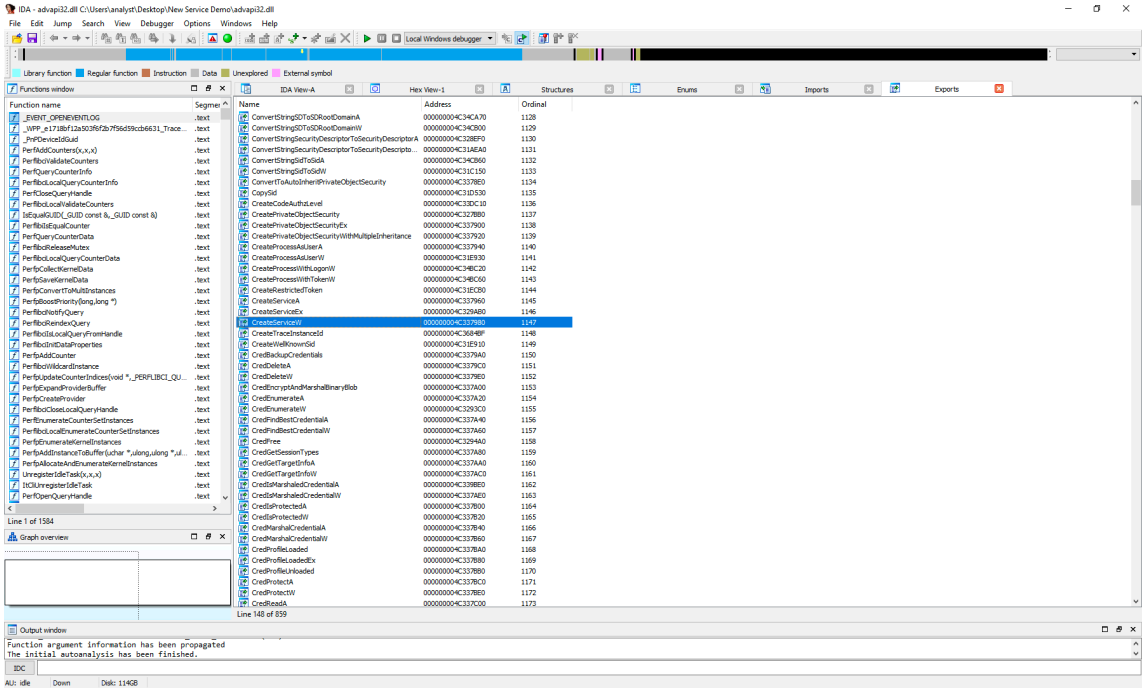
Find Exporting DLL

Requirements

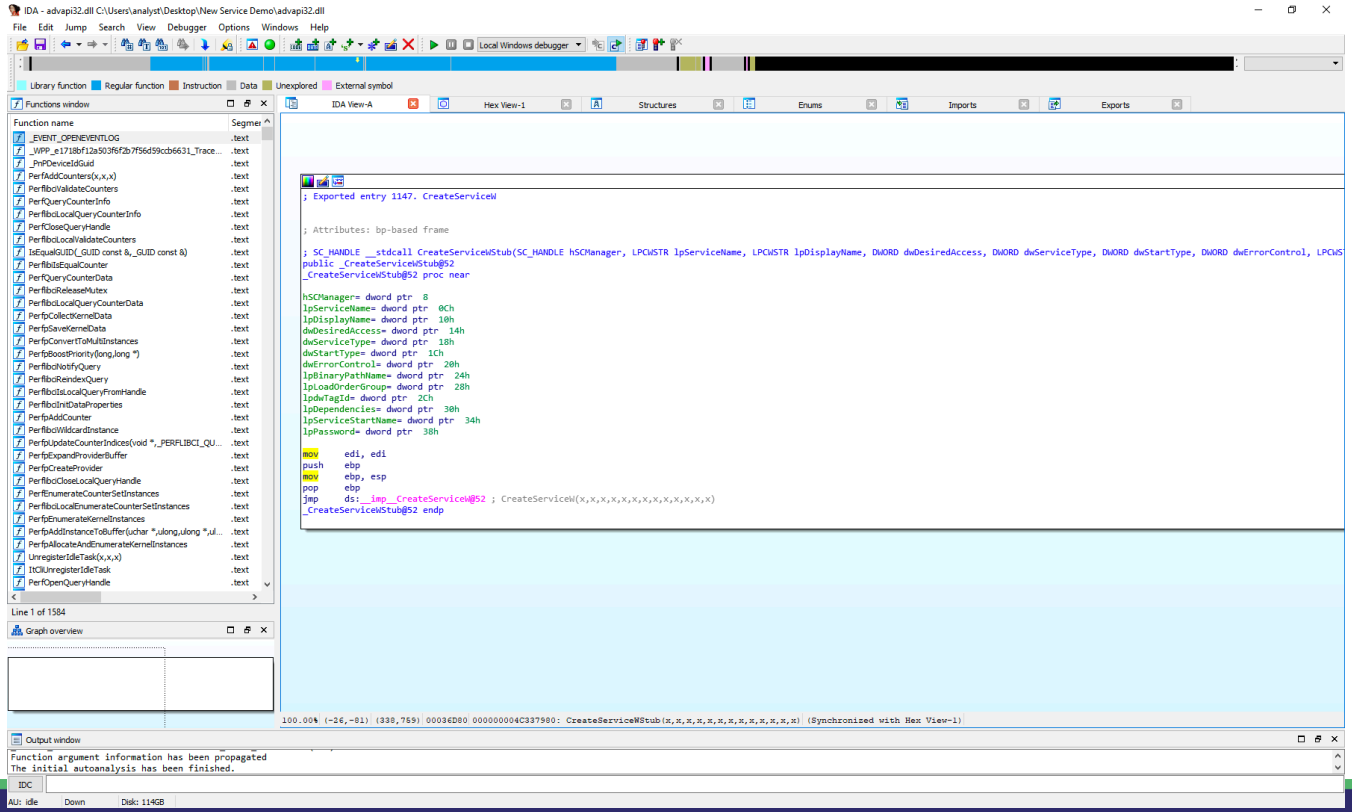
Minimum supported client	Windows XP [desktop apps only]
Minimum supported server	Windows Server 2003 [desktop apps only]
Target Platform	Windows
Header	winsvc.h (include Windows.h)
Library	Advapi32.lib
DLL	Advapi32.dll

https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-createservicew#requirements

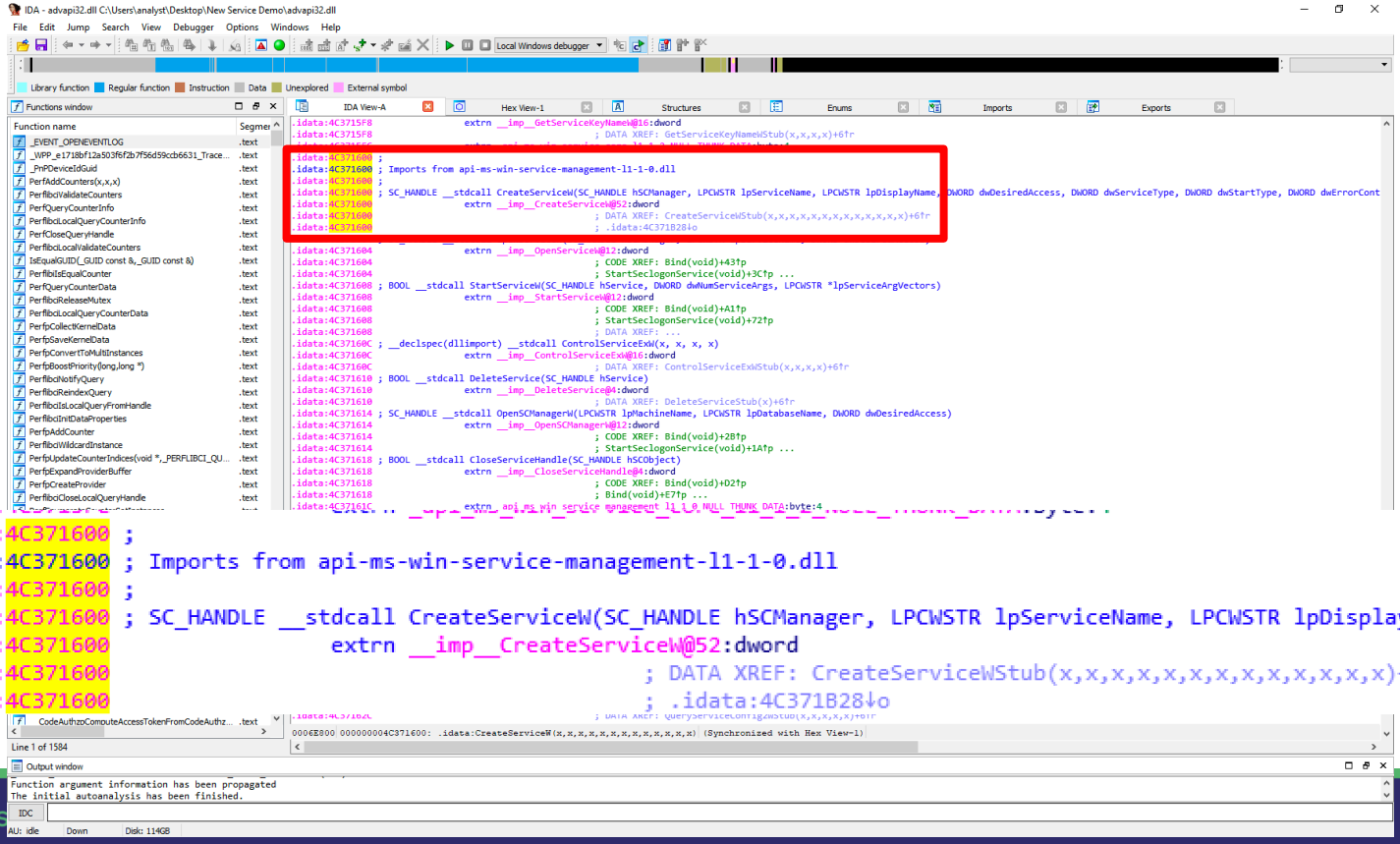
ADVAPI32.DLL – Export Table



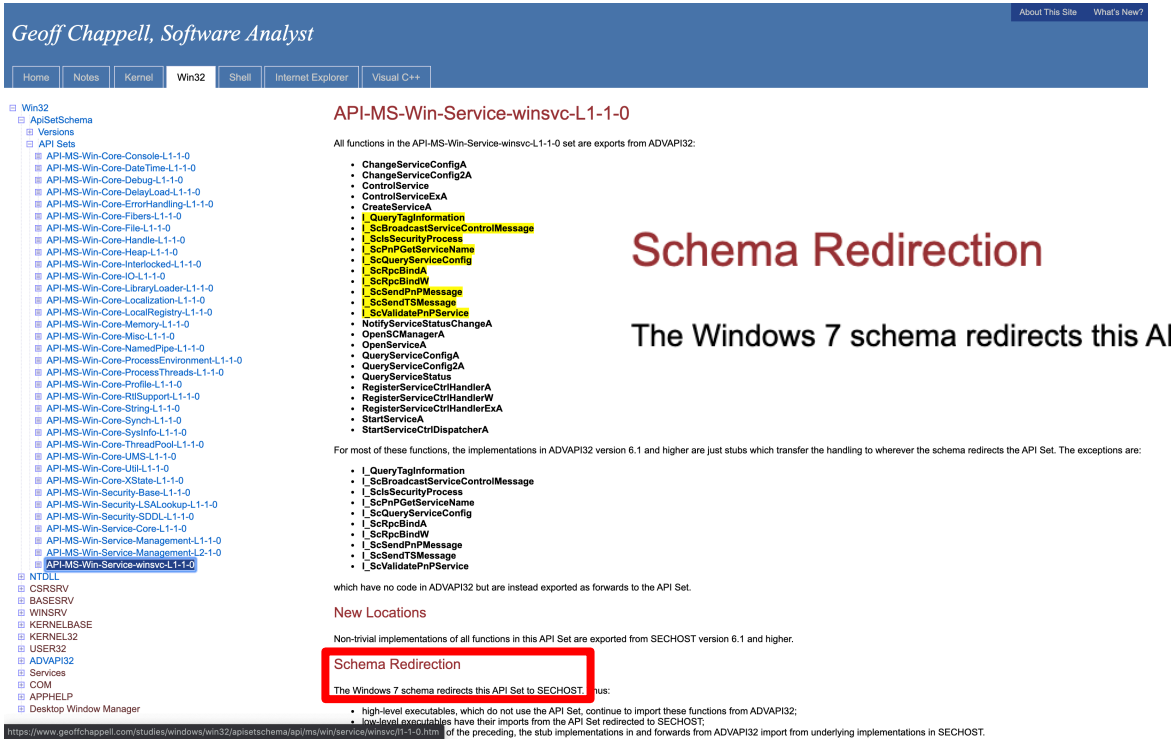
ADVAPI32.DLL – CreateServiceW



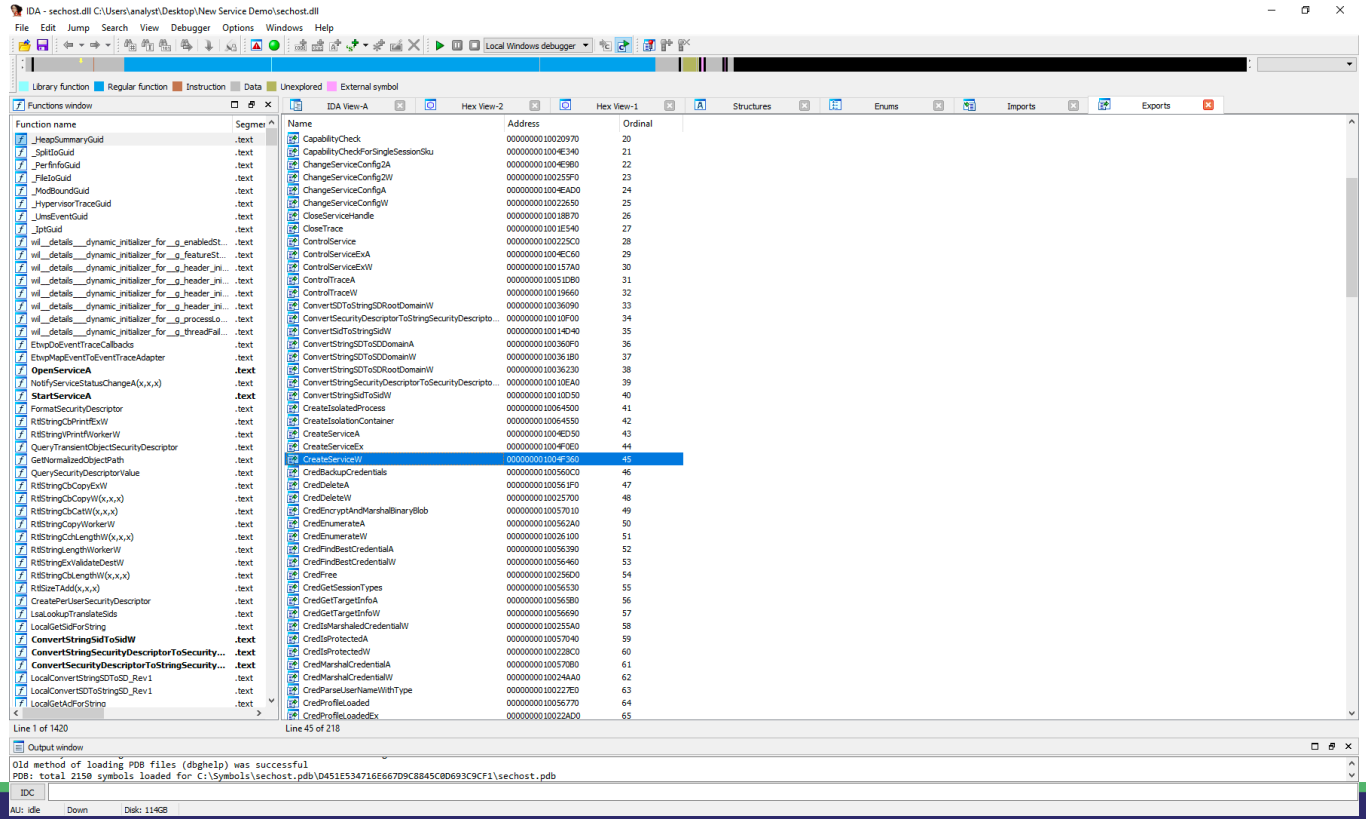
ADVAPI32.DLL – Forwarded Export



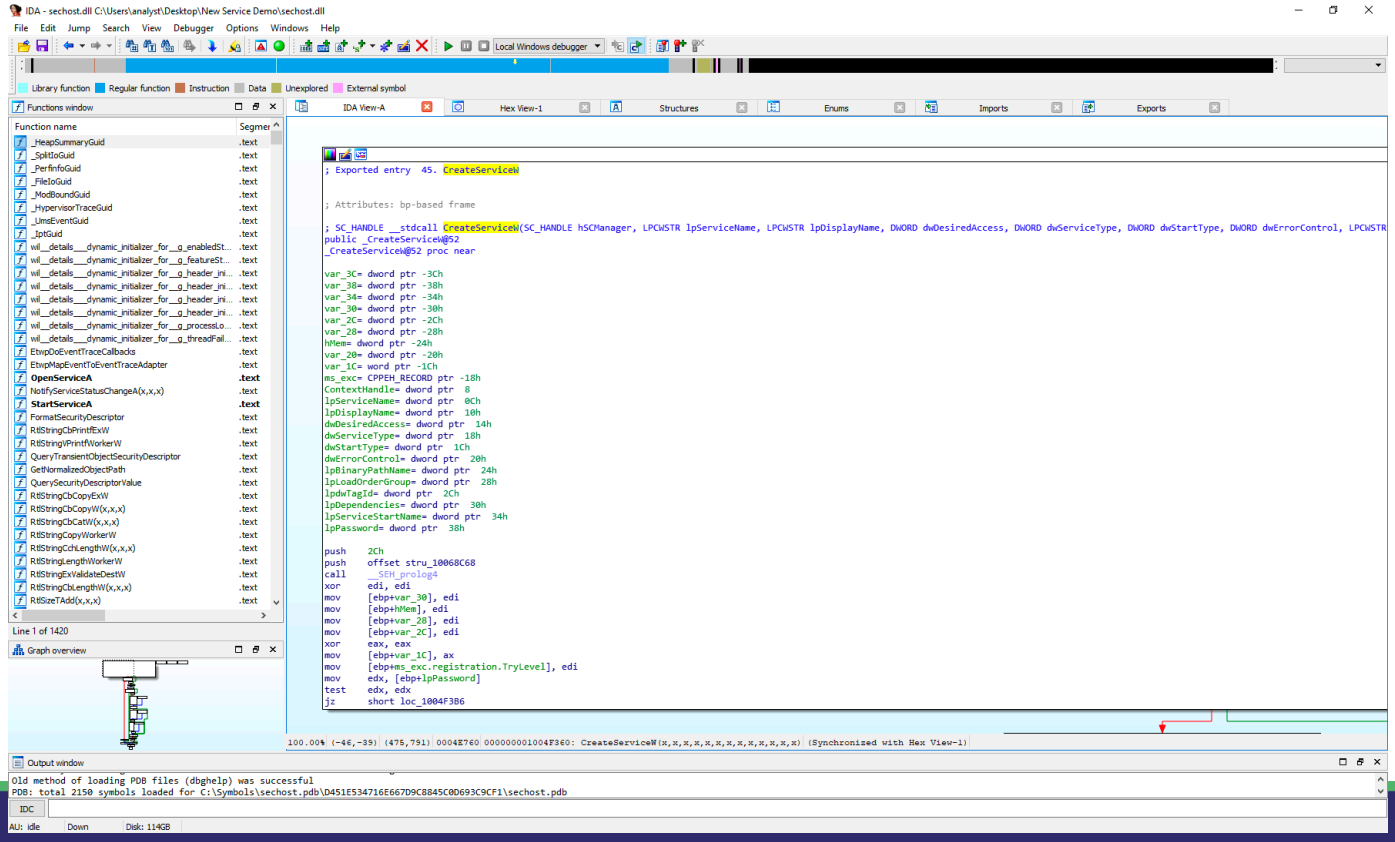
API Sets – API-MS-Win-Service-winsvc-L1-1-0



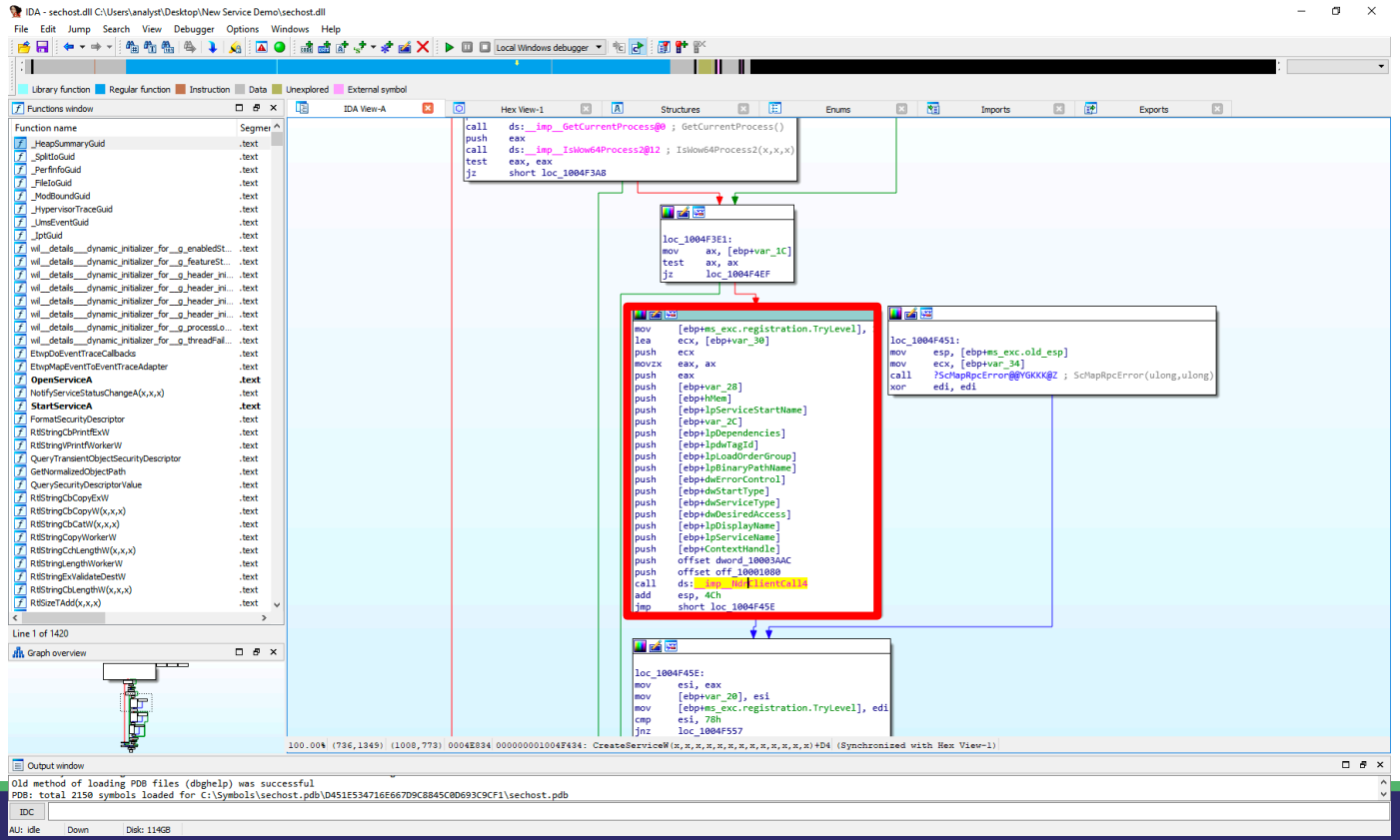
SECHOST.DLL – Export Table



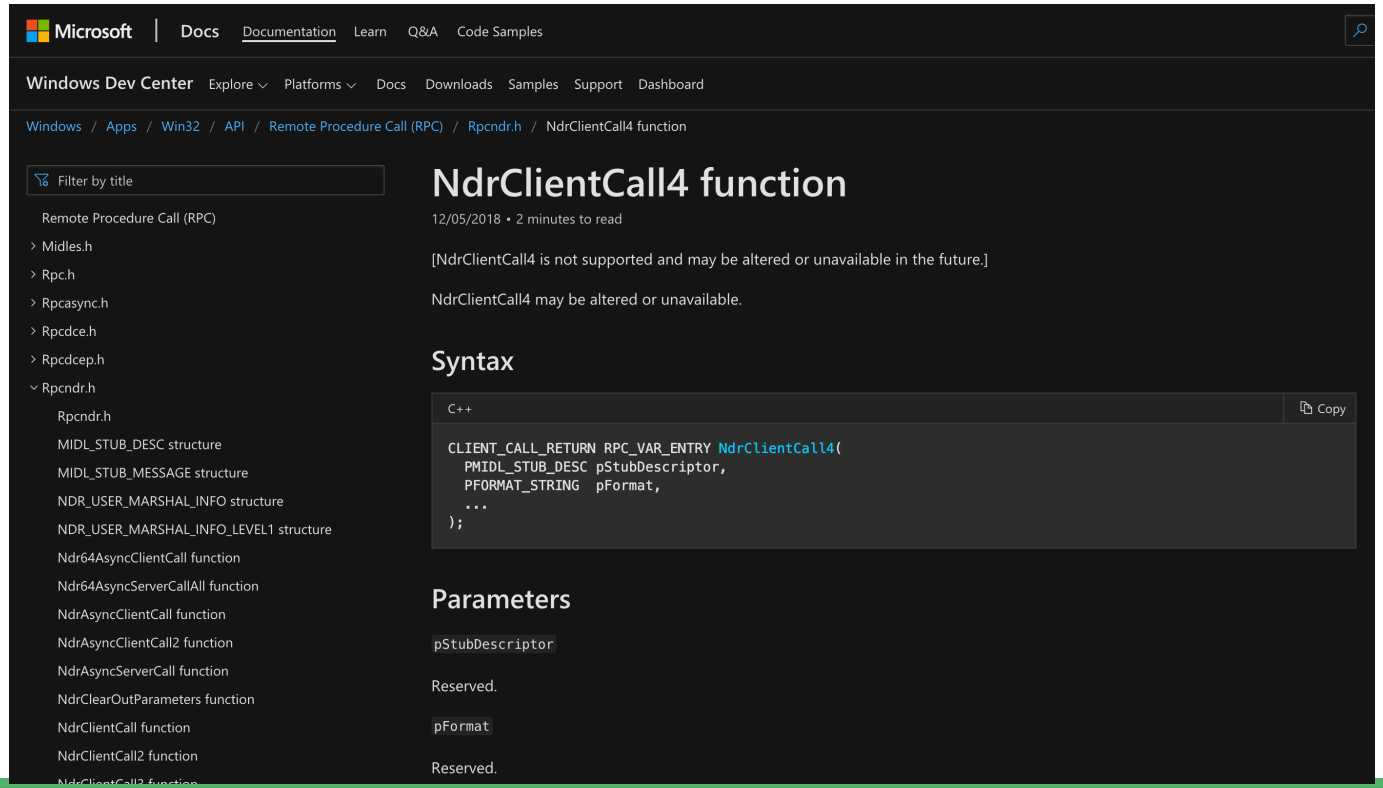
SECHOST.DLL - CreateServiceW



SECHOST.DLL – CreateServiceW Calls NdrClientCall4



NdrClientCall4



MIDL_STUB_DESC

MIDL_STUB_DESC structure

12/05/2018 • 2 minutes to read

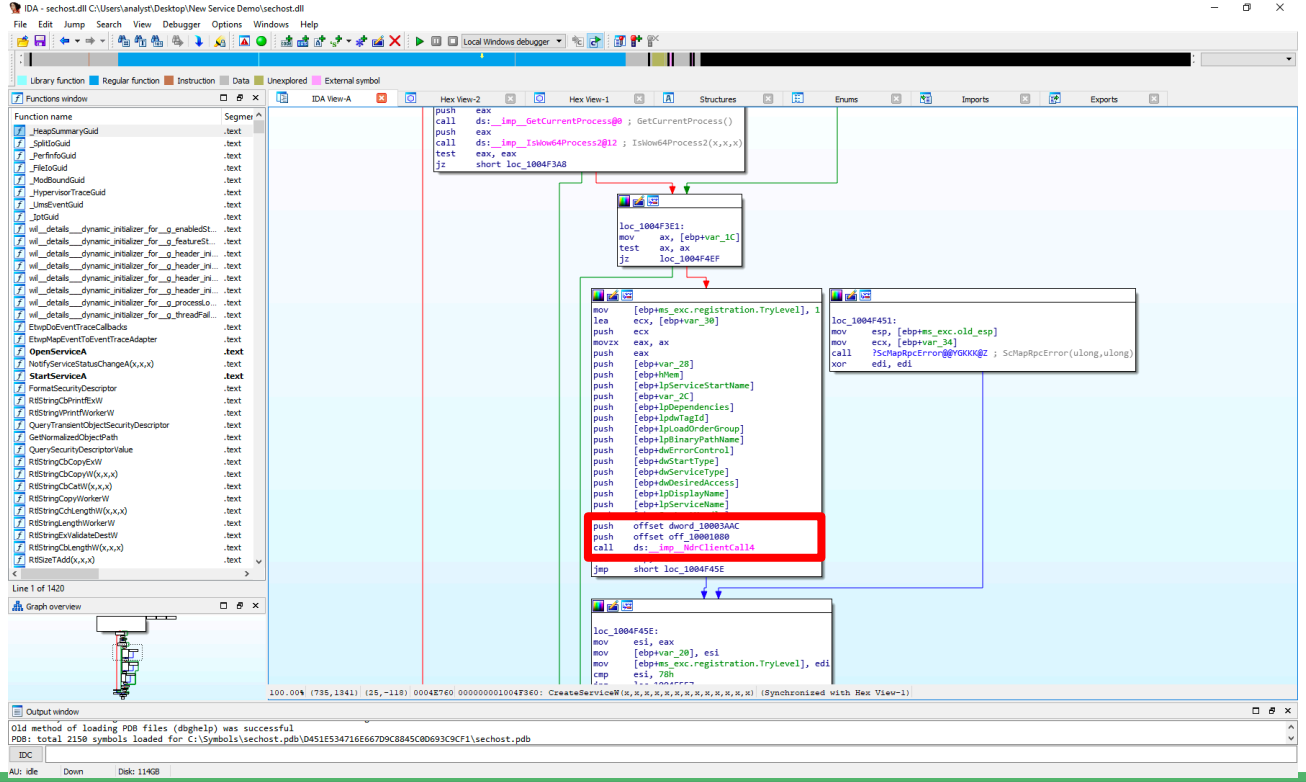
The **MIDL_STUB_DESC** structure is a MIDL-generated structure that contains information about the interface stub regarding RPC calls between the client and server.

Syntax

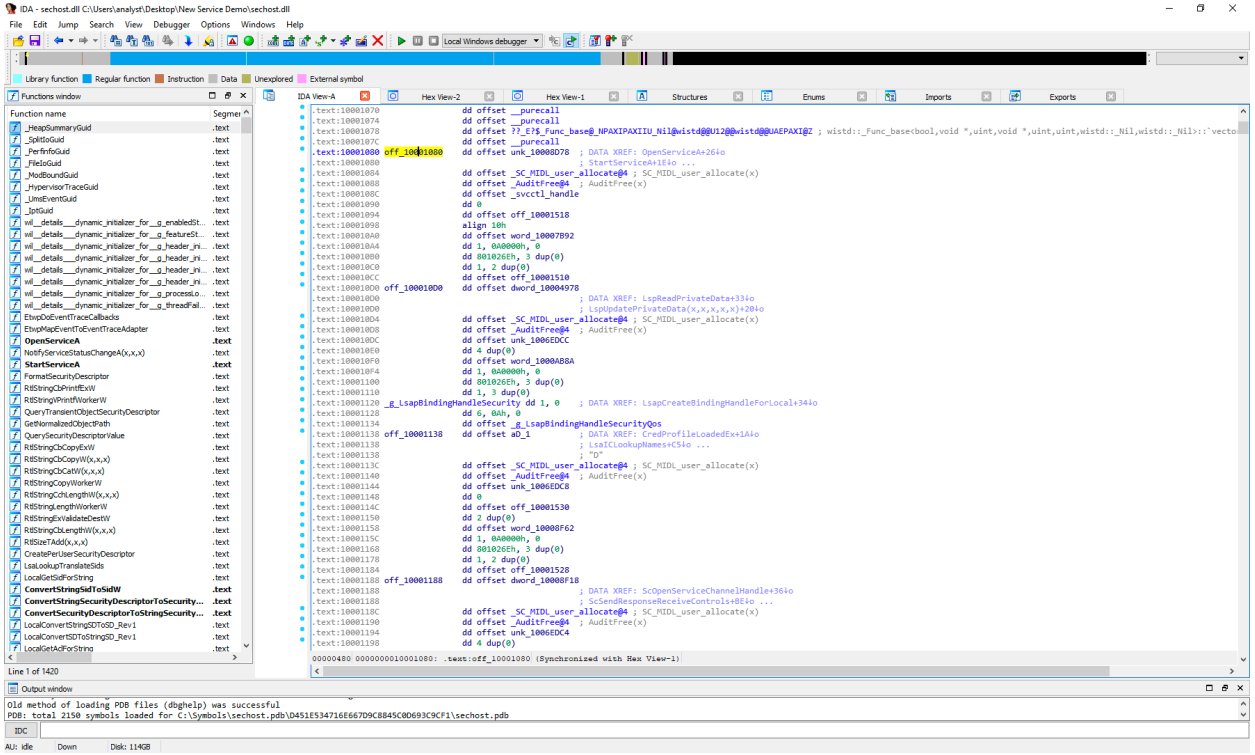
C++ Copy

```
typedef struct _MIDL_STUB_DESC {
  void                                *RpcInterfaceInformation;
  void * )(size_t)                                 *(pfnAllocate;
  void()(void *)                                 * pfnFree;
  union {
    handle_t              *pAutoHandle;
    handle_t              *pPrimitiveHandle;
    PGENERIC_BINDING_INFO pGenericBindingInfo;
  } IMPLICIT_HANDLE_INFO;
  const NDR_RUNDOWN                   *apfnNdrRundownRoutines;
  const GENERIC_BINDING_ROUTINE_PAIR  *aGenericBindingRoutinePairs;
  const EXPR_EVAL                     *apfnExprEval;
  const XMIT_ROUTINE_QUINTUPLE        *aXmitQuintuple;
  const unsigned char                 *pFormatTypes;
  int                                 fCheckBounds;
  unsigned long                       Version;
  MALLOC_FREE_STRUCT                  *pMallocFreeStruct;
  long                                MIDLVersion;
  const COMM_FAULT_OFFSETS            *CommFaultOffsets;
  const USER_MARSHAL_ROUTINE_QUADRUPLE *aUserMarshalQuadruple;
  const NDR_NOTIFY_ROUTINE            *NotifyRoutineTable;
  ULONG_PTR                           mFlags;
  const NDR_CS_ROUTINES               *CsRoutineTables;
  void                                *ProxyServerInfo;
  const NDR_EXPR_DESC                 *pExprInfo;
} MIDL_STUB_DESC;
```

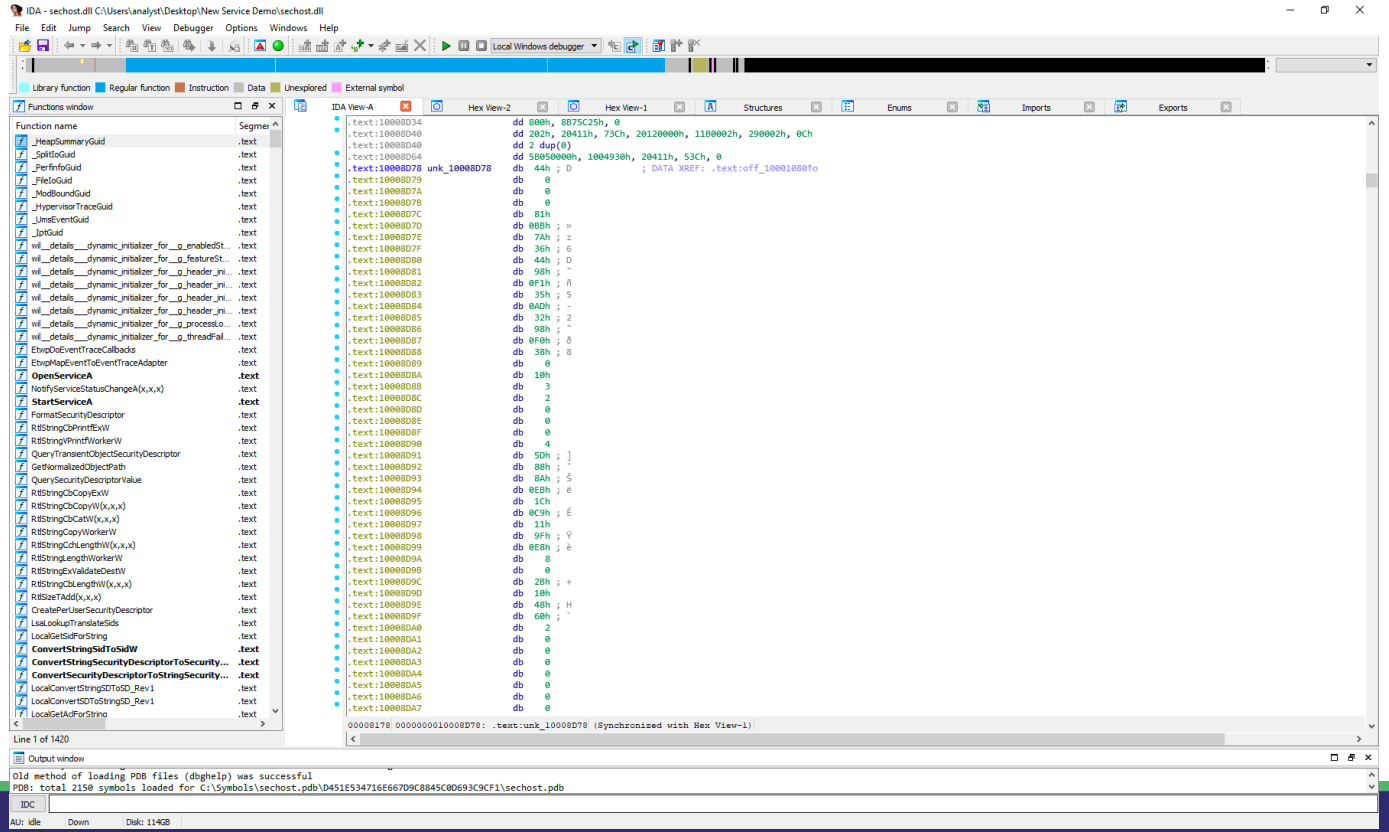
SECHOST.DLL – RpcInterfaceInformation



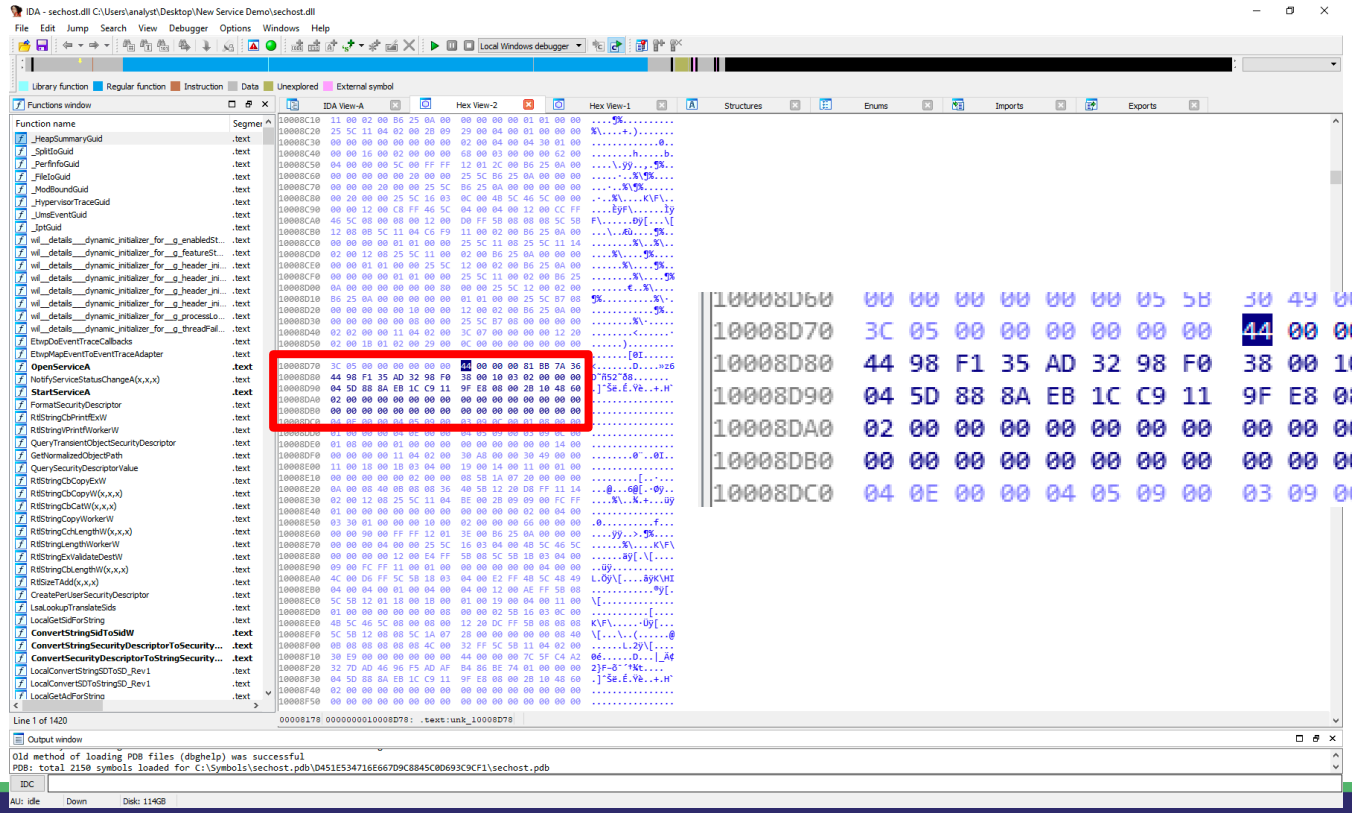
SECHOST.DLL – RpcInterfaceInformation



SECHOST.DLL – RpcInterfaceInformation

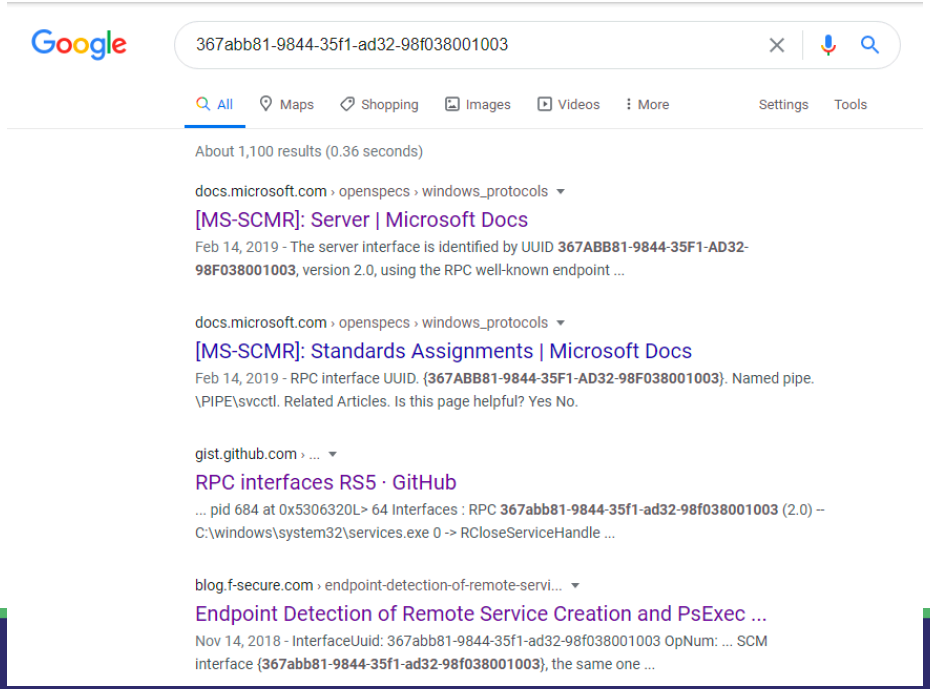


SECHOST.DLL – RpcInterfaceInformation

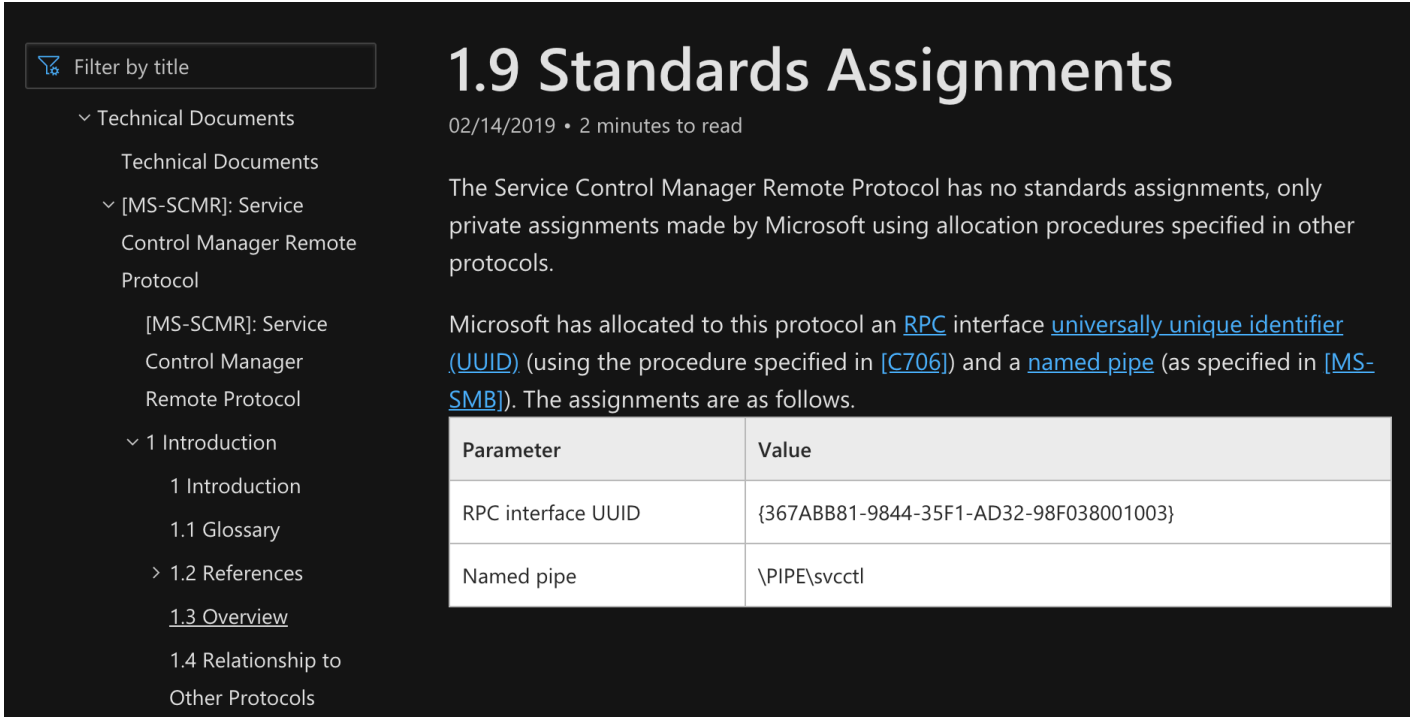


RPC Interface GUID

```
PS C:\Windows\system32> [Guid][Byte[]]@(0x81,0xBB,0x7A,0x36,0x44,0x98,0xF1,0x35,0xAD,0x32,0x98,0xF0,0x38,0x00,0x10,0x03)

Guid
----
367abb81-9844-35f1-ad32-98f038001003
```



MS-SCMR – Standards Assignment



Enigma0x3 – RPC Interface Inventory

gist.github.com/enigma0x3/2e549345e7f0ac88fad130e2444bb702#file-rpc_dump_rs5-txt-L62-L127

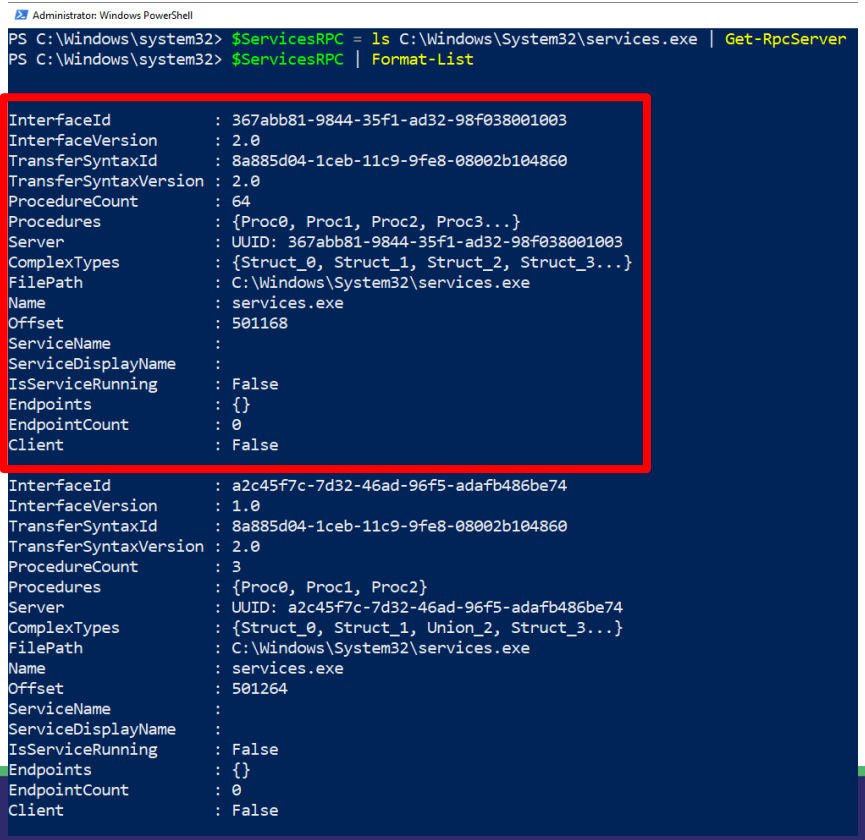
```
64

Interfaces :
RPC  367abb81-9844-35f1-ad32-98f038001003 (2.0) -- C:\windows\system32\services.exe
        0 -> RCloseServiceHandle
        1 -> RControlService
        2 -> RDeleteService
        3 -> RLockServiceDatabase
        4 -> RQueryServiceObjectSecurity
        5 -> RSetServiceObjectSecurity
        6 -> RQueryServiceStatus
        7 -> RSetServiceStatus
        8 -> RUnlockServiceDatabase
        9 -> RNotifyBootConfigStatus
        10 -> RI_ScSetServiceBitsW
        11 -> RChangeServiceConfigW
        12 -> RCreateServiceW
        13 -> REnumDependentServicesW
        14 -> REnumServicesStatusW
        15 -> ROpenSCManagerW
        16 -> ROpenServiceW
        17 -> RQueryServiceConfigW
        18 -> RQueryServiceLockStatusW
        19 -> RStartServiceW
        20 -> RGetServiceDisplayNameW
        21 -> RGetServiceKeyNameW
        22 -> CServiceRecord::GetStatusInternal
        23 -> RChangeServiceConfigA
        24 -> RCreateServiceA
        25 -> REnumDependentServicesA
        26 -> REnumServicesStatusA
        27 -> ROpenSCManagerA
        28 -> ROpenServiceA
        29 -> RQueryServiceConfigA
        30 -> RQueryServiceLockStatusA
        31 -> RStartServiceA
        32 -> RGetServiceDisplayNameA
```

Enigma0x3 – RPC Interface Inventory

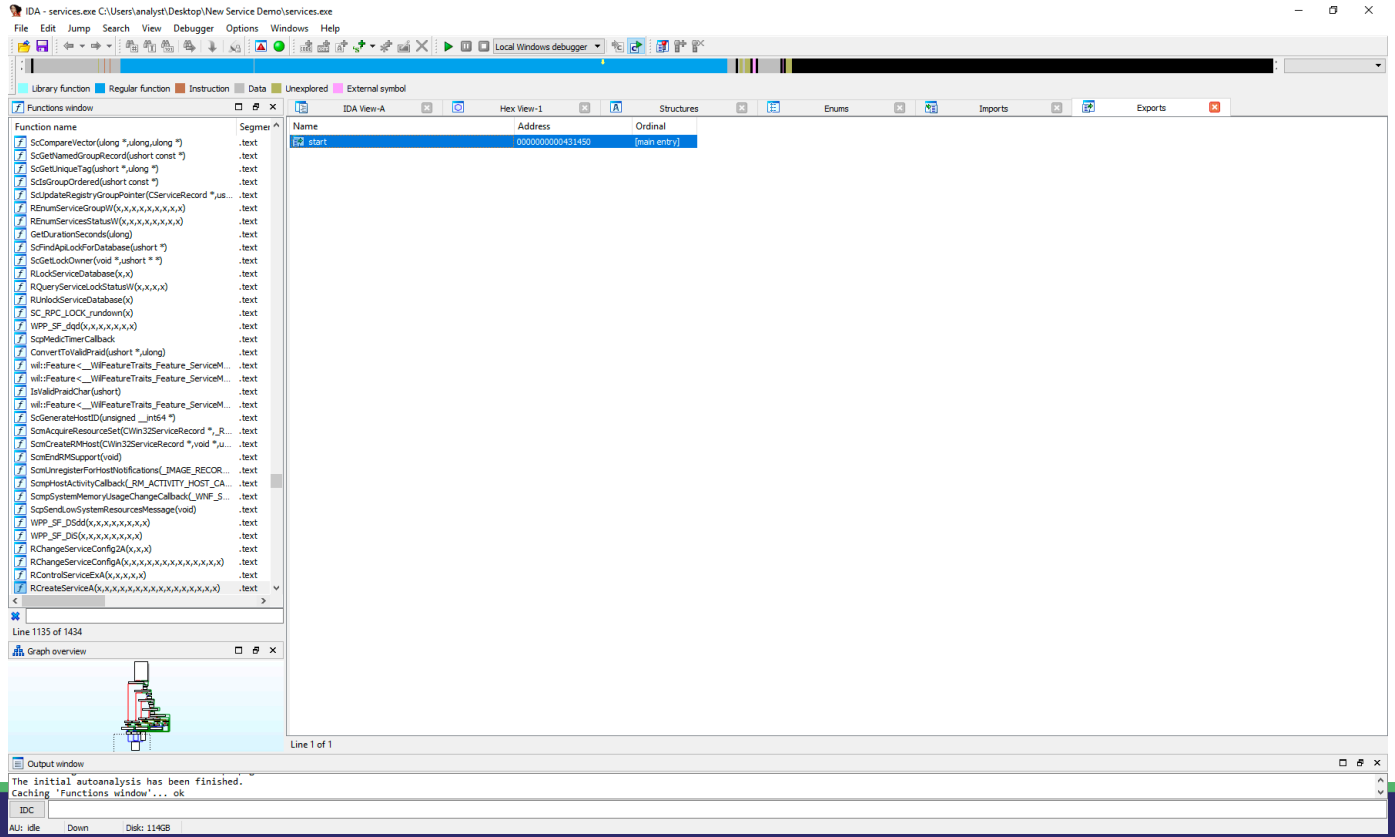
gist.github.com/enigma0x3/2e549345e7f0ac88fad130e2444bb702#file-rpc_dump_rs5-txt-L62-L127

```
64

Interfaces :
RPC 367abb81-9844-35f1-ad32-98f038001003 (2.0) - C:\windows\system32\services.exe
        0 -> RCloseServiceHandle
        1 -> RControlService
        2 -> RDeleteService
        3 -> RLockServiceDatabase
        4 -> RQueryServiceObjectSecurity
        5 -> RSetServiceObjectSecurity
        6 -> RQueryServiceStatus
        7 -> RSetServiceStatus
        8 -> RUnlockServiceDatabase
        9 -> RNotifyBootConfigStatus
        10 -> RI_ScSetServiceBitsW
        11 -> RChangeServiceConfigW
        12 -> RCreateServiceW
        13 -> REnumDependentServicesW
        14 -> REnumServicesStatusW
        15 -> ROpenSCManagerW
        16 -> ROpenServiceW
        17 -> RQueryServiceConfigW
        18 -> RQueryServiceLockStatusW
        19 -> RStartServiceW
        20 -> RGetServiceDisplayNameW
        21 -> RGetServiceKeyNameW
        22 -> CServiceRecord::GetStatusInternal
        23 -> RChangeServiceConfigA
        24 -> RCreateServiceA
        25 -> REnumDependentServicesA
        26 -> REnumServicesStatusA
        27 -> ROpenSCManagerA
        28 -> ROpenServiceA
        29 -> RQueryServiceConfigA
        30 -> RQueryServiceLockStatusA
        31 -> RStartServiceA
        32 -> RGetServiceDisplayNameA
```

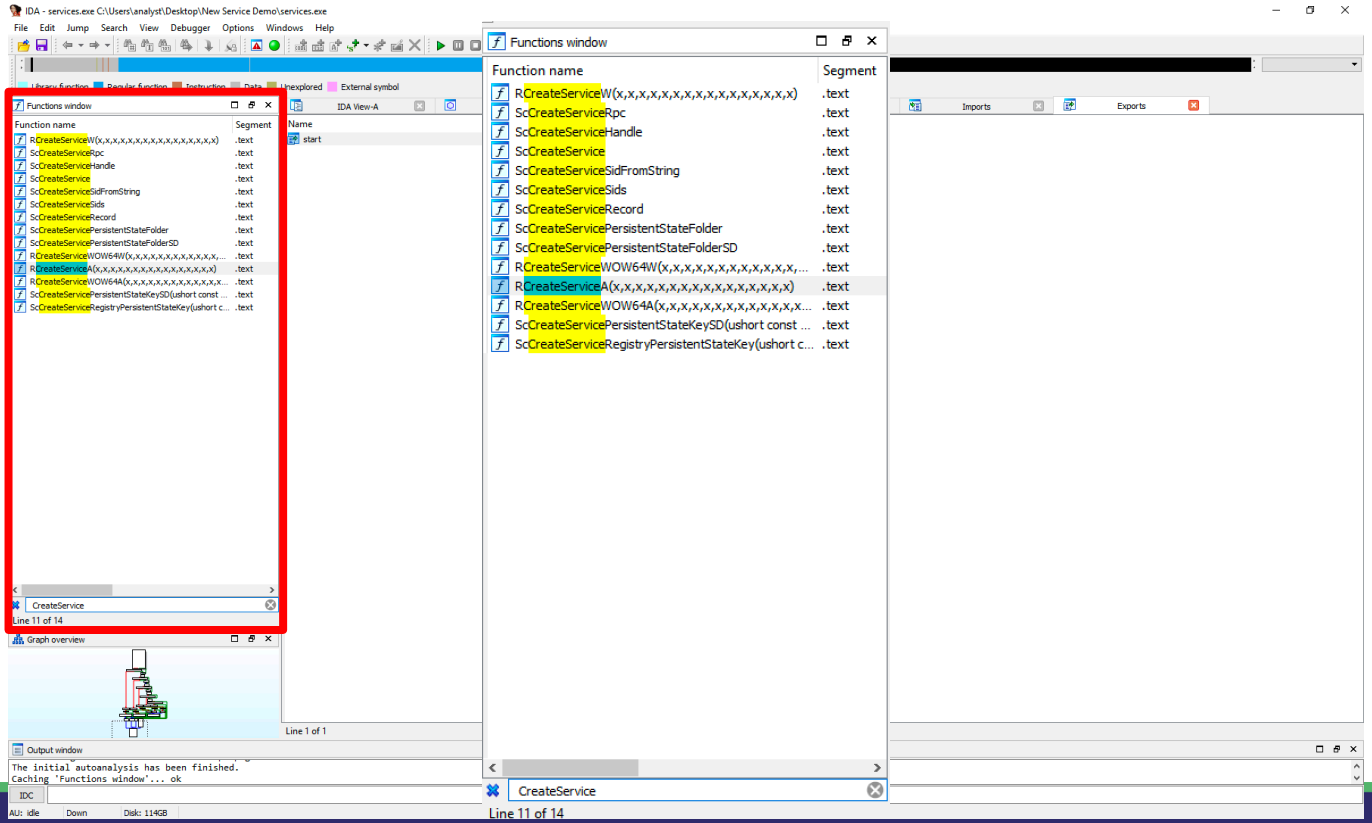
NtObjectManager – RPC Server



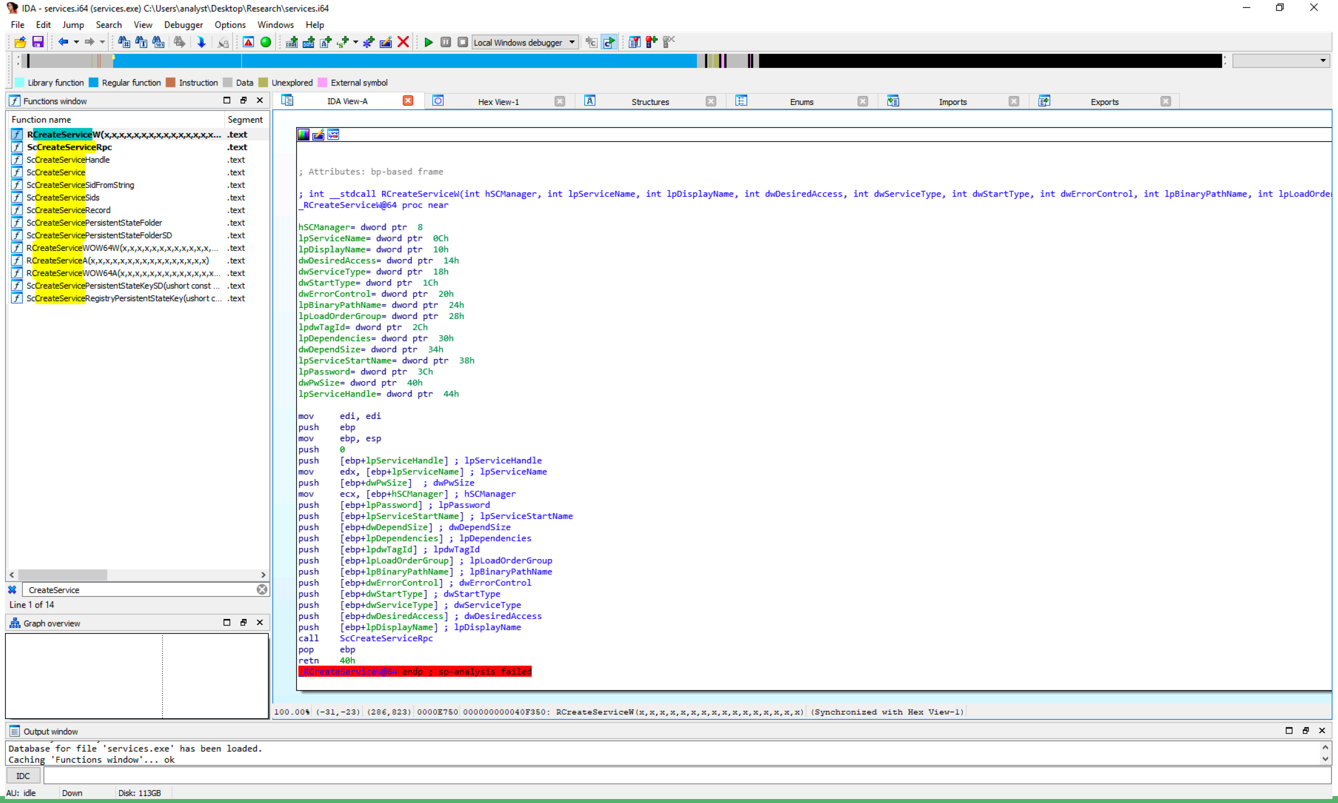
SERVICES.EXE – Exports Table



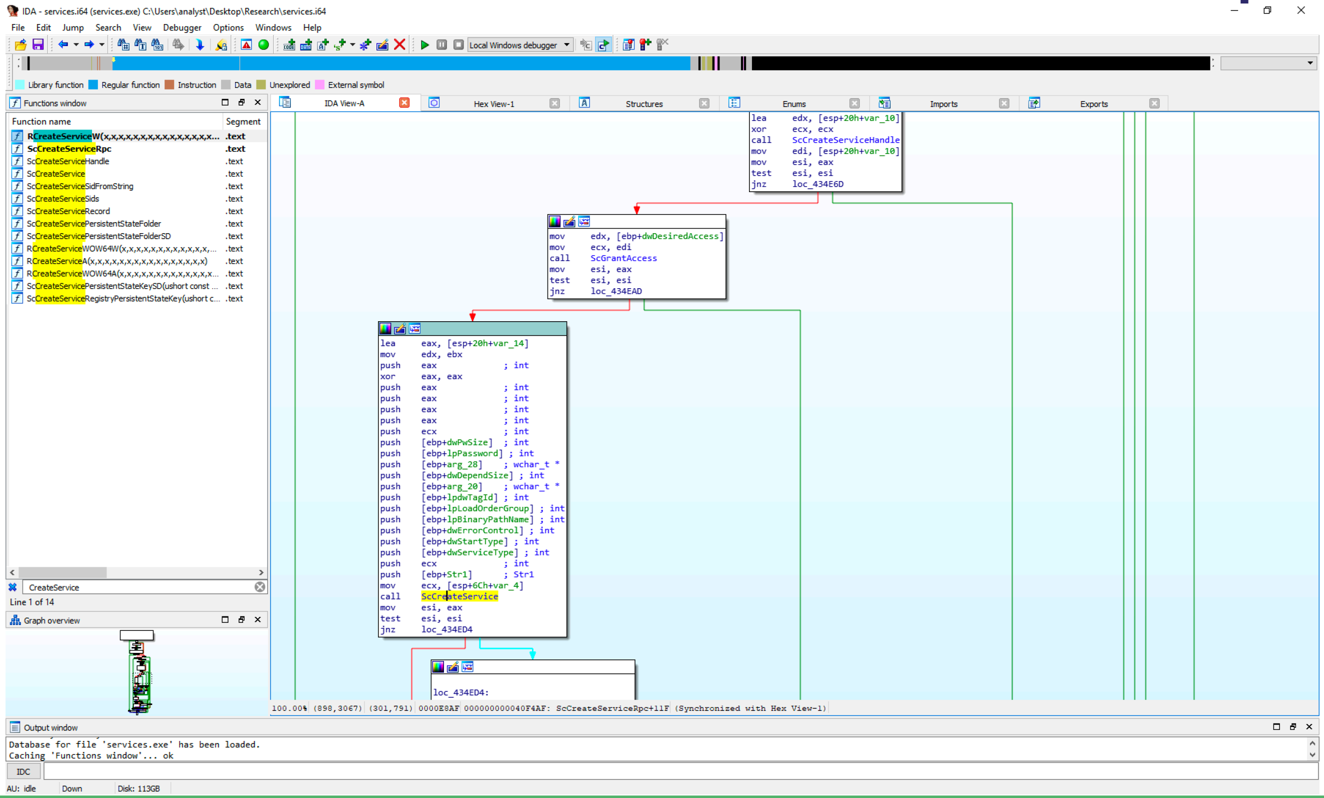
SERVICES.EXE – Sub Procedures



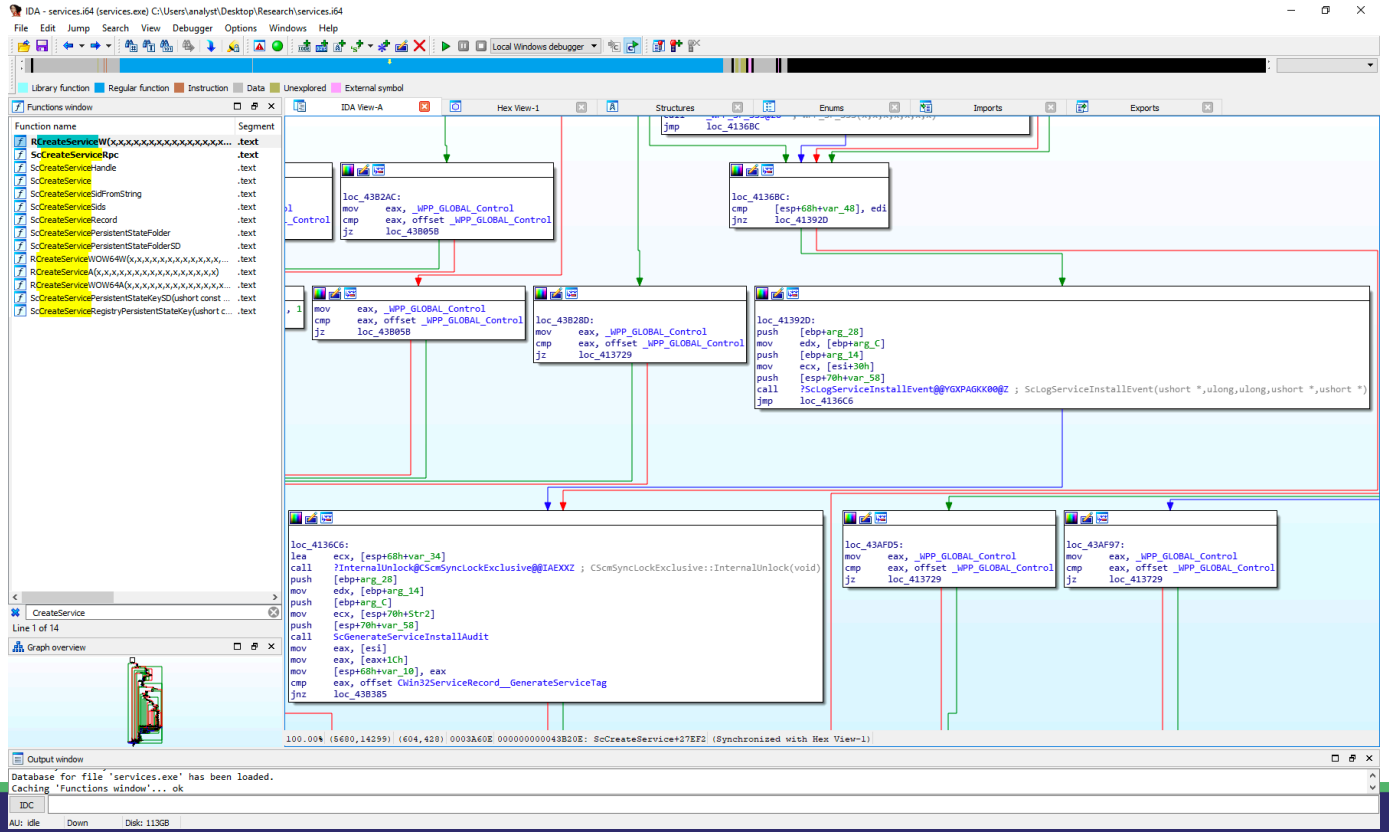
SERVICES.EXE - RCreateServiceW



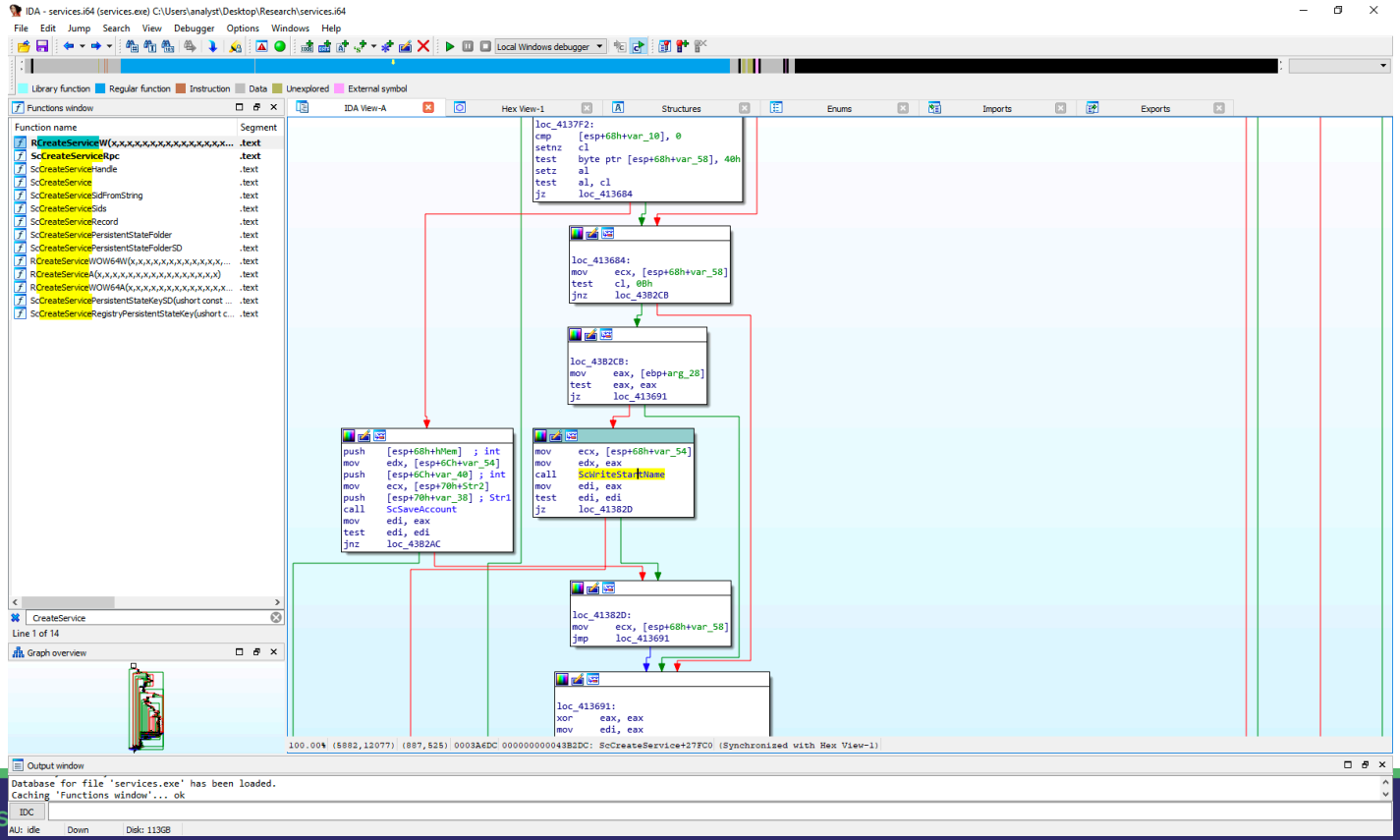
SERVICES.EXE - ScCreateServiceRpc



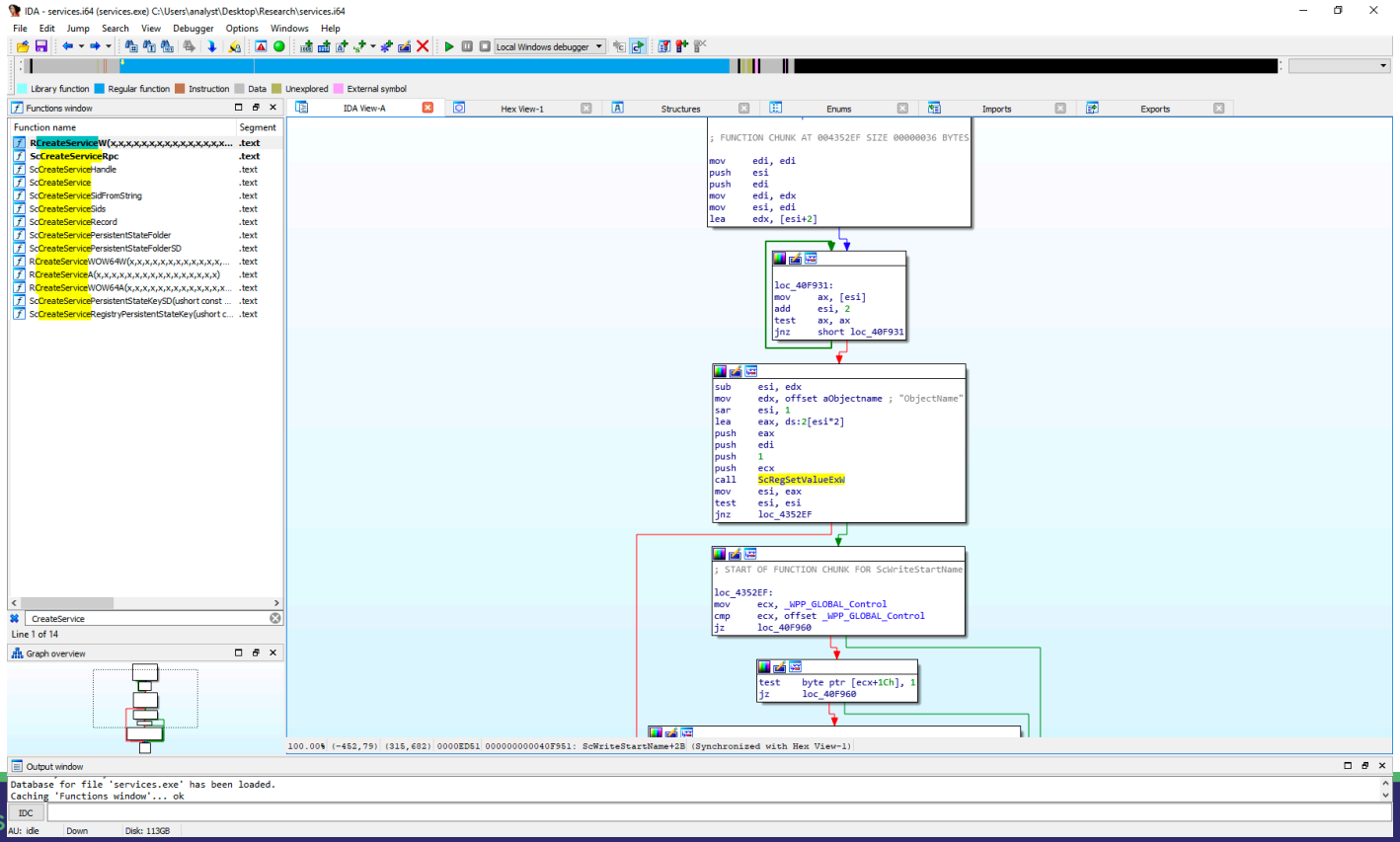
SERVICES.EXE – ScCreateService (Logging)



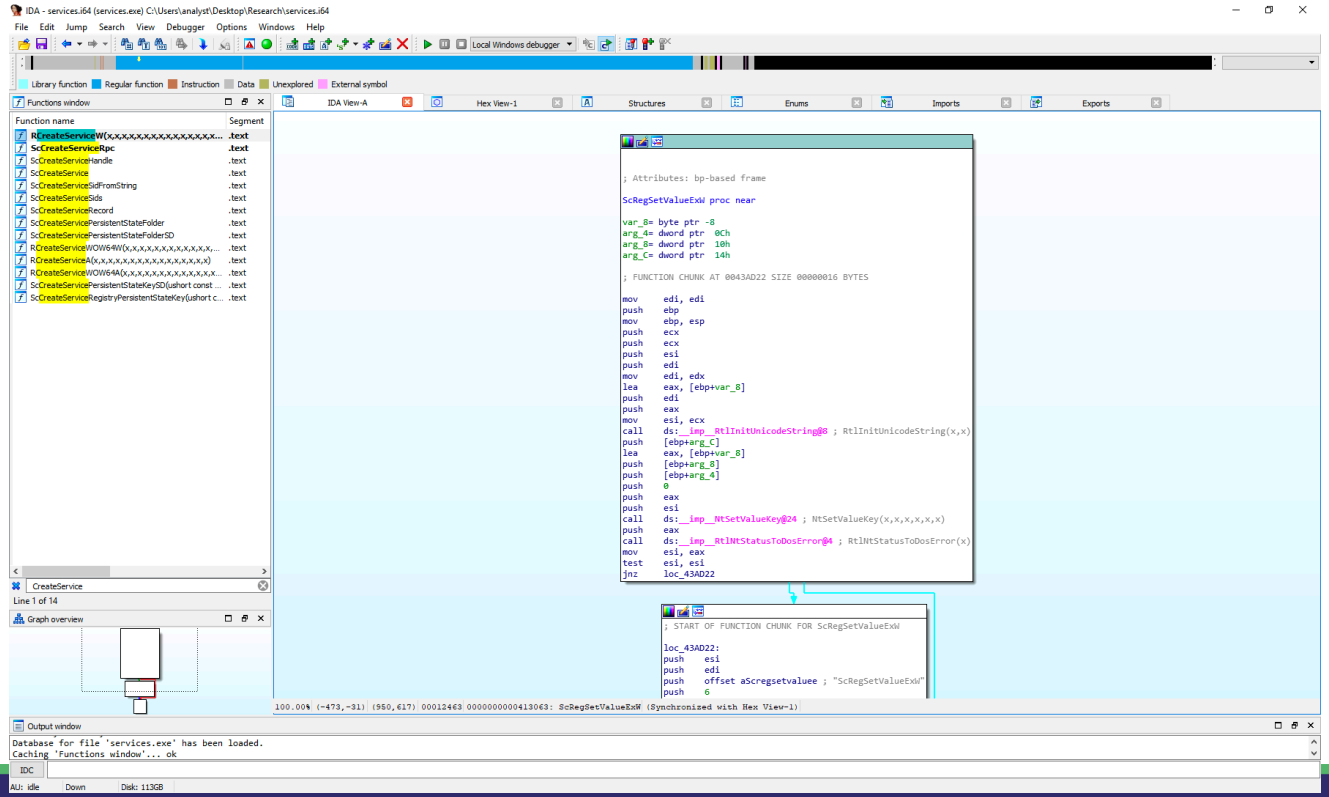
SERVICES.EXE – SCCreateService (Reg)



SERVICES.EXE - ScWriteStartName



SERVICES.EXE - ScRegSetValueExW



Event Tracing for Windows

How does the event log receive events?

- Event Tracing for Windows (ETW)
- Each event log receives events from a specific ETW provider
- An ETW provider is the source of events and has an installed manifest that we can retrieve.
- Note: not all ETW events are designed to be consumed by the event log so specialized tooling is required to capture these events which might be valuable from a detection perspective.
    - Some ETW events are in an unreadable format

What does ETW provide?

- Event Tracing for Windows (ETW) enables detection engineers to add context to current alerting structure.
  - Example: Windows Defender Advanced Threat Protection has "Miscellaneous" events which are derived of ETW events.
  - Example: ETW provider will log when a process is created in a suspended state.
    - Aka: Process Hollowing
- ETW aids in researching underlying technology, however there is not currently a way to utilize these logs at scale

Event Tracing for Windows - Overview

- Kernel-level tracing facility
  - Kernel or Application events
- Consumption of events in real time or from a log file
  - Intended use is to debug an application for performance issues
- Enabled/Disabled dynamically without requiring computer or application restarts.
- Consists of three components
  - Controllers - start/stop session & enable providers
  - Providers - responsible for generating the events
  - Consumers - responsible for consuming the events

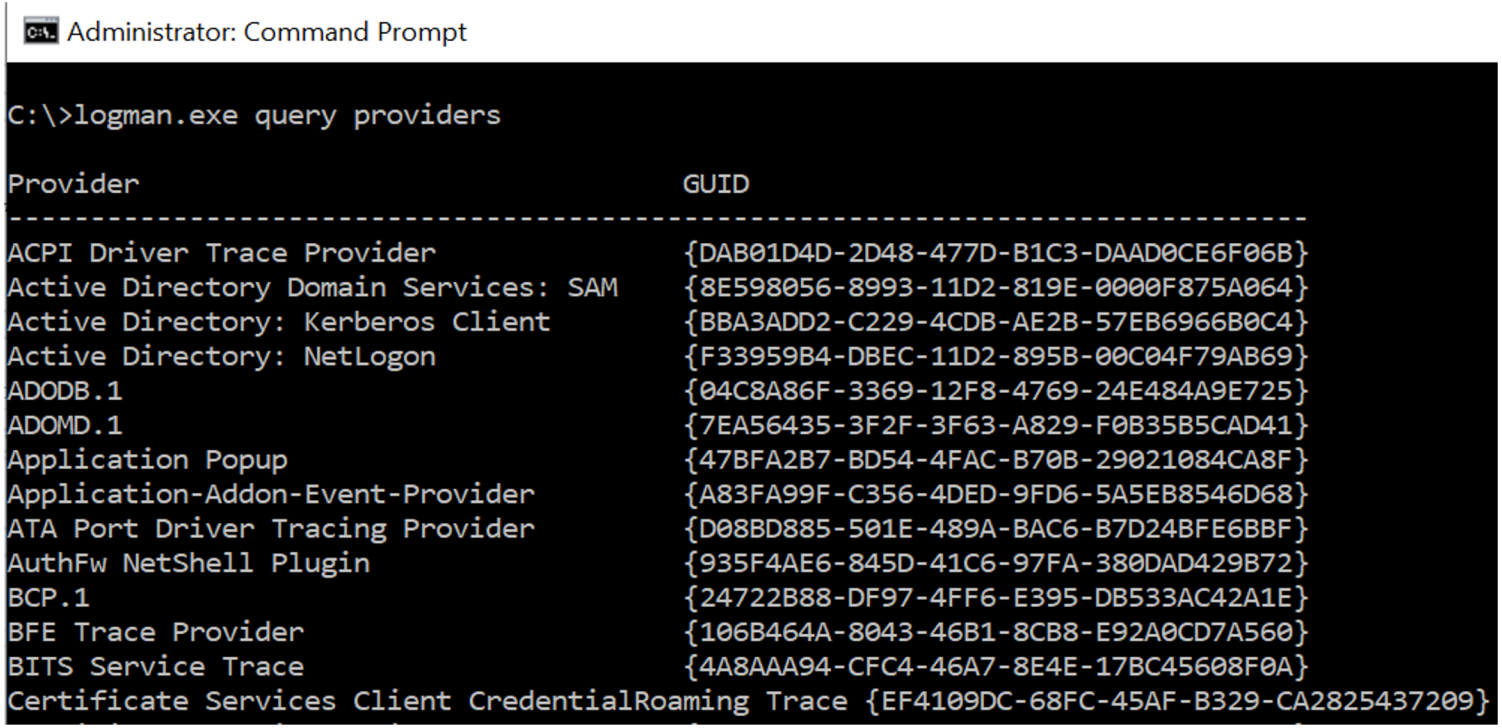
Source: About Event Tracing, https://docs.microsoft.com/en-us/windows/win32/etw/about-event-tracing

Event Tracing for Windows - Controllers

- Logman tool
  - Can create & manage trace sessions
  - Query Providers that are available
- Windows Performance System Tool
  - GUI based tool that provides the same data Logman provides plus more
- View the trace sessions on a given host:
  - logman query -ets
    - Data Collection Set (Trace Name)
    - Type (Trace)
    - Status (Running, Stopped)
- Trace Sessions subscribe to Providers

Source: About Event Tracing, https://docs.microsoft.com/en-us/windows/win32/etw/about-event-tracing

Registered Provider Enumeration



Investigate Supported ETW Provider Events

```
logman.exe query providers Microsoft-Windows-PowerShell
```

- List all supported keywords (used for filtering) and logging levels

```
wevtutil gp Microsoft-Windows-PowerShell /ge:true /gm:true
/f:xml > PowerShellEventSchema.xml
```

- Dump more detailed schema information

```
<event value="4104" version="1" opcode="15" channel="16" level="5" task="102"
keywords="0x8000000000000001" message="Creating Scriptblock text (%1 of %2):
%3
ScriptBlock ID: %4
Path: %5">
</event>
```

Investigate Supported ETW Provider Events

Built-in tools don't completely parse out ETW manifests. WEPExplorer can help.

Message	Fields
Computer Name $null or . resolve to LocalHost	
Resolving to default scheme http	
Remote shell name resolved to default Microsoft.PowerShell	
%3Context:%1User Data:%2	ContextInfo,UserData,Payload
%3Context:%1User Data:%2	ContextInfo,UserData,Payload
%3Context:%1User Data:%2	ContextInfo,UserData,Payload
%3Context:%1User Data:%2	ContextInfo,UserData,Payload
Creating Scriptblock text (%1 of %2):%3ScriptBlock ID: %4P...	MessageNumber,MessageTotal,ScriptBlockText,ScriptBlockId,Path

Investigate Supported ETW Provider Events

ETW events that don't specify an "Operational", "Admin", "Analytic" channel are not designed to be consumed/forwarded by the event log. Example:

Microsoft-Windows-LDAP-Client ETW Provider

- Debug and Analytic log sources are not logged to .EVTX (.ETL instead) and cannot be easily consumed/forwarded.

29	errors	Microsoft-Windows-LDAP-Client/Debug	Message
30	search	Microsoft-Windows-LDAP-Client/Debug	ScopeOfSearch,SearchFilter,DistinguishedName,AttributeList,ProcessId
31	performance	Microsoft-Windows-LDAP-Client/Debug	Message

Manually investigating registered ETW providers and their events can uncover some potentially interesting events from a security perspective!

Unsupported ETW Provider Events

What if you don’t have a **Channel** for the Provider?

Can you still consume events ? Yes….but...

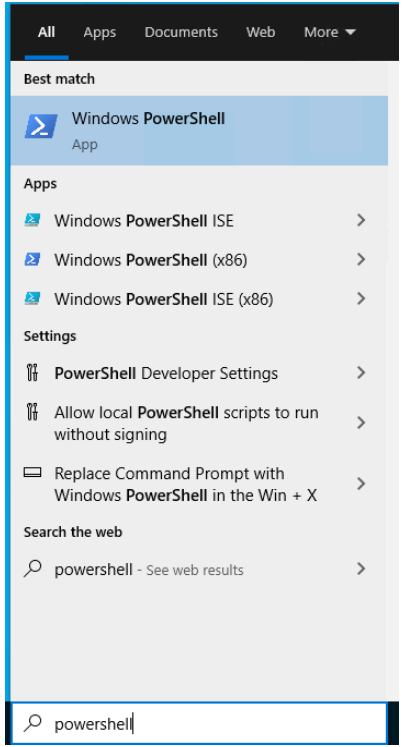
- Not real-time, manual creation of trace session with subscriber

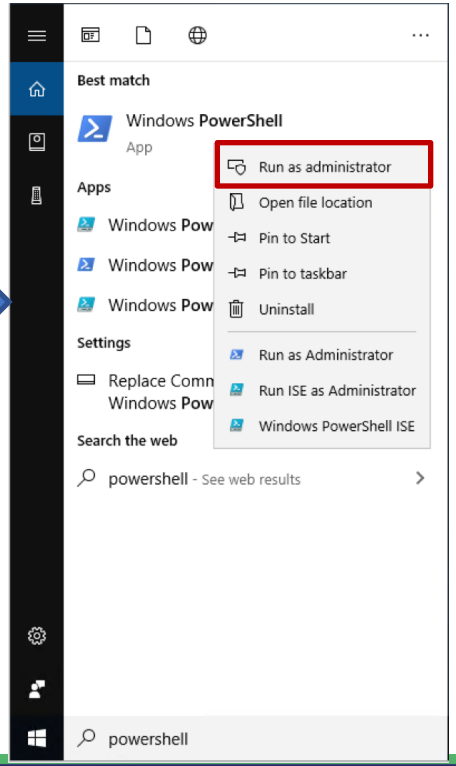
Example: **Microsoft-Windows-DotNETRuntime** Provider

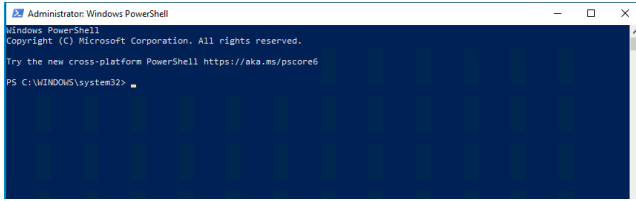
This solution could be useful for research but does not scale well...

Lab - Logman ETW

Starting Powerhsell – Logman ETW







Logman Query – Logman ETW

```
PS C:\Windows\system32> logman query -ets

Data Collector Set                      Type                          Status
-------------------------------------------------------------------------------
Circular Kernel Context Logger          Trace                         Running
AppModel                                Trace                         Running
SYSMON TRACE                            Trace                         Running
DiagLog                                 Trace                         Running
EventLog-Application                    Trace                         Running
EventLog-Microsoft-Windows-Sysmon-Operational Trace                         Running
EventLog-System                         Trace                         Running
LwtNetLog                               Trace                         Running
NtfsLog                                 Trace                         Running
ScreenOnPowerStudyTraceSession          Trace                         Running
TileStore                               Trace                         Running
UBPM                                    Trace                         Running
WdiContextLog                           Trace                         Running
WiFiSession                             Trace                         Running
umstartup                               Trace                         Running
UserNotPresentTraceSession              Trace                         Running
COM                                     Trace                         Running
Terminal-Services-LSM                   Trace                         Running
Terminal-Services-RCM                   Trace                         Running
WindowsUpdate_trace_log                 Trace                         Running
UserMgr                                 Trace                         Running
SHS-05222020-100819-7-5f                Trace                         Running
WDSC-05222020-100819-7-20               Trace                         Running
MpWppTracing-20200522-100819-00000003-ffffffff Trace                         Running
Diagtrack-Listener                      Trace                         Running
SysmonDnsEtwSession                     Trace                         Running
8696EAC4-1288-4288-A4EE-49EE431B0AD9    Trace                         Running

The command completed successfully.
```

Logman Query EventLog-System – Logman ETW

```
PS C:\Windows\system32> logman query "EventLog-System" -ets

Name:                 EventLog-System
Status:               Running
Root Path:            %systemdrive%\PerfLogs\Admin
Segment:              Off
Schedules:            On
Segment Max Size:     100 MB

Name:                 EventLog-System\EventLog-System
Type:                 Trace
Append:               Off
Circular:             Off
Overwrite:            Off
Buffer Size:          64
Buffers Lost:         0
Buffers Written:      146
Buffer Flush Timer:   1
Clock Type:           System
File Mode:            Real-time
```

Logman Query Providers – Logman ETW

```
PS C:\Windows\system32> logman query providers

Provider                                  GUID
-------------------------------------------------------------------------------
ACPI Driver Trace Provider                {DAB01D4D-2D48-477D-B1C3-DAAD0CE6F06B}
Active Directory Domain Services: SAM     {8E598056-8993-11D2-819E-0000F875A064}
Active Directory: Kerberos Client         {BBA3ADD2-C229-4CDB-AE2B-57EB6966B0C4}
Active Directory: NetLogon                {F33959B4-DBEC-11D2-895B-00C04F79AB69}
ADODB.1                                   {04C8A86F-3369-12F8-4769-24E484A9E725}
ADOMD.1                                   {7EA56435-3F2F-3F63-A829-F0B35B5CAD41}
Application Popup                         {47BFA2B7-BD54-4FAC-B70B-29021084CA8F}
Application-Addon-Event-Provider          {A83FA99F-C356-4DED-9FD6-5A5EB8546D68}
ATA Port Driver Tracing Provider          {D08BD885-501E-489A-BAC6-B7D24BFE6BBF}
AuthFw NetShell Plugin                    {935F4AE6-845D-41C6-97FA-380DAD429B72}
BCP.1                                     {24722B88-DF97-4FF6-E395-DB533AC42A1E}
BFE Trace Provider                        {106B464A-8043-46B1-8CB8-E92A0CD7A560}
BITS Service Trace                        {4A8AAA94-CFC4-46A7-8E4E-17BC45608F0A}
Certificate Services Client CredentialRoaming Trace {EF4109DC-68FC-45AF-B329-CA2825437209}
Certificate Services Client Trace         {F01B7774-7ED7-401E-8088-B576793D7841}
Circular Kernel Session Provider          {54DEA73A-ED1F-42A4-AF71-3E63D056F174}
Classpnp Driver Tracing Provider          {FA8DE7C4-ACDE-4443-9994-C4E2359A9EDB}
Critical Section Trace Provider           {3AC66736-CC59-4CFF-8115-8DF50E39816B}
DBNETLIB.1                                {BD568F20-FCCD-B948-054E-DB3421115D61}
Deduplication Tracing Provider            {5EBB59D1-4739-4E45-872D-B8703956D84B}
Disk Class Driver Tracing Provider        {945186BF-3DD6-4F3F-9C8E-9EDD3FC9D558}
Downlevel IPsec API                       {94335EB3-79EA-44D5-8EA9-306F49B3A041}
Downlevel IPsec NetShell Plugin           {E4FF10D8-8A88-4FC6-82C8-8C23E9462FE5}
Downlevel IPsec Policy Store              {94335EB3-79EA-44D5-8EA9-306F49B3A070}
Downlevel IPsec Service                   {94335EB3-79EA-44D5-8EA9-306F49B3A040}
EA IME API                                {E2A24A32-00DC-4025-9689-C108C01991C5}
```

Logman Query RPC Providers – Logman ETW

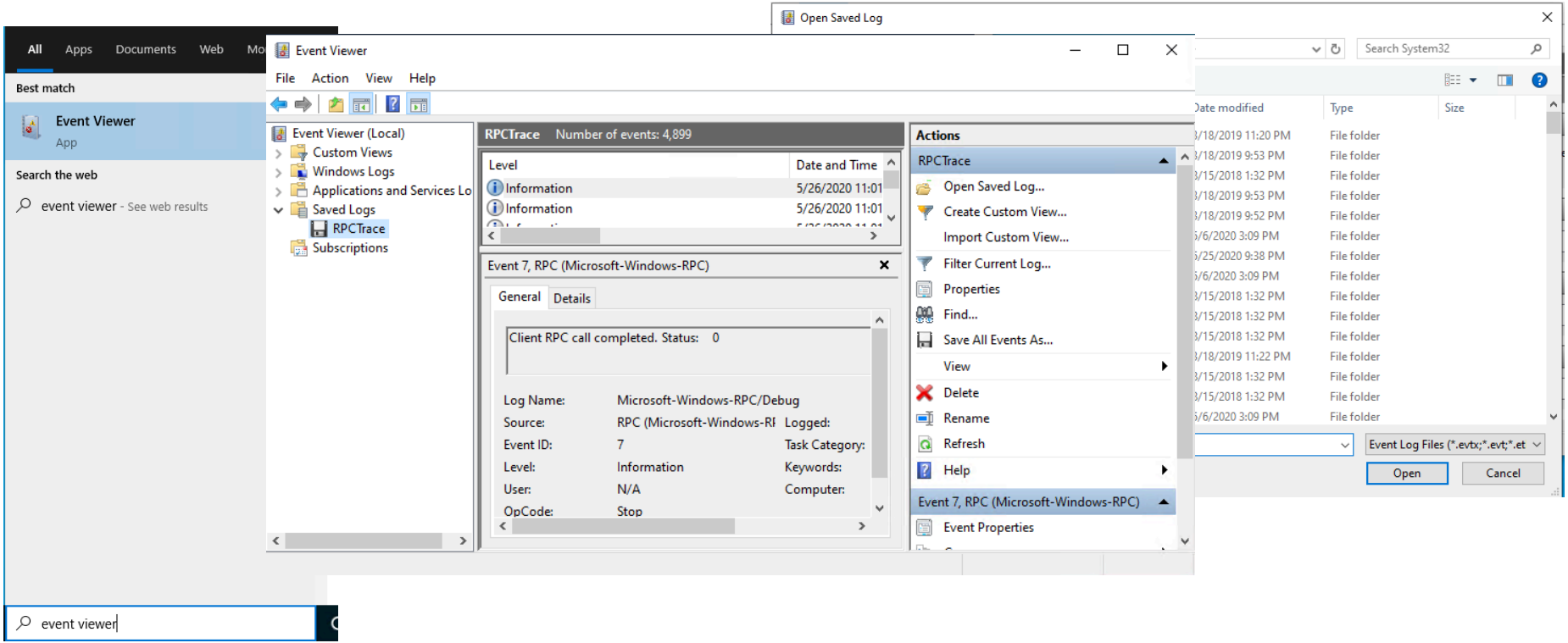
```
PS C:\Windows\system32> $ETW = logman query providers
PS C:\Windows\system32> $ETW | Where-Object { $_ -Like "*RPC*"}
Microsoft-Windows-RPC                       {6AD52B32-D609-4BE9-AE07-CE8DAE937E39}
Microsoft-Windows-RPC-Events                {F4AED7C7-A898-4627-B053-44A7CAA12FCD}
Microsoft-Windows-RPC-FirewallManager       {F997CD11-0FC9-4AB4-ACBA-BC742A4C0DD3}
Microsoft-Windows-RPC-Proxy-LBS             {272A979B-34B5-48EC-94F5-7225A59C85A0}
Microsoft-Windows-RPCSS                     {D8975F88-7DDB-4ED0-91BF-3ADF48C48E0C}
PS C:\Windows\system32>
```

RPC Trace – Logman ETW

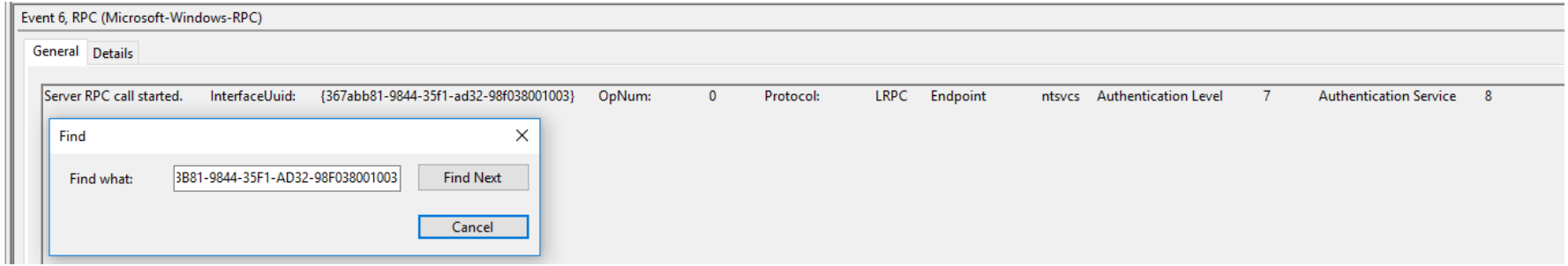
```
PS C:\WINDOWS\system32> tracerpt RPCTrace.etl -o RPCTrace.evtx -of EVTX

Input
---------------
File(s):
     RPCTrace.etl

100.00%

Output
---------------
DumpFile:           RPCTrace.evtx
```

```
PS C:\WINDOWS\system32> logman start RPCTrace -p Microsoft-Windows-RPC 0xffffffffffffffff win:Informational -ets
The command completed successfully.
PS C:\WINDOWS\system32> New-Service -Name "Test_Service" -BinaryPathName C:\Windows\System32\calc.exe

Status   Name               DisplayName
------   ----               -----------
Stopped  Test_Service       Test_Service


PS C:\WINDOWS\system32> logman stop RPCTrace -ets
The command completed successfully.
```

Uploading RPCTrace.evtx - Logman ETW



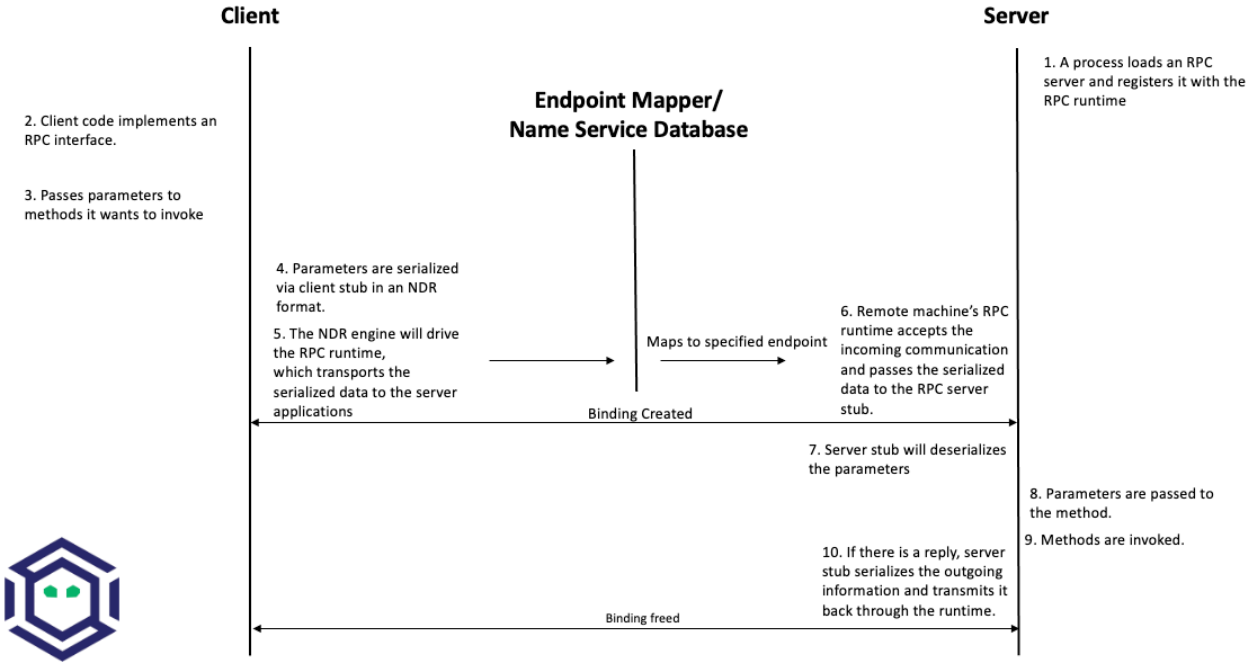
Looking for Service Creation – Logman ETW



Remote Procedure Call (RPC)

- Used for distributed client/server programs
- RPC Components:
  - Protocol
  - Interface
  - Method (Function)
  - Client/Server stubs
  - NDR Engine/Marshalling
  - RPC Run-Time
  - RPC endpoint mapper
  - Endpoint

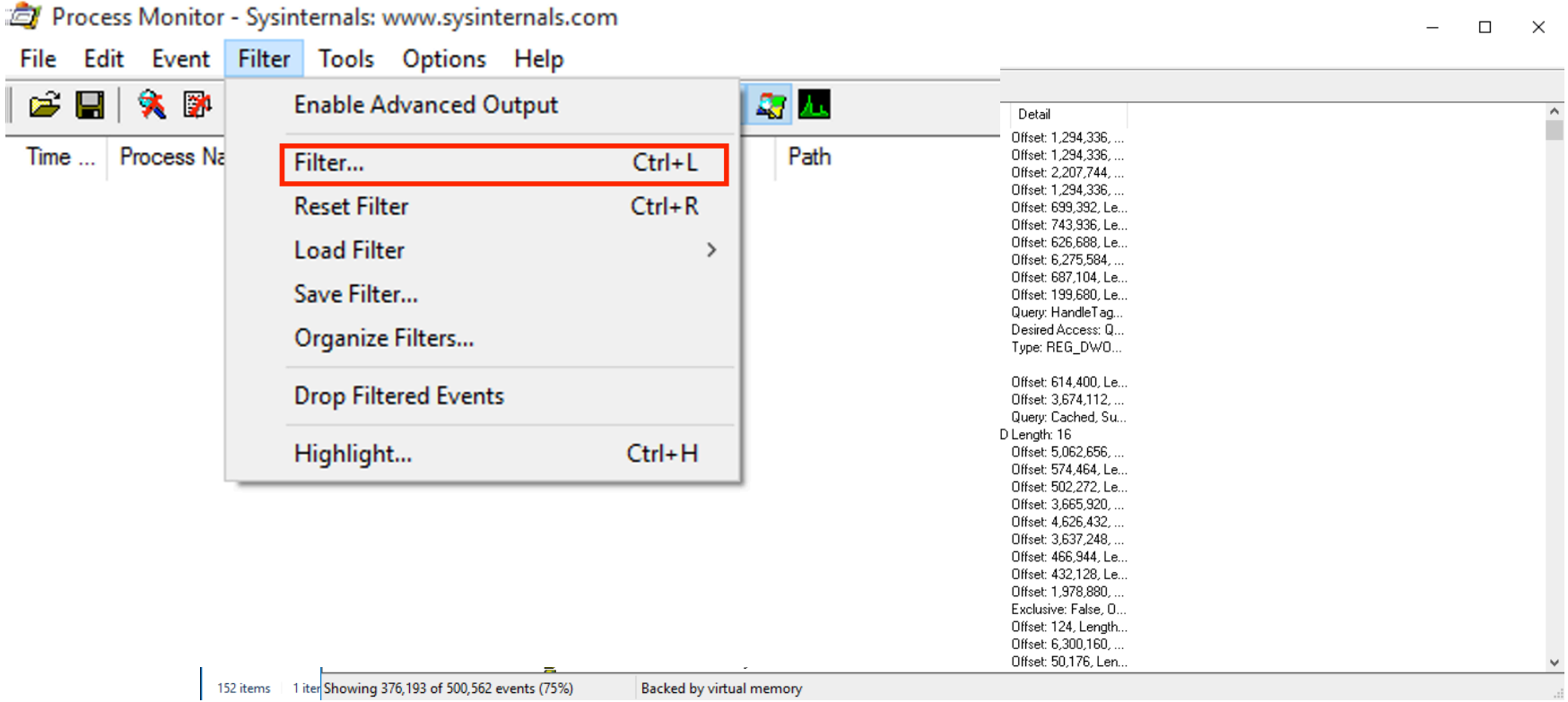
RPC Communication Example



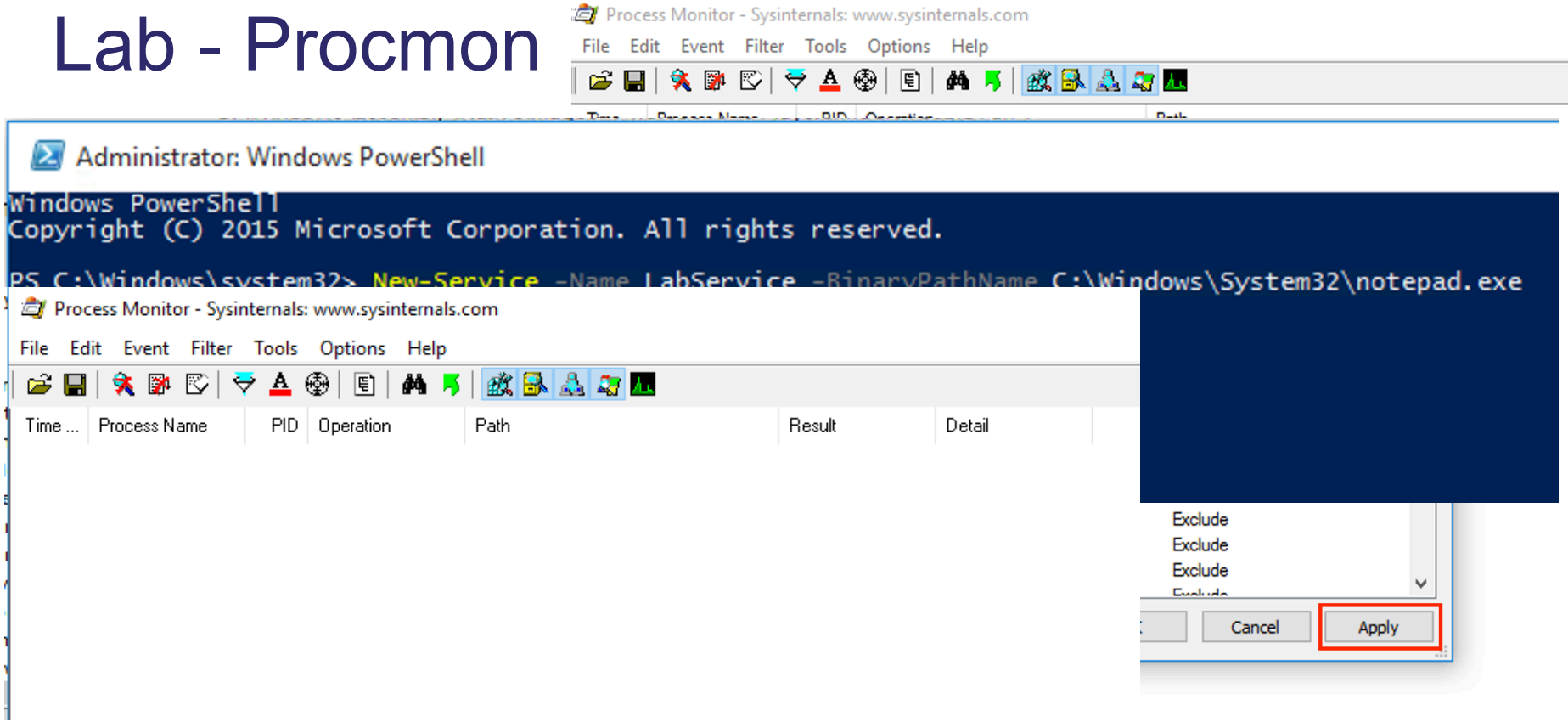
T1050 - Service Creation	
Tools	New-Service
Windows API	OpenSCManager/CreateService
RPC Interface/Named Pipe	367ABB81-9844-35F1-AD32-98F038001003 [MS-SCMR]
	\\PIPE\svcctl
Registry Service Database	HKLM\SYSTEM\CurrentControlSet\Services

Lab – Procmon

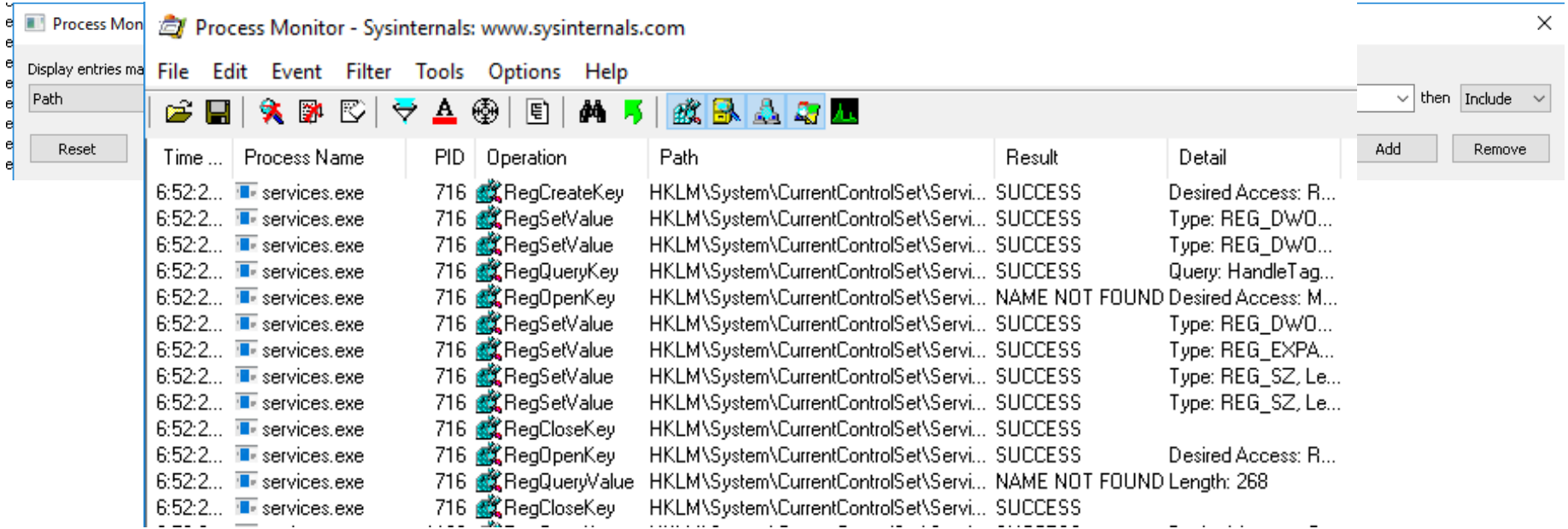
Lab - Procmon


Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help
Enable Advanced Output
Filter... Ctrl+L
Reset Filter Ctrl+R
Load Filter
Save Filter...
Organize Filters...
Drop Filtered Events
Highlight... Ctrl+H
Time ...
Process Na
Path
Detail
Offset: 1,294,336, ...
Offset: 1,294,336, ...
Offset: 2,207,744, ...
Offset: 1,294,336, ...
Offset: 699,392, Le...
Offset: 743,936, Le...
Offset: 626,688, Le...
Offset: 6,275,584, ...
Offset: 687,104, Le...
Offset: 199,680, Le...
Query: HandleTag...
Desired Access: Q...
Type: REG_DWO...
Offset: 614,400, Le...
Offset: 3,674,112, ...
Query: Cached, Su...
D Length: 16
Offset: 5,062,656, ...
Offset: 574,464, Le...
Offset: 502,272, Le...
Offset: 3,665,920, ...
Offset: 4,626,432, ...
Offset: 3,637,248, ...
Offset: 466,944, Le...
Offset: 432,128, Le...
Offset: 1,978,880, ...
Exclusive: False, O...
Offset: 124, Length...
Offset: 6,300,160, ...
Offset: 50,176, Len...
152 items
1 iter
Showing 376,193 of 500,562 events (75%)
Backed by virtual memory

Lab - Procmon


Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.
PS C:\Windows\system32> New-Service -Name LabService -BinaryPathName C:\Windows\System32\notepad.exe
Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help
Time ... Process Name PID Operation Path Result Detail
Exclude
Exclude
Exclude
Cancel
Apply

Lab - Procmon


Process Mon
Display entries ma
Path
Reset
Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help
then Include
Add Remove
Time ... Process Name PID Operation Path Result Detail
6:52:2... services.exe 716 RegCreateKey HKLM\System\CurrentControlSet\Servi... SUCCESS Desired Access: R...
6:52:2... services.exe 716 RegSetValue HKLM\System\CurrentControlSet\Servi... SUCCESS Type: REG_DWO...
6:52:2... services.exe 716 RegSetValue HKLM\System\CurrentControlSet\Servi... SUCCESS Type: REG_DWO...
6:52:2... services.exe 716 RegQueryKey HKLM\System\CurrentControlSet\Servi... SUCCESS Query: HandleTag...
6:52:2... services.exe 716 RegOpenKey HKLM\System\CurrentControlSet\Servi... NAME NOT FOUND Desired Access: M...
6:52:2... services.exe 716 RegSetValue HKLM\System\CurrentControlSet\Servi... SUCCESS Type: REG_DWO...
6:52:2... services.exe 716 RegSetValue HKLM\System\CurrentControlSet\Servi... SUCCESS Type: REG_EXPA...
6:52:2... services.exe 716 RegSetValue HKLM\System\CurrentControlSet\Servi... SUCCESS Type: REG_SZ, Le...
6:52:2... services.exe 716 RegSetValue HKLM\System\CurrentControlSet\Servi... SUCCESS Type: REG_SZ, Le...
6:52:2... services.exe 716 RegCloseKey HKLM\System\CurrentControlSet\Servi... SUCCESS
6:52:2... services.exe 716 RegOpenKey HKLM\System\CurrentControlSet\Servi... SUCCESS Desired Access: R...
6:52:2... services.exe 716 RegQueryValue HKLM\System\CurrentControlSet\Servi... NAME NOT FOUND Length: 268
6:52:2... services.exe 716 RegCloseKey HKLM\System\CurrentControlSet\Servi... SUCCESS

Abstraction Take Aways

- The best place for detection is dependent on your monitoring posture
- Don’t rely on a single detection/analytic
- Always strive to understand as much about the attack as possible when developing detection logic
- Take the easy wins when possible, but keep the full scope in mind
- There likely isn’t one answer that covers everything!

Demo – Bypassing Service Auditing

T1050 - Service Creation									
Tools	sc.exe	Powershell New-Service	Invoke-PSExec	CSExec	SharpSC	PSExec	reg.exe	regedit.exe	PS Registry Edit
Windows API	OpenSCManager/CreateService						RegKeyCreateEx/RegSetValueEx		
RPC Interface/Named Pipe	367ABB81-9844-35F1-AD32-98F038001003 [MS-SCMR]						338CD001-2244-31F1-AAAA-900038001003 [MS-RRP]		
	\\PIPE\svcctl						\\PIPE\winreg		
RCP Method	ROpenSCManager/RCreateService						BaseRegCreateKey		
Registry Service Database	HKLM\SYSTEM\CurrentControlSet\Services								

Further Research

- ~~PowerShell New-Service~~
- sc.exe /create
  - An alternate basic service creation utility
- Impacket
  - Can create a service through direct RPC calls
- reg.exe
  - Direct Manipulation of the Services Database
- PSExec
  - A popular administrative tool to execute code **remotely**
- Try everything on a remote system and see what is different!