T1546.011 - Application Shimming

Description from ATT&CK

> Adversaries may establish persistence and/or elevate privileges by executing malicious content triggered by application shims. The Microsoft Windows Application Compatibility Infrastructure/Framework (Application Shim) was created to allow for backward compatibility of software as the operating system codebase changes over time. For example, the application shimming feature allows developers to apply fixes to applications (without rewriting code) that were created for Windows XP so that it will work with Windows 10. (Citation: Elastic Process Injection July 2017)
> Within the framework, shims are created to act as a buffer between the program (or more specifically, the Import Address Table) and the Windows OS. When a program is executed, the shim cache is referenced to determine if the program requires the use of the shim database (.sdb). If so, the shim database uses hooking to redirect the code as necessary in order to communicate with the OS.
>
> A list of all shims currently installed by the default Windows installer (sdbinst.exe) is kept in:

> - `%WINDIR%\AppPatch\sysmain.sdb` and
> - `hklm\software\microsoft\windows nt\currentversion\appcompatflags\installedsdb`
>
> Custom databases are stored in:
>
> - `%WINDIR%\AppPatch\custom & %WINDIR%\AppPatch\AppPatch64\Custom` and
> - `hklm\software\microsoft\windows nt\currentversion\appcompatflags\custom`
>
> To keep shims secure, Windows designed them to run in user mode so they cannot modify the kernel and you must have administrator privileges to install a shim. However, certain shims can be used to Bypass User Account Control (UAC and RedirectEXE), inject DLLs into processes (InjectDLL), disable Data Execution Prevention (DisableNX) and Structure Exception Handling (DisableSEH), and intercept memory addresses (GetProcAddress).
>
> Utilizing these shims may allow an adversary to perform several malicious acts such as elevate privileges, install backdoors, disable defenses like Windows Defender, etc. (Citation: FireEye Application Shimming) Shims can also be abused to establish persistence by continuously being invoked by affected programs.

Atomic Tests

Atomic Test #1 - Application Shim Installation

Install a shim database. This technique is used for privilege escalation and bypassing user access control. Upon execution, "Installation of AtomicShim complete." will be displayed. To verify the shim behavior, run the AtomicTest.exe from the \T1546.011\bin directory. You should see a message box appear with "Atomic Shim DLL Test!" as defined in the AtomicTest.dll. To better understand what is happening, review the source code files is the \T1546.011\src directory.

Supported Platforms: Windows

auto_generated_guid: 9ab27e22-ee62-4211-962b-d36d9a0e6a18

Inputs:

Name	Description	Type	Default Value
file_path	Path to the shim database file	String	PathToAtomicsFolder\T1546.011\bin\AtomicShimx86.sdb

Attack Commands: Run with `command_prompt`! Elevation Required (e.g. root or admin)

```
sdbinst.exe #{file_path}
```

Cleanup Commands:

```
sdbinst.exe -u #{file_path} >nul 2>&1
```

Dependencies: Run with `powershell`!

Description: Shim database file must exist on disk at specified location (#{file_path})

Check Prereq Commands:

```
if (Test-Path #{file_path}) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
New-Item -Type Directory (split-path #{file_path}) -ErrorAction ignore | Out-Null
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/master/atomi
```

Description: AtomicTest.dll must exist at c:\Tools\AtomicTest.dll

Check Prereq Commands:

```
if (Test-Path c:\Tools\AtomicTest.dll) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
New-Item -Type Directory (split-path c:\Tools\AtomicTest.dll) -ErrorAction ignore
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/master/atomi
```

Atomic Test #2 - New shim database files created in the default shim database directory

Upon execution, check the "C:\Windows\apppatch\Custom" folder for the new shim database

https://www.fireeye.com/blog/threat-research/2017/05/fin7-shim-databases-persistence.html

Supported Platforms: Windows

auto_generated_guid: aefd6866-d753-431f-a7a4-215ca7e3f13d

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
Copy-Item $PathToAtomicsFolder\T1546.011\bin\T1546.011CompatDatabase.sdb C:\Window
Copy-Item $PathToAtomicsFolder\T1546.011\bin\T1546.011CompatDatabase.sdb C:\Window
```

Cleanup Commands:

```
Remove-Item C:\Windows\apppatch\Custom\T1546.011CompatDatabase.sdb -ErrorAction Ig
Remove-Item C:\Windows\apppatch\Custom\Custom64\T1546.011CompatDatabase.sdb -Error
```

Atomic Test #3 - Registry key creation and/or modification events for SDB

Create registry keys in locations where fin7 typically places SDB patches. Upon execution, output will be displayed describing the registry keys that were created. These keys can also be viewed using the Registry Editor.

https://www.fireeye.com/blog/threat-research/2017/05/fin7-shim-databases-persistence.html

Supported Platforms: Windows

auto_generated_guid: 9b6a06f9-ab5e-4e8d-8289-1df4289db02f

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
New-ItemProperty -Path HKLM:"\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompa
New-ItemProperty -Path HKLM:"\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompa
```

Cleanup Commands:

```
Remove-ItemProperty -Path HKLM:"\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppC
Remove-ItemProperty -Path HKLM:"\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppC
```