



Abstraction Workshop



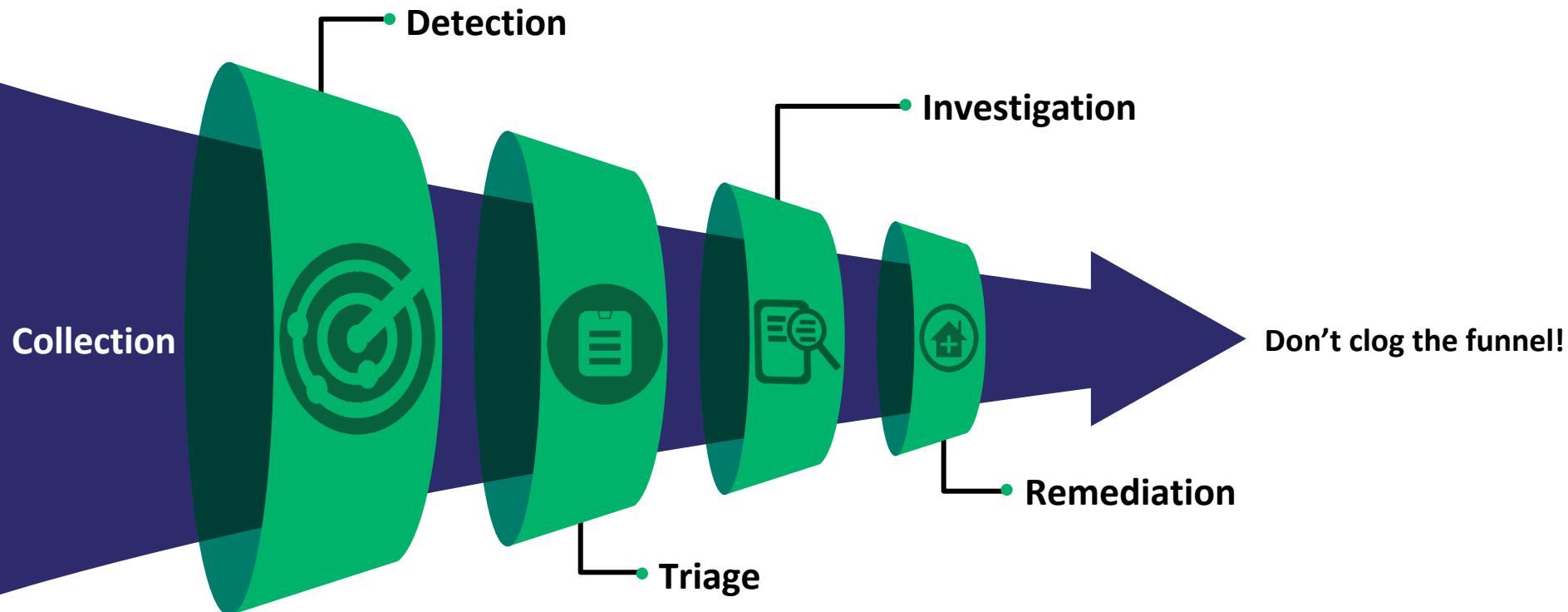
S P E C T E R O P S

Lab Infrastructure

Labs:

<https://jaredcatkinson.github.io/abstraction-workshop/>

Funnel of Fidelity

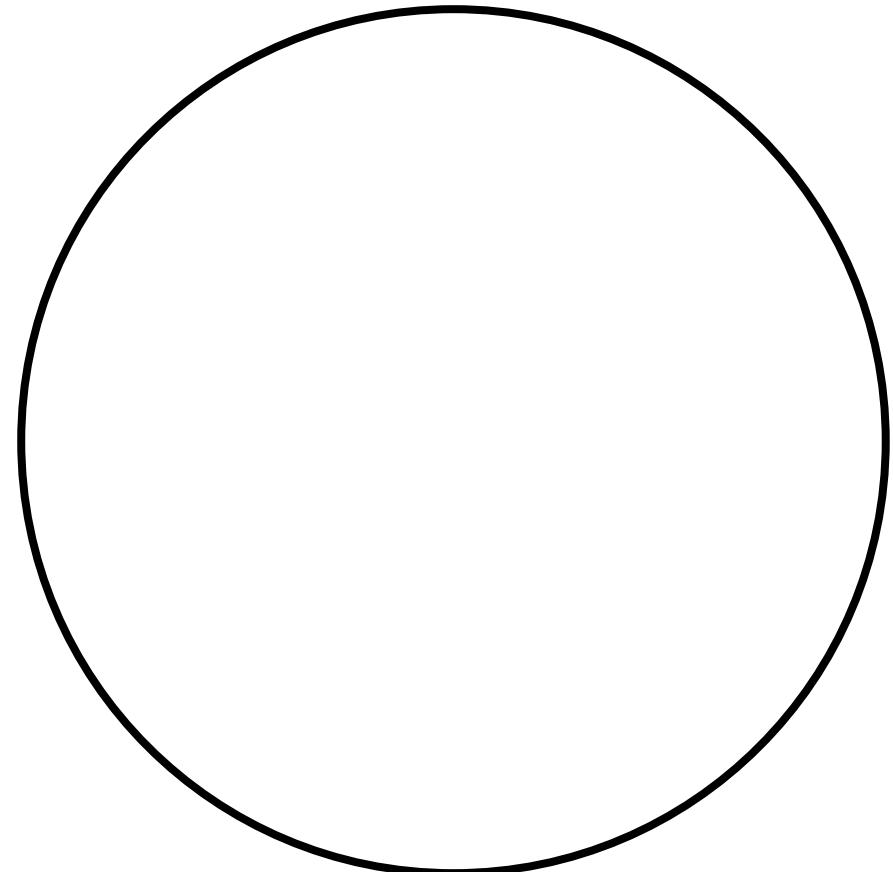


Capability Abstraction

- 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

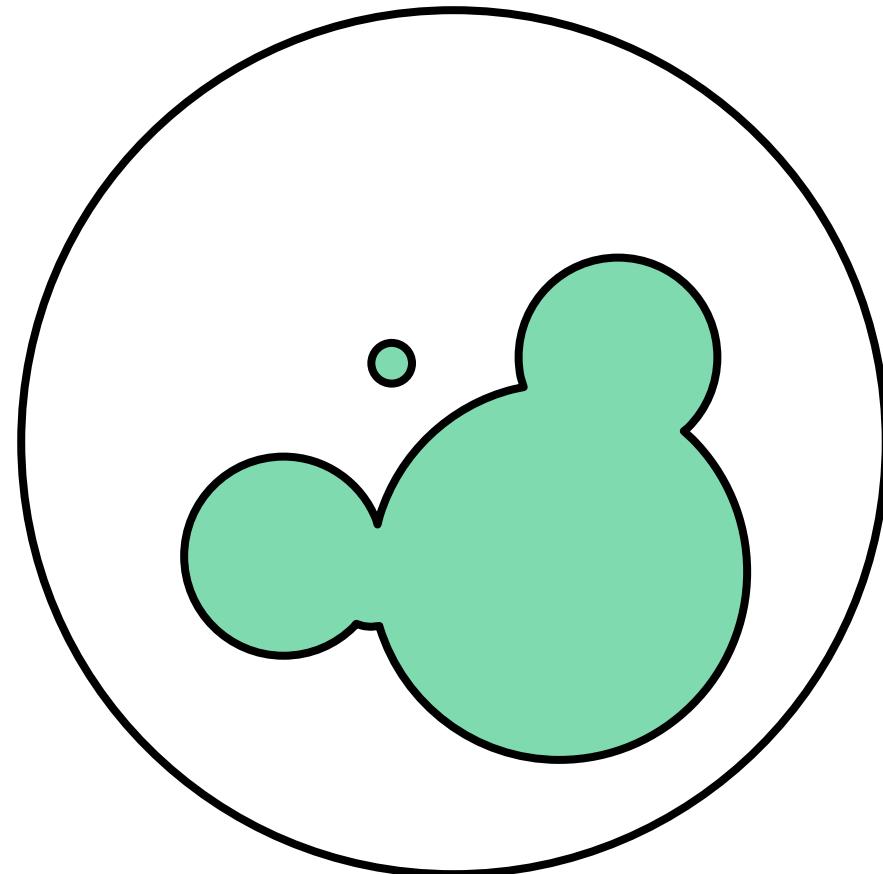
Population

- 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



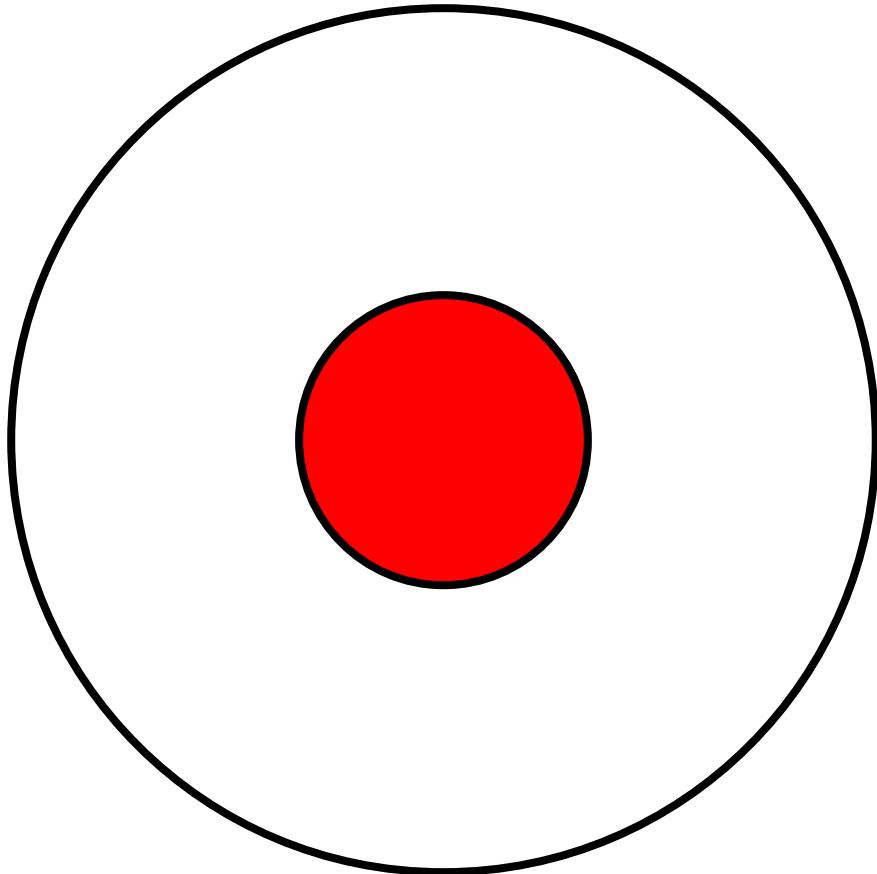
Classification Rule

- 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



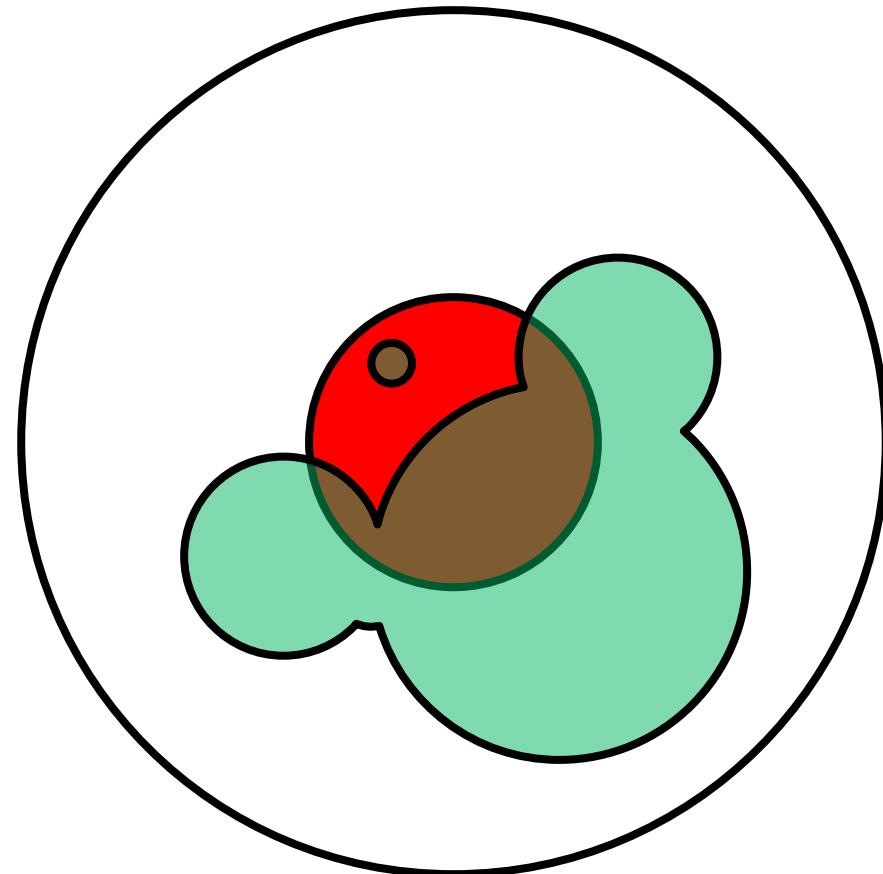
Condition Positive

- 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



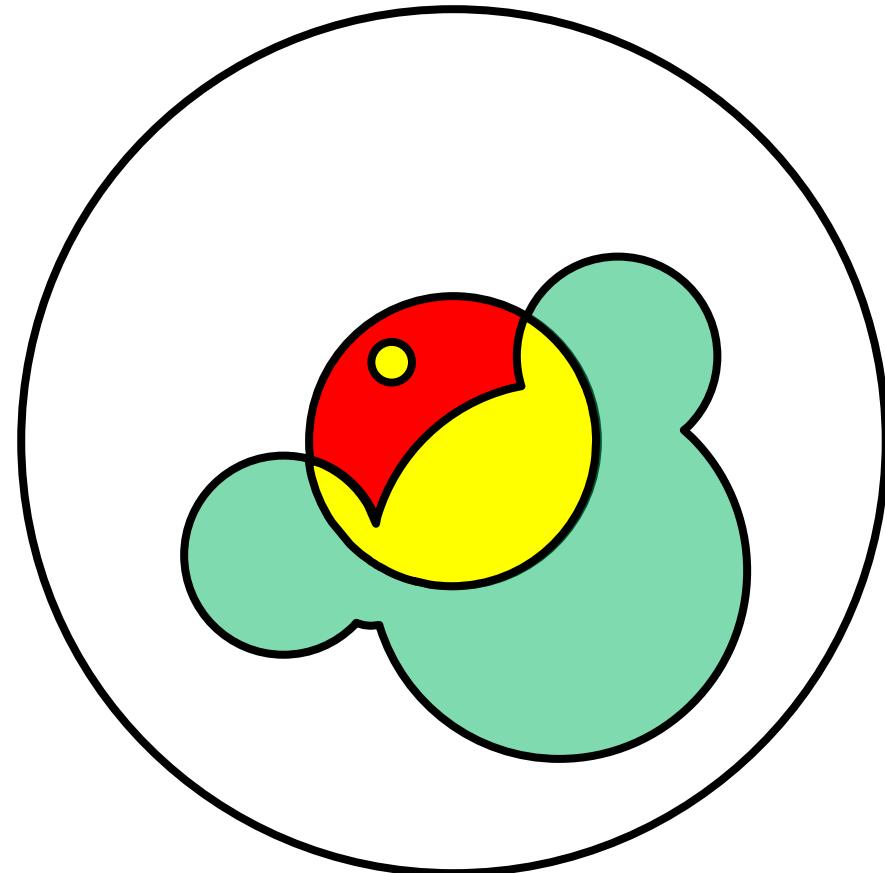
Condition Positive

- 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



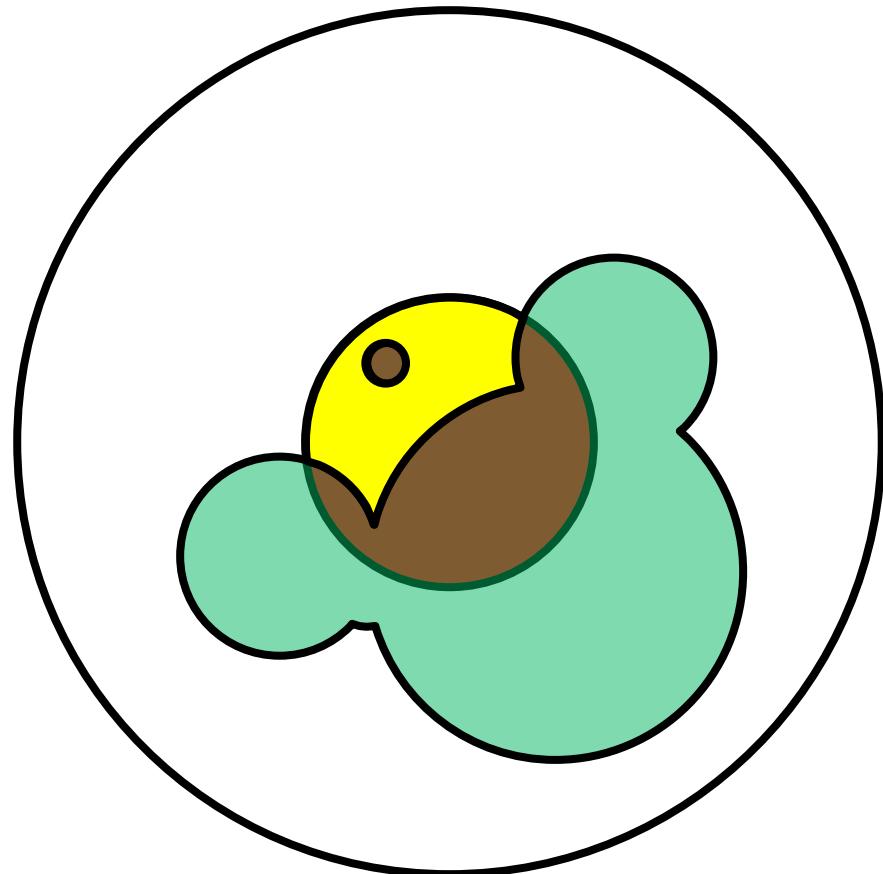
True Positives (TP)

- 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



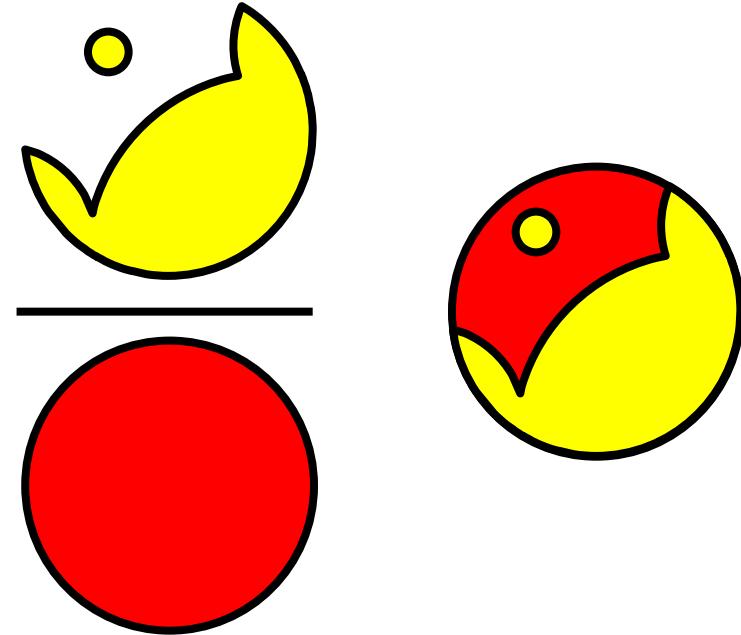
Sensitivity

- 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?

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

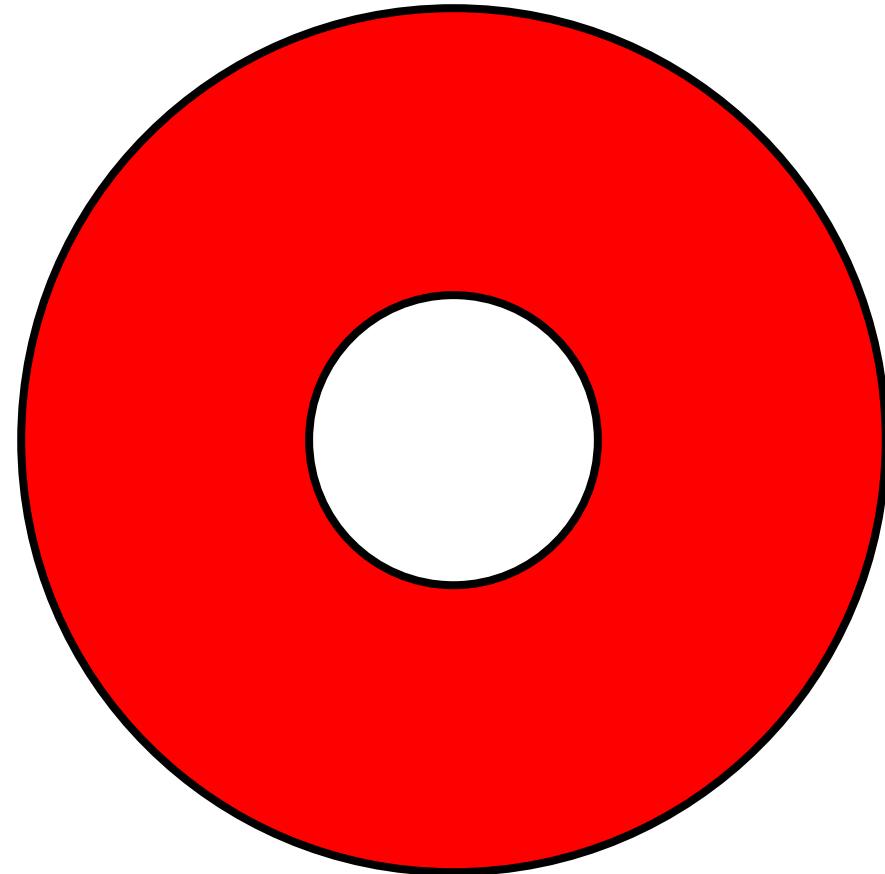
$$= \frac{\text{number of true positives}}{\text{total number of sick individuals in population}}$$

= probability of a positive test given that the patient has the disease



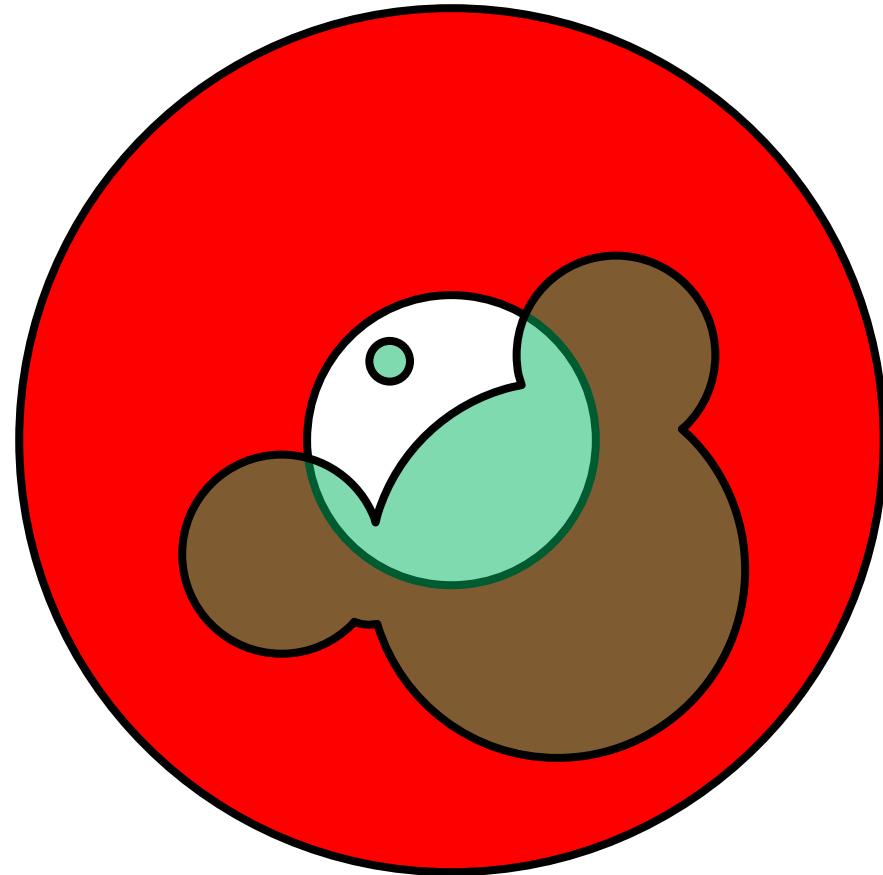
Condition Negative

- 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



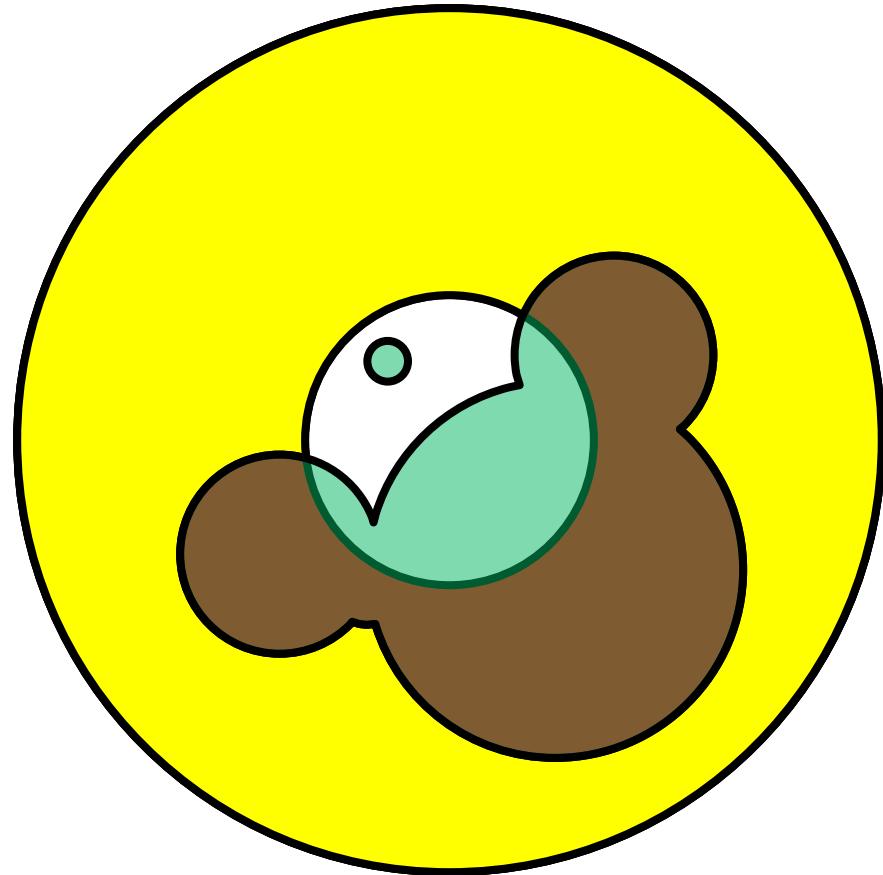
Condition Negative

- 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



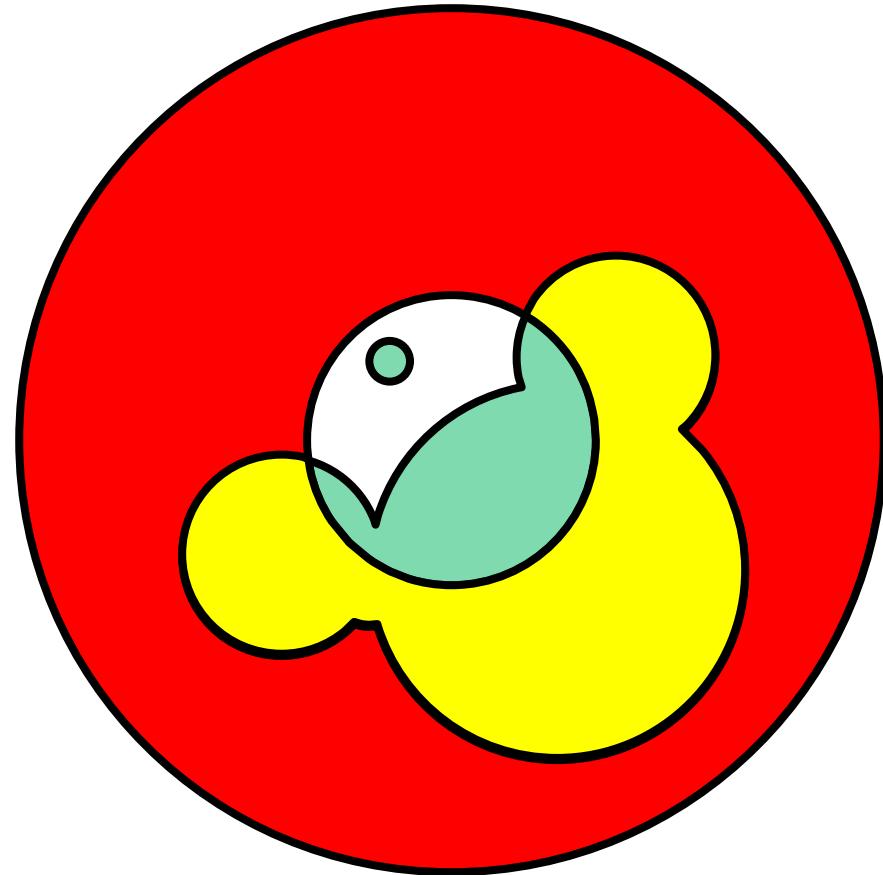
True Negatives (TN)

- 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



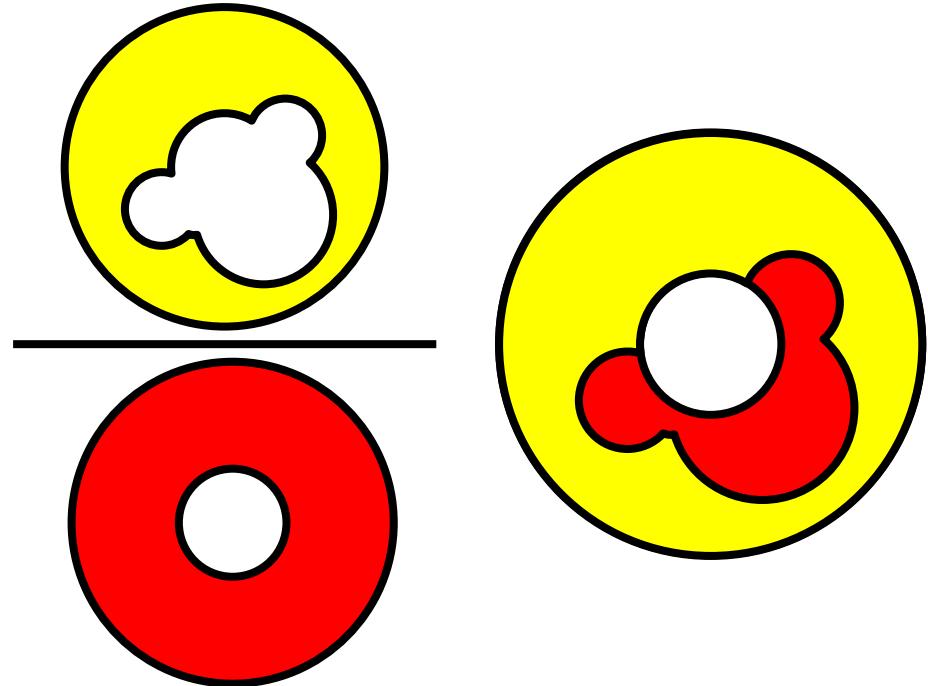
Specificity

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

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

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

= 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

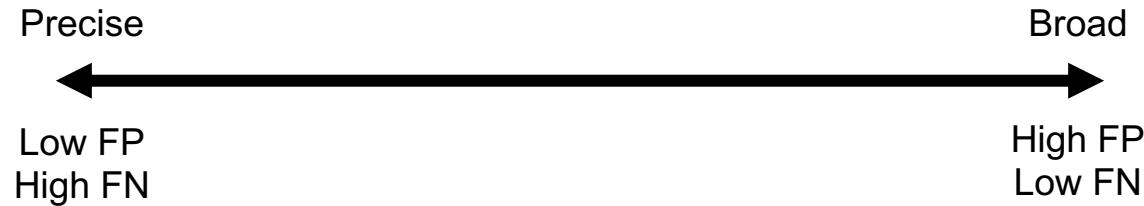
1 contributor

Executable File | 38 lines (38 sloc) | 1.12 KB

Raw Blame History

```
action: global
title: APT29 Google Update Service Install
id: c069f460-2b87-4010-8dcf-e45bab362624
description: This method detects malicious services mentioned in APT29 report by FireEye. The legitimate path for the Google update service names and executable locations used by APT29 are specific enough to be detected in log files.
references:
tags:
- attack.persistence
- attack.g0016
- attack.t1050
date: 2017/11/01
author: Thomas Patzke
logsource:
product: windows
service: system
detection:
service_install:
EventID: 7045
ServiceName: 'Google Update'
timeframe: 5m
condition: service_install | near process
falsepositives:
- Unknown
level: high
---
logsource:
category: process_creation
product: windows
detection:
process:
Image:
- 'C:\Program Files(x86)\Google\GoogleService.exe'
- 'C:\Program Files(x86)\Google\GoogleUpdate.exe'
fields:
- ComputerName
- User
- 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

```
1 title: Turla PNG Dropper Service
2 id: 1228f8e2-7e79-4dea-b0ad-c91fid5016c1
3 description: This method detects malicious services mentioned in Turla PNG dropper report by NCC Group in November 2018
4 references:
5   - https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2018/november/turla-png-dropper-is-back/
6 author: Florian Roth
7 date: 2018/11/23
8 tags:
9   - attack.persistence
10  - attack.g0010
11  - attack.t1050
12 logsource:
13   product: windows
14   service: system
15 detection:
16   selection:
17     EventID: 7045
18     ServiceName: 'WerFaultSvc'
19     condition: selection
20 falsepositives:
21   - unlikely
22 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

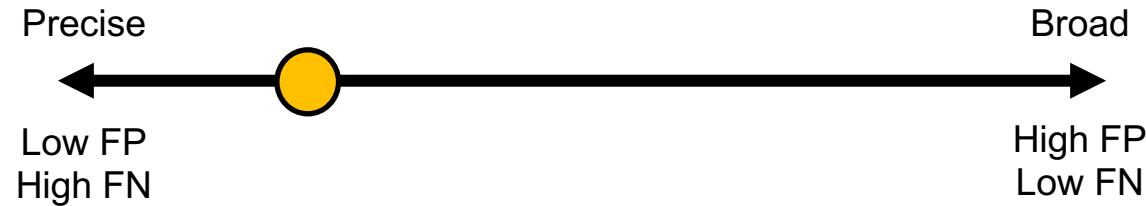
0 contributors

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 rules from 'apt' folder in respective folders 03ecb3b on Feb 1

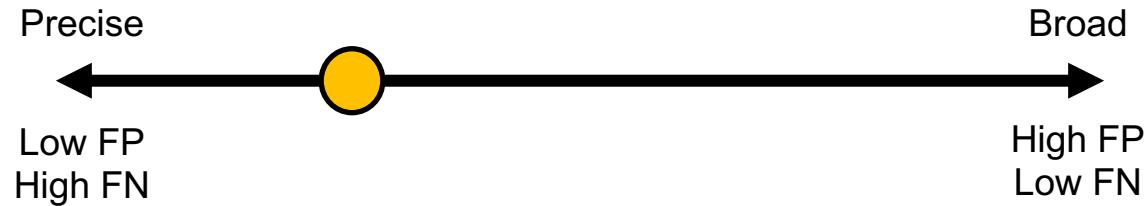
0 contributors

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       - 'srsservice'
20       - 'ipvpn'
21       - 'hkmvsc'
22   condition: selection
23 falsepositives:
24   - Unknown
25 level: high
```

Detection Spectrum



Specific Service Name

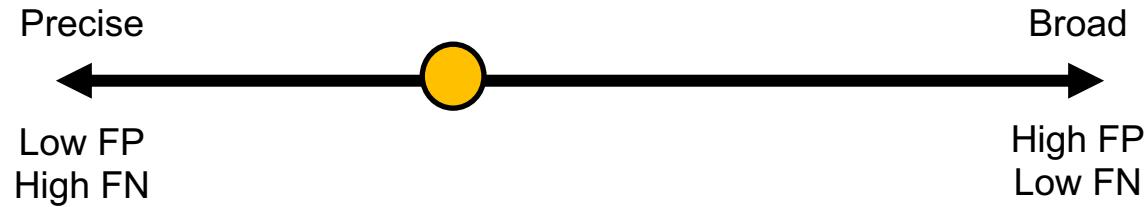
Tree: 48d95f027c ▾ [sigma / rules / windows / builtin / win_mal_service_installs.yml](#) Find file Copy path

 thomaspatzke Fixes 01d6c3b on Feb 16
2 contributors  

29 lines (29 sloc) 869 Bytes Raw Blame History

```
1 title: Malicious Service Installations
2 id: 5a105d34-05fc-401e-8553-272b45c1522d
3 description: Detects known malicious service installs that only appear in cases of lateral movement, credential dumping and oth
4 author: Florian Roth, Daniil Yugoslavskiy, oscd.community (update)
5 date: 2017/03/27
6 modified: 2019/11/01
7 tags:
8   - attack.persistence
9   - attack.privilege_escalation
10  - attack.t1003
11  - attack.t1035
12  - attack.t1050
13  - car.2013-09-005
14 logsources:
15   product: windows
16   service: system
17 detection:
18   selection:
19     EventID: 7045
20     malsvc_paexec:
21       ServiceFileName|contains: '\PAExec'
22     malsvc_wannacry:
23       ServiceName: 'msseccsvc2.0'
24     malsvc_persistence:
25       ServiceFileName|contains: 'net user'
26     condition: selection and 1 of malsvc_*
27 falsepositives:
28   - Penetration testing
29 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

thomasatzke Rule fixes 373424f on Feb 20

2 contributors

28 lines (28 sloc) | 846 Bytes

Raw Blame History

```
1 title: New Service Creation
2 id: 7fe71fc9-de3b-432a-8d57-8c809efc10ab
3 status: experimental
4 description: Detects creation if a new service
5 author: Timur Zinniatullin, Daniil Yugoslavskiy, oscd.community
6 date: 2019/10/21
7 modified: 2019/11/04
8 tags:
9   - attack.persistence
10  - attack.privilege_escalation
11  - attack.t1050
12 references:
13   - https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1050/T1050.yaml
14 logsource:
15   category: process_creation
16   product: windows
17 detection:
18   selection:
19     - Image|endswith: '\sc.exe'
20     - CommandLine|contains|all:
21       - 'create'
22       - 'binpath'
23     - Image|endswith: '\powershell.exe'
24     - CommandLine|contains: 'new-service'
25   condition: selection
26 falsepositives:
27   - Legitimate administrator or user creates a service for legitimate reason
28 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: 66bfef30-22a5-4fc4-ad44-8d81e60922ae
3 description: Detects rare service installs that only appear a few times per time frame and could reveal password dumpers, backdoors, or other malicious activity.
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.

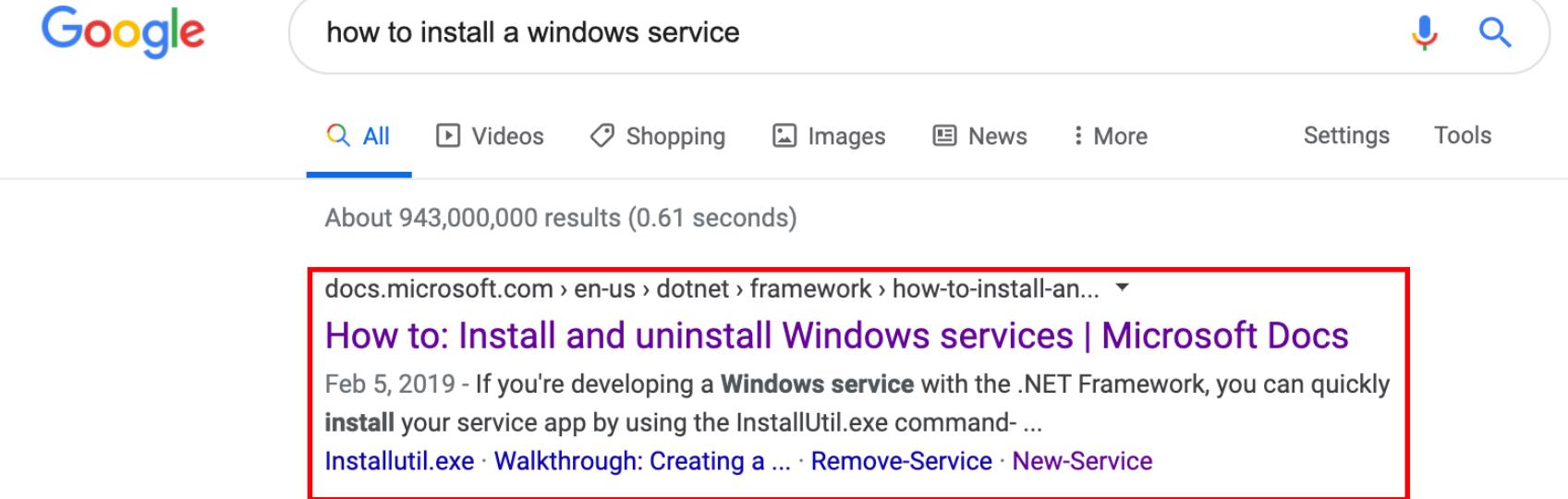
Questions

- 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 search results for "how to install a windows service". The search bar shows the query. Below it, a navigation bar includes All (selected), Videos, Shopping, Images, News, More, Settings, and Tools. A message indicates about 943,000,000 results found in 0.61 seconds. The top result is a Microsoft Docs page titled "How to: Install and uninstall Windows services | Microsoft Docs", dated Feb 5, 2019. The page content discusses using InstallUtil.exe to install a Windows service. A red box highlights the title and snippet of this result.

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

thomasatzke Rule fixes 373424f on Feb 20

2 contributors

28 lines (28 sloc) | 846 Bytes

Raw Blame History

```
1 title: New Service Creation
2 id: 7fe71fc9-de3b-432a-8d57-8c809efc10ab
3 status: experimental
4 description: Detects creation if a new service
5 author: Timur Zinniatullin, Daniil Yugoslavskiy, oscd.community
6 date: 2019/10/21
7 modified: 2019/11/04
8 tags:
9   - attack.persistence
10  - attack.privilege_escalation
11  - attack.t1050
12 references:
13   - https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1050/T1050.yaml
14 logsource:
15   category: process_creation
16   product: windows
17 detection:
18   selection:
19     - Image|endswith: '\sc.exe'
20     - CommandLine|contains|all:
21       - 'create'
22       - 'binpath'
23     - Image|endswith: '\powershell.exe'
24     - CommandLine|contains: 'new-service'
25   condition: selection
26 falsepositives:
27   - Legitimate administrator or user creates a service for legitimate reason
28 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

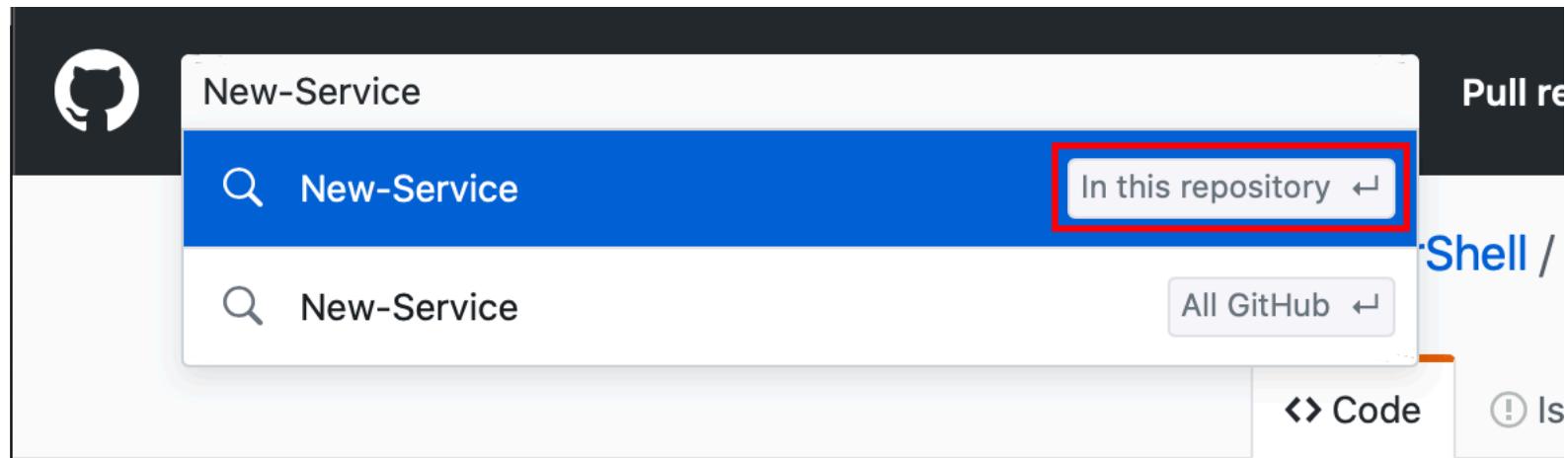
- Are there managed (ex. .NET) APIs that we should consider?
- What unmanaged function(s) is/are used by the tool(s) that we previously identified?
- What DLL(s) export these functions?
- Are there alternative functions that can be called?
- Are there any native, undocumented, or underlying functions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

Functions

The screenshot shows a GitHub repository page for 'PowerShell / PowerShell'. The search bar at the top is highlighted with a red box. The repository has 2,349 issues, 97 pull requests, 14 projects, and 3k forks. It has 7,936 commits, 2 branches, 0 packages, 79 releases, and 309 contributors. The latest commit was yesterday. The commit list includes changes for '.dependabot', '.devcontainer', '.github', '.poshchan', '.vscode', '.vsts-ci', 'CHANGELOG', 'assets', and 'demos'.

Commit	Description	Time Ago
.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



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 =
1998         "https://go.microsoft.com/fwlink/?LinkID=2096905")]
1999     ...
2000
2001     WriteNonTerminatingError(StartupType.ToString(), "New-Service",
2002         Name,
2003         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



openscmanagerw



All

Maps

Videos

Shopping

News

More

Settings

Tools

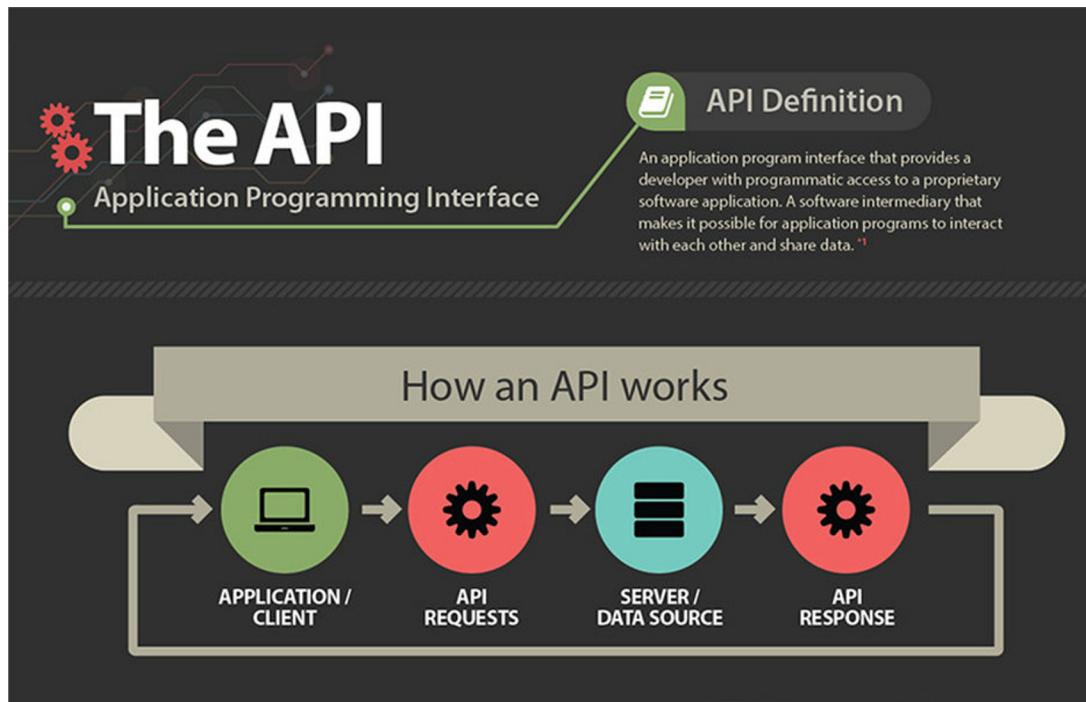
About 80,300 results (0.36 seconds)

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)



Windows API

- User-System programming 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

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

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

The screenshot shows a Microsoft Docs page for the Windows Dev Center. The top navigation bar includes links for Microsoft, Docs, Documentation (underlined), Learn, Q&A, and Code Samples. Below the navigation is a search bar. The main navigation menu includes Windows Dev Center, Explore, Platforms, Docs, Downloads, Samples, Support, and Dashboard. The breadcrumb trail shows Windows > Apps > Win32 > API > Winsvc.h > OpenSCManagerW function. On the left, there's a sidebar with a filter titled "Filter by title" and a list of related functions: OpenSCManagerA function, OpenSCManagerW function (which is selected and highlighted in grey), OpenServiceA function, OpenServiceW function, QUERY_SERVICE_CONFIGA structure, QUERY_SERVICE_CONFIGW structure, QUERY_SERVICE_LOCK_STATUSA structure, QUERY_SERVICE_LOCK_STATUSW structure, QueryServiceConfig2A function, QueryServiceConfig2W function, QueryServiceConfigA function, QueryServiceConfigW function, and QueryServiceDynamicInformation function.

OpenSCManagerW function

12/05/2018 • 2 minutes to read

Establishes a connection to the service control manager on the specified computer and opens the specified service control manager database.

Syntax

```
C++  
SC_HANDLE OpenSCManagerW(  
    LPCWSTR lpMachineName,  
    LPCWSTR lpDatabaseName,  
    DWORD   dwDesiredAccess  
) ;
```

Function Overview – Function Name

The screenshot shows a Microsoft Docs page for the Windows Dev Center. The URL is <https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-openscmanagerw>. The page title is "OpenSCManagerW function". A sidebar on the left lists various Windows API functions, with "OpenSCManagerW function" highlighted. The main content area includes the function's syntax in C++:

```
C++  
SC_HANDLE OpenSCManagerW(  
    LPCWSTR lpMachineName,  
    LPCWSTR lpDatabaseName,  
    DWORD    dwDesiredAccess  
) ;
```

Function Overview – Output Type

The screenshot shows a Microsoft Windows Dev Center page. At the top, there's a navigation bar with links for Microsoft Docs, Documentation (underlined), Learn, Q&A, and Code Samples. Below that is a secondary navigation bar for Windows Dev Center with links for Explore, Platforms, Docs, Downloads, Samples, Support, and Dashboard. The main content area has a breadcrumb trail: Windows / Apps / Win32 / API / Winsvc.h / OpenSCManagerW function. On the left, there's a sidebar with a 'Filter by title' input field containing 'OpenSCManagerA function' and a list of other functions: OpenSCManagerW function (highlighted in grey), OpenServiceA function, OpenServiceW function, QUERY_SERVICE_CONFIGA structure, QUERY_SERVICE_CONFIGW structure, QUERY_SERVICE_LOCK_STATUSA structure, QUERY_SERVICE_LOCK_STATUSW structure, QueryServiceConfig2A function, QueryServiceConfig2W function, QueryServiceConfigA function, QueryServiceConfigW function, and QueryServiceDynamicInformation function.

OpenSCManagerW function

12/05/2018 • 2 minutes to read

Establishes a connection to the service control manager on the specified computer and opens the specified service control manager database.

Syntax

```
C++  
SC_HANDLE OpenSCManagerW(  
    LPCWSTR lpMachineName,  
    LPCWSTR lpDatabaseName,  
    DWORD    dwDesiredAccess  
) ;
```

Copy

Function Overview – Parameters

The screenshot shows a Microsoft Docs page for the Windows Dev Center. The URL is <https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-openscmanagerw>. The page title is "OpenSCManagerW function". The content includes a brief description: "Establishes a connection to the service control manager on the specified computer and opens the specified service control manager database." Below this is a "Syntax" section with C++ code:

```
C++  
SC_HANDLE OpenSCManagerW(  
    _In_opt_ LPCWSTR lpMachineName,  
    _In_opt_ LPCWSTR lpDatabaseName,  
    _In_        DWORD   dwDesiredAccess  
) ;
```

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.

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

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

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

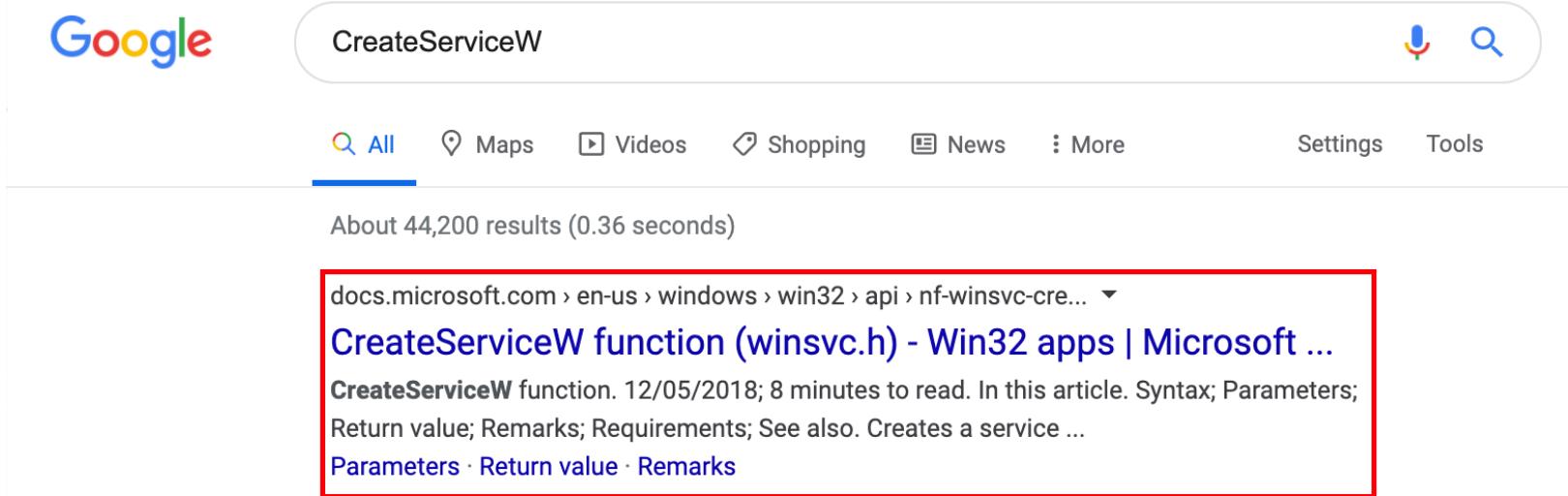
Requirements

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



A screenshot of a Google search results page. The search query "CreateServiceW" is entered in the search bar. Below the search bar, there are navigation links for All, Maps, Videos, Shopping, News, More, Settings, and Tools. A status message indicates "About 44,200 results (0.36 seconds)". The top result is a link to the Microsoft documentation for the `CreateServiceW` function, which is highlighted with a red box. The link text is "CreateServiceW function (winsvc.h) - Win32 apps | Microsoft ...". Below the link, a snippet of the documentation text is visible, mentioning the function's purpose and some parameters. There are also links for "Parameters", "Return value", and "Remarks".

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
);
```

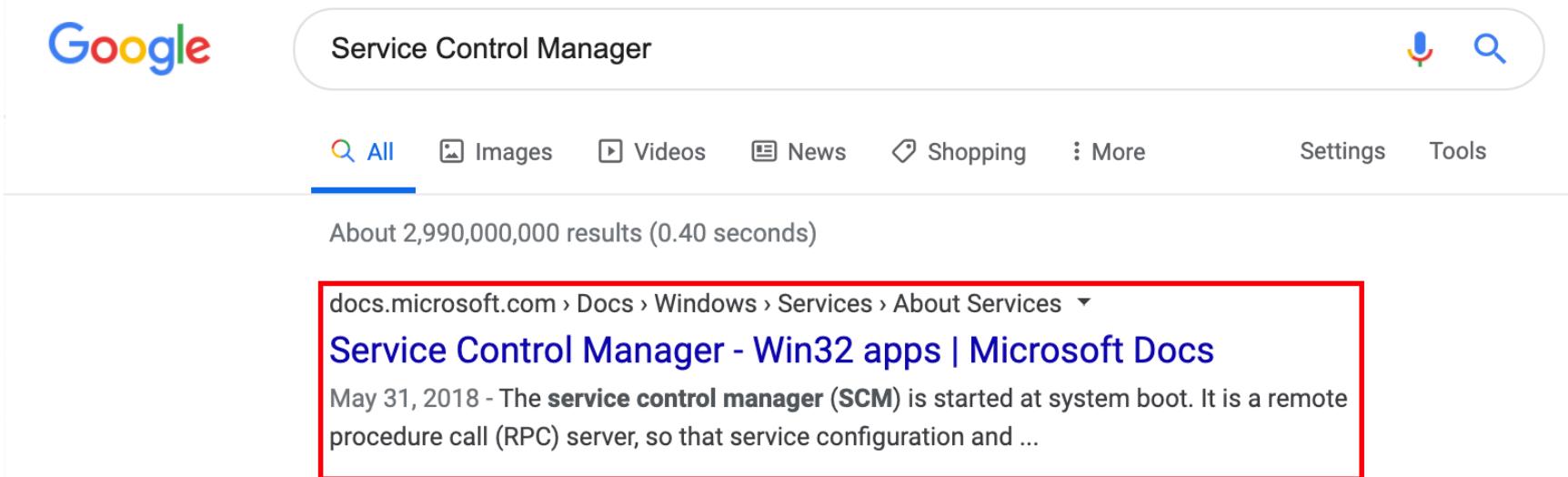
T1050 - Service Creation

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



A screenshot of a Google search results page. The search query "Service Control Manager" is entered in the search bar. Below the search bar, there are navigation links for All, Images, Videos, News, Shopping, More, Settings, and Tools. A status message indicates "About 2,990,000,000 results (0.40 seconds)". The top result is a link to "Service Control Manager - Win32 apps | Microsoft Docs" from docs.microsoft.com. This result is highlighted with a red rectangular box. The snippet from the Microsoft page describes the Service Control Manager (SCM) as a remote procedure call (RPC) server started at system boot.

Service Control Manager

All Images Videos News Shopping More Settings Tools

About 2,990,000,000 results (0.40 seconds)

docs.microsoft.com › Docs › Windows › Services › About Services ▾

Service Control Manager - Win32 apps | Microsoft Docs

May 31, 2018 - The **service control manager (SCM)** is started at system boot. It is a remote procedure call (RPC) server, so that service configuration and ...

Documentation



service control manager rpc site:docs.microsoft.com

All News Videos Images Shopping More Settings Tools

About 5,410 results (0.29 seconds)

docs.microsoft.com › Docs › Windows › Services › About Services ▾

Service Control Manager - Win32 apps | Microsoft Docs

May 31, 2018 - The service control manager (SCM) is started at system boot. It is a remote procedure call (RPC) server, so that service configuration and ...

People also ask

What does service control manager do? ▾

How do I start a service control manager? ▾

What is service control manager configuration tool? ▾

What services use RPC? ▾

Feedback

docs.microsoft.com › en-us › openspecs › windows_protocols › ms-sc... ▾

[MS-SCMR]: Service Control Manager Remote Protocol ...

Sep 23, 2019 - Specifies the Service Control Manager Remote Protocol, which is used for remotely managing the Service Control Manager (SCM), an RPC ...

[Previous Versions](#) · [Preview Versions](#)

docs.microsoft.com › en-us › openspecs › windows_protocols › ms-sc... ▾

[MS-SCMR]: Overview | Microsoft Docs

Feb 14, 2019 - The Service Control Manager Remote Protocol is a client/server protocol ... If the server accepts the request, it responds with an RPC context ...

docs.microsoft.com › ... › About Services › Service Programs ▾

Services and RPC/TCP - Win32 apps | Microsoft Docs

May 31, 2018 - ... the service control manager (SCM) supports remote procedure calls over both Transmission Control Protocol (RPC/TCP) and named pipes ...

Documentation

Filter by title

- Open Specifications
- Protocols
 - Protocols
 - Windows Protocols
 - Windows Protocols
- Technical Documents
 - Technical Documents
 - [MS-SCMR]: Service Control Manager Remote Protocol
 - [MS-SCMR]: Service Control Manager Remote Protocol
 - 1 Introduction
 - 1 Introduction
 - 1.1 Glossary
 - > 1.2 References
 - 1.3 Overview**
 - 1.4 Relationship to Other Protocols
 - 1.5 Prerequisites/Preconditions
 - 1.6 Applicability Statement
 - 1.7 Versioning and Capability Negotiation
 - 1.8 Vendor-Extensible Fields
 - 1.9 Standards Assignments

1.3 Overview

02/14/2019 • 2 minutes to read

The Service Control Manager Remote Protocol is a client/server protocol used for configuring and controlling [service](#) programs running on a remote computer. A remote service management session begins with the client initiating the connection request to the server. If the server grants the request, the connection is established. The client can then make multiple requests to modify, query the configuration, or start and stop services on the server by using the same session until the session is terminated.

A typical Service Control Manager Remote Protocol session involves the client connecting to the server and requesting to open the [SCM](#) on the server. If the server accepts the request, it responds with an [RPC context handle](#) to the client. The client uses this RPC context handle to operate on the server. This usually involves sending another request to the server and specifying the type of operation to perform and any specific parameters associated with that operation. If the server accepts this request, it attempts to perform the specified operation and responds to the client with the result of the operation. After the client is finished operating on the server, it terminates the protocol by sending a request to close the RPC context handle.

The Service Control Manager Remote Protocol maintains an internal database to store service program configurations and state. The Service Control Manager Protocol has exclusive access to this internal database. On one operating system instance there is only one SCM and one corresponding SCM database. Any updates to this internal database are made only through the Service Control Manager Remote Protocol. SCM takes care of serializing all concurrent accesses to the SCM database. The SCM database is resident in memory; it is recreated every time the SCM restarts (after each reboot). Part of the SCM database is retrieved from persistent storage (all information regarding registered services) and partially nonpersistent (current active state of the services). The persistent information is modified by the SCM when a service is added, configured, or deleted. Any attempt to directly modify the persistent part of the database directly in the persistent storage is not a supported scenario and will result in possible inconsistencies. Finally, if SCM were to be forcefully terminated, the operating system will shut down and restart.

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



scm database site:docs.microsoft.com



All Images News Videos Shopping More Settings Tools

About 5,470 results (0.40 seconds)

docs.microsoft.com > Docs > Windows > Services > About Services ▾

Service Control Manager - Win32 apps | Microsoft Docs

May 31, 2018 - The service functions provide an interface for the following tasks performed by the **SCM**: Maintaining the **database** of installed services. Starting ...

People also ask

What is SCM in Task Manager? ▾

How do I access service control manager? ▾

What is service control manager configuration tool? ▾

Where are Windows services installed? ▾

Feedback

docs.microsoft.com > ... > About Services > Service Control Manager ▾

Database of Installed Services - Win32 apps | Microsoft Docs

May 31, 2018 - The **database** is used by the **SCM** and programs that add, modify, or configure services. The following is the registry key for this **database**: ...

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]
Registry Service Database	\PIPE\svcctl
	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

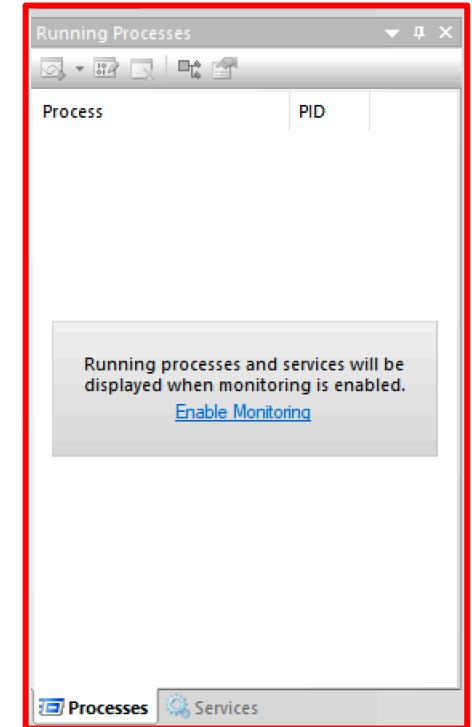
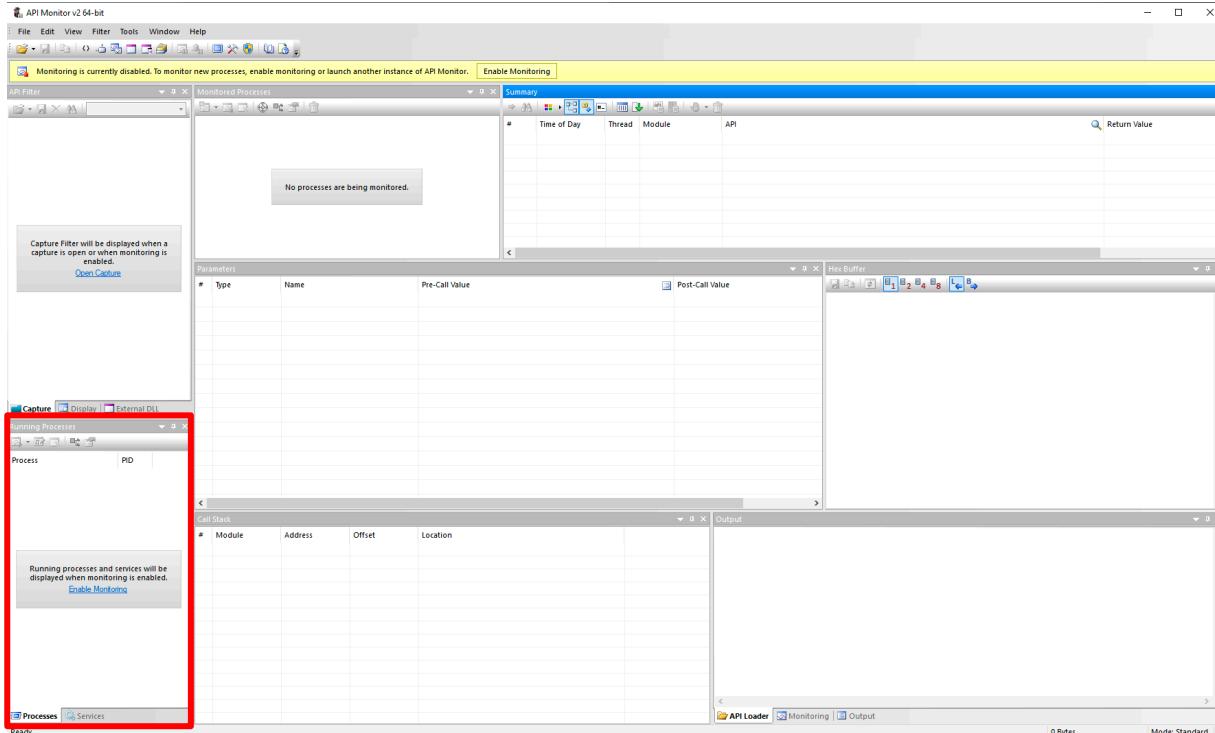
T1050 - Service Creation

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

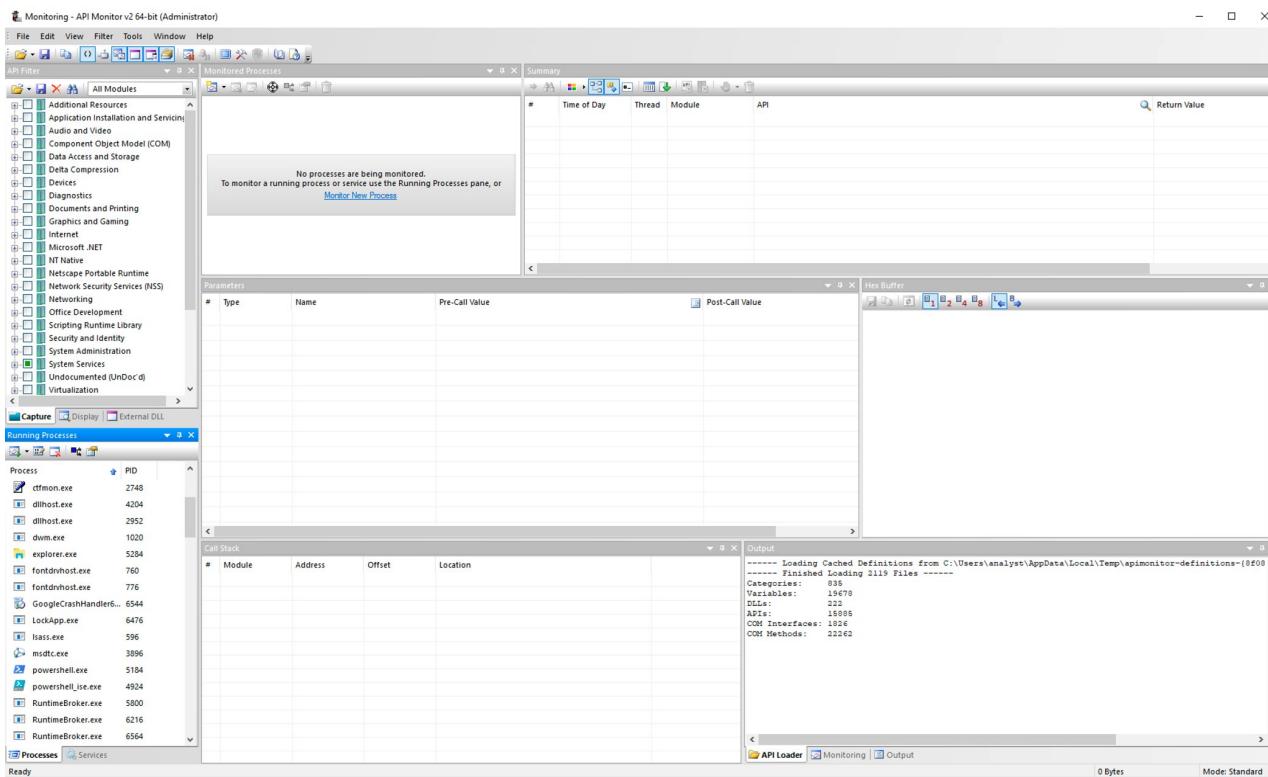
Functions

Dynamic Analysis - API Monitor

Enable Monitoring



Monitoring Enabled



Find the Correct PowerShell Process

The screenshot shows a Windows PowerShell session and the API Monitor tool running side-by-side.

Windows PowerShell Session:

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> $PID
5184
```

API Monitor Tool:

The API Monitor tool displays the following information:

- Running Processes pane:** Shows a list of running processes. A red box highlights the entry for "powershell.exe" with PID 5184.
- Monitored Processes pane:** Displays a summary table with columns: Time of Day, Thread, Module, and API. It shows no processes are being monitored.
- Call Stack pane:** Shows the call stack for the selected process (powershell.exe, PID 5184). It includes sections for Call Stack, Output, and API Loader.
- Output pane:** Displays log messages related to API definitions and file loading.

Legend:

- Process
- PID
- Module
- Address
- Offset
- Location

Output Log:

```
----- Loading Cached Definitions from C:\Users\analyst\AppData\Local\Temp\apimonitor-definitions-(#f08)
----- Finished Loading 2119 Files -----
Categories: 835
Variables: 19670
Functions: 122
APIs: 18938
COM Interfaces: 1826
COM Methods: 22262
```

Select Services Related API Functions

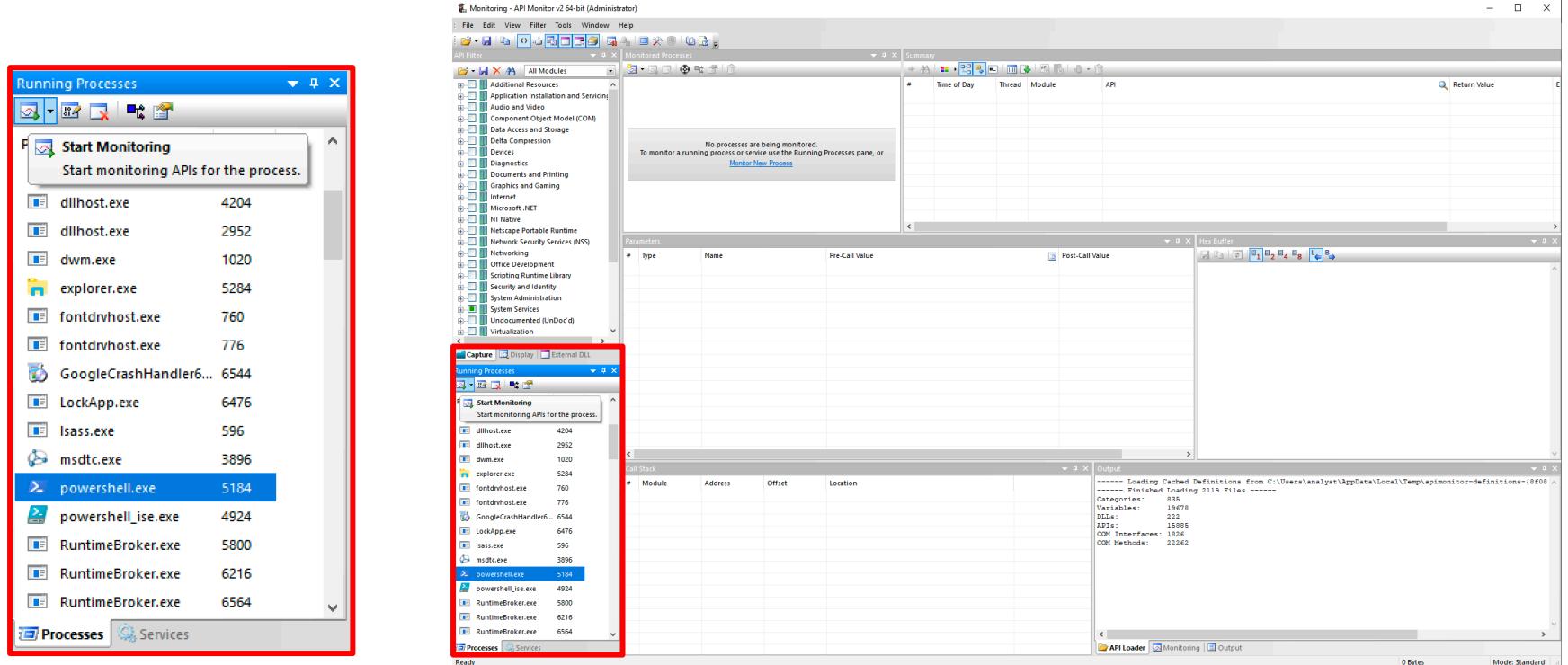
The screenshot shows the API Monitor interface with two main panes highlighted by red boxes.

Left Pane (API Filter): This pane displays a tree view of API categories. The 'Services' category under 'System Services' is expanded, and its sub-item 'Advapi32.dll' is selected. Other visible categories include Office Development, Scripting Runtime Library, Security and Identity, System Administration, Compression, Distributed Transaction Coordination, Dynamic-Link Libraries, Indexing Service, Interprocess Communications, Kernel Transaction Manager (KTM), Memory Management, Power Management, Processes and Threads, Remote Desktop Services, and Visual C++ Run-Time Library.

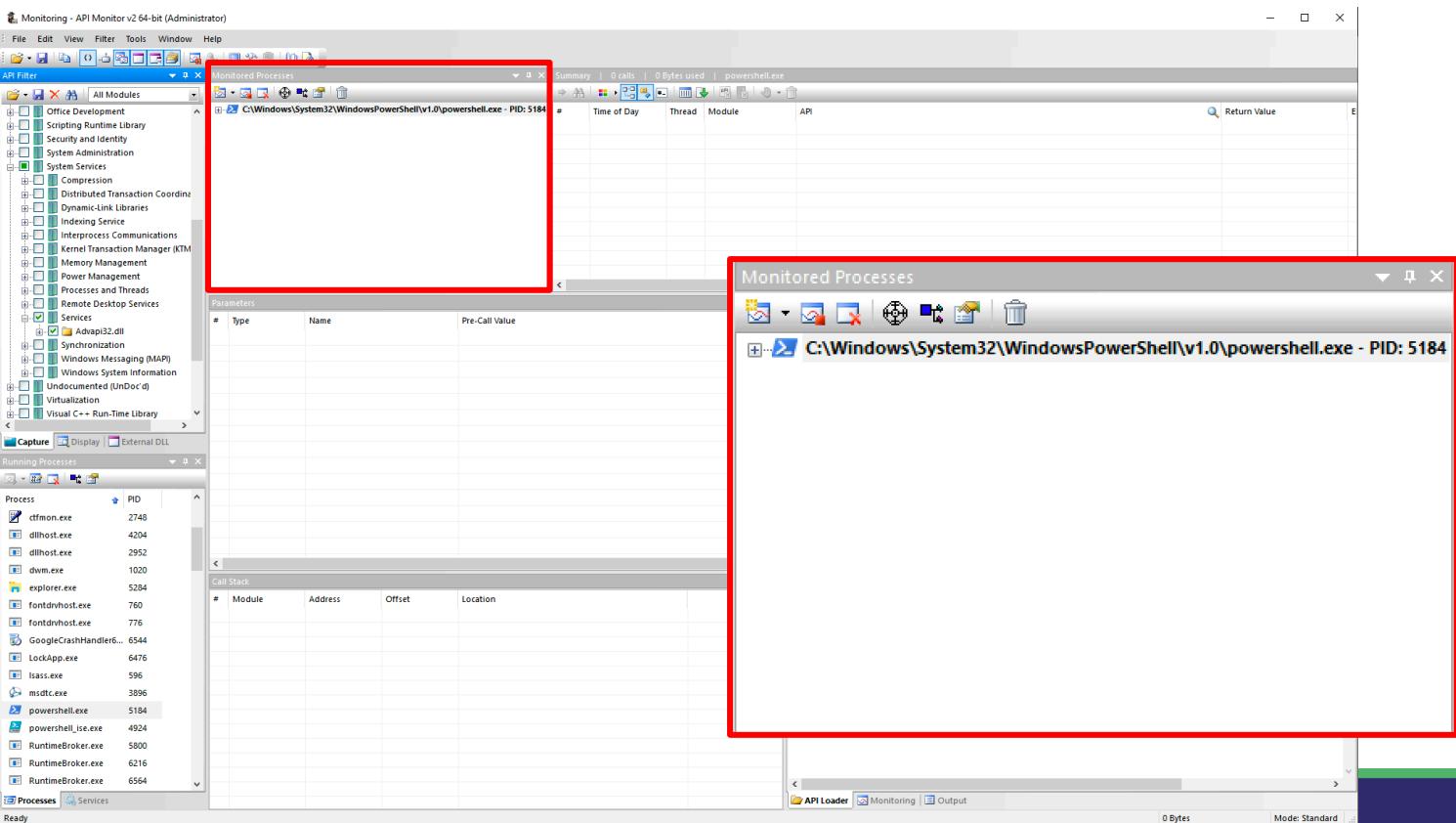
Right Pane (Monitoring): This pane shows monitoring data for the process 'powershell.exe'. It includes sections for Monitored Processes, Parameters, Call Stack, and Output. The 'Output' section displays log messages such as:

```
--> Loading Cached Definitions from C:\Users\analyst\AppData\LocalTemp\apimonitor-definitions-[#000]
----- Finished Loading 2115 Files -----
Categories: 835
Variables: 19678
Functions: 231
APIs: 15086
COM Interfaces: 1816
COM Methods: 22242
```

Start Monitoring PowerShell



Monitoring Started



Executing New-Service

Administrator: Windows PowerShell

```
PS C:\Windows\system32> New-Service -Name abstraction -BinaryPathName C:\Windows\notepad.exe -DisplayName abstraction -StartupType Automatic
```

Status	Name	DisplayName
Stopped	abstraction	abstraction

Monitoring - API Monitor v2.64-bit (Administrator)

File Edit View Filter Tools Window Help

API Filter Monitored Processes

All Modules

- Office Development
- Scripting Runtime Library
- Security and Identity
- System Administration
- System Services
 - Background
 - Distributed Transaction Coordinator
 - Dynamic Link Libraries
 - Indexing Service
 - Interprocess Communications
 - Kernel Transaction Manager (KTM)
 - Memory Management
 - Power Management
 - Processes and Threads
 - Remote Desktop Services
 - Services

Summary | 11 calls | 4 KB used | powershell.exe

#	Time of Day	Thread	Module	API	Return Value
1	12:41:15.398 PM	6	clr.dll	OpenSCManagerW (NULL, NULL, SC_MANAGER_CONNECT SC_MANAGER_CREATE_SERVICE)	0x000002142ad01670
2	12:41:15.398 PM	6	clr.dll	CreateServiceW (0x000002142ad01670, "abstraction", "abstraction", SERVICE_CHANGE_CONFIG, SERVICE...	0x000002142ad01b80
3	12:41:15.398 PM	6	clr.dll	ChangeServiceConfig2W (0x000002142ad01b80, SERVICE_CONFIG_DESCRIPTION, 0x000002142acec5c0)	TRUE
4	12:41:15.445 PM	6	clr.dll	OpenSCManagerW (NULL, NULL, SC_MANAGER_CONNECT)	0x000002142ad01580
5	12:41:15.445 PM	6	clr.dll	GetServiceKeyNameW (0x000002142ad01580, "abstraction", "", 0x000000644a30dc90)	
6	12:41:15.866 PM	6	clr.dll	OpenServiceW (0x000002142ad01580, "abstraction", SERVICE_QUERY_STATUS)	0x000002142ad01be0
7	12:41:15.866 PM	6	clr.dll	QueryServiceStatus (0x000002142ad01be0, 0x000000644a30dd28)	TRUE
8	12:41:15.866 PM	6	clr.dll	CloseServiceHandle (0x000002142ad01be0)	TRUE
9	12:41:15.882 PM	6	clr.dll	CloseServiceHandle (0x000002142ad01580)	TRUE
10	12:41:15.882 PM	6	clr.dll	CloseServiceHandle (0x000002142ad01b80)	TRUE
11	12:41:15.882 PM	6	clr.dll	CloseServiceHandle (0x000002142ad01670)	TRUE

Summary | 11 calls | 4 KB used | powershell.exe

#	Time of Day	Thread	Module	API	Return Value
1	12:41:15.398 PM	6	clr.dll	OpenSCManagerW (NULL, NULL, SC_MANAGER_CONNECT SC_MANAGER_CREATE_SERVICE)	0x000002142ad01670
2	12:41:15.398 PM	6	clr.dll	CreateServiceW (0x000002142ad01670, "abstraction", "abstraction", SERVICE_CHANGE_CONFIG, SERVICE...	0x000002142ad01b80
3	12:41:15.398 PM	6	clr.dll	ChangeServiceConfig2W (0x000002142ad01b80, SERVICE_CONFIG_DESCRIPTION, 0x000002142acec5c0)	TRUE
4	12:41:15.445 PM	6	clr.dll	OpenSCManagerW (NULL, NULL, SC_MANAGER_CONNECT)	0x000002142ad01580
5	12:41:15.445 PM	6	clr.dll	GetServiceKeyNameW (0x000002142ad01580, "abstraction", "", 0x000000644a30dc90)	
6	12:41:15.866 PM	6	clr.dll	OpenServiceW (0x000002142ad01580, "abstraction", SERVICE_QUERY_STATUS)	0x000002142ad01be0
7	12:41:15.866 PM	6	clr.dll	QueryServiceStatus (0x000002142ad01be0, 0x000000644a30dd28)	TRUE
8	12:41:15.866 PM	6	clr.dll	CloseServiceHandle (0x000002142ad01be0)	TRUE
9	12:41:15.882 PM	6	clr.dll	CloseServiceHandle (0x000002142ad01580)	TRUE
10	12:41:15.882 PM	6	clr.dll	CloseServiceHandle (0x000002142ad01b80)	TRUE
11	12:41:15.882 PM	6	clr.dll	CloseServiceHandle (0x000002142ad01670)	TRUE

Output

```
----- Loading Cached Definitions from C:\Users\analyst\AppData\Local\Temp\epimonitor-definitions-(0f00... ----- Finished Loading 2119 Files -----
```

Variables: 838

Functions: 1678

DLLs: 232

APIs: 16988

COM Interfaces: 1626

COM Methods: 22482

Processes Services

RuntimeBroker.exe 5800

RuntimeBroker.exe 6216

RuntimeBroker.exe 6564

Processes Services

Ready

SPECTER OPS

Calling OpenSCManagerW

The screenshot shows the API Monitor interface with the following details:

Monitored Processes: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe - PID: 5184

Summary: 11 calls | 4 KB used | powershell.exe

Call Stack:

#	Time of Day	Thread	Module	API	Return Value
1	12:41:15.390 PM	6	clr.dll	OpenSCManagerW (0x000002142ad01670 NULL, NULL, SC_MANAGER_CONNECT SC_MANAGER_CREATE_SERVICE)	0x000002142ad01670
2	12:41:15.390 PM	6	clr.dll	CreateServiceW (0x000002142ad01670, "abstraction", abstraction, SC_MANAGER_CONNECT, SERVICE_CONFIG_DESCRIPTION, 0x000002142ad0e5c0)	TRUE
3	12:41:15.390 PM	6	clr.dll	ChangeServiceConfigW (0x000002142ad01b80, SERVICE_CONFIG_DESCRIPTION, 0x000002142ad0e5c0)	TRUE
4	12:41:15.445 PM	6	clr.dll	OpenSCManagerW (NULL, NULL, SC_MANAGER_CONNECT)	0x000002142ad01580
5	12:41:15.445 PM	6	clr.dll	GetServiceKeyNameW (0x000002142ad01580, "abstraction", "", 0x00000064a3d0d0)	0x000002142ad01be0
6	12:41:15.866 PM	6	clr.dll	OpenServiceW (0x000002142ad01580, "abstraction", SERVICE_QUERY_STATUS)	0x000002142ad01be0
7	12:41:15.866 PM	6	clr.dll	QueryServiceStatus (0x000002142ad01b40, 0x0000064a3d0d28)	TRUE
8	12:41:15.866 PM	6	clr.dll	CloseServiceHandle (0x000002142ad01b40)	TRUE
9	12:41:15.882 PM	6	clr.dll	CloseServiceHandle (0x000002142ad01b50)	TRUE
10	12:41:15.882 PM	6	clr.dll	CloseServiceHandle (0x000002142ad01b80)	TRUE
11	12:41:15.882 PM	6	clr.dll	CloseServiceHandle (0x000002142ad01b70)	TRUE

Parameters: OpenSCManagerW (Advapi32.dll)

#	Type	Name	Pre-Call Value	Post-Call Value
1	LPCWSTR	lpMachineName	NULL	NULL
2	LPCWSTR	lpDatabaseName	NULL	NULL
3	DWORD	dwDesiredAccess	SC_MANAGER_CONNECT SC_MANAGER_CREATE_SERVICE	SC_MANAGER_CONNECT SC_MANAGER_CREATE_SERVICE

Return: 0x000002142ad01670

Calling CreateServiceW

The screenshot shows the API Monitor interface with several windows open. The main window displays a timeline of API calls made by the process powershell.exe (PID: 5184). A specific call to `CreateServiceW` is highlighted with a red box. The parameters for this call are shown in a detailed view, also highlighted with a red box. The parameters are:

#	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

The Post-Call Value column shows the return value for each parameter. The bottom right corner of the screenshot shows a hex dump of the memory buffer.

T1050 - Service Creation

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

Visit Function Documentation

The screenshot shows a Microsoft Docs page for the Windows Dev Center. The URL is <https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-createservicew>. The page title is "CreateServiceW function". The left sidebar lists various Windows API functions, with "CreateServiceW function" highlighted by a dark gray rectangle. The main content area contains the function's syntax in C++:

```
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
);
```

Find Exporting DLL

Requirements

Minimum supported client Windows XP [desktop apps only]

Minimum supported server Windows Server 2003 [desktop apps only]

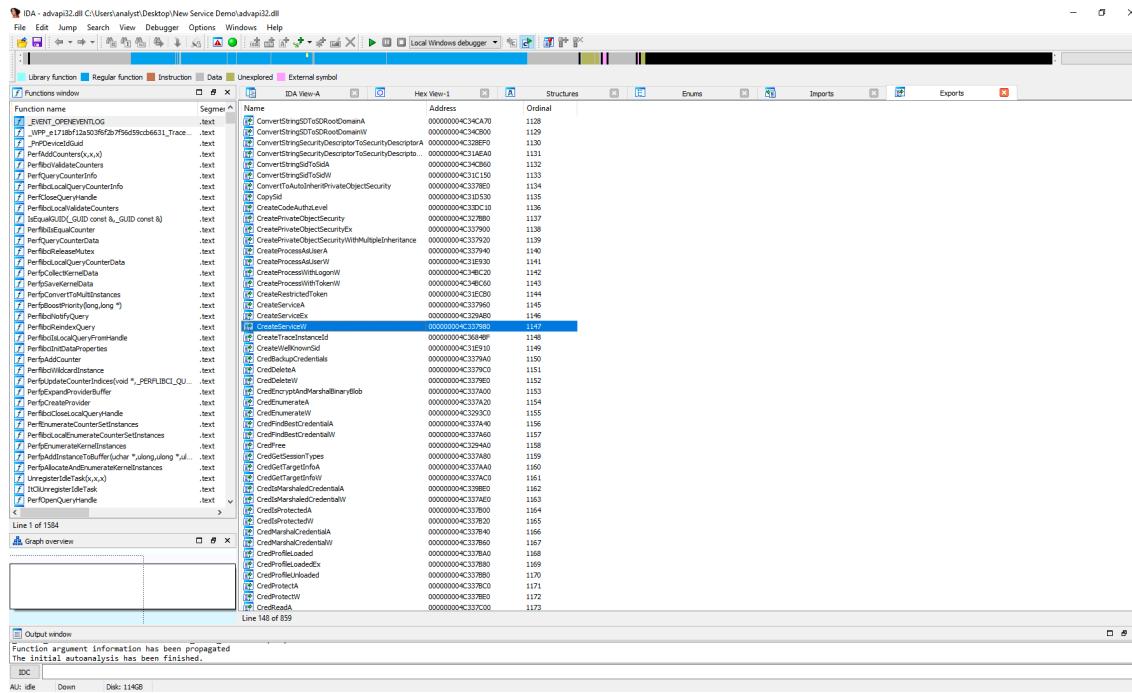
Target Platform Windows

Header winsvc.h (include Windows.h)

Library Advapi32.lib

DLL Advapi32.dll

ADVAPI32.DLL – Export Table



ADVAPI32.DLL – CreateServiceW

The screenshot shows the IDA Pro interface with the assembly view open. The function `_CreateServiceWStub@52` is selected. The assembly code is as follows:

```
; Exported entry 1147. CreateServiceW
; Attributes: bp-based frame
public _CreateServiceWStub@52
_CreateServiceWStub@52 proc near

hSCManager dword ptr 8
lpServiceName= dword ptr 0Ch
lpDisplayName= dword ptr 10h
dwDesiredAccess= dword ptr 14h
dwServiceType= dword ptr 15h
dwStartType= dword ptr 1Ch
dwErrorControl= dword ptr 28h
lpBinaryPathName= dword ptr 24h
lpLoadOrderGroup= dword ptr 28h
lpInstallContext= dword ptr 2Ch
lpDependencies= dword ptr 30h
lpServiceStartName= dword ptr 34h
lpPassword= dword ptr 38h
lpPassword2= dword ptr 38h

    mov    edi, edi
    push   ebp
    mov    ebp, esp
    pop    ebp
    jmp    ds:_imp__CreateServiceW@52 ; CreateServiceW((x,x,x,x,x,x,x,x,x,x))
_CreateServiceWStub@52 endp
```

The assembly code is annotated with comments explaining the parameters and the call to `>CreateServiceW`. The stack frame is established with `bp-based frame`. The function ends with a jump to the `ds:_imp__CreateServiceW@52` symbol.

ADVAPI32.DLL – Forwarded Export

The screenshot shows the assembly view of the `advapi32.dll` file in IDA Pro. The assembly code is color-coded to indicate different types of symbols:

- Library function**: Light blue background.
- Regular function**: Light green background.
- Instruction**: White background.
- Data**: Yellow background.
- Unexplored**: Pink background.
- External symbol**: White background with purple text.

A red box highlights a section of the assembly code that corresponds to the `CreateServiceW` function from the `api-ms-win-service-management-l1-1-0.dll`. The highlighted code includes:

```
extrn __imp__GetServiceKeyName@16:dword ; DATA XREF: GetServiceKeyNameStub(x,x,x,x)+6fr
idata:4C3715F8 ; .idata:4C3715F8 ; Imports from api-ms-win-service-management-l1-1-0.dll
idata:4C371600 ; .idata:4C371600 ; SC_HANDLE __stdcall CreateServiceW(SC_HANDLE hSCManager, LPCWSTR lpServiceName, LPCWSTR lpDisplayName, DWORD dwDesiredAccess, DWORD dwServiceType, DWORD dwStartType, DWORD dwErrorControl
idata:4C371600 ; .idata:4C371600 ; extrn __imp__CreateServiceW@52:dword
idata:4C371600 ; .idata:4C371600 ; ; DATA XREF: CreateServiceWStub(x,x,x,x,x,x,x,x,x,x)+6fr
idata:4C371600 ; .idata:4C371600 ; .idata:4C371B28@o
```

Below the assembly code, the `Exports` table shows the forwarded export:

Name	Address	Type
<code>CreateServiceW</code>	<code>idata:4C371600</code>	Forwarded

The bottom status bar indicates the assembly view is synchronized with the `Hex View-1`.

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

Geoff Chappell, Software Analyst

[About This Site](#) [What's New?](#)

Home Notes Kernel Win32 Shell Internet Explorer Visual C++

Win32

- ApisetSchema
- Versions
- API Sets
 - API-MS-Win-Core-Console-L1-1-0
 - API-MS-Win-Core-DateTime-L1-1-0
 - API-MS-Win-Core-Debug-L1-1-0
 - API-MS-Win-Core-DelayLoad-L1-1-0
 - API-MS-Win-Core-File-Handle-L1-1-0
 - API-MS-Win-Core-Fibers-L1-1-0
 - API-MS-Win-Core-File-L1-1-0
 - API-MS-Win-Core-Handle-L1-1-0
 - API-MS-Win-Core-Heap-L1-1-0
 - API-MS-Win-Core-Interlocked-L1-1-0
 - API-MS-Win-Core-Io-L1-1-0
 - API-MS-Win-Core-Job-Object-L1-1-0
 - API-MS-Win-Core-NamedPipe-L1-1-0
 - API-MS-Win-Core-ProcessEnvironment-L1-1-0
 - API-MS-Win-Core-ProcessThreads-L1-1-0
 - API-MS-Win-Core-Profiling-L1-1-0
 - API-MS-Win-Core-RISSupport-L1-1-0
 - API-MS-Win-Core-String-L1-1-0
 - API-MS-Win-Core-Synch-L1-1-0
 - API-MS-Win-Core-SyntInfo-L1-1-0
 - API-MS-Win-Core-Threadpool-L1-1-0
 - API-MS-Win-Core-Ums-L1-1-0
 - API-MS-Win-Core-XState-L1-1-0
 - API-MS-Win-Core-XStateR-L1-1-0
 - API-MS-Win-Security-SALookup-L1-1-0
 - API-MS-Win-Security-SDL-L1-1-0
 - API-MS-Win-Service-Core-L1-1-0
 - API-MS-Win-Service-Management-L1-1-0
 - API-MS-Win-Service-Management-L2-1-0
 - API-MS-Win-Service-winsvc-L1-1-0
- NTDLL
- CSRSRV
- BASESRV
- WINSRV
- KERNELBASE
- KERNEL32
- RPCRT4
- ADVAPI32
- Services
- COM
- APPHELP
- Desktop Window Manager

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

All functions in the API-MS-Win-Service-winsvc-L1-1-0 set are exports from ADVAPI32:

- ! ChangeServiceConfigA
- ! ChangeServiceConfig2A
- ! ControlService
- ! ControlServiceExA
- ! CreateServiceA
- ! ! QueryInformation
- ! ! SetServiceControlMessage
- ! ! ScsSecurityProcess
- ! ! ScpnPgtServiceName
- ! ! ScQueryServiceConfig
- ! ! ScQueryServiceStatus
- ! ! ScRpBindW
- ! ! ScSendDpNpMessage
- ! ! ScSendDfSMessage
- ! ! ScSendFpNpService
- ! ! NotifyServiceStatusChangeA
- ! OpenSCManagerA
- ! OpenSCManagerExA
- ! ! GetServiceConfigA
- ! ! GetServiceConfig2A
- ! ! GetServiceStatus
- ! ! RegisterServiceCtrlHandlerA
- ! ! RegisterServiceCtrlHandlerW
- ! ! StartServiceCtrlDispatcherA
- ! ! StartServiceCtrlDispatcherW

For most of these functions, the implementations in ADVAPI32 version 6.1 and higher are just stubs which transfer the handling to wherever the schema redirects the API Set. The exceptions are:

- ! ! QueryInformation
- ! ! SetServiceControlMessage
- ! ! ScsSecurityProcess
- ! ! ScpnPgtServiceName
- ! ! ScQueryServiceConfig
- ! ! ScRpBindA
- ! ! ScSendDpNpW
- ! ! ScSendDpNpMessage
- ! ! ScSendDfSMessage
- ! ! ScValidatePnPService

which have no code in ADVAPI32 but are instead exported as forwards to the API Set.

New Locations

Non-trivial implementations of all functions in this API Set are exported from SECHOST version 6.1 and higher.

Schema Redirection

The Windows 7 schema redirects this API Set to SECHOST, thus:

- high-level executables, which do not use the API Set, continue to import these functions from ADVAPI32;
- low-level executables have their imports from the API Set redirected to SECHOST;

<https://www.geoffchappell.com/studies/windows/win32/apisetschema/api/ms/win/service/management/l1-1-0.htm>

Schema Redirection

The Windows 7 schema redirects this API Set to SECHOST.

SECHOST.DLL – Export Table

The screenshot shows the IDA Pro interface with the title "IDA - sechost.dll C:\Users\analyst\Desktop\New Service Demo\sechost.dll". The main window displays the export table, which lists 1420 functions. The columns in the table are: Function name, Segmer..., Name, Address, and Ordinal. The "Name" column uses color coding to distinguish between different types of symbols: Library function (light blue), Regular function (dark blue), Instruction (yellow), Data (orange), Unexplored (pink), and External symbol (grey). The "Address" column shows memory addresses, and the "Ordinal" column shows the function's position in the export table. The bottom of the screen shows the status bar with "Line 1 of 1420" and "Line 45 of 218", and the output window with the message "Old method of loading PDB files (dbghelp) was successful".

Function name	Segmer...	Name	Address	Ordinal
7_HibernationGuid	.text	Capability_Check	0000000010020970	20
7_SleepGuid	.text	Capability_CheckForSingleSessionSku	000000001004E340	21
7_PerfmonGuid	.text	ChangeServiceConfig3A	000000001004E980	22
7_FileGuid	.text	ChangeServiceConfig2W	0000000010025590	23
7_MediaSyncGuid	.text	ChangeServiceConfigA	000000001004E4D0	24
7_HypervisorTraceGuid	.text	ChangeServiceConfigW	0000000010026250	25
7_UneventEventGuid	.text	CloseServiceHandle	0000000010018870	26
7_IptGuid	.text	CloseTrace	000000001001E5E0	27
wl_details__dynamic_initializer_for_g_enabledSt...	.text	ControlService	0000000010022520	28
wl_details__dynamic_initializer_for_g_featureSt...	.text	ControlServiceExA	000000001004EC60	29
wl_details__dynamic_initializer_for_g_header_in...	.text	ControlServiceExW	00000000100157A0	30
wl_details__dynamic_initializer_for_g_header_in...	.text	ControlTraceA	0000000010051D80	31
wl_details__dynamic_initializer_for_g_header_in...	.text	ControlTraceW	0000000010051960	32
wl_details__dynamic_initializer_for_g_header_in...	.text	ConvertStringToRfc2307DomainW	0000000010019000	33
wl_details__dynamic_initializer_for_g_header_in...	.text	ConvertSecurityDescriptorToStringSecurityDescripto...	000000001010F000	34
wl_details__dynamic_initializer_for_g_header_in...	.text	ConvertStringToRfc2307DomainW	0000000010014E40	35
wl_details__dynamic_initializer_for_g_header_in...	.text	ConvertStringToSidA	0000000010030690	36
wl_details__dynamic_initializer_for_g_header_in...	.text	ConvertStringToSidW	00000000100361B0	37
wl_details__dynamic_initializer_for_g_header_in...	.text	ConvertStringToOidDomainW	0000000010036230	38
wl_details__dynamic_initializer_for_g_header_in...	.text	ConvertStringToOidDomainW	0000000010010E00	39
wl_details__dynamic_initializer_for_g_header_in...	.text	ConvertStringToSidW	0000000010010D00	40
wl_details__dynamic_initializer_for_g_header_in...	.text	CreateIsolatedProcess	0000000010064500	41
wl_details__dynamic_initializer_for_g_header_in...	.text	CreateIsolationContainer	0000000010064550	42
wl_details__dynamic_initializer_for_g_header_in...	.text	CreateServiceA	000000001004ED50	43
wl_details__dynamic_initializer_for_g_header_in...	.text	CreateServiceEx	000000001004F0E0	44
GetNormalizedObjectPath	.text	CreateServiceW	0000000010043600	45
QuerySecurityDescriptorValue	.text	CredBackupCredentials	00000000100256C0	46
R5StringCopyExW	.text	CredDelete	00000000100256F0	47
R5StringCopyW(x,x)	.text	CredFree	0000000010025700	48
R5StringCmpW(x,x)	.text	CredEncryptAndMarshalBinaryBlob	0000000010057010	49
R5StringCopyWorkerW	.text	CredEnumerateA	00000000100562A0	50
R5StringCLengthW(x,x,x)	.text	CredEnumerateW	0000000010036100	51
R5StringCLengthWorkerW	.text	CredFindBestCredentialA	0000000010056300	52
R5StringCValidateDestW	.text	CredFindBestCredentialW	0000000010056400	53
R5StringCLengthW(x,x,x)	.text	CredFree	00000000100256D0	54
R5SizeAdd(x,x,x)	.text	CredGetSessionType	0000000010056530	55
CreatePerUserSecurityDescriptor	.text	CredGetTargetInfoA	0000000010056580	56
LsaLookupTranslateSids	.text	CredGetTargetInfoW	0000000010056660	57
LocateGsidForString	.text	CredMarshalCredentialW	00000000100235A0	58
ConvertStringSidToSIDW	.text	CredProtectData	0000000010027040	59
ConvertStringSecurityDescriptorToStringSecurity...	.text	CredProtectDataW	0000000010027070	60
ConvertStringSecurityDescriptorToStringSecurity...	.text	CredMarshalCredentialA	0000000010037080	61
LocateUserStringSidForString	.text	CredMarshalCredentialW	0000000010023440	62
LocalConvertSidToStringRev	.text	CredProfileNameWithType	00000000100222E0	63
LocalConvertSidToStringRev	.text	CredProfileLoad	0000000010056570	64
LocalGetAdForString	.text	CredProfileLoadEx	00000000100242D0	65

SECHOST.DLL - CreateServiceW

The screenshot shows the IDA Pro interface with the file `sechost.dll` open. The assembly view displays the `CreateServiceW` function, which is exported from the DLL. The assembly code is as follows:

```
; Exported entry 45. CreateServiceW
; Attributes: bp-based frame
; SC_HANDLE _stdcall CreateServiceW(SC_HANDLE hSCManager, LPCWSTR lpServiceName, DWORD dwDesiredAccess, DWORD dwServiceType, DWORD dwStartType, DWORD dwErrorControl, LPCWSTR lpServiceName, _CreateServiceW@52 proc near
var_3C= dword ptr -3Ch
var_38= dword ptr -38h
var_34= dword ptr -34h
var_30= dword ptr -30h
var_2C= dword ptr -2Ch
var_28= dword ptr -28h
Ms_ecx= word ptr -4Ah
var_20= dword ptr -20h
var_1C= word ptr -1Ch
Ms_exce= CPPEN_RECORD ptr -18h
ContextHandle= dword ptr 8
lpServiceName= dword ptr 0Ch
lpDisplayname= dword ptr 20h
dwDesiredAccess= dword ptr 14h
dwServiceType= dword ptr 18h
dwStartType= dword ptr 1Ch
dwErrorControl= dword ptr 20h
lpStartParameter= dword ptr 24h
lpLoadOrderGroup= dword ptr 28h
lpdwTagId= dword ptr 2Ch
lpDependencies= dword ptr 30h
lpServiceStartName= dword ptr 34h
lpPassword= dword ptr 38h

push 2Ch
push offset stru_100668C68
call STH prologue
xor edi, edi
mov [ebpvar_30], edi
mov [ebpvar_30], edi
mov [ebpvar_30], edi
mov [ebpvar_2C], edi
xor eax, eax
mov [ebpvar_1C], ax
mov [bpMs_exce.registration.TryLevel], edi
mov edx, [ebpIpPassword]
test edx, edx
je short loc_1004F3B6
```

The assembly code is annotated with comments explaining the parameters and local variables. The `Graph overview` window at the bottom left shows the control flow graph for the function. The `Output window` at the bottom right displays the message: "Old method of loading PDB files (dbghelp) was successful".

SECHOST.DLL – CreateServiceW Calls NdrClientCall4

IDA - sechost.dll C:\Users\analyst\Desktop\New Service Demo\sechost.dll

File Edit Jump Search View Debugger Options Windows Help

Local Windows debugger

Functions window

Segments

call ds: __imp_GetCurrentProcess@0 ; GetCurrentProcess()

push eax

call ds: __imp_IsWow64Process@12 ; IsWow64Process(x,x,x)

test eax, eax

short loc_1004F3A8

loc_1004F3E1:

mov ax, [ebp+var_1c]

test ax, ax

jz loc_1004F4EF

loc_1004F451:

mov esp, [ebp+ms_exc.old_esp]

mov ecx, [ebp+var_34]

call ?S�mapRpcError@@YAGKKK@Z ; S�mapRpcError(ulong,ulong)

xor edi, edi

loc_1004F45E:

mov esi, eax

mov [ebp+var_20], esi

mov [ebp+ms_exc.registration.TryLevel], edi

cmp esi, 78h

jnz loc_1004F557

Line 1 of 1420

Graph overview

100.00% (736,1349) (1000,773) 0004E034 00000001004F434: CreateServiceW(x,x)

Old method of loading PDB files (dbghelp) was successful

PDB: total 2159 symbols loaded for C:\Symbols\sechost.pdb|D451E534716E66709C8845C8D693C9CF1\sechost.pdb

IDC

AU: idle Down Disk: 114GB

NdrClientCall4

The screenshot shows a Microsoft Docs page for the **NdrClientCall4** function. The page has a dark theme. At the top, there's a navigation bar with links for Microsoft, Docs, Documentation, Learn, Q&A, and Code Samples. Below that is a secondary navigation bar for the Windows Dev Center with links for Explore, Platforms, Docs, Downloads, Samples, Support, and Dashboard. The main content area shows the **NdrClientCall4** function page. It includes a sidebar with a "Filter by title" input field and a list of related headers like `Remote Procedure Call (RPC)`, `Middles.h`, `Rpcch.h`, etc. The main content starts with the **NdrClientCall4** function title, its last update (12/05/2018), and a note that it is not supported. It then shows the **Syntax** section with C++ code:

```
C++  
CLIENT_CALL_RETURN RPC_VAR_ENTRY NdrClientCall4(  
    PMIDL_STUB_DESC pStubDescriptor,  
    PFORMAT_STRING pFormat,  
    ...  
)
```

Below the syntax is the **Parameters** section, which lists `pStubDescriptor` (Reserved), `pFormat` (Reserved), and `NdrClientCall2` (function).

IDL_STUB_DESC

IDL_STUB_DESC structure

12/05/2018 • 2 minutes to read

The **IDL_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++  
  
typedef struct _IDL_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 COMMFAULT_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;  
} IDL_STUB_DESC;
```

SECHOST.DLL – RpcInterfaceInformation

IDA - sechost.dll C:\Users\analyst\Desktop\New Service Demo\sechost.dll

File Edit Jump Search View Debugger Options Windows Help

Library function Regular function Instruction Data Unexplored External symbol

Functions window

Function name Segments

Hex View-A Hex View-1 Structures Enums Imports Exports

push eax
call ds: _imp__GetCurrentProcess@0 ; GetCurrentProcess()
push eax
push ds:
call ds: _imp__IsWow64Process@12 ; IsWow64Process2(x,x,x)
test ax, ax
jz short loc_1004F3A8

loc_1004F3E1:
mov ax, [ebp+var_1C]
test ax, ax
jz loc_1004F4EF

mov [ebp+ms_exc.registration.TryLevel], 1
lea ecx, [ebp+var_30]
push ecx
push ax
push [ebp+var_28]
push [ebp+var_2C]
push [ebp+lpServiceStartName]
push [ebp+var_2D]
push [ebp+lpDependencies]
push [ebp+lpInstallFlag]
push [ebp+lpInstallOrderGroup]
push [ebp+lpBinaryPathName]
push [ebp+lpServiceControl]
push [ebp+lpServiceType]
push [ebp+lpDesiredAccess]
push [ebp+lpDisplayName]
push [ebp+lpServiceName]
push offset dword_10001AAC
push offset off_10001080
call ds: _imp__NtClientCall14
jmp short loc_1004F45E

loc_1004F45E:
mov esi, eax
push [ebp+var_20], esi
mov [ebp+ms_exc.registration.TryLevel], edi
cmp esi, 7Bh
...
loc_1004F45E:

Line 1 of 1420

Graph overview

Output window

Old method of loading PDB files (dbghelp) was successful
PDB: total 2150 symbols loaded for C:\Symbols\sechost.pdb\0451E534716E66709C8845C00D693C9CF1\sechost.pdb

IDC

AU: Ide Down Disk: 114GB

SECHOST.DLL – RpcInterfaceInformation

The screenshot shows the IDA Pro interface with the file `sechost.dll` open. The assembly view displays the `RpcInterfaceInformation` function. The assembly code is heavily annotated with comments from the debugger, indicating various memory allocations, deallocations, and audit operations. The assembly code includes several `dd offset` instructions pointing to memory addresses like `off_10001080`, `off_10001088`, and `off_10001090`. The comments describe the purpose of these operations, such as allocating memory for `SC_MIDI_user_allocate` and performing audit operations via `AuditFree`.

```
IDA - sechost.dll C:\Users\analyst\Desktop\New Service Demo\sechost.dll
File Edit Jump Search View Debugger Options Windows Help
Functions window Segments ^ IDA View-A Hex View-2 Hex View-1 Structures Enums Imports Exports
Library function Regular function Instruction Data Unexplored External symbol
Function name
_HeapSummaryGuid
_SplitGuid
PerfInfoGuid
FileGuid
JetBindGuid
HypervisorFaceGuid
UmEventGuid
JptGuid
vil_details_dynamic_initializer_for_g_enabled...
vil_details_dynamic_initializer_for_g_features...
vil_details_dynamic_initializer_for_g_header...
vil_details_dynamic_initializer_for_g_header...
vil_details_dynamic_initializer_for_g_header...
vil_details_dynamic_initializer_for_g_header...
vil_details_dynamic_initializer_for_g_header...
vil_details_dynamic_initializer_for_g_header...
EthpDcEventTraceCallbacks
EventTraceEventFilterAdapter
OpenServiceA
NetServiceStatusChangeNotify(x,x,x)
StartServices
FormalSecurityDescriptor
RiStringQwHandleExW
RiStringQwHandleW
UserTransactObjectSecurityDescriptor
GetNormalizedObjectPath
QuerySearchDescriptorValue
RiStringQcCopyExW
RiStringQcCopyW(x,x,x)
RiStringQcCopyW(x,x,x)
RiStringQcLengthW(x,x,x)
RiStringQcLengthW(x,x,x)
RiStringQcLengthW(x,x,x)
RiStringQcValidateDestW
RiStringQcLengthW(x,x,x)
RiSizeTAdd(x,x)
CreatePeerSecurityDescriptor
LocalGetService
LocalEndService
ConvertStringToSdW
ConvertStringSecurityDescriptorToSecurity...
ConvertUserSecurityDescriptorToSecurity...
LocalConvertStringG0T05_D_Rev1
LocalConvertStringG0_D_Rev1
LocalGetAForString
Line 1 of 1420
Output window
OLD method of loading PDB files (debugHelp) was successful
PDB: total 2150 symbols loaded for C:\Symbol\sechost.pdb\451E534716E66709C8845C00693C9Cf1\sechost.pdb
IDC
AU: idle Down Disk: 114GB
```

00000400 0000000010001080: `text off_10001080` (Synchronized with Hex View-1)

SECHOST.DLL – RpcInterfaceInformation

IDA - sechost.dll C:\Users\analyst\Desktop\New Service Demo\sechost.dll

File Edit Jump Search View Debugger Options Windows Help

Local Windows debugger

Library function Regular function Instruction Data Unexplored External symbol

Functions window Segments

_HeapSummaryGuid .text

_SplitGuid .text

_PerfInfoGuid .text

_FieldGuid .text

_ModboundGuid .text

_UmEventTraceGuid .text

_UmEventGuid .text

_JniGuid .text

wl_details_dynamic_initializer_for__enabledIt .text

wl_details_dynamic_initializer_for__featureIt .text

wl_details_dynamic_initializer_for__header_mii .text

wl_details_dynamic_initializer_for__header_mni .text

wl_details_dynamic_initializer_for__header_mni .text

wl_details_dynamic_initializer_for__header_mni .text

wl_details_dynamic_initializer_for__processo... .text

wl_details_dynamic_initializer_for__threadDai... .text

EbtpProcessEventTraceCallbacks .text

EbtpProcessEventToEventTraceAdapter .text

OpenServiceA .text

NotifyServiceStatusChangeA(x,x,x) .text

StartServiceA .text

FormatServiceDescriptor .text

RtlStringCbFromSrv .text

RtlStringPrintFromWorkerW .text

QueryTraceEventToEventSecurityDescriptor .text

GetNormalizedObjectPath .text

QuerySecurityDescriptorValue .text

RtStringCbCopyExW .text

RtStringCbCopy((x,x,x)) .text

RtStringCbCopyCrtW((x,x,x)) .text

RtStringCopyWorkerW .text

RtStringChLengthW((x,x,x)) .text

RtStringLengthWorkerW .text

RtStringValidateTextW .text

RtStringCbLengthW((x,x,x)) .text

RtSizeTAddf((x,x,x)) .text

CreatePerUserSecurityDescriptor .text

LsaLookupTrustedSids .text

LocalConvertStringToSdW .text

ConvertStringSecurityDescriptorToSdW .text

ConvertFexentyDescriptorToSecurity... .text

ConvertFexentyDescriptorToSdStringSecurity... .text

LocalConvertStringToSd_D_Revi... .text

LocalConvertSDtoSdStringD_Revi... .text

LocalGetAclForString .text

Line 1 of 1420

Output window

Old method of loading PDB files (.dbghelp) was successful
PDB: total 2150 symbols loaded for C:\Symbol\0451E534716E66709C8845C006939CFC1\sechost.pdb

IDC

AU: idle Down Disk: 114GB

00008178 000000010008D78: .text:unk_10008D78 (Synchronized with Hex View-1)

SECHOST.DLL – RpcInterfaceInformation

IDA - sechost.dll C:\Users\analyst\Desktop\New Service Demo\sechost.dll

File Edit Jump Search View Debugger Options Windows Help

Local Windows debugger

Library function Regular function Instruction Data Unexplored External symbol

Functions window

Segments:

- Function name Segments
- HeadSecurityGuid .text
- SetGuid .text
- PerfGuid .text
- FidoGuid .text
- ModBoundGuid .text
- HypervisorTraceGuid .text
- UmEventGuid .text
- JptGuid .text
- Wl_detailed_dynamic_initializer_for_wl_enabledSt... .text
- Wl_detailed_dynamic_initializer_for_wl_featureSt... .text
- Wl_detailed_dynamic_initializer_for_wl_header_m... .text
- EtwObjEventTraceCallbacks .text
- EtwMapEventToEventTraceAdapter .text
- OpenService .text
- NetifServiceStatusChange(x,x,x)
- StartServiceA .text
- FormatServiceDescriptor .text
- RtStringCbHrtfEtw .text
- RtStringPrvWorkerW .text
- QueryTransactedObjectSecurityDescriptor .text
- GetThreadLocalObjectValue .text
- QuerySecurityDescriptorValue .text
- RtStringCbCopyExW .text
- RtStringCbCopy(x,x,x)
- RtStringCbChtW(x,x,x)
- RtStringCpWorkerW .text
- RtStringCpLengthW(x,x,x)
- RtStringCpLengthW(x,x,x)
- RtStringValidateDestW .text
- RtStringCbLengthW(x,x,x)
- RtStringCbLengthW(x,x,x)
- ConvertStringSrvToSdW .text
- ConvertStringSecurityDescriptorToSecurity... .text
- ConvertStringSecurityDescriptorToStringSecurity... .text
- LocalConvertStringSrvToSd_Rev1 .text
- LocalConvertSdToString_Sd_Rev1 .text
- LocalGetAForString .text

Line 1 of 1420

000008D60 00 00 00 00 00 00 00 05 58 30 49 00 01 11 04 02 00

000008D70 3C 05 00 00 00 00 00 00 00 44 00 00 00 81 BB 7A 36

000008D80 44 98 F1 35 AD 32 98 F0 38 00 10 03 02 00 00 00

000008D90 04 5D 88 8A EB 1C C9 11 9F E8 08 00 2B 10 48 60

000008DA0 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

000008DB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

000008DC0 04 0E 00 00 04 05 09 00 03 09 0C 00 01 08 00 00

Old method of loading PDB files (dbghelp) was successful
PDB: total 2158 symbols loaded for C:\Symbols\sechost.pdb\0451e534716E66709c8845c80693c9cf1\sechost.pdb

IDC: ide Down Disk: 114GB

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

A screenshot of a Google search results page. The search query is "367abb81-9844-35f1-ad32-98f038001003". The results are as follows:

- docs.microsoft.com › openspecs › windows_protocols**
[MS-SCMR]: Server | Microsoft Docs
Feb 14, 2019 - The server interface is identified by UUID **367ABB81-9844-35F1-AD32-98F038001003**, version 2.0, using the RPC well-known endpoint ...
- docs.microsoft.com › openspecs › windows_protocols**
[MS-SCMR]: Standards Assignments | Microsoft Docs
Feb 14, 2019 - RPC interface UUID. (**367ABB81-9844-35F1-AD32-98F038001003**). Named pipe. \\PIPE\svctrl. Related Articles. Is this page helpful? Yes No.
- gist.github.com › ...**
RPC interfaces RS5 · GitHub
... pid 684 at 0x5306320L> 64 Interfaces : RPC **367abb81-9844-35f1-ad32-98f038001003** (2.0) -- C:\windows\system32\services.exe 0 -> RCcloseServiceHandle ...
- blog.f-secure.com › endpoint-detection-of-remote-servi...**
Endpoint Detection of Remote Service Creation and PsExec ...
Nov 14, 2018 - InterfaceUuid: 367abb81-9844-35f1-ad32-98f038001003 OpNum: ... SCM interface (**367abb81-9844-35f1-ad32-98f038001003**), the same one ...

MS-SCMR – Standards Assignment

 Filter by title

- ✓ Technical Documents
 - Technical Documents
 - ✓ [MS-SCMR]: Service Control Manager Remote Protocol
 - [MS-SCMR]: Service Control Manager Remote Protocol
 - ✓ 1 Introduction
 - 1 Introduction
 - 1.1 Glossary
 - > 1.2 References
 - 1.3 Overview
 - 1.4 Relationship to Other Protocols

1.9 Standards Assignments

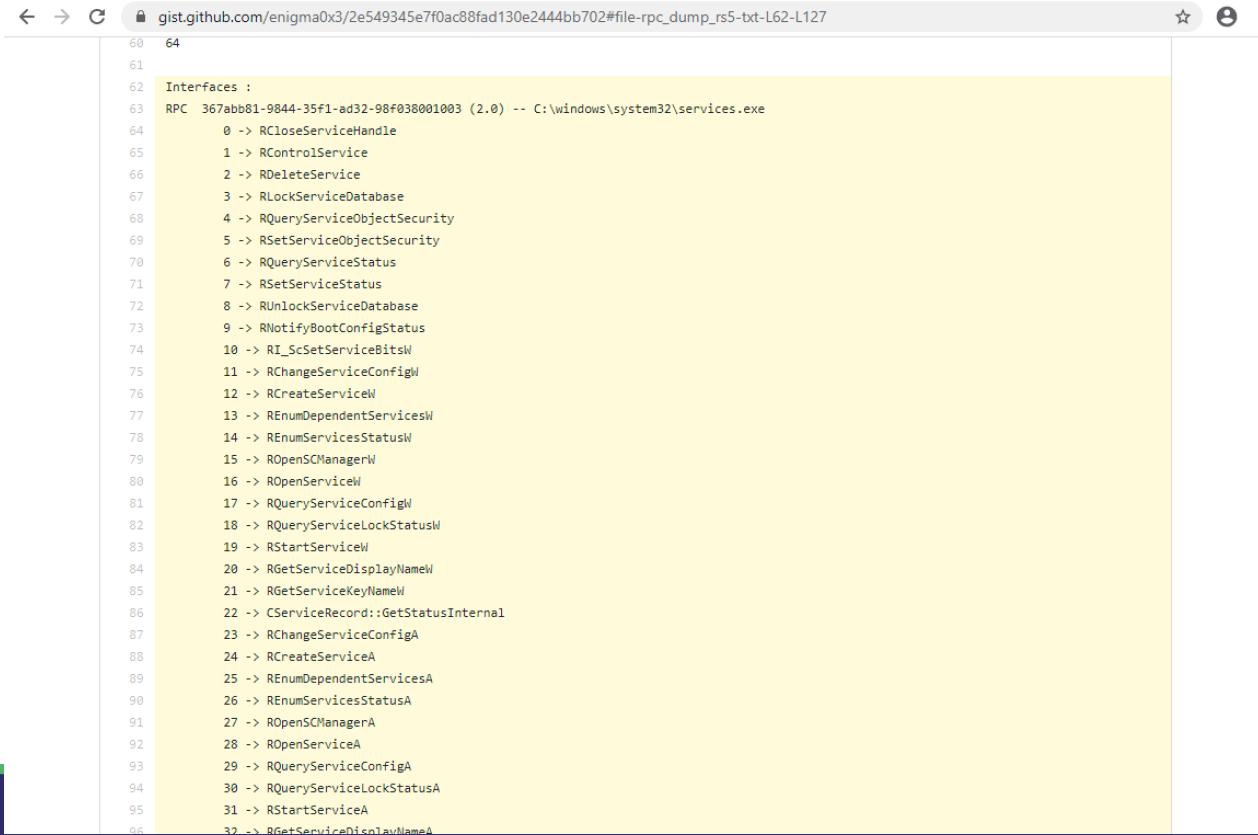
02/14/2019 • 2 minutes to read

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

Enigma0x3 – RPC Interface Inventory



```
60 64
61
62 Interfaces :
63 RPC 367abb81-9844-35f1-ad32-98f038001003 (2.0) -- C:\windows\system32\services.exe
64     0 -> RCloseServiceHandle
65     1 -> RControlService
66     2 -> RDeleteService
67     3 -> RLockServiceDatabase
68     4 -> RQueryServiceObjectSecurity
69     5 -> RSetServiceObjectSecurity
70     6 -> RQueryServiceStatus
71     7 -> RSetServiceStatus
72     8 -> RUnlockServiceDatabase
73     9 -> RNotifyBootConfigStatus
74    10 -> RI_ScSetServiceBitsW
75    11 -> RChangeServiceConfigW
76    12 -> RCreateServiceW
77    13 -> REnumDependentServicesW
78    14 -> REnumServicesStatusW
79    15 -> ROpenSCManagerW
80    16 -> ROpenServiceW
81    17 -> RQueryServiceConfigW
82    18 -> RQueryServiceLockStatusW
83    19 -> RStartServiceW
84    20 -> RGetServiceDisplayNameW
85    21 -> RGetServiceKeyNameW
86    22 -> CServiceRecord::GetStatusInternal
87    23 -> RChangeServiceConfigA
88    24 -> RCreateServiceA
89    25 -> REnumDependentServicesA
90    26 -> REnumServicesStatusA
91    27 -> ROpenSCManagerA
92    28 -> ROpenServiceA
93    29 -> RQueryServiceConfigA
94    30 -> RQueryServiceLockStatusA
95    31 -> RStartServiceA
96    32 -> RGetServiceDisplayNameA
```

Enigma0x3 – RPC Interface Inventory

```
60 64
61
62 Interfaces :
63 RPC 367abb81-9844-35f1-ad32-9bf038001003 (2.0) - C:\windows\system32\services.exe
64     0 -> RCcloseServiceHandle
65     1 -> RControlService
66     2 -> RDeleteService
67     3 -> RLockServiceDatabase
68     4 -> RQueryServiceObjectSecurity
69     5 -> RSetServiceObjectSecurity
70     6 -> RQueryServiceStatus
71     7 -> RSetServiceStatus
72     8 -> RUnlockServiceDatabase
73     9 -> RNotifyBootConfigStatus
74    10 -> RI_ScSetServiceBitsW
75    11 -> RChangeServiceConfigW
76    12 -> RCreateServiceW
77    13 -> REnumDependentServicesW
78    14 -> REnumServicesStatusW
79    15 -> ROpenSCManagerW
80    16 -> ROpenServiceW
81    17 -> RQueryServiceConfigW
82    18 -> RQueryServiceLockStatusW
83    19 -> RStartServiceW
84    20 -> RGetServiceDisplayNameW
85    21 -> RGetServiceKeyNameW
86    22 -> CServiceRecord::GetStatusInternal
87    23 -> RChangeServiceConfigA
88    24 -> RCreateServiceA
89    25 -> REnumDependentServicesA
90    26 -> REnumServicesStatusA
91    27 -> ROpenSCManagerA
92    28 -> ROpenServiceA
93    29 -> RQueryServiceConfigA
94    30 -> RQueryServiceLockStatusA
95    31 -> RStartServiceA
96    32 -> RGetServiceDisplayNameA
```

NtObjectManager – RPC Server

```
Administrator: Windows PowerShell
PS C:\Windows\system32> $ServicesRPC = ls C:\Windows\System32\services.exe | Get-RpcServer
PS C:\Windows\system32> $ServicesRPC | Format-List
```

```
InterfaceId      : 367abb81-9844-35f1-ad32-98f038001003
InterfaceVersion : 2.0
TransferSyntaxId : 8a885d04-1ceb-11c9-9fe8-08002b104860
TransferSyntaxVersion : 2.0
ProcedureCount   : 64
Procedures       : {Proc0, Proc1, Proc2, Proc3...}
Server           : UUID: 367abb81-9844-35f1-ad32-98f038001003
ComplexTypes     : {Struct_0, Struct_1, Struct_2, Struct_3...}
FilePath         : C:\Windows\System32\services.exe
Name             : services.exe
Offset           : 501168
ServiceName      :
ServiceDisplayName :
IsServiceRunning : False
Endpoints        : {}
EndpointCount    : 0
Client           : False
```

```
InterfaceId      : a2c45f7c-7d32-46ad-96f5-adafb486be74
InterfaceVersion : 1.0
TransferSyntaxId : 8a885d04-1ceb-11c9-9fe8-08002b104860
TransferSyntaxVersion : 2.0
ProcedureCount   : 3
Procedures       : {Proc0, Proc1, Proc2}
Server           : UUID: a2c45f7c-7d32-46ad-96f5-adafb486be74
ComplexTypes     : {Struct_0, Struct_1, Union_2, Struct_3...}
FilePath         : C:\Windows\System32\services.exe
Name             : services.exe
Offset           : 501264
ServiceName      :
ServiceDisplayName :
IsServiceRunning : False
Endpoints        : {}
EndpointCount    : 0
Client           : False
```

SERVICES.EXE – Exports Table

IDA - services.exe C:\Users\analyst\Desktop\New Service Demo\services.exe

File Edit Jump Search View Debugger Options Windows Help

Local Windows debugger

Library function Regular function Instruction Data Unexplored External symbol

Functions window IDA View-A Hex View-1 Structures Enums Imports Exports

Name Address Ordinal

start 00000000431450 [main entry]

ScCompareVector

- ↳ SdGetElementGroupRecord(ushort const *)
- ↳ SdGetUniqueTag(ushort *,ulong)
- ↳ SdSdsGroupOrdered(ushort const *)
- ↳ SdUpdateLegacyGroupPointer(CServiceRecord *,ulong)
- ↳ REnumServiceGroups(x,x,x,x,x,x,x,x)
- ↳ REnumServiceStatuses(x,x,x,x,x,x,x)
- ↳ SdFindAllocForDatabase(ushort *)
- ↳ SdGetLockOwner(void *userHandle *)
- ↳ RLockServiceDatabase(x,x)
- ↳ RQueryServiceLockStatus(x,x,x,x)
- ↳ RUUnloadServiceDatabase(x)
- ↳ SC_RPC_LOCK_rundown(x)
- ↳ WPR_SF_dedf(x,x,x,x,x)
- ↳ SqpMediaTimerCallback
- ↳ ConvertToValidProd(ushort *,ulong)
- ⋮

wl:Feature<_WIFeatureTraits_Feature_ServiceManager>

wl:Feature<_WIFeatureTraits_Feature_ServiceManager>

isValidPraIdChar(ushort)

wl:Feature<_WIFeatureTraits_Feature_ServiceManager>

ScGenerateHostID(unsigned __int64 *)

SomAcquireResourceSet((Win32ServiceRecord *)R,

SomCreateRHost(Win32ServiceRecord *,void *R,

SomRHostSupport(void),

SomUnregisterDefinitions(_IMAGE_RELOC),

SomRkHostActivityCallback(_PM_ACTIVITI_HOST_CALLBACK),

SomRSystemMemoryChangeCallback(_WNF_SYSTEM_MEMORY_CHANGE_CALLBACK),

SopSendLocalSystemResourceMessage(void)

WPR_SF_Dsdd(x,x,x,x,x,x)

WPR_SF_Dsfs(x,x,x,x,x,x)

RChangeServiceConfig2(x,x,x)

RChangeServiceConfig4(x,x,x,x,x,x,x,x,x,x,x)

RControlServiceExA(x,x,x,x)

RCreateServiceA(x,x,x,x,x,x,x,x,x,x,x)

Line 1135 of 1434

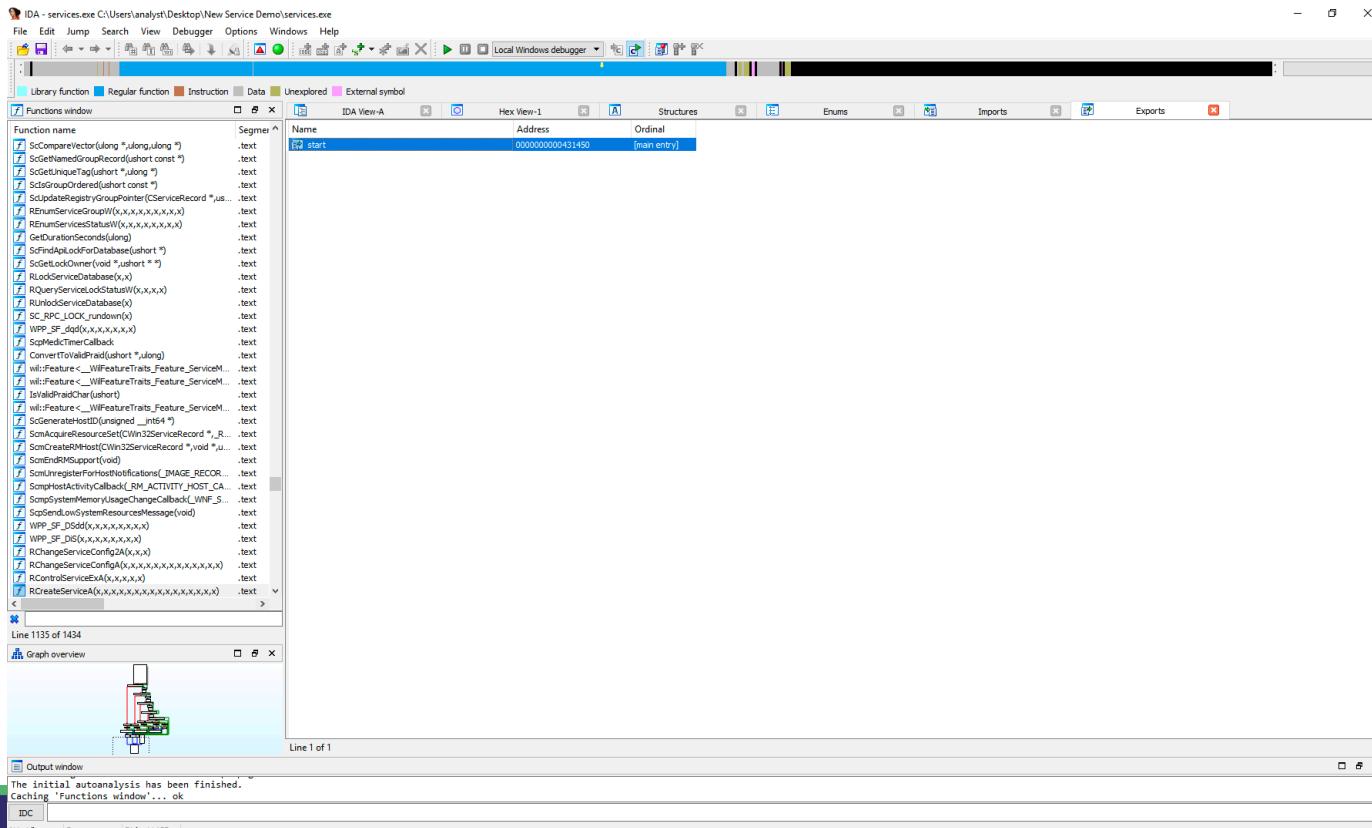
Graph overview

The initial autoanalysis has been finished.

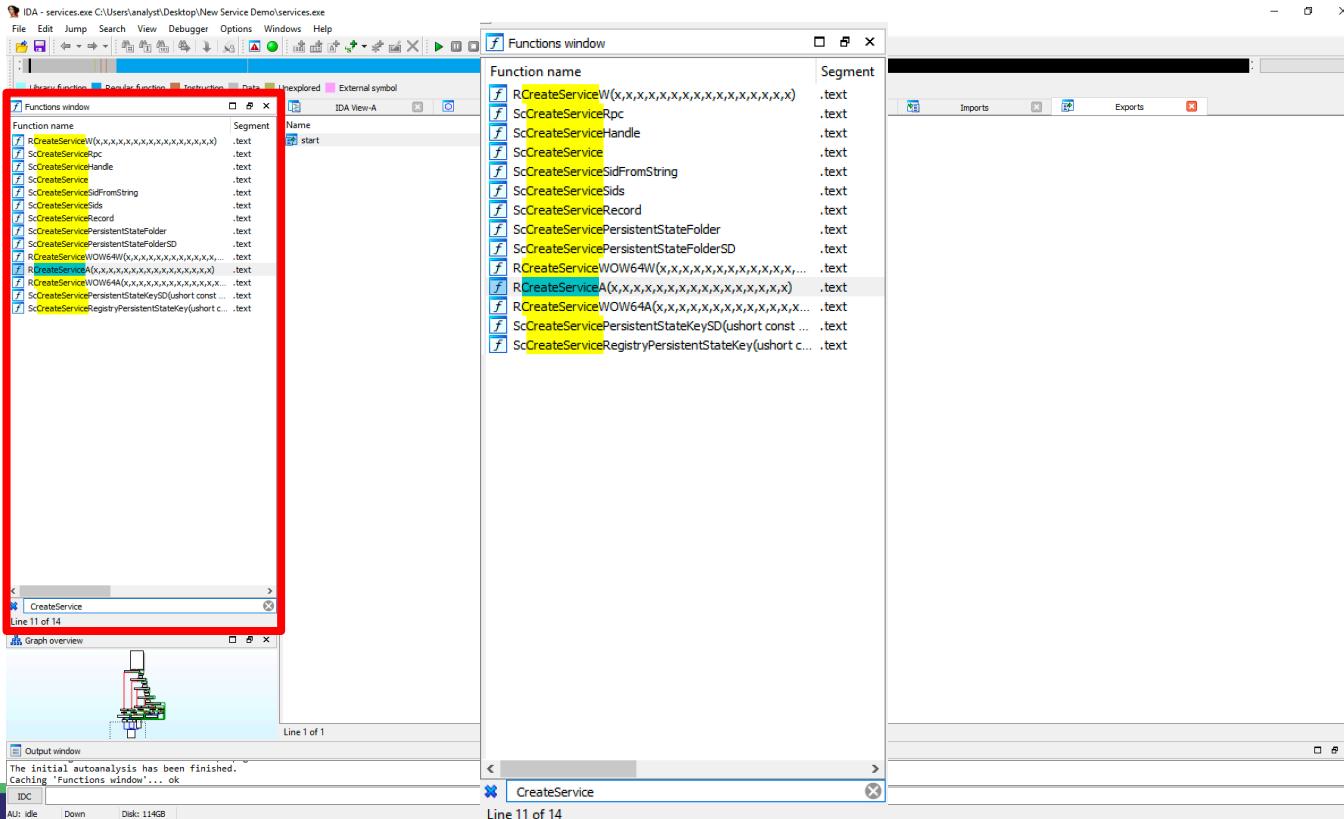
Caching 'Functions window'... ok

IDC

AU: idle Down Disk: 114GB



SERVICES.EXE – Sub Procedures



SERVICES.EXE - RCreateServiceW

The screenshot shows the IDA Pro debugger interface with the assembly view selected. The assembly window displays the code for the `RCreateServiceW` function. The code is written in Intel x86 assembly language. The stack frame is bp-based, and various registers (edi, ebp, esp) are used to hold pointers to parameters passed by reference. The function starts by pushing the current stack frame onto the stack. It then pushes several parameters onto the stack, including the service handle, service name, display name, start type, error control, binary path, and load order group. Finally, it calls the `ScCreateServiceRpc` function.

```
; Attributes: bp-based frame
; ...
int __stdcall RCreateServiceW(int hSCManager, int lpServiceName, int lpDisplayName, int dwDesiredAccess, int dwServiceType, int dwStartType, int dwErrorControl, int lpBinaryPathName, int lpLoadOrderGroup, int dwWaitHint)
{
    _RCreateService@4 proc near

        hSCManager= dword ptr  8
        lpServiceName= dword ptr  0Ch
        lpDisplayName= dword ptr  10h
        dwDesiredAccess= dword ptr  14h
        dwServiceType= dword ptr  18h
        dwStartType= dword ptr  1ch
        dwErrorControl= dword ptr  20h
        lpBinaryPathName= dword ptr  24h
        lpLoadOrderGroup= dword ptr  28h
        lpdwTagId= dword ptr  2Ch
        lpDependencies= dword ptr  30h
        dwDependSize= dword ptr  34h
        lpServiceHandle= dword ptr  38h
        lpPassword= dword ptr  3Ch
        dwWaitSize= dword ptr  40h
        lpServiceHandle= dword ptr  44h

        mov     edi, edi
        push    ebp
        mov     ebp, esp
        push    0
        push    [ebp+lpServiceHandle]; lpServiceHandle
        mov     edx, [ebp+lpServiceName]; lpServiceName
        push    [ebp+hSCManager]; hSCManager
        push    [ebp+lpPassword]; lpPassword
        push    [ebp+lpServiceStartName]; lpServiceStartName
        push    [ebp+dwDependSize]; dwDependSize
        push    [ebp+lpDependencies]; lpDependencies
        push    [ebp+lpdwTagId]; lpdwTagId
        push    [ebp+lpLoadOrderGroup]; lpLoadOrderGroup
        push    [ebp+lpBinaryPathName]; lpBinaryPathName
        push    [ebp+dwErrorControl]; dwErrorControl
        push    [ebp+dwStartType]; dwStartType
        push    [ebp+dwServiceType]; dwServiceType
        push    [ebp+dwDesiredAccess]; dwDesiredAccess
        push    [ebp+lpDisplayName]; lpDisplayName
        call    ScCreateServiceRpc
        pop    ebp
        retn   40h
    _RCreateService@4 endo ; synchronization failed
}
```

Output window:
Database 'services.exe' has been loaded.
Caching 'Functions window... ok'

SERVICES.EXE - ScCreateServiceRpc

The screenshot shows the IDA Pro debugger interface with the file 'services.exe' loaded. The assembly code for the function `ScCreateServiceRpc` is displayed in the main window. The code uses various registers (eax, edx, ecx, ebx, esp, esi, edi) and memory locations to perform operations such as `ScGrantAccess`, `ScCreateServiceHandle`, and `ScCreateService`. The assembly code is annotated with comments and assembly mnemonics. A graph overview window on the left shows the flow of control between different parts of the function. The bottom status bar indicates the assembly code is synchronized with the hex view.

```
lea    edx, [esp+20h+var_10]
xor    ecx, ecx
call   ScCreateServiceHandle
mov    edi, [esp+20h+var_10]
mov    edx, eax
test   esi, esi
jnz   loc_434ED0

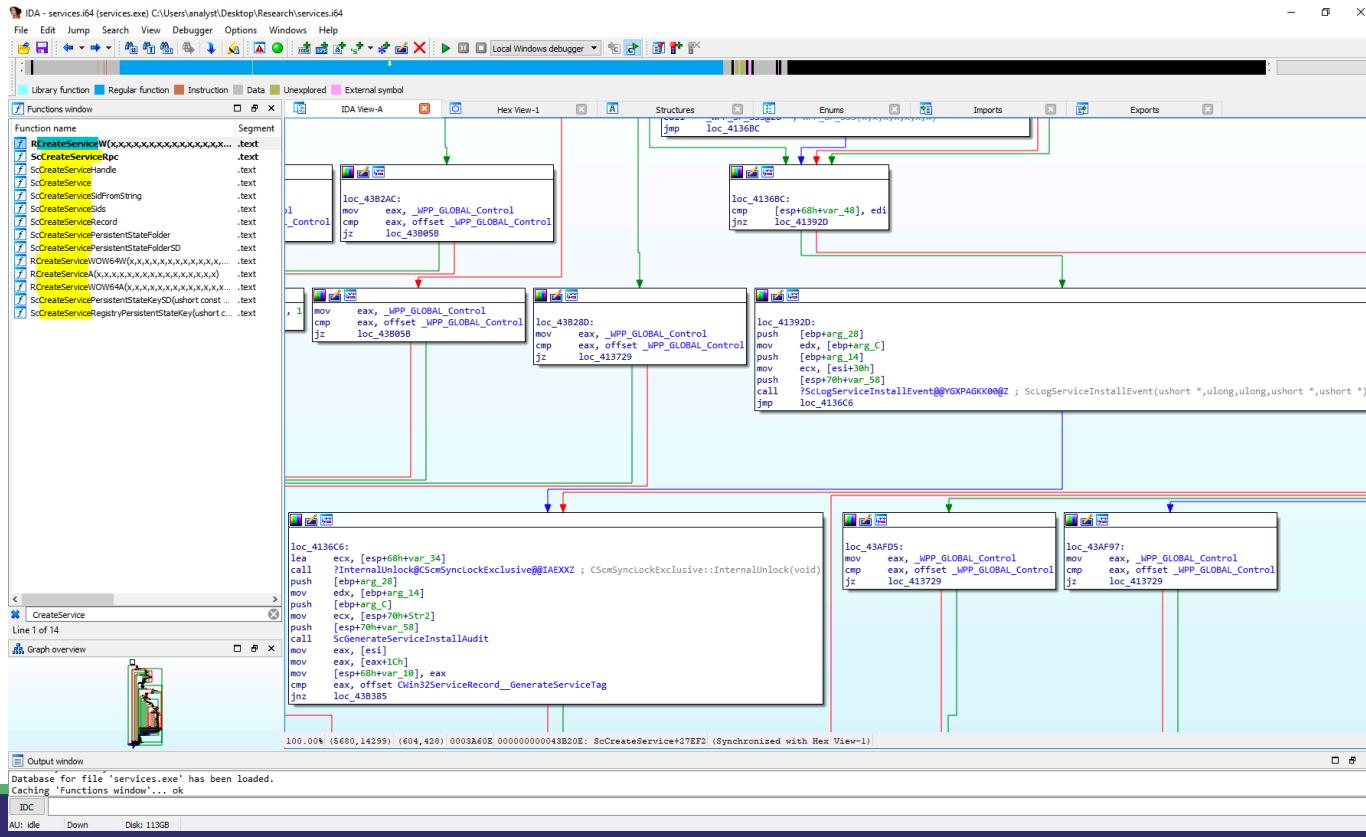
mov    edx, [ebp+dwDesiredAccess]
xor    eax, eax
push   edx, ebx
push   eax, eax ; int
push   [ebp+dwSize], edx
push   [ebp+lpPassword], edx
push   [ebp+arg_28], wchar_t *
push   [ebp+arg_20], wchar_t *
push   [ebp+arg_14], lmbtagid
push   [ebp+lpOrderGroup], int
push   [ebp+lpBinaryPathName], int
push   [ebp+dwErrorControl], int
push   [ebp+dwStartType], int
push   [ebp+dwServiceType], int
push   eax, eax ; int
push   [ebp+str1], str1
mov    ecx, [esp+eh+var_4]
call   ScCreateService
esi, edi
test   esi, esi
jnz   loc_434ED4
```

Database for file 'services.exe' has been loaded.
Caching 'Functions window'... ok

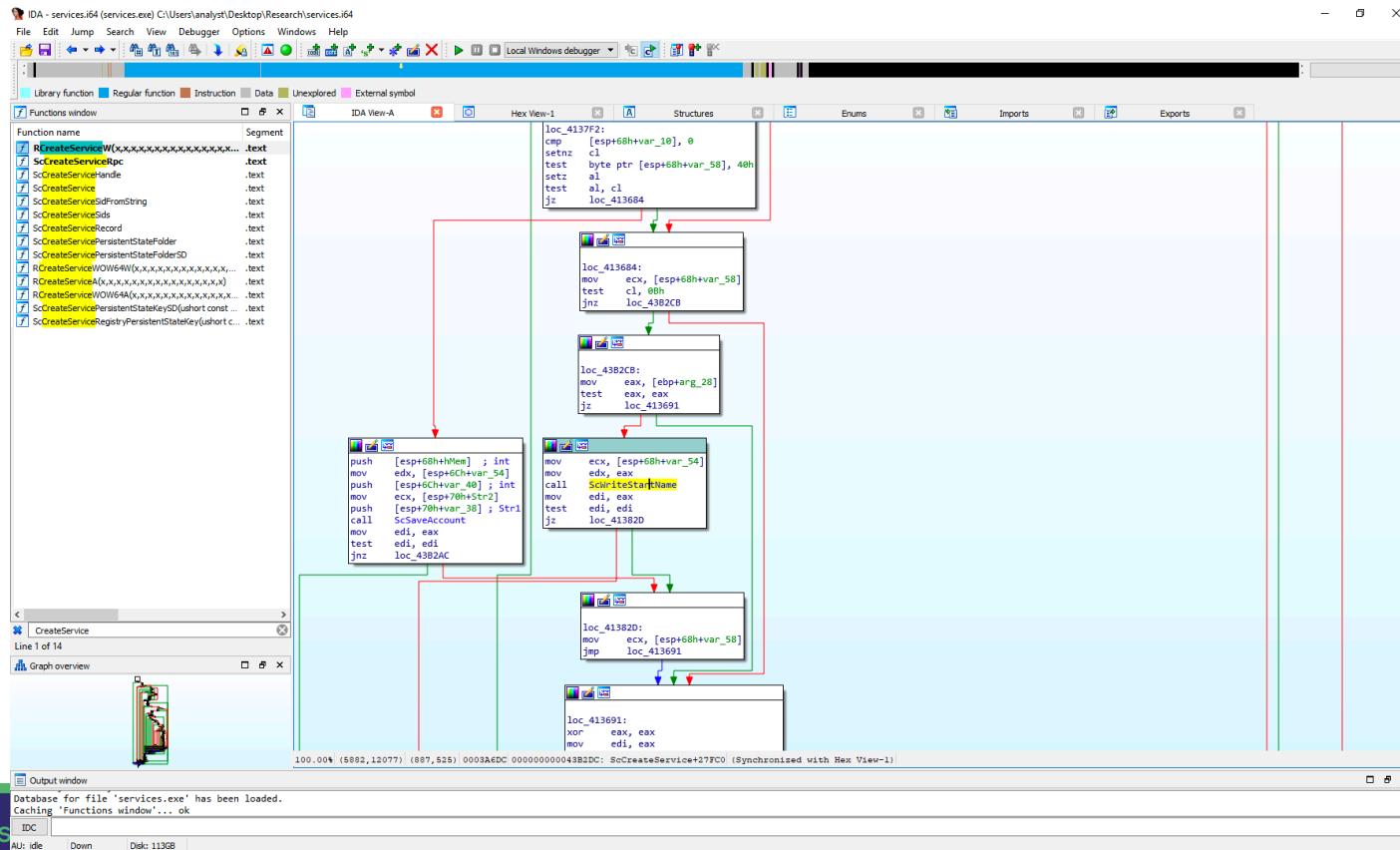
IDC

AU: idle Down Disk: 113GB

SERVICES.EXE – ScCreateService (Logging)



SERVICES.EXE – SCCreateService (Reg)



SERVICES.EXE - ScWriteStartName

IDA - services.i64 (services.exe) C:\Users\analyst\Desktop\Research\services.i64

File Edit Jump Search View Debugger Options Windows Help

Library function Regular function Instruction Data Unexplored External symbol

Functions window Segment

Function name .text

RCreateServiceW(xxxxxx...).text
ScCreateServiceRpc...
ScCreateServiceHandle...
ScCreateService...
ScCreateServiceSidFromString...
ScCreateServiceSids...
ScCreateServiceRecord...
ScCreateServicePersistentStateFolder...
ScCreateServicePersistentStateFolderSD...
RCreateServiceWOW64(x,x,x,x,x,x,x,x,x,x,...).text
RCreateServiceA(x,x,x,x,x,x,x,x,x,x,...).text
RCreateServiceWOW64A(x,x,x,x,x,x,x,x,x,x,...).text
ScCreateServicePersistentStateKeyDlshurst const ...
ScCreateServiceRegistryPersistentStateKey(ushort c...).text

; FUNCTION CHUNK AT 004352EF SIZE 00000036 BYTES

```
mov edi, edi
push esi
push edi
mov edi, edx
mov esi, edi
lea edx, [esi+2]
```

loc_40F931:

```
mov ax, [esi]
add esi, 2
test ax, ax
jnz short loc_40F931
```

loc_4352EF:

```
sub edi, edx
mov edx, offset aObjectName ; "ObjectName"
sar esi, 1
lea eax, ds:[esi*2]
push eax
push edi
push 1
push ecx
call ScRegSetValueExW
mov esi, eax
test esi, esi
jnz loc_4352EF
```

; START OF FUNCTION CHUNK FOR ScWriteStartName

loc_4352EF:

```
mov ecx, _WPP_GLOBAL_Control
cmp ecx, offset _WPP_GLOBAL_Control
jz loc_40F960
```

test byte ptr [ecx+1Ch], 1
jz loc_40F968

100.00% (-452,79) (315,682) 00000ED61 000000000040F951: ScWriteStartName+2B (Synchronized with Hex View-1)

Output window

Database for file 'services.exe' has been loaded.
Caching 'Functions window'... ok

IDC

AU: idle Down Disk: 113GB

Graph overview

125

SERVICES.EXE - ScRegSetValueExW

The screenshot shows the IDA Pro interface with the file "services.exe" loaded. The left pane displays a list of functions, and the right pane shows the assembly code for the selected function, `ScRegSetValueExW`.

Function List:

- `RCreateServiceWxxxxxxxxxxxxxx... .text`
- `ScCreateServiceRpc .text`
- `ScCreateServiceHandle .text`
- `ScCreateService .text`
- `ScCreateServiceFromString .text`
- `ScCreateServiceSdS .text`
- `ScCreateServiceRecord .text`
- `ScCreateServicePersistentStateFolder .text`
- `ScCreateServicePersistentStateSD .text`
- `RCreateServiceWOW4W(x,x,x,x,x,x,x,x,...) .text`
- `RCreateServiceA(x,x,x,x,x,x,x,x,x,x,x) .text`
- `RCreateServiceWOW4A(x,x,x,x,x,x,x,x,x,x,x) .text`
- `ScCreateServicePersistentStateSd(ushort const ...) .text`
- `ScCreateServicePersistentStateKey(uhort c... .text`

Assembly View:

```
; Attributes: bp-based frame
ScRegSetValueExW proc near
var_8=byte ptr -8
arg_4=dword ptr 0Ch
arg_0=dword ptr 10h
arg_C=dword ptr 14h

; FUNCTION CHUNK AT 0043AD22 SIZE 00000016 BYTES

mov edi, edi
push ebp
mov ebp, esp
push ecx
push ecx
push esi
push edi
mov edi, edx
lea eax, [ebp+var_8]
push edi
push eax
push mov esi, ecx
call ds:_imp__RtlInitUnicodeString@0 ; RtlInitUnicodeString(x,x)
push [ebp+arg_4]
lea eax, [ebp+var_8]
push eax
push [ebp+arg_4]
push [ebp+arg_4]
push 0
push eax
push esi
call ds:_imp__NtSetValueKey@24 ; NtSetValueKey(x,x,x,x,x)
push eax
call ds:_imp__RtlNtStatusToDosError@4 ; RtlNtStatusToDosError(x)
mov esi, eax
test esi, esi
jnc loc_43AD22
```

Call Graph:

A call graph overview window is visible in the bottom-left corner, showing the flow of control between different parts of the program.

Output Window:

```
Database for file 'services.exe' has been loaded.
Caching 'Functions window'... ok
```

IDC:

```
100.00% (-473,-31) (950,617) 00012463 0000000000413063: ScRegSetValueExW (Synchronized with Hex View-1)
```

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

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

Registered Provider Enumeration

```
C:\>Administrator: Command Prompt  
  
C:\>logman.exe 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}
```

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

Computer Name \$null or . resolve to LocalHost

Resolving to default scheme http

Remote shell name resolved to default Microsoft.PowerShell

%3Context:%1User Data:%2

%3Context:%1User Data:%2

%3Context:%1User Data:%2

%3Context:%1User Data:%2

Creating Scriptblock text (%1 of %2):%3Script Block ID: %4P...

Fields

ContextInfo,UserData,Payload

ContextInfo,UserData,Payload

ContextInfo,UserData,Payload

ContextInfo,UserData,Payload

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...

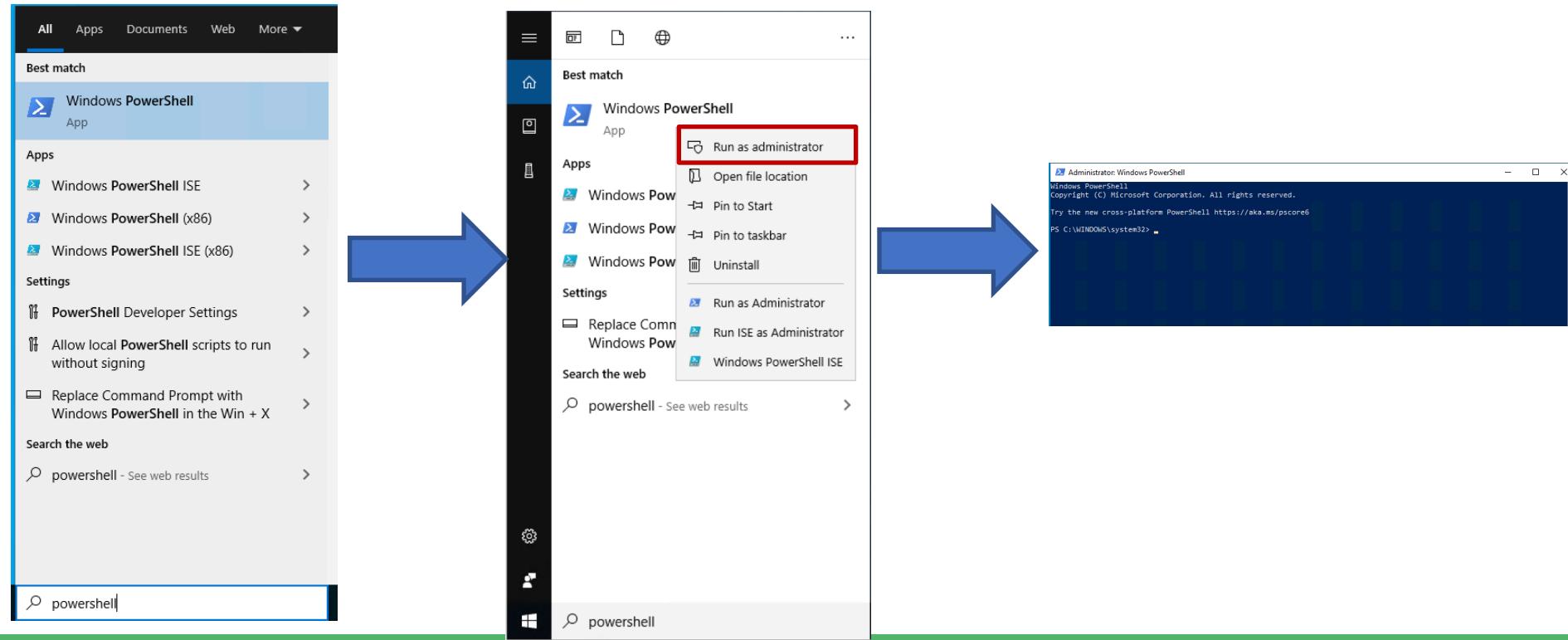
- Not real-time, manual creation of trace session with subscriber

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
EventLog-System               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
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            {F3395984-DBEC-11D2-895B-00C04F79AB69}
ADODB.1                                {04C8A86F-3369-12F8-4769-24E484A9E725}
ADOMD.1                                {7EA56435-3F2F-3F63-A829-F0B3585CAD41}
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}
Classppnp Driver Tracing Provider     {FA8DE7C4-ACDE-4443-9994-C4E2359A9EDB}
CriticalSection Trace Provider          {3AC66736-CC59-4CFF-8115-80F50E39816B}
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%

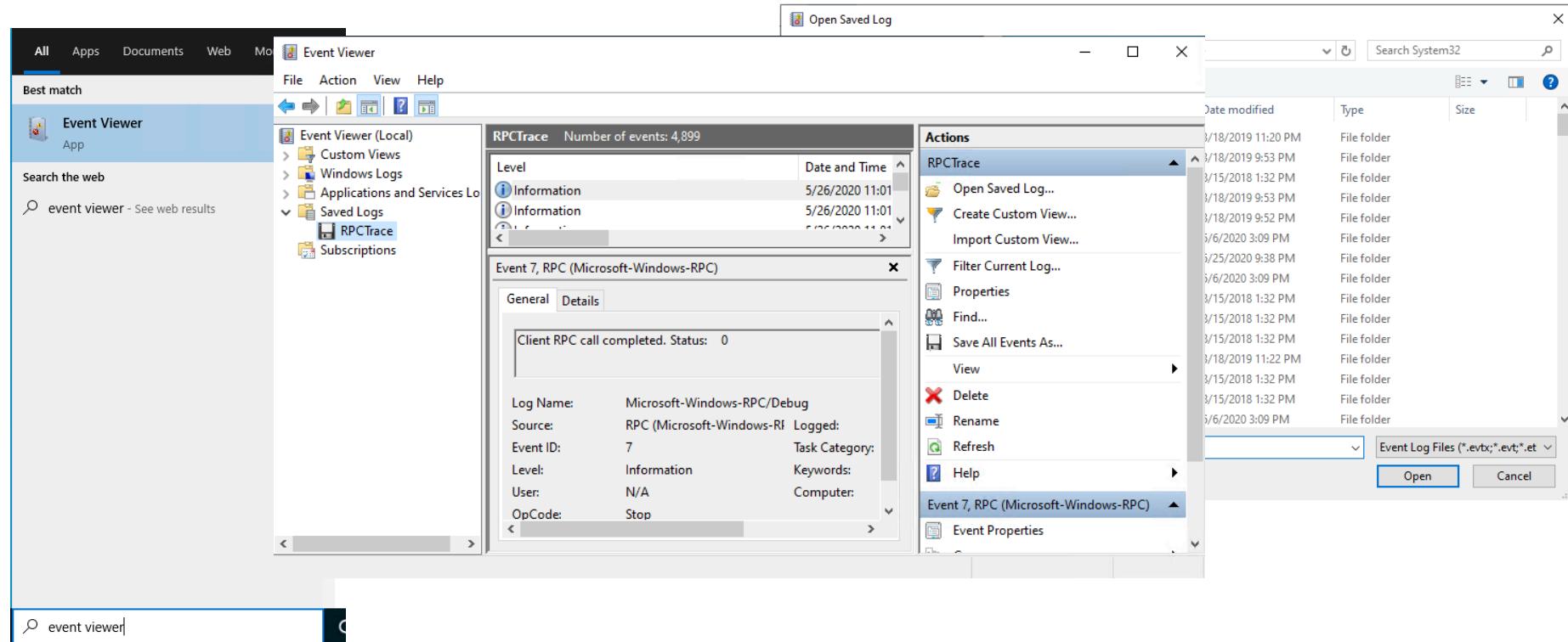
Output
-----
DumpFile:          RPCTrace.evtx
```

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

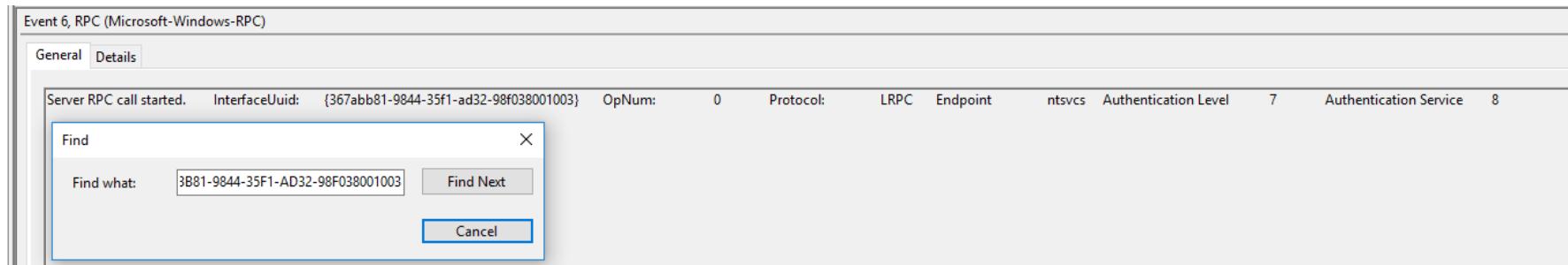
Status   Name           DisplayName
----   --
Stopped Test_Service  Test_Service

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

Uploading RPCTrace.evtx - Logman ETW



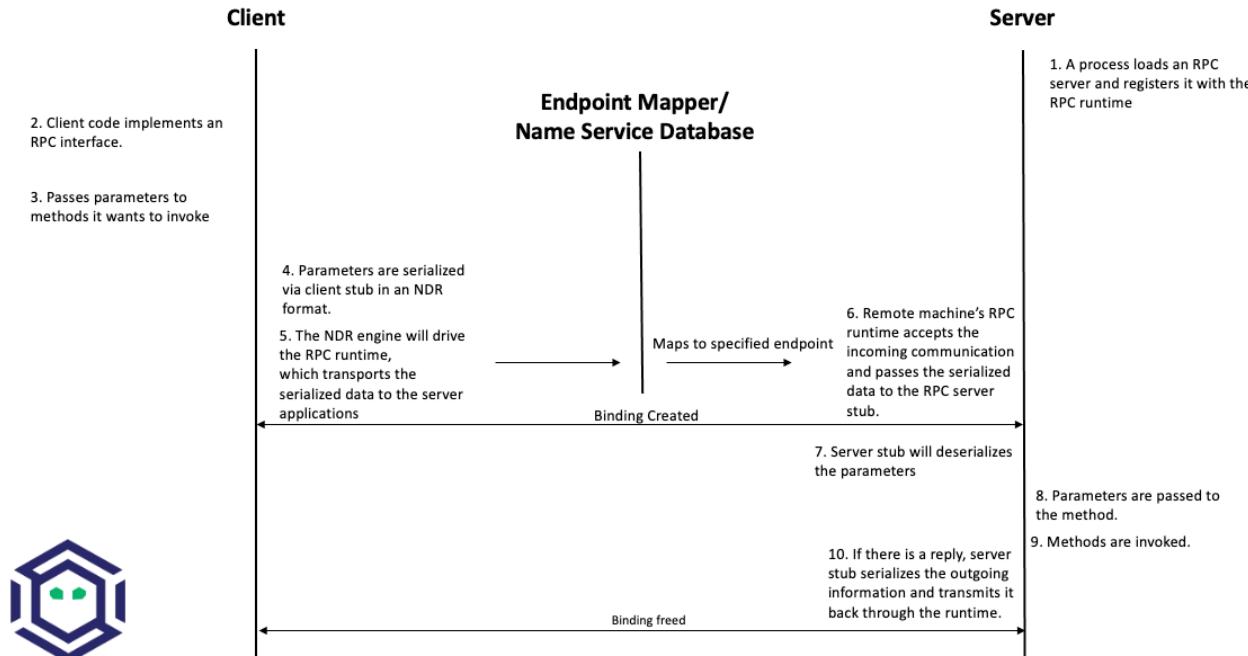
Looking for Service Creation – Logman ETW



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	367ABB81-9844-35F1-AD32-98F038001003
Interface/Named Pipe	[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

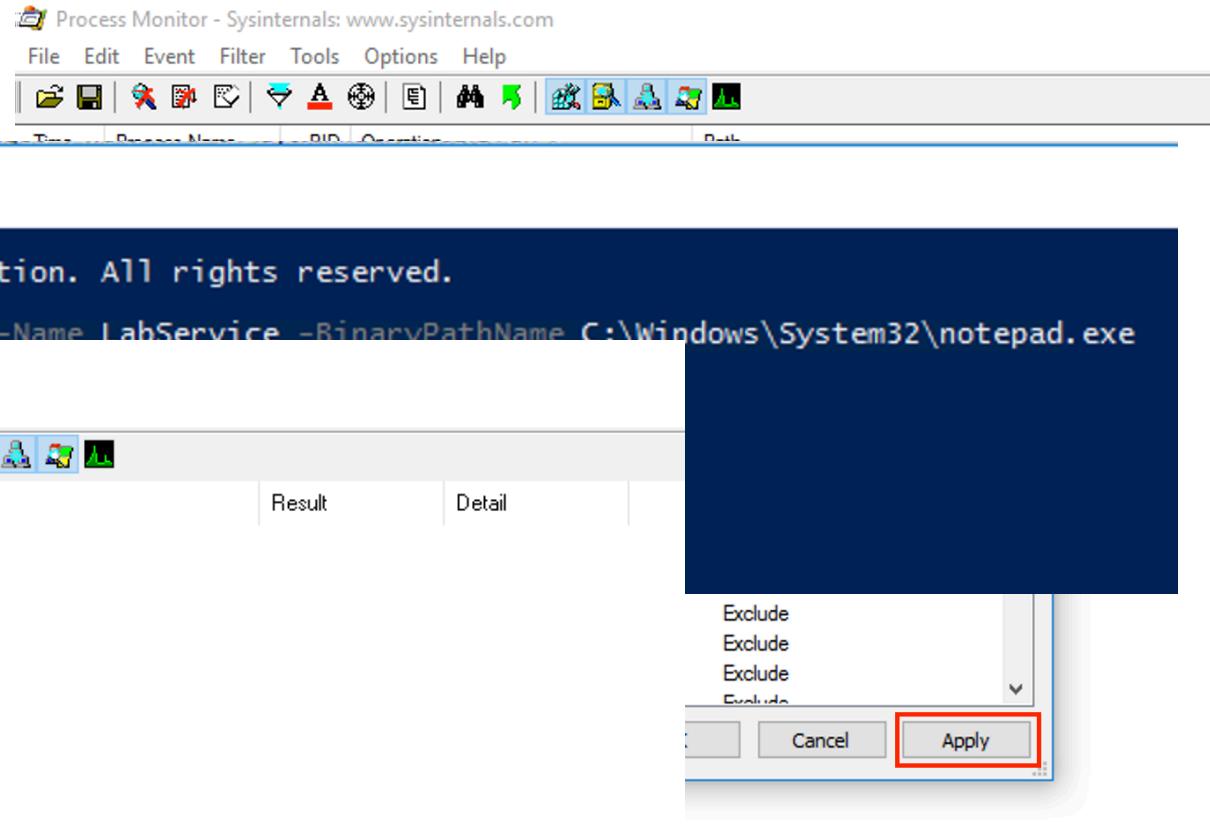
Highlight... Ctrl+H

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 item | Showing 376,193 of 500,562 events (75%) | Backed by virtual memory

Lab - Procmon



Lab - Procmon

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Path

Reset

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-PSEexec	CSEexec	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\svctrl						\\\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
- PSEExec
 - A popular administrative tool to execute code **remotely**
- Try everything on a remote system and see what is different!