# KAPE + EZ Tools and Beyond

Eric Zimmerman

Senior Vice President, Kroll

Certified Instructor & Author, SANS Institute

Eric Zimmerman's TOOLS

# Review: Architecture

**Parsers**

- In the form of a DLL

**Interfaces**

- CLI
- GUI
- PowerShell

**Outputs**

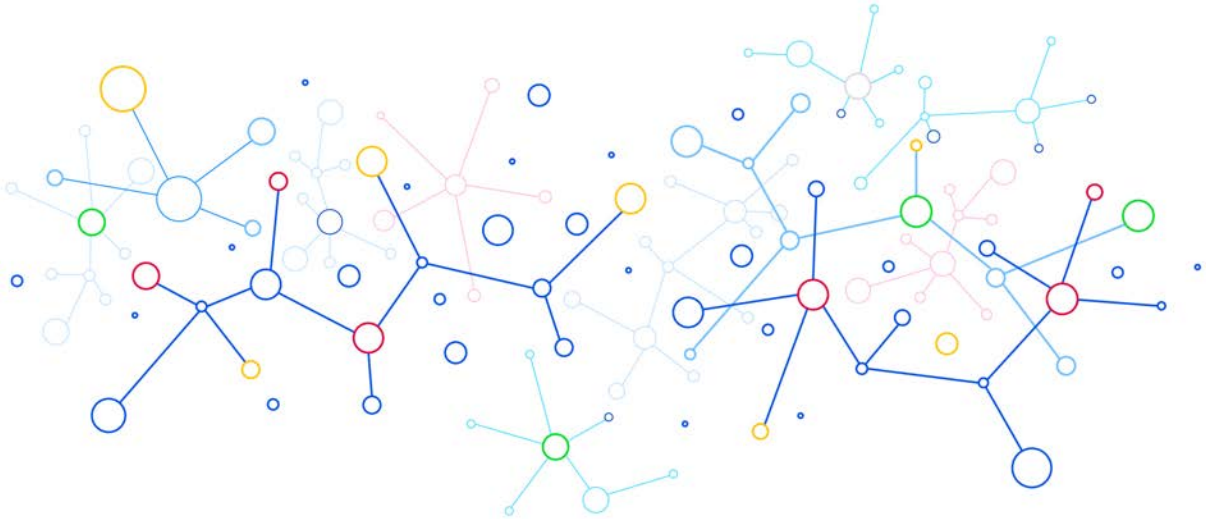
- CSV
- JSON
- XML

# Some new(ish) stuff

- EvtxECmd
- MFTECmd
- Many underlying parsers are showing up as Nuget packages
- With .NET Core 3 here, work will begin to migrate CLI apps to be core compliant (which means cross platform)

# EvtxECmd

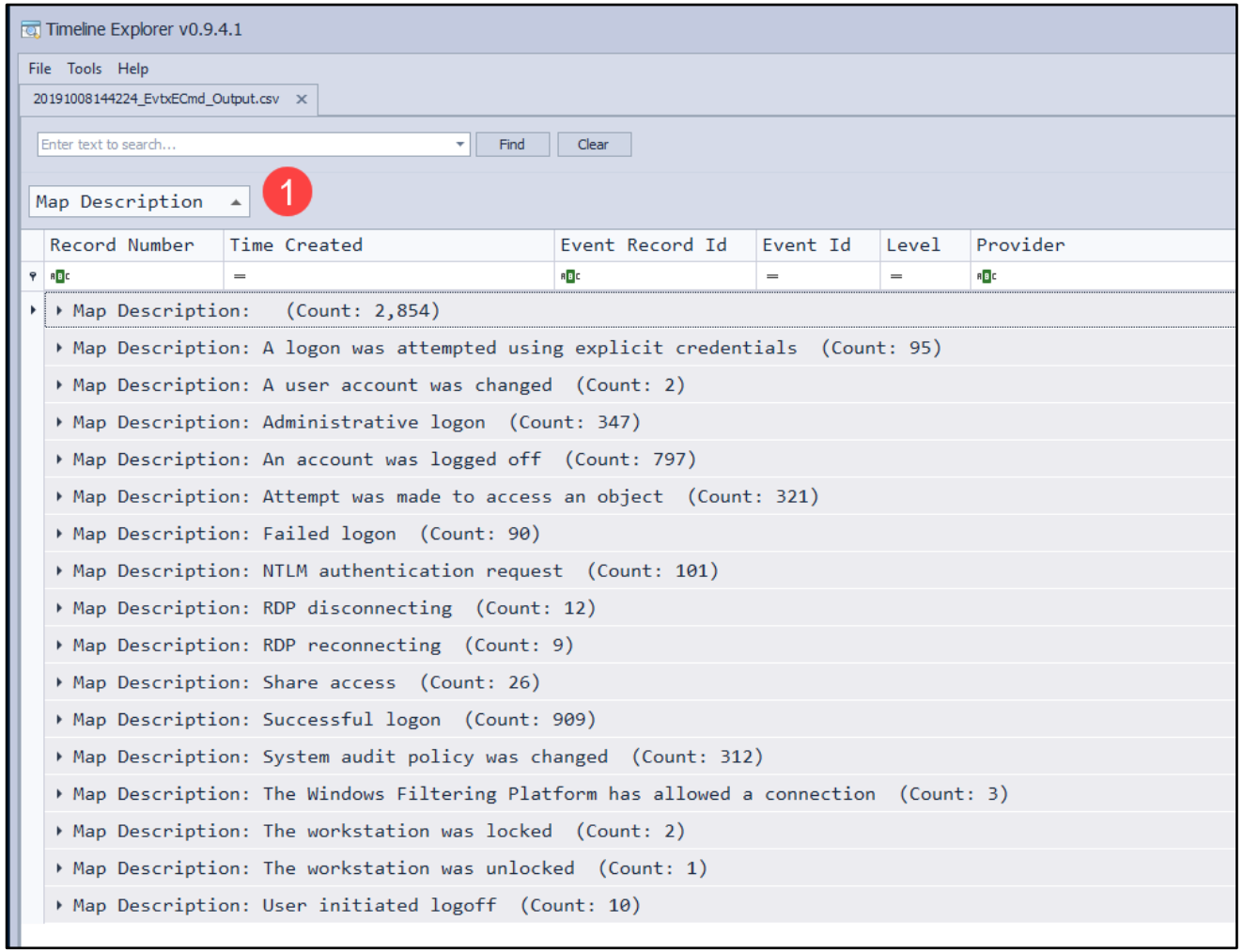
- Single file or recursive directory
- Export to CSV, JSON, and XML
  - Consistent CSV export regardless of event ID
- Flexible event ID inclusion/exclusion
- MAPS!

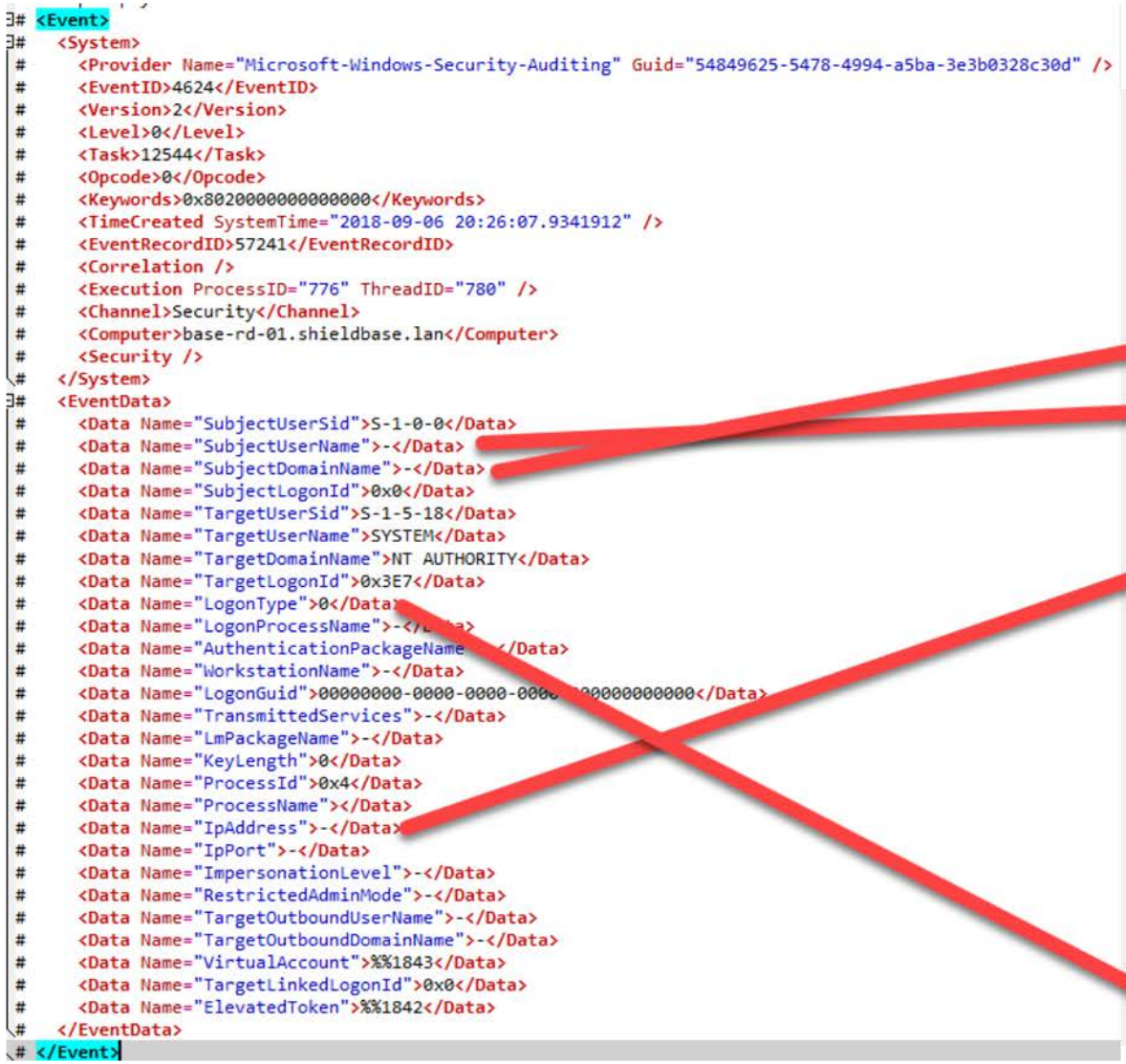


# Why maps?

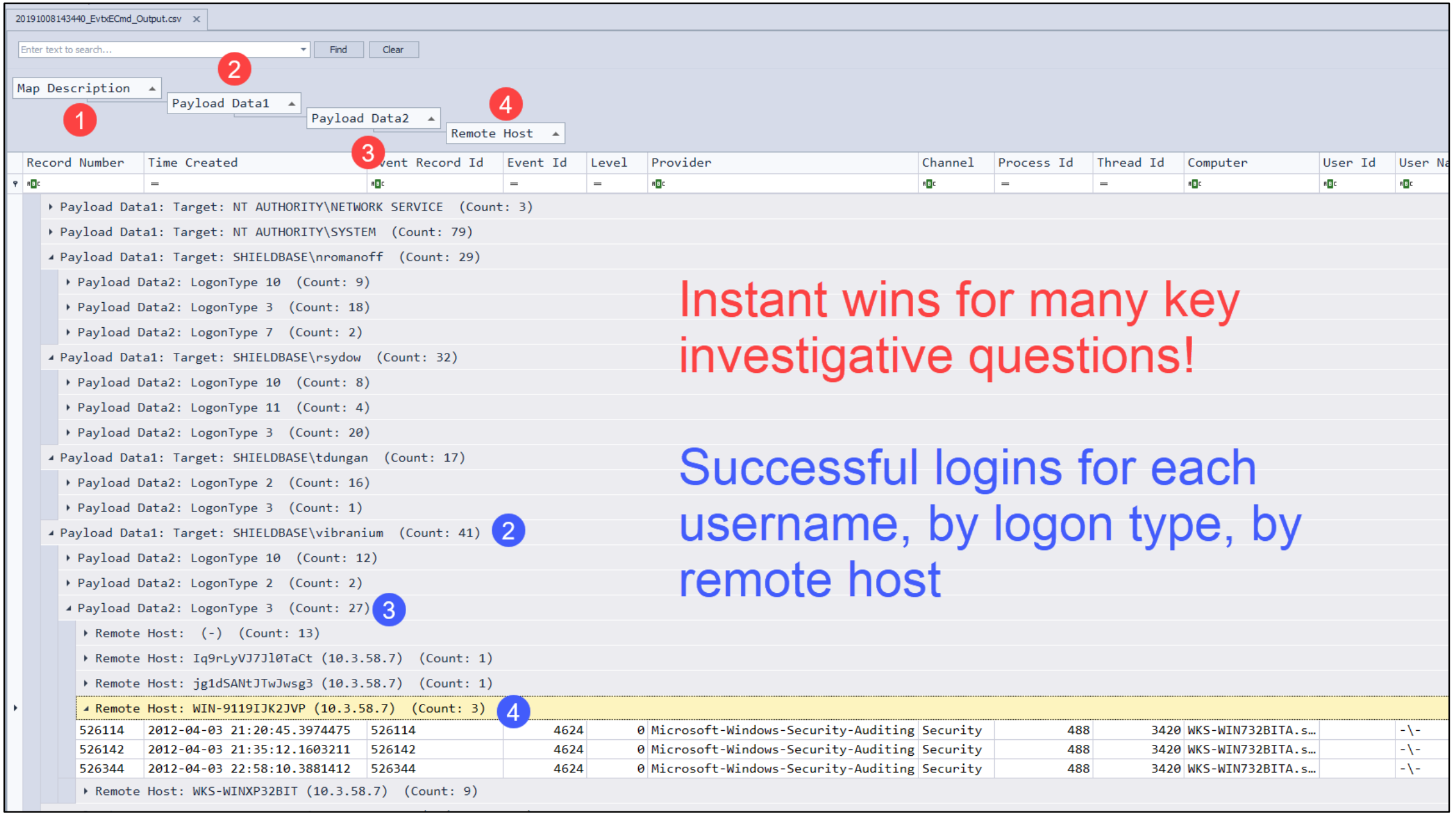
- Customizable
- Flexible
- Easy to make
- Why not?!

```
Event log details
Flags: IsDirty
Chunk count: 30
Stored/Calculated CRC: D4742453/D4742453
Earliest timestamp: 2012-03-29 05:02:59.2593562
Latest timestamp:   2012-04-07 17:38:09.4692524
Total event log records found: 5,892

Records included: 5,891 Errors: 1 Events dropped: 0

Errors
Record #21: Error: Root element is missing.

Processed 1 file in 2.1852 seconds


Files with errors
'D:\SynologyDrive\EventLogs\Romanoff\2\Security.evtx' error count: 1
```






| Map Description | User Name | Remote Host | Payload Data1 | Payload Data2 |
|---|---|---|---|---|
| | | | = Target: SHIELDBASE\vibranium | |
| Successful logon | -\- | WIN-9119IJK2JVP (10.3.58.7) | Target: SHIELDBASE\vibranium | LogonType 3 |
| Successful logon | SHIELDBASE\WKS-WIN732BITA$ | WKS-WIN732BITA (10.3.58.7) | Target: SHIELDBASE\vibranium | LogonType 10 |


20191008143440_EvtxECmd_Output.csv
Enter text to search...
Find
Clear
Map Description
Payload Data1
Payload Data2
Remote Host
1
2
3
4
Record Number
Time Created
Event Record Id
Event Id
Level
Provider
Channel
Process Id
Thread Id
Computer
User Id
User Na
Payload Data1: Target: NT AUTHORITY\NETWORK SERVICE (Count: 3)
Payload Data1: Target: NT AUTHORITY\SYSTEM (Count: 79)
Payload Data1: Target: SHIELDBASE\nromanoff (Count: 29)
Payload Data2: LogonType 10 (Count: 9)
Payload Data2: LogonType 3 (Count: 18)
Payload Data2: LogonType 7 (Count: 2)
Payload Data1: Target: SHIELDBASE\rsydow (Count: 32)
Payload Data2: LogonType 10 (Count: 8)
Payload Data2: LogonType 11 (Count: 4)
Payload Data2: LogonType 3 (Count: 20)
Payload Data1: Target: SHIELDBASE\tdungan (Count: 17)
Payload Data2: LogonType 2 (Count: 16)
Payload Data2: LogonType 3 (Count: 1)
Payload Data1: Target: SHIELDBASE\vibranium (Count: 41)
Payload Data2: LogonType 10 (Count: 12)
Payload Data2: LogonType 2 (Count: 2)
Payload Data2: LogonType 3 (Count: 27)
Remote Host: (-) (Count: 13)
Remote Host: Iq9rLyVJ7Jl0TaCt (10.3.58.7) (Count: 1)
Remote Host: jg1dSANtJTwJwsg3 (10.3.58.7) (Count: 1)
Remote Host: WIN-9119IJK2JVP (10.3.58.7) (Count: 3)
526114 2012-04-03 21:20:45.3974475 526114 4624 0 Microsoft-Windows-Security-Auditing Security 488 3420 WKS-WIN732BITA.s... -\-
526142 2012-04-03 21:35:12.1603211 526142 4624 0 Microsoft-Windows-Security-Auditing Security 488 3420 WKS-WIN732BITA.s... -\-
526344 2012-04-03 22:58:10.3881412 526344 4624 0 Microsoft-Windows-Security-Auditing Security 488 3420 WKS-WIN732BITA.s... -\-
Remote Host: WKS-WINXP32BIT (10.3.58.7) (Count: 9)
Instant wins for many key investigative questions!
Successful logins for each username, by logon type, by remote host

# MFTECmd

- Handles $MFT, $J, $Boot, and $SDS
  - No $LogFile (yet), as it's a horrible format to parse
- Exports to CSV, JSON, or body file
  - CSV has many "helper" filters, like Copied, timestomping, ADS info, etc.
- Has istat and fls emulation modes
- Supports many extended attributes
- For resident files, shows contents as ASCII and Unicode strings

# MFTECmd

| File Name | Extension | Is Directory | Has Ads | Is Ads | File Size | Created0x10 | Created0x30 | Last Modified0x10 | Last Modified0x30 |
|---|---|---|---|---|---|---|---|---|---|
| $MFT | | ☐ | ☐ | ☐ | 965738496 | 2018-12-18 02:11:08.2193228 | | 2018-12-18 02:11:08.2193228 | |
| $MFTMirr | | ☐ | ☐ | ☐ | 4096 | 2018-12-18 02:11:08.2193228 | | 2018-12-18 02:11:08.2193228 | |
| $LogFile | | ☐ | ☐ | ☐ | 67108864 | 2018-12-18 02:11:08.2193228 | | 2018-12-18 02:11:08.2193228 | |
| $Volume | | ☐ | ☐ | ☐ | 0 | 2018-12-18 02:11:08.2193228 | | 2018-12-18 02:11:08.2193228 | |
| $AttrDef | | ☐ | ☐ | ☐ | 2560 | 2018-12-18 02:11:08.2193228 | | 2018-12-18 02:11:08.2193228 | |
| . | | ☑ | ☐ | ☐ | 0 | 2018-04-11 21:04:33.5819148 | 2018-12-18 02:11:08.2193228 | 2019-10-08 16:16:49.0286425 | 2018-12-18 02:11:08.2193228 |
| $Bitmap | | ☐ | ☑ | ☐ | 15593280 | 2018-12-18 02:11:08.2193228 | | 2018-12-18 02:11:08.2193228 | |
| $Bitmap:$SRAT | | ☐ | ☐ | ☑ | 68 | 2018-12-18 02:11:08.2193228 | | 2018-12-18 02:11:08.2193228 | |
| $Boot | | ☐ | ☐ | ☐ | 8192 | 2018-12-18 02:11:08.2193228 | | 2018-12-18 02:11:08.2193228 | |
| $BadClus | | ☐ | ☑ | ☐ | 0 | 2018-12-18 02:11:08.2193228 | | 2018-12-18 02:11:08.2193228 | |

```
**** FILE NAME ****
Type: FileName, Attribute #: 0x6, Size: 0x90, Content size: 0x78, Name size: 0x0, Content offset: 0x18, Resident: True

File name: shutterstock_1379684540.jpg (Length: 0x1B)
Flags: Archive, Name Type: Windows, Reparse Value: 0x0, Physical Size: 0x1BA000, Logical Size: 0x1B97DD
Parent Mft Record: Entry/seq: 0x25884-0xB2

**** DATA ****
Type: Data, Attribute #: 0x8, Size: 0x70, Content size: 0x32, Name size: 0xF, Name: Zone.Identifier, Content offset: 0x38, Resident: True

Resident Data
Data: 5B-5A-6F-6E-65-54-72-61-6E-73-66-65-72-5D-0D-0A-5A-6F-6E-65-49-64-3D-33-0D-0A-48-6F-73-74-55-72-6C-3D-61-62-6F-75-74-3A-69-6E-74-65-72-6E-65-74-0D-0A

ASCII: [ZoneTransfer]
ZoneId=3
HostUrl=about:internet

Unicode: ??????????????????????????

PS D:\Tools>
```

# MFTECmd

```
Processed 'C:\temp\tout\c\$Secure_$SDS' in 0.0227 seconds

SDS entries found in 'C:\temp\tout\c\$Secure_$SDS': 13,380

Details for security record # 5455 (0x154F), Found in 'C:\temp\tout\c\$Secure_$SDS'
Hash value: F0EB9AB4, Offset: 0x408A60
Control flags: SeDaclPresent | SeSaclPresent | SeDaclAutoInherited | SeSaclAutoInherited | SeSelfRelative

Owner SID: S-1-5-18: An account that is used by the operating system.
Group SID: S-1-5-18: An account that is used by the operating system.

Discretionary Access Control List
ACE record count: 15
ACL type: Discretionary

------------ Ace record #0 ------------
Type: AccessAllowed
Flags: ObjectInheritAce | ContainerInheritAce | InheritedAce
SID: S-1-15-2-1: All applications running in an app package context.

------------ Ace record #1 ------------
Type: AccessAllowed
Flags: ObjectInheritAce | ContainerInheritAce | InheritedAce
SID: S-1-15-2-2

------------ Ace record #2 ------------
Type: AccessAllowed
Flags: ObjectInheritAce | ContainerInheritAce | InheritedAce
SID: S-1-5-32-545: A built-in group. After the initial installation of the operating system, the only memb
e Users group on the computer.

------------ Ace record #3 ------------
Type: AccessAllowed
Flags: ObjectInheritAce | ContainerInheritAce | InheritedAce
SID: S-1-15-2-1: All applications running in an app package context.

------------ Ace record #4 ------------
Type: AccessAllowed
Flags: ContainerInheritAce | InheritOnlyAce | InheritedAce
SID: S-1-15-2-1: All applications running in an app package context.

------------ Ace record #5 ------------
Type: AccessAllowed
Flags: InheritedAce
```

```
Contents for '.':

Key                     Name
11-11                   $Extend
231-3                   $Recycle.Bin
13536-21                __vssMount
161164-10               Config.Msi
145-2                   Documents and Settings
827639-1                Program Files
828947-1                Program Files (x86)
829048-1                ProgramData
828050-2                Recovery
83517-2                 System Volume Information
422358-29               temp
829186-1                Users
829243-1                Windows
4-4                     $AttrDef
8-8                     $BadClus
6-6                     $Bitmap
7-7                     $Boot
2-2                     $LogFile
0-1                     $MFT
1-1                     $MFTMirr
9-9                     $Secure
10-10                   $UpCase
3-3                     $Volume
491769-9                hiberfil.sys
139805-10               pagefile.sys
83531-2                 swapfile.sys
```

# Not just dead box files!

- EvtxECmd and MFTECmd can be executed on a running system
  - Just run as an admin
- Handles locked files with optional VSC processing
  - Automatic deduplication based on SHA-1
- Other tools with support
  - AmcacheParser
  - AppCompatCacheParser
  - bstrings
  - Registry Explorer/RECmd
  - ShellBags Explorer

# The problem

**Too many tools**

- How to ensure you run them all?
- Different tools have different syntax

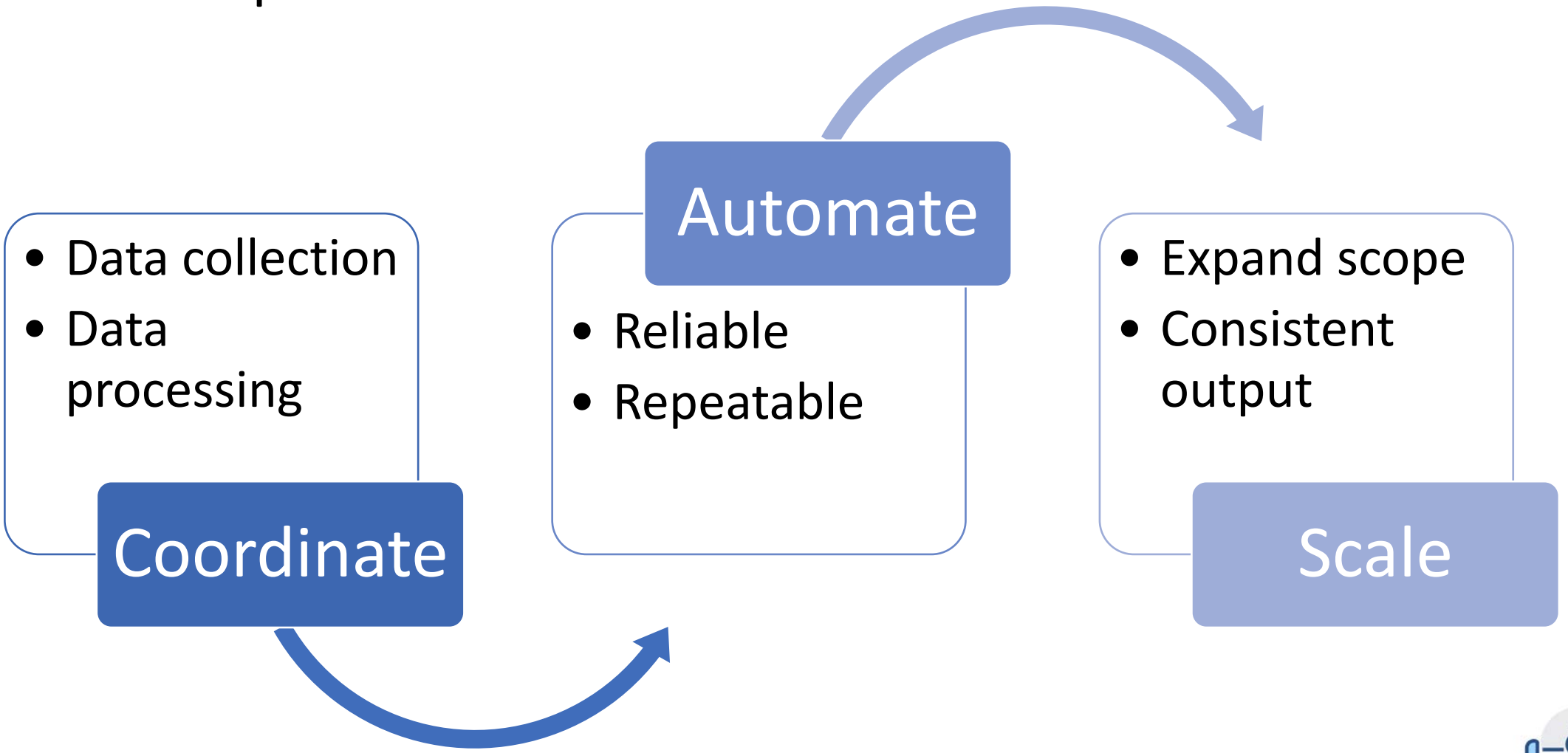
**Disparate output**

- Common categories of tools may not end up together
- Different output formats

**General**

- Using different options each time
- Omitting options
- Exporting data incorrectly

# The requirements



# The solution: KAPE!

## Search

- Mounted E01
- Live drive
- Directory
- F-Response
- Optional VSCs

## Collect

- Forensically sound
- Detailed copy log
- Optional VHD(x) or Zip container
  - SFTP/S3/Azure transfer

## Process

- Repeatable/scalable
- Extract actionable intelligence
- Facilitate artifact correlation/pivoting

# Why KAPE?

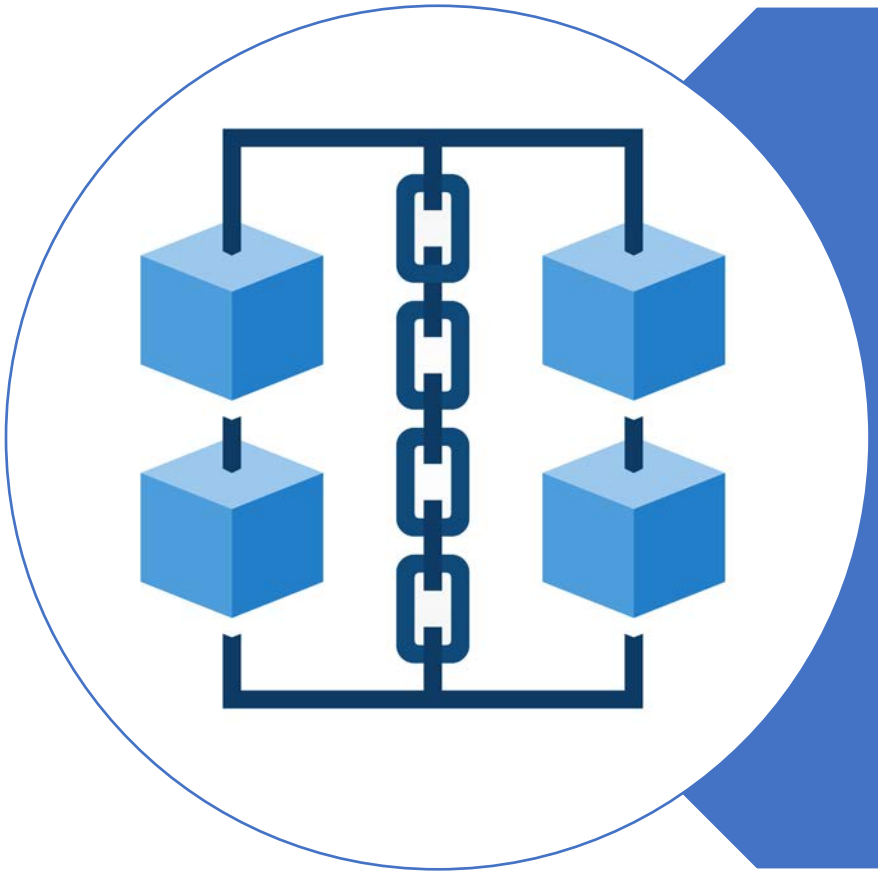
Customized to your needs

Extensibility by adding new targets and modules

Create consistent processing ***tool chains***

KAPE is the “glue” that ties things together, from collection to processing

# What is a toolchain?



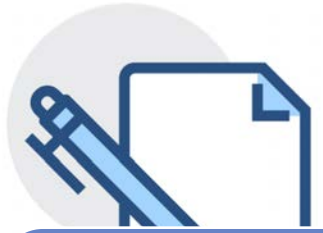
A set of targets and modules grouped together for a given investigative need that is:

- Thorough
- Repeatable
- Scalable
- Auditable

# Toolchain creation (Think ‘DFIR’)

**D**etermine

- What to collect
- What to process
- Where to put it



**F**abricate

- Custom compound targets
- Custom compound modules

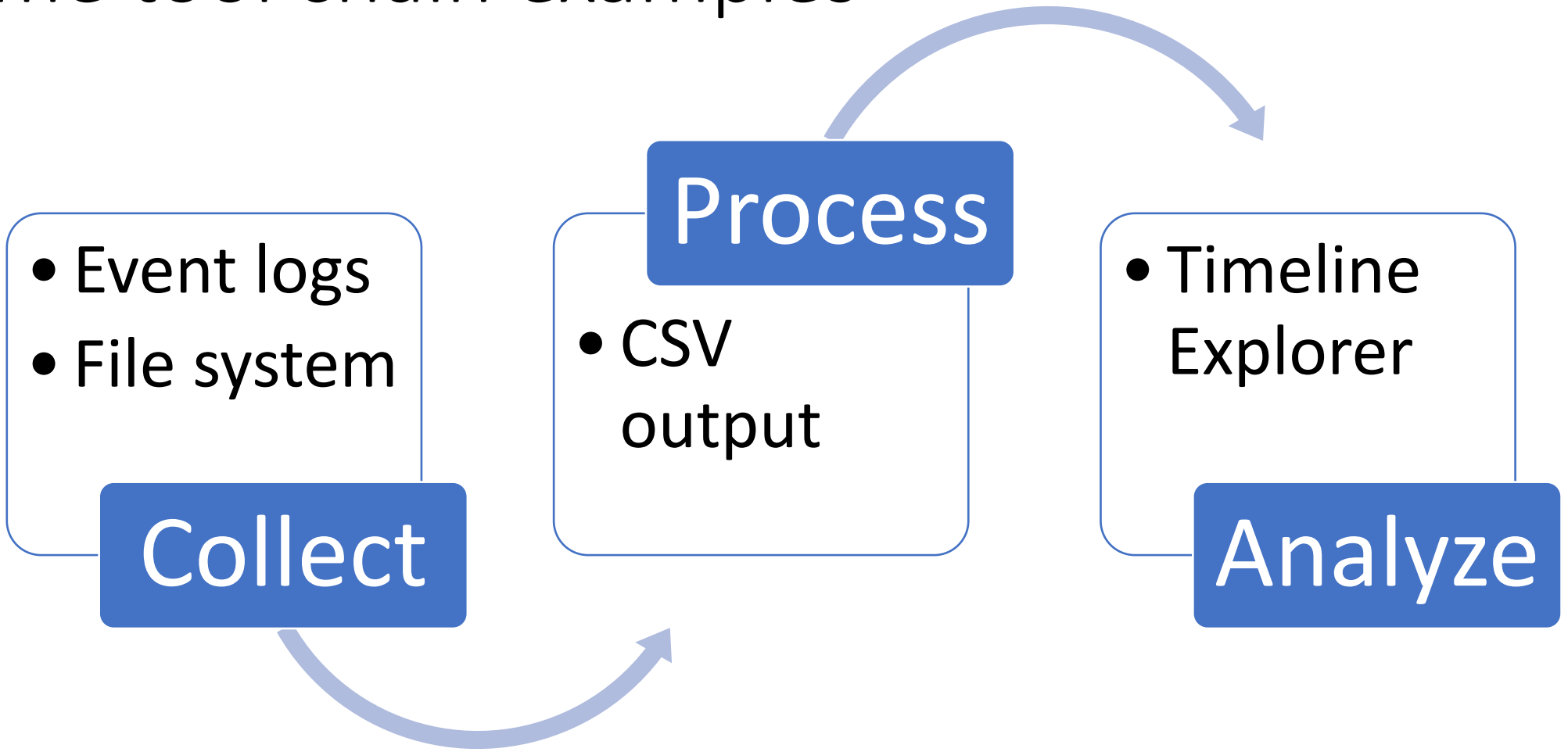
**I**nvoke

- KAPE using custom configurations

**R**eview

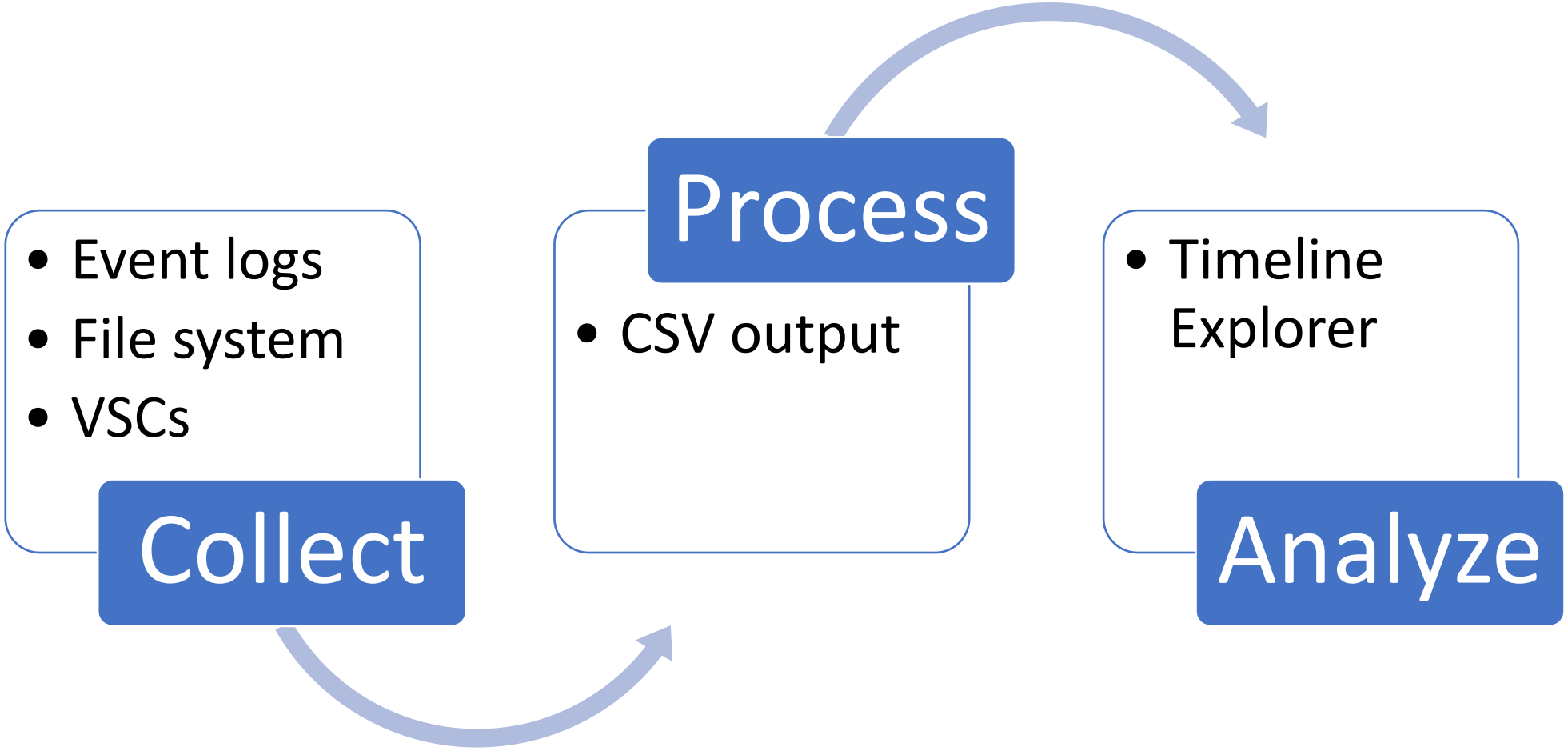
- Categories of output
- Grouped by question to be answered

# Some tool chain examples



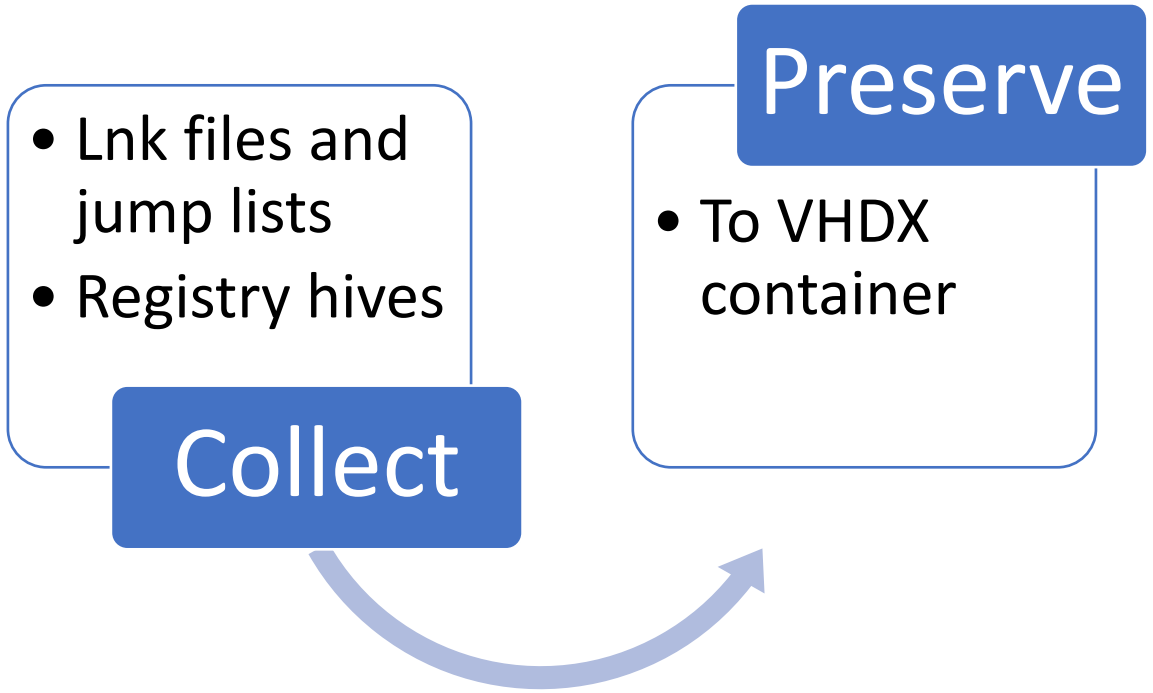

kape.exe --tsource c --tdest c:\temp\tout --tflush --target EventLogs,FileSystem --module EvtxECmd,MFTECmd_$J,MFTECmd_$MFT --mflush

# Same as previous, but with VSCs



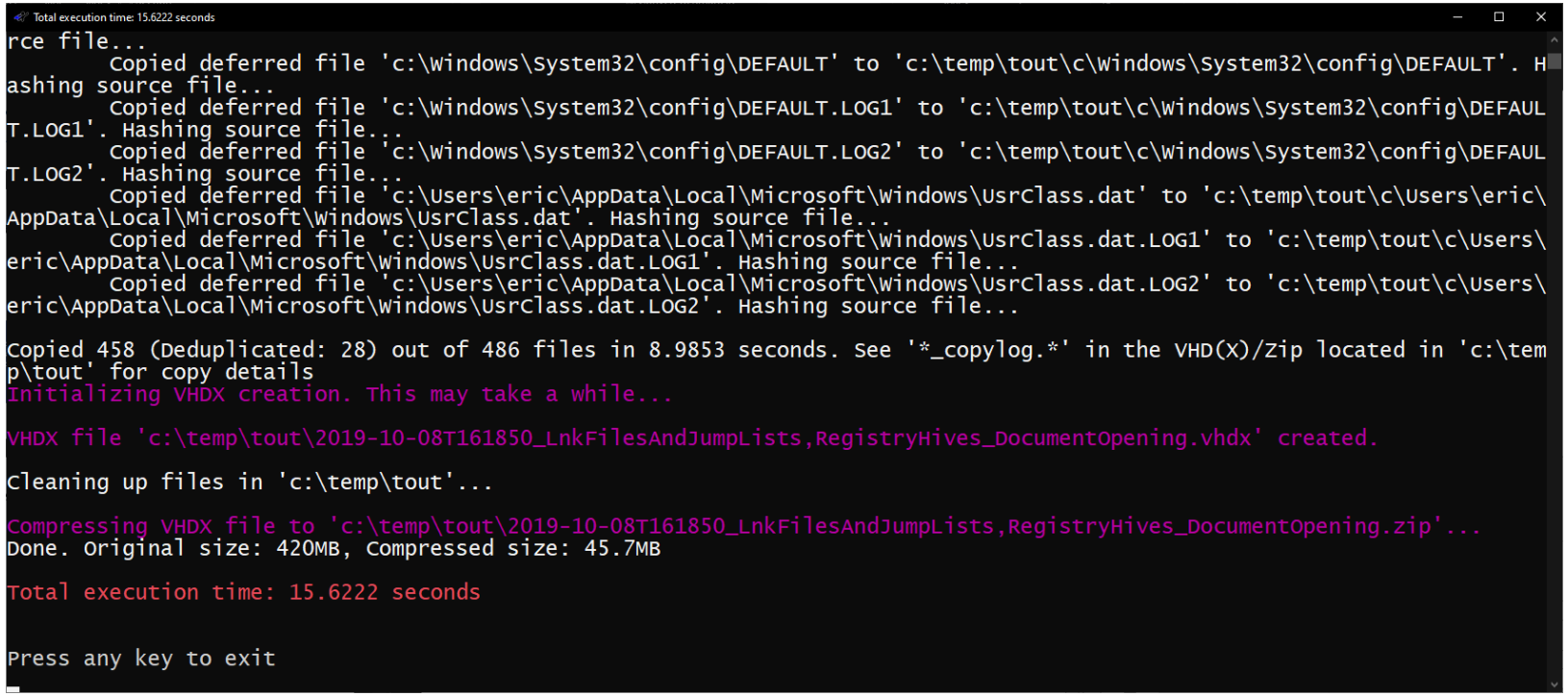

kape.exe --tsource c --tdest c:\temp\tout --tflush --target EventLogs,FileSystem **--vss** --module EvtxECmd,MFTECmd_$J,MFTECmd_$MFT --mflush

# Collect only, save to container




kape.exe --tsource c --tdest c:\temp\tout --tflush --targetLnkFilesAndJumpLists,RegistryHives **--vhdx DocumentOpening**

# Collect only, save to container

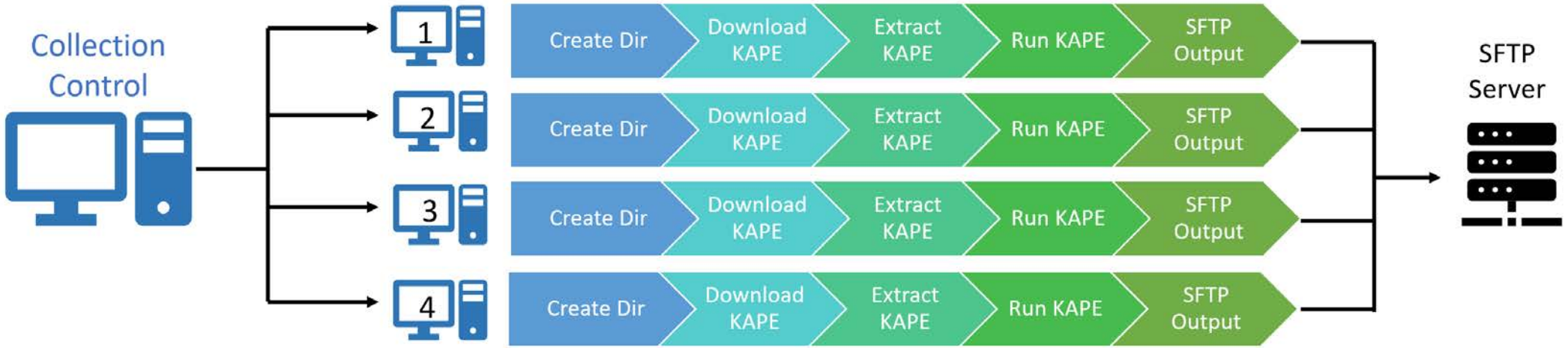




- VHDX can be mounted in Windows or loaded into any forensic tool that understands the container format
- Containers are compressed for faster transport if need be

# Yea, but does it scale?

- Call KAPE using Carbon Black, CrowdStrike Falcon, Tanium, etc.
- Using PowerShell
  - Async downloading, execution, and pushing of data to central location
  - More details available from Mark Hallman and Carlos Cajigas, including example code and set up step by step



# Questions?