The active use and exploitation of Microsoft's Application Compatibility Framework

Jon Erickson
- Jon Erickson (@2130706433)
- Sr. Labs Engineer at iSIGHT Partners
I’m not that Jon Erickson 😊

Although I would be happy to sign your book.
Best commercial cyber intelligence provider on the planet.

Highly Differentiated
- Forward looking, adversary focused intelligence, actionable advice
- Intelligence for multiple levels: executive, operational and technical
- Only vendor with true global intelligence collection presence

www.isightpartners.com
Agenda

- What is Application Compatibility
- Tools
- Prior Work
- EMET
- Real World Example
Fix’s Crash when alt-tab is pressed

Fix’s Crash when alt-tab is pressed

Agenda

- What is Application Compatibility
- **Tools**
- Prior Work
- EMET
- Real World Example
Tools

- Application Compatibility Toolkit (Microsoft)
- Sdb2xml (Microsoft)
- cdd (Alex Ionesceu)
- sdbinst (Microsoft)
- sdb-explorer (Jon Erickson)
Application Compatibility Toolkit

- Used to create and view SDB files
- Created by Heath Stewart (2007)
- Can dump patch_bits information
- Does not parse in-memory fix its.
Compatibility Database Dumper (CDD)

Compatibility Database Dumper (CDD) v1.0
Copyright (C) 2007 Alex Ionescu

http://www.alex-ionescu.com

usage: cdd.exe [-s][-e][-l][-f][-p][-d kernel-mode database file][-a user-mode database file]
-s Show shims
-e Show executables
-l Show layers
-f Show flags
-p Show patches
-d Use Blocked Driver Database from this path
-a Use Application Compatibility Database from this path
Installing SDB Files

sdbinst [-?] [-q] [-u] [-g] [-p] [-n[:WIN32|WIN64]] myfile.sdb | {guid} | "name"

-? - print this help text.
-p - Allow SDBs containing patches.
-q - Quiet mode: prompts are auto-accepted.
-u - Uninstall.
-g {guid} - GUID of file (uninstall only).
-n "name" - Internal name of file (uninstall only).

NOTE: Requires Administrator privileges

© Copyright 2014, iSIGHT Partners, Inc. All Rights Reserved

www.isightpartners.com
Installing SDB Files

- **Registry Locations**
  - HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Custom
  - HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\InstalledSDB

- **Default File Locations**
  - C:\Windows\AppPatch\Custom\
  - C:\Windows\AppPatch\Custom\Custom64\n
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Default)</td>
<td>REG_SZ</td>
<td>(value not set)</td>
</tr>
<tr>
<td>DatabaseDescription</td>
<td>REG_SZ</td>
<td>MSXML5: CVE-2012-1889</td>
</tr>
<tr>
<td>DatabaseInstallTimeStamp</td>
<td>REG_QWORD</td>
<td>0x1ceab146904e220 (130229543389946400)</td>
</tr>
<tr>
<td>DatabasePath</td>
<td>REG_SZ</td>
<td>C:\Windows\AppPatch\Custom\91d42a30-5434-48bc-9620-c00936f38898.sdb</td>
</tr>
<tr>
<td>DatabaseType</td>
<td>REG_DWORD</td>
<td>0x00010000 (65536)</td>
</tr>
</tbody>
</table>
Installing SDB Files

sdb-explorer.exe -r filename.sdb [-a application.exe]

- Does NOT show up in Add remove programs
- Does NOT copy SDB to default location
- Requires Administrator privileges

Note regarding 64bit Patches:

The path of the SDB file MUST contain Custom64
Agenda

- What is Application Compatibility
- Tools
- Prior Work
- EMET
- Real World Example
Prior Work

- **Alex Ionesceu**
  - Secrets of the Application Compatibility Database (SDB)
- **Mark Baggett**
  - Windows – Own3d by Default
- **Jon Erickson**
  - Persist-It – Using and Abusing Microsoft Fix It Patches
- **Chris Graham**
  - Shimming Your Way Past UAC
Secrets of the Application Compatibility Database (SDB) - Alex Ionesceu

1) Introduction
2) System Shims – The Most Interesting Ones
3) The Private Shim Engine Interface With The PE Loader
4) Built-in Shimmed Applications and Specific Shims – A Sample Never Released:
5) Tool 1 – CDD – Compatibility Database Dumper
6) Flag Shims – LUA and Installer Flags
7) The Run-Time In-Memory Patching Behavior and Analysis
8) The System Blocked Driver Database – The Kernel Side of SDB
9) Conclusion and Tool 2
System Shims

- C:\Windows\AppPatch\n  - sysmain.sdb
  - drvmain.sdb
  - msimain.sdb
  - pcamain.sdb
System Shims

- `sdb-explorer -t sysmain.sdb`

```
27650 TAG 7004 - SHIM
27656 TAG 6001 - NAME: ShowWindowIE
2765c TAG 600a - DLLFILE: AcGenral.DLL
27662 TAG 9010 - FIX_ID: {A10EB3FC-E61C-4651-845F-ABC35EF40C25}
27678 TAG 1002 - GENERAL
2767a TAG 4028 - DESCRIPTION_RC_ID: 60192 (0xeb20)
27680 TAG 7004 - SHIM
27686 TAG 6001 - NAME: Shrinker
2768c TAG 600a - DLLFILE: AcLayers.DLL
27692 TAG 9010 - FIX_ID: {17F7E7EF-065A-4424-9A7F-07D5DBA9DFE2}
276a8 TAG 1002 - GENERAL
276aa TAG 4028 - DESCRIPTION_RC_ID: 60193 (0xeb21)
```
%WINDIR%\AppPatch\en-US\AcRes.dll.mui– has descriptions in its string table.

60192, "Shim which intercepts the ShowWindow API call and fixes the problem due to tabbed browsing architecture changes in IE. The window an application finds is not the top level window any more, but the child tab window. Caught the ShowWindow API which checks if the class and process name of the window is IE's tab window. And then calls the real ShowWindow on the top level parent window."

60193, "This compatibility fix fixes problems with any application that uses the Shrinker resource compression library. This library hacks resource functions in ntdll and kernel32 and redirect calls into their own function routines. But Ntdll code has different opcodes in Windows XP. The program failed to find the opcode signature and decided to cancel WriteProcessMemory call to write their redirection. Because of this, the necessary decompression of program code and resources were not executed and caused access violation. Shrinker compatibility fix resolves this by providing necessary opcode signature so the app could write those redirection into ntdll."
System Shims

- %WINDIR%\AppPatch\ja-JP\AcRes.dll.mui – has descriptions in its string table.

- **60192**, "ShowWindow API 呼び出しを途中で取得し、IE のタブ ブラウズ アーキテクチャ変更による問題を修正する shim です。アプリケーションが検索するウィンドウは最上位ウィンドウではなく、子タブのウィンドウになります。ウィンドウのクラスとプロセス名が IE のタブ ウィンドウであるかどうかを確認する ShowWindow API をキャッチしました。次に、最上位の親ウィンドウで実際の ShowWindow を呼び出します。"

- **60193**, "この互換性修正プログラムは、Shrinker リソース圧縮ライブラリを使用するアプリケーションの問題を修正します。このライブラリは、ntdll と kernel32 のリソース関数をハッキングし、呼び出しをリソース関数の関数ルーチンにリダイレクトします。ただし、Windows XP では Ntdll コードに異なるオペコードがあります。プログラムはオペコード署名の確認に失敗し、リダイレクトを書き込むための WriteProcessMemory 呼び出しの取り消しを決定しました。このため、プログラム コードとリソースの圧縮解除が実行されず、アクセス違反が発生しました。Shrinker 互換性修正プログラムは、アプリケーションがリダイレクトを ntdll に書き込めるように必要なオペコード署名を提供して、この問題を解決します。

© Copyright 2014, iSIGHT Partners, Inc. All Rights Reserved  www.isightpartners.com
Prior Work

- Alex Ionesceu
  - Secrets of the Application Compatibility Database (SDB)
- **Mark Baggett**
  - Windows – Own3d by Default
- Jon Erickson
  - Persist-It – Using and Abusing Microsoft Fix It Patches
- Chris Graham
  - Shimming Your Way Past UAC
Mark Baggett
- Windows - Owned By Default! (DerbyCon 2013)
  - Process Execution Redirection
  - API Hooking
  - Hiding in the File System
  - Hiding in the Registry
  - Disable Security Features of the OS
  - Execute Backdoors
InjectDll Details

2524a TAG 7004 - SHIM
25250 TAG 6001 - NAME: InjectDll
25256 TAG 600a - DLLFILE: AcGenral.DLL
2525c TAG 9010 - FIX_ID: {GUID}
25272 TAG 1002 - GENERAL
25274 TAG 4028 - DESCRIPTION_RC_ID: 60155 (0xeafb)

- AcGenral.dll
- NS_InjectDll::NotifyFn()

- LoadLibraryW()
RedirectEXE

26dbc TAG 7004 - SHIM
  26dc2 TAG 6001 - NAME: RedirectEXE
  26dc8 TAG 600a - DLLFILE: AcGenral.DLL
  26dce TAG 9010 - FIX_ID: {GUID}
  26de4 TAG 1002 - GENERAL
  26de6 TAG 4028 - DESCRIPTION_RC_ID: 60176 (0xeb10)

AcGenral.dll
   NS_RedirectEXE::NotifyFn()

   CreateProcessA()
   CloseHandle(hProcess)
   CloseHandle(hThread)
   ExitProcess()
Prior Work

- Alex Ionesceu
  - Secrets of the Application Compatibility Database (SDB)
- Mark Baggett
  - Windows – Own3d by Default
- Jon Erickson
  - Persist-It – Using and Abusing Microsoft Fix It Patches
- Chris Graham
  - Shimming Your Way Past UAC
Fix-It Patch Analysis

- How is this different from patches released on patch Tuesday?
  - BinDiff mshtml.dll from MS13-097 vs. MS14-010
    - 465 Different matched functions
    - 16 unmatched functions
  - Fix It Patch for CVE-2013-3893
    - 2 Changes
Preventing 0-Day Exploitation

- CVE-2014-4114/ CVE-2014-6352 (October/November 2014)
  - OLE Packager Vulnerability

- CVE-2014-0322 (February 2014)
  - IE Use After Free

- CVE-2013-3893 (September 2013)
  - IE Memory Corruption

- CVE-2012-4792 (December 2012)
  - IE Use After Free

- CVE-2012-1889 (June 2012)
  - XML Core Services
Sandworm CVE-2014-4114

PowerPoint Lure

EVIL.inf

InfDefaultInstall.exe "EVIL.inf"
Sandworm CVE-2014-4114

PowerPoint Lure

EVIL.inf

InfDefaultInstall.exe “EVIL.inf”
Steps:
• PowerPoint Loads packager.dll to handle Packager Object
• Packager.dll copies evil.inf to Temp folder
• PowerPoint Animation invokes packager.dll DoVerb command
• DoVerb command performs “right click” context menu action for install.
• Packager.dll launches InfDefaultInstall to handle “install” action.
Sandworm CVE-2014-4114 Fix

Steps:
- PowerPoint Loads packager.dll to handle Packager Object
- Packager.dll copies evil.inf to Temp folder
  - Mark file unsafe
- PowerPoint Animation invokes packager.dll DoVerb command
- DoVerb command performs “right click” context menu action for install.
- Packager.dll launches InfDefaultInstall to handle “install” action.
  - Checks for unsafe flag
  - Prompts Users

FIXED October 2014
Sandworm CVE-2014-4114 Fix

InfDefaultInstall.exe “EVIL.inf”
CVE-2014-6352

- CVE-2014-4114 Bypasses

Google found in the wild

IDLE starts
Executing special .py file

Haifei Li found and notified Microsoft
CVE-2014-6352

- CVE-2014-4114 Bypasses

Google found in the wild

IDLE starts
And executes special .py file

(Haifei Li found)
Sandworm Bypass Fix-It

1cae TAG 7007 - EXE
1cb4 TAG 6001 - NAME: POWERPNT.exe
1cba TAG 6006 - APP_NAME: POWERPNT.exe
1cc0 TAG 6005 - VENDOR: Microsoft
1cc6 TAG 9004 - EXE_ID: {D82187EB-A66D-4A6A-B6BA-0F5738B5D08E}
1cdc TAG 9011 - APP_ID: {F503FB56-18CF-4B58-80D0-02AC0D38D698}
1cf2 TAG 7008 - MATCHING_FILE
  1cf8 TAG 6001 - NAME: *
  1cfe TAG 6009 - COMPANY_NAME: Microsoft Corporation
1d04 TAG 7008 - MATCHING_FILE
  1d0a TAG 6001 - NAME: %windir%\System32\packager.dll
  1d10 TAG 5002 - BIN_FILE_VERSION: 6.1.7601.18601
  1d1a TAG 400b - PE_CHECKSUM: 79169 (0x13541)
1d20 TAG 700a - PATCH_REF
  1d26 TAG 6001 - NAME: ef1de1e8-f835-470d-819c-228118f7eb22
  1d2c TAG 4005 - PATCH_TAGID: 972 (0x3cc)
Sandworm Bypass Fix-It

- Output from sdb-explorer
With Fix It

db OCCh  ;
db OCCh  ;
db OCCh  ;
db OCCh  ;
db OCCh  ;
db OCCh  ;

; __imp__CreatePopupMenu@0  ; CreatePopupMenu()

cmp   eax, edi
mov   [ebp+var_628], eax
jl    loc_2FA59D4
call  ds::__imp__CreatePopupMenu@0  ; CreatePopupMenu()
Steps:
• PowerPoint Loads packager.dll to handle Packager Object.
• Packager.dll copies evil.inf to Temp folder.
  - Mark file unsafe (from CVE-2014-4114 Fix).
• PowerPoint Animation invokes packager.dll DoVerb command.
  • DoVerb command performs “right click” context menu action for edit.
    • Prompts User
• Packager.dll launches IDLE to handle “Edit with IDLE” action.
• IDLE Looks for python file with specific name and executes it.
  • Doesn’t care about unsafe flag.

FIXED November 2014
CVE-2014-6352 Fix

IDLE starts Executing evil2.py
CVE-2014-6352 Fix

PowerPoint Lure

EVIL.py

IDLE starts
Executing evil2.py
Application Compatibility Toolkit has no concept of in-memory patches
Analyzing CVE-2014-6253 Fix-It

- October 21\textsuperscript{st}, 1 week after CVE-2014-4114 was patched.


\begin{verbatim}
%windir\%System32\packager.dll (6.0.6002.19192) Checksum = (0x1708a)
%windir\%System32\packager.dll (6.0.6002.23496) Checksum = (0x1a612)
%windir\%System32\packager.dll (6.1.7601.18601) Checksum = (0x13541)
%windir\%System32\packager.dll (6.1.7601.22809) Checksum = (0x171ab)
%windir\%System32\packager.dll (6.2.9200.17121) Checksum = (0x14f94)
%windir\%System32\packager.dll (6.2.9200.21237) Checksum = (0x17675)
%windir\%System32\packager.dll (6.3.9600.17341) Checksum = (0x173b6)
%windir\%SysWOW64\packager.dll (6.0.6002.19192) Checksum = (0x1708a)
%windir\%SysWOW64\packager.dll (6.0.6002.23496) Checksum = (0x1a612)
%windir\%SysWOW64\packager.dll (6.1.7601.18601) Checksum = (0x13541)
%windir\%SysWOW64\packager.dll (6.1.7601.22809) Checksum = (0x171ab)
%windir\%SysWOW64\packager.dll (6.2.9200.17121) Checksum = (0x14f94)
%windir\%SysWOW64\packager.dll (6.2.9200.21237) Checksum = (0x17675)
%windir\%SysWOW64\packager.dll (6.3.9600.17341) Checksum = (0x173b6)
\end{verbatim}
Analyzing CVE-2014-6253 (sdb-explorer)

- `sdb-explorer.exe -t my.sdb`
- Prints Tree View, similar to sdb2xml
Analyzing CVE-2014-6253 (sdb2xml)

- sdb2xml my.sdb

```xml
<EXE>
  <NAME type="xs:string">POWERPNT.exe</NAME>
  <APP_NAME type="xs:string">POWERPNT.exe</APP_NAME>
  <VENDOR type="xs:string">Microsoft</VENDOR>
  <EXE_ID type="xs:string" baseType="xs:base64Binary">{e3d43c8e-908f-42fa-b4cd-d16aa46ef2ed}</EXE_ID>
  <APP_ID type="xs:base64Binary"/>
  <MATCHING_FILE>
    <NAME type="xs:string">*</NAME>
    <COMPANY_NAME type="xs:string">Microsoft Corporation</COMPANY_NAME>
  </MATCHING_FILE>
  <MATCHING_FILE>
    <NAME type="xs:string">%windir%\System32\packager.dll</NAME>
    <BIN_FILE_VERSION type="xs:long">1688854653393177</BIN_FILE_VERSION>
    <PE_CHECKSUM type="xs:int">94635</PE_CHECKSUM>
  </MATCHING_FILE>
  <PATCH_REF>
    <NAME type="xs:string">556af74c-d4d4-4191-8850-72d0247e3368</NAME>
    <PATCH_TAGID type="xs:int">746</PATCH_TAGID>
  </PATCH_REF>
</EXE>
```
- patch, patchbits, patchref, patch_tag_id, checksum
sdb-explorer.exe -p CVE-2014-6352-32bit-Shim.sdb 0x2ea
sdb-explorer.exe -i -p CVE-2014-6352-32bit-Shim.sdb 0x2ea

```python
from idaapi import *

base = idaapi.get_imagebase();
addr = 0;

addr = base + 0x58d4;
print "Patching: 0x%x 5 bytes" % (addr)
idaapi.patch_many_bytes(addr, "\xcc\xcc\xcc\xcc\xcc\xcc");
```
- sdb-explorer
- Explanation of in-memory patch file format.
  - Lots of details
- Shows how to analyze fix-it patches
- Showed how to create your own in-memory patches.
Prior Work

- Alex Ionesceu
  - Secrets of the Application Compatibility Database (SDB)
- Mark Baggett
  - Windows – Own3d by Default
- Jon Erickson
  - Persist-It – Using and Abusing Microsoft Fix It Patches
- Chris Graham
  - Shimming Your Way Past UAC
Shimming Your Way Past UAC

- Windows signed files with “AutoElevate” permission.
  - Example: SndVol.exe

- Uses RedirectEXE shim type.

- Steps:
  - Create Shim for SndVol.exe that does RedirectEXE to evil.exe
  - Register Shim
  - Start SndVol.exe
Agenda

- What is Application Compatibility
- Tools
- Prior Work
- EMET
- Real World Example
- The Enhanced Mitigation Experience Toolkit
- DEP
- SEHOP
- Null Page
- Heap Spray Protection
- EAF
- Mandatory ASLR
- ROP Detection
- Attack Surface Reduction
- %WINDIR%\AppPatch\EMET.dll
- %WINDIR%\AppPatch\AppPatch64\EMET.dll
EMET Shim
Agenda

- What is Application Compatibility
- Tools
- Prior Work
- EMET
- Real World Example
Real World Examples

- Search Protect
- BlackEnergy
- Win32/Farfli.BEK
- Potentially unwanted program (PUP)
  - Adware

- Now using Application Compatibility to persist
  - Uses InjectDll Shim
  - Loads Search Protect library into browsers
Recently targeting Ukrainian Government.

Uses UAC Bypass.
- Same technique as Chris Graham

Driver Signing Bypass
- Shim?
- To patch user32.dll-mui

Win32/Farfli.BEK Persistence

- Anton Cherepanov – ESET Reported at ZeroNights 2014

Win32/Farfli.BEK drops following files:
%WINDIR%\AppPatch\msimain.mui – raw code
%WINDIR%\AppPatch\AcProtect.dll

Drops Shim DataBase & registers it:
%WINDIR%\AppPatch\Custom\%GUID%.sdb
Win32/Farfli.BEK Persistence

- EMET-Style sdb
What Can You Do?

- Disable the Shim Engine
  - I do NOT recommend this
  - Breaks EMET
  - Disables 0day Fix-Its

- GPEdit.msc
  - Administrative Templates \ Windows Components \ Application Compatibility \ Turn off Application Compatibility Engine
Recommendations

- Search your registry and File System
  - Your system will have SDB Files, there are defaults
  - Use the knowledge you gained

- AutoRuns (SysInternals) does not consider Application Compatibility Fixes
  - They are aware and are working on it 😊

- Add signatures to SDB files (Microsoft)

- Notification of non-signed SDB files running, or about to run (Microsoft)
Application Compatibility Toolkit is a new method attackers are using today.

This is not a vulnerability

This is a feature that attackers are abusing

Defenders should start looking for this on the machines in their networks.

SDB File requires Administrator privilege to install
References

- ddilabs.blogspot.com/2014/05/shimming-your-way-past-uac.html
Thanks

- Kat, Josh, Sam, zen, Mac, Mike, Dave, Sean, Darel, Brad A., Matt G., Mark B., Chris G, Mark R., Microsoft, iSIGHT Partners, and all others who will remain nameless.
Questions

- jerickson <@> isightpartners.com

- Source Code:
  - https://github.com/evil-e/sdb-explorer