Case Name: The NTFS File System

Couse Name: IST402

Instructor: Robert Price

Date: 02/27/2025

Examiner Name: Jaspreet Singh

Table of Contents

List of Illustrative Materials

NTFS File System

HxD Application

Autopsy 4.2.0

EXECUTIVE SUMMARY

Background

The primary file system for Windows is the New Technology File System, also known as (NTFS), and is the primary focus of this lab's analysis. The main goal was to understand its structure, security characteristics, and forensic value. NTFS's primary purpose is to provide the ability to store large files efficiently and provide advanced security features, making this a strong focus in forensic investigations examining NTFS and exploring raw disk data using a HEX editor. This forensic tool allows the examiner to view and analyze the raw data of a file in a hexadecimal format. Hashing to confirm image integrity, which plays a vital role in verifying the authentication of images, and utilizing Autopsy, an open-source forensic tool whose purpose is to analyze disk images, recover deleted files, and extract artifacts from NTFS partitions to analyze an NTFS partition, were among the examiner's primary activities.

In addition, in this lab, the examiner looks at the following characteristics of NTFS: Alternate Data Streams (ADS), a feature in the NTFS file system that allows compatibility with older versions of Mac OS. ADS can also be used to determine whether an individual is trying to hide data on their system. Encryption file system (EFS) is another NTFS feature that allows one to encrypt files and folders. Timestomp is used in the command prompt to change files, such as Modified, Access, and Created times.

Overall, by utilizing these NTFS features and forensic tools, the examiner can detect hidden data or files, verify a file's integrity, and track any modifications made to data or files.

Evidence

Description	Hash Algorithm	Hash Value	Examiner
Evidence	CRC32	93644201	Jaspreet Singh
Evidence	MD5	A2BA635AAF7AEEF814C6EA41A968E5DB	Jaspreet Singh
Evidence	SHA-1	CF27F528D469012B59EC6440D3A4085261176FD	Jaspreet Singh

COLLECTION AND ANALYSIS

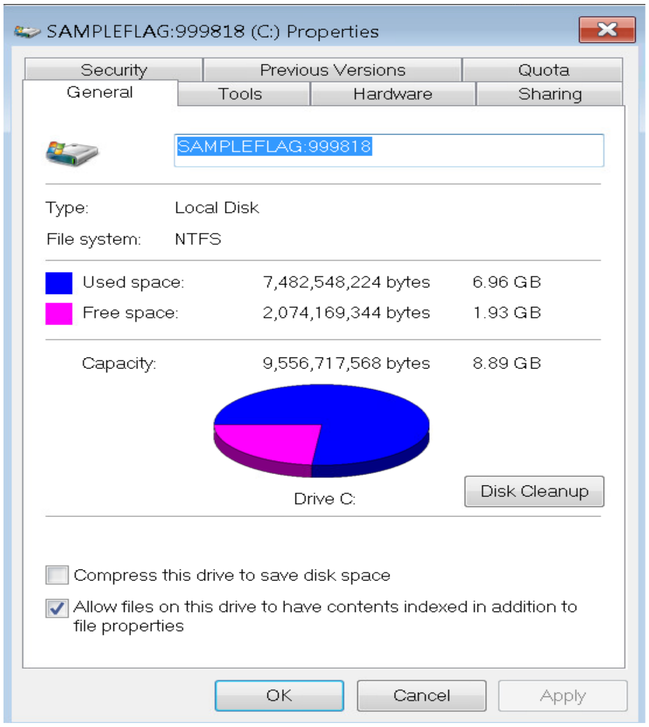
Collection

Through this process, the examiner could acquire and retrieve information from an NTFS partition without modifying the original evidence by looking at a disk image. This would then allow the integrity of the original evidence to remain intact when extracting data. Imaging and hashing the NTFS file system in this way would ensure that data is not altered and remains consistent when the examiner executes a forensic analysis. The examiner will use forensic tools to verify, interpret, and analyze the NTFS partition and the respective artifacts during this forensic analysis of the Windows file system, such as the HEX editor, command prompts, and Autopsy. Through hashing, the examiner will ensure that the image obtained can be trusted and has not been altered for forensic analysis.

Analysis

Local Drive NTFS File System

The examiner examined the local drive (C:) called SAMPLEFLAG:999818(C:) and from there the examiner opened the properties and saw the file system was NTFS meaning that the local drive supports advanced security permissions, features, encryption, ACLs and ADS.



SAMPLEFLAG:999818 (C:) Properties

SAMPLEFLAG:999818

Type: Local Disk

File system: NTFS

🟦 Used space:	7,482,548,224 bytes	6.96 GB
🟪 Free space:	2,074,169,344 bytes	1.93 GB
Capacity:	9,556,717,568 bytes	8.89 GB

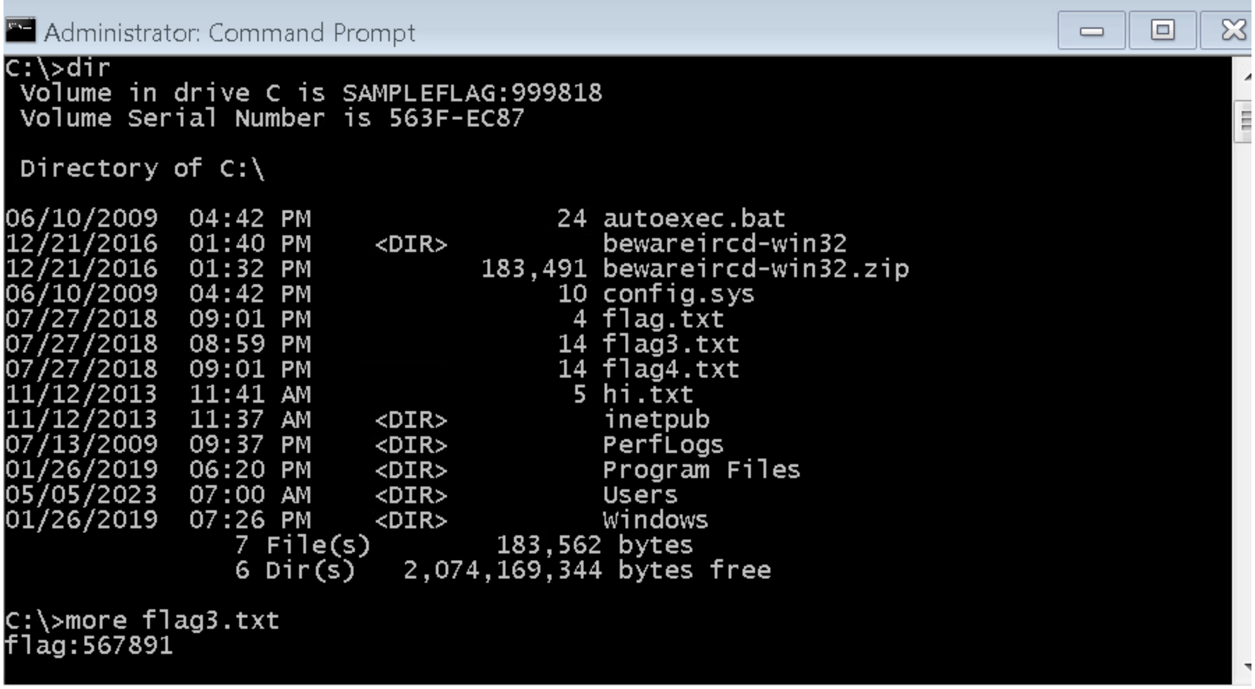
☐ Compress this drive to save disk space

☑ Allow files on this drive to have contents indexed in addition to file properties

Figure 1: Viewing Local Drive (C:) Properties

Command Prompt showing the Directory

In the command prompt the examiner is looking at the directory of the local drive (C:) by using the command “dir” the examiner can see the entire directory of SAMPLEFLAG:999818.

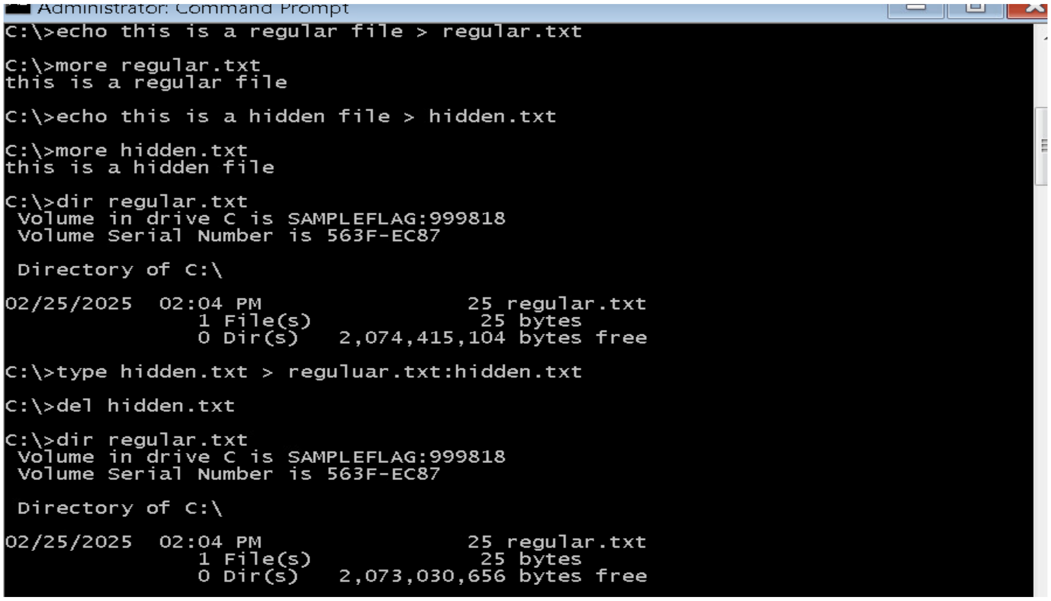


```
C:\>dir
 Volume in drive C is SAMPLEFLAG:999818
 Volume Serial Number is 563F-EC87

 Directory of C:\

06/10/2009  04:42 PM                24 autoexec.bat
12/21/2016  01:40 PM    <DIR>          bewareircd-win32
12/21/2016  01:32 PM           183,491 bewareircd-win32.zip
06/10/2009  04:42 PM                10 config.sys
07/27/2018  09:01 PM                 4 flag.txt
07/27/2018  08:59 PM                14 flag3.txt
07/27/2018  09:01 PM                14 flag4.txt
11/12/2013  11:41 AM                 5 hi.txt
11/12/2013  11:37 AM    <DIR>          inetpub
07/13/2009  09:37 PM    <DIR>          PerfLogs
01/26/2019  06:20 PM    <DIR>          Program Files
05/05/2023  07:00 AM    <DIR>          Users
01/26/2019  07:26 PM    <DIR>          Windows
               7 File(s)        183,562 bytes
               6 Dir(s)   2,074,169,344 bytes free

C:\>more flag3.txt
flag:567891
```

Figure 2: Viewing SAMPLEFLAG:999818 Drive Directory

Creating ADS

The examiner in the command prompt is creating an Alternate Data Stream (ADS) to allow files to store hidden data. So, in this the examiner created a file called regular.txt and then hid that file within another file called hidden.txt.

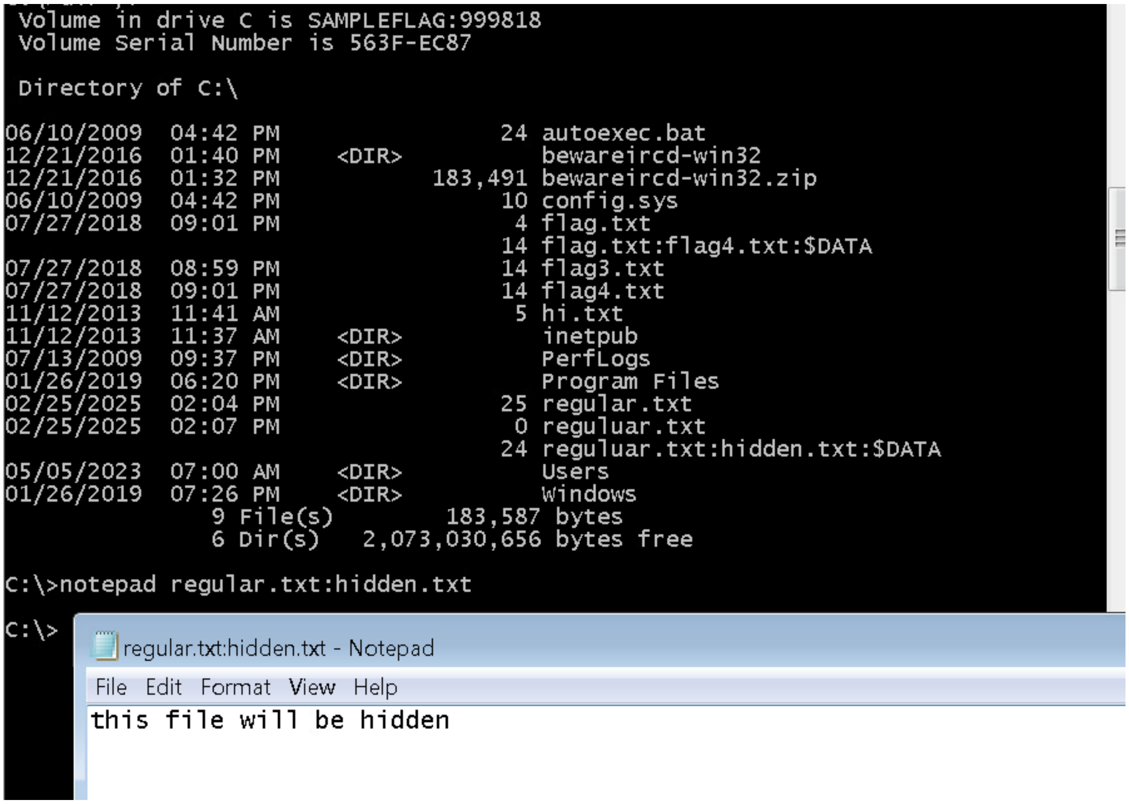


```
C:\>echo this is a regular file > regular.txt

C:\>more regular.txt
this is a regular file

C:\>echo this is a hidden file > hidden.txt

C:\>more hidden.txt
this is a hidden file

C:\>dir regular.txt
 Volume in drive C is SAMPLEFLAG:999818
 Volume Serial Number is 563F-EC87

 Directory of C:\

02/25/2025  02:04 PM                25 regular.txt
               1 File(s)             25 bytes
               0 Dir(s)   2,074,415,104 bytes free

C:\>type hidden.txt > reguluar.txt:hidden.txt

C:\>del hidden.txt

C:\>dir regular.txt
 Volume in drive C is SAMPLEFLAG:999818
 Volume Serial Number is 563F-EC87

 Directory of C:\

02/25/2025  02:04 PM                25 regular.txt
               1 File(s)             25 bytes
               0 Dir(s)   2,073,030,656 bytes free
```

Figure 3: Creating ADS in Command Prompt

Viewing ADS

The examiner used the command prompt to view the content within the hidden.txt file created to reveal the information in regular.txt in a notepad file.



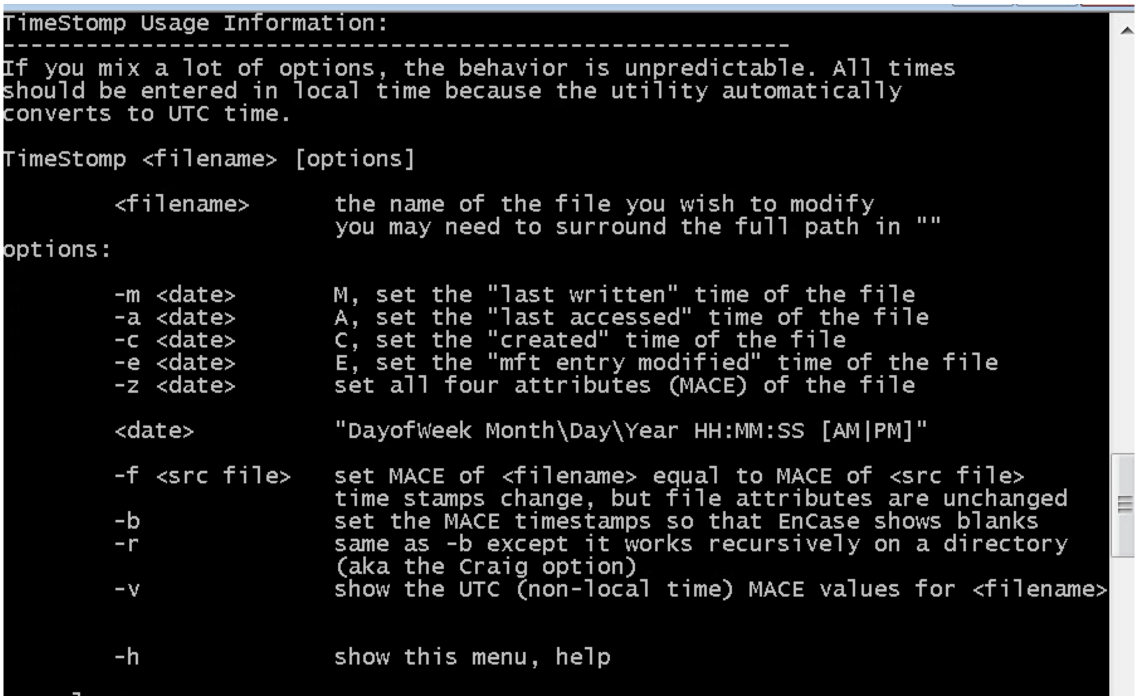
```
 Volume in drive C is SAMPLEFLAG:999818
 Volume Serial Number is 563F-EC87

 Directory of C:\

06/10/2009  04:42 PM                24 autoexec.bat
12/21/2016  01:40 PM    <DIR>          bewareircd-win32
12/21/2016  01:32 PM           183,491 bewareircd-win32.zip
06/10/2009  04:42 PM                10 config.sys
07/27/2018  09:01 PM                 4 flag.txt
                                    14 flag.txt:flag4.txt:$DATA
07/27/2018  08:59 PM                14 flag3.txt
07/27/2018  09:01 PM                14 flag4.txt
11/12/2013  11:41 AM                 5 hi.txt
11/12/2013  11:37 AM    <DIR>          inetpub
07/13/2009  09:37 PM    <DIR>          PerfLogs
01/26/2019  06:20 PM    <DIR>          Program Files
02/25/2025  02:04 PM                25 regular.txt
02/25/2025  02:07 PM                 0 reguluar.txt
                                    24 reguluar.txt:hidden.txt:$DATA
05/05/2023  07:00 AM    <DIR>          Users
01/26/2019  07:26 PM    <DIR>          Windows
               9 File(s)        183,587 bytes
               6 Dir(s)   2,073,030,656 bytes free

C:\>notepad regular.txt:hidden.txt

C:\>
```

```
this file will be hidden
```

Figure 4: Viewing ADS in command prompt.

Timestomp in Command Prompt

In the command prompt the examiner can view a files timestomps data. It allows for the manipulation of file data such as modified, accessed, created, and MFT entry modified. It's often used by hackers to change timestamps, so it makes it difficult for forensic analysis.

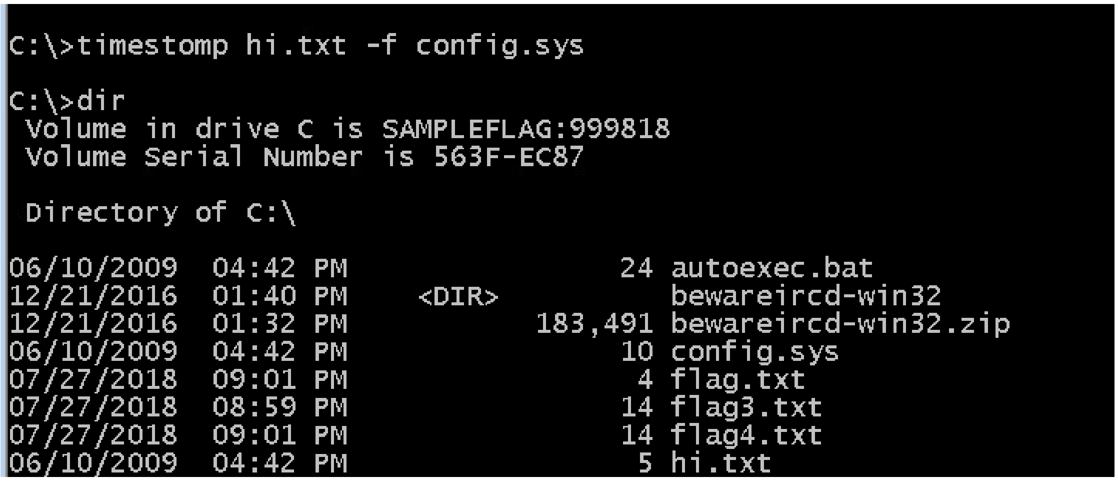


```
TimeStomp Usage Information:
----------------------------------------------------------
If you mix a lot of options, the behavior is unpredictable. All times
should be entered in local time because the utility automatically
converts to UTC time.

TimeStomp <filename> [options]

        <filename>      the name of the file you wish to modify
                        you may need to surround the full path in ""
options:

        -m <date>       M, set the "last written" time of the file
        -a <date>       A, set the "last accessed" time of the file
        -c <date>       C, set the "created" time of the file
        -e <date>       E, set the "mft entry modified" time of the file
        -z <date>       set all four attributes (MACE) of the file

        <date>          "DayofWeek Month\Day\Year HH:MM:SS [AM|PM]"

        -f <src file>   set MACE of <filename> equal to MACE of <src file>
                        time stamps change, but file attributes are unchanged
        -b              set the MACE timestamps so that EnCase shows blanks
        -r              same as -b except it works recursively on a directory
                        (aka the Craig option)
        -v              show the UTC (non-local time) MACE values for <filename>


        -h              show this menu, help
```

Figure 5: Timestomp analysis in Command Prompt

Using Timestomp on a .txt file.

In the command prompt the examiner is using the Timestomp tool to manipulate the file ht.txt and copies timestomps from config.sys into the txt file.

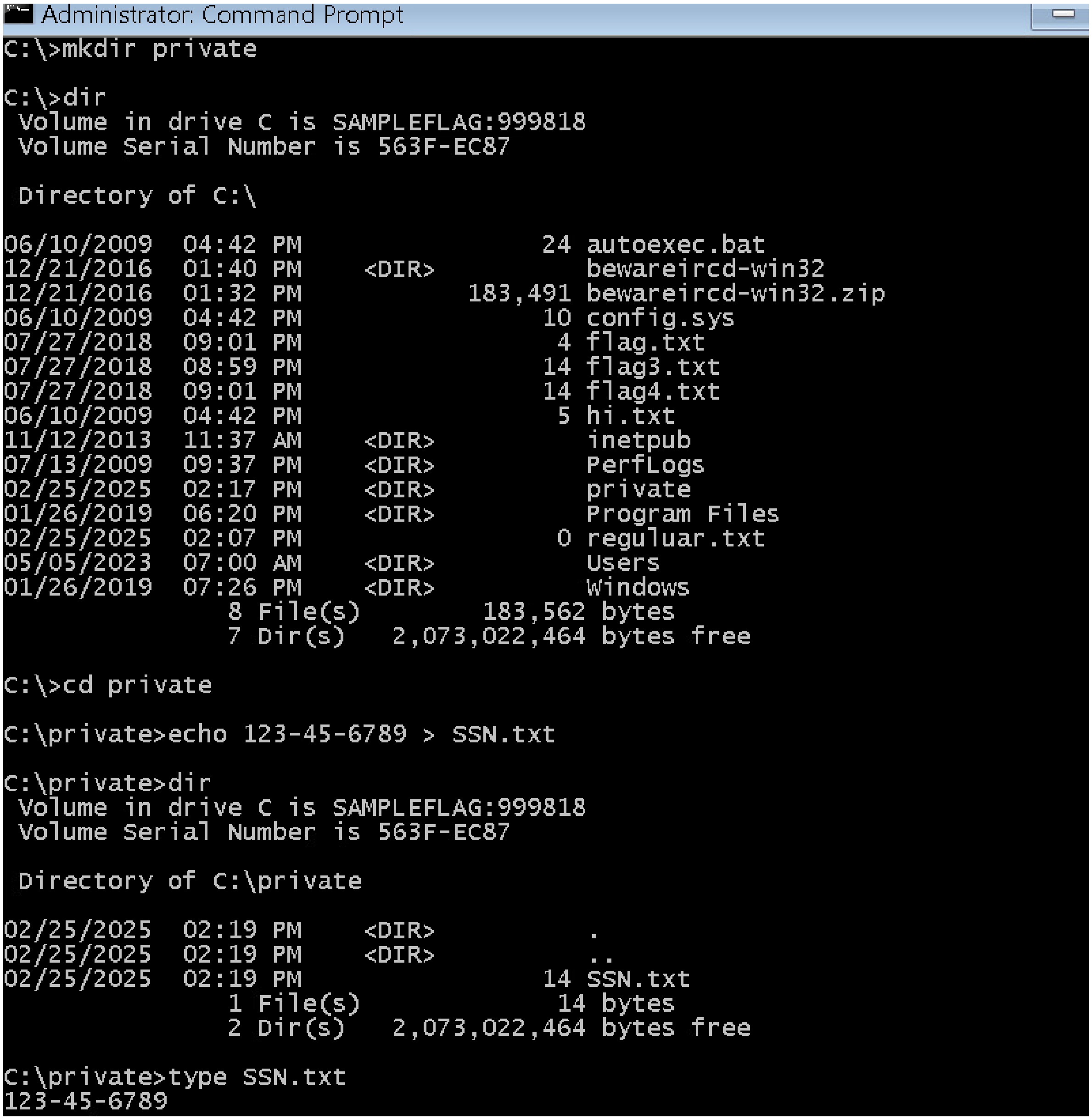


```
C:\>timestomp hi.txt -f config.sys

C:\>dir
 Volume in drive C is SAMPLEFLAG:999818
 Volume Serial Number is 563F-EC87

 Directory of C:\

06/10/2009  04:42 PM                24 autoexec.bat
12/21/2016  01:40 PM    <DIR>          bewareircd-win32
12/21/2016  01:32 PM           183,491 bewareircd-win32.zip
06/10/2009  04:42 PM                10 config.sys
07/27/2018  09:01 PM                 4 flag.txt
07/27/2018  08:59 PM                14 flag3.txt
07/27/2018  09:01 PM                14 flag4.txt
06/10/2009  04:42 PM                 5 hi.txt
```

Figure 6: Timestomp on .txt file

Creating a new Directory and File then Testing

The examiner created a new directory in drive C: called private and created a new file called SSN.txt with an inputter SSN. Then the examiner checked the C:\privat directory and verified that the new SSN.txt file created was present and viewed the files contents.

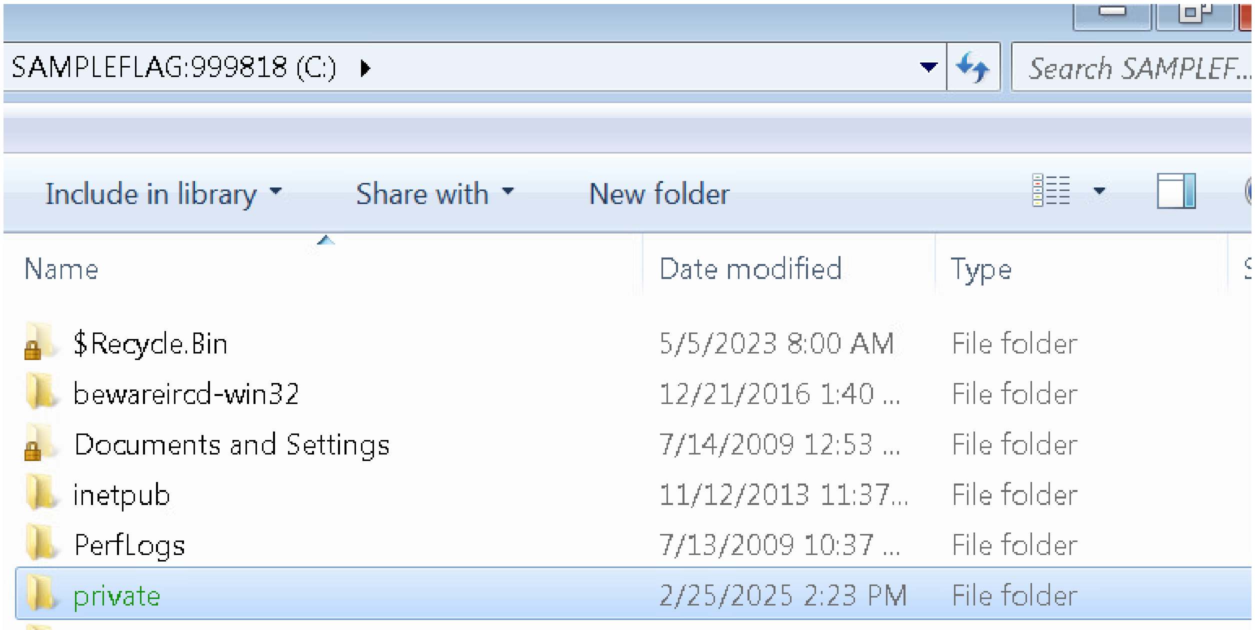


```
C:\>mkdir private

C:\>dir
 Volume in drive C is SAMPLEFLAG:999818
 Volume Serial Number is 563F-EC87

 Directory of C:\

06/10/2009  04:42 PM                24 autoexec.bat
12/21/2016  01:40 PM    <DIR>          bewareircd-win32
12/21/2016  01:32 PM           183,491 bewareircd-win32.zip
06/10/2009  04:42 PM                10 config.sys
07/27/2018  09:01 PM                 4 flag.txt
07/27/2018  08:59 PM                14 flag3.txt
07/27/2018  09:01 PM                14 flag4.txt
06/10/2009  04:42 PM                 5 hi.txt
11/12/2013  11:37 AM    <DIR>          inetpub
07/13/2009  09:37 PM    <DIR>          PerfLogs
02/25/2025  02:17 PM    <DIR>          private
01/26/2019  06:20 PM    <DIR>          Program Files
02/25/2025  02:07 PM                 0 reguluar.txt
05/05/2023  07:00 AM    <DIR>          Users
01/26/2019  07:26 PM    <DIR>          Windows
               8 File(s)        183,562 bytes
               7 Dir(s)   2,073,022,464 bytes free

C:\>cd private

C:\private>echo 123-45-6789 > SSN.txt

C:\private>dir
 Volume in drive C is SAMPLEFLAG:999818
 Volume Serial Number is 563F-EC87

 Directory of C:\private

02/25/2025  02:19 PM    <DIR>          .
02/25/2025  02:19 PM    <DIR>          ..
02/25/2025  02:19 PM                14 SSN.txt
               1 File(s)             14 bytes
               2 Dir(s)   2,073,022,464 bytes free

C:\private>type SSN.txt
123-45-6789
```

Figure 7: Creating new private directory and SSN.txt file

Viewing drive (C:) Private Folder

The examiner went into the local drive (C:) and saw the private folders properties and went to advanced. From there the examiner encrypted the contents and applied the changes to the entire private folder and its contents and now the folder has green text to show that it's encrypted.

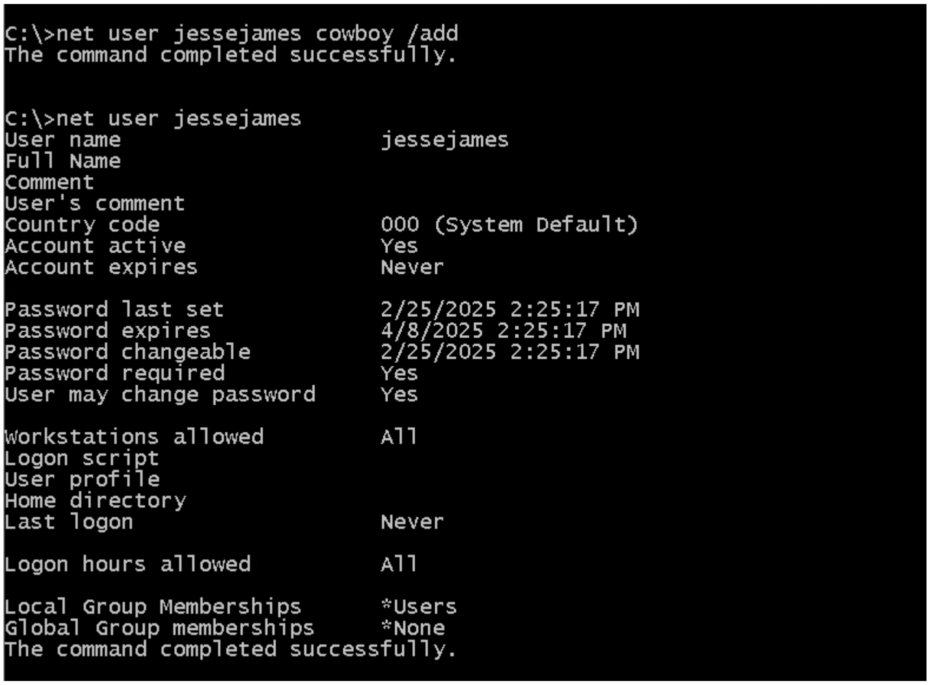


Name	Date modified	Type
$Recycle.Bin	5/5/2023 8:00 AM	File folder
bewareircd-win32	12/21/2016 1:40 ...	File folder
Documents and Settings	7/14/2009 12:53 ...	File folder
inetpub	11/12/2013 11:37...	File folder
PerfLogs	7/13/2009 10:37 ...	File folder
private	2/25/2025 2:23 PM	File folder

Figure 8: Encrypting private folder in drive (C:)

Creating new User in command prompt

The examiner in the command prompt is adding a new user called Jesse James with the password cowboy and then verifying after creating the user if it's in the system.

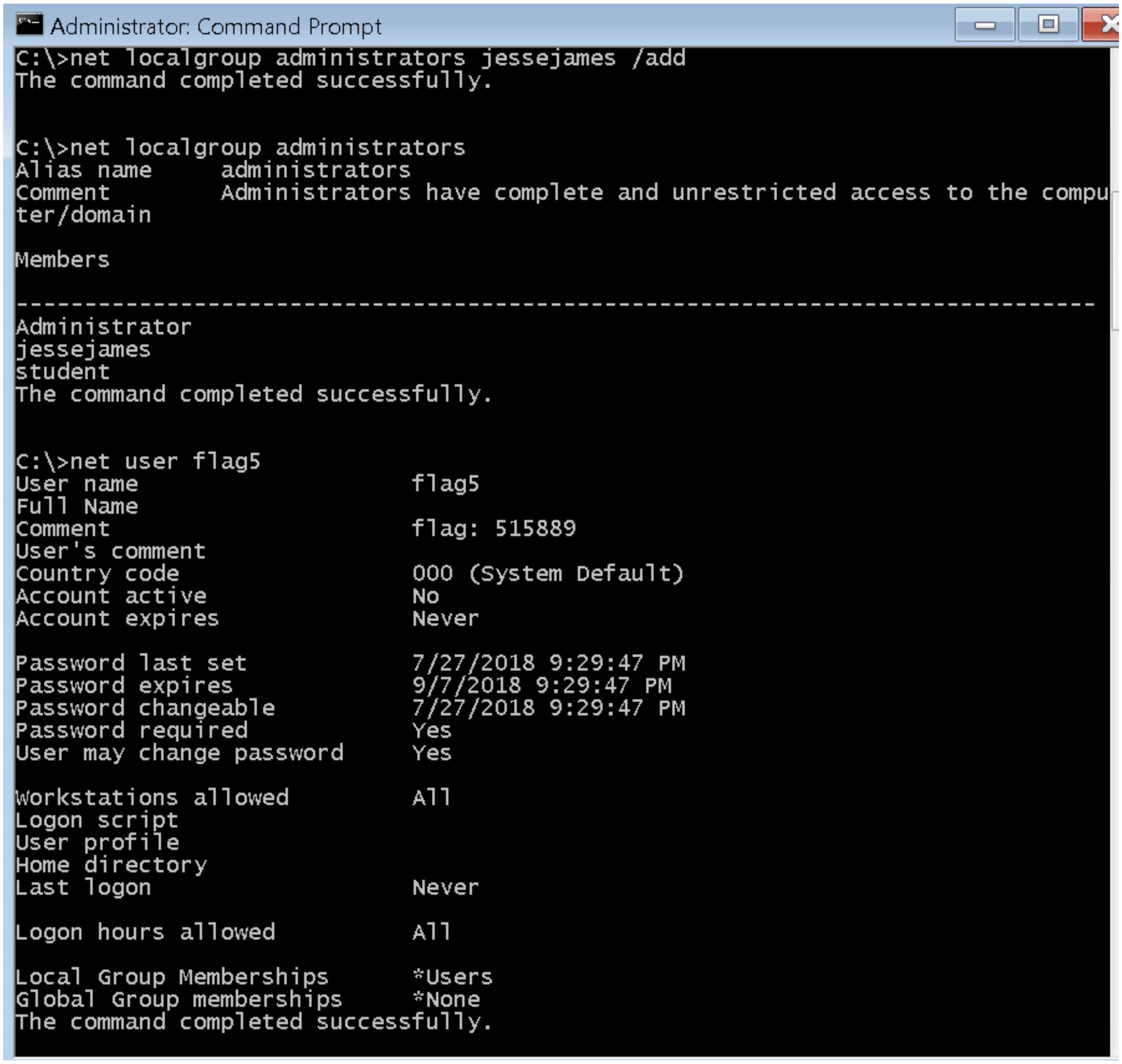


```
C:\>net user jessejames cowboy /add
The command completed successfully.


C:\>net user jessejames
User name                    jessejames
Full Name
Comment
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            2/25/2025 2:25:17 PM
Password expires             4/8/2025 2:25:17 PM
Password changeable          2/25/2025 2:25:17 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   Never

Logon hours allowed          All

Local Group Memberships      *Users
Global Group memberships     *None
The command completed successfully.
```

Figure 9: Creating new user

Adding New User as Administrator

The examiner after verifying that jessejames has been added will then add jessejames as an administrator and once that is completed. The examiner can verify that both jessejames and student are the two administrators on the device.



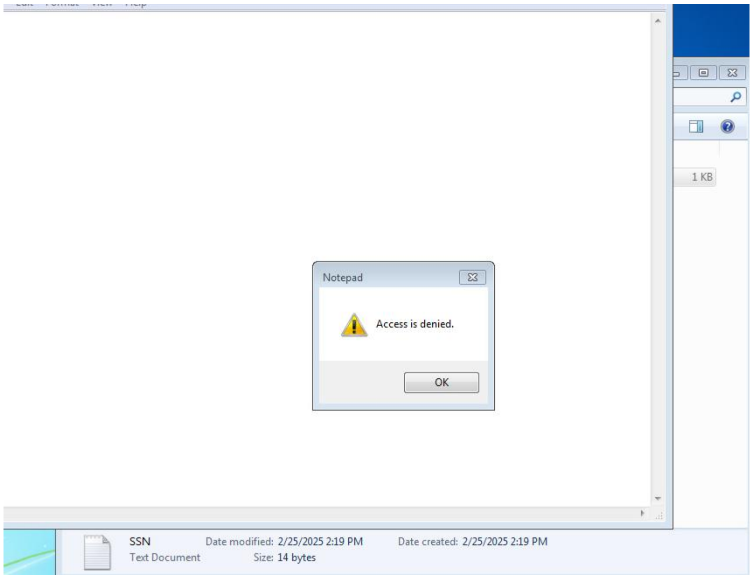
Administrator: Command Prompt

```
C:\>net localgroup administrators jessejames /add
The command completed successfully.


C:\>net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the compu
ter/domain

Members

-------------------------------------------------------------------------------
Administrator
jessejames
student
The command completed successfully.


C:\>net user flag5
User name                    flag5
Full Name
Comment                      flag: 515889
User's comment
Country code                 000 (System Default)
Account active               No
Account expires              Never

Password last set            7/27/2018 9:29:47 PM
Password expires             9/7/2018 9:29:47 PM
Password changeable          7/27/2018 9:29:47 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   Never

Logon hours allowed          All

Local Group Memberships      *Users
Global Group memberships     *None
The command completed successfully.
```

Figure 10: Adding new user as administrator

Switching users to test accessibility to the encrypted private folder

The examiner logged out of student's profile and logged into the new administrator file jessejames. The examiner let the environment boot up fully then went into the local drive (C:) on jessejames to try to open the encrypted file in private folder called SSN.txt. But the user was denied access to the file since the file is encrypted. The user does not have proper authority or key to open the file therefore cannot access the file.


Notepad

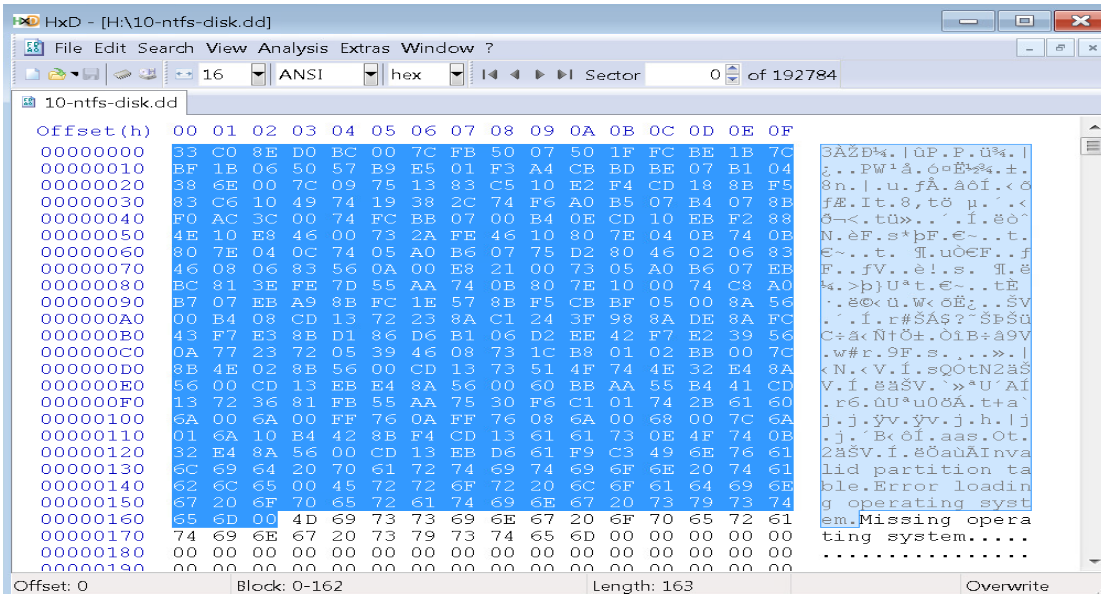
Access is denied.

SSN
Text Document
Date modified: 2/25/2025 2:19 PM
Size: 14 bytes
Date created: 2/25/2025 2:19 PM

Figure 11: Denied access to file SSN.txt

Using HxD Application

The examiner opened the HxD application and went to the computer's local drive (C:) to open a disk image called 10-ntfs-disk.dd-shortcut on the application. From there the examiner selected the sector size of 512 bytes and highlighted from bytes 00000000 to 00000162 to show the following message on the ASCII table on the right "Error loading operating system".

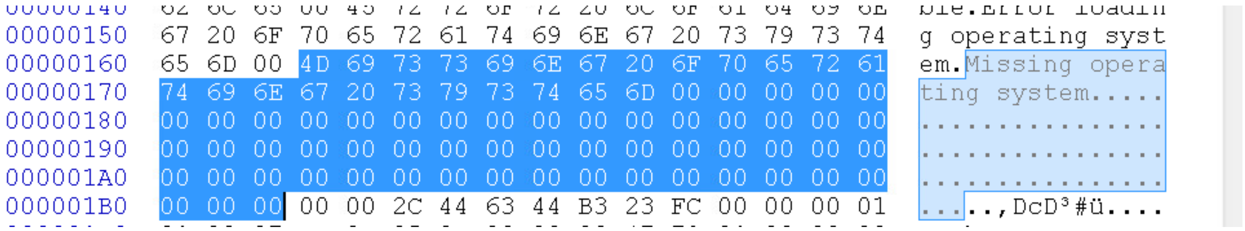

HxD - [H:\10-ntfs-disk.dd]
File Edit Search View Analysis Extras Window ?
16
ANSI
hex
Sector
0
of 192784
10-ntfs-disk.dd

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000   33 C0 8E D0 BC 00 7C FB 50 07 50 1F FC BE 1B 7C   3ÀŽÐ¼.|ûP.P.ü¾.|
00000010   BF 1B 06 50 57 B9 E5 01 F3 A4 CB BD BE 07 B1 04   ¿..PW¹å.ó¤Ë½¾.±.
00000020   38 6E 00 7C 09 75 13 83 C5 10 E2 F4 CD 18 8B F5   8n.|.u.ƒÅ.âôÍ.‹õ
00000030   83 C6 10 49 74 19 38 2C 74 F6 A0 B5 07 B4 07 8B   ƒÆ.It.8,tö µ.´.‹
00000040   F0 AC 3C 00 74 FC BB 07 00 B4 0E CD 10 EB F2 88   ð¬<.tü»..´.Í.ëòˆ
00000050   4E 10 E8 46 00 73 2A FE 46 10 80 7E 04 0B 74 0B   N.èF.s*þF.€~..t.
00000060   80 7E 04 0C 74 05 A0 B6 07 75 D2 80 46 02 06 83   €~..t. ¶.uÒ€F..ƒ
00000070   46 08 06 83 56 0A 00 E8 21 00 73 05 A0 B6 07 EB   F..ƒV..è!.s. ¶.ë
00000080   BC 81 3E FE 7D 55 AA 74 0B 80 7E 10 00 74 C8 A0   ¼.>þ}Uªt.€~..tÈ 
00000090   B7 07 EB A9 8B FC 1E 57 8B F5 CB BF 05 00 8A 56   ·.ë©‹ü.W‹õË¿..ŠV
000000A0   00 B4 08 CD 13 72 23 8A C1 24 3F 98 8A DE 8A FC   .´.Í.r#ŠÁ$?˜ŠÞŠü
000000B0   43 F7 E3 8B D1 86 D6 B1 06 D2 EE 42 F7 E2 39 56   C÷ã‹Ñ†Ö±.ÒîB÷â9V
000000C0   0A 77 23 72 05 39 46 08 73 1C B8 01 02 BB 00 7C   .w#r.9F.s.¸..».|
000000D0   8B 4E 02 8B 56 00 CD 13 73 51 4F 74 4E 32 E4 8A   ‹N.‹V.Í.sQOtN2äŠ
000000E0   56 00 CD 13 EB E4 8A 56 00 60 BB AA 55 B4 41 CD   V.Í.ëäŠV.`»ªU´AÍ
000000F0   13 72 36 81 FB 55 AA 75 30 F6 C1 01 74 2B 61 60   .r6.ûUªu0öÁ.t+a`
00000100   6A 00 6A 00 FF 76 0A FF 76 08 6A 00 68 00 7C 6A   j.j.ÿv.ÿv.j.h.|j
00000110   01 6A 10 B4 42 8B F4 CD 13 61 61 73 0E 4F 74 0B   .j.´B‹ôÍ.aas.Ot.
00000120   32 E4 8A 56 00 CD 13 EB D6 61 F9 C3 49 6E 76 61   2äŠV.Í.ëÖaùÃInva
00000130   6C 69 64 20 70 61 72 74 69 74 69 6F 6E 20 74 61   lid partition ta
00000140   62 6C 65 00 45 72 72 6F 72 20 6C 6F 61 64 69 6E   ble.Error loadin
00000150   67 20 6F 70 65 72 61 74 69 6E 67 20 73 79 73 74   g operating syst
00000160   65 6D 00 4D 69 73 73 69 6E 67 20 6F 70 65 72 61   em.Missing opera
00000170   74 69 6E 67 20 73 79 73 74 65 6D 00 00 00 00 00   ting system.....
00000180   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00000190   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Offset: 0          Block: 0-162          Length: 163          Overwrite
```

Figure 12: Loading disk image on HxD Application

Examining ASCII table

The examiner highlights the following bytes from 00000163 to 000001B2 to show the rest of the message which shows “Missing operating system” on the ASCII table.

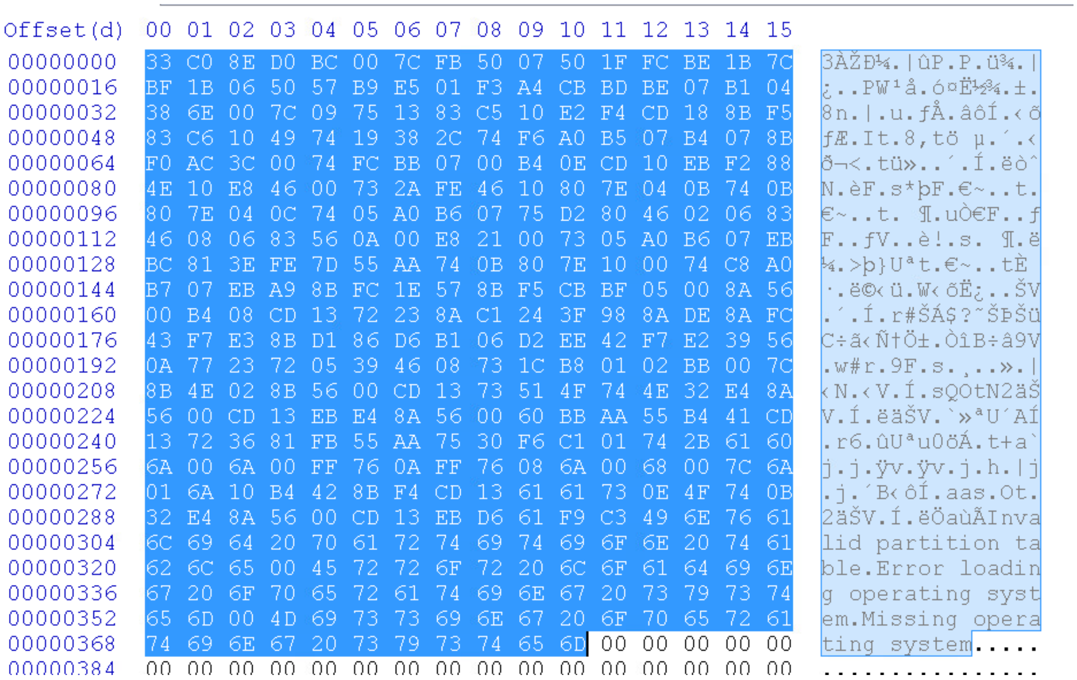


```
00000140  62 6C 65 00 45 72 72 6F 72 20 6C 6F 61 64 69 6E  ble.Error loadin
00000150  67 20 6F 70 65 72 61 74 69 6E 67 20 73 79 73 74  g operating syst
00000160  65 6D 00 4D 69 73 73 69 6E 67 20 6F 70 65 72 61  em.Missing opera
00000170  74 69 6E 67 20 73 79 73 74 65 6D 00 00 00 00 00  ting system.....
00000180  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001B0  00 00 00 00 00 2C 44 63 44 B3 23 FC 00 00 00 01  ....,DcD³#ü....
```

Figure 13: Rest of ASCII Message

Viewing from h to d

The examiner changes the offset located on the upper left from h to d which is decimal values. The examiner switches from hexadecimal to decimal because it is more natural for people to understand and that also makes it easier to understand partitions.

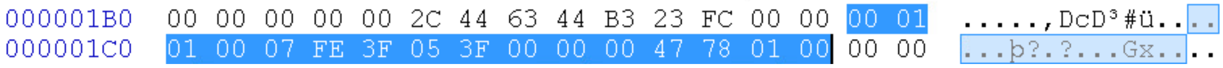


```
Offset(d) 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15

00000000  33 C0 8E D0 BC 00 7C FB 50 07 50 1F FC BE 1B 7C  3ÀŽĐ¼.|ûP.P.ü¾.|
00000016  BF 1B 06 50 57 B9 E5 01 F3 A4 CB BD BE 07 B1 04  ¿..PW¹å.ó¤Ë½¾.±.
00000032  38 6E 00 7C 09 75 13 83 C5 10 E2 F4 CD 18 8B F5  8n.|.u.ƒÅ.âôÍ.‹õ
00000048  83 C6 10 49 74 19 38 2C 74 F6 A0 B5 07 B4 07 8B  ƒÆ.It.8,tö µ.´.‹
00000064  F0 AC 3C 00 74 FC BB 07 00 B4 0E CD 10 EB F2 88  ð¬<.tü»..´.Í.ëòˆ
00000080  4E 10 E8 46 00 73 2A FE 46 10 80 7E 04 0B 74 0B  N.èF.s*þF.€~..t.
00000096  80 7E 04 0C 74 05 A0 B6 07 75 D2 80 46 02 06 83  €~..t. ¶.uÒ€F..ƒ
00000112  46 08 06 83 56 0A 00 E8 21 00 73 05 A0 B6 07 EB  F..ƒV..è!.s. ¶.ë
00000128  BC 81 3E FE 7D 55 AA 74 0B 80 7E 10 00 74 C8 A0  ¼.>þ}Uªt.€~..tÈ 
00000144  B7 07 EB A9 8B FC 1E 57 8B F5 CB BF 05 00 8A 56  ·.ë©‹ü.W‹õË¿..ŠV
00000160  00 B4 08 CD 13 72 23 8A C1 24 3F 98 8A DE 8A FC  .´.Í.r#ŠÁ$?˜ŠÞŠü
00000176  43 F7 E3 8B D1 86 D6 B1 06 D2 EE 42 F7 E2 39 56  C÷ã‹Ñ†Ö±.ÒîB÷â9V
00000192  0A 77 23 72 05 39 46 08 73 1C B8 01 02 BB 00 7C  .w#r.9F.s.¸..».|
00000208  8B 4E 02 8B 56 00 CD 13 73 51 4F 74 4E 32 E4 8A  ‹N.‹V.Í.sQOtN2äŠ
00000224  56 00 CD 13 EB E4 8A 56 00 60 BB AA 55 B4 41 CD  V.Í.ëäŠV.`»ªU´AÍ
00000240  13 72 36 81 FB 55 AA 75 30 F6 C1 01 74 2B 61 60  .r6.ûUªu0öÁ.t+a`
00000256  6A 00 6A 00 FF 76 0A FF 76 08 6A 00 68 00 7C 6A  j.j.ÿv.ÿv.j.h.|j
00000272  01 6A 10 B4 42 8B F4 CD 13 61 61 73 0E 4F 74 0B  .j.´B‹ôÍ.aas.Ot.
00000288  32 E4 8A 56 00 CD 13 EB D6 61 F9 C3 49 6E 76 61  2äŠV.Í.ëÖaùÃInva
00000304  6C 69 64 20 70 61 72 74 69 74 69 6F 6E 20 74 61  lid partition ta
00000320  62 6C 65 00 45 72 72 6F 72 20 6C 6F 61 64 69 6E  ble.Error loadin
00000336  67 20 6F 70 65 72 61 74 69 6E 67 20 73 79 73 74  g operating syst
00000352  65 6D 00 4D 69 73 73 69 6E 67 20 6F 70 65 72 61  em.Missing opera
00000368  74 69 6E 67 20 73 79 73 74 65 6D 00 00 00 00 00  ting system.....
00000384  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Figure 14: Switching Offset from h to d

Viewing the first partition

The examiner looks at each partition of the disk image which has a total of 64 bytes in the HxD application. First the examiner starts with the first partition which is from 1BE to 1CD. This first partition shows that it’s a non-bootable partition with the entry of 00 at the beginning.

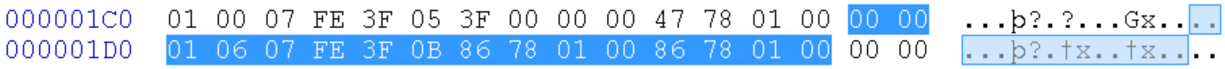


```
000001B0  00 00 00 00 00 2C 44 63 44 B3 23 FC 00 00 00 01  .....,DcD³#ü...
000001C0  01 00 07 FE 3F 05 3F 00 00 00 47 78 01 00 00 00  ...þ?.?...Gx....
```

Figure 15: First Partition of Disk Image

Viewing the second partition

The examiner looks at the next 16-byte partition from 1CE to 1DD.

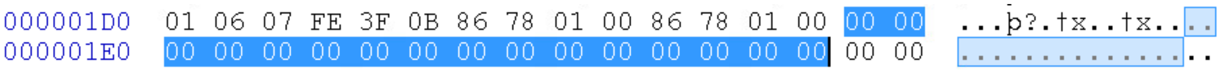


```
000001C0  01 00 07 FE 3F 05 3F 00 00 00 47 78 01 00 00 00  ...þ?.?...Gx....
000001D0  01 06 07 FE 3F 0B 86 78 01 00 86 78 01 00 00 00  ...þ?.†x..†x....
```

Figure 16: Second Partition of Disk Image

Viewing the third partition

The examiner looks at the third 16-byte partition from 1DE to 1ED.

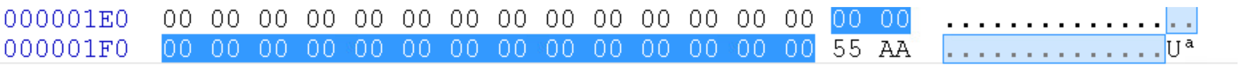


```
000001D0  01 06 07 FE 3F 0B 86 78 01 00 86 78 01 00 00 00  ...þ?.†x..†x....
000001E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Figure 17: Third Partition of Disk Image

Viewing the fourth partition

The examiner looks at the last 16-byte partition of the 64-byte disk image from m 1EE to 1FD.

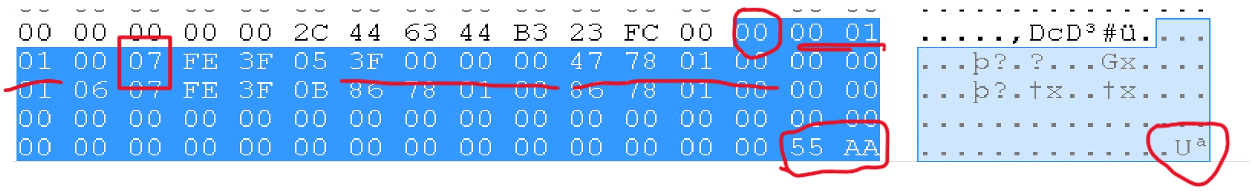


```
000001E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AA  ..............Uª
```

Figure 18: Fourth Partition of Disk Image

Viewing the full partition

After the examiner analyzes the four separate partitions, they can now see the full 64-byte disk image and see that in the full partition the next three bytes after the 00 non-bootable indicate the head, sector, and cylinder known as the CHS address (0,1,1). Then the examiner sees 07 after the CHS address indication the partition type which is 07 hence indicating it's a NTFS partition. Then the examiner looks at bytes 3F 00 00 00 indicating the Logical Block Addressing (LBA) which is used by computers to access and locate data in storage. Then the next four bytes the examiner sees is 47 78 01 00 which indicates the size in sectors of the partition which means that each entry is 16 bytes long. Finally at the end the examiner sees 55 AA which signifies an MBR signature.

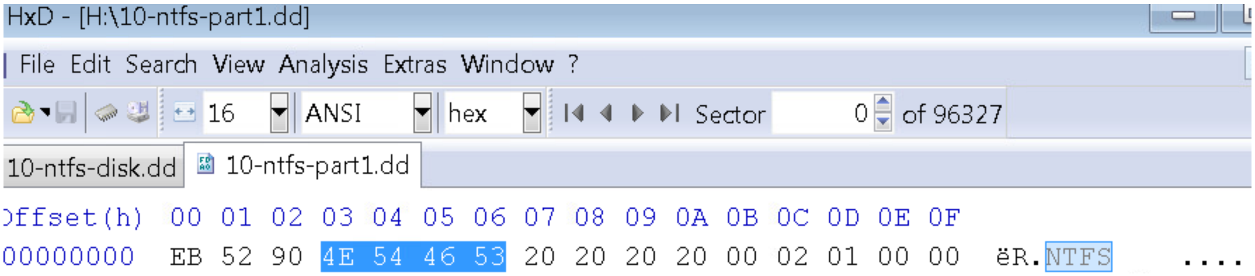


```
00 00 00 00 00 2C 44 63 44 B3 23 FC 00 00 00 01  ......,DcD³#ü...
01 00 07 FE 3F 05 3F 00 00 00 47 78 01 00 00 00  ...þ?.?...Gx....
01 06 07 FE 3F 0B 86 78 01 00 86 78 01 00 00 00  ...þ?.†x..†x....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AA  ..............Uª
```

Figure 19: Full partition explanation

Opening New Disk Image

The examiner opens new disk image from the local drive (C:) called 10-ntfs-part1.dd with a sector size of 512 bytes and in the partition the examiner analyzes the following bytes 4E 54 46 53 and sees in the message on the ASCII table on right “NTFS”.



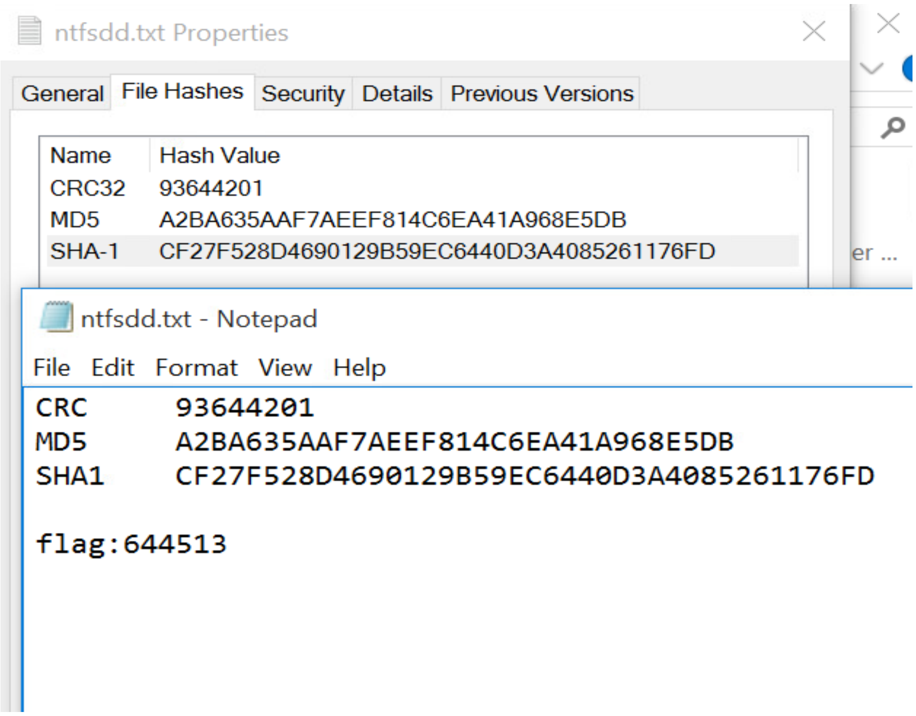
HxD - [H:\10-ntfs-part1.dd]
File Edit Search View Analysis Extras Window ?
16
ANSI
hex
Sector 0 of 96327
10-ntfs-disk.dd
10-ntfs-part1.dd

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000  EB 52 90 4E 54 46 53 20 20 20 20 00 02 01 00 00  ëR.NTFS    ....
```

Figure 20: New disk image NTFS message

Verifying Evidence Hash Values

The examiner has now logged into windows 10 and went to the Local Drive(C:) file and opened images. Then clicked the file called ntfsdd.txt. From there the examiner can see the CRC, MD5, and SHA1 values. The examiner from there can go into the ntfsdd.txt files properties and then look at the hash values in the properties. Then the examiner can review both the hash values in notepad and properties to verify that each match is correct.



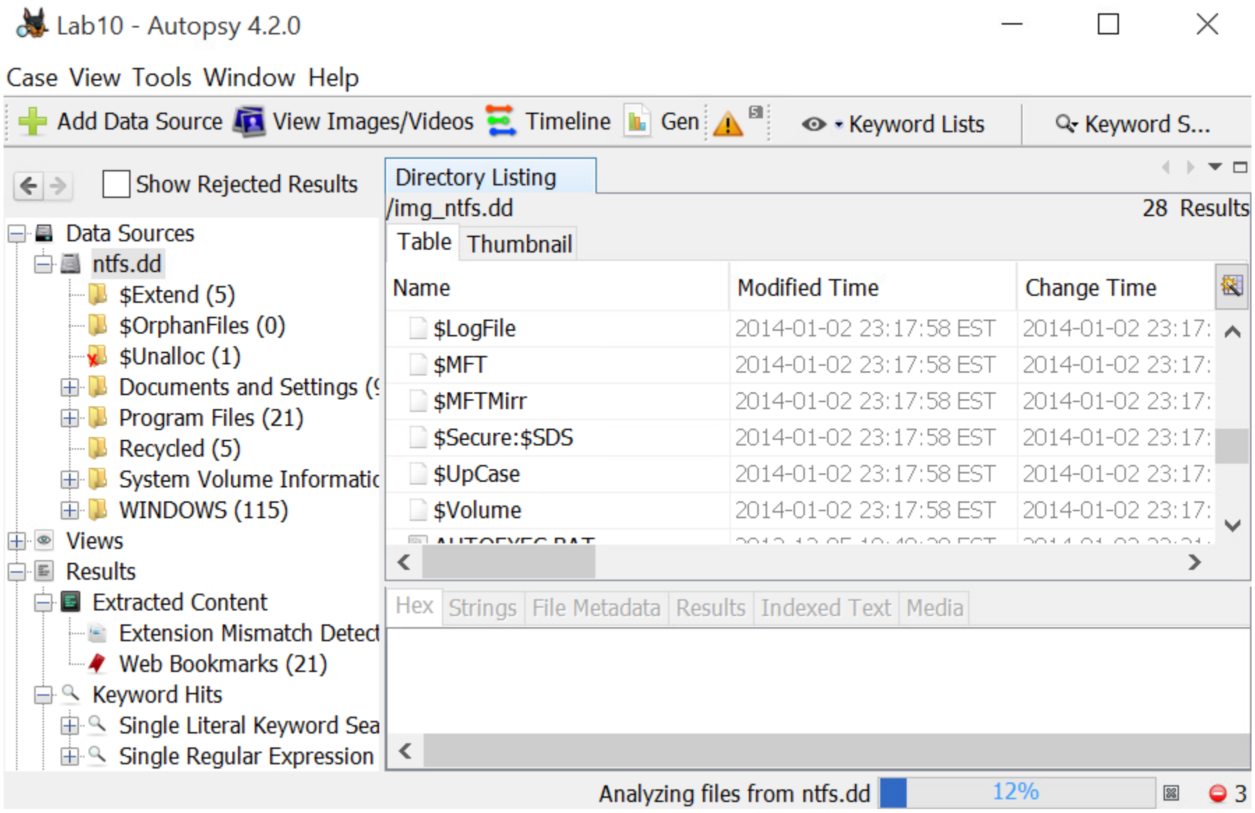
Name	Hash Value
CRC32	93644201
MD5	A2BA635AAF7AEEF814C6EA41A968E5DB
SHA-1	CF27F528D4690129B59EC6440D3A4085261176FD

```
CRC     93644201
MD5     A2BA635AAF7AEEF814C6EA41A968E5DB
SHA1    CF27F528D4690129B59EC6440D3A4085261176FD

flag:644513
```

Figure 21: Verifying hash values

NTFS Partition with Autopsy

The examiner used Autopsy 4.2.0 application to create a new case named Lab0 for the base directory C:\images. Then the examiner inputted the case number as '1' and examiner 'Jaspreet'. Once created the examiner added data by browsing the C:\images and clicked on the ntfs.dd file. Once this file is selected the examiner configures the ingest modules and selects all the things to search for when reviewing this ntfs.dd file. Then once the examiner selects all the things to search for and sees the results the examiner looks through the created table data on the file. Then the examiner sees that the NTFS file system includes the Master File Table ($MFT) which is used to help with file metadata in NTFS and the Master File Table Mirror ($MFTMirr) which is used to recover files and check integrity. This is important because these files are critical for the NTFS file system structure and for forensic analysis.



/img_ntfs.dd 28 Results

Name	Modified Time	Change Time
$LogFile	2014-01-02 23:17:58 EST	2014-01-02 23:17:
$MFT	2014-01-02 23:17:58 EST	2014-01-02 23:17:
$MFTMirr	2014-01-02 23:17:58 EST	2014-01-02 23:17:
$Secure:$SDS	2014-01-02 23:17:58 EST	2014-01-02 23:17:
$UpCase	2014-01-02 23:17:58 EST	2014-01-02 23:17:
$Volume	2014-01-02 23:17:58 EST	2014-01-02 23:17:

Analyzing files from ntfs.dd 12% 3

Figure 22: Autopsy File Partition

Conclusion

Overall, through this lab the examiner was able to analyze and understand the NTFS file system using the following forensic tools HxD and Autopsy as well as command prompts to examine the raw disk data, verify the data's integrity, and reveal any hidden or deleted information. Through this process the examiner was able to gain hands-on experience in understanding the structure of NTFS file system as well as exploring how partitions worked and identifying critical system files. By using the forensic tools, it ensured that the examiner followed a structured approach when conducting the forensic analysis on the data and ensured that the integrity of the data through the investigation remained in tact through the analysis of key NTFS features such as ADS, Timestomp manipulation, and $MFT.

NTFS File System

This is primarily used in windows operating systems to provide a secure and modern file system. This file system included advanced features such as ADS, ACLs, journaling, and file encryption, all of which is useful and impactful during a forensic investigation. THE NTFS file system also keeps a Master file Table, which stores all the metadata about all files and directories within it, making it a vital resource for digital analysis. By understanding the key features and structures of the NTFS file system the examiner can recover deleted content or files, track the modification of files, and detect any techniques or actions conducted by malicious characters during an investigation.

HxD Application

This is a hex editor application that allows the examiner to modify and or view raw disk image data on a byte level. This tool is essential to examine the Master Boot Record (MBR) of 512 bytes in length and helps to partition the table into sections to help the examiner locate and interpret partition structures. This application utilizes the switching of offsets from hexadecimal (h) to decimal (d) to make the partitions easier to compare with other forensic tools as well as make it easier to interpret and read the data. By understanding the key features of this hex editor application, it can play a critical role in tracing deleted files, recovering hidden data, and overall providing a forensic analysis of a disk image during an investigation.

Autopsy

Autopsy is an open-source forensic tool that is used to analyze NTFS disk images and extract any crucial artifacts within it. Through this the examiner was able to examine the disk image of the NTFS file system including the $MFT and $MFTMirr, which include stored metadata about every file on the system. This forensic tool allowed the examiner to recover any deleted files, detect possible techniques used by attackers to hide data, and inspect timestamps of files. This tool is crucial to a forensic investigation because it allows for more simplified and structured interfaces to browse a file system searching for key hidden artifacts and generating a report on the file system based on the selected prompts by an examiner.