



CyberSleuth Solutions Ltd. Digital Forensic Investigation Report

| | |
|-------------------------|----------------|
| Investigators Full name | Ahmed Rashwan |
| Investigators ID | 12842713 |
| Case Number | CASE-2024-0008 |
| Version | 1.0 |

Table of Contents

| | |
|---|-----------|
| TABLE OF CONTENTS | 2 |
| TABLE OF FIGURES | 3 |
| INTRODUCTION | 4 |
| DIGITAL FORENSIC METHODOLOGY | 4 |
| DATA COLLECTION | 4 |
| EXAMINATION | 5 |
| <i>Tools Used</i> | 5 |
| <i>Findings</i> | 5 |
| List of Users | 5 |
| Suspicious Evidence | 5 |
| Obtaining Passwords for Password Protected Files | 6 |
| INSTALL_AIM[1].exe..... | 10 |
| <i>Analysis</i> | 13 |
| Credit Card Theft..... | 13 |
| <i>Drug Manufacturing</i> | 13 |
| <i>Theft of Pharmaceutical Drugs</i> | 13 |
| Check Washing | 14 |
| REPORTING | 14 |
| <i>Likelihood of each crime</i> | 14 |
| <i>Timeline</i> | 14 |
| APPLICABLE REGULATIONS..... | 16 |
| LEGAL AND ETHICAL CONSIDERATIONS | 16 |
| CONCLUSION | 16 |
| REFERENCES | 17 |
| APPENDIX A (CHAIN OF CUSTODY)..... | 18 |
| APPENDIX B (AUTHORIZATION LETTER) | 19 |
| APPENDIX C (SCREENSHOTS OF CREDIT CARD THEFT) | 20 |
| APPENDIX D (SCREENSHOTS OF DRUG MANUFACTURING) | 22 |
| APPENDIX E (SCREENSHOTS OF PHARMACEUTICAL THEFT) | 25 |
| APPENDIX F (SCREENSHOTS OF CHECK WASHING) | 26 |
| APPENDIX G (CASE NOTES)..... | 26 |

Table of Figures

| | |
|---|----|
| Image 1: Static Analysis by hybrid-analysis.com | 10 |
| Image 2 Chain of custody: | 18 |
| Image 3: Authorization letter..... | 19 |
| Image 4: Credit Card theft Evidence 1 | 20 |
| Image 5: Credit Card theft Evidence 2 | 20 |
| Image 6: Credit Card theft Evidence 3 | 20 |
| Image 7: Credit Card theft Evidence 4 | 20 |
| Image 8: Credit Card theft Evidence 5 | 21 |
| Image 9: Credit Card theft Evidence 6 | 21 |
| Image 10: Credit Card theft Evidence 7 | 22 |
| Image 11: Credit Card theft Evidence 9 | 22 |
| Image 12: Discussion of cooking drugs (Meth) | 22 |
| Image 13: Drug Manufacturing Evidence | 23 |
| Image 14: Location of camping/Drug cooking | 24 |
| Image 15: Drug Theft From Pharmacy Evidence | 25 |
| Image 16: Check Washing Evidence..... | 26 |
| Image 17: Check Washing Evidence 2 | 26 |

Introduction

This is a comprehensive digital forensic report conducted on a given USB that was suspected to have suspicious evidence. This report has been carried out and followed in accordance with the authorization letter issued by Oliver Mitchell, CEO of CyberSleuth Solutions Ltd as well as the supervision of Sarah Johnson and strictly follows the guidelines and procedures of the NIST (National Institute of Standards and Technology) 800-86 titled **Guide to Integrating Forensic Techniques into Incident Response**.

Digital Forensic Methodology

Data Collection

This is the first stage of the digital forensic investigation and report as part of the **NIST 800-86 guidelines**, this stage involves going through a systematic and fixed methodology of data collection in a way that maintains confidentiality and integrity as well as documenting a chain of custody form throughout the process of collection.

As per the authorization letter that CyberSleuth Solutions had provided, Sara Johnson has provided me with the evidence given to us by the client where I am given permission to acquire and collect the evidence which is a **Yellow INTEGRAL USB device**.

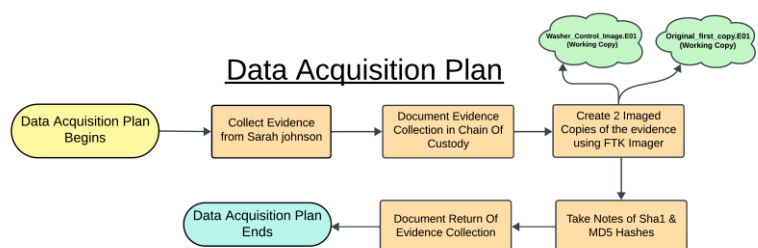
Following this, 2 imaged copies of the evidence were created, one being the working copy and one being the control copy. **The control copy is not** fragmented or compressed to make sure I have the purest form of the evidence as my control copy and **the working copy** has been compressed to create a smaller image and save on storage space, as throughout the investigation, the E01 file had to be moved to the cloud to not only maintain integrity of the 2 images but also to be later transferred to a Kali Linux virtual machine where further investigation was carried out. An E01 filetype was chosen for its many advantages for example, it is widely compatible with many digital forensic tools and has built-in data integrity features which makes it a better option for maintaining integrity for a digital forensic investigation.

Upon collecting the evidence and successfully creating 2 images of the evidence using the forensic tool **FTK Imager**, the evidence was returned to Sarah Johnson and was signed of upon collection and return by Sarah Johnson. Below is an Image of the chain of custody form that has been used throughout the collection phase of the investigation.

A write blocker was also used specifically the **Orion USB write blocker**, a test was run on a random USB to verify the write blocker was active and working and then the USB was plugged in while the write blocker was turned on to ensure no changes were made to the original evidence. (Orion USB Write Blocker -Orion

Forensics LAB Thailand, n.d.)

This is an image of the Data Acquisition Plan that was created as per the NIST 800-86 guidelines under section **3.1.2** called “**Acquiring the Data**”.



Examination

The examination stage involves examining the collected evidence image file and any files or data that could be extracted from the evidence image file as well as documenting the findings found.

Tools Used

With the methodology used for the examination stage, a variety of industry-standard digital forensics tools were used generate image files of the evidence, examine the data inside of the evidence, maintain integrity throughout the investigation as well as bypass any password-protected files found in the evidence.

- **AcessData FTK imager tool 3.1.2.0** was used to create an E01 image of the evidence file.
- **Autopsy 4.19.3** was used to investigate the E01 file of the suspect's machine.
- **Hashcat 2.6.2** was used to crack password-protected files and retrieve the password.
- **Office2John.py** was used to extract a password hash from an office document.
- **OneDrive** was used to store a copy of the imaged evidence to maintain integrity.
- **Corbett Backup Restore Wizard 4.5.0.0** converter to convert email messages to a readable pdf document.
- **Eric Zimmermans ShellBags Explorer 2.0.0.0** to analyse .DAT files found.
- **Eric Zimmermans Registry Explorer 2.0.0.0** to view registry files.
- **Orion USB write blocker 1.0.0** to ensure integrity when imaging the original evidence.

Each one of these tools were chosen to be used for its reliability and efficiency when it comes to carrying out a digital forensic investigation.

Findings

Below are all the findings related to users, items, events, places and suspicious activities or files in a tabular format.

List of Users

This is a list of all the user accounts found in the “*Washer 17.E01*” file and their corresponding email addresses.

| Users | Email |
|---------------------------|--------------------------|
| John Washer | chkwasher@comcast.net |
| Rasco Bad Guy | txkidd@swbell.net |
| Wes Mantooth | dollarhyde86@comcast.net |
| Billy Bob Brubeck | - |
| Mr Smee | smee.rox@gmail.com |
| The Wolf | - |
| Captain Hook | - |
| Artimus | - |
| Guest | - |
| Administrator | - |
| David Thomas (Skimmerman) | skimmerman27@hotmail.com |

Suspicious Evidence

This is a table created to showcase all the interesting files and directories found throughout the investigation. The files were found using the tool Autopsy and were extracted to the default **/exports** directory in autopsy. The table mentions the files found, their path and descriptions.

| File Name | Relative Path | Description |
|--|---|--|
| Derived HTML Files | | |
| F003815.html | /img_Washer 17.E01/vol_vol2/\$CarvedFiles/1/ | This file contains chat logs between John Washer with username Washergonebad and Rasco Badguy with username rbadguy2424 discussing meeting up and cooking meth. |
| /f0003908.html | /img_Washer 17.E01/vol_vol2/\$CarvedFiles/1/ | This file shows a google search query that was carried out regarding credit card skimming, this shows the suspect john washer was looking into learning more about this crime or how to potentially carry it out. |
| 0 | /img_Washer 17.E01/vol_vol2/Documents and Settings/Administrator/Local Settings/Temporary Internet Files/Content.IE5/O6O6PIEA/index[1].pl/index[1].pl/ | This is an html page created to serve as a platform where new users who would like to carry out check washing fraud can create an account and the platform offers insight as to what kind of preventive measures are in place and what chemicals to use. |
| Microsoft Office Documents & PDF Files | | |
| How To Steal Credit Numbers.doc | /img_Washer 17.E01/vol_vol2/Documents and Settings/Administrator/My Documents/ | This document was Password-Protected , when cracked, the file mentions abusing AOL's passprogram to steal credit card numbers. By sending an email to passprogram@aol with a specific subject line and the sender's screen name, alongside other information mentioned it is possible to successfully bypass security measures and receive the credit card number requested thus causing credit card theft . |
| SLIST.doc | /img_Washer 17.E01/vol_vol2/Documents and Settings/Administrator/My Documents/ | This document was Password-Protected , when cracked the file contains 4 credit cards with their corresponding expiry date, CVC number and a letter at the end specifying what type of card it is, for example M for Mastercard, D for debit card etc. |
| Washers To Do List.doc.doc | /img_Washer 17.E01/vol_vol2/Documents and Settings/Administrator/Desktop/Stuff/ | This document contains a list of things John Washer wants to do including buying peanut butter, killing familiars, burying Wes's enemies who is his friend and accomplice, the document also mentions that John washer will confess to the police. |
| X marks the spot.doc | /img_Washer 17.E01/vol_vol2/Documents and Settings/Administrator/Local Settings/Temporary Internet Files/Content.IE5/O6O6PIEA/ | This document is a Password-Protected document, when cracked the document reveals a map location of where John Washer and his accomplices are going to be Drug Manufacturing. |
| The Dealz.doc | /img_Washer 17.E01/vol_vol2/Documents and Settings/Administrator/My Documents/ | This is a word document but when attempted to open the file, it prompts for file conversion and to choose an encoding type, after many attempts, none of the prompted file type encodings made the document readable. |
| ALLSTATE CREDIT AGENCY.pdf | /img_Washer 17.E01/vol_vol2/Documents and Settings/Administrator/My Documents/ | This is a Password-Protected document, obtaining the password was not successful as the wordlists did not contain the password for this document. |
| Card_Printing_101.pdf | Inside the sent items.dbx file | A file containing information about credit card printing and the different credit cards and ID cards and how they are printed. |
| Malicious .exe File (Malware) | | |
| Install_AIM59[1].exe | /img_Washer 17.E01/vol_vol2/Documents and Settings/Administrator/Local Settings/Temporary Internet Files/Content.IE5/8VHFTJ2/ | This file was suspected to be malware, and therefore has been examined using dynamic analysis by utilising websites such as any.run & tria.ge . |
| Microsoft Outlook express .dbx Files | | |
| Sent Items.dbx | /img_Washer 17.E01/vol_vol2/Documents and Settings/Administrator/Local Settings/Application Data/Identities/{6B6FD541-F2AF-4EFB-AF50-EC531BF02474}/Microsoft/Outlook Express/ | This is a saved Microsoft Outlook express folder of all the sent items from john washer's email address. |
| Inbox.dbx | /img_Washer 17.E01/vol_vol2/Documents and Settings/Administrator/Local Settings/Application Data/Identities/{6B6FD541-F2AF-4EFB-AF50-EC531BF02474}/Microsoft/Outlook Express/ | This is a saved Microsoft Outlook express folder of all the inbox items from john washer's email address. |
| Deleted Items.dbx | /img_Washer 17.E01/vol_vol2/Documents and Settings/Administrator/Local Settings/Application Data/Identities/{6B6FD541-F2AF-4EFB-AF50-EC531BF02474}/Microsoft/Outlook Express/ | This is a saved Microsoft Outlook express folder of all the deleted items from john washer's email address. |

Obtaining Passwords for Password Protected Files

There were 4 password protected files, and this section of the report showcases how the passwords were obtained. After extracting the 3 passwords shared between a chat log in the **F003815.html** file and adding it to a file called *extractedpw_wordlist.txt* with the following passwords:

- ✓ M3th1sR1sky
- ✓ attica
- ✓ Outt0st3a1

This password file was used alongside the extracted password hashes that was obtained using the tool *Office2John.py* and was used with the tool *hashcat* to crack the hashes of the following password protected documents.

How To Steal Credit Numbers.doc

The password hash was extracted from the document using the command `python3 office2john.py passwd_protected/ How\ To\ Steal\ Credit\ Numbers.doc > stealCreditCards_hash.txt` and then hashcat was used to crack the password using the command `hashcat -a -m 9700 stealCreditCards_hash.txt extractedpw_wordlist.txt` which revealed the password to be `0utt0st3a1`.

```
(kali@kali)-[~/Downloads/forensics]
└─$ python3 office2john.py passwd_protected/How\ To\ Steal\ Credit\ Numbers.doc
How To Steal Credit Numbers.doc:$oldoffice$1*7ee44ef0aa2d1a422b775a6f6b0b3c95*7d6eb910777b1ee5d280c67ebad30005*1ac0ac77c1aad17005dcdbde6480c1444:::passwd_protected/How To Steal Credit Numbers.doc
```

Screenshot 1: Extracted Hash

```
(kali@kali)-[~/Downloads/forensics]
└─$ hashcat -m 9700 -a 0 stealcreditcards_hash.txt extractedpw_wordlist.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 4.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.7, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

* Device #1: cpu-penryn-11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz, 4105/8274 MB (2048 MB allocatable), 2MCU

/usr/share/hashcat/OpenCL/m09700_a0-optimized.cl: Pure kernel not found, falling back to optimized kernel
Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 15

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Optimized-Kernel
* Zero-Byte
* Precompute-Init
* Not-Iterated
* Single-Hash
* Single-Salt

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 0 MB

Dictionary cache hit:
* Filename ..: extractedpw_wordlist.txt
* Passwords ..: 3
* Bytes ..: 29
* Keyspace ..: 3

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.

$oldoffice$1*7ee44ef0aa2d1a422b775a6f6b0b3c95*7d6eb910777b1ee5d280c67ebad30005*1ac0ac77c1aad17005dcdbde6480c1444:0utt0st3a1

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 9700 (MS Office <= 2003 $0/$1, MD5 + RC4)
Hash.Target.....: $oldoffice$1*7ee44ef0aa2d1a422b775a6f6b0b3c95*7d6eb910777b1ee5d280c67ebad30005*1ac0ac77c1aad17005dcdbde6480c1444
Time.Started.....: Wed Mar 20 22:24:25 2024 (0 secs)
Time.Estimated...: Wed Mar 20 22:24:25 2024 (0 secs)
Kernel.Feature...: Optimized Kernel
Guess.Base.....: File (extractedpw_wordlist.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 6345 H/s (0.01ms) @ Accel:1024 Loops:1 Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 3/3 (100.00%)
Rejected.....: 0/3 (0.00%)
Restore.Point...: 0/3 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1...: M3th1sR1skY -> 0utt0st3a1
Hardware.Mon.#1..: Util: 8%

Started: Wed Mar 20 22:24:23 2024
Stopped: Wed Mar 20 22:24:27 2024
```

Screenshot 2: Cracked Password

SLIST.doc

The password hash was extracted from the document using the command `python3 office2john.py passwd_protected/SLIST.doc > slist_hash.txt` and then hashcat was used to crack the password using the command `hashcat -a -m 9700 slist_hash.txt extractedpw_wordlist.txt` which revealed the password to be `attica`.

```
(kali@kali)-[~/Downloads/forensics]
└─$ python3 office2john.py passwd_protected/SLIST.doc
SLIST.doc:$oldoffice$1*6ba43f09f1c25635eab6254ae8b2b443*3c45ad977786a6efec971ac5b7e7f644*b08a2ebcbf4a47bb82750c4a12d5c87a:::passwd_protected/SLIST.doc
```

Screenshot 3: Extracting Hash

```

└─$ hashcat -m 9700 -a 0 slist_hash.txt extractedpw_wordlist.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 4.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.7, SLEEP, DISTRO, POCL_DEBUG) -
Platform #1 [The pocl project]

=====
* Device #1: cpu-penryn-11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz, 4105/8274 MB (2048 MB allocatable), 2MCU

/usr/share/hashcat/OpenCL/m09700_a0-optimized.cl: Pure kernel not found, falling back to optimized kernel
Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 15

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Optimized-Kernel
* Zero-Byte
* Precompute-Init
* Not-Iterated
* Single-Hash
* Single-Salt

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 0 MB

Dictionary cache built:
* Filename..: extractedpw_wordlist.txt
* Passwords.: 3
* Bytes.....: 30
* Keyspace..: 3
* Runtime ...: 0 secs

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.

$oldoffice$1*6ba43f09f1c25635eab6254ae8b2b443*3c45adc977786a6efec971acb5e7f644*b08a2ebcbf4a47bb82750c4a12d5c87a:att
ica

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 9700 (MS Office ≤ 2003 $0/$1, MD5 + RC4)
Hash.Target.....: $oldoffice$1*6ba43f09f1c25635eab6254ae8b2b443*3c45a ... d5c87a
Time.Started.....: Wed Mar 20 22:19:33 2024 (0 secs)
Time.Estimated...: Wed Mar 20 22:19:33 2024 (0 secs)
Kernel.Feature...: Optimized Kernel
Guess.Base.....: File (extractedpw_wordlist.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 46 H/s (0.01ms) @ Accel:1024 Loops:1 Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 3/3 (100.00%)
Rejected.....: 0/3 (0.00%)
Restore.Point....: 0/3 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: M3th1sR1sky → 0utt0st3a1
Hardware.Mon.#1..: Util: 34%

Started: Wed Mar 20 22:19:30 2024
Stopped: Wed Mar 20 22:19:34 2024

```

Screenshot 4: Cracking password of SLIST

This file had contained 4 stolen credit cards details that included the 16 digits, the CVC number, the expiry date and a letter corresponding with what type of card it is.

| | | | |
|---------------------|------|-------|---|
| 1234 5678 1234 1234 | 232 | 10/09 | M |
| 0012 3330 3330 3030 | 676 | 03/10 | M |
| 2145 0909 9888 0989 | 998 | 02/10 | V |
| 1929 000986 12345 | 4253 | 11/09 | A |

X marks the spot.doc

Using the wordlist *10-million.txt* file and *Office2John.py* script, the password hash was extracted from the document using the command `python3 office2john.py passwd_protected/X\ marks\ the\ spot.doc > xmarksthespot_hash.txt` and then hashcat was used to crack the password using the command `hashcat -a -m 9700 xmarksthespot_hash.txt 10-million.txt` which revealed the password to be `camp`.

```
(kali@kali)-[~/Downloads/forensics]
$ python3 office2john.py passwd_protected/X\ marks\ the\ spot.doc
X marks the spot.doc:$oldoffice$1*ce6447a9c582476651158cf84e997bb0*86863fda5dfffd262ab99d3af808f89d8*6df23348ba793b231fa767d1fdf3714b:::passwd_protected/X m
arks the spot.doc
```

Screenshot 5: extracting hash

```
kali@kali: ~/Downloads/forensics
File Actions Edit View Help

/usr/share/hashcat/OpenCL/m09700_a0-optimized.cl: Pure kernel not found, falling back to optimized kernel
Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 15

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Optimized-Kernel
* Zero-Byte
* Precompute-Init
* Not-Iterated
* Single-Hash
* Single-Salt

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 0 MB

Dictionary cache built:
* Filename..: Password/hashcat-6.2.6/10-million.txt
* Passwords.: 999998
* Bytes.....: 8529108
* Keyspace...: 999998
* Runtime...: 0 secs

$oldoffice$1*ce6447a9c582476651158cf84e997bb0*86863fda5dfffd262ab99d3af808f89d8*6df23348ba793b231fa767d1fdf3714b:camp

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 9700 (MS Office ≤ 2003 $0/$1, MD5 + RC4)
Hash.Target.....: $oldoffice$1*ce6447a9c582476651158cf84e997bb0*86863... f3714b
Time.Started.....: Thu Mar 21 01:10:02 2024 (0 secs)
Time.Estimated...: Thu Mar 21 01:10:02 2024 (0 secs)
Kernel.Feature...: Optimized Kernel
Guess.Base.....: File (Password/hashcat-6.2.6/10-million.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 372.3 kH/s (2.86ms) @ Accel:1024 Loops:1 Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 14340/999998 (1.43%)
Rejected.....: 4/14340 (0.03%)
Restore.Point....: 12290/999998 (1.23%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: pulled → 06121989
Hardware.Mon.#1..: Util: 14%

Started: Thu Mar 21 01:10:00 2024
Stopped: Thu Mar 21 01:10:04 2024
```

Screenshot 6: cracking password

ALLSTATE CREDIT AGENCY.pdf

The password hash was obtained using the *pdf2john.pl* perl script which is part of the same suite of *office2john.py* but whilst extracting the password hash was successful, obtaining the password and successfully cracking it was not.

```
(kali@kali)-[~/Downloads/forensics]
└─$ perl Password/john-bleeding-jumbo/run/pdf2john.pl passwd_protected/credit.pdf
passwd_protected/credit.pdf:$pdf$2*3*128*-1028*1*16*9642f477b8408c6af0d023b675c0dff8*32*b3816bb9a8e56b138cb24a485992b0200000000930a0000000000000000*32*d
46e99212fbd6ec007245a5c2cbc2287a87aba69362b9fdb5994d4b01e6c5b16

(kali@kali)-[~/Downloads/forensics]
└─$
```

Screenshot 7: extracted hash

INSTALL_AIM[1].exe

This executable file is suspected to be malicious software or malware so static analysis and dynamic analysis was conducted and the findings are shown in this section of the report.

Static Analysis

For static analysis, manalyzer.com was used which a website that does static analysis on a given executable. The website revealed a lot of different information like all 72 antivirus vendors on virustotal.com revealed that the file is safe as shown below.

Plugin Output

| | | |
|------------|---|---|
| Info | Matching compiler(s): | Wise Installer Stub |
| Suspicious | Strings found in the binary may indicate undesirable behavior. | Contains another PE executable: <ul style="list-style-type: none">This program cannot be run in DOS mode. |
| Info | The PE contains common functions which appear in legitimate applications. | Possibly launches other programs: <ul style="list-style-type: none">CreateProcessA Can create temporary files: <ul style="list-style-type: none">GetTempPathACreateFileA |
| Info | The PE is digitally signed. | Signer: America Online Issuer: VeriSign Class 3 Code Signing 2004 CA |
| Safe | VirusTotal score: 0/72 (Scanned on 2024-03-28 04:03:19) | All the AVs think this file is safe. |

Image 1: Static Analysis by hybrid-analysis.com

The program also imports multiple kernel32.dll API functions including the following:

| API FUNCTION | DESCRIPTION |
|---------------------|---|
| mCreateFileMappingA | Creates a named or unnamed file mapping object for a specific file. |
| CreateProcessA | Retrieves the command-line string for the current process. |
| GetCommandLineA | Retrieves the command-line string for the current process. |
| CloseHandle | Closes an already open handle object |
| WriteFile | Writes data to the specified file or input/output (I/O) device. |
| DeleteFileA | Deletes an existing file. |
| GetTempFileNameA | Creates a unique temporary file name. |
| CreateFileA | Creates or opens a file or I/O device. |

Dynamic Analysis

For dynamic analysis, websites like Any.run, tria.ge, and hybrid-analysis.com were used to conduct an in-depth analysis of the executable. (Free Automated Malware Analysis Service - Powered by Falcon Sandbox, 2019) (ANY.RUN - Interactive Online Malware Sandbox, n.d.)

The executable was uploaded to any.run for dynamic analysis and the website revealed that the executable was indeed malicious with many indicators. The second the file is run, an executable file is dropped `install_aim59[1].exe` which then installs `AIM_INSTALLER_DERANDOMIZED.EXE` which is responsible for querying the register and scanning for artifacts as well as creating a writeable file in the system directory of the machine, additionally it starts up a `cmd.exe` to be able to execute commands as well as dropping C-runtime libraries.

The MITRE ATT&CK Matrix below shows all the attack techniques and description that was used by the malware as well as a graph of how everything happened. (MITRE, n.d.)

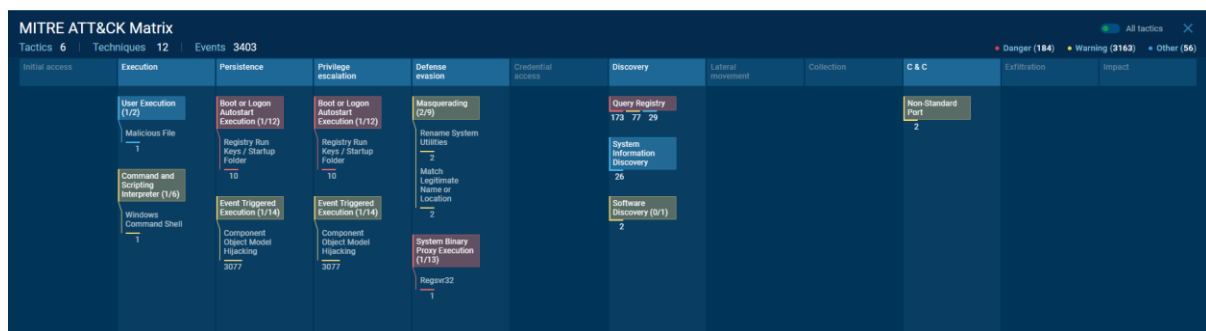


Figure 1: MITRE ATT&CK MATRIX

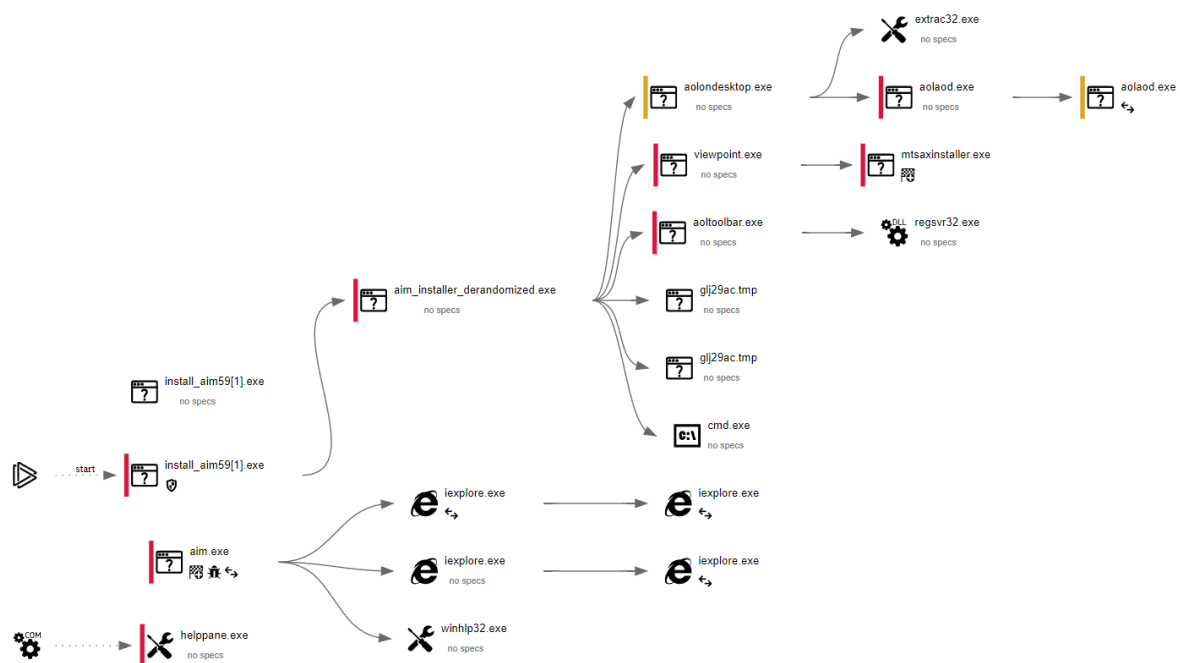


Figure 2: FLOW GRAPH OF ATTACK EXECUTION

The executable also loads very suspicious and unusual API functions such as the following which can all be used to aid in achieving malicious goals for the threat actor:

- LoadLibraryA
- GetModuleFileNameA
- WinExec

- GetTempPathA
- GetModuleHandleA
- GetTempFileNameA
- GetProcAddress

The malware ended up disguising itself as a trojan masquerading as the AIM software and instead installs and harms devices by modifying the registry, running arbitrary commands by spinning up a windows shell and dropping unusual malicious API calls as well as using access token manipulation to evade antivirus detection and carry out privilege escalation.

Analysis

This section of the report outlines all the device revolving the many crimes found during this investigation and are broken down into headings of each crime. The analysis of each crime and the evidence found to support the likelihood of whether the crime happened or not.

Credit Card Theft

After carefully examining the evidence there was many occasions where John Washer, Rasco Badguy and Wes Mantooth as well as Skimmerman were all involved both directly and indirectly in credit card theft, with the many emails exchange that happened there is much concrete evidence towards the conclusion that all 4 suspects were involved in credit card theft.

Inside the inbox.dbx file is a document sent through by Rasco Badguy to John Washer and Wes Mantooth which is a document on how to steal credit numbers, when password cracked and opened the file shows steps of a technique to carry out to gather someone's real credit card information by sending an email to the passprogram@aol.com with a specific code being h-kt0e-429-2391- and filling out more information in the body, you pretended to access your billing account by sending the email and in return AOL would mistakenly send the credit card details, they discussed that it was attempted multiple times and that it works perfectly. This email shows that this crime has been carried out multiple times.

Documents like SLIST.doc that had stolen credit card details are even more solid evidence that all 4 suspects mainly John washer and Rasco Badguy were possibly involved in credit card theft in all its forms including, credit card skimming, credit card printing, and more. The files supporting this conclusion were all mainly gathered from the inbox.dbx file and sent items.dbx files. Skimmerman was mainly the person with the most knowledge of how to carry out credit card skimming and theft and was sharing it with John Washer as well as Rasco Badguy to help them carry out this crime.

See **Appendix C** for screenshots of the evidence collected.

Drug Manufacturing

Throughout the investigation, there was much evidence that was discovered that supported the crime of Drug Manufacturing such as meth. In the first file that was found named `F003815.html`, there was evidence of chat logs between Rasco Badguy and John Washer making plans to meetup somewhere where an old lady was cooking meth. Further investigation found that Rasco Badguy and John Washer had made plans to go 'camping' somewhere secluded to cook more drugs, evidence to support this was found in the email files. Specifically, when a document called `X marks the spot.doc` was shared mentioning that they will be `cooking` up where they are going camping and can't afford any interruptions, and throughout the investigation the location of where they are supposedly cooking was revealed to be Red Feathers Lake Colorado, this was revealed through an email with subject 'Camping Directions'. Based on the evidence found, there is no doubt that Meth was being cooked in their camping location.

See **Appendix D** for screenshots of the evidence collected.

Theft of Pharmaceutical Drugs

Whilst there is very little evidence to support this crime being committed, there is one very strong undeniable evidence that was found which was an email specifically in the sent items.dbx file which mentions how Wes Mantooth stole some drugs from the pharmacy counter because a lady left it in the counter and he took it without the lady catching him in the act, Wes Mantooth told

this to John washer via email and therefore this proves that Wes Mantooth stole and took drugs from a pharmacy. See **Appendix E** for screenshots of the evidence collected.

Check Washing

During the investigation, check washing was brought up a few times, whilst not as much as credit card theft, there was a few evidence that was found related to check washing. Starting off with the file named `0` where it is an html derived file showing a platform and a web page related to check washing and it is welcoming new users to check washing as well as explaining what check washing is and how it can be a source of easy income. The second evidence supporting check washing was in the sent Items.dbx email file which shows a conversation between John Washer and Wes Mantooth discussing check washing and sending links to each other about the idea `<http://celtickane.com/projects/washing/>`. See **Appendix F** for screenshots of the evidence collected.

Reporting

Likelihood of each crime

Based on the evidence collected and the analysis conducted on the image, the following table has been created to showcase the likelihood of whether the crime happened or not.

| Likelihood | Key |
|------------------|-----|
| Very Likely | |
| Likely | |
| Partially Likely | |
| Unlikely | |
| Very Unlikely | |

| Crime | Likelihood |
|-------------------------------|------------|
| Credit Card Theft | |
| Drug Manufacturing | |
| Check Washing | |
| Theft of Pharmaceutical Drugs | |

Timeline

The timeline shown here is a very shortened version of the full timeline, because of the nature of how verbose the evidence that was found was, a simpler and smaller version of the timeline was created for this.

11-09-2003

- **07:41:00:** "options.doc" Document Created
- **08:23:00:** "options.doc" Document Last Saved

25-07-2007

- **02:26:23:** Malware file created.
- **02:26:41:** Malware file modified.
 - John Washer conducts web searches on "how to steal checks."
- **21:07:47:** "How to steal credit numbers.doc" created.
- **21:07:55:** "How to steal credit numbers.doc" created.

- **21:08:55:** Document modified a minute later.
 - **21:26:25:** "The Dealz.doc" file changed and modified.
 - Shortly after: "X marks the spot.doc" modified and changed.
 - **21:31:41:** "ALLSTATE CREDIT AGENCY.pdf" file created and accessed
- 02-08-2007**
- Web searches related to credit card printers and making credit cards.
 - Google Dorking for a PDF, likely finding the "Card_Printing_101.pdf" file.
 - Searches span from **20:03 to 20:10.**
 - **21:08:10:** "ALLSTATE CREDIT AGENCY.pdf" file modified
- 12-02-2008**
- **00:46:00:** "Washer To Do List.doc.doc" Document Created
 - **00:55:00:** "Washer To Do List.doc.doc" Document Last Saved
 - **11:20:50:** "NTUSER.DAT" File Modified
- 13-02-2008**
- **00:46:03:** "changing your identity" google web search
 - **19:30:47:** "Install_AIM59[1].exe" File Changed
 - **19:30:56:** "Install_AIM59[1].exe" File Accessed
 - **19:36:08:** "X marks the spot.doc" File Changed

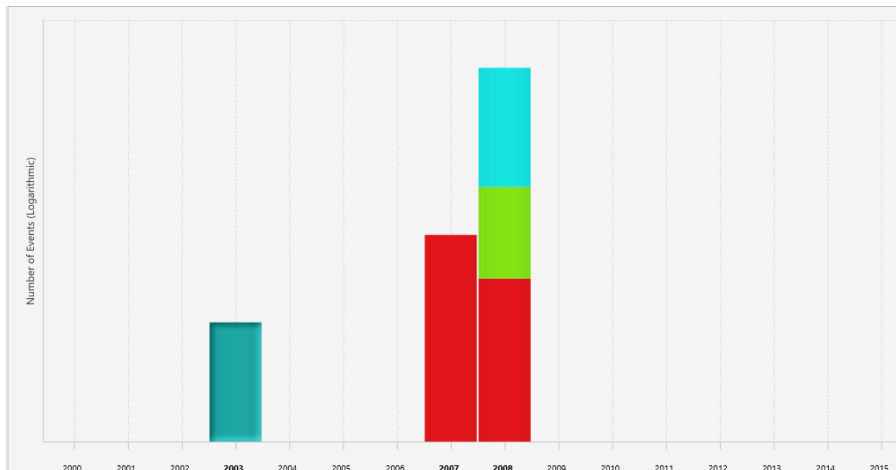
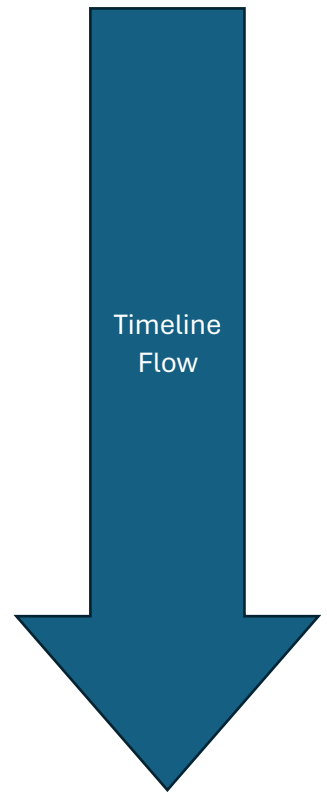


Figure 3: Timeline

Applicable Regulations

Overall, this digital forensic report conducted by CyberSleuth Solutions Ltd., follows the ACPO (**Association of Chief Police Officers**) guidelines as well as RIPA (**Regulation of Investigatory Powers Act 2000**) throughout the entirety of the investigation from its initial point of collection all the way through to its analysis and reporting stage. The steps taken to create this comprehensive report is strictly in adherence to the NIST (**National Institute of Standards and Technology**) 800-86 guidelines, titled "**Guide to Integrating Forensic Techniques into Incident Response**," The report outlines each phase of the methodology set forth by NIST and documents the findings, tools used and a track of the chain of custody as well as the case notes. Furthermore, the analysis of the evidence file provided specifically the crimes such as credit card theft, drug manufacturing, theft of pharmaceutical drugs, and check washing, is conducted with carefully with precision thorough attention.

Legal and Ethical Considerations

Throughout the digital forensic investigation, I have taken all the necessary steps alongside Sarah Johnsons supervision to follow the NIST 800-86 guidelines correctly and carefully as well as prioritizing the principals of confidentiality and integrity and the preservation of the evidence in a chain of custody form that can be found in **Appendix A**. On the ethical side, the use of digital forensic tools was chosen to maintain the principles as well as complying with ACPO guidelines and RIPA.

Conclusion

In conclusion, there are many copious evidence supporting some of the crimes mentioned in this report of the suspects carrying out the crime, some crimes had clearer evidence than others, and while the investigation was carried out, I, Ahmed Rashwan made sure to comply with all necessary guidelines, standards, regulations and policies and made sure that all the evidence found was clearly shown in this report in a manner that meets necessary guidelines as well as the NIST 800-86 guidelines.

References

- ANY.RUN - Interactive Online Malware Sandbox. (n.d.). Any.run. <https://any.run/>
- Athena Forensics. (2018). *An Explanation of ACPO Guidelines for Digital Based Evidence*. Athena Forensics. <https://athenaforensics.co.uk/acpo-guidelines-for-computer-forensics/>
- Download AccessData FTK Imager by AccessData Group, LLC. (n.d.). Accessdata-Ftk-Imager.software.informer.com. Retrieved March 29, 2024, from <https://accessdata-ftk-imager.software.informer.com/download/>
- Email Backup Wizard to Backup & Restore Emails From 209+ Platforms. (n.d.). Corbettsoftware.com. Retrieved March 29, 2024, from <https://corbettsoftware.com/backup-restore/>
- John the Ripper. (2023, October 22). GitHub. <https://github.com/openwall/john/tree/bleeding-jumbo>
- Kent, K., Chevalier, S., Grance, T., & Dang, H. (2006). *Special Publication 800-86 Guide to Integrating Forensic Techniques into Incident Response Recommendations of the National Institute of Standards and Technology*. <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-86.pdf>
- legislation.gov.uk. (2019). *Regulation of Investigatory Powers Act 2000*. Legislation.gov.uk. <https://www.legislation.gov.uk/ukpga/2000/23/contents>
- MITRE. (n.d.). MITRE ATT&CK™. Mitre.org. <https://attack.mitre.org/>
- Orion USB Write Blocker -Orion Forensics LAB Thailand. (n.d.). Orion Forensics Thailand. Retrieved March 25, 2024, from <http://www.orionforensics.com/forensics-tools/orion-usb-write-blocker/>
- Sandbox for High-Volume Automated Malware Analysis. (n.d.). Tria.ge. <https://tria.ge/>
- Williams, J. (2012). *ACPO Good Practice Guide ACPO Good Practice Guide for Digital Evidence for Digital Evidence*. https://www.digital-detective.net/digital-forensics-documents/ACPO_Good_Practice_Guide_for_Digital_Evidence_v5.pdf

Appendix A (Chain Of Custody)

Digital Forensics Acquisition Form



Case Information

Case Number: Case-2024-0008 Evidence Bag Number M32228965

Evidence Number: EXH-2024-748

Date/Time of Collection: 16/02/2024 09:47am Officer's Signature of Collection Rosstaylor

Date/Time of Return: 16/02/2024 11:03am Officer's Signature of Return Rosstaylor

Device Type: USB

Device Make/Model: Vendor Co ProductCode USB Device

Device Description: Yellow USB with the word integral // USB Device with suspicious evidence

Serial Number: 7782251041355063269

Storage/Capacity: 15000 MB

Evidence Provider: DIGIMA

Location of Device: Coventry University EEC building first floor outside EC1-13

Acquisition Notes

| |
|--|
| |
|--|

Image 2 Chain of custody:

Appendix B (Authorization Letter)



Image 3: Authorization letter

Appendix C (Screenshots of Credit card Theft)

| Hex | Text | Application | Source File Metadata | OS Account | Data Artifacts | Analysis Results | Context | Annotations | Other Occurrences |
|--|------|-------------|----------------------|------------|----------------|------------------|---------|-------------|-------------------|
| Result: 10792 of 11095 Result < > | | | | | | | | | |
| Web Search Term: credit card printer filetype:pdf Time: 2007-08-02 20:09:56 BST Domain: google.com Program Name: Internet Explorer Analyzer | | | | | | | | | |
| Source Host: Washer 17.E01_38 Host Data Source: Washer 17.E01 File: /img_Washer 17.E01/vol_vol2/Documents and Settings/Administrator/Local Settings/Temporary Internet Files/Content.IE5/index.dat | | | | | | | | | |

Image 4: Credit Card theft Evidence 1

| Hex | Text | Application | Source File Metadata | OS Account | Data Artifacts | Analysis Results | Context | Annotations | Other Occurrences |
|--|------|-------------|----------------------|------------|----------------|------------------|---------|-------------|-------------------|
| Result: 10663 of 11095 Result < > | | | | | | | | | |
| Web Search Term: credit card printer Time: 2007-08-02 20:05:39 BST Domain: tele-pak.com Program Name: Internet Explorer Analyzer | | | | | | | | | |
| Source Host: Washer 17.E01_38 Host Data Source: Washer 17.E01 File: /img_Washer 17.E01/vol_vol2/Documents and Settings/Administrator/Local Settings/Temporary Internet Files/Content.IE5/index.dat | | | | | | | | | |

Image 5: Credit Card theft Evidence 2

| Hex | Text | Application | Source File Metadata | OS Account | Data Artifacts | Analysis Results | Context | Annotations | Other Occurrences |
|--|------|-------------|----------------------|------------|----------------|------------------|---------|-------------|-------------------|
| Visit Details Date Accessed: 2007-08-02 20:05:20 BST Domain: ebay.com URL: http://search.express.ebay.com/expansion/_?_nkw=credit card printer&_ipg=5&uaslop=0&fromCoreSearch=1&ru=http%3A%2F%2Fsearch.ebay.com%3A80%2Fsearch%2Fsearch.htm Program Name: Internet Explorer Analyzer | | | | | | | | | |
| Source Host: Washer 17.E01_38 Host Data Source: Washer 17.E01 File: /img_Washer 17.E01/vol_vol2/Documents and Settings/Administrator/Local Settings/Temporary Internet Files/Content.IE5/index.dat | | | | | | | | | |

Image 6: Credit Card theft Evidence 3


| Content | Message Header | Hex View | Raw Message |
|--|----------------|----------|-------------|
| Rasco Badguy <txkidd@swbell.net> You will love this.... To : John Washer <chkwasher@comcast.net> Cc : Wes Mantooth <dollarhyde86@comcast.net>  How To Steal Credit Numbers.doc 27 KB | | | |
| This doc says it all guys.....ran across it and my buddy skimmerman used it a few times.....worked like a charm..... R~ | | | |

Image 7: Credit Card theft Evidence 4

How To Steal Credit Numbers

Ok this information is about the same as how to steal passwords but its credit cards your stealing this time.

First off aol has a passprogram that not only has to do with passwords but also credit card numbers off of aol billing. So what you do is go to write email and put passprogram@aol in the send box.

Next- in the subject box put in h-kte-429-2391- so aol gets a message that will let you by the pass block. Next go to the first line (where you would write an email) and type your screen name and real-credit card number and the name on the credit card, so the reciever will read it an send it past thinking you are going into your billing account.

Next- in the 2nd line put in a fake persons name like joe brown and that fake person would be likely to be in aol billing. If not try a different name. Next in the 3rd line put in nothing, just leave it blank and that is it in one day aol will send you the credit card number of whoever you wanted. It is even easier than stealing passwords.

Image 8: Credit Card theft Evidence 5

Content Message Header Hex View Raw Message


Rasco Badguy <txkidd@swbell.net> 23/07/2007 19:59

RE: Stuff


To: John Washer <chkwasher@comcast.net>

SLIST.doc
26 KB

Got this little device from a friend you used to be a waiter. He said it was a money making item for him but things got a little hot so he need to dump it.



I have attached a small document with some of the information it can provide. I will get with you later on the password or just ask Wes. We vacationed there once. My buddy also has a lot of copies of these -



R~

Image 9: Credit Card theft Evidence 6

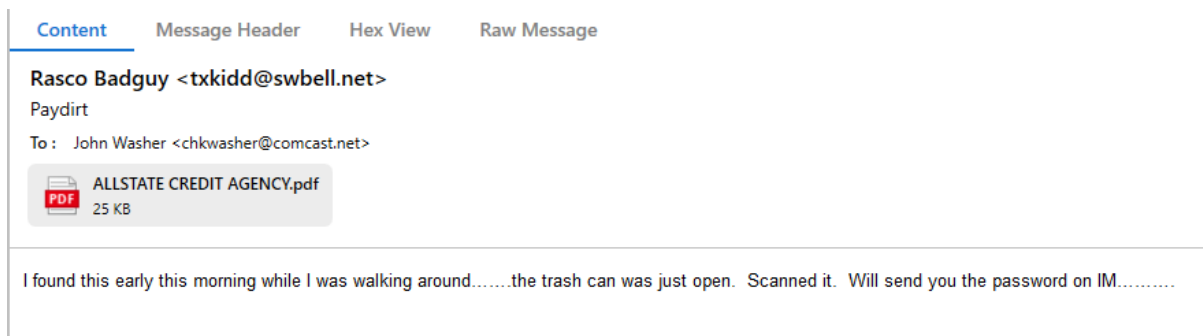


Image 10: Credit Card theft Evidence 7

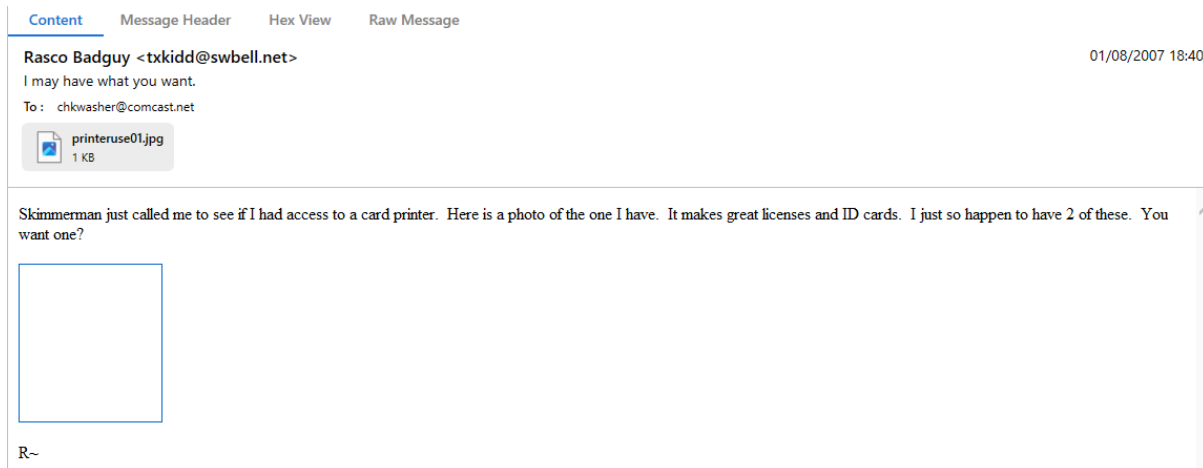


Image 11: Credit Card theft Evidence 9

Appendix D (Screenshots of Drug Manufacturing)

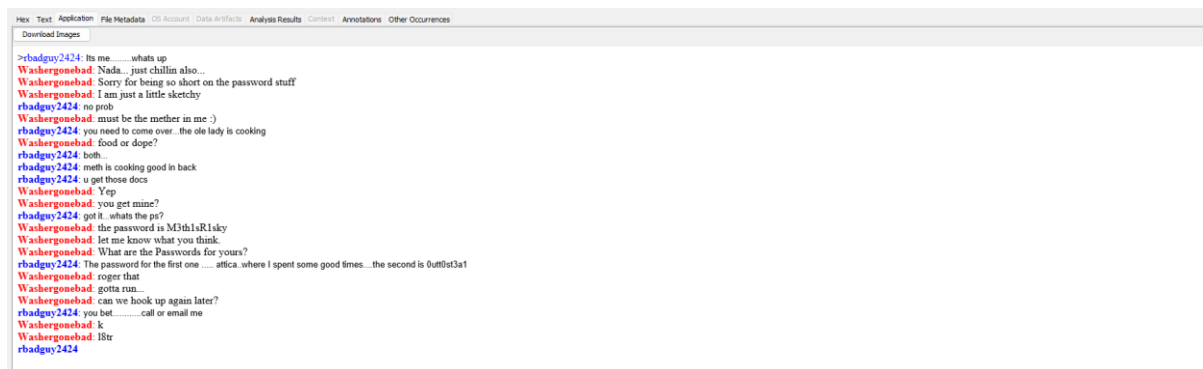


Image 12: Discussion of cooking drugs (Meth)

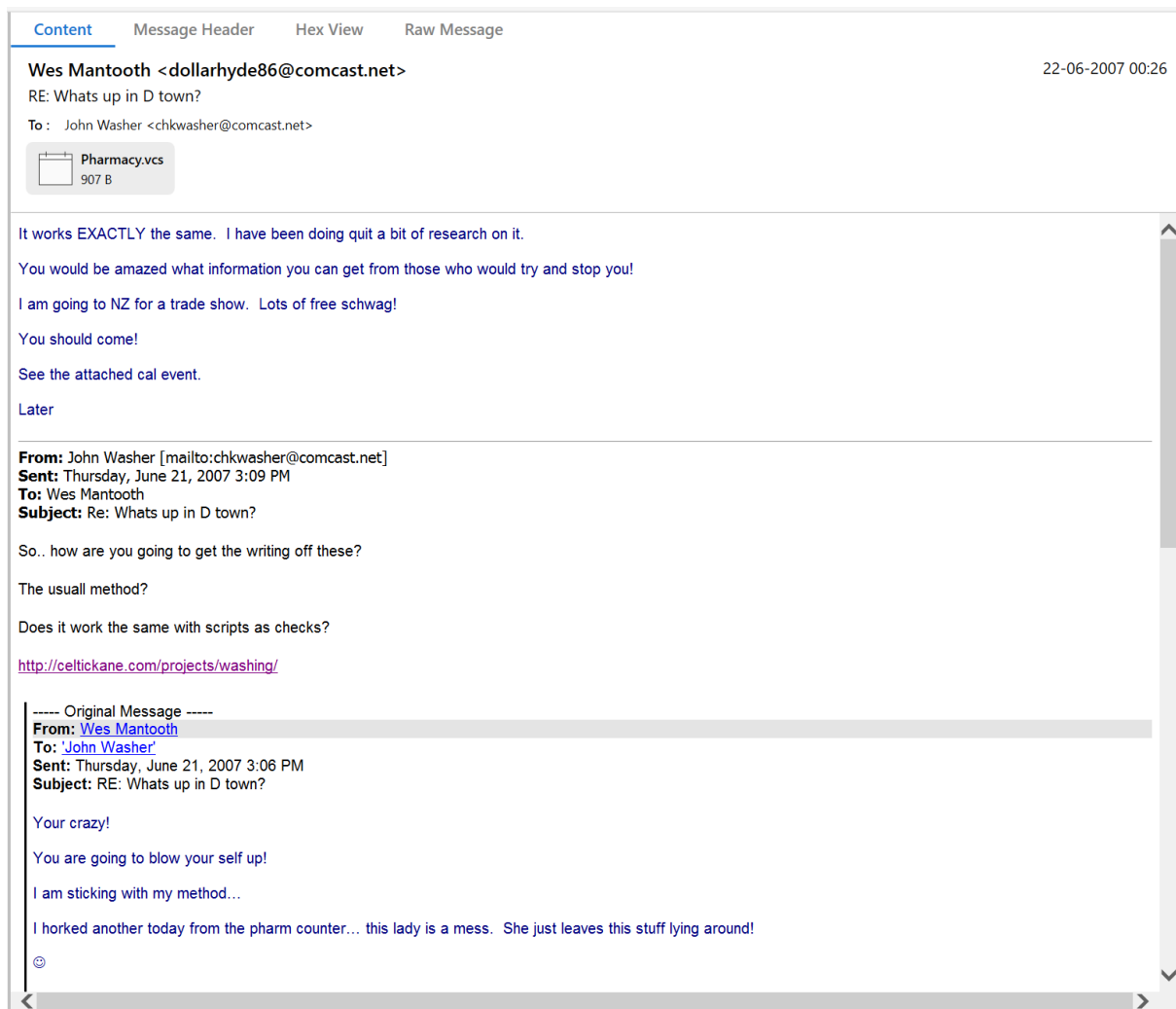


Image 13: Drug Manufacturing Evidence

Ok, so it is not an x... More like an "0". Here is where we are meeting. Please delete this and SHRED it when you are done. We are going to be cooking up there so we can't afford ANY interruptions if you know what I mean.

See you there!

JW

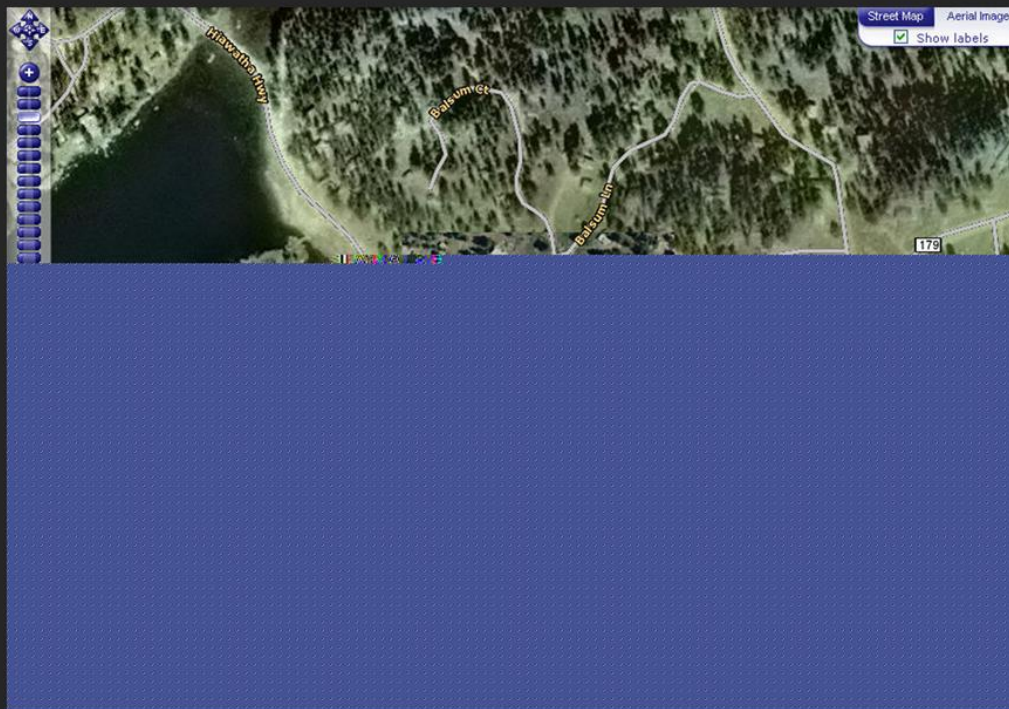


Image 14: Location of camping/Drug cooking

Appendix E (Screenshots of Pharmaceutical Theft)

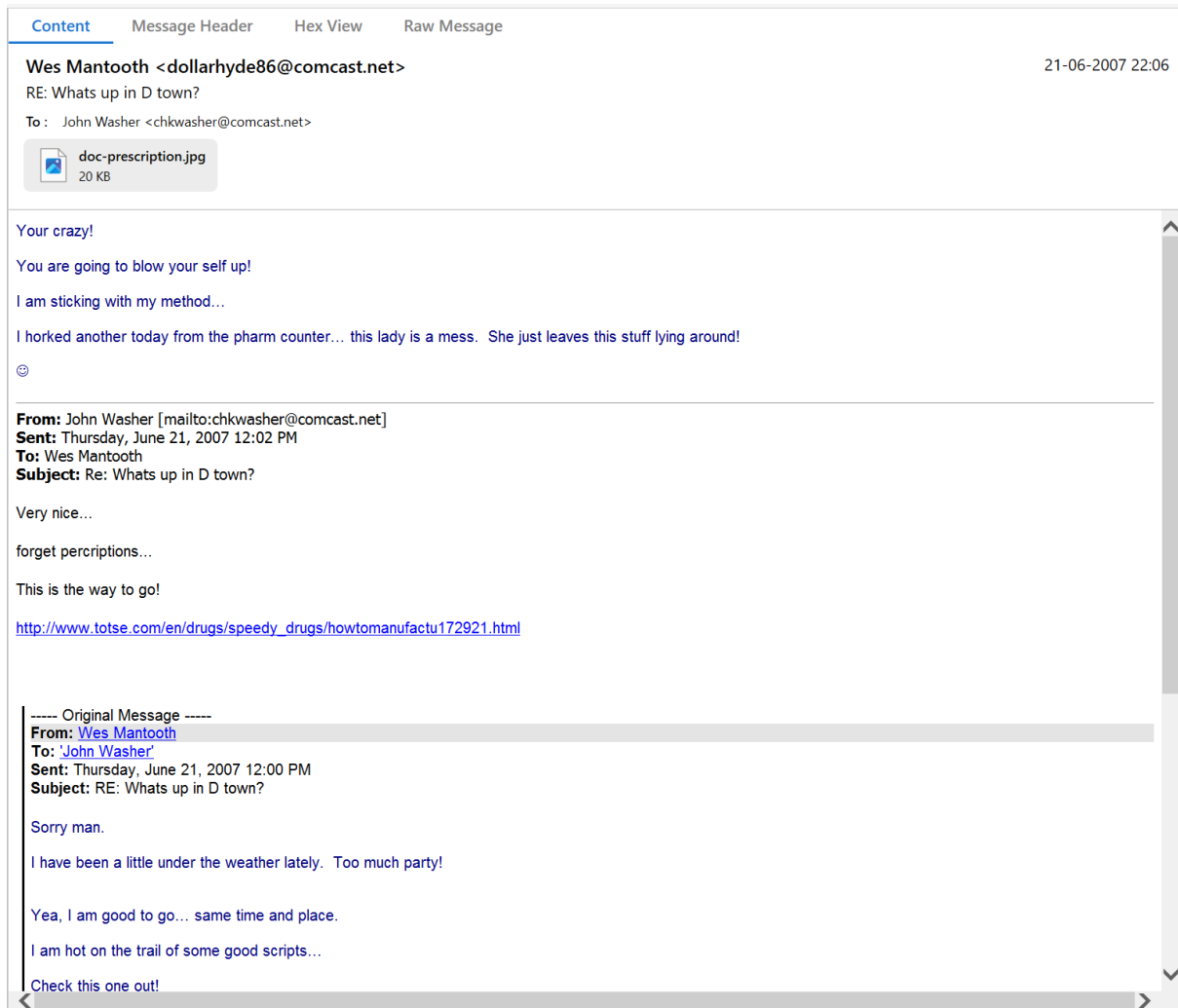


Image 15: Drug Theft From Pharmacy Evidence

Appendix F (Screenshots of Check Washing)

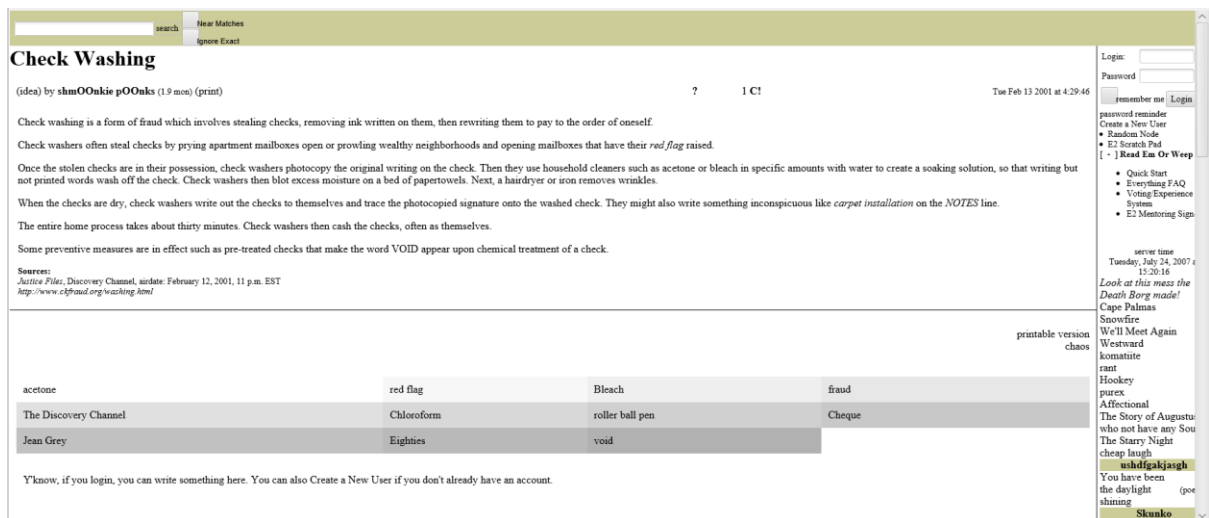


Image 16: Check Washing Evidence

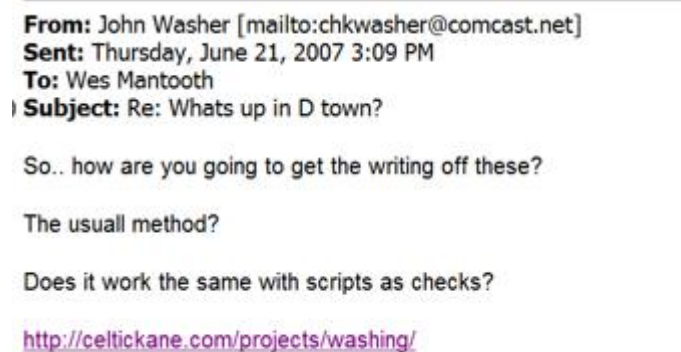


Image 17: Check Washing Evidence 2

Appendix G (Case Notes)

| Action Taken | Date & Time |
|---|--------------------|
| Uploaded control image and working copy image to OneDrive to maintain integrity | 22-03-2024,01:21am |
| Found F003815.html | 22-03-2024,01:24am |
| Found f0003908.html | 22-03-2024,01:26am |
| Found ALLSTATE CREDIT AGENCY.pdf | 22-03-2024,01:26am |
| Found How To Steal Credit Numbers.doc | 22-03-2024,02:13am |
| Found Washers To Do List.doc.doc | 22-03-2024,02:22am |
| Found SLIST.doc | 22-03-2024,02:30am |
| Found The Dealz.doc | 22-03-2024,03:36am |
| Install AIM59[1].exe | 22-03-2024,03:36am |
| Found Found Card_Printing_101.pdf | 22-03-2024,11:58am |
| Found Sent Items.dbx | 23-03-2024,03:56am |
| Found Inbox.dbx | 23-03-2024,04:06am |
| Found Deleted Items.dbx | 23-03-2024,04:37am |
| Found X marks the spot.doc | 23-03-2024,12:37pm |
| Found 0 | 25-03-2024,16:30pm |

