

Product ▾Solutions ▾Resources ▾Open Source ▾Enterprise ▾Pricing

🔍

Sign in

Sign up

📄redcanaryco / atomic-red-teamPublic

🔔Notifications

 Fork2.8k

 Star9.7k

<>Code

🕒Issues6

🔗Pull requests5


🎬Actions

📖Wiki

🛡️Security

📊Insights

atomic-red-team / atomics / T1574.011 / T1574.011.md

 CircleCI Atomic Red Team doc...

Generate docs from job=genera...

7091fa8 · 2 years ago

🕒History

T1574.011 - Services Registry Permissions Weakness

Description from ATT&CK

Adversaries may execute their own malicious payloads by hijacking the Registry entries used by services. Adversaries may use flaws in the permissions for Registry keys related to services to redirect from the originally specified executable to one that they control, in order to launch their own code when a service starts. Windows stores local service configuration information in the Registry under `HKLM\SYSTEM\CurrentControlSet\Services`. The information stored under a service's Registry keys can be manipulated to modify a service's execution parameters through tools such as the service controller, `sc.exe`, [PowerShell] (<https://attack.mitre.org/techniques/T1059/001>), or [Reg] (<https://attack.mitre.org/software/S0075>). Access to Registry keys is controlled through access control lists and user permissions. (Citation: Registry Key Security)(Citation: malware_hides_service)

If the permissions for users and groups are not properly set and allow access to the Registry keys for a service, adversaries may change the service's `binPath/ImagePath` to point to a different executable under their control. When the service starts or is restarted, then the adversary-controlled program will execute, allowing the adversary to establish persistence and/or privilege escalation to the account context the service is set to execute under (local/domain account, SYSTEM, LocalService, or NetworkService).

Adversaries may also alter other Registry keys in the service's Registry tree. For example, the `FailureCommand` key may be changed so that the service is executed in an elevated context anytime the service fails or is intentionally corrupted.(Citation: Kansa Service related collectors)(Citation: Tweet Registry Perms Weakness)

The `Performance` key contains the name of a driver service's performance DLL and the names of several exported functions in the DLL.(Citation: microsoft_services_registry_tree) If the `Performance` key is not already present and if an adversary-controlled user has the `Create Subkey` permission, adversaries may create the `Performance` key in the service's Registry tree to point to a malicious DLL. (Citation: insecure_reg_perms)

Files

f339e7d

🔍

Go to file

> .github

> atomic_red_team

atomic-red-team / atomics / T1574.011 / T1574.011.md ↑ Top


Preview


Code

Blame

105 lines (56 loc) · 5.04 KB




Raw





⋮

```
troj_zegost) Additionally, if adversaries launch their malicious services using
svchost.exe, the service's file may be identified using
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\servicename\Parameter
s\ServiceDll .(Citation: malware_hides_service)
```

- ▼  atomics
- >  Indexes
- >  T1003.001
- >  T1003.002
- >  T1003.003
- >  T1003.004
- >  T1003.005
- >  T1003.006
- >  T1003.007
- >  T1003.008
- >  T1003
- >  T1006
- >  T1007
- >  T1010
- >  T1012
- >  T1014
- >  T1016
- >  T1018
- >  T1020
- >  T1021.001
- >  T1021.002
- >  T1021.003
- >  T1021.006
- >  T1027.001
- >  T1027.002
- >  T1027.004
- >  T1027
- >  T1030
- >  T1033
- >  T1036.003
- >  T1036.004
- >  T1036.005
- >  T1036.006
- >  T1036
- >  T1037.001
- >  T1037.002
- >  T1037.004
- >  T1037.005
- >  T1039
- >  T1040

Atomic Tests

- [Atomic Test #1 - Service Registry Permissions Weakness](#)
- [Atomic Test #2 - Service ImagePath Change with reg.exe](#)

Atomic Test #1 - Service Registry Permissions Weakness

Service registry permissions weakness check and then which can lead to privilege escalation with ImagePath. eg. reg add "HKLM\SYSTEM\CurrentControlSet\Services#{weak_service_name}" /v ImagePath /d "C:\temp\AtomicRedteam.exe"

Supported Platforms: Windows

auto_generated_guid: f7536d63-7fd4-466f-89da-7e48d550752a

Inputs:

Name	Description	Type	Default Value
weak_service_name	weak service check	String	weakservicename

Attack Commands: Run with powershell !

```
get-acl REGISTRY::HKLM\SYSTEM\CurrentControlSet\Services\* | FL
get-acl REGISTRY::HKLM\SYSTEM\CurrentControlSet\Services\#{weak_service_name}
```

Atomic Test #2 - Service ImagePath Change with reg.exe

Change Service registry ImagePath of a benign service to a malicious file

Supported Platforms: Windows

auto_generated_guid: f38e9eea-e1d7-4ba6-b716-584791963827

Inputs:

Name	Description	Type	Default Value
weak_service_name	weak service name	String	calcservice
weak_service_path	weak service path	String	%windir%\system32\win32calc.exe
malicious_service_path	malicious service path	String	%windir%\system32\cmd.exe

Attack Commands: Run with command_prompt !

```
reg.exe add "HKLM\SYSTEM\CurrentControlSet\Services\#{weak_service_name}
```

Cleanup Commands:

```
sc.exe delete #{weak_service_name}
```

Dependencies: Run with powershell !

Description: The service must exist ({weak_service_name})

Check Prereq Commands:

```
if (Get-Service {weak_service_name}) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
sc.exe create {weak_service_name} binpath= "{weak_service_path}"
```