

# Process Injection and Persistence using Application Shimming

Nov 12, 2018

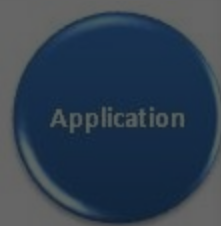
Microsoft provides Shims to developers mainly for backward compatibility, but malware can take advantage of shims to target an executable for both persistence and injection.

As the Windows operating system evolves from version to version, changes to the implementation of some functions may affect applications that depend on them.

Because of the nature of software, modifying the function again to resolve this compatibility issue could break additional applications or require Windows to remain the same regardless of the improvement that the alternative implementation could offer.

Using the Shim Infrastructure, developers can create a shim for a particular application (and its dependencies) that will be loaded by Windows outside the core Windows functionality.

The Shim Infrastructure is a framework that allows developers to create shims. Specifically, it leverages the Shim Engine, which is a component of Windows (the shim).



from <https://docs.microsoft.com/en-us/windows/desktop/shim/shim-engine>

**So, shims are essentially a way to create a shim that runs the Shim Engine and applies the appropriate fixes to the application, thus allowing it to persist and inject.**

## How can be created?

Microsoft allows anyone to create and install Shim database (sdb) files. These database files contain the specific details on how Windows should manipulate (in other words 'shim') a target program with predefined 'Fixes'. Microsoft provides also a free tool called the Application Compatibility Administrator which allows users to create and apply specific fixes such as **'DisableNX', 'ModifyShellLinkPath', 'VirtualRegistry', 'DisableAdvancedPCClientHardening', 'ForceAdminAccess', 'InjectDll', 'DisableSeh', 'ShellExecuteXP'** and many others.

After, the Application Compatibility Toolkit can be used to create and install a shim by guiding the user through a simple wizard.

The installer will create a GUID, copy the sdb file in to

```
%SystemRoot%\AppPatch\Custom\<GUID>.sdb
```

then add a registry key using an internal database name in the format of

```
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Custom\<GUID>.sdb
and
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\InstalledSDB\<GUID>.sdb
```

Obviously if a user/malware has administrative access,they could simply add the keys to the registry directly.

After the sdb file installation is complete all processes launched after that point will be subjected to the file matching rules of this shim database.

## How shim cache can be useful for a malware?

Nearly every process is vulnerable to shim injection and all modern Windows OS versions support shims and natively ship with the auto-elevated shim database installer **sdbinst.exe**.

Custom fixes can be defined in the form of a user supplied DLL file, furthermore fixes are not considered executable even though they can contain shellcode.

The shim engine (**shimeng.dll**) will not shim certain hard coded modules such as: **NT Symbolic Debugger** (NTSD), **WinDbg** or **Software License Service** (slsvc.exe) and it will intercept **GetProcAddress()** in the event an application attempts to dynamically call a function that the shim engine has manipulated.

## How analyze a shim database?

There are a few c++ parsers for Application Shimming Database (SDB) files, here is a simple python script that uses the `sdb_dump_raw.py` parser.

### Examples



`sdb_dump_raw.py`

This script dumps

```
$python sdb_dump_raw.py sdb
<INDEXES>
  <INDEX>
    <INDEX_TAG type='integer'>0x7007</INDEX_TAG>
    <INDEX_KEY type='integer'>0x600b</INDEX_KEY>
    <INDEX_FLAGS type='integer'>0x1</INDEX_FLAGS>
    <INDEX_BITS type='hex'>0000000000000000000000000000000000000000000000000000000000000000</INDEX_BITS>
  </INDEX>
  <INDEX>
    <INDEX_TAG type='integer'>0x7007</INDEX_TAG>
    <INDEX_KEY type='integer'>0x6020</INDEX_KEY>
    <INDEX_FLAGS type='integer'>0x1</INDEX_FLAGS>
    <INDEX_BITS type='hex'>0000000000000000000000000000000000000000000000000000000000000000</INDEX_BITS>
  </INDEX>
  <INDEX>
    <INDEX_TAG type='integer'>0x7007</INDEX_TAG>
    <INDEX_KEY type='integer'>0x9004</INDEX_KEY>
    <INDEX_FLAGS type='integer'>0x1</INDEX_FLAGS>
    <INDEX_BITS type='hex'>0000000000000000000000000000000000000000000000000000000000000000</INDEX_BITS>
  </INDEX>
</INDEXES>
<DATABASE>
  <OS_PLATFORM type='integer'>0x1</OS_PLATFORM>
```



andreafortuna.org asks for your consent to use your personal data to:

-  Personalised advertising and content, advertising and content measurement, audience research and services development
-  Store and/or access information on a device

Your personal data will be processed and information from your device (cookies, unique identifiers, and other device data) may be stored by, accessed by and shared with [134 TCF vendor\(s\)](#) and [63 ad partner\(s\)](#), or used specifically by this site or app.

Some vendors may process your personal data on the basis of legitimate interest, which you can object to by managing your options below. Look for a link at the bottom of this page to manage or withdraw consent in privacy and cookie settings.

[Manage your preferences](#)

[Accept all cookies](#)

```
<NAME type='stringref'>0x6</NAME>
<DATABASE_ID type='guid'>XXXXXXX-XX-XXXX-XXXX-XXXX...</DATABASE_ID>
<LIBRARY>
  <SHIM>
    <NAME type='stringref'>0x30</NAME>
    <DLLFILE type='stringref'>0x52</DLLFILE>
  </SHIM>
</LIBRARY>
<EXE>
  <NAME type='stringref'>0x7e</NAME>
  <APP_NAME type='stringref'>0x9c</APP_NAME>
  <EXE_ID type='hex'>YYYYY-YYY-YYY-YYYYY...</EXE_ID>
  <MATCHING_FILE>
    <NAME type='stringref'>0xbe</NAME>
  </MATCHING_FILE>
  <SHIM_REF>
    <NAME type='stringref'>0x30</NAME>
    <SHIM_TAGID type='integer'>0x47c</SHIM_TAGID>
  </SHIM_REF>
</EXE>
</DATABASE>
<STRINGTABLE>
  <STRINGTABLE>
  <STRINGTABLE>
  <STRINGTABLE>
  <STRINGTABLE>
  <STRINGTABLE>
  <STRINGTABLE>
  <STRINGTABLE>
</STRINGTABLE>
```

sdb\_dump\_data.bat

This script dumps the `shimdb` database and resolves value references. It can be used to generate a shim database for a specific application.



```
$python sdb_dump_data.py
<DATABASE>
  <OS_PLATFORM type='stringref'>0x1</OS_PLATFORM>
  <NAME type='stringref'>0x1</NAME>
  <DATABASE_ID type='guid'>XXXXXXX-XX-XXXX-XXXX-XXXX...</DATABASE_ID>
  <LIBRARY>
    <SHIM>
      <NAME type='stringref'>0x30</NAME>
      <DLLFILE type='stringref'>0x52</DLLFILE>
    </SHIM>
  </LIBRARY>
  <EXE>
    <NAME type='stringref'>calc.exe</NAME>
    <APP_NAME type='stringref'>XXXEngine_Apps</APP_NAME>
    <EXE_ID type='hex'>YYYYY-YYY-YYY-YYYYY...</EXE_ID>
    <MATCHING_FILE>
      <NAME type='stringref'>*</NAME>
    </MATCHING_FILE>
    <SHIM_REF>
      <NAME type='stringref'>XXXEngine_Shim</NAME>
      <SHIM_TAGID type='integer'>0x47c</SHIM_TAGID>
    </SHIM_REF>
  </EXE>
</DATABASE>
```

sdb\_dump\_shims.py

This script dumps the `DATABASE` element of a shim database, resolves value references, and substitutes complete shim definitions for `SHIM_REF` elements.



andreafortuna.org asks for your consent to use your personal data to:

-  Personalised advertising and content, advertising and content measurement, audience research and services development
-  Store and/or access information on a device

Your personal data will be processed and information from your device (cookies, unique identifiers, and other device data) may be stored by, accessed by and shared with [134 TCF vendor\(s\)](#) and [63 ad partner\(s\)](#), or used specifically by this site or app.

Some vendors may process your personal data on the basis of legitimate interest, which you can object to by managing your options below. Look for a link at the bottom of this page to manage or withdraw consent in privacy and cookie settings.

Accept

Decline

```
<DATABASE>
  <OS_PLATFORM type='integer'>0x1</OS_PLATFORM>
  <NAME type='stringref'>XXXEngine_Database</NAME>
  <DATABASE_ID type='guid'>XXXXXXX-XXXX-XXX-XXX-XX...</DATABASE_ID>
</LIBRARY>
  <SHIM>
    <NAME type='stringref'>XXXEngine_Shim</NAME>
    <DLLFILE type='stringref'>Custom\xxx.dll</DLLFILE>
  </SHIM>
</LIBRARY>
<EXE>
  <NAME type='stringref'>calc.exe</NAME>
  <APP_NAME type='stringref'>XXXEngine_Apps</APP_NAME>
  <EXE_ID type='hex'>YYYYY-YYY-YYY-YYYYY...</EXE_ID>
  <MATCHING_FILE>
    <NAME type='stringref'>*</NAME>
  </MATCHING_FILE>
  <SHIM>
    <!-- SHIM_REF name:'XXXEngine_Shim' offset:0x47c -->
    <NAME type='stringref'>XXXEngine_Shim</NAME>
    <DLLFILE type='stringref'>Custom\xxx.dll</DLLFILE>
  </SHIM>
</EXE>
</DATABASE>
```

Another useful tool

```
shims - full v
Usage
shims -listsdb
shims -stats
shims -sdb <D

Enumerate opt
-apps
-exes
-fixes
-shims
-patches
-layers
-flags
-tag <#>
-guids
-retainable
```

## Reference

- Understanding
- Microsoft Ap
- python-sdb
- Windows Sh



### andreafortuna.org asks for your consent to use your personal data to:

- Personalised advertising and content, advertising and content measurement, audience research and services development
- Store and/or access information on a device

Your personal data will be processed and information from your device (cookies, unique identifiers, and other device data) may be stored by, accessed by and shared with [134 TCF vendor\(s\)](#) and [63 ad partner\(s\)](#), or used specifically by this site or app.

Some vendors may process your personal data on the basis of legitimate interest, which you can object to by managing your options below. Look for a link at the bottom of this page to manage or withdraw consent in privacy and cookie settings.



## Andrea Fortuna

Andrea Fortuna  
[andrea@andreafortuna.org](mailto:andrea@andreafortuna.org)

- [andreafortuna](#)
- [andrea-fortuna](#)
- [andrea](#)

Cybersecurity expert, software developer,  
experienced digital forensic analyst, musician