

Case Details	
Case Name:	WK#2 - Memory Image Analysis
Course Name:	CYB651
Professor:	Michael Woods
Date:	03/29/2020
Analyst Name:	Brian T. Carr

Table of Contents

List of Illustrative Materials.....	3
Tables.....	3
Figures.....	3
Executive Summary	4
Background	4
Request.....	4
Summary of Findings.....	4
Evidence.....	5
Collection and Analysis	6
Collection and Evidence Designations	6
Analysis.....	6
Conclusion	12
References	13
Appendix.....	14
Appendix A: Analyst Workstation Specifications	14
Appendix B: Software Applications Utilized	15

List of Illustrative Materials

Tables

Table 1: Case evidence items.....	5
Table 2: Analyst workstation specifications	14
Table 3: Tools used for collection and analysis.....	15

Figures

Figure 1: MD5 hashes of the evidence files.....	6
Figure 2: Volatility imageinfo plugin results for mem.dmp.	6
Figure 3: Volatility pslist plugin results for mem.dmp.....	7
Figure 4: Volatility connscan plugin results for mem.dmp.	7
Figure 5: Console commands captured in mem.dmp.....	8
Figure 6: Geolocation information for 67.205.91.17 (iplocation.net, 2020).	9
Figure 7: Volatility hivelist plugin results for mem.dmp.	9
Figure 8: Volatility hashdump results for mem.dmp.	10
Figure 9: Bash script to create audit file.	10
Figure 10: Creation of audit-20200328.txt.	10
Figure 11: Hash Suite results for hashes.txt.....	11
Figure 12: Edward Rainden's password hash verified with CrackStation. (Defuse Security, 2019)	11

Executive Summary

Background

On 3/23/2020, defense contractor Edward Rainden was suspected to have leaked classified information to various news outlets. At the time of this report, the whereabouts of the suspect was unknown, and there was an international manhunt regarding the suspect and the suspect was seeking asylum in a foreign country. Prior to fleeing the United States, the suspect executed code which caused NSA systems to read “NO MORE LIES! THE TRUTH IS FREE”. The suspect did not have permission to access or utilize the systems in question.

At the time of the incident, the suspect was suspected to have exfiltrated additional data. Evidence technicians collected an additional file which was believed to be the suspect’s TrueCrypt container. The Analyst was requested to obtain the password utilized by the suspect in hopes that it was the same password that the TrueCrypt container was utilizing.

Request

The Office of General Counsel (OGC) requested that the Analyst analyze the memory capture image to determine or locate:

- The operating system of the computer.
- Which processes were running at the time of the memory capture.
- If any of those running processes could be used to exfiltrate data.
- What network TCP connections existed at the time of the memory capture.
- Where the TCP connections traced back to.
- Information related to any commands executed.
- Information related to the system registry.
- Any password hashes stored in the memory capture image.

Summary of Findings

The Analyst performed an analysis of the mem.dmp memory capture image and determined that the suspect was utilizing the FileZilla FTP client to exfiltrate data to 67.205.19.17. This IPv4 address corresponded to iWeb Technologies Inc., a company located in Quebec. The Analyst was able to successfully extract the suspect’s password hash and decrypted it using Hash Suite, and CrackStation. The suspect’s password was determined to be: OPEN. Additionally, the Analyst located console information revealing that the suspect exfiltrated five text files and one PDF.

Evidence

Table 1 outlines the evidence items involved in this investigation.

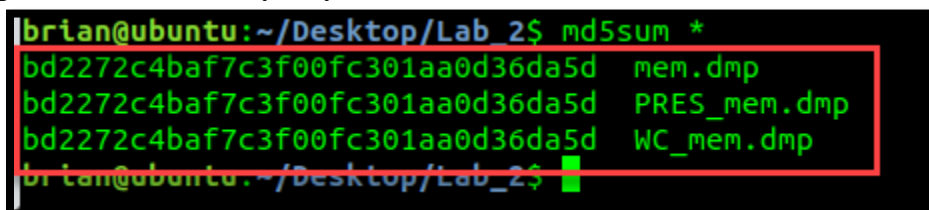
Table 1: Case evidence items

Description	Designation	Filename	MD5 Hash
Evidence Examined	Working Copy	mem.dmp	bd2272c4baf7c3f00fc301aa0d36da5d
Evidence Created	Preservation Copy	PRES_mem.dmp	bd2272c4baf7c3f00fc301aa0d36da5d
Evidence Provided	Preservation Copy	WC_mem.dmp	bd2272c4baf7c3f00fc301aa0d36da5d
Supplemental Files	Supplemental Files	audit-20200328.txt	4acf932fadff07c5ab1777d57958edc8

Collection and Analysis

Collection and Evidence Designations

On 3/23/2020, a memory capture image of Edward Raindens's laptop RAM (SN: S23SDD23) was provided to the Analyst by evidence technicians via Engage. The Analyst utilized the Volatility Framework to obtain relevant information from the memory capture image. The Analyst began by first downloading the memory capture image from the Analyst's engage shell. The Analyst was provided with an MD5 hash which corresponded to the original memory capture image. The provided MD5 hash was: bd2272c4baf7c3f00fc301aa0d36da5d. It can be outlined in *Figure 1* that each of the files had a hash value which corresponded to the one provided to the Analyst by evidence technicians.

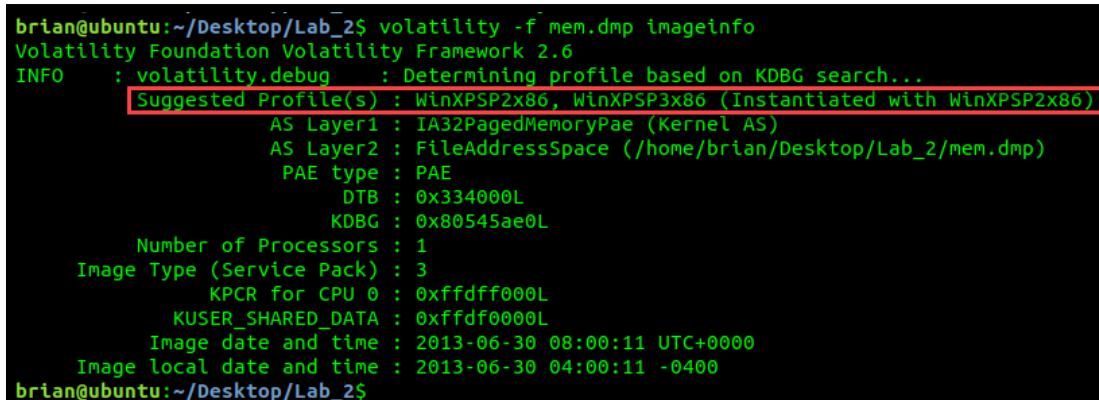
A terminal window with a black background and green text. The prompt is 'brian@ubuntu:~/Desktop/Lab_2\$'. The command 'md5sum *' has been executed, resulting in three lines of output: 'bd2272c4baf7c3f00fc301aa0d36da5d mem.dmp', 'bd2272c4baf7c3f00fc301aa0d36da5d PRES_mem.dmp', and 'bd2272c4baf7c3f00fc301aa0d36da5d WC_mem.dmp'. A red rectangular box highlights these three lines of output.

```
brian@ubuntu:~/Desktop/Lab_2$ md5sum *
bd2272c4baf7c3f00fc301aa0d36da5d mem.dmp
bd2272c4baf7c3f00fc301aa0d36da5d PRES_mem.dmp
bd2272c4baf7c3f00fc301aa0d36da5d WC_mem.dmp
brian@ubuntu:~/Desktop/Lab_2$
```

Figure 1: MD5 hashes of the evidence files.

Analysis

The working copy, mem.dmp, was Analyzed within both Linux and Windows virtual machines on the Analyst's forensic workstation. Both the Windows and Linux virtual machines were isolated from any network connection. The Analyst began the analysis of the working copy by first obtaining the proper profile which the Volatility Framework required for additional plugins. The Analyst can be seen utilizing the imageinfo plugin in *Figure 2*. Additionally, the results from the imageinfo plugin revealed that the profile of mem.dmp was either WinXPSP2x86 or WinXPSP3x86. The Analyst opted to use the WinXPSP2x86 profile for the remainder of the investigation.

A terminal window with a black background and green text. The prompt is 'brian@ubuntu:~/Desktop/Lab_2\$'. The command 'volatility -f mem.dmp imageinfo' has been executed. The output shows 'Volatility Foundation Volatility Framework 2.6' and 'INFO : volatility.debug : Determining profile based on KDBG search...'. A red rectangular box highlights the line 'Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)'. Below this, various system details are listed, including AS Layer1, AS Layer2, PAE type, DTB, KDBG, Number of Processors, Image Type, KPCR, KUSER_SHARED_DATA, and Image date and time.

```
brian@ubuntu:~/Desktop/Lab_2$ volatility -f mem.dmp imageinfo
Volatility Foundation Volatility Framework 2.6
INFO : volatility.debug : Determining profile based on KDBG search...
Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
AS Layer1 : IA32PagedMemoryPae (Kernel AS)
AS Layer2 : FileAddressSpace (/home/brian/Desktop/Lab_2/mem.dmp)
PAE type : PAE
DTB : 0x334000L
KDBG : 0x80545ae0L
Number of Processors : 1
Image Type (Service Pack) : 3
KPCR for CPU 0 : 0xffdff000L
KUSER_SHARED_DATA : 0xffdff000L
Image date and time : 2013-06-30 08:00:11 UTC+0000
Image local date and time : 2013-06-30 04:00:11 -0400
brian@ubuntu:~/Desktop/Lab_2$
```

Figure 2: Volatility imageinfo plugin results for mem.dmp.

The Analyst then utilized the pslist plugin of the Volatility Framework to obtain information regarding the running processes contained within the memory capture image. The Analyst was

specifically interested in any processes which would be utilized to exfiltrate data. The pslist plugin results for mem.dmp can be seen in *Figure 3*. The FileZilla application can be seen outlined in *Figure 3*. FileZilla is a free FTP application. (FileZilla, 2020)

```
brian@ubuntu:~/Desktop/Lab_2$ volatility -f mem.dmp --profile=WinXPSP2x86 pslist
Volatility Foundation Volatility Framework 2.6
```

Offset(V)	Name	PID	PPID	Thds	Hnds	Sess	Wow64	Start	Exit
0x823c8830	System	4	0	57	258	-----	0		
0x822265e8	smss.exe	540	4	3	19	-----	0	2013-06-30 07:58:40 UTC+0000	
0x81f465a8	csrss.exe	612	540	10	397	0	0	2013-06-30 07:58:42 UTC+0000	
0x81f4bda0	winlogon.exe	636	540	25	524	0	0	2013-06-30 07:58:42 UTC+0000	
0x822dcca8	services.exe	680	636	15	259	0	0	2013-06-30 07:58:42 UTC+0000	
0x820fe978	lsass.exe	692	636	24	349	0	0	2013-06-30 07:58:42 UTC+0000	
0x822d9458	vmacthlp.exe	852	680	1	25	0	0	2013-06-30 07:58:42 UTC+0000	
0x82318258	svchost.exe	868	680	20	201	0	0	2013-06-30 07:58:42 UTC+0000	
0x81d9b020	svchost.exe	944	680	9	238	0	0	2013-06-30 07:58:43 UTC+0000	
0x81ea8020	svchost.exe	1036	680	67	1183	0	0	2013-06-30 07:58:43 UTC+0000	
0x81e2f020	svchost.exe	1096	680	7	87	0	0	2013-06-30 07:58:43 UTC+0000	
0x81fbb7c0	svchost.exe	1164	680	15	200	0	0	2013-06-30 07:58:44 UTC+0000	
0x8228e230	spoolsv.exe	1416	680	16	148	0	0	2013-06-30 07:58:44 UTC+0000	
0x82290558	explorer.exe	1660	1600	14	324	0	0	2013-06-30 07:58:50 UTC+0000	
0x82104228	vmtoolsd.exe	1760	1660	5	215	0	0	2013-06-30 07:58:50 UTC+0000	
0x820fa020	vmtoolsd.exe	2012	680	9	278	0	0	2013-06-30 07:59:02 UTC+0000	
0x82106da0	filezilla.exe	244	1660	6	130	0	0	2013-06-30 07:59:05 UTC+0000	
0x81d86980	wmlprvse.exe	568	868	8	181	0	0	2013-06-30 07:59:10 UTC+0000	
0x81fa33d8	TPAutoConnSvc.e	988	680	5	99	0	0	2013-06-30 07:59:10 UTC+0000	
0x822dc368	alg.exe	736	680	7	104	0	0	2013-06-30 07:59:10 UTC+0000	
0x821f2b28	TPAutoConnect.e	316	988	1	62	0	0	2013-06-30 07:59:11 UTC+0000	
0x82269780	wscntfy.exe	1244	1036	1	28	0	0	2013-06-30 07:59:12 UTC+0000	
0x82302350	cmd.exe	1504	1660	1	32	0	0	2013-06-30 07:59:27 UTC+0000	
0x82051da0	wuauclt.exe	1984	1036	8	175	0	0	2013-06-30 07:59:54 UTC+0000	

Figure 3: Volatility pslist plugin results for mem.dmp

The Analyst then utilized the connscan plugin for the Volatility Framework to obtain a list of active network connections contained within the memory capture image. The results of the connscan plugin revealed that the process associated with a process ID (PID) of 244 was connected to 67.206.91.17 over port 21. This information can be found within *Figure 4*. This information is relevant as the PID of FileZilla was 244, and port 21 is commonly utilized for FTP. This suggests that FileZilla was utilized for FTP purposes during the time of the memory capture.

```
brian@ubuntu:~/Desktop/Lab_2$ volatility -f mem.dmp --profile=WinXPSP2x86 connscan
Volatility Foundation Volatility Framework 2.6
```

Offset(P)	Local Address	Remote Address	Pid
0x0213d008	192.168.65.128:1026	67.205.91.17:21	244
0x02246008	192.168.65.128:1036	192.168.65.130:445	1164

Figure 4: Volatility connscan plugin results for mem.dmp.

After obtaining the TCP connections, the Analyst then obtained the executed console commands which were captured in the memory capture image. The Analyst obtained this information with the consoles plugin of the Volatility Framework. The results of the consoles plugin show that the suspect executed a batch file named script.bat from the C:\WINDOWS\system32\hidden\ directory. Upon executing this, the suspect was informed that five text files and one PDF were exfiltrated. The exfiltrated files can be seen highlighted in *Figure 5*.

```
Cmd #0 at 0x4f1fa0: C:\WINDOWS\system32\hidden\script.bat
----
Screen 0x4f2ab0 X:80 Y:300
Dump:

EDWARD RAINDEN'S CONFIDENTIAL INFORMATION EXTRACTION SCRIPT

STARTING EXTRACTION...
The command completed successfully.

\\NSA-SERVER-001\Users\erainden\Documents\TOP_SECRET_DOCS\Area_51.txt
\\NSA-SERVER-001\Users\erainden\Documents\TOP_SECRET_DOCS\List_of_Secret_Agents.
txt
\\NSA-SERVER-001\Users\erainden\Documents\TOP_SECRET_DOCS\Missile_Launch_Codes.tx
t
\\NSA-SERVER-001\Users\erainden\Documents\TOP_SECRET_DOCS\Nuclear.pdf
\\NSA-SERVER-001\Users\erainden\Documents\TOP_SECRET_DOCS\Safe_House_Locations.t
xt
\\NSA-SERVER-001\Users\erainden\Documents\TOP_SECRET_DOCS\Stuxnet_Program.txt
6 file(s) copied.
EXTRACTION COMPLETE!

SENDING HEROIC MESSAGE TO GOVERNMENT EMPLOYEES...
The message was successfully sent to domain NSA.

C:\Documents and Settings\Edward>
```

Figure 5: Console commands captured in mem.dmp.

Once the Analyst determined that the FileZilla application was utilizing an FTP connection with a public IPv4 address, the Analyst then proceeded to determine who was associated with the public IPv4 address. To do this, the Analyst utilized the resources at iplocation.net. The iplocation.net results for 67.205.91.17 can be seen in *Figure 5*. The results of iplocation.net showed that the 67.205.91.17 geolocated back to Canada. Additionally, the IPv4 address belonged to iWeb Technologies Inc. (iplocation.net, 2020)

Geolocation data from IP2Location (Product: DB6, updated on 2020-3-1)			
IP Address	Country	Region	City
67.205.91.17	Canada 🇨🇦	Quebec	Montreal
ISP	Organization	Latitude	Longitude
iWeb Technologies Inc.	Not Available	45.5088	-73.5878
Geolocation data from ipinfo.io (Product: API, real-time)			
IP Address	Country	Region	City
67.205.91.17	Canada 🇨🇦	Quebec	Notre-Dame-de-Grâce
ISP	Organization	Latitude	Longitude
iWeb Technologies Inc.	iWeb Technologies Inc. (iweb.com)	45.4594	-73.5501
Geolocation data from DB-IP (Product: Full, 2020-3-1)			
IP Address	Country	Region	City
67.205.91.17	Canada 🇨🇦	Quebec	Montreal
ISP	Organization	Latitude	Longitude
iWeb Technologies Inc.	FileGenie Inc	45.5017	-73.5673

Figure 6: Geolocation information for 67.205.91.17 (iplocation.net, 2020).

The Analyst then proceeded to obtain registry information from the memory capture image by utilizing the hivelist plugin. The Analyst can be seen utilizing the hivelist plugin in *Figure 6*.

```
brian@ubuntu:~/Desktop/Lab_2$ volatility -f mem.dmp --profile=WinXPSP2x86 hivelist
Volatility Foundation Volatility Framework 2.6
Virtual Physical Name
-----
0xe1eaf008 0x0d38d008 \Device\HarddiskVolume1\Documents and Settings\Edward\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat
0xe1ebc008 0x0d7b7008 \Device\HarddiskVolume1\Documents and Settings\Edward\NTUSER.DAT
0xe19b9760 0x0b82d760 \Device\HarddiskVolume1\Documents and Settings\LocalService\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat
0xe19a54f8 0x0b4e54f8 \Device\HarddiskVolume1\Documents and Settings\LocalService\NTUSER.DAT
0xe197eb60 0x0b3d4b60 \Device\HarddiskVolume1\Documents and Settings\NetworkService\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat
0xe18ef9c8 0x0aade9c8 \Device\HarddiskVolume1\Documents and Settings\NetworkService\NTUSER.DAT
0xe1830b60 0x0930eb60 \Device\HarddiskVolume1\WINDOWS\system32\config\software
0xe1822758 0x09156758 \Device\HarddiskVolume1\WINDOWS\system32\config\default
0xe1827b60 0x090edeb60 \Device\HarddiskVolume1\WINDOWS\system32\config\SAM
0xe155c758 0x09567758 \Device\HarddiskVolume1\WINDOWS\system32\config\SECURITY
0xe13d8b60 0x02e61b60 [no name]
0xe1035b60 0x02abbb60 \Device\HarddiskVolume1\WINDOWS\system32\config\system
0xe102e008 0x02ab5008 [no name]
brian@ubuntu:~/Desktop/Lab_2$
```

Figure 7: Volatility hivelist plugin results for mem.dmp.

The Analyst then used the hashdump plugin for the Volatility Framework to obtain any raw hashes from the mem.dmp. The hashdump plugin revealed that there were five hashes located which corresponded to account credentials. The hashes can be seen in *Figure 7*. The hash related

to the suspect's password can be seen outlined in *Figure 7*. The Analyst then output the hashes to a hash file named hashes.txt in the /Desktop/Lab_2/ directory.

```
brian@ubuntu:~/Desktop/Lab_2$ volatility -f mem.dmp --profile=WinXPSP2x86 hashdump
Volatility Foundation Volatility Framework 2.6
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:674f6dc2528d6640264b5b9be2ec691c:b184491432abbbd9a7756b5fa2eff83b:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:9f316b40ca9cee0d212e3c237c4979ff:::
Edward:1003:d277b934f8354db0aad3b435b51404ee:ceafbb1486b589426d0ffc095dec7c3e:::
brian@ubuntu:~/Desktop/Lab_2$
```

Figure 8: Volatility hashdump results for mem.dmp.

The Analyst was requested to provide a supplemental file containing the results of the various Volatility Framework plugins. The Analyst wrote a bash script to create the supplemental file. The script was named audit-log.sh, the source code of audit-log.sh can be seen in *Figure 8*.

```
#!/bin/bash
echo Hello, welcome to Brian's volatility script...
echo This script will obtain information from a memory image file by utilizing six volatility plugins.
#User input
read -p "Please enter the file name of the memory image you wish to analyze: " in_file
read -p "Please enter the name of the output file you wish to create: " out_file
read -p "Please enter the profile which should be used for the selected memory image: " profile
#Volatility commands sent to output file
volatility -f $in_file imageinfo >> $out_file
volatility -f $in_file $profile pslist >> $out_file
volatility -f $in_file $profile connscan >> $out_file
volatility -f $in_file $profile consoles >> $out_file
volatility -f $in_file $profile hivelist >> $out_file
volatility -f $in_file $profile hashdump >> $out_file
echo $out_file has been successfully created!
echo See you soon!
```

Figure 9: Bash script to create an audit file.

The bash script prompted the Analyst for the name of the input file which was to be processed by the various volatility plugins, the name of the output file, and the profile which should be used. The values which the Analyst had entered into the script can be seen outlined in *Figure 9*. After the script was run, the supplemental file was created. The supplemental file was named audit-20200328.txt.

```
brian@ubuntu:~/Desktop/Lab_2$ ./audit-log.sh
Hello, welcome to Brian's volatility script...
This script will obtain information from a memory image file by utilizing six volatility plugins.
Please enter the file name of the memory image you wish to analyze: mem.dmp
Please enter the name of the output file you wish to create: audit-20200328.txt
Please enter the profile which should be used for the selected memory image: WinXPSP2x86
Volatility Foundation Volatility Framework 2.6
INFO : volatility.debug : Determining profile based on KDBG search...
Volatility Foundation Volatility Framework 2.6
Volatility Foundation Volatility Framework 2.6
Volatility Foundation Volatility Framework 2.6
Volatility Foundation Volatility Framework 2.6
Volatility Foundation Volatility Framework 2.6
audit-20200328.txt has been successfully created!
See you soon!
brian@ubuntu:~/Desktop/Lab_2$
```

Figure 10: Creation of audit-20200328.txt.

Once the supplemental file containing the results from each volatility plugin was created, the Analyst then proceeded to crack the password for the suspect's account. To do this the Analyst first copied the hashes.txt file previously created to a Windows 10 VM which had Hash Suite installed. The Analyst then imported the hashes.txt file into Hash Suite, before allows the program to crack the hash. The Hash Suite results for hashes.txt can be seen in *Figure 10*. Additionally, it can be seen in *Figure 10*, that user Edwards password was OPEN.

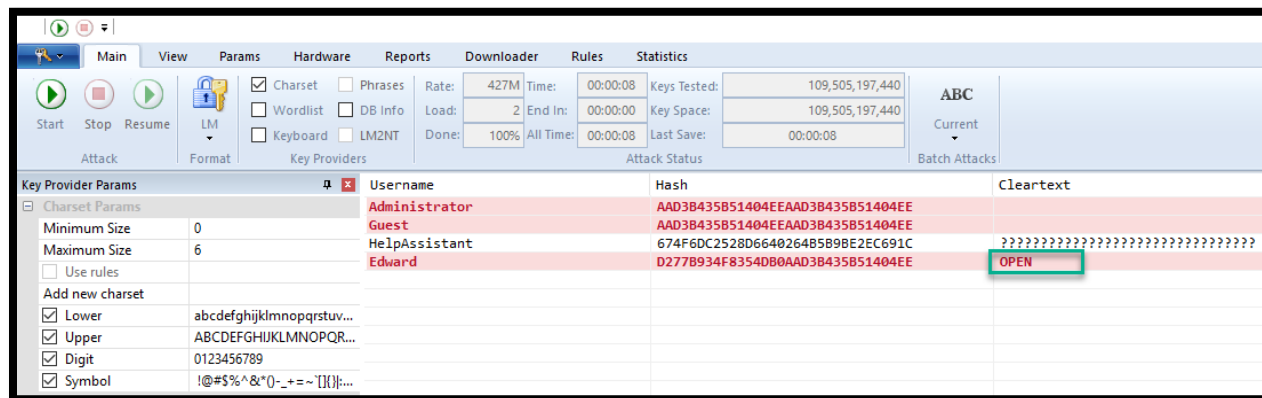


Figure 11: Hash Suite results for hashes.txt.

After determining that the suspect's password was OPEN, the Analyst then proceeded to use an additional online hash cracking tool named CrackStation from a VM with internet connectivity. In *Figure 11* it can be seen that CrackStation determined that the suspect's password hash corresponded to the password OPEN. (Defuse Security, 2019) This confirms that the suspect's password was OPEN. This concluded the Analyst's investigation.

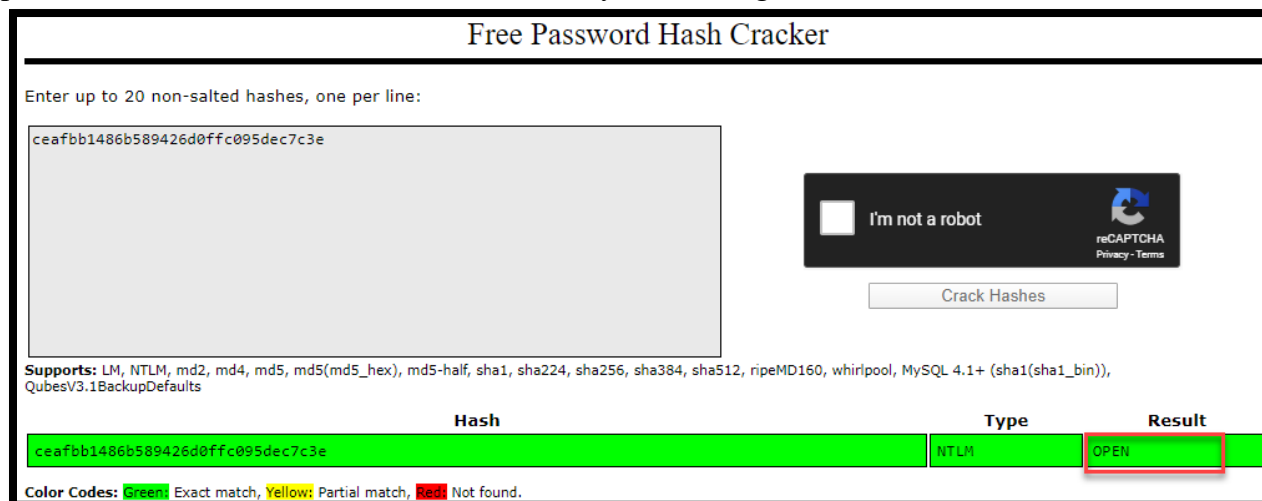


Figure 12: Edward Rainden's password hash verified with CrackStation. (Defuse Security, 2019)

Conclusion

The Office of General Counsel (OGC) requested that the Analyst analyze the memory capture image to determine or locate:

- The operating system of the computer.
- Which processes were running at the time of the memory capture.
- If any of those running processes could be used to exfiltrate data.
- What network TCP connections existed at the time of the memory capture.
- Where the TCP connections traced back to.
- Information related to any commands executed.
- Information related to the system registry.
- Any password hashes stored in the memory capture image.

The Analyst performed an analysis of the mem.dmp memory capture image and determined that the suspect was utilizing the FileZilla FTP client to exfiltrate data to 67.205.19.17. This IPv4 address corresponded to iWeb Technologies Inc., a company located in Quebec. The Analyst was able to successfully extract the suspect's password hash and decrypted it using Hash Suite, and CrackStation. The suspect's password was determined to be: OPEN. The Analyst located console information revealing that the suspect exfiltrated five text files and one PDF. Additionally, the Analyst created a supplemental file that contained the information requested by the OGC. This file was named audit-20200328.txt.

References

Defuse Security. (2019, May 27). *CrackStation*. Retrieved from CrackStation.net:

<https://crackstation.net/>

FileZilla. (2020). *Overview*. Retrieved from filezilla-project.org: <https://filezilla-project.org/>

iplocation.net. (2020). *Where is Geolocation of an IP Address?* Retrieved from iplocation.net:

<https://www.iplocation.net/>

Appendix

Appendix A: Analyst Workstation Specifications

Table 2 below outlines various Analyst workstations specifications.

Table 2: Analyst workstation specifications

System Info	Analyst's Machine Information
Computer Name:	CYB-Ubuntu
Operating System (OS) Name:	Ubuntu Linux
OS Version:	Version 6.1 (Build 7601: Service Pack 1)
System Make/Model:	MSI GS75-Stealth 9SF
System Serial Number:	9S717G111243ZJA000621
Time Zone of Analyst Machine:	(UTC-05:00) Eastern Time (US & Canada)
Automatically adjust clock for Daylight Saving Time Setting:	ENABLED
Analyst Time Verification:	System date/time is consistent with the time zone listed above, as verified by http://nist.time.gov/ .

Appendix B: Software Applications Utilized

Table 3 below outlines the various software applications that were utilized during the collection and analysis.

Table 3: Tools used for collection and analysis

Software Application	Version	Tool Website/Download Location
HashSuite	3.5.1	https://hashsuite.openwall.net/
IPlocation	N/A	https://iplocation.net
CrackStation	N/A	https://crackstation.net/
Volatility	2.6	https://github.com/volatilityfoundation/volatility