

# Cyber Defense Organization

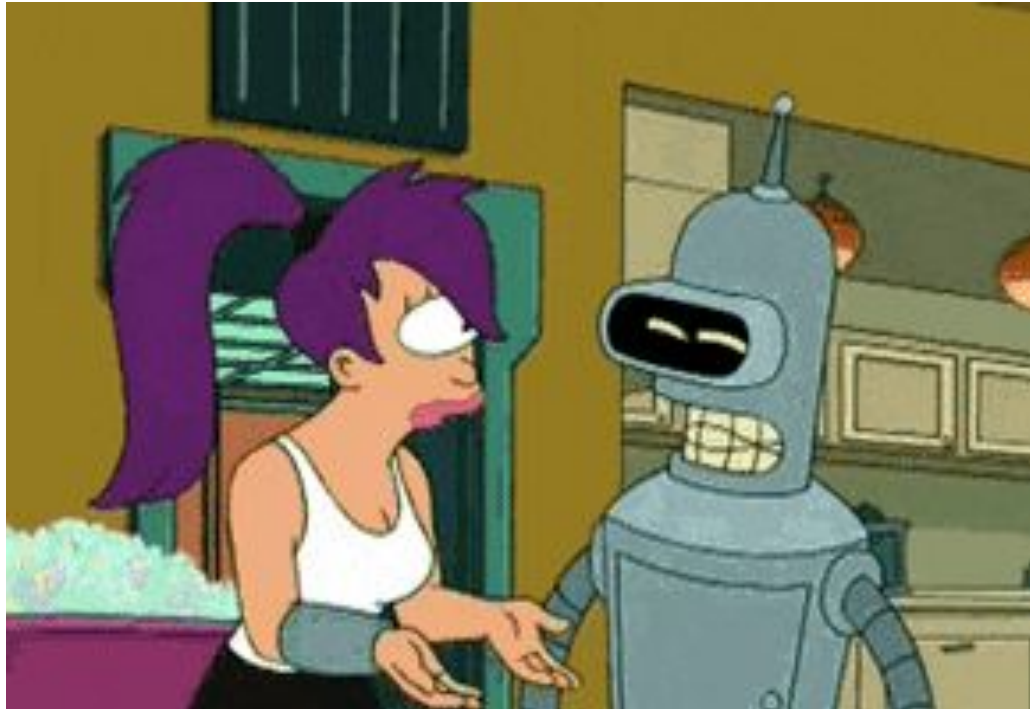
Spring 2020 - Windows Forensics



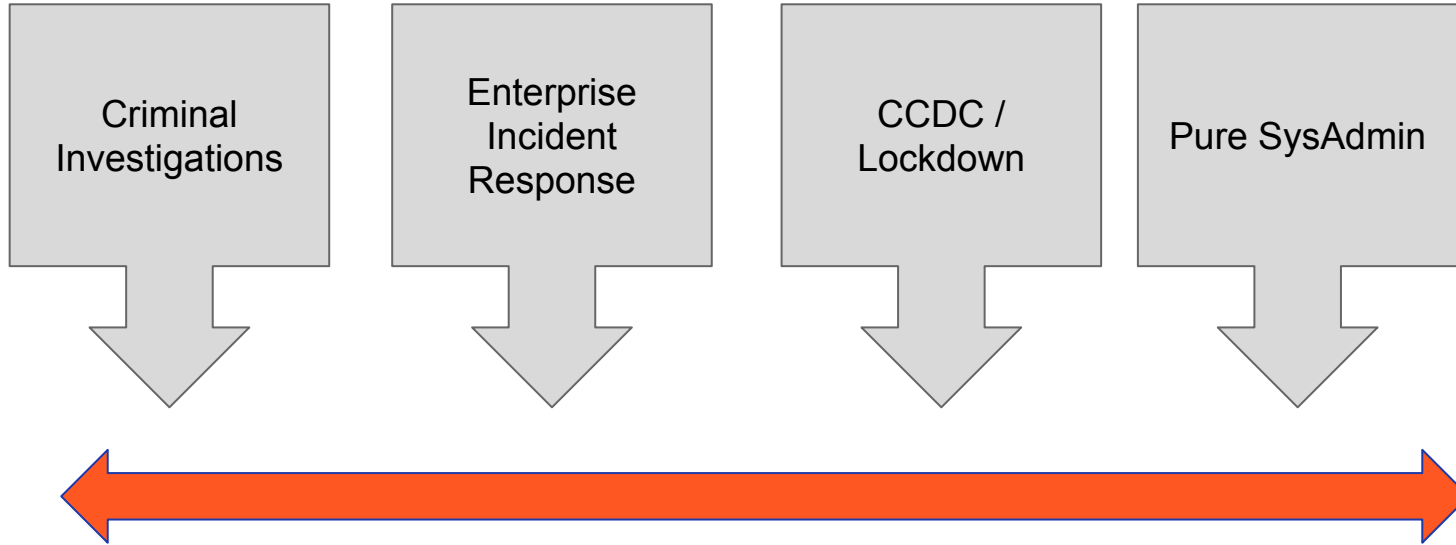
**Word of the week**

**Incident responder** - I need you to take down  
your entire environment while I investigate.

**Sysadmin** -



# **Onto Real world Forensics**



## **Forensics - System Administration Spectrum**

# Dead Box vs Live Response

## Dead Box

- You have access to a full bit-for-bit copy of the disk.
- This can give you a lot more filesystem data, and timelines can be populated with much more info.
- You lose access to volatile data in memory.

## Live Response

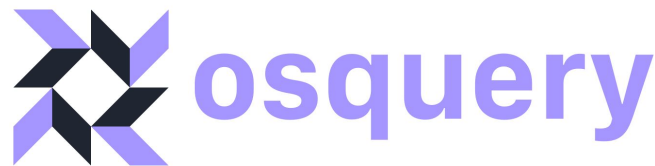
- You install an agent or remotely grab important artifacts.
- You have access to real time events and volatile data.
- Unless you want to wait, you won't get access to every file.

# Live Incident Response at Scale

Most live incident response involved installing an agent, telling it to grab some key artifacts, and then investigating with those artifacts.

You don't often get a full bit for bit copy.

You'll need to put together a story with just bits of information.



# Goals

# Goals

- High level overview a three important sources of forensic artifacts.
- Give you just enough information to get started.
- Using a few artifacts to piece together a story.
- Let you loose on an investigation.

# Event Logs

Security Number of events: 26,290

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	3/6/2020 10:39:28 AM	Microsoft Windows security auditing.	4624	Logon

Event 4624, Microsoft Windows security auditing. X

General Details

An account was successfully logged on.

Subject:

- Security ID: SYSTEM
- Account Name: WILLIAMSMITH-PCS
- Account Domain: WORKGROUP
- Logon ID: 0x3E7

Logon Information:

- Logon Type: 5
- Restricted Admin Mode: -
- Virtual Account: No
- Elevated Token: Yes

Impersonation Level: Impersonation

Log Name: Security

Source: Microsoft Windows security    Logged: 3/6/2020 10:39:28 AM

Event ID: 4624    Task Category: Logon

Level: Information    Keywords: Audit Success

User: N/A    Computer: WilliamSmith-PC

OpCode: Info

More Information: [Event Log Online Help](#)

**Structure: Single Log** (Liam's full name is William, who knew 0.0)

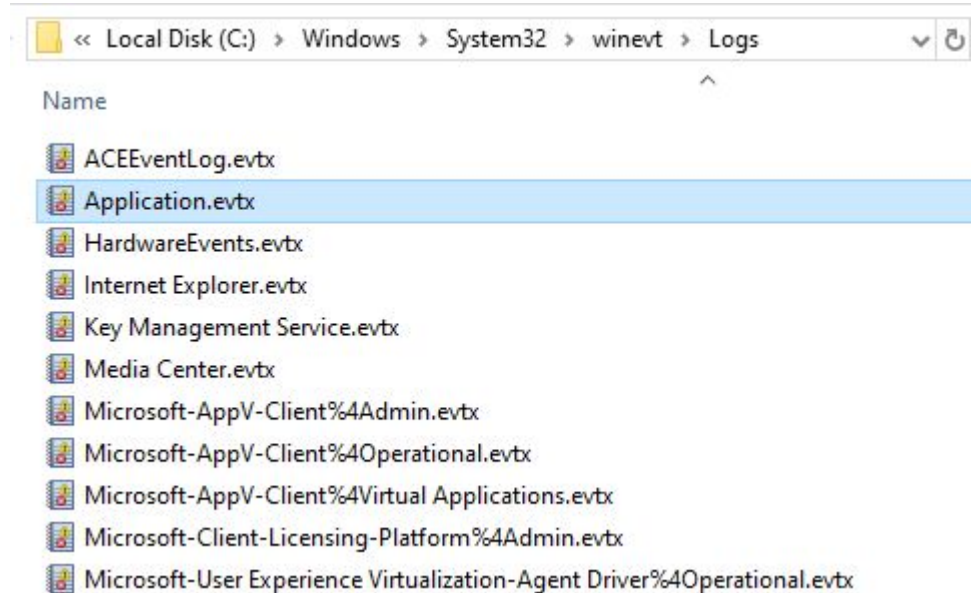
# Log Components

<b>Event ID</b>	A required integer used to categorize events in the event log.
<b>Time and Date</b>	When the event log was recorded, not always when the event happened. UTC.
<b>Log Name</b>	Name of the log on disk.
<b>Source</b>	Program / Process / Task that is writing this log. (WindowsUpdateClient, DistributedCOM)
<b>Task Category</b>	Basic information. (Bumper sticker of the log).
User, Computer, Process ID, Session ID, and a few more.	

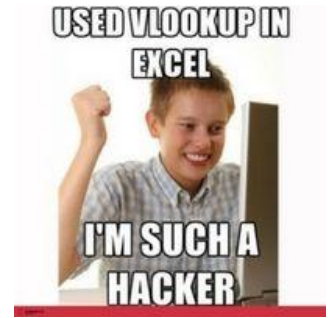
# Event Log Location

All versions of Windows maintain three “core” event logs: **Application**, **System**, and **Security**. Activities related to user programs and commercial off-the-shelf applications populate the Application log.

However there are other important logs!



# What Matters



Windows Event logs track a lot of things, so let's cut to the bone.

- Understand Logon Events, and Logon Types. Lateral movement is tracked here.
- Services are tracked as well. Service install (4697), Service timeout reached (7009).
- Know your event ids, they can make researching a log much easier.
  - [www.myeventlog.com](http://www.myeventlog.com)
- Sysmon is amazing and can add a lot.
- IRL - pump all those event logs to a spreadsheet, sort by time, look at them in Excel.

# **File System**

# New Technology File System - NTFS

NTFS is a proprietary file system developed by Microsoft in the early 1990s.

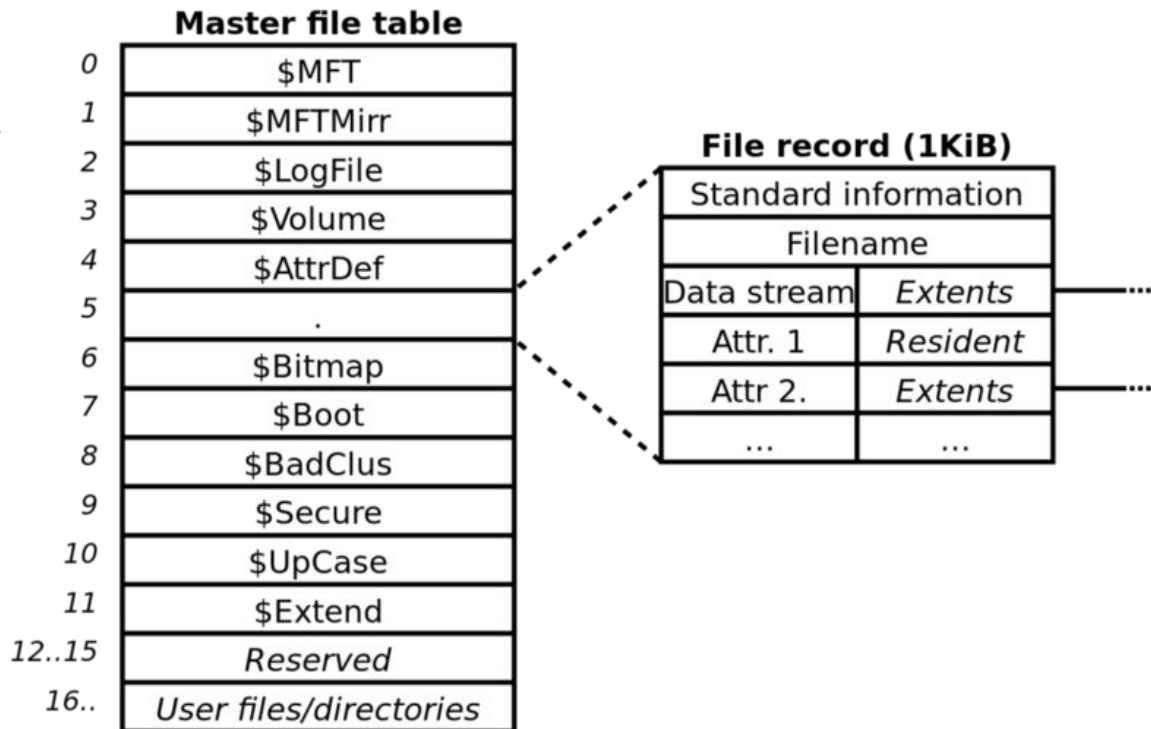
- Hard-links
- Improved performance, reliability and disk space utilization
- Security access control lists
- File system journaling



# \$MFT

The Master File Table is a list of file records (entries), one for each file and directory on volume.

With the MFT you have the authoritative catalog of all files that exist on that volume.



# \$MFT Entries

	A	B	C	D	F	G	I	L
1	EntryNumber ▼	SequenceNumber ▼	InUse ▼	ParentEntryNumber ▼	ParentPath ▼	FileName ▼	FileSize ▼	IsDirectory ▼
2	0	1	TRUE	5	.	\$MFT	115081216	FALSE
3	1	1	TRUE	5	.	\$MFTMirr	4096	FALSE
4	2	2	TRUE	5	.	\$LogFile	24494080	FALSE
5	3	3	TRUE	5	.	\$Volume	0	FALSE
6	4	4	TRUE	5	.	\$AttrDef	2560	FALSE
7	5	5	TRUE	5	.	.	0	TRUE

Record Number	An integer used to identify the given MFT Entry
Record Type	Whether it is a file or directory.
Parent Record Number	The parent MFT entry number.
Active/Inactive Flag	MFT entries for deleted files or directories are marked "Inactive."
Attributes	\$STANDARD_INFORMATION, \$FILENAME, and \$DATA.

# \$SI & \$FN & Time Stamps

**\$STANDARD\_INFO** can be modified by user level processes like timestomp.

**\$FILE\_NAME** can only be modified by the system kernel.

The MAC(b) times are derived from file system metadata and they stand for:

1. Modified
2. Accessed
3. Changed (\$MFT Modified)
4. Birth (file creation time)

FileName	Created0x10	Created0x30	LastModified0x10	LastModified0x30
__output	2019-10-21 02:33:39		2019-10-21 02:33:48	2019-10-21 02:33:39
LastRecordChange0x10	LastRecordChange0x30	LastAccess0x10	LastAccess0x30	
2019-10-21 02:33:48	2019-10-21 02:33:39	2019-10-21 02:33:39		

# Windows® Time Rules

## \$ STANDARD\_INFORMATION

File Creation	File Access	File Modification	File Rename	File Copy	Local File Move	Volume File Move (move via CLI)	Volume File Move (cut/paste via Explorer)	File Deletion
Modified – Time of File Creation	Modified – No Change	Modified – Time of Data Modification	Modified – No Change	Modified – Inherited from Original	Modified – No Change	Modified – Inherited from Original	Modified – Inherited from Original	Modified – No Change
Access – Time of File Creation	Access – Time of Access (No Change only on NTFS Win7+)	Access – No Change	Access – No Change	Access – Time of File Copy	Access – No Change	Access – Time of File Move via CLI	Access – Time of Cut/Paste	Access – No Change
Metadata – Time of File Creation	Metadata – No Change	Metadata – Time of Data Modification	Metadata – Time of File Rename	Metadata – Time of File Copy	Metadata – Time of Local File Move	Metadata – Inherited from Original	Metadata – Inherited from Original	Metadata – No Change
Creation – Time of File Creation	Creation – No Change	Creation – No Change	Creation – No Change	Creation – Time of File Copy	Creation – No Change	Creation – Time of File Move via CLI	Creation – Inherited from Original	Creation – No Change

## \$ FILENAME

File Creation	File Access	File Modification	File Rename	File Copy	Local File Move	Volume File Move (move via CLI)	Volume File Move (cut/paste via Explorer)	File Deletion
Modified – Time of File Creation	Modified – No Change	Modified – No Change	Modified – No Change	Modified – Time of File Copy	Modified – No Change	Modified – Time of Move via CLI	Modified – Time of Cut/Paste	Modified – No Change
Access – Time of File Creation	Access – No Change	Access – No Change	Access – No Change	Access – Time of File Copy	Access – No Change	Access – Time of Move via CLI	Access – Time of Cut/Paste	Access – No Change
Metadata – Time of File Creation	Metadata – No Change	Metadata – No Change	Metadata – No Change	Metadata – Time of File Copy	Metadata – No Change	Metadata – Time of Move via CLI	Metadata – Time of Cut/Paste	Metadata – No Change
Creation – Time of File Creation	Creation – No Change	Creation – No Change	Creation – No Change	Creation – Time of File Copy	Creation – No Change	Creation – Time of Move via CLI	Creation – Time of Cut/Paste	Creation – No Change

**Time Stamp Analysis**

# \$USN\$J

Update Sequence Number (USN) journal, named \$UsnJrnl, tracks changes to files. A given entry includes the type of change event that occurred, its corresponding timestamp, the file name, its attributes, and the MFT entry identifiers for the file and its parent directory.

Located - \ \$Extend\ \$UsnJrnl

\$J : Has the actual change log records. (It's a stream).

Name	EntryNumber	SequenceNumber	ParentEntryNumber	UpdateSequenceNumber	UpdateTimestamp	UpdateReasons
_output	109375	2	5	9411840	2019-10-21 02:31:29	FileCreate
_output	109375	2	5	9411920	2019-10-21 02:31:29	DataExtend   FileCreate
_output	109375	2	5	9412000	2019-10-21 02:31:29	DataExtend   FileCreate   Close

# What Matters

While “Fileless” malware is the trend, if anything touches disk, the file system itself will have traces of it.

- When you have a time frame, file system activity can provide a wealth of information.
- Understanding the Time Stamps, what they mean, and how to use them.
- You can find evidence of deleted files, and even the exact time a file was deleted.\*

\*\$I and \$R and \$Recycle.Bin... story for another time.



# Registry

# What is the Registry?

The Windows Registry is a hierarchical database that stores low-level settings for the Microsoft Windows operating system and for applications that opt to use the registry.

- HKEY\_CLASSES\_ROOT (**HKCR**)
- HKEY\_CURRENT\_USER (**HKCU**)
- HKEY\_LOCAL\_MACHINE (**HKLM**)
- HKEY\_USER (**HKU**)
- HKEY\_CURRENT\_CONFIG (**HKCC**)

Look at it With RegEdit, or Registry Explorer.

## Location on Disk

Hive	Location
AmCache.hve	C:\Windows\AppCompat\Programs\Amcache.hve
SAM	C:\Windows\System32\config\SAM
SECURITY	C:\Windows\System32\config\SECURITY
SOFTWARE	C:\Windows\System32\config\SOFTWARE
SYSTEM	C:\Windows\System32\config\SYSTEM
NTUSER.DAT	C:\Users\(\Username)\NTUSER.DAT
USRCLASS.DAT	C:\Users\(\Username)\AppData\Local\Microsoft\Windows\USRCLASS.DAT

# Demo Time

THIS IS NOT REGISTRY FORENSICS

Many analysts kick off an examination by loading Registry hives into a viewer, without really understanding what it is they're looking for; this will often result in “no findings” and a great deal of time spent finding this out. If you understand what you're interested in and what you're looking for, you can find it very quickly.

**Harlan Carvey**

# So wait, how then?

You break it up by category, and ask: what do I need?

<b>Program Execution</b>	UserAssist, RecentApps, Shimcache, Amcache.hve, MRU, BAM/DAM
<b>Evidence of Interaction</b>	ShellBags, WordWheelQuery
<b>External Drive</b>	USB/USBSTOR, MountedDevices, Volume Serial Number, Volume Name
<b>Computer Configuration</b>	Timezone, NTFS Settings, Computer Name.

# SYSTEM - AppCompat (ShimCache)

What?	Everytime a program is run, the execution is recorded, and after reboot it written to the AppComatCache.
Where?	SYSTEM: ControlSet001\Control\Session Manager\AppCompatCache
Why?	Let me repeat what it is. EVERYTIME a program is run you have a record.
Cache Entry Pos.	Everytime a program is executed it is placed on top of the cash entry position, pushing all other entries "down" one place. 0 is the most recent, and the higher the position number the farther back.
Program Name	This artifact records the full path of the program executed.
Modified Time	<b>Important:</b> This artifact records a time stamp. This time stamp is not the time in which the program was executed. Many people make this mistake. Be better than <del>me</del> them. This time stamp is actually the standard information and tree modified time of the executed file. Don't worry if that doesn't make any sense, just means you should brush up on your NTFS time stamps.

Type viewer   Slack viewer   **AppCompatCache**

Drag a column header here to group by that column

	Cache Entry Po...	Program Name	Modified Time
▼	=	nuc	=
	0	C:\Users\WILLIA~1\AppData\Local\Temp\076FB4A8-E0AA-4D52-B181-604BF46056E6\dismhost.exe	2019-03-19 04:46:23
	1	C:\Users\William Smith\AppData\Local\Temp\076FB4A8-E0AA-4D52-B181-604BF46056E6\DismHost.exe	2019-03-19 04:46:23
	2	C:\Users\William Smith\Desktop\RegistryExplorer_RECmd\RegistryExplorer\RegistryExplorer.exe	2019-04-02 22:20:34
	3	C:\Users\William Smith\AppData\Roaming\autopsy\python_modules\Parse_USNJ\parseusn.exe	2017-10-13 19:36:46
	4	C:\Users\William Smith\AppData\Roaming\autopsy\python_modules\Thumbcache_parser\thumbcache_viewer_cmd.exe	2017-10-13 19:36:46
	5	C:\Program Files\Autopsy-4.13.0\autopsy\ESEDatabaseView\ESEDatabaseView.exe	2019-10-10 22:40:04
	6	C:\Program Files\Autopsy-4.13.0\autopsy\plaso\plaso-20180818-amd64\psort.exe	2019-10-10 22:39:28
	7	C:\Program Files\Autopsy-4.13.0\autopsy\plaso\plaso-20180818-amd64\log2timeline.exe	2019-10-10 22:39:28
	8	C:\Exclusions\CDQR\cdqr.exe	2020-02-28 01:21:21
▶	9	C:\Exclusions\Posh64.exe	2019-10-25 14:55:15
	10	C:\Exclusions\Posh32.exe	2019-10-25 14:55:11

**AppCompat (ShimCache)**

# NTUSER.DAT - UserAssist

Values

UserAssist

Drag a column header here to group by that column

	Program Name	Run Counter	Focus Count	Focus Time	Last Executed
▼	ABC	=	=	ABC	=
	{Program Files X64}\Everything\Everything.exe	8	15	0d, 0h, 03m, 18s	2020-03-06 18:34:04
	{Windows}\regedit.exe	2	1	0d, 0h, 00m, 51s	2020-03-06 18:11:13
	Microsoft.Office.EXCEL.EXE, 15	13	30	0d, 0h, 15m, 13s	2020-03-06 17:53:05

What?	<b>GUI-based</b> programs launched from the desktop are tracked in the launcher on a Windows System.
Where?	NTUSER.DAT: Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist
Important	Because this is in the NTUSER.DAT, you can attribute these program executions to a user.

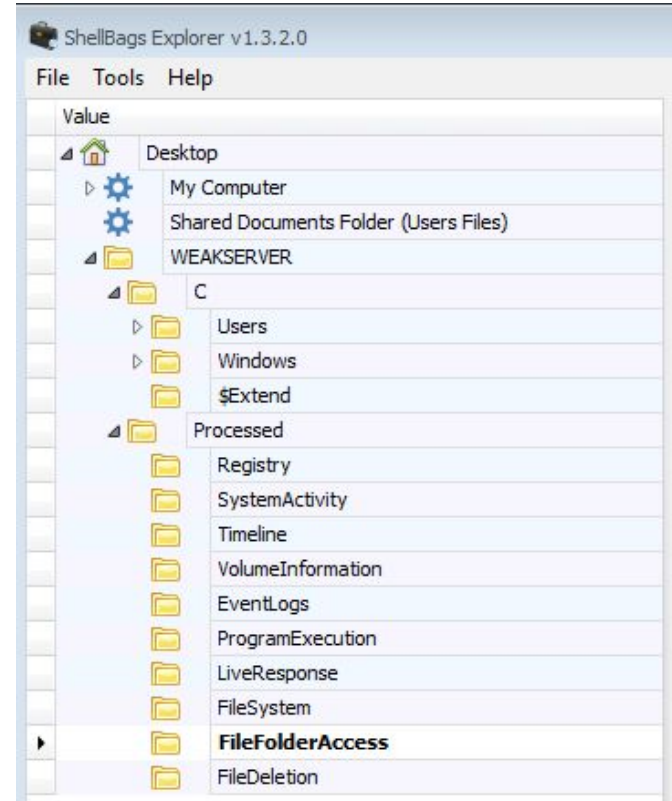
Key name	# values	# subkeys	Last write timestamp
ABC	=	=	=
 PsExec	1	0	2019-08-29 19:28:41

Values			
Drag a column header here to group by that column			
	Value Name	Value Type	Data
▼	ABC	ABC	ABC
▶	EulaAccepted	RegDword	1

**Timestamp rant - PSExec EULA**

# Evidence of Interaction - ShellBags

In a nutshell, shellbags help track views, sizes and positions of a folder window when viewed through Windows Explorer; this includes network folders and removable devices.



# What's Important

- LastWriteTime. That's all you get. Registry keys don't have any other time stamps.
- Know the common program execution artifacts, and their caveats.
- Learn to think “what do I need? What can the registry provide?”

**Activity Time**

# Windows Forensics Activity Sheet

Go to your Downloads file and unzip the weak server zip file

<https://tinyurl.com/windowsforensics>

# Scenario

On 2019-10-21 the weakadmin user logged into the WEAKSERVER Windows Server noticed “owned.txt” on the desktop.

They told the incident response team, and user IncidentResponder logged in and confirmed the existence of the file. They then used CyLR to collect forensically relevant artifacts for analysis.

They used KAPE to parse the artifacts and generate a few reports.

# Choose your own adventure

- CTF style question and answer.
- Look through the reports and parsed artifacts to figure out what happened.
- Parse them yourself.

**Next Time... Spring Break THEN IOT!**



# Add us on Social Media!

Twitter: **@ualbanyCDO**



Instagram: **ualbany\_cdo**



Website: **uacyber.org**



Myinvolvement: **Cyber Defense Org**

**We have a discord!**

