

**Team Members: Aidan Fetter, Dylan Greene, Heath Baron, Alex Adrian, Dallas DeSimone**

Who collected the device, from whom, when, serial number of the device etc.?

Who collected the device?

Dylan Greene

Who was it collected from?

Ujan Mukhopadhyay

When was it collected?

October 31<sup>st</sup>

Serial number of the device?

DA55D802

What operations you performed, when, what results it yielded.

I first collected the device and connected it to my Kali Linux Virtual Machine. From there, the first operation I did was mount the drive to Kali and then perform an image command on the device. Running `df -h` yielded that the drive was under the `/dev/sdb1` directory within Kali. With this directory, I ran `dd if=/dev/sdb of=image.dd` to image the suspect USB device to my local machine. I also ran `lsusb -v` to obtain the serial number and extra information on the device whilst the imaging was in progress. After the imaging was completed, the device was unmounted and removed from the local machine. Utilizing the SleuthKit commands, I ran `mm1s image.dd` to obtain the Partition Table of the image and verify that the image was successful.

#### DOS Partition Table

Offset Sector: 0

Units are in 512-byte sectors

	Slot	Start	End	Length	Description
000:	Meta	0000000000	0000000000	0000000001	Primary Table (#0)
001:	-----	0000000000	0000009567	0000009568	Unallocated
002:	000:000	0000009568	0015974399	0015964832	Win95 FAT32 (0x0b)

I then ran the following commands to discover the existing and deleted files on the image. For the existing files, `fls -o 9568 image.dd`, and for the deleted files, `fls -o 9568 -Frd image.dd`. The results of both commands listed roughly 100+ files and directories within the image.

```

r/r * 139:      System Volume Information/_FS0.LOG
r/r * 5026565: {73F77E4E-5A17-46E5-A5FC-8A061047725F}v14.36.32532/_ackages/vcRuntimeMinimum_x86/_ab1.cab
r/r * 5026568: {73F77E4E-5A17-46E5-A5FC-8A061047725F}v14.36.32532/_ackages/vcRuntimeMinimum_x86/vc_runtimeMinimum_x86.msi
r/r * 19:       New Text Document (2).txt
r/r * 22:       _FS0.TMP
r/r * 1076869: {0caa97f1-4764-4ef2-a28f-908585b2609b}/_tate.rsm
r/r * 1076874: {0caa97f1-4764-4ef2-a28f-908585b2609b}/windowsdesktop-runtime-6.0.23-win-x86.exe
r/r * 1098377: {0D02D706-44F2-4957-A448-E7259A0B56B9}v40.68.31219/windowsdesktop-runtime-5.0.17-win-x86.msi
r/r * 1921413: {2BB7BEBF-308B-4A9D-B1E0-1BBE7C8F5EA4}v3.11.6150.0/_ib.msi
r/r * 2201477: {2CCD08A5-5FA3-4218-964E-6426FA3F28E8}v3.11.6150.0/_xe.msi
r/r * 2222853: {2FB71770-2C2E-42A3-9136-5101D1E930F4}v3.11.5150.0/_auncher.msi
r/r * 2240261: {5BC2F455-DDC6-468D-A7CE-2982DDAFBC77}v3.11.6150.0/_cltk.msi
r/r * 2351365: {6FAD6842-68E9-4804-887D-52762A67617D}v3.11.6150.0/_est.msi
r/r * 2476040: {7C0437DA-6703-47F1-A116-CD138B0768AD}v48.92.2594/dotnet-runtime-6.0.23-win-x64.msi
r/r * 3322373: {8bdfef669-9705-4184-9368-db9ce581e0e7}/_tate.rsm
r/r * 3322376: {8bdfef669-9705-4184-9368-db9ce581e0e7}/VC_redist.x64.exe
r/r * 3342981: {8fb37bcd-c3ab-4dc2-a7df-3d52ce16f512}/_tate.rsm
r/r * 3342984: {8fb37bcd-c3ab-4dc2-a7df-3d52ce16f512}/WindowsSensor.x64.exe
r/r * 3369989: {9dfff3540-fc85-4ed5-ac84-9e3c7fd8bece}/_tate.rsm
r/r * 3369992: {9dfff3540-fc85-4ed5-ac84-9e3c7fd8bece}/vcredist_x86.exe
r/r * 3384840: {9E82832F-8E44-4C36-B66B-051F3FFA24D7}v48.92.2594/dotnet-host-6.0.23-win-x86.msi
r/r * 3408266: {9F191CDC-6122-4376-A0CA-B98501C749AE}v8.10.0/VMware Horizon Media Optimization for Microsoft Teams (x64).msi
r/r * 3662981: {0025DD72-A959-45B5-A0A3-7EFEB15A8050}v14.36.32532/_ackages/vcRuntimeAdditional_amd64/_ab1.cab
r/r * 3662985: {0025DD72-A959-45B5-A0A3-7EFEB15A8050}v14.36.32532/_ackages/vcRuntimeAdditional_amd64/vc_runtimeAdditional_x64.msi
r/r * 3847173: {33d1fd90-4274-48a1-9bc1-97e33d9c2d6f}/_tate.rsm
r/r * 3847176: {33d1fd90-4274-48a1-9bc1-97e33d9c2d6f}/vcredist_x86.exe

```

**Figure 2 - Deleted Files List**

```

d/d 5: System Volume Information
r/r 8: Textbook 1.pdf
d/d * 13: {73F77E4E-5A17-46E5-A5FC-8A061047725F}v14.36.32532
r/r 16: New Text Document.txt
r/r * 19: New Text Document (2).txt
r/r 21: Autorun.inf
r/r * 22: _FS0.TMP
r/r 28: CIS 4203_Digital Forensics_FA23_Mukhopadhyay.docx.PFILE
r/r 30: 48DD6352E6FB
d/d * 34: {0caa97f1-4764-4ef2-a28f-908585b2609b}
d/d * 39: {0D02D706-44F2-4957-A448-E7259A0B56B9}v40.68.31219
d/d * 44: {2BB7BEBF-308B-4A9D-B1E0-1BBE7C8F5EA4}v3.11.6150.0
d/d * 49: {2CCD08A5-5FA3-4218-964E-6426FA3F28E8}v3.11.6150.0
d/d * 54: {2FB71770-2C2E-42A3-9136-5101D1E930F4}v3.11.5150.0
d/d * 59: {5BC2F455-DDC6-468D-A7CE-2982DDAFBC77}v3.11.6150.0
d/d * 64: {6FAD6842-68E9-4804-887D-52762A67617D}v3.11.6150.0
d/d * 69: {7C0437DA-6703-47F1-A116-CD138B0768AD}v48.92.2594
d/d * 73: {8bdfef669-9705-4184-9368-db9ce581e0e7}
d/d * 77: {8fb37bcd-c3ab-4dc2-a7df-3d52ce16f512}
d/d * 81: {9dfff3540-fc85-4ed5-ac84-9e3c7fd8bece}
d/d * 86: {9E82832F-8E44-4C36-B66B-051F3FFA24D7}v48.92.2594
d/d * 91: {9F191CDC-6122-4376-A0CA-B98501C749AE}v8.10.0

```

**Figure 1 - Directory Content List**

However, I came up with a solution while manually doing some of the first few extractions. This whole line of work is in Linux, of which Bash scripts exist that can take advantage of direct terminal interaction without much coding needed. I had devised a script that ended up being revised to consolidate the work and time to extract the files we wanted. Both scripts take input from the user for information on the image. They have the user input the image filename, offset number and inode(s) to check and/or extract. Utilizing both scripts, I extracted the first bulk of files successfully up to sector number 1077555. The remaining existing and deleted files were in more obscure sectors of the image, of which the second version of the script handled.

Next task I had was to extract these files from the image so we can check for any important information that may be dwelling in them. For this task, I utilized the `icat` command using a piped output to a file – command to see what datatype is detected. If nothing is found, we will skip the file; otherwise, I will extract it to the local machine for further inspection using the redirect ‘>’ to save the carved file to a local filename in the system. `icat -o 9568 image.dd | file - and icat -o 9568 image.dd > [inode#_filename]`. Knowing I would have to run the `icat` command at least 1 time to check if a file was empty and then potentially a 2<sup>nd</sup> time to extract it, made me realize this process will *take time & be tedious*.

The script is named Finder.sh, the below screenshot is the first version, which loops from a specified starting inode and ending inode. It extracts any file that isn't classified as "empty".

```
1 #!/bin/bash
2
3 # Finder.sh Ver 1.0
4 # Menu based version of the inodeExtractor.sh
5 # This is an older version, which makes you specify the info
6 # per usage (fixed in Ver 2.0)
7
8 menu() {
9     echo "Select an option:"
10    echo "1. Scan"
11    echo "2. Extract"
12    echo "3. Exit"
13 }
14
15 scan() {
16     read -p "Enter image file: " image_file
17     read -p "Enter offset: " offset
18     read -p "Enter output directory: " output_directory
19
20     if [ ! -f "$image_file" ]; then
21         echo "Image file not found."
22         return 1
23     fi
24
25     if [ ! -d "$output_directory" ]; then
26         echo "Output directory does not exist."
27         return 1
28     fi
29
30     read -p "Enter inode to scan: " inode
31
32     # Use icat to extract the file with the specified inode
33     icat -o "$offset" "$image_file" "$inode" | file -
34 }
35
36 extract() {
37     read -p "Enter image file: " image_file
38     read -p "Enter offset: " offset
39     read -p "Enter output directory: " output_directory
40
41     if [ ! -f "$image_file" ]; then
42         echo "Image file not found."
43         return 1
44     fi
45
46     if [ ! -d "$output_directory" ]; then
47         echo "Output directory does not exist."
48         return 1
49     fi
50
51     read -p "Enter inode to extract: " inode
52
53     # Use icat to extract the file with the specified
54     # inode and save it to the output directory
55     icat -o "$offset" "$image_file" "$inode" > "$output_directory/$inode"
56     echo "File extracted to: $output_directory/$inode"
57 }
58
59 # Main program
60 while true; do
61     menu
62     read -p "Enter your choice: " choice
63
64     case "$choice" in
65         1)
66             scan
67             ;;
68         2)
69             extract
70             ;;
71         3)
72             echo "Exiting the program."
73             exit 0
74             ;;
75         *)
76             echo "Invalid choice. Please try again."
77             ;;
78     esac
79 done
80
```

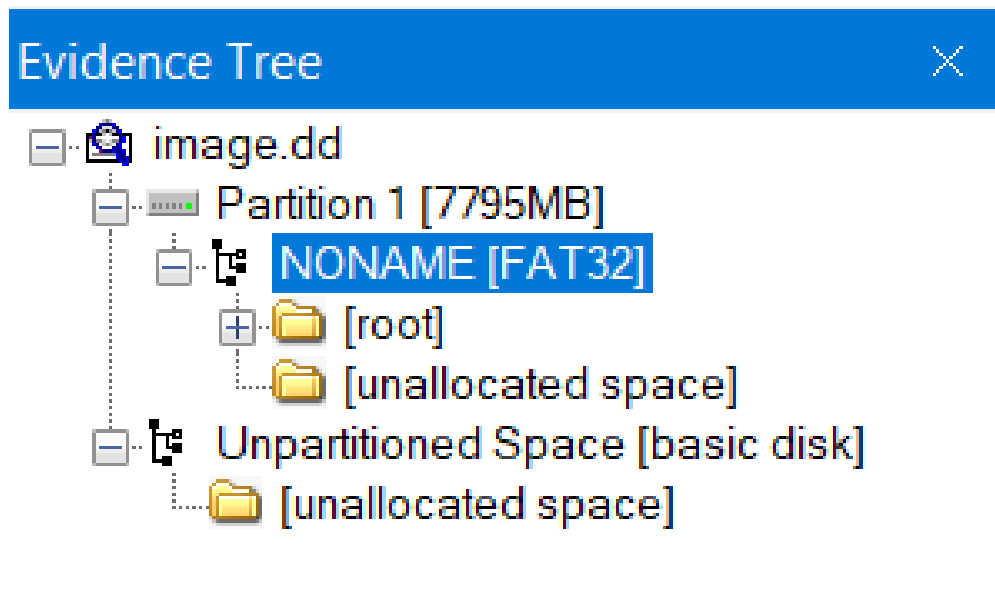
The reason this had to be revised to the new version is due to the nature that it is slow. With the first bulk run it took several hours to collect any files between inodes 0 to 1,077,555.

You will find below the screenshot of the second version, which was designed to take care of the extra inodes that the original would have taken possibly days to loop from start to finish of all 15,964,831 sectors of the 7.7GB image file. This version will take input at the call of the script where the user specifies the image filename, sector offset for the partition to be searched, and the output directory where the extracted files are stored. `./Finder.sh image.dd 9568 InodeOut`

```
1 #!/bin/bash
2
3 # Finder.sh Ver 2.0
4 # Updated and Current Finder.sh script
5 # Menu based inodeExtractor
6 # Improved workflow to take input arguments from command call
7 # * Main requirements are provided in arguments at runtime
8 # * inode # is provided with each scan or extract option
9
10 if [ $# -ne 3 ]; then
11     echo "Usage: $0 image_file offset output_directory"
12     exit 1
13 fi
14
15 image_file="$1"
16 offset="$2"
17 output_directory="$3"
18
19 # Check for valid "image_file" and valid "output_directory"
20 if [ ! -f "$image_file" ]; then
21     echo "Image file not found."
22     exit 1
23 fi
24
25 if [ ! -d "$output_directory" ]; then
26     echo "Output directory does not exist."
27     exit 1
28 fi
29
30 menu() {
31     echo "++++Main=Menu++++"
32     echo "||      Select an option:      ||"
33     echo "||      1. Scan                  ||"
34     echo "||      2. Extract               ||"
35     echo "||      3. Exit                 ||"
36     echo "++++"
37 }
38
39 scan() {
40     read -p "Enter inode to scan: " inode
41
42     # Use icat to extract the file with the specified inode
43     echo ""
44     echo "Result:"
45     icat -o "$offset" "$image_file" "$inode" | file -
46     echo ""
47 }
48
49 extract() {
50     read -p "Enter inode to extract: " inode
51
52     # Use icat to extract the file with the specified inode and save it
53     # to the output directory
54     echo ""
55     icat -o "$offset" "$image_file" "$inode" > "$output_directory/$inode"
56     echo "File extracted to: $output_directory/$inode"
57     echo ""
58 }
59
60 # Main program
61 while true; do
62     menu
63     read -p "Enter your choice: " choice
64
65     case "$choice" in
66         1)
67             scan
68             ;;
69         2)
70             extract
71             ;;
72         3)
73             echo "Exiting the program."
74             exit 0
75             ;;
76         *)
77             echo ""
78             echo "Invalid choice. Please try again."
79             echo ""
80             ;;
81     esac
82 done
```

Name of the files, condition in which it was found, contents.

The below image displays the output after placing the .dd file into FTK Imager:



NONAME is the name of the drive attached to the partition. The tree below shows the contents of the directories found:

[root]/

|— !FS0.TMP – Does not provide much information other than the fact that Wix was used on the Windows system

```

11a0 0D 00 00 00 57 00 69 00-78 00 42 00 75 00 6E 00  ...W-i-x-B-u-n-
11b0 64 00 6C 00 65 00 4E 00-61 00 6D 00 65 00 02 00 d-l-e-N-a-m-e-...
11c0 00 00 30 00 00 00 4D 00-69 00 63 00 72 00 6F 00  ...M-i-c-r-o-
11d0 73 00 6F 00 66 00 74 00-20 00 57 00 69 00 6E 00 s-o-f-t- W-i-n-
11e0 64 00 6F 00 77 00 73 00-20 00 44 00 65 00 73 00 d-o-w-s- D-e-s-
11f0 6B 00 74 00 6F 00 70 00-20 00 52 00 75 00 6E 00 k-t-o-p- R-u-n-
1200 74 00 69 00 6D 00 65 00-20 00 2D 00 20 00 36 00 t-i-m-e- --6-
1210 2E 00 30 00 2E 00 32 00-33 00 20 00 28 00 78 00 .0..2-3- -(x-
1220 38 00 36 00 29 00 01 00-00 00 01 00 00 00 17 00 8-6-) .....
1230 00 00 57 00 69 00 78 00-42 00 75 00 6E 00 64 00  ...W-i-x-B-u-n-d-
1240 6C 00 65 00 4F 00 72 00-69 00 67 00 69 00 6E 00 l-e-O-r-i-g-i-n-
1250 61 00 6C 00 53 00 6F 00-75 00 72 00 63 00 65 00 a-l-S-o-u-r-c-e-
1260 02 00 00 00 5A 00 00 00-43 00 3A 00 5C 00 57 00  ...Z--C-:\W-
1270 49 00 4E 00 44 00 4F 00-57 00 53 00 5C 00 53 00 I-N-D-O-W-S-\S-
1280 6F 00 66 00 74 00 77 00-61 00 72 00 65 00 44 00 o-f-t-w-a-r-e-D-
1290 69 00 73 00 74 00 72 00-69 00 62 00 75 00 74 00 i-s-t-r-i-b-u-t-
12a0 69 00 6F 00 6E 00 5C 00-44 00 6F 00 77 00 6E 00 i-o-n-\D-o-w-n-
12b0 6C 00 6F 00 61 00 64 00-5C 00 49 00 6E 00 73 00 l-o-a-d-\I-n-s-
12c0 74 00 61 00 6C 00 6C 00-5C 00 77 00 69 00 6E 00 t-a-l-l-\w-i-n-
12d0 64 00 6F 00 77 00 73 00-64 00 65 00 73 00 6B 00 d-o-w-s-d-e-s-k-
12e0 74 00 6F 00 70 00 2D 00-72 00 75 00 6E 00 74 00 t-o-p- --r-un-t-
12f0 69 00 6D 00 65 00 2D 00-36 00 2E 00 30 00 2E 00 i-m-e- --6-..0-..
1300 32 00 33 00 2D 00 77 00-69 00 6E 00 2D 00 78 00 2-3--w-i-n--x-
1310 38 00 36 00 2E 00 65 00-78 00 65 00 01 00 00 00 8-6-..e-x-e-...
1320 01 00 00 00 1D 00 00 00-57 00 69 00 78 00 42 00  ...W-i-x-B-
1330 75 00 6E 00 64 00 6C 00-65 00 4F 00 72 00 69 00 u-n-d-l-e-O-r-i-
1340 67 00 69 00 6E 00 61 00-6C 00 53 00 6F 00 75 00 g-i-n-a-l-S-o-u-
1350 72 00 63 00 65 00 46 00-6F 00 6C 00 64 00 65 00 r-c-e-F-o-l-d-e-
1360 72 00 02 00 00 00 31 00-00 00 43 00 3A 00 5C 00 r-----l--C-:\
1370 57 00 49 00 4E 00 44 00-4F 00 57 00 53 00 5C 00 W-I-N-D-O-W-S-\
1380 53 00 6F 00 66 00 74 00-77 00 61 00 72 00 65 00 S-o-f-t-w-a-r-e-
1390 44 00 69 00 73 00 74 00-72 00 69 00 62 00 75 00 D-i-s-t-r-i-b-u-
13a0 74 00 69 00 6F 00 6E 00-5C 00 44 00 6F 00 77 00 t-i-o-n-\D-o-w-
13b0 6E 00 6C 00 6F 00 61 00-64 00 5C 00 49 00 6E 00 n-l-o-a-d-\I-n-
13c0 73 00 74 00 61 00 6C 00-6C 00 5C 00 01 00 00 00 s-t-a-l-l-\.....

```

— 48DD6352E6FB – Contains the following text:

26542179D2DB1848E0362F06D3154A4F07D700BB

This does not have any significance based on our investigation.

— 5473B06CA73CC00C7D9577CF3A7FB5B07BD6D0CA – Empty Directory

— 7772AD4B5C741E4056069CDBBDBD50598B4141B9 – Empty Directory

— Autorun.inf – Contains the following:

```
[autorun]
open=wubi.exe --cdmenu
icon=wubi.exe,0
label=Install Ubuntu

[Content]
MusicFiles=false
PictureFiles=false
VideoFiles=false
[autorun]
open=setup.exe
icon=setup.exe,0
label=My install CD
shell\readme\command=notepad readme.txt
shell\readme=Read & Me
shell=readme
[ExclusiveContentPaths]
\pictures
\music
more music\special
[IgnoreContentPaths]
pictures
\music
more music\special
```

It looks like the file opens Ubuntu, runs a setup executable, opens up notepad, and displays the content of readme.txt

└─ BDD.LOG – Contains the following:

```
<![LOG[Property LogPath is now =
C:\MININT\SMSOSD\OSDLOGS]LOG]!><time="13:47:20.000+000" date="11-15-2021"
component="StartMBAMEncryption" context="" type="1" thread=""
file="StartMBAMEncryption">

<![LOG[Property AddRegFile is now =
AddMBAMRegEntries.reg]LOG]!><time="13:47:20.000+000" date="11-15-2021"
component="StartMBAMEncryption" context="" type="1" thread=""
file="StartMBAMEncryption">

<![LOG[Property RemoveRegFile is now =
RemoveMBAMRegEntries.reg]LOG]!><time="13:47:20.000+000" date="11-15-2021"
component="StartMBAMEncryption" context="" type="1" thread=""
file="StartMBAMEncryption">

<![LOG[Property WaitForEncryption is now = true]LOG]!><time="13:47:20.000+000"
date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread=""
file="StartMBAMEncryption">

<![LOG[Microsoft Deployment Toolkit version:
6.3.8330.1000]LOG]!><time="13:47:20.000+000" date="11-15-2021"
```

```
component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[Property Debug is now = FALSE]LOG]!><time="13:47:20.000+000" date="11-15-  
2021" component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[using C:\WINDOWS\syntax\reg.exe for Reg  
Import]LOG]!><time="13:47:21.000+000" date="11-15-2021"  
component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[About to run command: C:\WINDOWS\syntax\reg.exe IMPORT  
"C:\_SMSTaskSequence\Packages\LKL00200\AddMBAMRegEntries.reg"]LOG]!><time="13:4  
7:21.000+000" date="11-15-2021" component="StartMBAMEncryption" context=""  
type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Command has been started (process ID 3820)]LOG]!><time="13:47:21.000+000"  
date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[Return code from command = 0]LOG]!><time="13:47:21.000+000" date="11-15-  
2021" component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[Waiting for Encryption to Start]LOG]!><time="13:47:22.000+000" date="11-15-  
2021" component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[Encryption Started]LOG]!><time="13:47:37.000+000" date="11-15-2021"  
component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[About to run command: C:\WINDOWS\syntax\reg.exe IMPORT  
"C:\_SMSTaskSequence\Packages\LKL00200\RemoveMBAMRegEntries.reg"]LOG]!><time="1  
3:47:37.000+000" date="11-15-2021" component="StartMBAMEncryption" context=""  
type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Command has been started (process ID 6588)]LOG]!><time="13:47:37.000+000"  
date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[Return code from command = 0]LOG]!><time="13:47:37.000+000" date="11-15-  
2021" component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 2% Complete.]LOG]!><time="13:47:37.000+000" date="11-15-  
2021" component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```



```
<![LOG[Encrypting C: drive 7% Complete.]LOG]!><time="13:48:37.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 11% Complete.]LOG]!><time="13:49:38.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 17% Complete.]LOG]!><time="13:50:38.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 24% Complete.]LOG]!><time="13:51:38.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 31% Complete.]LOG]!><time="13:52:38.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 38% Complete.]LOG]!><time="13:53:38.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 44% Complete.]LOG]!><time="13:54:38.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 49% Complete.]LOG]!><time="13:55:38.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 55% Complete.]LOG]!><time="13:56:38.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 61% Complete.]LOG]!><time="13:57:39.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 68% Complete.]LOG]!><time="13:58:39.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 74% Complete.]LOG]!><time="13:59:39.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 81% Complete.]LOG]!><time="14:00:39.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 87% Complete.]LOG]!><time="14:01:39.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 94% Complete.]LOG]!><time="14:02:39.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 100% Complete.]LOG]!><time="14:03:39.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[StartMBAMEncryption processing completed successfully.]LOG]!><time="14:04:39.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

It looks like this file is showing the encryption process for the C: drive on the OS. Through reading the logs, it is clear that Microsoft BitLocker Administration and Management (MBAM) Encryption is being used.

└─ CFD2B7732BB65D5356414B4D2C037465C4520711 – Empty Directory

└─ CIS 4203\_Digital Forensics\_FA23\_Mukhopadhyay.docx.PFILE – This file contained a lot of random symbols and random text. However, it also contained two pieces of text that could be of use: [umukhopadhyay@floridapoly.edu](mailto:umukhopadhyay@floridapoly.edu) and [administrator@FLORIDAPOLY](mailto:administrator@FLORIDAPOLY). We can also see that Microsoft Enhanced Cryptographic Provider v1.0 was used.

```
R|dp
sE(M
6229962695ea56c9a94aa6817c5950e3_451ab7ed-e83d-4c6a-9f7e-825e83d05384
Microsoft Enhanced Cryptographic Provider v1.0
Ujan Mukhopadhyay(umukhopadhyay@floridapoly.edu)
7VCC
G-\|
0J8[
I4uR4$
R|dp
administrator(administrator@FLORIDAPOLY)
BG 0
)MDB
Ax#<
```

— DES Key.txt – Does not contain any content. However, this could be very important to decrypt a potential DES encryption used on the drive. We tried using John the Ripper and Hashcat to see if we could brute force, but it didn't come up with anything.

— New Text Document (2).txt – Empty File

— New Text Document (2).txt.copy0 – Empty File

— New Text Document.txt – Contains the following:

```
hEcBxRXwEVeHYjgmQzCxvYPDRMiTh/GzbdNZ+TJr4OkkclQLa1gvBzJnphiJ3PYWPBBFFN2nNP  
a+dmhvRi++AhTae/r1E6dbNyXNvzj+sAy/cRle2Dt1Sw==
```

Originally, we thought this was encode using base64. However, since “+” and “/” is contained within the string, this cannot be the case. After researching potential encodings and hash look-alikes, we were still unable to determine what this says in plaintext.

— StartMBAMEncryption.log – Contains the following:

```
<![LOG[Property LogPath is now =  
C:\MININT\SMSOSD\OSDLOGS]LOG]!><time="13:47:20.000+000" date="11-15-2021"  
component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">  
  
<![LOG[Property AddRegFile is now =  
AddMBAMRegEntries.reg]LOG]!><time="13:47:20.000+000" date="11-15-2021"  
component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">  
  
<![LOG[Property RemoveRegFile is now =  
RemoveMBAMRegEntries.reg]LOG]!><time="13:47:20.000+000" date="11-15-2021"  
component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">  
  
<![LOG[Property WaitForEncryption is now = true]LOG]!><time="13:47:20.000+000"  
date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">  
  
<![LOG[Microsoft Deployment Toolkit version:  
6.3.8330.1000]LOG]!><time="13:47:20.000+000" date="11-15-2021"  
component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">  
  
<![LOG[Property Debug is now = FALSE]LOG]!><time="13:47:20.000+000" date="11-15-  
2021" component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">  
  
<![LOG[using C:\WINDOWS\sysnative\reg.exe for Reg  
Import]LOG]!><time="13:47:21.000+000" date="11-15-2021"
```

```
component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[About to run command: C:\WINDOWS\syste\native\reg.exe IMPORT  
"C:\_SMSTaskSequence\Packages\LKL00200\AddMBAMRegEntries.reg"]LOG]!><time="13:4  
7:21.000+000" date="11-15-2021" component="StartMBAMEncryption" context=""  
type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Command has been started (process ID 3820)]LOG]!><time="13:47:21.000+000"  
date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[Return code from command = 0]LOG]!><time="13:47:21.000+000" date="11-15-  
2021" component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[Waiting for Encryption to Start]LOG]!><time="13:47:22.000+000" date="11-15-  
2021" component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[Encryption Started]LOG]!><time="13:47:37.000+000" date="11-15-2021"  
component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[About to run command: C:\WINDOWS\syste\native\reg.exe IMPORT  
"C:\_SMSTaskSequence\Packages\LKL00200\RemoveMBAMRegEntries.reg"]LOG]!><time="1  
3:47:37.000+000" date="11-15-2021" component="StartMBAMEncryption" context=""  
type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Command has been started (process ID 6588)]LOG]!><time="13:47:37.000+000"  
date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[Return code from command = 0]LOG]!><time="13:47:37.000+000" date="11-15-  
2021" component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 2% Complete.]LOG]!><time="13:47:37.000+000" date="11-15-  
2021" component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 7% Complete.]LOG]!><time="13:48:37.000+000" date="11-15-  
2021" component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 11% Complete.]LOG]!><time="13:49:38.000+000" date="11-  
15-2021" component="StartMBAMEncryption" context="" type="1" thread=""  
file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 17% Complete.]LOG]!><time="13:50:38.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 24% Complete.]LOG]!><time="13:51:38.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 31% Complete.]LOG]!><time="13:52:38.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 38% Complete.]LOG]!><time="13:53:38.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 44% Complete.]LOG]!><time="13:54:38.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 49% Complete.]LOG]!><time="13:55:38.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 55% Complete.]LOG]!><time="13:56:38.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 61% Complete.]LOG]!><time="13:57:39.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 68% Complete.]LOG]!><time="13:58:39.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 74% Complete.]LOG]!><time="13:59:39.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 81% Complete.]LOG]!><time="14:00:39.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 87% Complete.]LOG]!><time="14:01:39.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 94% Complete.]LOG]!><time="14:02:39.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[Encrypting C: drive 100% Complete.]LOG]!><time="14:03:39.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

```
<![LOG[StartMBAMEncryption processing completed successfully.]LOG]!><time="14:04:39.000+000" date="11-15-2021" component="StartMBAMEncryption" context="" type="1" thread="" file="StartMBAMEncryption">
```

The file contains logs regarding the way the C: drive was encrypted. The logs show that the register modules AddMBAMRegEntries.reg and RemoveMBAMRegEntries.reg were imported and Microsoft Deployment Toolkit version 6.3.8330.1000 was used to start the encryption process. It can also be said that the process was successful as indicated by the last log entry.

└─ System Volume Information - [Directory](#)

└─ !FS0.LOG – [Empty Directory](#)

└─ IndexerVolumeGuid – [Contains the following:](#)

```
{ 6 1 7 E 4 9 C 6 - 6 E B C - 4 4 B 0 - 9 9 6 E - 6 8 E F 9 9 6 D 0 3 1 D }
```

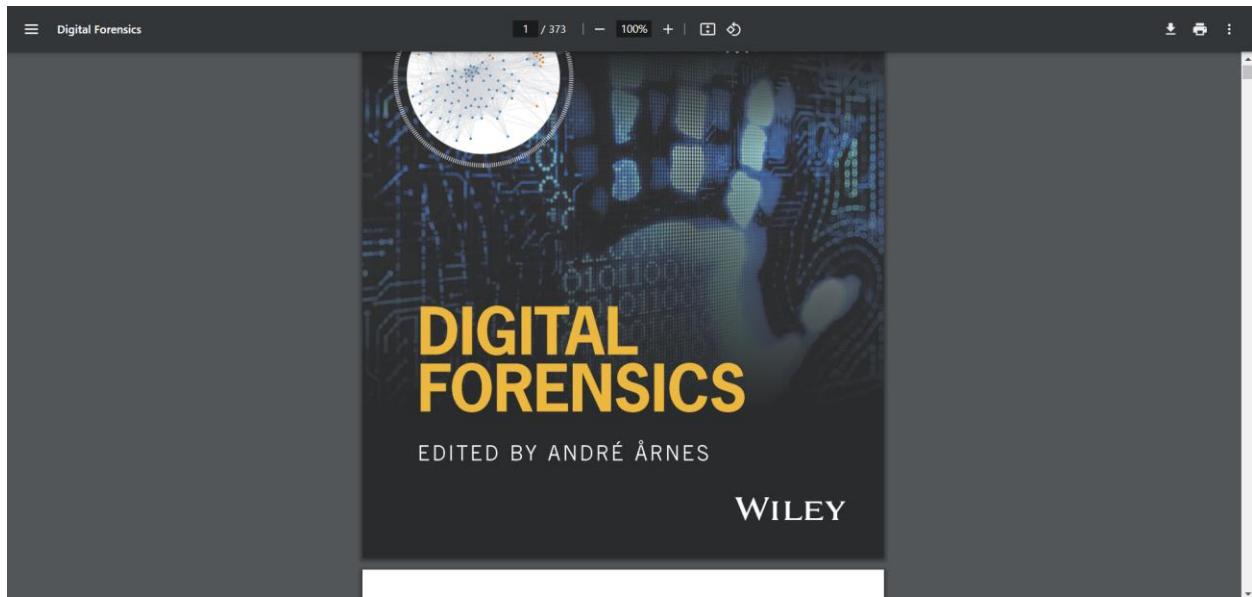
This does not have any significance based on our investigation.

└─ WPSettings.dat – [Contains the following:](#)

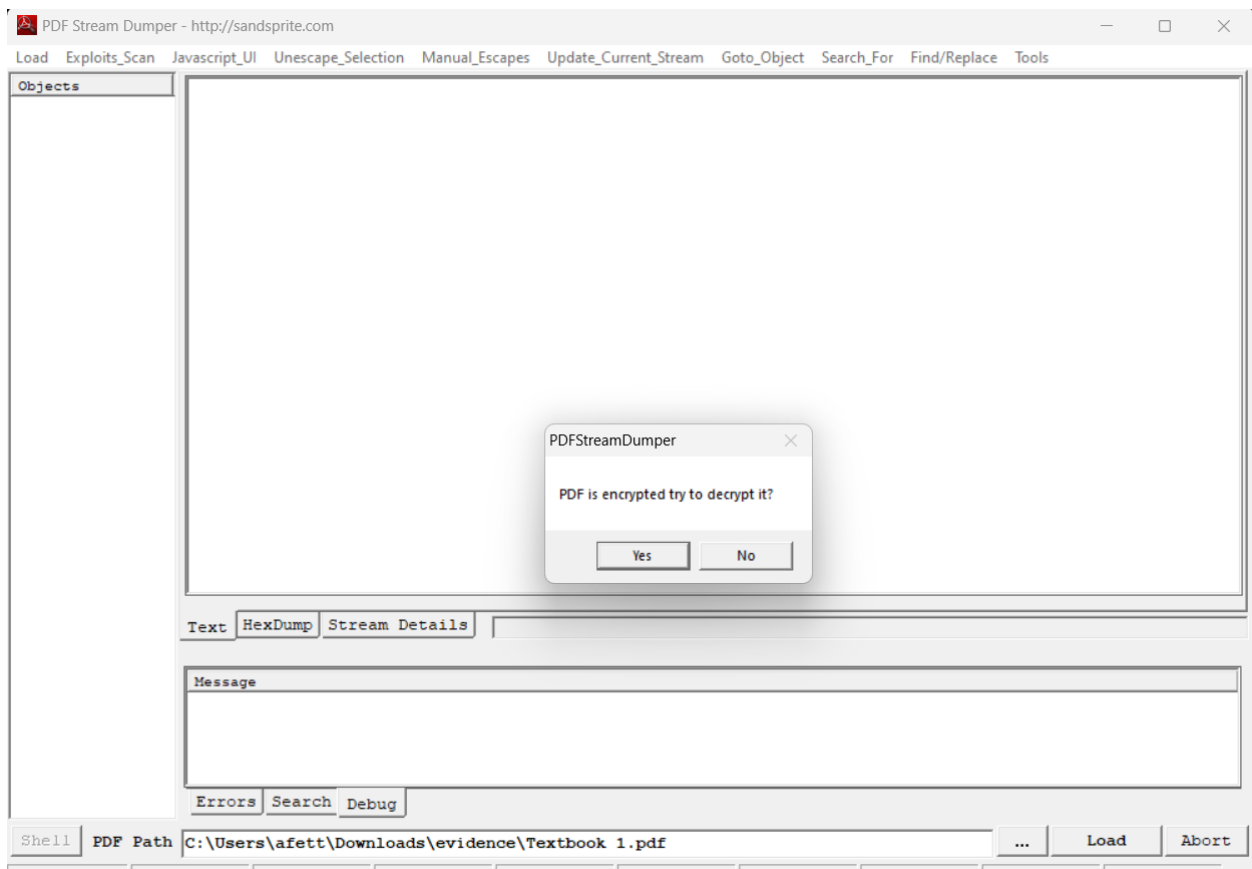
```
^Û'³!“
```

This does not have any significance based on our investigation.

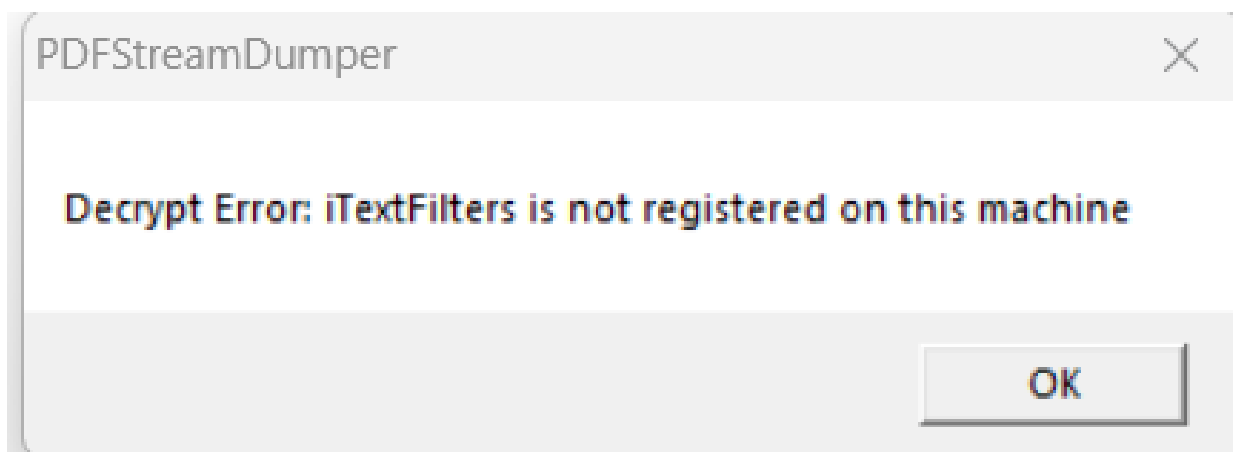
└─ Textbook 1.pdf – [Contains a PDF of the textbook for the class:](#)



When putting the file into PDF Stream Dumper, we got the following result:

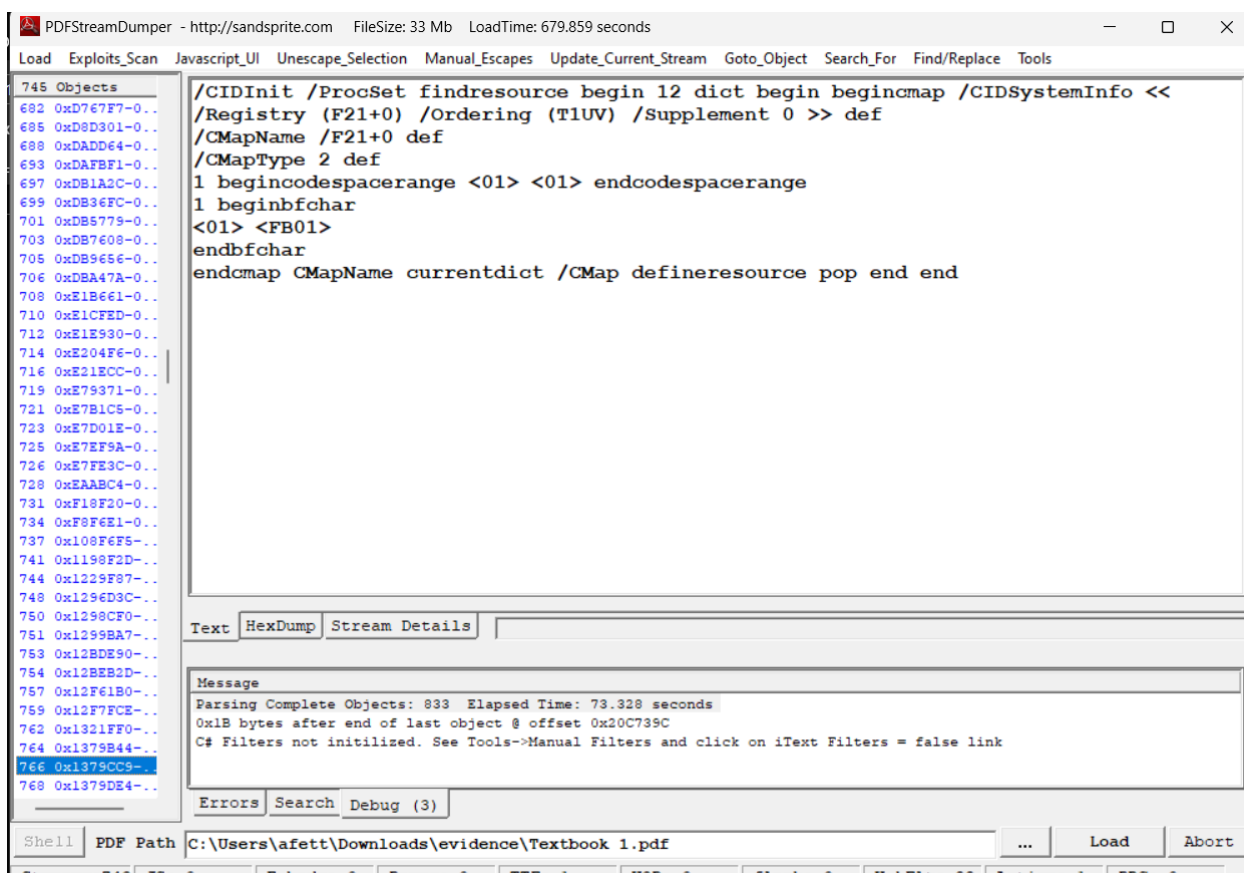


Based on the above screenshot, we figured out that the PDF has been changed and encrypted. While this looks promising, the application displays the following after pressing “Yes”:



iText is a developer tool used to create and manipulate PDF files in Java and .NET. This gives us an insight as to how the PDF was changed. Although this is a good lead, iText needs to be purchased in order to install it; therefore, we decided to move on and dig deeper.

No relevant information was found in the hex stream. However, it did further confirm that the pdf content has been encoded.



Since we did not receive any additional information from the dump, we decided to go with another method.



1. The hash of the textbook from the disk image is different from the hash of our original copy of the textbook. This could mean nothing since any small thing could have changed the hash, but it could also mean that there is something hidden in the PDF.
2. We took several actions to try and uncover anything hidden within the PDF file. We first checked if there was any interesting exif data using exiftool and didn't find anything interesting. We then tried some file carving tools like binwalk and foremost to see if there were any embedded files. The output of these tools didn't show anything of interest to our case.

└─ VARIABLES.DAT - Contains the following:

```
<?xml version="1.0"?>
```

```
<MediaVarList Version="4.00.5345.0000"><var  
name="ADDREGFILE"><![CDATA[AddMBAMRegEntries.reg]]></var><var  
name="LOGPATH"><![CDATA[C:\MININT\SMSOSD\OSDLOGS]]></var><var  
name="REMOVEREGFILE"><![CDATA[RemoveMBAMRegEntries.reg]]></var><var  
name="WAITFORENCRYPTION"><![CDATA[true]]></var><var  
name="DEBUG"><![CDATA[FALSE]]></var></MediaVarList>
```

1. The XML declaration `<?xml version="1.0"?>` indicates that this is an XML document with version 1.0.
2. The root element is `<MediaVarList>` with an attribute `Version="4.00.5345.0000"`.
3. Within the `<MediaVarList>` element, there are several `<var>` elements, each representing a variable with a name and a value. The values are enclosed in CDATA sections (`<![CDATA[...]]>`), which is used to include characters that would otherwise be treated as markup.
  - The first `<var>` element has a name of "ADDREGFILE" and a value of "AddMBAMRegEntries.reg".
  - The second `<var>` element has a name of "LOGPATH" and a value of "C:\MININT\SMSOSD\OSDLOGS".
  - The third `<var>` element has a name of "REMOVEREGFILE" and a value of "RemoveMBAMRegEntries.reg".
  - The fourth `<var>` element has a name of "WAITFORENCRYPTION" and a value of "true".
  - The fifth `<var>` element has a name of "DEBUG" and a value of "FALSE".

└─ VARIABLES.DAT.LOCK – Empty File

└─ adtz\_globals.dll – Empty File

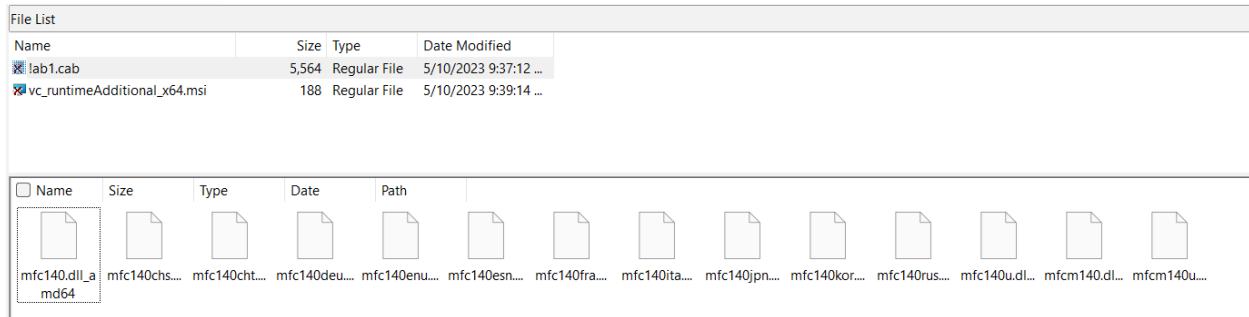
└─ cislog.txt – Empty File

└─ {0025DD72-A959-45B5-A0A3-7EFEB15A8050}\v14.36.32532 - Directory

└─ Packages - Directory

└─ vcRuntimeAdditional\_amd64 - Directory

└─ lab1.cab – This cabinet file contains 14 .dll\_amd64 files




A Dynamic Link Library (DLL) file contains code and data that can be used by multiple programs on the operating system at once. Although some of the files contained some plaintext in them that didn't have much meaning, the majority of the files contained random letters and symbols.




└─ vc\_runtimeAdditional\_x64.msi – Contains a Windows Installer file






▼ Earlier this year

	<b>vc_runtimeAdditional_x64.msi</b>	5/10/2023 5:39 AM	Windows Installer ...	188 KB
<div> Item type: Windows Installer Package  Authors: Microsoft Corporation  Title: Installation Database  Subject: Visual C++ 2022 X64 Additional Runtime  This installer database contains the logic and data required to install Microsoft Visual C++ 2022 X64 Additional Runtime - 14.  Date modified: 5/10/2023 5:39 AM  Size: 188 KB  Path: vc_runtimeAdditional_x64 (C:\Users\afett\Downloads) </div>				

Based on this, we concluded that this directory did not add any relevant information to the investigation as it's only contains installation data for Microsoft Visual Studio 2022 X64.

- └─ {010792BA-551A-3AC0-A7EF-0FAB4156C382}v12.0.40664 - [Directory](#)
  - └─ !ackages – [Directory](#)
    - └─ vcRuntimeAdditional\_amd64 - [Directory](#)
      - └─ !ab1.cab – This cabinet file is similar to an earlier file in which several files are embedded within it. Instead of dll\_amd64 files, however, this file contains 14 .mfc120\_x64 files.


File List				
Name	Size	Type	Date Modified	
 !ab1.cab	5,458	Regular File	5/24/2017 5:07:46 ...	
 !ab1.cab.FileSlack	3	File Slack		
 vc_runtimeAdditional_x64.msi	140	Regular File	5/24/2017 5:09:00 ...	

<input type="checkbox"/> Name	Size	Type	Date	Path
 F_CENTRAL_mfc120_x64				
F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...
F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...
F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...
F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...
F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...
F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...
F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...
F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...
F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...
F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...
F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...
F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...	F_CENTRA...

None of the files had any significance based on our investigation.

- └─ vc\_runtimeAdditional\_x64.msi – [Windows installer for](#)

▼ A long time ago

 <b>vc_runtimeAdditional_x64.msi</b>	5/24/2017 1:09 PM	Windows Installer ...	140 KB
---	-------------------	-----------------------	--------

Item type: Windows Installer Package

Authors: Microsoft Corporation

Title: Installation Database

Subject: Visual C++ 2013 x64 Additional Runtime

This installer database contains the logic and data required to install Microsoft Visual C++ 2013 x64 Additional Runtime - 12.

Date modified: 5/24/2017 1:09 PM

Size: 140 KB

Path: vc\_runtimeAdditional\_x64 (C:\Users\afett\Downloads)

Based on this, we concluded that this directory did not add any relevant information to the investigation as it's only contains extra installation data for Microsoft Visual Studio 2022 X64.

└─ {042d26ef-3dbe-4c25-95d3-4c1b11b235a7} - Directory

| └─ !tate.rsm – Contains the following:

```


File Edit View
B
WixBundleForcedRestartPackage WixBundleLastUsedSource
WixBundleName < Microsoft Visual C++ 2013 Redistributable (x64) - 12.0.40664 WixBundleOrig
inalSource m C:\Program Files (x86)\Common Files\Adobe\Adobe Desktop Common\Runtime\customhook

```

This does not have any significance based on our investigation.

| └─ vcredist\_x64.exe – Contains the following executable:

▼ A long time ago

 <b>vcredist_x64.exe</b>	11/22/2021 1:00 PM	Application	456 KB
---	--------------------	-------------	--------

File description: Microsoft Visual C++ 2013 Redistributable (x64) - 12.0.40664

Company: Microsoft Corporation

File version: 12.0.40664.0

Date created: 10/26/2023 8:48 AM

Size: 455 KB

Path: vcredist\_x64 (C:\Users\afett\Downloads)

Based on this, we concluded that this directory did not add any relevant information to the investigation as it's only contains data pertaining to Visual Studio C++ 2013.

**The rest of the directories/files listed below did not contain anything significant to the investigation:**

└─ {098c6ff7-1af1-4c4a-b86f-c60608c98e31} - Directory

| └─ !tate.rsm

- | └─ windowsdesktop-runtime-5.0.17-win-x86.exe
- | └─ {0D02D706-44F2-4957-A448-E7259A0B56B9}v40.68.31219 - [Directory](#)
- | └─ windowsdesktop-runtime-5.0.17-win-x86.msi
- | └─ {0caa97f1-4764-4ef2-a28f-908585b2609b} - [Directory](#)
- | └─ !tate.rsm
- | └─ windowsdesktop-runtime-6.0.23-win-x86.exe
- | └─ {1870DD0E-1583-44FF-8265-A9D1692CD89C}v48.92.2594 - [Directory](#)
- | └─ dotnet-host-6.0.23-win-x64.msi
- | └─ {2BB7BEBF-308B-4A9D-B1E0-1BBE7C8F5EA4}v3.11.6150.0 - [Directory](#)
- | └─ !ib.msi
- | └─ {2CCD08A5-5FA3-4218-964E-6426FA3F28E8}v3.11.6150.0 - [Directory](#)
- | └─ !xe.msi
- | └─ {2FB71770-2C2E-42A3-9136-5101D1E930F4}v3.11.5150.0 - [Directory](#)
- | └─ !launcher.msi
- | └─ {33d1fd90-4274-48a1-9bc1-97e33d9c2d6f} - [Directory](#)
- | └─ !tate.rsm
- | └─ vcredist\_x86.exe
- | └─ {37B8F9C7-03FB-3253-8781-2517C99D7C00}v11.0.61030 - [Directory](#)
- | └─ !ackages - [Directory](#)
- | └─ vcRuntimeAdditional\_amd64 - [Directory](#)
- | └─ !ab1.cab
- | └─ vc\_runtimeAdditional\_x64.msi
- | └─ {410c0ee1-00bb-41b6-9772-e12c2828b02f} - [Directory](#)
- | └─ !tate.rsm
- | └─ VC\_redist.x86.exe
- | └─ {41C15E27-311D-46D2-B507-42C4F4C2EA0D}v48.92.2594 - [Directory](#)
- | └─ windowsdesktop-runtime-6.0.23-win-x86.msi
- | └─ {50825ABC-26FD-4260-AAC1-A5F40C4A6CE3}v48.92.2594 - [Directory](#)

- | └─ dotnet-hostfxr-6.0.23-win-x86.msi
- | └─ {50908576-1AF3-495B-82CE-C390DE362701}v6.32.14651.0 - [Directory](#)
- | └─ CsFirmwareAnalysis.msi
- | └─ {53CF6934-A98D-3D84-9146-FC4EDF3D5641}v12.0.40664 - [Directory](#)
- | └─ !ackages - [Directory](#)
- |     └─ vcRuntimeMinimum\_amd64
- |         └─ !ab1.cab
- |         └─ vc\_runtimeMinimum\_x64.msi
- | └─ {5420831A-BFDB-4331-9A54-EEF5EA52F0D4}v6.55.16950.0 - [Directory](#)
- | └─ CsDeviceControl.msi
- | └─ {54DE7EA9-E391-4BD2-A373-3A72A18EBDB5}v40.68.31213 - [Directory](#)
- | └─ dotnet-host-5.0.17-win-x86.msi
- | └─ {59650A2A-3839-46EC-9D9C-6B3B1C743C55}v40.68.31213 - [Directory](#)
- | └─ dotnet-runtime-5.0.17-win-x86.msi
- | └─ {5BC2F455-DDC6-468D-A7CE-2982DDAFBC77}v3.11.6150.0 - [Directory](#)
- | └─ !cltk.msi
- | └─ {65D1D415-E699-4A82-8D6D-F8B7488D9954}v3.11.6150.0 - [Directory](#)
- | └─ !ore.msi
- | └─ {6FAD6842-68E9-4804-887D-52762A67617D}v3.11.6150.0 - [Directory](#)
- | └─ !est.msi
- | └─ {73F77E4E-5A17-46E5-A5FC-8A061047725F}v14.36.32532 - [Directory](#)
- | └─ !ackages - [Directory](#)
- |     └─ vcRuntimeMinimum\_x86 - [Directory](#)
- |         └─ !ab1.cab
- |         └─ vc\_runtimeMinimum\_x86.msi
- | └─ {774C54C9-575B-4611-81C5-06466534F750}v3.11.6150.0 - [Directory](#)
- | └─ !ip.msi
- | └─ {7C0437DA-6703-47F1-A116-CD138B0768AD}v48.92.2594 - [Directory](#)

- | └─ dotnet-runtime-6.0.23-win-x64.msi
- | └─ {8122DAB1-ED4D-3676-BB0A-CA368196543E}v12.0.40664 - [Directory](#)
- | └─ !ackages - [Directory](#)
- | └─ vcRuntimeMinimum\_x86 - [Directory](#)
- | └─ !abl.cab
- | └─ vc\_runtimeMinimum\_x86.msi
- | └─ {8bdfc669-9705-4184-9368-db9ce581e0e7} - [Directory](#)
- | └─ !tate.rsm
- | └─ VC\_redist.x64.exe
- | └─ {8fb37bcd-c3ab-4dc2-a7df-3d52ce16f512} - [Directory](#)
- | └─ !tate.rsm
- | └─ WindowsSensor.x64.exe
- | └─ {995CC82C-E3E8-4BB5-9AB8-2B95C611D59D}v48.92.2594 - [Directory](#)
- | └─ dotnet-hostfxr-6.0.23-win-x64.msi
- | └─ {9E82832F-8E44-4C36-B66B-051F3FFA24D7}v48.92.2594 - [Directory](#)
- | └─ dotnet-host-6.0.23-win-x86.msi
- | └─ {9F191CDC-6122-4376-A0CA-B98501C749AE}v8.10.0 - [Directory](#)
- | └─ VMware Horizon Media Optimization for Microsoft Teams (x64).msi
- | └─ {9dff3540-fc85-4ed5-ac84-9e3c7fd8bece} - [Directory](#)
- | └─ !tate.rsm
- | └─ vcredist\_x86.exe
- | └─ {A4533B93-1B02-49BD-9231-AB892B620514}v8.10.0.37628 - [Directory](#)
- | └─ Components.cab
- | └─ Core.cab
- | └─ RMKSComponents.cab
- | └─ VMware Horizon View Client (x64).msi
- | └─ {A5731924-4225-4B73-BEEB-4A575133E8BB}v3.11.6150.0 - [Directory](#)
- | └─ !oc.msi

- |— {AA393199-374C-4AD1-9245-6CBB254D8146}v48.92.2594 – [Empty Directory](#)
- |— {AF01038B-6523-4EA7-9D9E-4F1E2927D88B}v40.68.31213 – [Empty Directory](#)
- |— {B175520C-86A2-35A7-8619-86DC379688B9}v11.0.61030 – [Empty Directory](#)
- |— {B1F7D579-C5B4-4859-9A53-BE40E5AAC4A7}v3.11.6150.0 – [Empty Directory](#)
- |— {BD95A8CD-1D9F-35AD-981A-3E7925026EBB}v11.0.61030 – [Empty Directory](#)
- |— {C2C59CAB-8766-4ABD-A8EF-1151A36C41E5}v14.36.32532 – [Empty Directory](#)
- |— {CF2BEA3C-26EA-32F8-AA9B-331F7E34BA97}v11.0.61030 – [Empty Directory](#)
- |— {D401961D-3A20-3AC7-943B-6139D5BD490A}v12.0.40664 – [Empty Directory](#)
- |— {D5D19E2F-7189-42FE-8103-92CD1FA457C2}v14.36.32532 – [Empty Directory](#)
- |— {DDC730D4-A94C-4C97-89D6-B7F532413A73}v3.11.6150.0 – [Empty Directory](#)
- |— {E90CEEAE-E798-4E92-8E58-E4543F641DF7}v48.92.2594 – [Empty Directory](#)
- |— {EA60D78B-BB3E-44EB-94B7-AF50586E79D9}v3.11.6150.0 – [Empty Directory](#)
- |— {F415A60E-479F-46DF-B76C-66F080FE7204}v8.10.0 – [Directory](#)
- |   └— VMware Horizon HTML5 MMR (x64).msi
- |— {F904FE12-B193-44E0-949C-D8084D051B21}v7.03.17506.0 – [Empty Directory](#)
- |— {ca67548a-5ebe-413a-b50c-4b9ceb6d66c6} – [Empty Directory](#)
- |— {f8e8ac13-7063-40e6-81dd-7ddcc3781ecd} – [Empty Directory](#)
- |— {ff117c0e-a559-4b5e-a773-93c94ec4ac73} – [Empty Directory](#)

## Screenshots of the operations

The below images show the analysis tree of the .dd file when opened in Autopsy:



