# Email Analysis

The following files were found on Thunderbird Profile:

|  |  |  |
| --- | --- | --- |
| File Name | Type | MD5 Checksum |
| INBOX.msf | Mozilla Mork database, version 1.4 | 02713a86ad1d35ddc732e7b7e34a1462 |
| INBOX | ASCII text | 23f181d07b615d61482404f1f2f29b2e |

The following message could be recovered:

|  |
| --- |
| To: "jones.sally1993@gmail.com" <jones.sally1993@gmail.com>  From: Biochemistry Campus IT Department <jason\_halloween@protonmail.com>  Reply-To: Biochemistry Campus IT Department <jason\_halloween@protonmail.com>  Subject: Important Security Update  Date: Mon, 12 Nov 2018 16:53:10 +0000 |
| Dear user,  We have been informed of a vulnerability on the workstations connected to our campus network. This vulnerability, which is tied to improper network configurations, has been rated "Critical". If left unattended, it is likely that your personal information will be leaked to third-parties.  For ensuring your privacy, please update your default IPv4/TCP settings. You may do so automatically by executing the patch located in the attachments of this message.  -- Sid Wilkes  Technical Support - Information Technology Department |
| Attachment:  A single file named “main” which appears to be:   |  | | --- | | ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32, BuildID[sha1]=28ba79c778f7402713aec6af319ee0fbaf3a8014, stripped | | **MD5 Checksum**: 324ddc336159dd62e182e3abf12c9b0a | |

Artifact 1: A suspicious message found at Sally's Inbox

We were able to extract the main file from the email.

This message is suspicious, because the sender name doesn’t match the email address.

The team decided to carefully analyze the message’s header in order to extract more information about the sender identity.

It seems like the sender use **ProtonMail ISP** which claims to protect the privacy of the users: By default, we do not keep any IP logs which can be linked to your anonymous email account. For more detailed approach on this email analysis see Header Analysis.

A curious fact is that a file with the same name exists on Downloads folder (under Sally’s Home Directory). We checked if the file is the same as the attachment as it’s very likely that Sally downloaded it.

|  |  |  |
| --- | --- | --- |
| File Name | Type | MD5 Checksum |
| main | XML 1.0 document, ASCII text | 1b38180828dbef5dc4a7e32589def5ae |

Artifact 2: Files found on Sally's Download folder

The main file found at Sally’s Download folder appears to be a XML file with some configuration about the desktop environment, thus revealing no interest to this investigation. See more information on Memory Analysis section at the bottom of this page.

## Header Analysis

In order to obtain more information about the attacker we decided to take a close look at the message header.

|  |
| --- |
| **Received**: from mail-40136.protonmail.ch (mail-40136.protonmail.ch. [**185.70.40.136**])  by mx.google.com with ESMTPS id n7-v6si8421705wma.39.2018.11.12.08.53.26  for <jones.sally1993@gmail.com>  (version=TLS1\_2 cipher=ECDHE-RSA-AES128-GCM-SHA256 bits=128/128);  Mon, 12 Nov 2018 08:53:27 -0800 (PST)  Received-SPF: pass (google.com: domain of jason\_halloween@protonmail.com designates **185.70.40.136** as permitted sender) client-ip=**185.70.40.136**; |

Evidence 1: First received SMTP stamp on malicious email [Artifact 1]

This reveals a dead end because the mail provider guarantees the privacy of the sender by removing the client source IP. We can only backtrack the email sender back to the ProtonMail’s ISP. A possible step is to request a warrant to get the information stored at ISP based on Message ID. It is possible that ISP store the clients IP and links them to the Messages ID but not expose that information publicly.

# Memory Analysis

Using the tool Volatility, we checked for malicious process running on Sally’s system and searching for suspicious processes.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Offset | Name | Pid | PPid | Uid | Start Time UTC +0000 |
| 0xffff91d5b56b5b00 | main | 14919 | 1211 | 1000 | 2018-11-12 17:15:45 |
| 0xffff91d5b56b0000 | main | 14921 | 14919 | 1000 | 2018-11-12 17:15:45 |

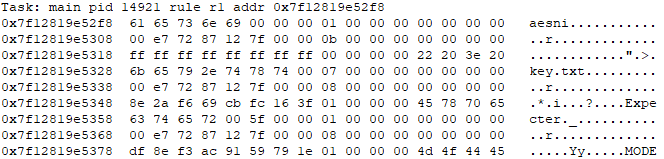
Artifact 3: Suspicious process running on Sally's computer

These two main processes started at the same time, and one was created by the first one. The Parent Process Id (**PPid**) shows this.

Using Volatility tool, we were able to discover that the main process was on Sally’s Downloads folder. We suspect that the file was replaced because the file types and checksums don't match and fls tool shows that this file was realloc.

|  |  |  |  |
| --- | --- | --- | --- |
| Pid | Uid | Gid | Path |
| 14919 | 1000 | 1000 | /home/sally/Downloads/main |
| 14921 | 1000 | 1000 | /home/sally/Downloads/main |

We started by searching the two processes IDs in the memory dump looking for the string “**aes**”. Our goal is trying to get the key from the memory. We know that malware was running during the memory snapshot, so it is very likely that the key is stored plaintext somewhere in the memory.

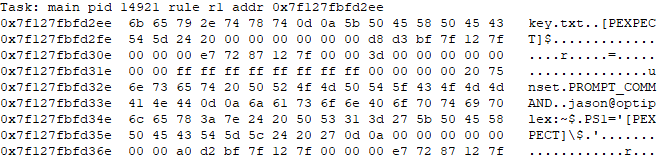


Artifact 4: Search results of AES in memory dump

(You can check the entire search result at file **yarascan.14921.txt**)

We suspect that the key was stored (even temporarily) on a file named “key.txt”.

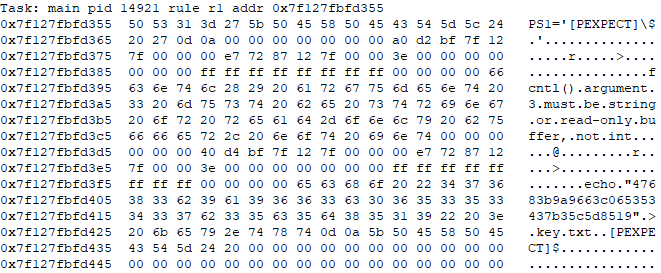
Next step is to search for that file in memory dump. Using a YaraScan Volatility feature, we look for “**key.txt**” in all memory (and not in particular process ID), and the following result appears:



Artifact 5: “key.txt” Search results in memory dump

(You can check the entire content of this result at file **yarascan.key.txt**)

We also found what appears to be a UNIX based bash prompt “jason@optiplex:~$ ” Our first guest relies on the hypothesis that **PS1** is a **bash** **environment** **variable**, so we decided to scan the memory looking for the content of this mysterious variable.



Artifact 6: Results of PS1 search on memory

*(You can check the entire content of this result at file* ***y*arascan.PS1.14921.txt**)

And Jackpot! We found interesting results **[Artifact 6]**. The **echo** command found contains, potentially, the key used by the ransomware to encrypt Sally’s documents.

|  |
| --- |
| 47 68 3b 9a 96 63 c0 65 35 34 37 b3 5c 5d 85 19 |

Evidence 2: AES symmetric key

The key has exactly **16 bytes**, which corresponds to **128-bit key** that we were trying to find.

The team put effort on developing a script to recover the files. We found a script on the Internet[[1]](#footnote-2) that allowed us to recover the encrypted files.

We changed the script to ignore the first 128 bits of each encrypted file and used those bits as the initial value for the counter.

|  |  |
| --- | --- |
| File Name | MD5 Checksum |
| AS\_09125\_050118150001\_A03f00d0.png.encrypted | 6e684be3134831bd07b81b165e28010f |
| AS\_09125\_050118150001\_A03f01d0.png.encrypted | de7674d7a23907429d648b4e0eeec6f6 |
| AS\_09125\_050118150001\_A03f02d0.png.encrypted | fed81b29f036d7999e46d4c09821980a |
| AS\_09125\_050118150001\_A03f03d0.png.encrypted | 2020b659af4a0b68c43cb34564ce9b3e |
| AS\_09125\_050118150001\_A03f04d0.png.encrypted | b3bbe351f2331f742391a3986eab8397 |
| AS\_09125\_050118150001\_A03f05d0.png.encrypted | 3910fae2d5659c9ada25d71e8e1f3cde |
| Image\_Processing\_with\_ImageJ.pdf.encrypted | add2eb2adabae079a7bfa3f59baf9235 |
| paper\_draft.txt.encrypted | 284638519f010804036a263f03423a1e |

Evidence 3: List of encrypted files

Executing the following command allowed us to recover all encrypted files:

|  |
| --- |
| python aes-ctr.py -d -i ../**ENCRYPTED\_FILE.png.encrypted** -o ../**FILE.png** -k **AES\_KEY** -iv **COUNTER** |

Command 1: Example of decryption tool usage

|  |  |
| --- | --- |
| File Name | MD5 Checksum |
| AS\_09125\_050118150001\_A03f00d0.png | b58303dd6f4026663fb1aacaccf5bf94 |
| AS\_09125\_050118150001\_A03f01d0.png | 1e33b87269c463474f68df10d95eb67b |
| AS\_09125\_050118150001\_A03f02d0.png | defa8c84d13338cf83668cf44ccbe016 |
| AS\_09125\_050118150001\_A03f03d0.png | 32de7caaac1e191febe5c7e4d48c839a |
| AS\_09125\_050118150001\_A03f04d0.png | 1a6093f96040770a97dd257a3d487231 |
| AS\_09125\_050118150001\_A03f05d0.png | f75baf3c3f4e06d14355133a6edae13b |
| Image\_Processing\_with\_ImageJ.pdf | 23f432689a13006cfe0e982f8ae71459 |
| paper\_draft.txt | aa4d4b8006c1941ffa3684f26747b696 |

Artifact 7: List of recovered files

## Malware analysis

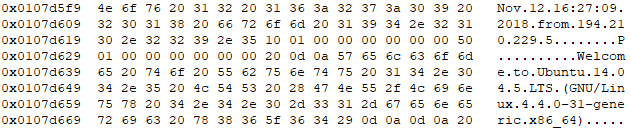
We suspect that the attachment (main) is a **remote access tool** (RAT) that allowed to wait and receive malicious instructions. This suspicion is based on test that we performed on the malware running on a controlled environment (VirtualBox). The malware opened two ports and did not performed encryption on baited PDF at Document Folder.

We tried to decompile the executable using Snowman tool, but no interesting results were found.

We also searched for malware evidence at: /tmp/\_MEIlXS6RU but again no relevant information was found.

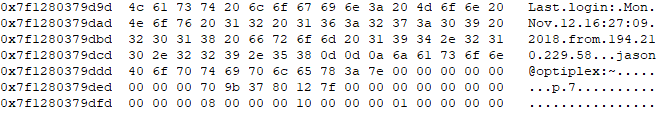
## Attacker Identity

Sally received an email from someone called Jason Halloween. We started by looking for “**jason**” in the memory dump.



Evidence 4: Results of "Jason" search on memory dump

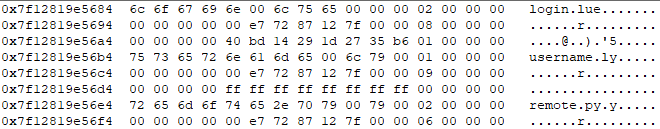
*(You can check the entire content of this result at file* ***volshell.lastlogin.txt***)



Evidence 5: SSH welcome message

This appears to be a welcome message that is present in most of Linux distributions when we connect via SSH.

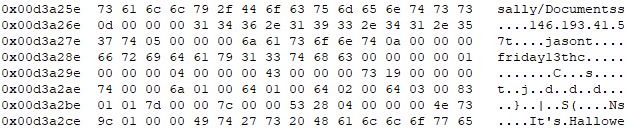
We were able to determine that Sally’s Linux Version is Ubuntu 16.04.5 LTS by extracting the /etc/issue file. We can now compare the two versions (Sally’s Version) with version found on **Evidence 4**. The IP address **194.210.229.58** is, probably, the last IP where **jason** logged in to that machine from his own computer! Let’s take a closer look using volshell:



Evidence 6: Results of "login" search in memory dump

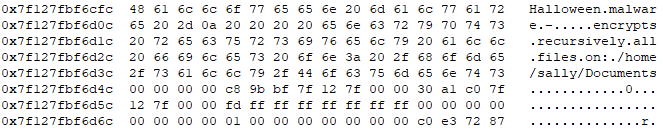
We even suspect that remote.py is a script that Jason used to connected to Sally’s PC and get along with the encryption process.

We decided to try some combination of useful keywords in the context of main executable, and more interesting results appears:



Evidence 7: Result of "sally/" search in memory dump

*(You can check the entire content of this result at file* ***Sallybarra.14921*.txt**)



*(You can check the entire content of this result at file* ***Halloween.14921*.txt**)

Evidence 8: Results of "Halloween" search in memory dump

We suspect that the malware was communicating with the **146.193.41.57** IP address. This is the first hint that leads to Jason’s real identity. The following steps are discovering the ISP of this address and request a warrant for discover who is Jason.

1. <https://github.com/rdomanski/AES-CTR> [↑](#footnote-ref-2)