

# C#'ening Your Forensic Tools

Eric R. Zimmerman

Senior director, Kroll Cyber Security

[eric.zimmerman@kroll.com](mailto:eric.zimmerman@kroll.com)

501-313-3778

@EricRZimmerman

<https://binaryforay.blogspot.com/>



# What tools are there?

- 100% managed C# code for:

- Lnk files
- Registry hives
- Jump lists
- Prefetch
- ExtensionBlocks (shell items)
- OleCF
- Amcache
- Appcompatcache (shimcache)
- WMI/CIM\*
- MFT\*

\* Coming soon! 10

# Why use these tools?

- Extract mass quantities of forensic data
- Often first to support new formats and/or features
- Simple, easy to understand code
- Fast, open source, and free
- Seamless integration with PowerShell
- When issues arise, fail noisily!



# Why use these tools?

- Support for key artifacts showing a range of activity
  - Evidence of execution
  - Directory traversal
  - File access
  - Persistence
- Find things other tools miss
- Export to any format you like
  - Json, csv, xml, html, etc.





# How can they be used?

- All parsers are standalone projects
- This allows for anyone to integrate core parsers into a larger tool chain vs only consuming output from existing tools
- Multiple front ends: Secondary projects use the parsers and wrap them
  - Command line tools
  - Awesome GUIs

JLECmd version 0.9.6.1

Author: Eric Zimmerman (saericzimmerman@gmail.com)

<https://github.com/EricZimmerman/JLECmd>

d	Directory to recursively process. Either this or -f is required
f	File to process. Either this or -d is required
q	Only show the filename being processed vs all output.
all	Process all files in directory vs. only files matching
csv	Directory to save CSV (tab separated) formatted results
html	Directory to save xhtml formatted results to. Be sure to use --pretty
json	Directory to save json representation to. Use --pretty
pretty	When exporting to json, use a more human readable layout
ld	Include more information about lnk files
fd	Include full information about lnk files (Alternatively use --ld)
dumpTo	Directory to save exported lnk files
dt	The custom date/time format to use when displaying time stamps
mp	Display higher precision for time stamps. Default is false
withDir	When true, show contents of Directory not accounted for in debug mode

Examples: JLECmd.exe -f "C:\Temp\f01b4d95cf55d32a.customDestinations-ms" --mp  
JLECmd.exe -f "C:\Temp\f01b4d95cf55d32a.automaticDestinations-ms" --mp

Modified on: 2016-10-25 12:54:19 +00:00

Last accessed on: 2016-10-10 00:12:58 +00:00

Executable name: 7ZFM.EXE

Hash: 3129C294

File size (bytes): 56,188

Version: Windows 10

Run count: 6

Last run: 2016-10-25 12:54:17 +00:00

Other run times: 2016-10-15 14:49:06 +00:00, 2016-10-13 17:16:05 +00:00, 2016-10-13 00:02:20 +00:00, 2016-10-10 00:22:11 +00:00, 2016-10-10 00:12:52 +00:00

Volume information:

#0: Name: \VOLUME{01d203b1729eca21-6c72c785} Serial: 6C72C785 Created: 2016-08-31 17:59:41 +00:00 Directories: 27 File references: 97

Directories referenced: 27

00: \VOLUME{01d203b1729eca21-6c72c785}\PROGRAM FILES  
01: \VOLUME{01d203b1729eca21-6c72c785}\PROGRAM FILES (X86)  
02: \VOLUME{01d203b1729eca21-6c72c785}\PROGRAM FILES (X86)\STARDOCK  
03: \VOLUME{01d203b1729eca21-6c72c785}\PROGRAM FILES (X86)\STARDOCK\START10  
04: \VOLUME{01d203b1729eca21-6c72c785}\PROGRAM FILES\7-ZIP  
05: \VOLUME{01d203b1729eca21-6c72c785}\PROGRAM FILES\GPSOFTWARE

AppCompatCache Parser version 0.9.2.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)

<https://github.com/EricZimmerman/AppCompatCacheParser>

c	The ControlSet to parse. Default is to detect the current control set.
d	Debug mode
h	Full path to SYSTEM hive file to process. If this option is not specified, the live Registry will be used
s	(REQUIRED) Directory to save results
t	Sorts timestamps in descending order
dt	The custom date/time format to use when displaying time stamps. Default is: yyyy-MM-dd HH:mm:ss K

Example: AppCompatCacheParser.exe -s c:\temp -t -c 2

Registry Explorer v0.8.1.0

File Tools Options Bookmarks (26/0) View Help

Registry hives (3) Available bookmarks (63/1)

Key name	# val...	Last write timesta...
D:\Dropbox\RegistryHives\UsrClass.dat		2015-08-03 02:1...
S-1-5-21-3967952009-1209047751-3562736479-1000_CI...	0	2015-07-25 16:2...
D:\Dropbox\RegistryHives\SYSTEM		2013-08-22 14:5...
CsTool-CreateHive-{00000000-0000-0000-0000-00000000...}	0	2015-02-24 03:2...
ControlSet001	0	2013-08-22 15:3...
DriverDatabase	3	2015-01-23 16:3...
HardwareConfig	2	2015-02-24 03:2...
MountedDevices	65	2015-02-23 17:3...
RNG	2	2015-02-24 03:2...
Select	4	2013-08-22 13:2...
Setup	12	2015-02-24 03:2...
WPA	0	2015-02-24 03:2...
Unassociated deleted records	0	
D:\Dropbox\RegistryHives\NTUSER.DAT		2013-08-22 13:2...
CsTool-CreateHive-{00000000-0000-0000-0000-00000000...}	0	2014-11-28 16:5...
AppEvents	0	2014-05-20 14:1...
AppLifeUpdatesShortcuts	1	2014-05-20 18:4...
Console	39	2014-10-15 20:2...
Control Panel	0	2014-11-06 16:2...
Environment	3	2014-06-27 19:1...
EUDC	0	2014-05-20 14:1...
Identities	0	2014-05-20 18:4...
Keyboard Layout	0	2014-05-20 14:1...
Network	0	2014-10-24 15:1...
Printers	0	2014-10-10 20:4...
Software	0	2014-12-08 13:5...
System	0	2014-05-20 14:1...
Associated deleted records	0	
Unassociated deleted records	0	

Values

Drag a column header here to group by that column

Value Name	Value Type	Data	Value Slack
\DosDevices\C:	RegBinary	BD-C3-FF-A5-00-00-...	
\\Volume{10feca75-e030-11e3-8250-806e6f6e963}	RegBinary	BD-C3-FF-A5-00-00-...	
\\Volume{10feca76-e030-11e3-8250-806e6f6e963}	RegBinary	BD-C3-FF-A5-00-00-...	
\\Volume{10feca77-e030-11e3-8250-806e6f6e963}	RegBinary	E1-1F-F1-07-00-00-1...	
\\Volume{10feca7d-e030-11e3-8250-806e6f6e963}	RegBinary	5C-00-3F-00-3F-00-5...	
\\Volume{10feca7e-e030-11e3-8250-806e6f6e963}	RegBinary	5C-00-3F-00-3F-00-5...	
\DosDevices\D:	RegBinary	44-4D-49-4F-3A-49-4...	18-52-73-00
\DosDevices\E:	RegBinary	E1-1F-F1-07-00-00-1...	
\DosDevices\F:	RegBinary	44-4D-49-4F-3A-49-4...	D8-3F-0C-00
\DosDevices\G:	RegBinary	5C-00-3F-00-3F-00-5...	
\DosDevices\H:	RegBinary	5C-00-3F-00-3F-00-5...	
\\Volume{14fc45bc-e0e5-11e3-8258-ac220b2a5a56}	RegBinary	C8-E7-21-B6-00-04-0...	
\DosDevices\I:	RegBinary	44-4D-49-4F-3A-49-4...	44-00-69-00-73-00-68-00-26-00-5...
\\Volume{14fc504e-e0e5-11e3-8258-ac220b2a5a56}	RegBinary	5F-00-3F-00-3F-00-5...	00-00
\\Volume{14fc5203-e0e5-11e3-8258-ac220b2a5a56}	RegBinary	5F-00-3F-00-3F-00-5...	00-00
\\Volume{14fc5203-e0e5-11e3-8258-ac220b2a5a56}	RegBinary	5F-00-3F-00-3F-00-5...	00-00

Type viewer

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11
00000000	5C	00	3F	00	3F	00	5C	00	53	00	43	00	53	00	49	00	23	00
00000012	43	00	64	00	52	00	6F	00	6D	00	26	00	56	00	65	00	6E	00
00000024	5F	00	41	00	53	00	55	00	53	00	26	00	50	00	72	00	6F	00
00000036	64	00	5F	00	44	00	52	00	57	00	2D	00	32	00	34	00	42	00
00000048	31	00	53	00	54	00	5F	00	5F	00	63	00	63	00	23	00	35	00
0000005A	26	00	32	00	66	00	66	00	34	00	33	00	33	00	33	00	34	00
0000006C	26	00	30	00	26	00	30	00	30	00	30	00	30	00	30	00	30	00
0000007E	23	00	78	00	35	00	33	00	66	00	35	00	36	00	33	00	30	00
00000090	64	00	2D	00	62	00	36	00	62	00	66	00	2D	00	31	00	31	00

Current offset: 0 (0x0) Bytes selected: 0 (0x0) Value: \\Volume{10feca7d-e030-11e3-8250-806e6f6e963} Collapse all hives

Key: MountedDevices

Last write: 2015-02-23 17:31:31 +00:00 | 65 of 65 values shown (100.00 %) | Load complete

Hidden keys: 0 | 216

ShellBags Explorer v0.7.0.0 - UsrClass MTP.dat

File Tools Help

Desktop

- Téléphone
- Photos - Carte mémoire
- Khazam - Fichiers
- Documents ordinateur bas - A TRIER
- Cours
- My Computer
- Downloads
- C:
- F:
- Desktop
- Videos
- Pictures
- A1-830
- Mémoire de stockage interne
- Video
- Documents
- H:
- I:
- Music
- G:
- D:
- C:
- C:
- C:
- C:
- DLNA Media Servers Data Source
- Thunder\_Q45
- Apple iPod
- H:\
- Wave 3
- Recycle bin
- Control Panel
- My Games
- Nouveau dossier
- Search Folder
- Search Folder
- Photos
- Portable
- Musique
- User Libraries
- OneDrive

Details Hex view

Value: A1-830

Shell Type: Root folder

Bag Path: BagMRU\

Absolute Path: Desk

# Child Bags: 2

Last Write Time: 8/2

Hex Value: 5C-01-2F-5F-00-30-00-35-00-61-00-2D-00-34-00-26-43-E6-26-46-9E-8C-00-00-74-1A-59-

'UsrClass MTP.dat' registry hive data loaded in 1.8953 seconds! 2 shell bags loaded

JumpList Explorer v0.3.0.0

File Help

Drag a column header here to group by that column

Source File Name	Jump List Type	App ID	App ID Description	Lnk File Count	File Size
D:\Dropbox\Jump lists and Inks\5f7b5f1e01b83767	Automatic	5f7b5f1e01b83767	Unknown AppId		673
D:\Dropbox\Jump lists and Inks\9839aec31243a928(ExtraDirectories)	Automatic	9839aec31243a928(ExtraDirectories)	Unknown AppId		12
D:\Dropbox\Jump lists and Inks\ITA_Jum...	Automatic	f01b4d95cf55d32a	Windows Explorer Windows 8.1.		7
D:\Dropbox\Jump lists and Inks\469e4a7...	Automatic	469e4a7982cea4d4	Windows Wordpad		1
D:\Dropbox\Jump lists and Inks\ITA_Jum...	Automatic	fe725be78b8614c4	Unknown AppId		0

```
{
  __type: "Prefetch.Version30, Prefetch",
  RawBytes: HgAAAFNDQ0ERAAAFNsAADcAWgBGAE0ALgBFAFgARQAAAAAAAAAAAAA
  SourceFilename: "c:\Windows\Prefetch\7ZFM.EXE-3129C294.pf",
  SourceCreatedOn: 2016-10-10T00:12:58.2945850+00:00,
  SourceModifiedOn: 2016-10-25T12:54:19.7247787+00:00,
  SourceAccessedOn: 2016-10-10T00:12:58.2945850+00:00,
  Header:
  {
    Version: Win10,
    Signature: SCCA,
    FileSize: 56188,
    ExecutableFilename: 7ZFM.EXE,
    Hash: 3129C294
  },
  FileMetricsOffset: 304,
  FileMetricsCount: 81,
  TraceChainsOffset: 2896,
  TraceChainsCount: 4623,
  FilenameStringsOffset: 39880,
  FilenameStringsSize: 11906,
  VolumesInfoOffset: 51792,
  VolumeCount: 1,
  VolumesInfoSize: 4396,
  TotalDirectoryCount: 27,
  LastRunTimes:
  [
    2016-10-25T12: 54:17.7485709+00:00,
    2016-10-15T14: 49:06.8183459+00:00,
    2016-10-13T17: 16:05.1050550+00:00,
    2016-10-13T00: 02:20.7711422+00:00,
    2016-10-10T00: 22:11.4654395+00:00,
    2016-10-10T00: 12:52.5187725+00:00
  ],
  VolumeInformation:
  [
    {

```

```
__type: "ExtensionBlocks.Beef0004, ExtensionBlocks",
CreatedOnTime: 2012-12-03T22:05:30.0000000+00:00,
LastAccessTime: 2016-09-01T15:44:18.0000000+00:00,
Identifier: 42,
MFTInformation:
{
  MFTEntryNumber: 528,
  MFTSequenceNumber: 2,
  Note: NTFS
},
LongName: Documents,
LocalisedName: "@shell132.dll,-21770",
Message: "",
Size: 104,
Version: 8,
Signature: 3203334148,
VersionOffset: 24
}
```

# automaticDestinations-ms.json

```
...40.....50.....60.....70.....80.....90.....100.....110.....
},
{
  __type: "Lnk.ShellItems.ShellBag0X00, Lnk",
  PropertyStore:
  {
    Sheets:
    [
      {
        Size: 174,
        Version: 31-53-50-53,
        GUID: b725f130-47ef-101a-a5f1-02608c9eebac,
        PropertyNames:
        {
          10: Hardware.xlsx,
          15: "10/17/2014 10:25:06",
          12: 196357,
          13: 32,
          14: "12/03/2014 15:30:22",
          16: "12/03/2014 15:30:22"
        },
        PropertySheetType: Numeric
      },
      {
        Size: 49,
        Version: 31-53-50-53,
        GUID: 446d16b1-8dad-4870-a748-402ea43d788c,
        PropertyNames:
        {
          100: 3109811667573631259
        },
        PropertySheetType: Numeric
      },
      {
        Size: 652,
        Version: 31-53-50-53,
        GUID: 28636aa6-953d-11d2-b5d6-00c04fd918d0,
        PropertyNames:
        {

```



# Yea, but how easy is it?

- June 22, Andrew Case asks about Mark Woan's Jumplister tool and support for Windows 10 on Twitter
- Project started on June 23. Within a few hours, working prototype built using JumpList project

JumpList Explorer - vX

File Help

Drag a column header here to group by that column

Source File Name	Jump List Type	App ID	App ID Description	Dest List Entry Count	Pinned Entry Count	Last Entry Number
D:\Temp\1b4dd67f29cb1962.automatic...	Automatic	1b4dd67f29cb1962	Windows Explorer Pinned and Recent.	4	0	4
D:\Temp\1b4dd67f29cb1962.automatic...	Automatic	f01b4d95cf55d32a	Windows Explorer Windows 8.1.	9	0	9

Name

f01b4d95cf55d32a.automaticDestinations-ms

Entry #1: 7

Entry #1: 9

Entry #1: 8

Entry #1: 1

Entry #1: 2

Entry #1: 6

Entry properties

Entry #

Path

knownfolder:{374DE290-123F-4565-9164-39C4925E46...}

Hostname

desktop-annf9d

MAC address

00:15:5d:01:6d:02

Created

11/24/2015 6:30:28 PM +00:00

Last modified

11/24/2015 6:31:14 PM +00:00

Volume birth droid

4c01e77a-2661-40c9-b838-14ff153a26f5

Volume droid

4c01e77a-2661-40c9-b838-14ff153a26f5

File birth droid

6fcbcd8f-92d9-11e5-88e0-00155d016d02

File droid

6fcbcd8f-92d9-11e5-88e0-00155d016d02

Pinned

☒

Hostname

Volume Droid

Volume Birth ...

File Droid

File Birth Droid

Entry Number

Created On

Last Modified

Pinned

Path

Mac Address

Lnk

desktop-annf... 4c01e77a-26... 4c01e77a-26... a21b9d4a-92... a21b9d4a-92... 7 11/24/2015 6... 11/24/2015 8... ☐ C:\Temp 00:15:5d:01:6d:02 Source file: D:... desktop-annf... 4c01e77a-26... 4c01e77a-26... a21b9d50-92... a21b9d50-92... 9 11/24/2015 6... 11/24/2015 8... ☐ C:\Temp\1 00:15:5d:01:6d:02 Source file: D:... desktop-annf... 4c01e77a-26... 4c01e77a-26... a21b9d4b-92... a21b9d4b-92... 8 11/24/2015 6... 11/24/2015 8... ☐ C:\ 00:15:5d:01:6d:02 Source file: D:... desktop-annf... 4c01e77a-26... 4c01e77a-26... 6fcbcd8e-92... 6fcbcd8e-92... 1 11/24/2015 6... 11/24/2015 6... ☒ C:\Users\le\D... 00:15:5d:01:6d:02 Source file: D:... desktop-annf... 4c01e77a-26... 4c01e77a-26... 6fcbcd8f-92... 6fcbcd8f-92... 2 11/24/2015 6... 11/24/2015 6... ☒ knownfolders... 00:15:5d:01:6d:02 Source file: D:... desktop-annf... 4c01e77a-26... 4c01e77a-26... 6fcbcd8a-92... 6fcbcd8a-92... 6 11/24/2015 6... 11/24/2015 6... ☐ knownfolders... 00:15:5d:01:6d:02 Source file: D:... desktop-annf... 4c01e77a-26... 4c01e77a-26... 6fcbcd8d-92... 6fcbcd8d-92... 5 11/24/2015 6... 11/24/2015 6... ☐ knownfolders... 00:15:5d:01:6d:02 Source file: D:... desktop-annf... 4c01e77a-26... 4c01e77a-26... 6fcbcd8c-92... 6fcbcd8c-92... 4 11/24/2015 6... 11/24/2015 6... ☒ knownfolders... 00:15:5d:01:6d:02 Source file: D:... desktop-annf... 4c01e77a-26... 4c01e77a-26... 6fcbcd8b-92... 6fcbcd8b-92... 3 11/24/2015 6... 11/24/2015 6... ☒ knownfolders... 00:15:5d:01:6d:02 Source file: D:... Status info

Eric Zimmerman @EricRZimmerman · Jun 23

@505Forensics @lee\_whitfield @atrc things are progressing well

2

# Looking under the covers

```
foreach (var file in fb.FileNames)
{
    try
    {
        var b = File.OpenRead(file).ReadByte();

        var fi = new FileInfo(file);

        switch (b)
        {
            case 0xD0: //automatic
                var auto = JumpList.JumpList.LoadAutoJumpList(file);
                var autoj = new JumpListOverview(file, JumpListOverview.JumpListTypes.Automatic,
                    auto.AppId.AppId, auto.AppId.Description, fi.Length, auto.DestListEntries.Count);

                _autoJumpLists.Add(file, auto);
                _jumpListOverviews.Add(autoj);

                break;

            case 0x02: //custom
                var custom = JumpList.JumpList.LoadCustomJumpList(file);
```

Determine type

Parse file

- Many projects have constructors that take multiple input formats
  - File name
  - Byte array
- This allows for maximum flexibility depending on how data is made available

# Looking under the covers

```
rootNode.SetValue("Name", Patn.GetFilename(jla.SourceFile));
//
rootNode.ImageIndex = 0;
rootNode.SelectImageIndex = 0;

rootNode.Tag = jla;

foreach (var destListEntry in jla.DestListEntries)
{
    var childNode = treeJumpList.AppendNode(null, rootNode);

    var target = GetAbsolutePathFromTargetIDs(destListEntry.Lnk.TargetIDs);

    if (target.Length == 0)
    {
        target = $"{destListEntry.Lnk.NetworkShareInfo.NetworkShareName}\\\\{destListEntry.Lnk.CommonPath}";
    }

    childNode.SetValue("Name",
        $"Entry #: {destListEntry.EntryNumber.ToString().PadLeft(4, '0')} - {target}");
    childNode.Tag = destListEntry;
}
```

Iterate destlist

Easy access to  
Lnk details

- Provide strongly typed objects and lists
- Most also provide access to the raw bytes for verification, etc.

# Deeper still!

```
//export lnks if requested
if (_fluentCommandLineParser.Object.LnkDumpDirectory.Length > 0)
{
    _logger.Info("");
    _logger.Warn(
        $"Dumping lnk files to '{_fluentCommandLineParser.Object.LnkDumpDirectory}'");

    if (Directory.Exists(_fluentCommandLineParser.Object.LnkDumpDirectory) == false)
        Directory.CreateDirectory(_fluentCommandLineParser.Object.LnkDumpDirectory);

    foreach (var processedCustomFile in _processedCustomFiles)
        foreach (var entry in processedCustomFile.Entries)
        {
            if (entry.LnkFiles.Count == 0)
                continue;

            var outDir = Path.Combine(_fluentCommandLineParser.Object.LnkDumpDirectory,
                Path.GetFileName(processedCustomFile.SourceFile));

            if (Directory.Exists(outDir) == false)
                Directory.CreateDirectory(outDir);

            entry.DumpAllLnkFiles(outDir, processedCustomFile.AppId.AppId);
        }
}
```



- Helper methods for common operations
  - Dumping lnk files
  - Extracting bytes from OleCf containers
- Same option exists for automatic jump lists which extracts lnk files not tracked by DestList



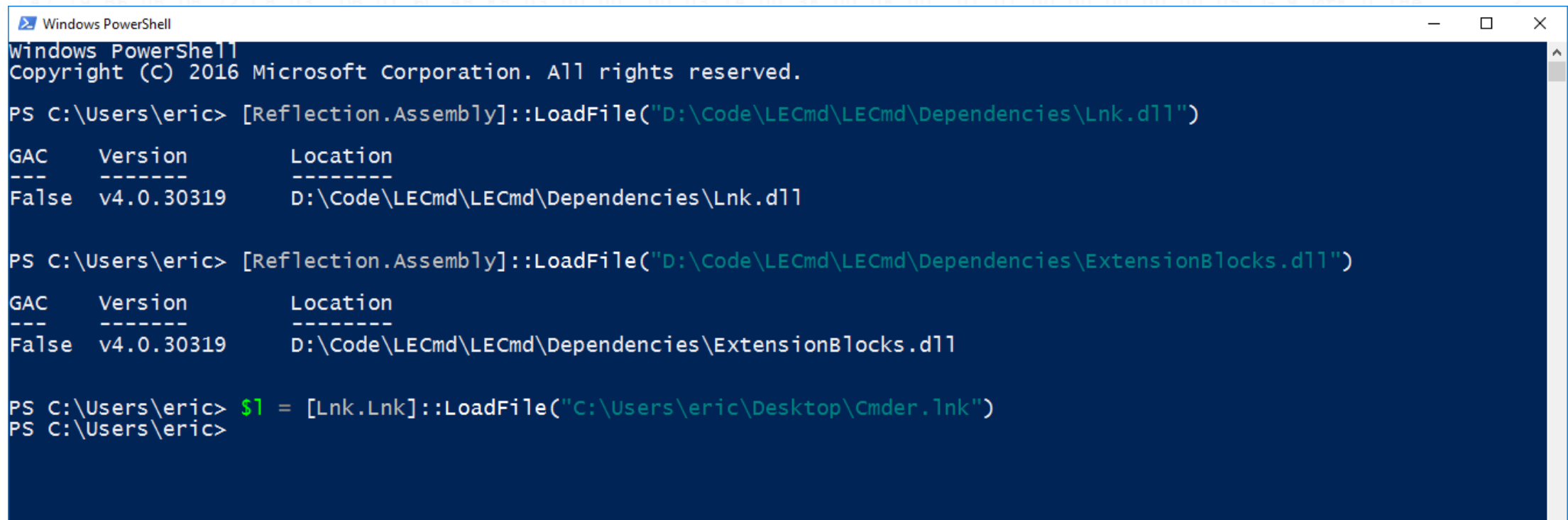
# Programmatic access to everything

The screenshot shows a file explorer window with a tree view on the left and a details pane on the right. The tree view shows a directory structure starting with 'jla' and 'Appld'. The details pane shows properties for the selected directory, including 'Source', 'Appld', 'Header', 'Count', 'Directory Name', 'Directory Type', 'Color', 'Previous Directory Id', 'Next Directory Id', 'Sub Directory Id', 'ClassId', 'CreationTime', 'DirectorySize', 'DirectoryType', 'FirstDirectorySectorId', 'ModifiedTime', 'NextDirectoryId', 'NodeColor', 'PreviousDirectoryId', 'SubDirectoryId', and 'UserFlags'.

Property	Value
{> Source: D:\Dropbox\Jump lists and Inks\5f7b5f1e01b83767.automaticDestinations-ms	Appld: 5f7b5f1e01b83767 ==:
{5f7b5f1e01b83767 ==> Unknown Appld}	
{Header: Version: 3NumberOfEntries: 673NumberOfPinnedEntries: 0LastEntryNumber: 1132LastRevisionNumber: 3704U	
673	
Count = 673	
Count = 675	
{Directory Name: Root EntryDirectory Type: RootStorageNode Color: RedPrevious Directory Id: -1Next Directory Id: -1Su	
{00450020-006e-0074-7200-790000000000}	
null	
"Root Entry"	
611136	
RootStorage	
3	
{2/22/2016 6:09:43 PM +00:00}	
-1	
Red	
-1	
23	
0	
{Directory Name: DestListDirectory Type: StreamNode Color: RedPrevious Directory Id: -1Next Directory Id: -1Sub Direct	
{Directory Name: 1Directory Type: StreamNode Color: BlackPrevious Directory Id: -1Next Directory Id: -1Sub Directory Id	
{Directory Name: 388Directory Type: StreamNode Color: BlackPrevious Directory Id: -1Next Directory Id: -1Sub Directory	
{Directory Name: 3f2Directory Type: StreamNode Color: BlackPrevious Directory Id: 637Next Directory Id: -1Sub Director	
{Directory Name: 29eDirectory Type: StreamNode Color: BlackPrevious Directory Id: 498Next Directory Id: 103Sub Direct	
{Directory Name: 5DDirectory Type: StreamNode Color: BlackPrevious Directory Id: 2Next Directory Id: -1Sub Directory Id:	
{Directory Name: 2dDirectory Type: StreamNode Color: BlackPrevious Directory Id: 30Next Directory Id: 12Sub Directory	
{Directory Name: 3bDirectory Type: StreamNode Color: BlackPrevious Directory Id: -1Next Directory Id: -1Sub Directory I	

- Get everything, not low hanging fruit
- Leave no byte behind!

# Extending PowerShell



```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\eric> [Reflection.Assembly]::LoadFile("D:\Code\LECmd\LECmd\Dependencies\Lnk.dll")

GAC      Version      Location
---      -
False    v4.0.30319      D:\Code\LECmd\LECmd\Dependencies\Lnk.dll

PS C:\Users\eric> [Reflection.Assembly]::LoadFile("D:\Code\LECmd\LECmd\Dependencies\ExtensionBlocks.dll")

GAC      Version      Location
---      -
False    v4.0.30319      D:\Code\LECmd\LECmd\Dependencies\ExtensionBlocks.dll

PS C:\Users\eric> $1 = [Lnk.Lnk]::LoadFile("C:\Users\eric\Desktop\Cmder.lnk")
PS C:\Users\eric>
```

- In newer versions of PowerShell, can also use `Add-Type -path D:\Code\LECmd\LECmd\Dependencies\Lnk.dll -ReferencedAssemblies D:\Code\LECmd\LECmd\Dependencies\ExtensionBlocks.dll`

# Extending PowerShell

```
Windows PowerShell
PS C:\Users\eric> $!

TargetIDs      : {Type: Root folder: GUID, Value: My Computer
                  , Type: Drive letter, Value: D:
                  , Type: Directory, Value: cmdr130
                  Extension blocks found: 1
                  ----- Block 0 (Beef0004)-----
                  Long name: cmdr130
                  Created: 4/4/2016 11:59:32 AM +00:00
                  Last access: 7/20/2016 6:32:58 PM +00:00
                  MFT entry/sequence #: 41993/14 (0xA409/0xE)
                  File system hint: NTFS
                  -----
                  Short name: cmdr130
                  Modified: 7/20/2016 6:32:58 PM +00:00
                  , Short name: Cmdr.exe
                  File size: 129,536
                  Modified On: 7/14/2016 7:22:52.000 AM +00:00
                  Type: File, Value: Cmdr.exe
                  Extension blocks found: 1
                  ----- Block 0 (Beef0004)-----
                  Long name: Cmdr.exe
                  Created: 7/20/2016 6:32:18 PM +00:00
                  Last access: 7/20/2016 6:32:18 PM +00:00
                  MFT entry/sequence #: 123161/1 (0x1E119/0x1)
                  File system hint: NTFS
                  -----
```

# Extending PowerShell

```
PS C:\Users\eric> $1.TargetIDs[2]
```

```
LastModificationTime : 7/20/2016 6:32:58 PM +00:00
LastAccessTime       :
ShortName             : cmdr130
FriendlyName         : Directory
Value                : cmdr130
ExtensionBlocks       : {Signature: 0xbeef0004
                        Size: 66
                        Version: 9
                        Version Offset: 0x18

                        Identifier: 2E (Windows 8.1, 10)

                        Created On: 4/4/2016 11:59:32 AM +00:00
                        Last Access: 7/20/2016 6:32:58 PM +00:00

                        Long Name: cmdr130

                        MFT Entry Number: 41993
                        MFT Sequence Number: 14

                        File system hint: NTFS
                        }
```

- All properties are available and can be accessed individually as needed



# Putting it all together

Mount  
disk

\$MFT

JumpLists

Ink files

Prefetch

Registry hives

json/csv/xml

ELK, Splunk, SIEM, etc

# So who is using it?

- I actively maintain parsers and front ends
- SANS includes many tools in SIFT workstation in FOR408/FOR508
- Troy Larson has integrated several of the parsers into his Azure forensics stack
- You (hopefully!)

# How to get involved?

- Visit my GitHub page at <https://github.com/EricZimmerman>
- Fork projects
- ???
- Profit

# Questions?

