Lab Report

Name:	Inor Wang
Title:	Triage Capture with Kape
Case:	25-T106
Date:	10/17/2025

Table of Contents

Document Revision History

Name	Revision Date	Version	Description
Inor Wang	10/17/2025	0.1	Draft

Executive Summary

On October 17, 2025, the examiner, Inor Wang, conducted a triage acquisition and review of Windows artifacts from a domain controller (DC01) and a workstation (DESKTOP) for Case 25-T106. **The objective was to rapidly preserve high-value artifacts (e.g., registry hives, Prefetch, Amcache, RecentFileCache) to support timeline reconstruction around a suspected September intrusion**. Triage was performed using KAPE with the !SANS_Triage compound target; images were mounted read-only with Arsenal Image Mounter to maintain forensic soundness.

Evidentiary integrity was verified against provided cryptographic hashes (MD5/SHA1/SHA256) prior to processing. KAPE was executed with Container=ZIP, Deduplicate=On, and Process VSCs=Off, writing to a controlled evidence directory. Tooling included KAPE/gkape, PowerShell (for hashing and quick target inspections), and standard Windows file system viewers to confirm pathing and collected profile content.

Key findings from the examination are as follows:

- **Target set validation & scope:** !SANS_Triage ID 1bfbd59d-6c58-4eeb-9da7-1d9612b79964 ; KapeTriage ID a745b730-d6b7-4cb7-9847-4e896d9f3c52 ; !SANS_Triage 21 total (18 active, 3 inactive) ; KapeTriage 14 active ; MessagingClients in !SANS_Triage only ; EvidenceOfExecution targets = Prefetch, RecentFileCache, Amcache, Syscache ; Prefetch paths = C:\Windows\Prefetch\ and C:\Windows.old\Windows\Prefetch\ ; FileMask *.pf.
- **Acquisition results (DC01):** 18 targets matched ; 323 files found ; 298 copied ; 25 duplicates skipped ; 7 deferred (UnauthorizedAccessException) ; ZIP ≈ 40.1 MB ; root folder F:\ ; $MFT present ; user profiles = Administrator (Default excluded) ; The command is ".\kape.exe --tsource F: --tdest "C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped" --tflush --target !SANS_Triage --zip dc01_collection_v1 --gui" ; Used F: as the mount letter.
- **Acquisition results (DESKTOP):** 18 targets matched ; 3,183 files found ; 2,633 copied ; 550 duplicates skipped ; 9 deferred (UnauthorizedAccess / NotSupported / long-path) ; ZIP ≈ 95.3 MB ; user profiles = Admin, Administrator, mortysmith, ricksanchez (Default

excluded) ; SAM SHA256 = 6429738382642740791F9934C58A846E697A1871BBF244B5381ABFC20883C9C ; Used E: as the mount letter.

In conclusion, the KAPE triage successfully preserved core Windows execution and account artifacts from both **DC01** and **DESKTOP**, producing verifiable, deduplicated ZIP archives suitable for deeper analysis. The collections include Prefetch/Amcache/RecentFileCache/Syscache, registry hives (including **SAM**), and user profile data necessary to build a timeline of execution and login activity around the suspected September incident. Deferred items were limited and attributable to access restrictions, unsupported objects, or long-path limitations; these can be re-attempted with targeted methods if needed. Overall, the triage provides a sound evidentiary foundation for subsequent timeline, persistence, and attribution analysis.

Synopsis

Memory and disk images for a domain controller (DC01) and a workstation (DESKTOP) were provided for offline triage to answer the lab's investigative questions per the rubric (verification, mounting via Arsenal Image Mounter, and KAPE collection using !SANS_Triage/KapeTriage). The KAPE triage captures serve as the primary evidence set, documenting system configuration and execution artifacts (e.g., Prefetch and EvidenceOfExecution), user profiles, registry hives, target counts, duplicate/deferred files, archive sizes, and required hashes to support each graded item. The client requested a step-by-step, reproducible workflow with annotated screenshots and explicit timestamps supporting each finding.

Client Questions:

Compare the !SANS_Triage to the KapeTriage collection

1. What is the ID of !SANS_Triage?
2. What is the ID of KapeTriage?
3. How many target collections are in !SANS_Triage?
4. How many target collections are in KapeTriage?
5. The !SANS_Triage collects information from messaging clients, does Kape Triage?
6. Find the EvidenceofExecution collection. What targets are in this collection?
7. Find the Prefetch collection; there are two targets in this collection. What are the target paths this is collecting from?
8. What is the file mask for this collection?
9. Before you execute the triage collection, ensure all is configured properly. What is the command line output for the DC01 collection?

KAPE collection analysis DC01

1. There should have been 18 targets found, how many files were found/collected?
2. How many files were "deferred?"
3. Of the files copied, how many were duplicated?
4. What is the size (in MB) of the KAPE archive (ZIP) file for the domain controller image?
5. Extract the DC01 collection. What is the name of the root folder in this collection? Why is it named this way? Why not "C:\"

6. Is the MFT file in this collection? What is the file's name?
7. What user profiles were captured in this collection? (Do not include default).

KAPE collection analysis DESKTOP

1. There should have been 18 targets found, how many diles were found/collected?
2. How many files were "deferred?"
3. Of the files copied, how many were duplicated?
4. What is the size (in MB) of the KAPE archive (ZIP) file for the desktop image?
5. What user profiles were captured in this collection? (Do not include default).
6. Extract the SAM registry hive from this capture. What is the SHA256 hash of this file?

Scope of Work:

- Acquisition and triage of **DC01-E01.zip** and **DESKTOP-E01.zip** (mounted via Arsenal), not *link-file-capture.vmdk.*
- Verification of evidentiary integrity using MD5, SHA1, and SHA256 cryptographic hashes.

Evidence Analyzed

This section provides details of the digital evidence collected

Evidence ID	**E001**
Name	DC01-E01.zip
Type	Zip archive data, at least v2.0 to extract, compression method=store
Size	4,836,649,413 bytes (4.83 GB)
MD5	E57FC636E833C5F1AB58DFACE873BBDE
SHA1	29F841501B76CE461EC1049E21D769D151246D69
SHA256	EFE06D12388DBC000FA4AE306746DDACA3893A6CDBD55311B52F5833E717ACD9

Evidence ID	**E002**
Name	DESKTOP-E01.zip
Type	Zip archive data, at least v2.0 to extract, compression method=deflate
Size	6.37 GB
MD5	71C5C3509331F472ABCDF81EB6EFFF07
SHA1	C56B619B5A4ADD5F269FB6731543CF7BA759DB0D
SHA256	N/A (Not Provided)

Tools Used

Workstation

Hostname	Operating System	Build	Physical / Virtual	Built
IS-4523-001-WINDOWS	Windows 11	2021	Virtual	09/06/2025

Software

Name	Version	Release	Purpose
Arsenal Image Mounter (Arsenal Recon)	3.11.307	Apr 2025	Mount the DC01 and DESKTOP disk images read-only as local disks, assign drive letters to the system volume (with Windows, Program Files, Users), and expose the file system so KAPE can target the correct mounted volume without altering evidence.
Kape (Kroll)	1.2.0.0	Jun 2025	Run the !SANS_Triage (and compare to KapeTriage) collection against the mounted volume to rapidly acquire key artifacts (e.g., Prefetch, EvidenceOfExecution, registry hives, event logs, LNK/Jumplists, user profiles) into a ZIP named DC01/DESKTOP, producing the counts, deferred/duplicate stats, and data needed to answer the rubric questions.
PowerShell (Windows PowerShell)	5.1	2016	Search and document .tkape configs (Select-String for Name/Path/FileMask/ID), compute hashes (e.g., Get-FileHash -Algorithm SHA256 .\SAM), and capture command-line evidence for repeatability.
Windows Explorer (File Explorer)	N/A	N/A	Validate archive contents and folder mappings (e.g., root F:\/E:\), confirm presence of key artifacts ($MFT, registry hives, \Users* profiles), and take screenshots of directory listings for the report.
Notepad	N/A	N/A	View and document .tkape target files for !SANS_Triage, KapeTriage, EvidenceOfExecution, and Prefetch (recording IDs, target names, paths, and FileMask values).

Analysis Findings

Overview of Examination Procedures

The examiner mounted the provided images (DC01-E01.zip, DESKTOP-E01.zip) read-only using Arsenal Image Mounter and identified the system volumes. Using KAPE (gkape v1.2.0.0), the !SANS_Triage compound target was executed against each mounted volume with Container=ZIP, Deduplicate=On, Process VSCs=Off. Command lines and run logs were preserved to document targets matched, files found/copied, duplicates skipped, and deferred items. Post-acquisition, artifacts were spot-verified (e.g., presence of $MFT, registry hives, user profiles) and hashes were computed where required (e.g., SAM).

Additional targeted analysis was performed using:

- **Arsenal Image Mounter (Arsenal Recon)** — Mounted DC01-E01.zip and DESKTOP-E01.zip read-only and identified the Windows system volumes (mounted as F: and E:).
- **KAPE / gkape (Kroll)** — Executed the !SANS_Triage compound target with Container=ZIP, Deduplicate=On, Process VSCs=Off; preserved command lines and logs.
- **PowerShell** — Quick inspections of .tkape target files (e.g., Select-String for Name:, Path:, FileMask:), and hashing with Get-FileHash (e.g., SAM SHA256).
- **Windows Explorer** — Validated archive contents, confirmed root folder mapping (e.g., F:\), and verified presence of key artifacts (e.g., $MFT, registry hives, user profile folders).
- **Notepad** — Viewed and documented .tkape configurations (IDs, targets, paths, file masks) for !SANS_Triage, KapeTriage, EvidenceOfExecution, and Prefetch.

Throughout the process, all findings were documented, and cryptographic hash values were maintained for validation.

Evidence Reviewed

1. **DC01-E01.zip (E001): Windows system image (domain controller)**
2. **DESKTOP-E01.zip (E002): Windows desktop workstation image**

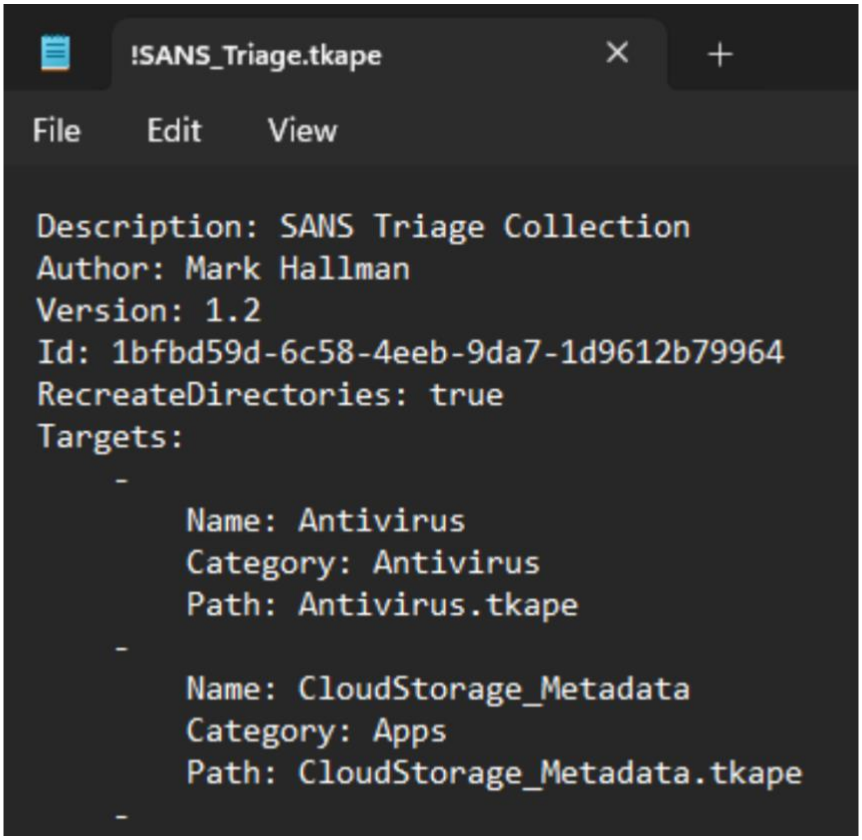
Key Findings

Compare the !SANS_Triage to the KapeTriage collection

1. What is the ID of !SANS_Triage?

- **Analysis Performed:**
    - The !SANS_Triage.tkape file was analyzed through the Notepad application.
    - The examiner navigated to the !SANS_Triage.tkape which is a configuration file that tells KAPE what forensic artifacts to collect and where to find them on a system or disk image.
    - Path: *C:\Users\inorw\Desktop\Tools\Kape\Targets\Compound\!SANS_Triage.tkape*
- **Answer:**
  The ID of !SANS_Triage is "**1bfbd59d-6c58-4eeb-9da7-1d9612b79964**", as shown in Figure 1.

- **Supporting Evidence:**



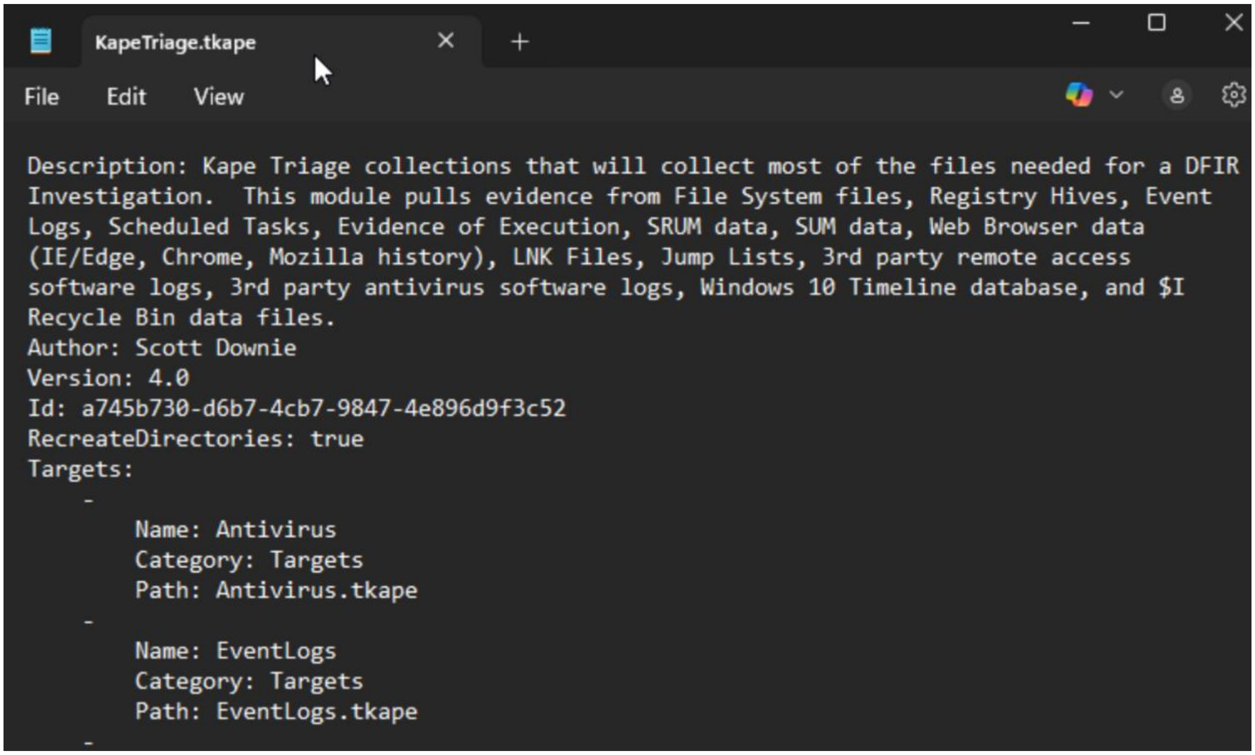
```
Description: SANS Triage Collection
Author: Mark Hallman
Version: 1.2
Id: 1bfbd59d-6c58-4eeb-9da7-1d9612b79964
RecreateDirectories: true
Targets:
    -
        Name: Antivirus
        Category: Antivirus
        Path: Antivirus.tkape
    -
        Name: CloudStorage_Metadata
        Category: Apps
        Path: CloudStorage_Metadata.tkape
    -
```

Figure 1. ID information of !SANS_Triage.tkape

2. What is the ID of KapeTriage?

- **Analysis Performed:**
    - The KapeTriage.tkape file was analyzed through the Notepad application.
    - The examiner navigated to the KapeTriage.tkape which is a configuration file that tells KAPE what forensic artifacts to collect and where to find them on a system or disk image.
    - Path: *C:\Users\inorw\Desktop\Tools\Kape\Targets\Compound\KapeTriage.tkape*
- **Answer:**
  The ID of KapeTriage is "**a745b730-d6b7-4cb7-9847-4e896d9f3c52**", as shown in Figure 2.

- **Supporting Evidence:**



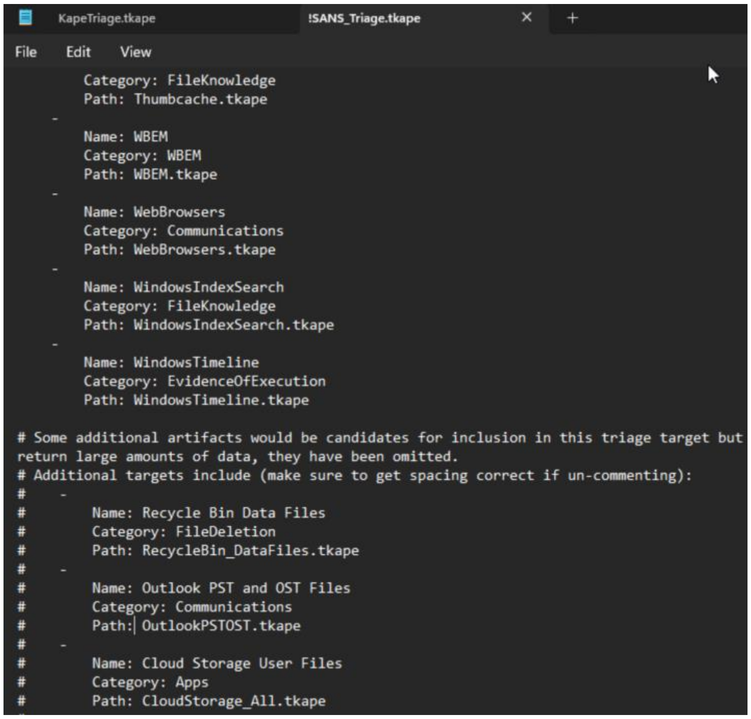
```
Description: Kape Triage collections that will collect most of the files needed for a DFIR
Investigation.  This module pulls evidence from File System files, Registry Hives, Event
Logs, Scheduled Tasks, Evidence of Execution, SRUM data, SUM data, Web Browser data
(IE/Edge, Chrome, Mozilla history), LNK Files, Jump Lists, 3rd party remote access
software logs, 3rd party antivirus software logs, Windows 10 Timeline database, and $I
Recycle Bin data files.
Author: Scott Downie
Version: 4.0
Id: a745b730-d6b7-4cb7-9847-4e896d9f3c52
RecreateDirectories: true
Targets:
    -
        Name: Antivirus
        Category: Targets
        Path: Antivirus.tkape
    -
        Name: EventLogs
        Category: Targets
        Path: EventLogs.tkape
    -
```

Figure 2. ID information of KapeTriage.tkape

3. How many target collections are in !SANS_Triage?

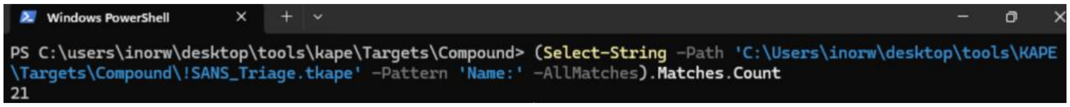
- **Analysis Performed:**
  - The !SANS_Triage.tkape file was analyzed through the Powershell and Notepad application.
  - The examiner accessed the !SANS_Triage.tkape file through Notepad and noted that there were three additional targets that were not in use, as shown in Figure 3.
  - Individuals are able to count the number of targets within the collection files, the examiner created a quick command to search for the count of targets (specifically for "Name:"), as shown in Figure 4.
  - Path: *C:\Users\inorw\Desktop\Tools\Kape\Targets\Compound\!SANS_Triage.tkape*
  - Command: *(Select-String -Path 'C:\Users\inorw\Desktop\Tools\KAPE\Targets\Compound\!SANS_Triage.tkape' -Pattern 'Name:' -AllMatches).Matches.count*
- **Answer:**
  In total, there are 21 target collections as shown in Figure 4 however, there are currently **18 active target collections** in !SANS_Triage and **3 inactive target collections**, as shown in Figure 3.

- **Supporting Evidence:**



```
        Category: FileKnowledge
        Path: Thumbcache.tkape
    -
        Name: WBEM
        Category: WBEM
        Path: WBEM.tkape
    -
        Name: WebBrowsers
        Category: Communications
        Path: WebBrowsers.tkape
    -
        Name: WindowsIndexSearch
        Category: FileKnowledge
        Path: WindowsIndexSearch.tkape
    -
        Name: WindowsTimeline
        Category: EvidenceOfExecution
        Path: WindowsTimeline.tkape

# Some additional artifacts would be candidates for inclusion in this triage target but
return large amounts of data, they have been omitted.
# Additional targets include (make sure to get spacing correct if un-commenting):
#    -
#        Name: Recycle Bin Data Files
#        Category: FileDeletion
#        Path: RecycleBin_DataFiles.tkape
#    -
#        Name: Outlook PST and OST Files
#        Category: Communications
#        Path: OutlookPSTOST.tkape
#    -
#        Name: Cloud Storage User Files
#        Category: Apps
#        Path: CloudStorage_All.tkape
```

Figure 3. Accessed the !SANS_Triage.tkape file through Notepad



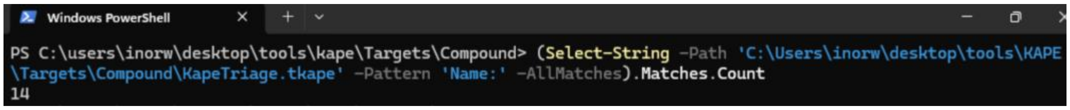
```
PS C:\users\inorw\desktop\tools\kape\Targets\Compound> (Select-String -Path 'C:\Users\inorw\desktop\tools\KAPE
\Targets\Compound\!SANS_Triage.tkape' -Pattern 'Name:' -AllMatches).Matches.Count
21
```

Figure 4. Command to retrieve the amount of target collections in !SANS_Triage.tkape

4. How many target collections are in KapeTriage?

- **Analysis Performed:**
  - The KapeTriage.tkape file was analyzed through the Powershell application.
  - The examiner created a quick command to search for the count of targets (specifically for “Name:”).
    - Command: *(Select-String -Path ‘C:\Users\inorw\Desktop\Tools\KAPE\Targets\Compound\KapeTriage.tkape’ -Pattern ‘Name:’ -AllMatches).Matches.count*
- **Answer:**
  There are currently **14 active target collections in KapeTriage.tkape**, as shown in Figure 5.

- **Supporting Evidence:**



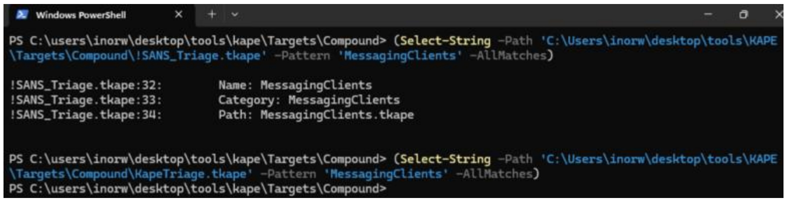
```
PS C:\users\inorw\desktop\tools\kape\Targets\Compound> (Select-String -Path 'C:\Users\inorw\desktop\tools\KAPE
\Targets\Compound\KapeTriage.tkape' -Pattern 'Name:' -AllMatches).Matches.Count
14
```

Figure 5. Command to retrieve the amount of target collections in KapeTriage.tkape

5. The !SANS_Triage collects information from messaging clients, does KapeTriage?

- **Analysis Performed:**
  - The KapeTriage.tkape and !SANS_Triage file was analyzed through the Powershell application.
  - The examiner used the following commands to search for the string, “MessagingClients”, to find out if that target collection is enabled for Kape Triage.
    - !SANS_Triage Command: *(Select-String -Path ‘C:\Users\inorw\Desktop\Tools\KAPE\Targets\Compound\!SANS_Triage.tkape’ -Pattern ‘MessagingClients’ -AllMatches)*
    - KapeTriage Command: *(Select-String -Path ‘C:\Users\inorw\Desktop\Tools\KAPE\Targets\Compound\KapeTriage.tkape’ -Pattern ‘MessagingClients’ -AllMatches)*
- **Answer:**
  As shown in Figure 6, **Kape Triage does not collect information from messaging clients** the way that !SANS_Triage does.

- **Supporting Evidence:**



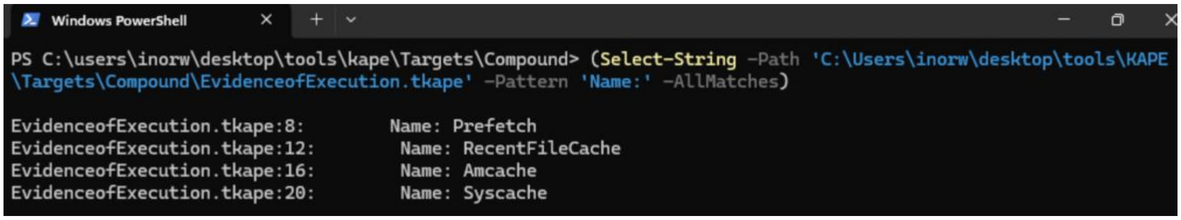
```
PS C:\users\inorw\desktop\tools\kape\Targets\Compound> (Select-String -Path 'C:\Users\inorw\desktop\tools\KAPE
\Targets\Compound\!SANS_Triage.tkape' -Pattern 'MessagingClients' -AllMatches)

!SANS_Triage.tkape:32:        Name: MessagingClients
!SANS_Triage.tkape:33:        Category: MessagingClients
!SANS_Triage.tkape:34:        Path: MessagingClients.tkape


PS C:\users\inorw\desktop\tools\kape\Targets\Compound> (Select-String -Path 'C:\Users\inorw\desktop\tools\KAPE
\Targets\Compound\KapeTriage.tkape' -Pattern 'MessagingClients' -AllMatches)
PS C:\users\inorw\desktop\tools\kape\Targets\Compound>
```

Figure 6. Command to figure out if KapeTriage collects information from messaging clients the way !SANS_Triage does

6. Find the EvidenceofExecution collection. What targets are in this collection?

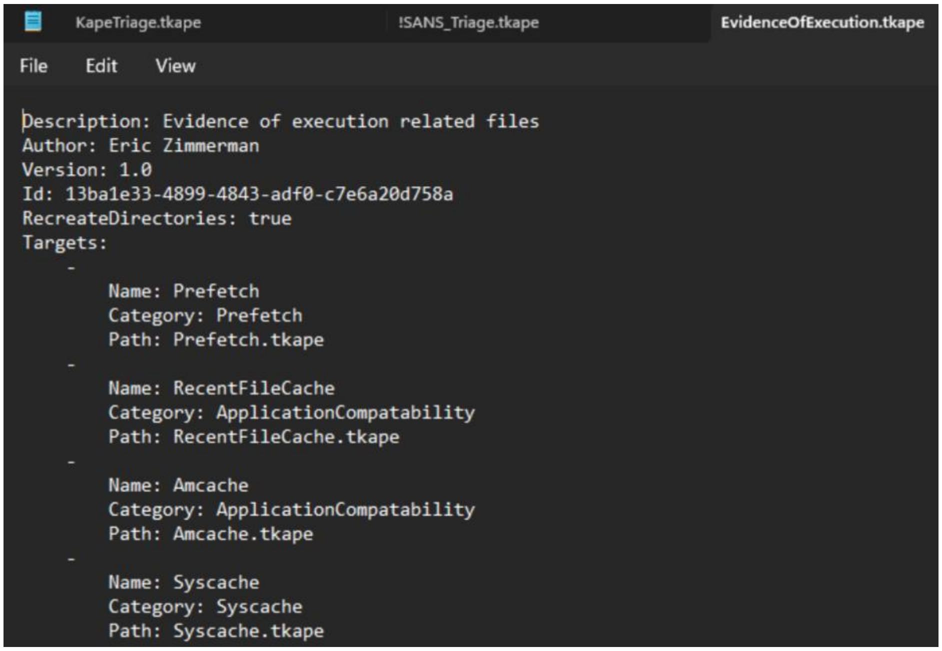
- **Analysis Performed:**
  - The EvidenceOfExecution.tkape file was analyzed through the Powershell and Notepad application.
  - The examiner used the following command to search for the string, "Name:", to find out if that target is enabled for the EvidenceOfExecution collection.
    - Command: *(Select-String -Path 'C:\Users\inorw\Desktop\Tools\KAPE\Targets\Compound\EvidenceofExecution.tkape' -Pattern 'Name:' -AllMatches)*
  - As shown in Figure 7 and 8, the following targets in the collection are: Prefetch, RecentFileCache, Amcache, and Syscache.
  - Path: *C:\Users\inorw\Desktop\Tools\KAPE\Targets\Compound\EvidenceofExecution.tkape*
- **Answer:**
  As shown in Figure 7 and 8, the **Prefetch, RecentFileCache, Amcache, Syscache targets** are in the EvidenceOfExecution collection.

- **Supporting Evidence:**



```
PS C:\users\inorw\desktop\tools\kape\Targets\Compound> (Select-String -Path 'C:\Users\inorw\desktop\tools\KAPE
\Targets\Compound\EvidenceofExecution.tkape' -Pattern 'Name:' -AllMatches)

EvidenceofExecution.tkape:8:        Name: Prefetch
EvidenceofExecution.tkape:12:        Name: RecentFileCache
EvidenceofExecution.tkape:16:        Name: Amcache
EvidenceofExecution.tkape:20:        Name: Syscache
```

Figure 7. Command to find what the targets are in the EvidenceOfExecution collection



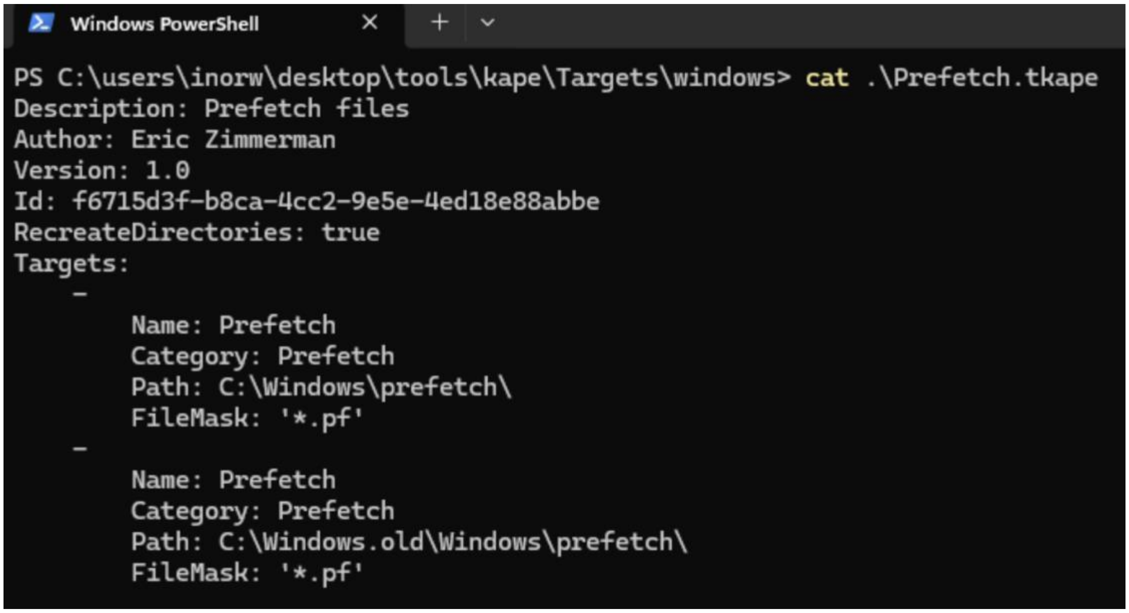
```
Description: Evidence of execution related files
Author: Eric Zimmerman
Version: 1.0
Id: 13ba1e33-4899-4843-adf0-c7e6a20d758a
RecreateDirectories: true
Targets:
    -
        Name: Prefetch
        Category: Prefetch
        Path: Prefetch.tkape
    -
        Name: RecentFileCache
        Category: ApplicationCompatability
        Path: RecentFileCache.tkape
    -
        Name: Amcache
        Category: ApplicationCompatability
        Path: Amcache.tkape
    -
        Name: Syscache
        Category: Syscache
        Path: Syscache.tkape
```

Figure 8. Accessed EvidenceOfExecution.tkape within Notepad

7. Find the Prefetch collection; there are two targets in this collection. What are the target paths this is collecting from?

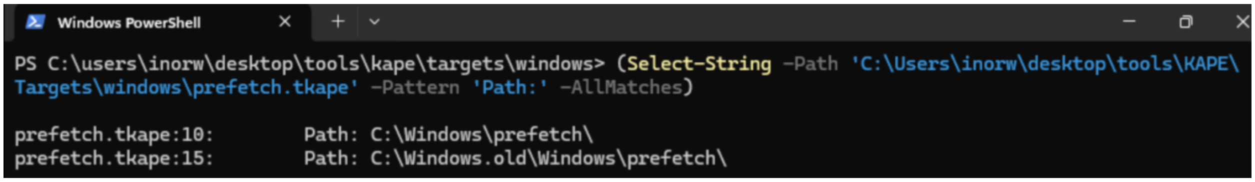
- **Analysis Performed:**
    - The Prefetch.tkape file was analyzed through the Powershell application.
    - The examiner used the following command to search for the string, “Path:”, to find out the target paths for the EvidenceOfExecution collection, as shown in Figure 10.
        - Command: *(Select-String -Path ‘C:\Users\inorw\Desktop\Tools\KAPE\Targets\Windows\Prefetch.tkape’ -Pattern ‘Path:’ -AllMatches)*
    - The examiner accessed the contents of the Prefetch.tkape file via the cat command (*cat .\Prefetch.tkape*).
        - As shown in Figure 9, there are two targets in the Prefetch collection with their corresponding target paths listed.
    - Path: *C:\Users\inorw\Desktop\Tools\KAPE\Targets\Windows\Prefetch.tkape*
- **Answer:**
  As shown in Figure 7, the target paths of the two targets in the Prefetch collection are: **C:\Windows\prefetch\** AND **C:\Windows.old\Windows\prefetch\.** The reason for why the target path matters is because it’s where KAPE actually looks for artifacts.

- **Supporting Evidence:**



```
PS C:\users\inorw\desktop\tools\kape\Targets\windows> cat .\Prefetch.tkape
Description: Prefetch files
Author: Eric Zimmerman
Version: 1.0
Id: f6715d3f-b8ca-4cc2-9e5e-4ed18e88abbe
RecreateDirectories: true
Targets:
    -
        Name: Prefetch
        Category: Prefetch
        Path: C:\Windows\prefetch\
        FileMask: '*.pf'
    -
        Name: Prefetch
        Category: Prefetch
        Path: C:\Windows.old\Windows\prefetch\
        FileMask: '*.pf'
```

Figure 9. Reading the contents of Prefetch.tkape



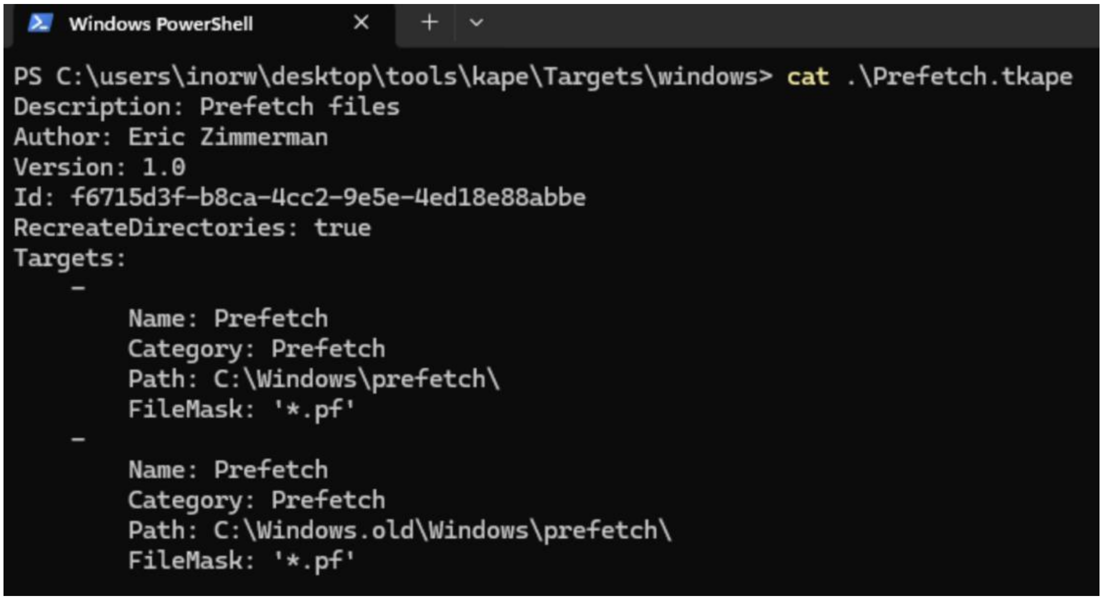
```
PS C:\users\inorw\desktop\tools\kape\targets\windows> (Select-String -Path 'C:\Users\inorw\desktop\tools\KAPE\
Targets\windows\prefetch.tkape' -Pattern 'Path:' -AllMatches)

prefetch.tkape:10:        Path: C:\Windows\prefetch\
prefetch.tkape:15:        Path: C:\Windows.old\Windows\prefetch\
```

Figure 10. Command to find the target paths of Prefetch.tkape file

8. What is the file mask for this collection?

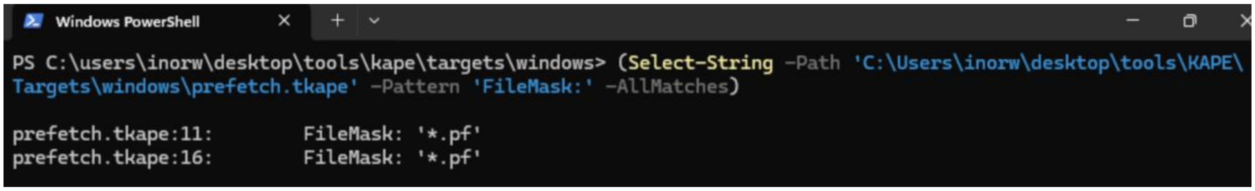
- **Analysis Performed:**
    - The Prefetch.tkape file was analyzed through the Powershell application.
    - The examiner used the following command to search for the string, “FileMask:”, to find out the file mask for the EvidenceOfExecution collection, as shown in Figure 12.
        - Command: *(Select-String -Path ‘C:\Users\inorw\Desktop\Tools\KAPE\Targets\Windows\Prefetch.tkape’ -Pattern ‘FileMask:’ -AllMatches)*
    - The examiner accessed the contents of the Prefetch.tkape file via the cat command (*cat .\Prefetch.tkape*).
        - As shown in Figure 11, there are two targets in the Prefetch collection with their FileMask listed.
    - Path: *C:\Users\inorw\Desktop\Tools\KAPE\Targets\Windows\Prefetch.tkape*
- **Answer:**
  As shown in Figure 11 and 12, the file mask for the Prefetch collection is **‘*.pf’**. The reason for why this may matter is because the file mask tells KAPE which files to collect (and which to ignored) at a given path. It’s essentially a filename filter. In this case, it is only collecting Windows Prefetch files since the file mask is ‘*.pf’.

- **Supporting Evidence:**



```
PS C:\users\inorw\desktop\tools\kape\Targets\windows> cat .\Prefetch.tkape
Description: Prefetch files
Author: Eric Zimmerman
Version: 1.0
Id: f6715d3f-b8ca-4cc2-9e5e-4ed18e88abbe
RecreateDirectories: true
Targets:
    -
        Name: Prefetch
        Category: Prefetch
        Path: C:\Windows\prefetch\
        FileMask: '*.pf'
    -
        Name: Prefetch
        Category: Prefetch
        Path: C:\Windows.old\Windows\prefetch\
        FileMask: '*.pf'
```

Figure 11. Reading the contents of Prefetch.tkape file



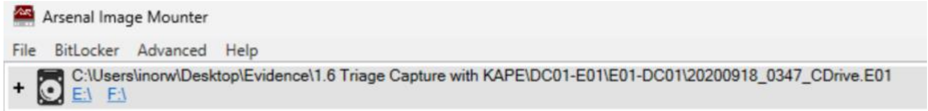
```
PS C:\users\inorw\desktop\tools\kape\targets\windows> (Select-String -Path 'C:\Users\inorw\desktop\tools\KAPE\
Targets\windows\prefetch.tkape' -Pattern 'FileMask:' -AllMatches)

prefetch.tkape:11:        FileMask: '*.pf'
prefetch.tkape:16:        FileMask: '*.pf'
```

Figure 12. Command to find the file mask within the Prefetch.tkape file

9. Before you execute the triage collection, ensure all is configured properly. What is the command line output for the DC01 collection?

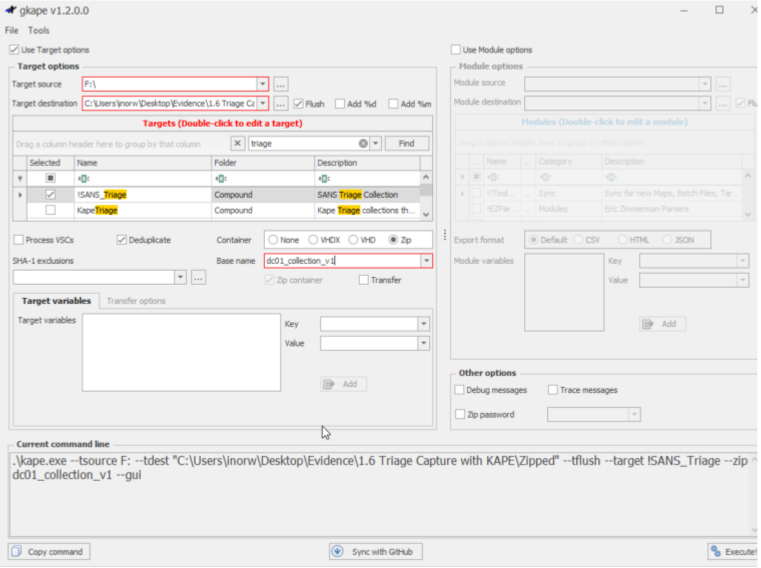
- **Analysis Performed:**
  - The examiner mounted the DC01 CDrive image using the Arsenal Image Mounter application as shown in Figure 13. The drive containing the CDrive is F:\.
  - The examiner ensured all is configured properly before executing the triage collection as shown in Figure 14.
    - Opened gkape v1.2.0.0 and confirmed Target options with the Target source to the mounted DC image volume of F: and the Target destination to “C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped”.
    - Selected target collection: !SANS_Triage
    - Ensured Process VSCs were unchecked; Deduplicate checked; container = zip
  - The examiner then executed the command and the command line output is shown in Figure 15
- **Answer:**
  The command line output for the DC01 collection is shown in Figure 15. The command is “.\kape.exe --tsource F: --tdest “C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped” --tflush --target !SANS_Triage --zip dc01_collection_v1 --gui”, as shown in Figure 14.

- **Supporting Evidence:**



C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\DC01-E01\E01-DC01\20200918_0347_CDrive.E01
E:\ F:\

Figure 13. Successfully mounted the DC01 CDrive Image


gkape v1.2.0.0
File
Tools
Targets (Double-click to edit a target)
Drag a column header here to group by that column
Find
Transfer options
Add
Copy command
Sync with GitHub
Execute!

☑ Use Target options

Target options

Target source: F:\

Target destination: C:\Users\inorw\Desktop\Evidence\1.6 Triage Ca ☑ Flush □ Add %d □ Add %m

Search: triage

Selected	Name	Folder	Description
☑	!SANS_Triage	Compound	SANS Triage Collection
□	KapeTriage	Compound	Kape Triage collections th...

□ Process VSCs ☑ Deduplicate Container: ○ None ○ VHDX ○ VHD ● Zip

SHA-1 exclusions

Base name: dc01_collection_v1

☑ Zip container □ Transfer

Target variables: Key, Value

□ Use Module options

Module options: Module source, Module destination

Export format: ● Default ○ CSV ○ HTML ○ JSON

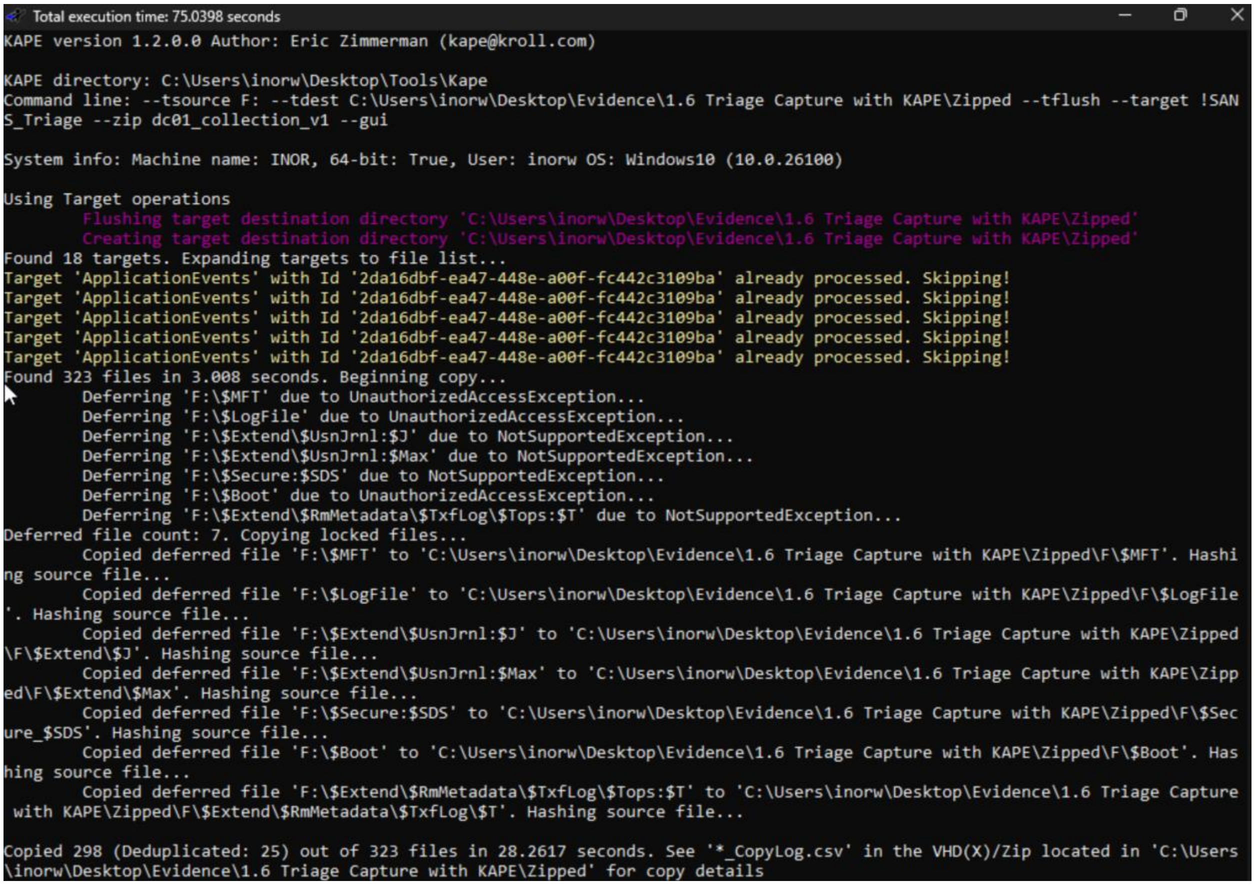
Module variables: Key, Value

Other options: □ Debug messages □ Trace messages □ Zip password

Current command line

```
.\kape.exe --tsource F: --tdest "C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped" --tflush --target !SANS_Triage --zip
dc01_collection_v1 --gui
```

Figure 14. gkape with all correct configurations for triage collection



```
Total execution time: 75.0398 seconds
KAPE version 1.2.0.0 Author: Eric Zimmerman (kape@kroll.com)

KAPE directory: C:\Users\inorw\Desktop\Tools\Kape
Command line: --tsource F: --tdest C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped --tflush --target !SAN
S_Triage --zip dc01_collection_v1 --gui

System info: Machine name: INOR, 64-bit: True, User: inorw OS: Windows10 (10.0.26100)

Using Target operations
        Flushing target destination directory 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped'
        Creating target destination directory 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped'
Found 18 targets. Expanding targets to file list...
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Found 323 files in 3.008 seconds. Beginning copy...
        Deferring 'F:\$MFT' due to UnauthorizedAccessException...
        Deferring 'F:\$LogFile' due to UnauthorizedAccessException...
        Deferring 'F:\$Extend\$UsnJrnl:$J' due to NotSupportedException...
        Deferring 'F:\$Extend\$UsnJrnl:$Max' due to NotSupportedException...
        Deferring 'F:\$Secure:$SDS' due to NotSupportedException...
        Deferring 'F:\$Boot' due to UnauthorizedAccessException...
        Deferring 'F:\$Extend\$RmMetadata\$TxfLog\$Tops:$T' due to NotSupportedException...
Deferred file count: 7. Copying locked files...
        Copied deferred file 'F:\$MFT' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\F\$MFT'. Hashi
ng source file...
        Copied deferred file 'F:\$LogFile' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\F\$LogFile
'. Hashing source file...
        Copied deferred file 'F:\$Extend\$UsnJrnl:$J' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped
\F\$Extend\$J'. Hashing source file...
        Copied deferred file 'F:\$Extend\$UsnJrnl:$Max' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipp
ed\F\$Extend\$Max'. Hashing source file...
        Copied deferred file 'F:\$Secure:$SDS' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\F\$Sec
ure_$SDS'. Hashing source file...
        Copied deferred file 'F:\$Boot' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\F\$Boot'. Has
hing source file...
        Copied deferred file 'F:\$Extend\$RmMetadata\$TxfLog\$Tops:$T' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture
 with KAPE\Zipped\F\$Extend\$RmMetadata\$TxfLog\$T'. Hashing source file...

Copied 298 (Deduplicated: 25) out of 323 files in 28.2617 seconds. See '*_CopyLog.csv' in the VHD(X)/Zip located in 'C:\Users
\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped' for copy details
```

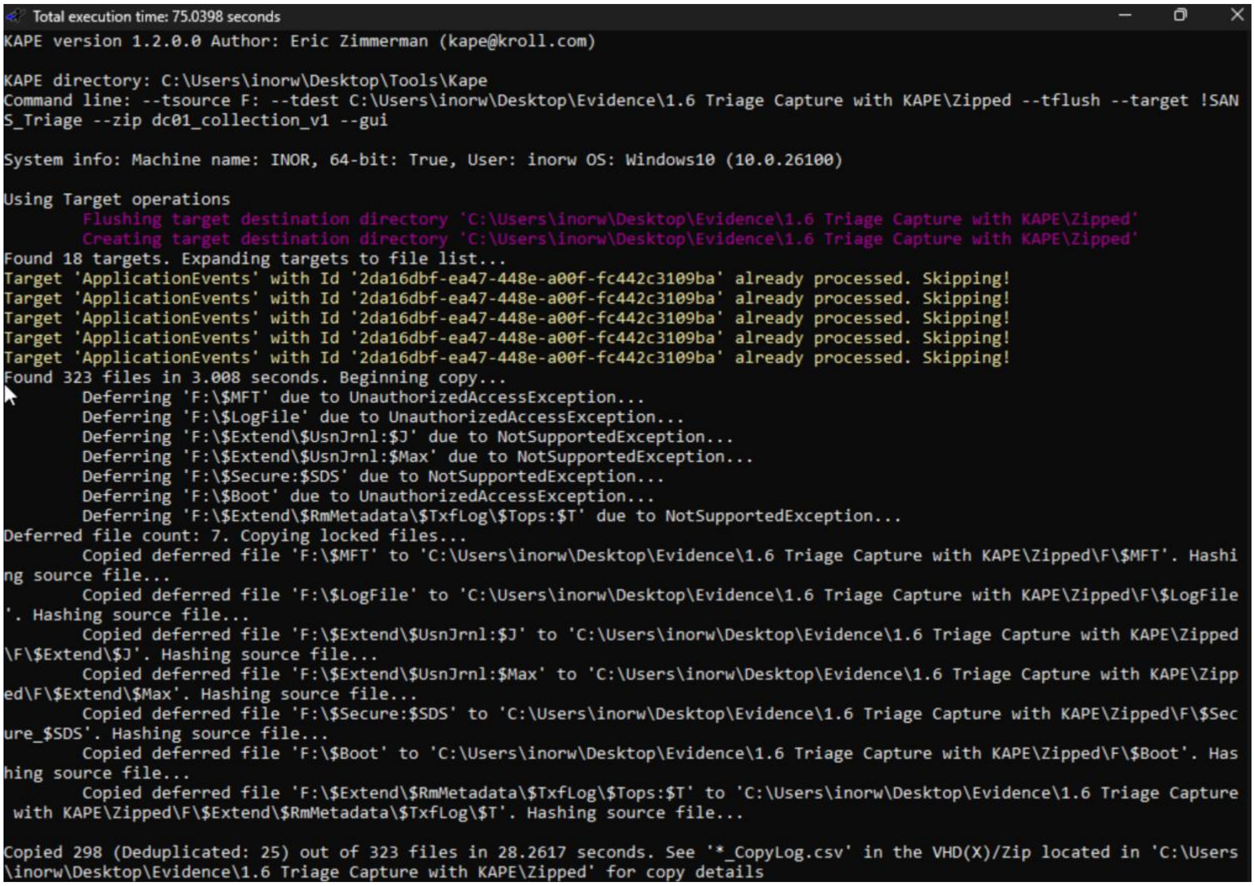
Figure 15. Command line output for the DC01 collection

KAPE collection analysis DC01

1. There should have been 18 targets found, how many files were found/collected?

- **Analysis Performed:**
    - The F:\ drive which was mounted from DC01 evidence's CDrive was analyzed through the gkape application.
    - As noted previously, the examiner properly configured the triage collection for DC01 which is shown in Figure 14.
    - As shown in Figure 16, there are 18 found targets, which is what should have been found. Additionally, there are 323 found files and the application copied 298 (deduplicated 25).
- **Answer:**
  As confirmed in the question, there are **18 found targets, 323 found files, and collected 298 files**, as shown in Figure 16. These counts mean KAPE matched 18 target definitions, located 323 candidate artifact files across those targets, and actually copied 298 of them to the collection after de-duplicating (skipping 25 duplicate/overlapping files).

- **Supporting Evidence:**



Total execution time: 75.0398 seconds

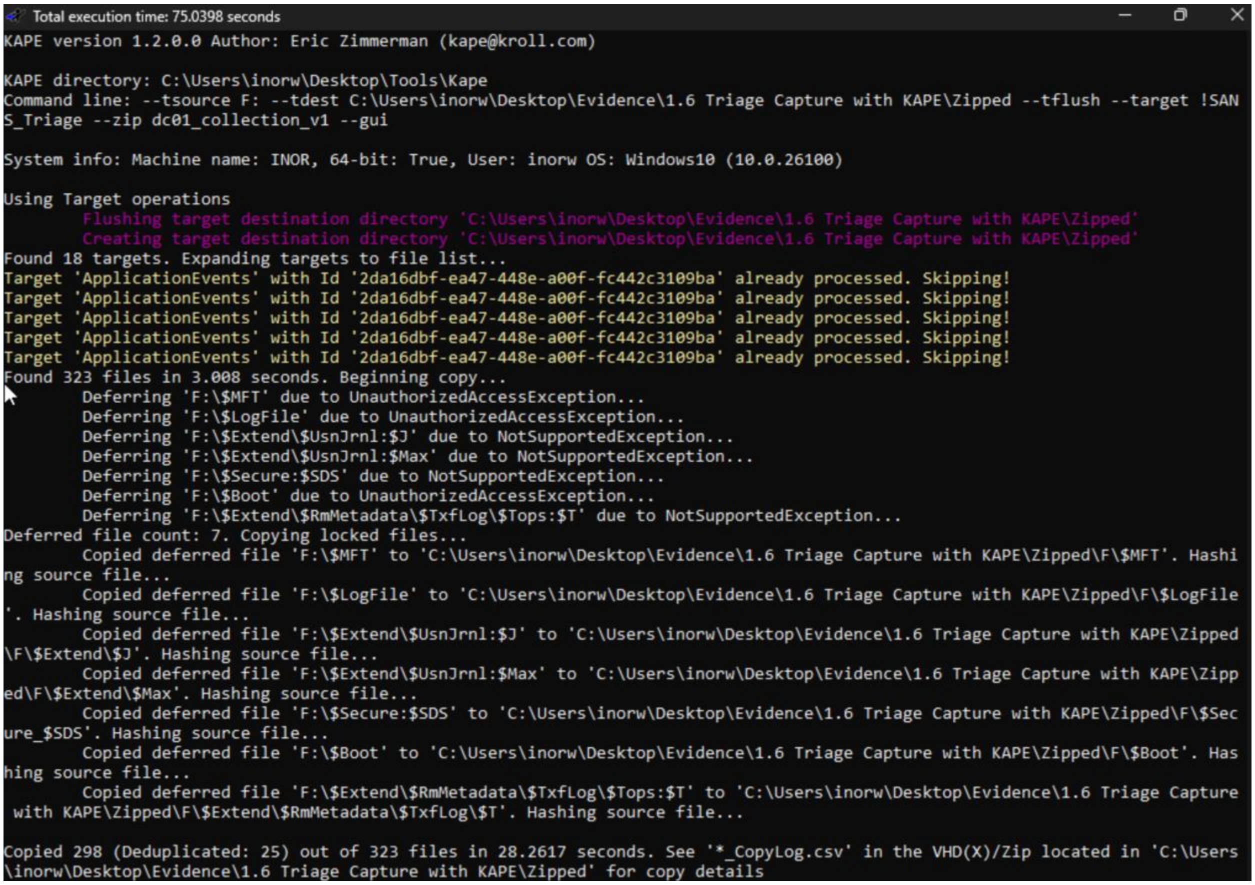
```
KAPE version 1.2.0.0 Author: Eric Zimmerman (kape@kroll.com)

KAPE directory: C:\Users\inorw\Desktop\Tools\Kape
Command line: --tsource F: --tdest C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped --tflush --target !SAN
S_Triage --zip dc01_collection_v1 --gui

System info: Machine name: INOR, 64-bit: True, User: inorw OS: Windows10 (10.0.26100)

Using Target operations
        Flushing target destination directory 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped'
        Creating target destination directory 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped'
Found 18 targets. Expanding targets to file list...
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Found 323 files in 3.008 seconds. Beginning copy...
        Deferring 'F:\$MFT' due to UnauthorizedAccessException...
        Deferring 'F:\$LogFile' due to UnauthorizedAccessException...
        Deferring 'F:\$Extend\$UsnJrnl:$J' due to NotSupportedException...
        Deferring 'F:\$Extend\$UsnJrnl:$Max' due to NotSupportedException...
        Deferring 'F:\$Secure:$SDS' due to NotSupportedException...
        Deferring 'F:\$Boot' due to UnauthorizedAccessException...
        Deferring 'F:\$Extend\$RmMetadata\$TxfLog\$Tops:$T' due to NotSupportedException...
Deferred file count: 7. Copying locked files...
        Copied deferred file 'F:\$MFT' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\F\$MFT'. Hashi
ng source file...
        Copied deferred file 'F:\$LogFile' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\F\$LogFile
'. Hashing source file...
        Copied deferred file 'F:\$Extend\$UsnJrnl:$J' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped
\F\$Extend\$J'. Hashing source file...
        Copied deferred file 'F:\$Extend\$UsnJrnl:$Max' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipp
ed\F\$Extend\$Max'. Hashing source file...
        Copied deferred file 'F:\$Secure:$SDS' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\F\$Sec
ure_$SDS'. Hashing source file...
        Copied deferred file 'F:\$Boot' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\F\$Boot'. Has
hing source file...
        Copied deferred file 'F:\$Extend\$RmMetadata\$TxfLog\$Tops:$T' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture
 with KAPE\Zipped\F\$Extend\$RmMetadata\$TxfLog\$T'. Hashing source file...

Copied 298 (Deduplicated: 25) out of 323 files in 28.2617 seconds. See '*_CopyLog.csv' in the VHD(X)/Zip located in 'C:\Users
\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped' for copy details
```

Figure 16. Command line output for DC01 collection

2. How many files were "deferred?"

- **Analysis Performed:**
  - The F:\ drive which was mounted from DC01 evidence's CDrive was analyzed through the gkape application.
  - As noted previously, the examiner properly configured the triage collection for DC01 which is shown in Figure 14.
  - As shown in Figure 17, there are 7 deferred files all due to UnauthorizedAccessException.
- **Answer:**
  There are **7 deferred files** all due to UnauthorizedAccessException, as shown in Figure 17. Deferred files are artifacts KAPE found but couldn't copy during the run, typically because the path was locked or permission restricted. In this case, access was denied.

- **Supporting Evidence:**



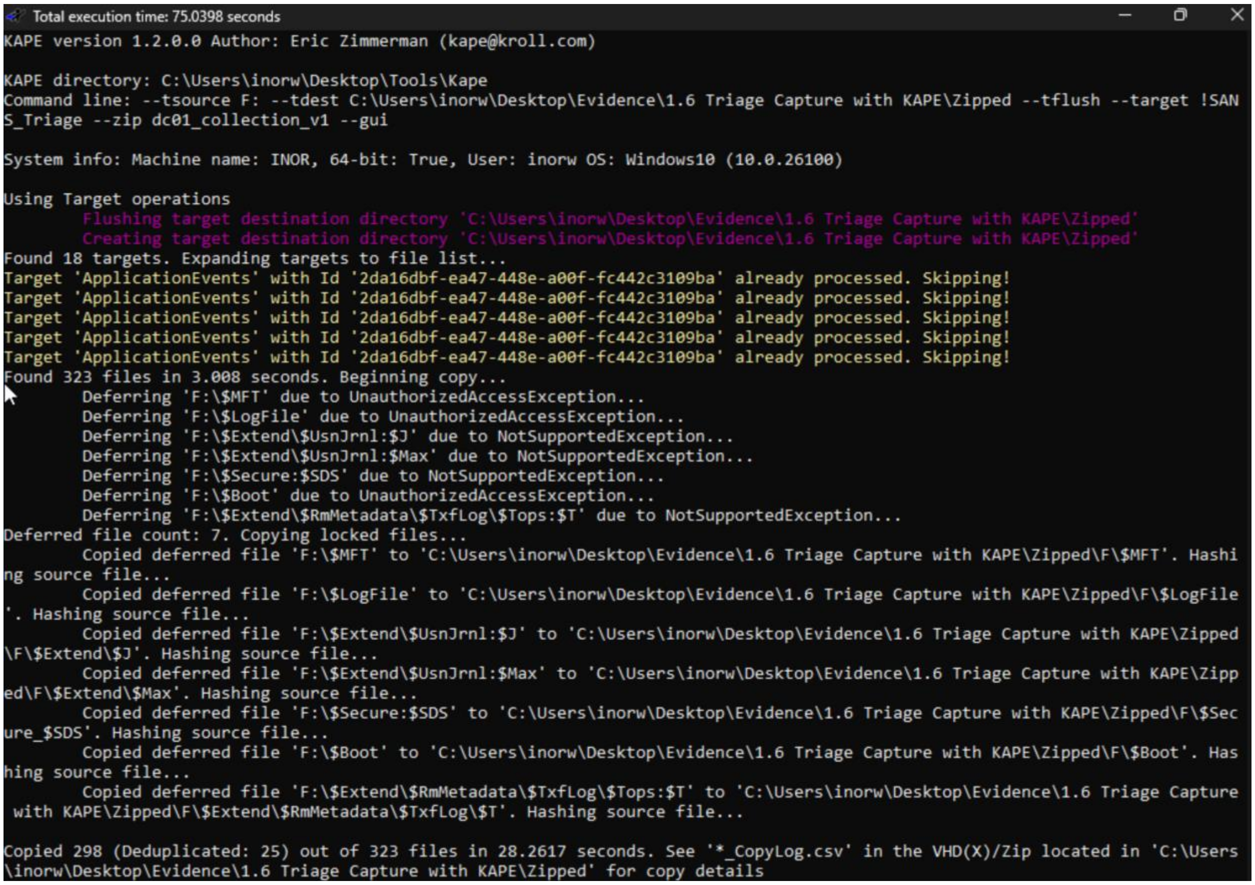
```
Total execution time: 75.0398 seconds
KAPE version 1.2.0.0 Author: Eric Zimmerman (kape@kroll.com)

KAPE directory: C:\Users\inorw\Desktop\Tools\Kape
Command line: --tsource F: --tdest C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped --tflush --target !SAN
S_Triage --zip dc01_collection_v1 --gui

System info: Machine name: INOR, 64-bit: True, User: inorw OS: Windows10 (10.0.26100)

Using Target operations
        Flushing target destination directory 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped'
        Creating target destination directory 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped'
Found 18 targets. Expanding targets to file list...
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Found 323 files in 3.008 seconds. Beginning copy...
        Deferring 'F:\$MFT' due to UnauthorizedAccessException...
        Deferring 'F:\$LogFile' due to UnauthorizedAccessException...
        Deferring 'F:\$Extend\$UsnJrnl:$J' due to NotSupportedException...
        Deferring 'F:\$Extend\$UsnJrnl:$Max' due to NotSupportedException...
        Deferring 'F:\$Secure:$SDS' due to NotSupportedException...
        Deferring 'F:\$Boot' due to UnauthorizedAccessException...
        Deferring 'F:\$Extend\$RmMetadata\$TxfLog\$Tops:$T' due to NotSupportedException...
Deferred file count: 7. Copying locked files...
        Copied deferred file 'F:\$MFT' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\F\$MFT'. Hashi
ng source file...
        Copied deferred file 'F:\$LogFile' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\F\$LogFile
'. Hashing source file...
        Copied deferred file 'F:\$Extend\$UsnJrnl:$J' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped
\F\$Extend\$J'. Hashing source file...
        Copied deferred file 'F:\$Extend\$UsnJrnl:$Max' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipp
ed\F\$Extend\$Max'. Hashing source file...
        Copied deferred file 'F:\$Secure:$SDS' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\F\$Sec
ure_$SDS'. Hashing source file...
        Copied deferred file 'F:\$Boot' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\F\$Boot'. Has
hing source file...
        Copied deferred file 'F:\$Extend\$RmMetadata\$TxfLog\$Tops:$T' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture
 with KAPE\Zipped\F\$Extend\$RmMetadata\$TxfLog\$T'. Hashing source file...

Copied 298 (Deduplicated: 25) out of 323 files in 28.2617 seconds. See '*_CopyLog.csv' in the VHD(X)/Zip located in 'C:\Users
\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped' for copy details
```

Figure 17. Command line output for DC01 collection

3. Of the files copied, how many were duplicated?

- **Analysis Performed:**
  - The F:\ drive which was mounted from DC01 evidence's CDrive was analyzed through the gkape application.
  - As noted previously, the examiner properly configured the triage collection for DC01 which is shown in Figure 14.
  - As shown in Figure 18, of the files copied, there were 25 duplicated files. Copied 298 (with 25 duplicated files) out of 323 files in 28.2617 seconds.
- **Answer:**
  Of the files copied, there were **25 duplicated files**, as shown in Figure 18.

- **Supporting Evidence:**



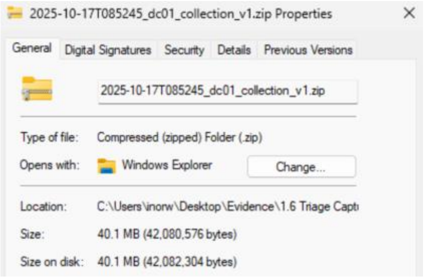
```
Total execution time: 75.0398 seconds
KAPE version 1.2.0.0 Author: Eric Zimmerman (kape@kroll.com)

KAPE directory: C:\Users\inorw\Desktop\Tools\Kape
Command line: --tsource F: --tdest C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped --tflush --target !SAN
S_Triage --zip dc01_collection_v1 --gui

System info: Machine name: INOR, 64-bit: True, User: inorw OS: Windows10 (10.0.26100)

Using Target operations
        Flushing target destination directory 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped'
        Creating target destination directory 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped'
Found 18 targets. Expanding targets to file list...
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Found 323 files in 3.008 seconds. Beginning copy...
        Deferring 'F:\$MFT' due to UnauthorizedAccessException...
        Deferring 'F:\$LogFile' due to UnauthorizedAccessException...
        Deferring 'F:\$Extend\$UsnJrnl:$J' due to NotSupportedException...
        Deferring 'F:\$Extend\$UsnJrnl:$Max' due to NotSupportedException...
        Deferring 'F:\$Secure:$SDS' due to NotSupportedException...
        Deferring 'F:\$Boot' due to UnauthorizedAccessException...
        Deferring 'F:\$Extend\$RmMetadata\$TxfLog\$Tops:$T' due to NotSupportedException...
Deferred file count: 7. Copying locked files...
        Copied deferred file 'F:\$MFT' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\F\$MFT'. Hashi
ng source file...
        Copied deferred file 'F:\$LogFile' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\F\$LogFile
'. Hashing source file...
        Copied deferred file 'F:\$Extend\$UsnJrnl:$J' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped
\F\$Extend\$J'. Hashing source file...
        Copied deferred file 'F:\$Extend\$UsnJrnl:$Max' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipp
ed\F\$Extend\$Max'. Hashing source file...
        Copied deferred file 'F:\$Secure:$SDS' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\F\$Sec
ure_$SDS'. Hashing source file...
        Copied deferred file 'F:\$Boot' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\F\$Boot'. Has
hing source file...
        Copied deferred file 'F:\$Extend\$RmMetadata\$TxfLog\$Tops:$T' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture
 with KAPE\Zipped\F\$Extend\$RmMetadata\$TxfLog\$T'. Hashing source file...

Copied 298 (Deduplicated: 25) out of 323 files in 28.2617 seconds. See '*_CopyLog.csv' in the VHD(X)/Zip located in 'C:\Users
\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped' for copy details
```

Figure 18. Command line output for DC01 collection

4. What is the size (in MB) of the KAPE archive (ZIP) file for the domain controller image?

- **Analysis Performed:**
    - The F:\ drive which was mounted from DC01 evidence's CDrive was analyzed through the gkape application.
    - After the command ran successfully, a zip file was created in the destination folder set by the examiner.
        - Path: *C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\2025-10-*17T085245_dc01*_collection_v1.zip*
    - The examiner opened the properties of the KAPE archive (ZIP) file for the domain controller (DC01) image as shown in Figure 19.
- **Answer:**
  The size (in MB) of the KAPE archive (ZIP) file for DC01 is **40.1 MB**, as shown in Figure 19.

- **Supporting Evidence:**



2025-10-17T085245_dc01_collection_v1.zip

Type of file: Compressed (zipped) Folder (.zip)

Opens with: Windows Explorer

Location: C:\Users\inorw\Desktop\Evidence\1.6 Triage Capt

Size: 40.1 MB (42,080,576 bytes)

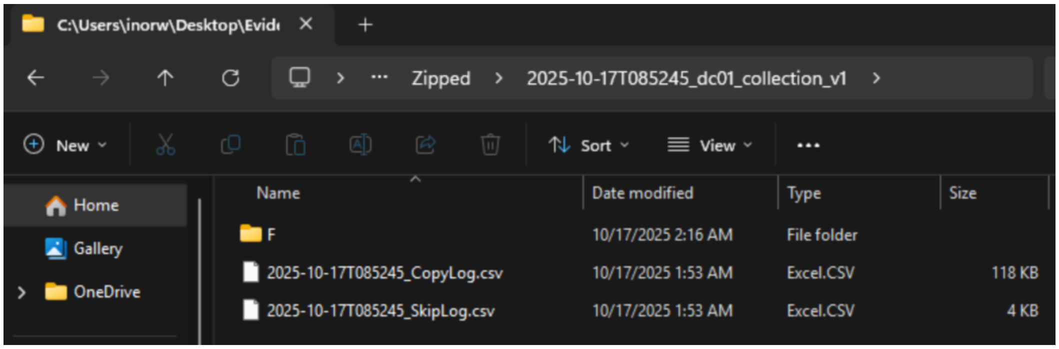
Size on disk: 40.1 MB (42,082,304 bytes)

Figure 19. Properties tab of the output DC01 collection zip file

5. Extract the DC01 collection. What is the name of the root folder in this collection? Why is it named this way? Why not "C:\"

- **Analysis Performed:**
    - The F:\ drive which was mounted from DC01 evidence's CDrive was analyzed through the gkape application.
    - After the command ran successfully, a zip file was created in the destination folder set by the examiner.
        - Path: *C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\2025-10-*17T085245_dc01*_collection_v1.zip*
        - The examiner extracted the DC01 collection and noticed that the name of the root folder in the collection is "F:\" and not "C:\", as shown in Figure 20.
    - The image file was initially mounted through Arsenal Image Mounter and the application mounted the image's root folder to F:\.
- **Answer:**
  The name of the root folder in the DC01 collection is "F:\", as shown in Figure 20. The reason for why it is named "F:\" and not "C:\" is because the Target source is the mounted (from Arsenal Image Mounter) DC01 image file which is "F:\".

- **Supporting Evidence:**


C:\Users\inorw\Desktop\Evid
Zipped
2025-10-17T085245_dc01_collection_v1
New
Sort
View
Home
Gallery
OneDrive

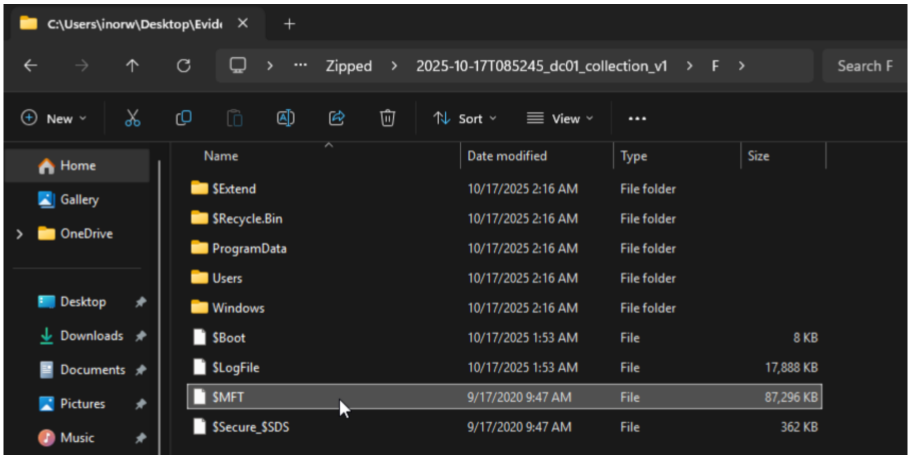
Name	Date modified	Type	Size
F	10/17/2025 2:16 AM	File folder	
2025-10-17T085245_CopyLog.csv	10/17/2025 1:53 AM	Excel.CSV	118 KB
2025-10-17T085245_SkipLog.csv	10/17/2025 1:53 AM	Excel.CSV	4 KB

Figure 20. Shown filesystem of the DC01 Collection with the F drive folder

6. Is the MFT file in this collection? What is the file's name?

- **Analysis Performed:**
  - The F:\ drive which was mounted from DC01 evidence's CDrive was analyzed through the gkape application.
  - After the command ran successfully, a zip file was created in the destination folder set by the examiner.
    - Path: *C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\2025-10-*17T085245_dc01_*collection_v1.zip*
    - The examiner extracted the DC01 collection, and opened the root folder which is F.
    - Within the F folder, which is the CDrive of the image, it contained the MFT file, as shown in Figure 21.
- **Answer:**
  Yes, the MFT file is in this collection and the file's name is $MFT, as shown in Figure 21

- **Supporting Evidence:**



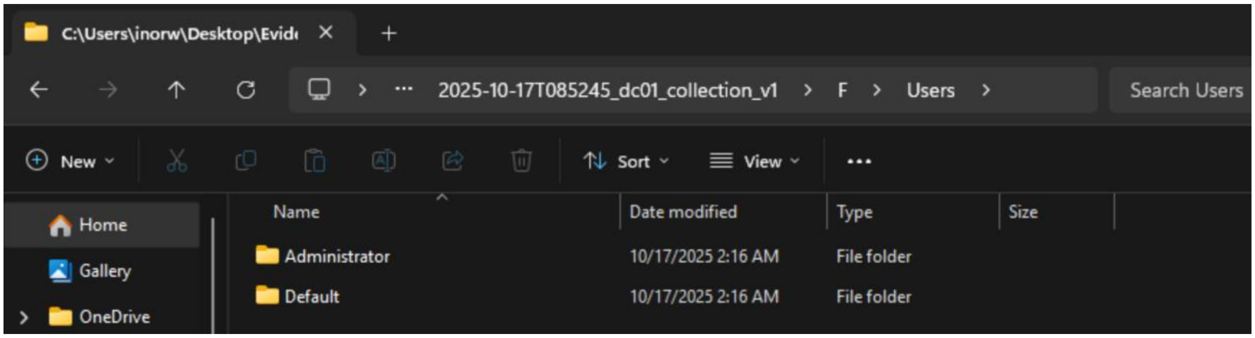
Name	Date modified	Type	Size
$Extend	10/17/2025 2:16 AM	File folder	
$Recycle.Bin	10/17/2025 2:16 AM	File folder	
ProgramData	10/17/2025 2:16 AM	File folder	
Users	10/17/2025 2:16 AM	File folder	
Windows	10/17/2025 2:16 AM	File folder	
$Boot	10/17/2025 1:53 AM	File	8 KB
$LogFile	10/17/2025 1:53 AM	File	17,888 KB
$MFT	9/17/2020 9:47 AM	File	87,296 KB
$Secure_$SDS	9/17/2020 9:47 AM	File	362 KB

Figure 21. Shown filesystem of the DC01 CDrive

7. What user profiles were captured in this collection? (Do not include default).

- **Analysis Performed:**
  - Mounted the DC01 image (via Arsenal Image Mounter) and extracted the !SANS_Triage ZIP.
  - Browsed the extracted triage to \Users\ for the mounted system volume (shown as F: in this collection).
  - Reviewed the profile folders present, excluding default.
- **Answer:**
  The **Administrator** user profile was captured in this collection.

- **Supporting Evidence:**


C:\Users\inorw\Desktop\Evid
2025-10-17T085245_dc01_collection_v1 › F › Users ›
Search Users
New
Sort
View
Home
Gallery
OneDrive

Name	Date modified	Type	Size
Administrator	10/17/2025 2:16 AM	File folder	
Default	10/17/2025 2:16 AM	File folder	

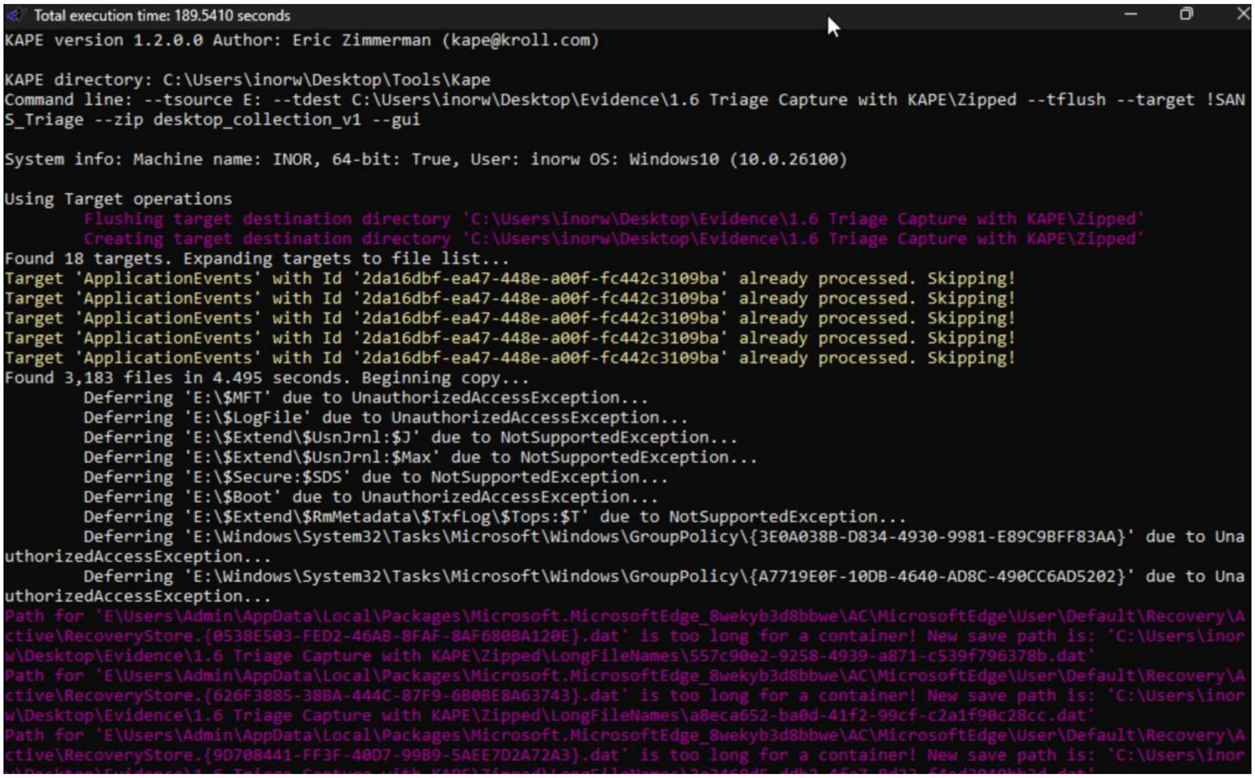
Figure 22. Extracted DC01 triage → F:\Users\ showing Administrator and Default; only Administrator counts as a user profile for this question (Default excluded).

KAPE collection analysis DESKTOP

1. There should have been 18 targets found, how many files were found/collected?

- **Analysis Performed:**
  - Mounted the DESKTOP image (via Arsenal Image Mounter) and the set the mounted Windows volume as KAPE source.
  - Ran !SANS_Triage to a ZIP destination and monitored the console output, as shown in Figure 23.
- **Answer:**
  As confirmed in the question, there are **18 found targets, 3183 found files**, as shown in Figure 23. These counts mean KAPE matched 18 target definitions, located 3183 candidate artifact files across those targets.

- **Supporting Evidence:**



Total execution time: 189.5410 seconds

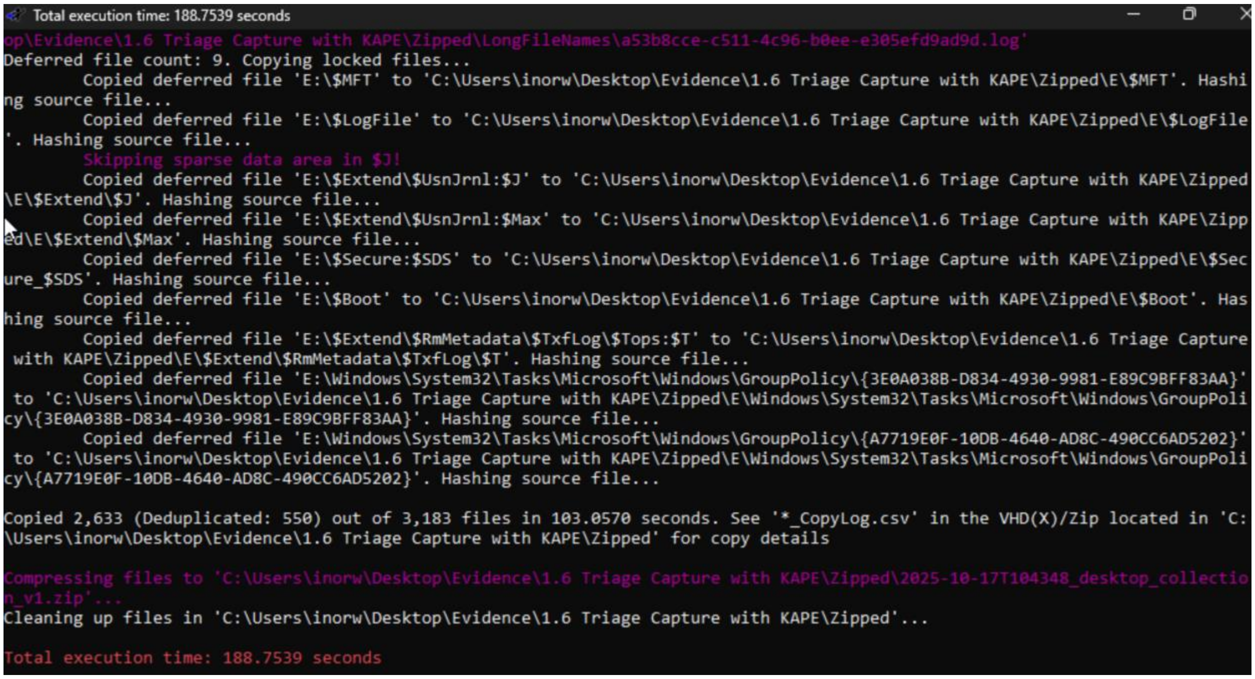
```
KAPE version 1.2.0.0 Author: Eric Zimmerman (kape@kroll.com)

KAPE directory: C:\Users\inorw\Desktop\Tools\Kape
Command line: --tsource E: --tdest C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped --tflush --target !SAN
S_Triage --zip desktop_collection_v1 --gui

System info: Machine name: INOR, 64-bit: True, User: inorw OS: Windows10 (10.0.26100)

Using Target operations
        Flushing target destination directory 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped'
        Creating target destination directory 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped'
Found 18 targets. Expanding targets to file list...
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Target 'ApplicationEvents' with Id '2da16dbf-ea47-448e-a00f-fc442c3109ba' already processed. Skipping!
Found 3,183 files in 4.495 seconds. Beginning copy...
        Deferring 'E:\$MFT' due to UnauthorizedAccessException...
        Deferring 'E:\$LogFile' due to UnauthorizedAccessException...
        Deferring 'E:\$Extend\$UsnJrnl:$J' due to NotSupportedException...
        Deferring 'E:\$Extend\$UsnJrnl:$Max' due to NotSupportedException...
        Deferring 'E:\$Secure:$SDS' due to NotSupportedException...
        Deferring 'E:\$Boot' due to UnauthorizedAccessException...
        Deferring 'E:\$Extend\$RmMetadata\$TxfLog\$Tops:$T' due to NotSupportedException...
        Deferring 'E:\Windows\System32\Tasks\Microsoft\Windows\GroupPolicy\{3E0A038B-D834-4930-9981-E89C9BFF83AA}' due to Una
uthorizedAccessException...
        Deferring 'E:\Windows\System32\Tasks\Microsoft\Windows\GroupPolicy\{A7719E0F-10DB-4640-AD8C-490CC6AD5202}' due to Una
uthorizedAccessException...
Path for 'E\Users\Admin\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\MicrosoftEdge\User\Default\Recovery\A
ctive\RecoveryStore.{0538E503-FED2-46AB-8FAF-8AF680BA120E}.dat' is too long for a container! New save path is: 'C:\Users\inor
w\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\LongFileNames\557c90e2-9258-4939-a871-c539f796378b.dat'
Path for 'E\Users\Admin\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\MicrosoftEdge\User\Default\Recovery\A
ctive\RecoveryStore.{626F3885-38BA-444C-87F9-6B0BE8A63743}.dat' is too long for a container! New save path is: 'C:\Users\inor
w\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\LongFileNames\a8eca652-ba0d-41f2-99cf-c2a1f90c28cc.dat'
Path for 'E\Users\Admin\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\MicrosoftEdge\User\Default\Recovery\A
ctive\RecoveryStore.{9D708441-FF3F-40D7-99B9-5AEE7D2A72A3}.dat' is too long for a container! New save path is: 'C:\Users\inor
```

Figure 23. KAPE console for DESKTOP showing command line, 18 targets, 3,183 files found, and deferral messages (e.g., UnauthorizedAccessException, NotSupportedException, path too long).

2. How many files were “deferred?”

- **Analysis Performed:**
  - Mounted the DESKTOP image with Arsenal and set the mounted Windows volume (E:) as the KAPE source.
  - Ran !SANS_Triage to a ZIP destination (no VSCs), then reviewed the KAPE console for outcome lines.
  - Noted “Deferred file” entries and their reasons; counted 9 deferred files, primarily due to UnauthorizedAccessException (access denied) and NotSupportedException (unsupported object/path).
- **Answer:**
  There are **9 deferred files**, mostly due to UnauthorizedAccessException and NotSupportedException (as shown in Figure 23), as shown in Figure 24. Deferred files are artifacts KAPE found but couldn’t copy during the run, typically because the path was locked or permission restricted. In this case, access was denied to some artifacts and the operation isn’t supported for some of the artifact’s objects/paths.

- **Supporting Evidence:**



Total execution time: 188.7539 seconds

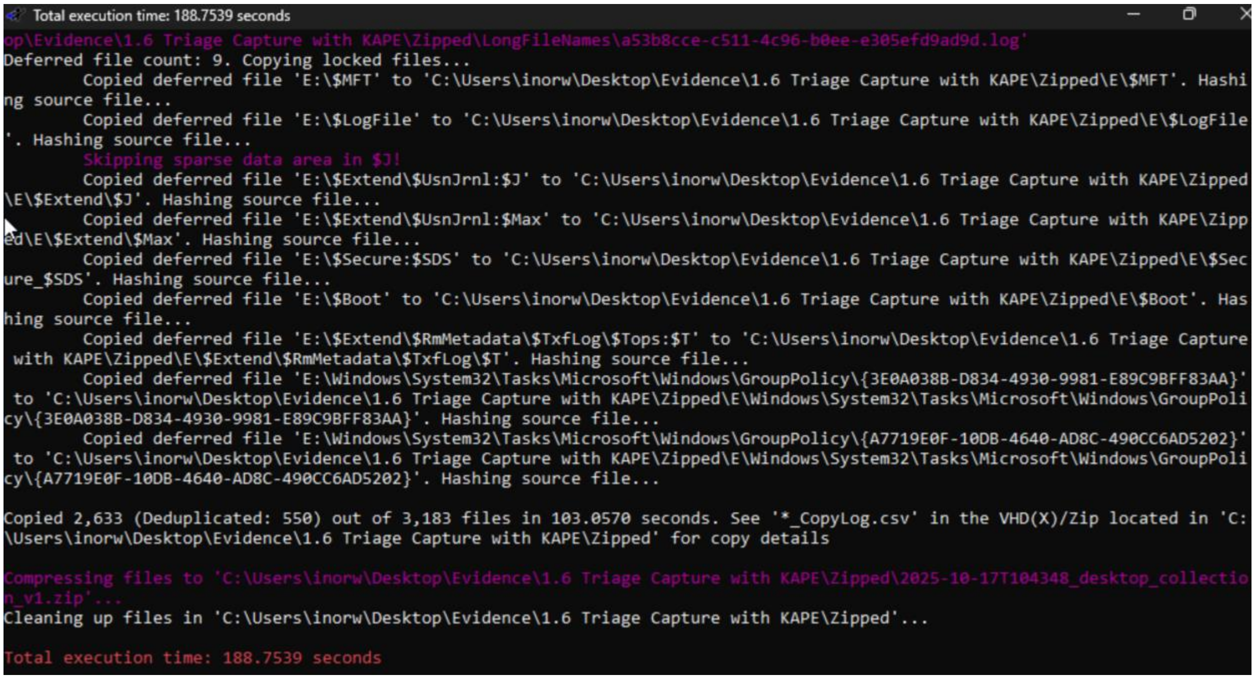
```
op\Evidence\1.6 Triage Capture with KAPE\Zipped\LongFileNames\a53b8cce-c511-4c96-b0ee-e305efd9ad9d.log'
Deferred file count: 9. Copying locked files...
        Copied deferred file 'E:\$MFT' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\E\$MFT'. Hashi
ng source file...
        Copied deferred file 'E:\$LogFile' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\E\$LogFile
'. Hashing source file...
        Skipping sparse data area in $J!
        Copied deferred file 'E:\$Extend\$UsnJrnl:$J' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped
\E\$Extend\$J'. Hashing source file...
        Copied deferred file 'E:\$Extend\$UsnJrnl:$Max' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipp
ed\E\$Extend\$Max'. Hashing source file...
        Copied deferred file 'E:\$Secure:$SDS' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\E\$Sec
ure_$SDS'. Hashing source file...
        Copied deferred file 'E:\$Boot' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\E\$Boot'. Has
hing source file...
        Copied deferred file 'E:\$Extend\$RmMetadata\$TxfLog\$Tops:$T' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture
 with KAPE\Zipped\E\$Extend\$RmMetadata\$TxfLog\$T'. Hashing source file...
        Copied deferred file 'E:\Windows\System32\Tasks\Microsoft\Windows\GroupPolicy\{3E0A038B-D834-4930-9981-E89C9BFF83AA}'
 to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\E\Windows\System32\Tasks\Microsoft\Windows\GroupPoli
cy\{3E0A038B-D834-4930-9981-E89C9BFF83AA}'. Hashing source file...
        Copied deferred file 'E:\Windows\System32\Tasks\Microsoft\Windows\GroupPolicy\{A7719E0F-10DB-4640-AD8C-490CC6AD5202}'
 to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\E\Windows\System32\Tasks\Microsoft\Windows\GroupPoli
cy\{A7719E0F-10DB-4640-AD8C-490CC6AD5202}'. Hashing source file...

Copied 2,633 (Deduplicated: 550) out of 3,183 files in 103.0570 seconds. See '*_CopyLog.csv' in the VHD(X)/Zip located in 'C:
\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped' for copy details

Compressing files to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\2025-10-17T104348_desktop_collectio
n_v1.zip'...
Cleaning up files in 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped'...

Total execution time: 188.7539 seconds
```

Figure 24. KAPE console excerpt for DESKTOP showing multiple Deferred file messages

3. Of the files copied, how many were duplicated?

- **Analysis Performed:**
  - Mounted the DESKTOP image with Arsenal and set the mounted Windows volume (E:) as the KAPE source.
  - Ran !SANS_Triage to a ZIP destination (no VSCs), then reviewed the KAPE console for outcome lines.
  - As shown in Figure 25, of the files copied, there were 550 duplicated files. Copied 2633 (with 550 duplicates) out of 3183 files in 103.0570 seconds.
- **Answer:**
  Of the files copied, there were **550 duplicated file**s, as shown in Figure 25.

- **Supporting Evidence:**



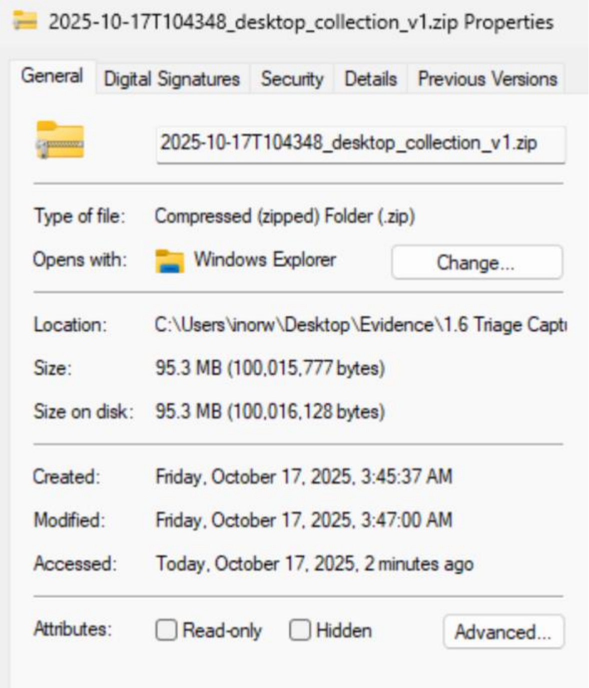
```
Total execution time: 188.7539 seconds
op\Evidence\1.6 Triage Capture with KAPE\Zipped\LongFileNames\a53b8cce-c511-4c96-b0ee-e305efd9ad9d.log'
Deferred file count: 9. Copying locked files...
        Copied deferred file 'E:\$MFT' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\E\$MFT'. Hashi
ng source file...
        Copied deferred file 'E:\$LogFile' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\E\$LogFile
'. Hashing source file...
        Skipping sparse data area in $J!
        Copied deferred file 'E:\$Extend\$UsnJrnl:$J' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped
\E\$Extend\$J'. Hashing source file...
        Copied deferred file 'E:\$Extend\$UsnJrnl:$Max' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipp
ed\E\$Extend\$Max'. Hashing source file...
        Copied deferred file 'E:\$Secure:$SDS' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\E\$Sec
ure_$SDS'. Hashing source file...
        Copied deferred file 'E:\$Boot' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\E\$Boot'. Has
hing source file...
        Copied deferred file 'E:\$Extend\$RmMetadata\$TxfLog\$Tops:$T' to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture
 with KAPE\Zipped\E\$Extend\$RmMetadata\$TxfLog\$T'. Hashing source file...
        Copied deferred file 'E:\Windows\System32\Tasks\Microsoft\Windows\GroupPolicy\{3E0A038B-D834-4930-9981-E89C9BFF83AA}'
 to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\E\Windows\System32\Tasks\Microsoft\Windows\GroupPoli
cy\{3E0A038B-D834-4930-9981-E89C9BFF83AA}'. Hashing source file...
        Copied deferred file 'E:\Windows\System32\Tasks\Microsoft\Windows\GroupPolicy\{A7719E0F-10DB-4640-AD8C-490CC6AD5202}'
 to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\E\Windows\System32\Tasks\Microsoft\Windows\GroupPoli
cy\{A7719E0F-10DB-4640-AD8C-490CC6AD5202}'. Hashing source file...

Copied 2,633 (Deduplicated: 550) out of 3,183 files in 103.0570 seconds. See '*_CopyLog.csv' in the VHD(X)/Zip located in 'C:
\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped' for copy details

Compressing files to 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\2025-10-17T104348_desktop_collectio
n_v1.zip'...
Cleaning up files in 'C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped'...

Total execution time: 188.7539 seconds
```

Figure 25. KAPE console excerpt for DESKTOP showing how many files were duplicated

4. What is the size (in MB) of the KAPE archive (ZIP) file for the desktop image?

- **Analysis Performed:**
  - Mounted the DESKTOP image with Arsenal and set the mounted Windows volume (E:) as the KAPE source.
  - Ran !SANS_Triage to a ZIP destination (no VSCs).
  - After it ran successfully, a zip file was created in the destination folder set by the examiner.
    - Path: *C:\Users\inorw\Desktop\Evidence\1.6 Triage Capture with KAPE\Zipped\2025-10-17T104348_desktop_collection_v1.zip*
  - The examiner opened the properties of the KAPE archive (ZIP) file for the desktop image as shown in Figure 26.
- **Answer:**
  The size (in MB) of the KAPE archive (ZIP) file for DESKTOP is **95.3 MB**, as shown in Figure 26.

- **Supporting Evidence:**



2025-10-17T104348_desktop_collection_v1.zip Properties

2025-10-17T104348_desktop_collection_v1.zip

Type of file: Compressed (zipped) Folder (.zip)

Opens with: Windows Explorer

Location: C:\Users\inorw\Desktop\Evidence\1.6 Triage Captu

Size: 95.3 MB (100,015,777 bytes)

Size on disk: 95.3 MB (100,016,128 bytes)

Created: Friday, October 17, 2025, 3:45:37 AM

Modified: Friday, October 17, 2025, 3:47:00 AM

Accessed: Today, October 17, 2025, 2 minutes ago

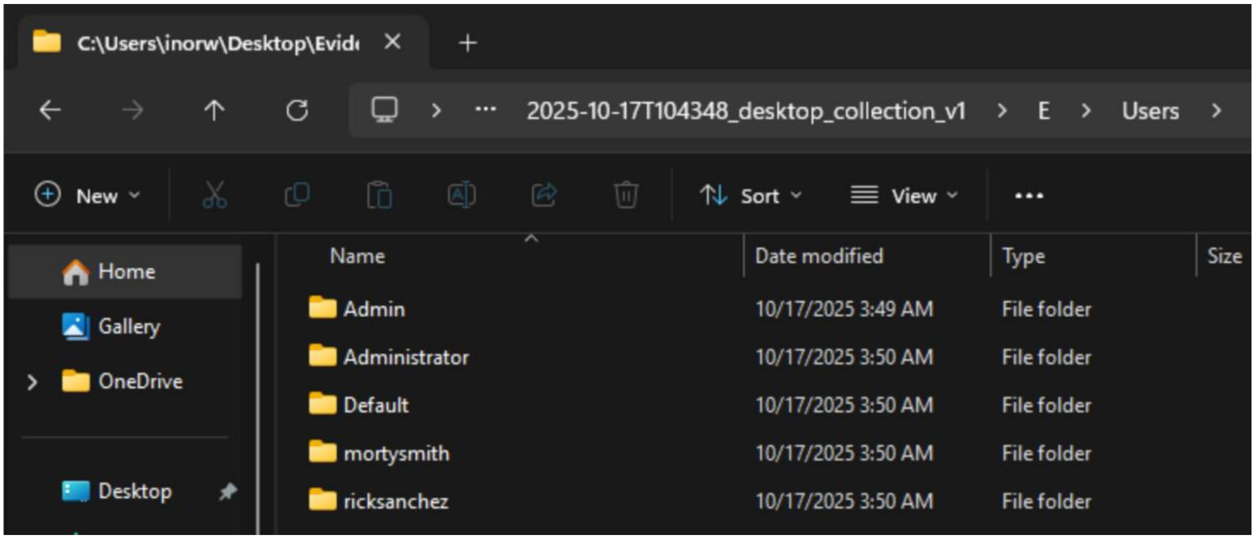
Attributes: □ Read-only □ Hidden

Figure 26. Properties tab of the KAPE archive (ZIP) file for DESKTOP

5. What user profiles were captured in this collection? (Do not include default).

- **Analysis Performed:**
  - Opened the extracted triage for DESKTOP and navigated to the \Users\ folder via the filesystem, as shown in Figure 27.
  - Reviewed the top-level profile folders and excluded the built-in Default profile as instructed.
- **Answer:**
  The non-default user profiles captured in this collection are: **Admin, Administrator, mortysmith, ricksanchez**, as shown in Figure 27. It seems like someone created the user account, "Admin", to trick system admins into thinking it is the normal Administrator account.

- **Supporting Evidence:**



C:\Users\inorw\Desktop\Evid

2025-10-17T104348_desktop_collection_v1 > E > Users >

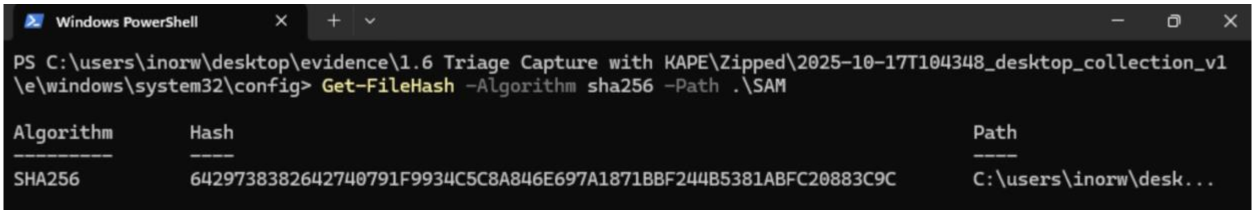
Name	Date modified	Type	Size
Admin	10/17/2025 3:49 AM	File folder	
Administrator	10/17/2025 3:50 AM	File folder	
Default	10/17/2025 3:50 AM	File folder	
mortysmith	10/17/2025 3:50 AM	File folder	
ricksanchez	10/17/2025 3:50 AM	File folder	

Figure 27. Extracted triage path ...\desktop_collection_v1\E\Users\ showing profile folders Admin, Administrator, Default (excluded), mortysmith, and ricksanchez.

6. Extract the SAM registry hive from this capture. What is the SHA256 hash of this file?

- **Analysis Performed:**
  - The examiner navigated to the extracted triage folder for DESKTOP and into the System32\config directory.
  - Opened Powershell in the directory and computed the SHA256 hash of the SAM hive using the following command, as shown in Figure 28:
    - Command: *Get-FileHash -Algorithm SHA256 -Path .\SAM*
- **Answer:**
  The examiner extracted the SAM registry hive from the DESKTOP capture and the SHA256 hash of the file is: “**6429738382642740791F9934C58A846E697A1871BBF244B5381ABFC20883C9C**”.

- **Supporting Evidence:**



```
PS C:\users\inorw\desktop\evidence\1.6 Triage Capture with KAPE\Zipped\2025-10-17T104348_desktop_collection_v1
\e\windows\system32\config> Get-FileHash -Algorithm sha256 -Path .\SAM

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          6429738382642740791F9934C5C8A846E697A1871BBF244B5381ABFC20883C9C       C:\users\inorw\desk...
```

*Figure 28. PowerShell Get-FileHash -Algorithm SHA256 -Path .\SAM executed in the triage directory, displaying the SHA256 digest for the **SAM** registry hive.*

Conclusion

The examiner, Inor Wang, enjoyed this lab! There is no critique from me. Thank you.

References

Carvey, H. A. (2014). Windows forensic analysis toolkit: Advanced analysis techniques for Windows 8 (Fourth edition). Syngress.

Johansen, G., & Safari, an O. M. C. (2020). Digital Forensics and Incident Response—Second Edition.

Ligh, M. H., Case, A., Levy, J., & Walters, A. (2014). The art of memory forensics: Detecting malware and threats in Windows, Linux, and Mac memory. Wiley.

Malware forensics field guide for Windows systems digital forensics field guides. (2012). Syngress.

Oettinger, W., & Safari, an O. M. C. (2020). Learn Computer Forensics.

Reddy, N. (2019). Practical cyber forensics: An incident-based approach to forensic investigations. APress. https://doi.org/10.1007/978-1-4842-4460-9.