Johnny AppCompatCache

The Ring of Malware

PRESENTED BY: Mary Singh & Brice Daniels, Senior Consultants

JULY 9, 2013

Introductions

- Mary is an Incident Responder / Forensic Analyst
- Brice is an Incident Responder / Proactive Assessor
- APT and Financial Cases

@marycheese

@theonehiding

Mandiant: Experts in Advanced Targeted Threats

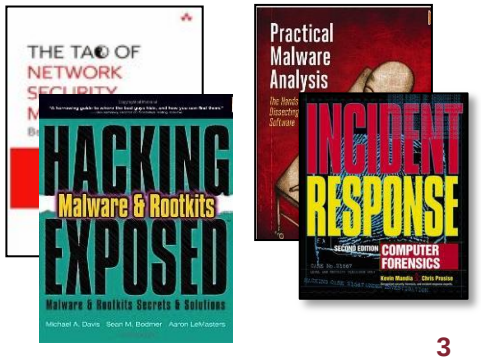
- **Expert Responders for Critical Security Incidents**
  - Incident responders to the biggest breaches
  - We train the FBI & Secret Service
  - Our consultants wrote the book (literally) on incident response
  - Clients include more than 33% of Fortune 500

Gartner 2013 CoolVendor

- **Our Products Are Based on Our Experience**
  - Built to find and stop advanced attackers
  - We use our own products in our investigations
  - SC Magazine 2012 & 2013 **"Best Security Company"**

SC MAGAZINE AWARDS 2013 WINNER Honored in the U.S. Best Security Company

- **Nationwide Presence**
  - 350+ employees
  - Offices in DC, New York, LA, San Francisco & Dublin



Mandiant's Unique Approach

Security incident response management platform

High-end, white glove incident response & security consulting services

Unmatched intelligence about tools & tactics of advanced attack groups directly from the front lines

Agenda

- Types of Attackers
- Application Compatibility Cache
  - Overview
  - Registry Key
  - Structure
- ShimCacheParser
- Case Study #1 – Stacking, a system in time saves 9
  Case Study #2 – What is seen, cannot be unseen
- Cash out
- Q&A

Types of Attackers

Application Compatibility Cache
ShimCacheParser
Case Studies
Cash out

All Threat Actors Are Not Equal

	Nuisance Threats	Economic Espionage	Organized Crime	Hacktivists
Objective	Launch Points & Nuisance	Economic Advantage	Financial Gain	Defamation, Press & Policy
Example	Botnets & Spam	Advanced Persistent Threat	Credit Card Theft	Anonymous & Lulzsec
Targeted	✗	✓	✓	✓
Persistent	✗	✓	✓	✗

*Attacks which are targeted **and** persistent pose the greatest challenge and the greatest risk.*

~~*Types of Attackers*~~

Application Compatibility Cache

ShimCacheParser
Case Studies
Cash out

Application Compatibility Cache

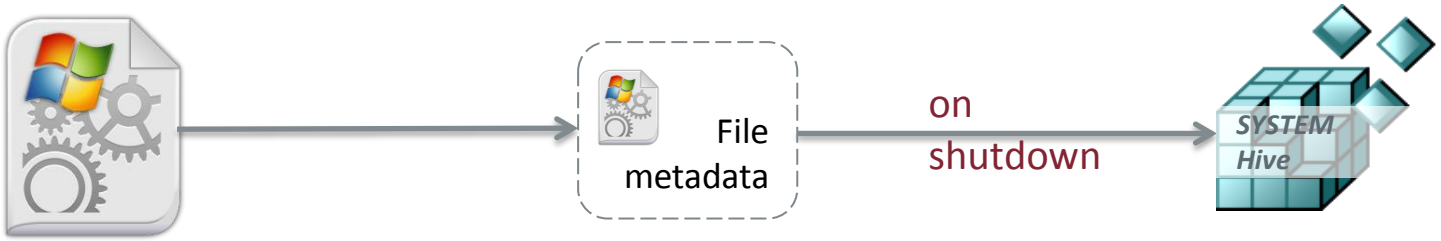
- Overview
  - Created by Microsoft to identify application compatibility issues, helps developers troubleshoot legacy functions
    - Windows looks at AppCompatCache to determine if modules require shimming for compatibility
  - The Cache data tracks file path, size, last modified time, and last execution time (depending on OS)
  - Most recent on top, written on shutdown
- Registry key

XP	`HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatibility\AppCompatCache`
Non-XP	`HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache\AppCompatCache`

That Shim is so Cache

- AppCompatCache is the key, but it's a.k.a. "ShimCache"
- What is a "shim" anyway?
  - Small library that intercepts an API and fixes compatibility
  - Helps legacy apps that rely on incorrect / old functionality
  - For Linux types… "Wine" is an example of a shim which enables Windows apps to run on Unix-y OSes
- Caveats…
  - Unavailable on Windows 2000 or older
  - Only files with specific extensions are logged (e.g. ".exe",".bat",".dll")
  - Registry updated only on system shutdown
  - Presence in Vista+ doesn't prove execution (more later)
- Handy protip: wood shims fix a non-level cabinet or align a door

Writing and Reading the Cache



1. File executed (file created also tracked in Vista+)

2. File metadata saved in a data structure

3. Data structure added to registry on shutdown

- Note: some forensic tools do not read AppCompatCache
  - Why? The information is stored as “REG_BINARY” data
  - Many of these tools don’t parse “big data” values
  - e.g. WRR, WRA, EnCase, Registry Viewer will not display the data in AppCompatCache correctly
- Some registry tools can read this data:
  - Regripper with updated plugin
  - Lock and Code Registry Browser
  - TZWorks Yaru, more…

AppCompat Structure

- AppCompat Structure (3 formats)
  - There are differences between Windows OS versions
  - Different metadata stored
- Three unique binary data structure formats each with 32/64bit versions stored in a series of records

- <u>Windows XP</u>
  - Full path of file
  - Last Modified Date
  - File size (when executed)
  - Last time the file was executed
  - 96 entries
  - Header: “0xDEADBEEF”

```
typedef struct AppCompatibilityEntry{
    WCHAR Path[MAX_PATH+4];
    FILETIME LastModTime;
    DWORD  dwFileSize;
    DWORD  dwFileSizeHigh;
    FILETIME LastExecTime;
};
```

AppCompat Structure

- Windows Server 2003
  - Last Modified Date
  - Full file path
  - File size (when executed)
  - 512 entries
  - Header: “0xBADC0FFE”

```
typedef struct AppCompat_Entry32_Type1 {
    USHORT wLength;
    USHORT wMaximumLength;
    DWORD  dwPathOffset;
    FILETIME qwFileTime;
    DWORD dwFileSize;
    DWORD dwFileSizeHigh;
};
```

- Windows Vista+
  - Last Modified Date
  - Full file path
  - File size
  - Binary “execution” flag
  - Logs files executed and/or created
  - 1024 entries
  - Header: “0xBADC0FEE”

```
typedef struct AppCompat_Entry32_Type2 {
    USHORT wLength;
    USHORT wMaximumLength;
    DWORD  dwPathOffset;
    FILETIME qwFileTime;
    DWORD dwFileFlags;
    DWORD dwFlags;
    DWORD dwBlobSize;
    DWORD dwBlobOffset;
};
```

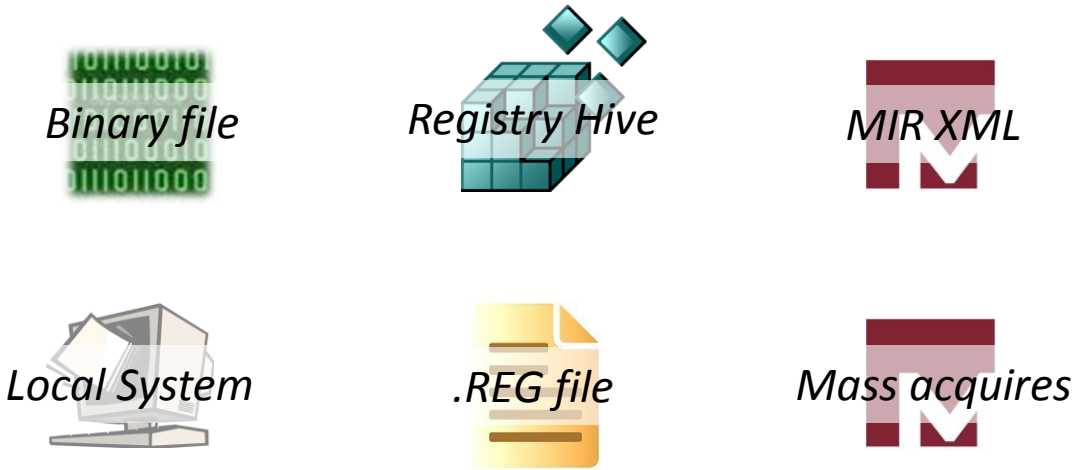
~~*Types of Attackers*~~
~~*Application Compatibility Cache*~~

ShimCacheParser

Case Studies
Cash out

ShimCacheParser

- ShimCacheParser.py
  - Automatically locates AppCompatCache related keys, determines their structure type and exports the data
  - 6 types of input:



- Download at https://github.com/mandiant/ShimCacheParser

ShimCacheParser

- Output in CSV format

```
> ShimCacheParser.py -i D:\case\SYSTEM -o D:\case\output.txt
```

Last Modified	Last Update	Path	File Size	Process Exec Flag
08/27/12 19:53:26	N/A	C:\Windows\system32\sql.exe	N/A	No
08/27/12 19:52:34	N/A	C:\Users\joeuser\AppData\Local\Temp\tmp83e46c15\12345.exe	N/A	Yes
07/14/09 01:14:41	N/A	C:\Windows\system32\svchost.exe	N/A	No
08/24/12 19:19:59	N/A	C:\Windows\system32\b.exe	N/A	No
07/14/09 01:14:12	N/A	C:\Windows\system32\at.exe	N/A	No
08/24/12 19:37:47	N/A	C:\Windows\system32\msabc.exe	N/A	No
07/14/09 01:14:27	N/A	C:\Windows\system32\net1.exe	N/A	No
07/14/09 01:14:45	N/A	C:\Windows\system32\whoami.exe	N/A	No
07/14/09 01:14:27	N/A	C:\Windows\system32\NETSTAT.EXE	N/A	No
08/24/12 19:16:36	N/A	C:\Users\joeuser\AppData\Local\Temp\tmp591d39cc\12345.exe	N/A	Yes

~~*Types of Attackers*~~
~~*Application Compatibility Cache*~~
~~*ShimCacheParser*~~

Case Studies

Cash Out

Case Study #1 – What is EVERYONE executing?!

- Use the AppCompatCache to find evil everywhere!
- Situation
  - 30,000 node Windows AD domain
  - Active attacker
- Benefits
  - Fast results
  - Develop investigative leads

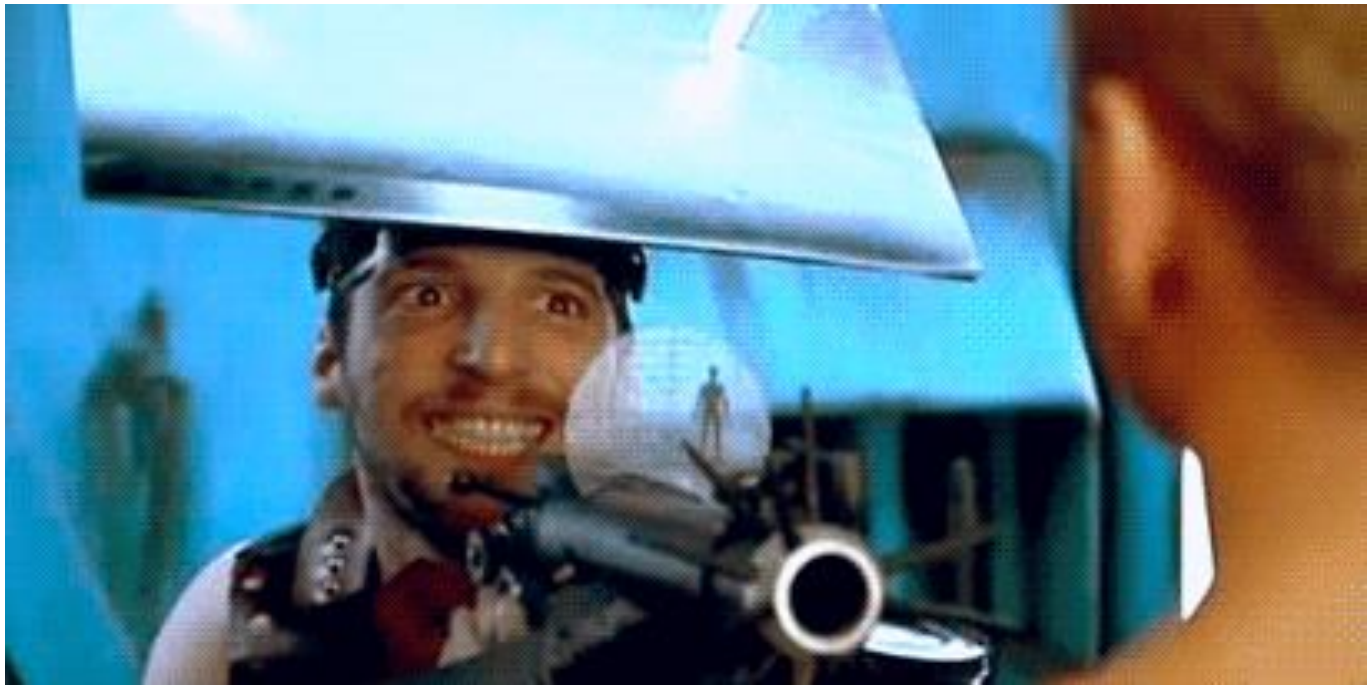
Steps

1. Collect registry keys from your network
   1. Use IOC Finder to collect AppCompatCache keys as MIR XML – except IOC Finder is a ~14MB executable
   2. Use <2KB batch script to export keys to a .reg file
      - https://github.com/theonehiding/ShimCacheCollector
2. Run ShimCacheParser.py across the set
3. Analyze

4. … Profit!

Gimme Da **CACHE**!

Gimme Da Cache!



Exporting AppCompat Keys

- Two commands
  - reg export [key] [file]
  - regedit /e [file] [key]

```
rem For Windows 7

reg export "HKLM\SYSTEM\CurrentControlSet\Control\Session
  Manager\AppCompatCache" %TEMP%\reg_0.reg /y >nul 2>1

rem For Windows XP

regedit /e %TEMP%\reg_0.reg
  "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session
  Manager\AppCompatibility" >nul 2>1
```

Examinations at Scale

- Suspicious filenames
  - Pwdump.exe
- Utilities possibly used by the attacker
  - at.exe
  - psexec.exe / psexesvc.exe
- Suspicious paths
  - C:\Program Files\
  - C:\Recycler\
  - C:\Windows\addins\
  - C:\ProgramData\
- File sizes of known malware

Time Lining

- AppCompatCache only provides file modified times
  - Except for Windows XP
  - Could be modified by the attacker
- Could correspond to the creation time
  - Tools like psexesvc.exe

Hunting an older attacker

- Data in Windows 7, Windows 2008 remains longer
- Servers may contain older records
  - AppCompat data serialized only on shutdown

Stacking Execution History

- Stacking helps find needles in haystacks
- Use to help find files masquerading as legitimate

Count	File Path	File Size	Executed
2	c:\Windows\System32\scvhost.exe	N/A	Yes
763	c:\Windows\System32\svchost.exe	N/A	Yes

- Complicated by temporary files
  - Hotfix installers with Purposely unique directories

System	Date Modified	Date Executed	File Path	File Size	Executed
{Win-7}	2011-07-09 13:23:02	N/A	c:\01882cb67ce37b6f7137\Setup.exe	N/A	Yes
{Win-7}	2011-01-11 12:41:10	N/A	c:\01e45c03736f85750ed2\install.exe	N/A	Yes
{Win-XP}	2010-07-05 13:15:53	2013-06-10 15:23:39	c:\01f1236656cecae4125c\update\update.exe	755,576	N/A

Case Study #2 – I see what you did there

- Attacker Type: Organized Crime
- Target: Corporation
  - *Filenames changed to protect the innocent (and our NDA)

- Discovered bad file “cdel.exe” (variant of Citadel)
  - C:\Users\mary\AppData\Roaming\Gappy\cdel.exe
  - The file was timestomped
  - Extracted filename creation date from $MFT
    - August 20, 2012 18:06:49

Standard Information Attribute vs Filename Information

	Std Info	SIA	SIA	SIA	Filename	Filename	Filename	Filename
Name	Created	Modified	Accessed	Entry Md	Created	Modified	Accessed	Entry Md
cdel.exe	06/12/12 13:10:41	06/12/12 13:10:41	06/12/12 13:10:41	10/03/12 09:12:23	08/20/12 18:06:49	08/20/12 18:06:49	08/20/12 18:06:49	08/20/12 18:06:49

Output from ShimCacheParser

- Imported SYSTEM hive, exported 1,022 rows

Last Modified	Last Update	Path	File Size	Process Exec Flag
04/04/05 23:58:34	N/A	C:\Progra~2\Adobe\Adobe Version Cue CS2\data\database\bin\mysqladmin.exe	N/A	Yes
12/07/09 06:19:00	N/A	C:\Program Files (x86)\UltraVNC\WinVNC.exe	N/A	Yes
07/14/09 01:39:15	N/A	C:\Windows\system32\LogonUI.exe	N/A	Yes
07/14/09 01:39:37	N/A	C:\Windows\system32\SearchFilterHost.exe	N/A	Yes
07/14/09 01:39:37	N/A	C:\Windows\system32\SearchProtocolHost.exe	N/A	Yes
02/07/12 03:38:24	N/A	C:\Program Files\Common Files\Autodesk Shared\AcHelp2.exe	N/A	Yes
07/14/09 01:14:35	N/A	C:\Windows\sysWow64\SearchProtocolHost.exe	N/A	Yes

- Since we know “cdel.exe” is bad, search for that…
  - 3 entries for cdel.exe, next to 2 entries for “shoe1.exe”
  - Note: the Last Modified date matches the Standard Info Attribute

Last Modified	Last Update	Path	Size	Exec Flag
06/12/12 17:10:41	N/A	C:\Users\mary\AppData\Roaming\Gappy\cdel.exe	N/A	Yes
10/03/12 13:12:21	N/A	C:\Users\mary\AppData\Local\Temp\tmpebc090bd\shoe1.exe	N/A	Yes
02/15/12 08:12:04	N/A	C:\Users\mary\AppData\Roaming\Gappy\cdel.exe	N/A	Yes
09/24/12 13:09:49	N/A	C:\Users\mary\AppData\Local\Temp\tmp6e3a4f14\shoe1.exe	N/A	Yes
08/23/10 18:01:54	N/A	C:\Users\mary\AppData\Roaming\Gappy\cdel.exe	N/A	Yes

Analyzing ShimCacheParser output

- Look around malicious filenames, lines before and after
  - Remember: the most recent entries are on top
- Good ol' Timeline analysis
  - Check 8/20/12 "cdel.exe" creation date, 8/21/12 was earliest :-(
  - Extracted & parsed the backup copy of the SYSTEM registry hive :-)
- 3 entries for "c123.exe", 2 more entries for "cdel.exe", "shoe1.exe"
  - Note: no other evidence of "shoe1.exe" or "c123.exe"!

Last Modified	Last Update	Path	Size	Exec Flag
08/27/12 19:52:34	N/A	C:\Users\mary\AppData\Local\Temp\tmp83e46c15\c123.exe	N/A	Yes
08/24/12 19:16:36	N/A	C:\Users\mary\AppData\Local\Temp\tmp591d39cc\c123.exe	N/A	Yes
08/24/12 13:07:33	N/A	C:\Users\mary\AppData\Local\Temp\tmpc0803709\c123.exe	N/A	Yes
08/21/12 13:14:21	N/A	C:\Users\mary\AppData\Local\Temp\tmp4313f0ee\shoe1.exe	N/A	Yes
02/25/11 18:28:08	N/A	C:\Users\mary\AppData\Roaming\Gapiy\cdel.exe	N/A	Yes
08/20/12 18:06:49	N/A	C:\Users\mary\AppData\Local\Temp\1jfmlsif.exe	N/A	Yes

Break the case open!

Last Modified	Last Update	Path	Size	Exec Flag
08/24/12 19:16:36	N/A	C:\Users\mary\AppData\Local\Temp\tmp591d39cc\c123.exe	N/A	Yes
07/14/09 01:14:27	N/A	C:\Windows\SysWOW64\NETSTAT.EXE	N/A	Yes
07/14/09 01:14:45	N/A	C:\Windows\SysWOW64\whoami.exe	N/A	Yes
07/14/09 01:14:27	N/A	C:\Windows\SysWOW64\net1.exe	N/A	Yes
08/24/12 19:37:47	N/A	C:\Windows\SysWOW64\msbad.exe	N/A	Yes
07/14/09 01:14:42	N/A	C:\Windows\SysWOW64\taskkill.exe	N/A	Yes
08/24/12 20:49:00	N/A	C:\Windows\SysWOW64\msevil.exe	N/A	Yes
07/14/09 01:14:20	N/A	C:\Windows\SysWOW64\find.exe	N/A	Yes
12/27/10 15:01:12	N/A	C:\Windows\SysWOW64\schtasks.exe	N/A	Yes
07/14/09 01:14:12	N/A	C:\Windows\SysWOW64\at.exe	N/A	Yes
07/14/09 01:14:27	N/A	C:\Windows\SysWOW64\net.exe	N/A	Yes
07/14/09 01:14:21	N/A	C:\Windows\SysWOW64\HOSTNAME.EXE	N/A	Yes
07/14/09 01:14:21	N/A	C:\Windows\SysWOW64\ipconfig.exe	N/A	Yes

What now?

- Knew about “cdel.exe”
- Discovered new things:
  - 3 new filenames: “1jfmlsif.exe”, “shoe1.exe”, and “c123.exe”
  - 2 new backdoors: “msbad.exe”, “msevil.exe”
    - Still on the system: hashes, file size, file ownership
- Potentially more…
  - Compromised account “mary” (3 files saved in this profile)
  - Network information from the backdoors
  - Scheduled Task log files
  - Timeframe of compromise
- Search logs for activity
  - DNS logs, EVT log process monitoring, lateral movement, etc.
- Use the Indicators of Compromise (IOCs) for other systems

~~*Types of Attackers*~~
~~*Application Compatibility Cache*~~
~~*ShimCacheParser*~~
~~*Case Studies*~~

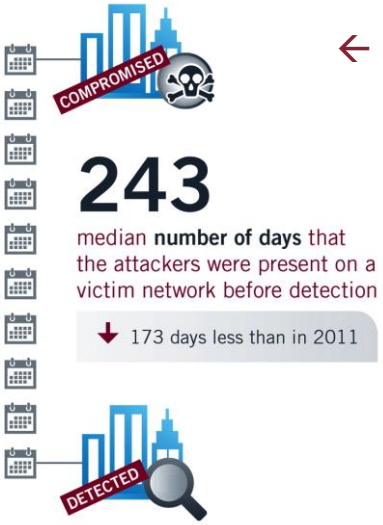
Cash Out

Cash Out

- What have we learned?
  - The AppCompatCache tracks file metadata for investigators like Last Modified date, full path, and file size
  - Most recent events are on top
  - New entries are written on shutdown
- Takeaways:
  - Source of evidence for deleted files
  - Use AppCompatCache along with your timelines to reconstruct and determine attacker activity
  - Plug IOCs back into an investigation to find more

Q&A

- **Email**
  - **mary.singh@mandiant.com | brice.daniels@mandiant.com**
- **ShimCacheParser Whitepaper**
  - www.mandiant.com/library/Whitepaper_ShimCacheParser.pdf
- **Additional Resources**
  - Mandiant Blog: blog.mandiant.com
  - Mandiant Reports:
    - M-Trends www.mandiant.com/m-trends
    - APT1 Report: www.mandiant.com/apt1



← (M-Trends snippet)