Cyber Defense Project

Name: Abel Benedict
Enroll-No: 20162171001
Class & Section: CS-74

Malware Analysis

Static Analysis

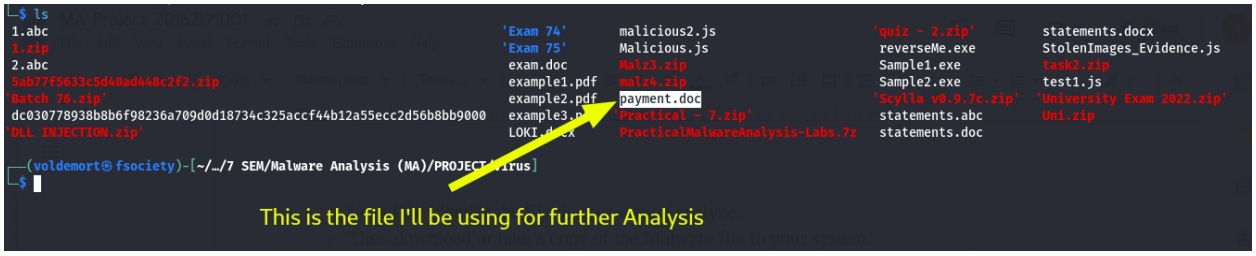
Here are the steps I'll be following throughout this Static Analysis.

- ☐ Obtain the Malware File
- ☐ Check File Information
- ☐ Extract Strings
- ☐ Analyze File Headers
- ☐ Perform Anti-Virus Scan

Obtain the Malware File:

- Let's first identify the file that we wanna analyze.
- Then download or take a copy of the Malware file to your system.

```
└─$ git clone https://github.com/kunaldg01/Virus.git
Cloning into 'Virus'...
remote: Enumerating objects: 156, done.
remote: Counting objects: 100% (74/74), done.
remote: Compressing objects: 100% (56/56), done.
remote: Total 156 (delta 21), reused 58 (delta 13), pack-reused 82
Receiving objects: 100% (156/156), 48.09 MiB | 2.76 MiB/s, done.
Resolving deltas: 100% (48/48), done.
```



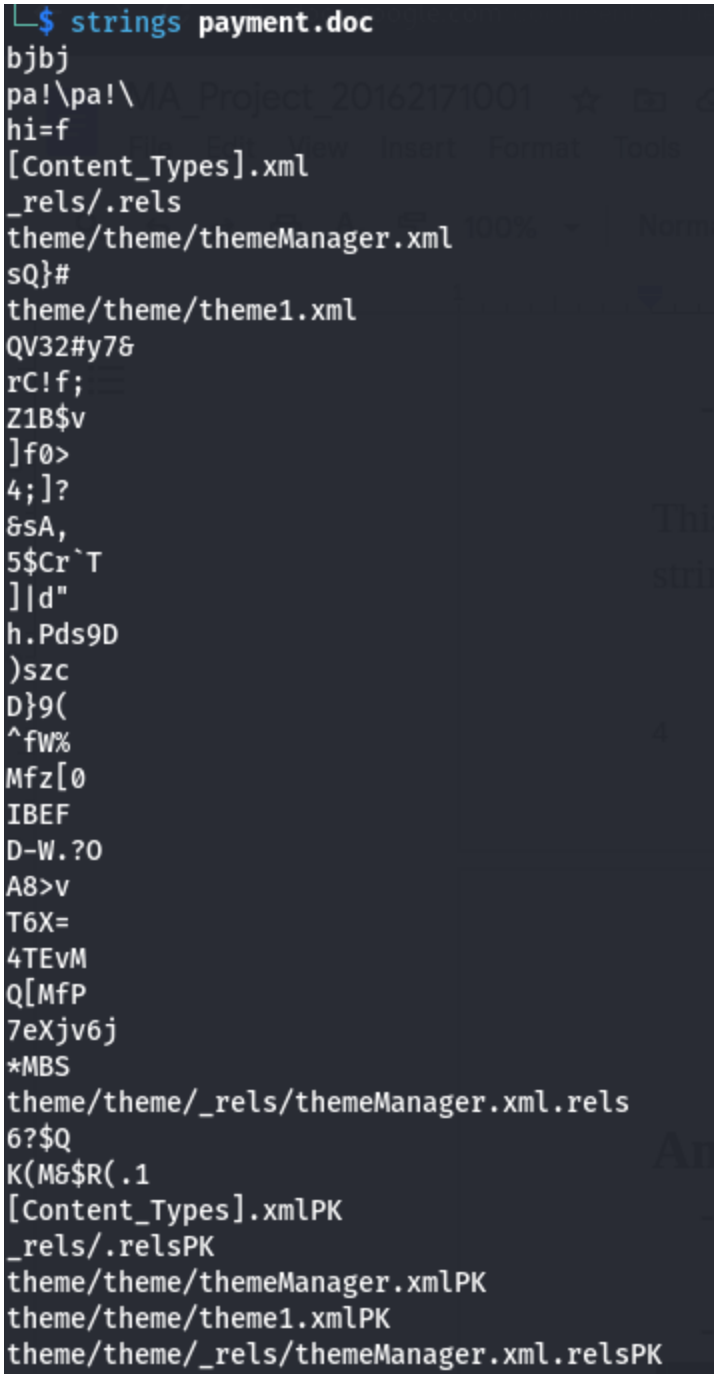
Check File Information:

- Use the "file" command to determine the file type and basic information.
- Run the following command.

```
└─$ file payment.doc
payment.doc: Composite Document File V2 Document, Little Endian, Os: Windows, Version 6.1, Code page: 1252, Author: admin, Template: Normal.dotm, Last Saved By: admin, Revision Number: 1, Name of Creating Application: Microsoft Office Word, Total Editing Time: 03:00, Create Time/Date: Wed Jun 29 17:09:00 2016, Last Saved Time/Date: Wed Jun 29 17:12:00 2016, Number of Pages: 1, Number of Words: 0, Number of Characters: 0, Security: 0
```

Extract Strings:

- Let's now extract readable strings from the file using the "strings" command.
- Run the following command.

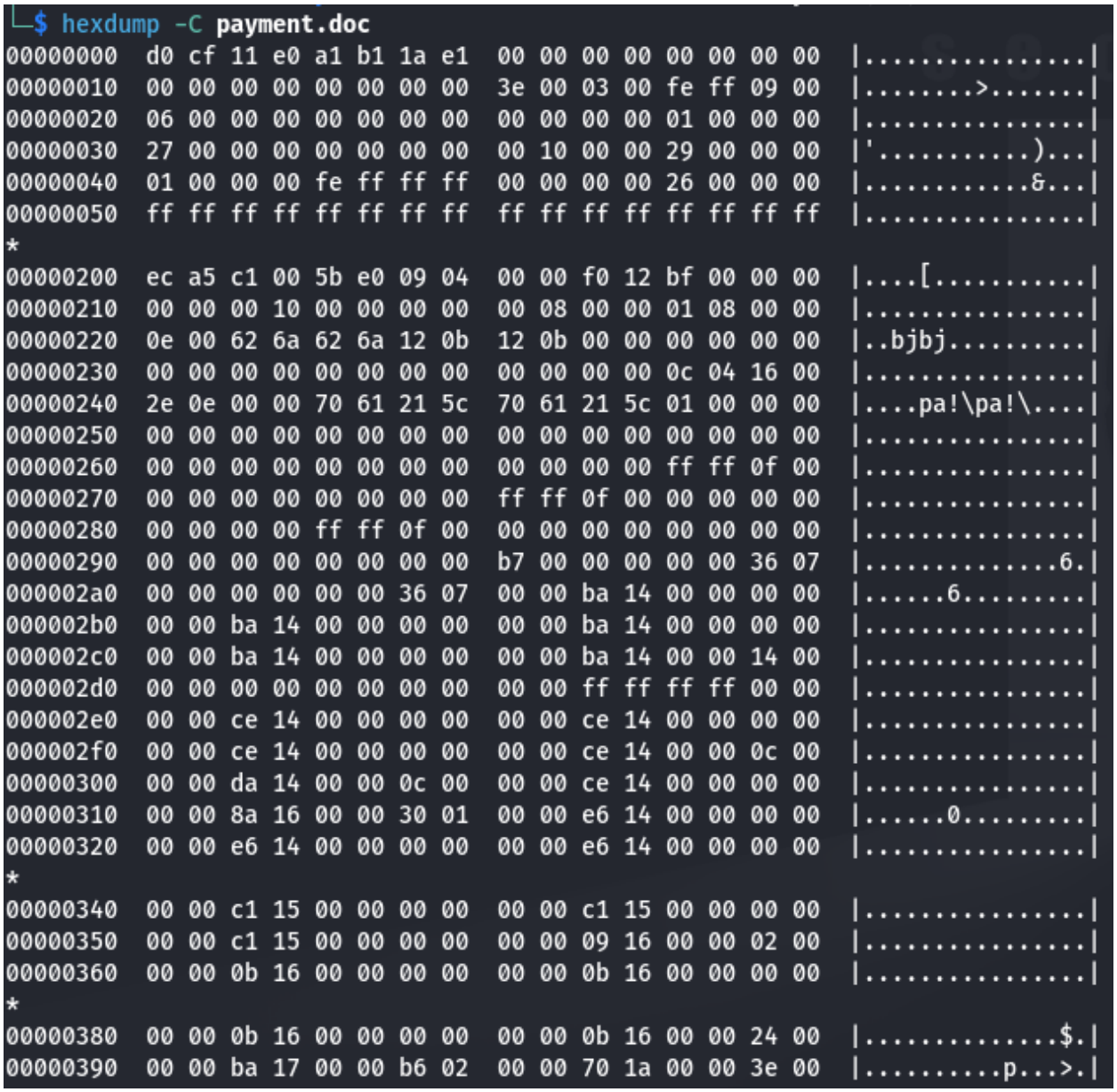


```
└─$ strings payment.doc
bjbj
pa!\pa!\
hi=f
[Content_Types].xml
_rels/.rels
theme/theme/themeManager.xml
sQ}#
theme/theme/theme1.xml
QV32#y7&
rC!f;
Z1B$v
]f0>
4;]?
&sA,
5$Cr`T
]|d"
h.Pds9D
)szc
D}9(
^fW%
Mfz[0
IBEF
D-W.?O
A8>v
T6X=
4TEvM
Q[MfP
7eXjv6j
*MBS
theme/theme/_rels/themeManager.xml.rels
6?$Q
K(M&$R(.1
[Content_Types].xmlPK
_rels/.relsPK
theme/theme/themeManager.xmlPK
theme/theme/theme1.xmlPK
theme/theme/_rels/themeManager.xml.relsPK
```

This output is too big to display on the terminal, so I've pasted the strings below.

Analyze File Headers:

- Use “hexdump” command to analyze the headers from this malware file.
- Run the following command.



```
└─$ hexdump -C payment.doc
00000000  d0 cf 11 e0 a1 b1 1a e1  00 00 00 00 00 00 00 00  |................|
00000010  00 00 00 00 00 00 00 00  3e 00 03 00 fe ff 09 00  |........>.......|
00000020  06 00 00 00 00 00 00 00  00 00 00 00 01 00 00 00  |................|
00000030  27 00 00 00 00 00 00 00  00 10 00 00 29 00 00 00  |'...........)...|
00000040  01 00 00 00 fe ff ff ff  00 00 00 00 26 00 00 00  |............&...|
00000050  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  |................|
*
00000200  ec a5 c1 00 5b e0 09 04  00 00 f0 12 bf 00 00 00  |....[...........|
00000210  00 00 00 10 00 00 00 00  00 08 00 00 01 08 00 00  |................|
00000220  0e 00 62 6a 62 6a 12 0b  12 0b 00 00 00 00 00 00  |..bjbj..........|
00000230  00 00 00 00 00 00 00 00  00 00 00 00 0c 04 16 00  |................|
00000240  2e 0e 00 00 70 61 21 5c  70 61 21 5c 01 00 00 00  |....pa!\pa!\....|
00000250  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000260  00 00 00 00 00 00 00 00  00 00 00 00 ff ff 0f 00  |................|
00000270  00 00 00 00 00 00 00 00  ff ff 0f 00 00 00 00 00  |................|
00000280  00 00 00 00 ff ff 0f 00  00 00 00 00 00 00 00 00  |................|
00000290  00 00 00 00 00 00 00 00  b7 00 00 00 00 00 36 07  |..............6.|
000002a0  00 00 00 00 00 00 36 07  00 00 ba 14 00 00 00 00  |......6.........|
000002b0  00 00 ba 14 00 00 00 00  00 00 ba 14 00 00 00 00  |................|
000002c0  00 00 ba 14 00 00 00 00  00 00 ba 14 00 00 14 00  |................|
000002d0  00 00 00 00 00 00 00 00  00 00 ff ff ff ff 00 00  |................|
000002e0  00 00 ce 14 00 00 00 00  00 00 ce 14 00 00 00 00  |................|
000002f0  00 00 ce 14 00 00 00 00  00 00 ce 14 00 00 0c 00  |................|
00000300  00 00 da 14 00 00 0c 00  00 00 ce 14 00 00 00 00  |................|
00000310  00 00 8a 16 00 00 30 01  00 00 e6 14 00 00 00 00  |......0.........|
00000320  00 00 e6 14 00 00 00 00  00 00 e6 14 00 00 00 00  |................|
*
00000340  00 00 c1 15 00 00 00 00  00 00 c1 15 00 00 00 00  |................|
00000350  00 00 c1 15 00 00 00 00  00 00 09 16 00 00 02 00  |................|
00000360  00 00 0b 16 00 00 00 00  00 00 0b 16 00 00 00 00  |................|
*
00000380  00 00 0b 16 00 00 00 00  00 00 0b 16 00 00 24 00  |..............$.|
00000390  00 00 ba 17 00 00 b6 02  00 00 70 1a 00 00 3e 00  |..........p...>.|
```

Because the output is too big, I couldn’t show everything in the screenshot.

Let’s now look at the file headers based on this hexdump.

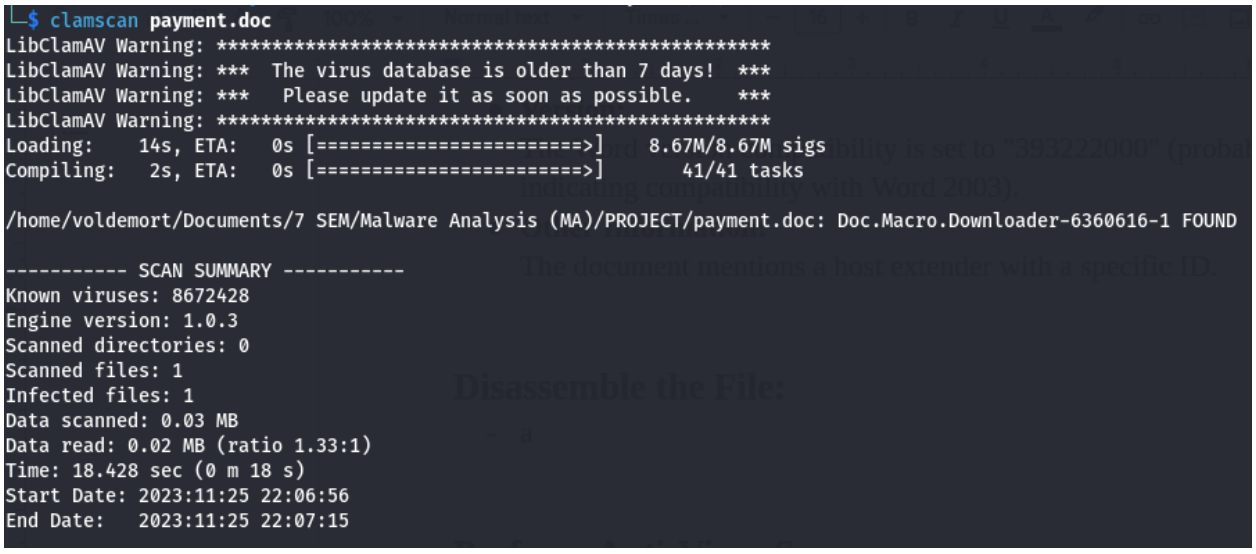
- **File Type:**

The file begins with the signature "D0CF11E0A1B11AE1" (not visible in the provided snippet, but common for MS Office files). The document type is identified as "Microsoft Word 97-2003 Document" in the ASCII representation.

- **Structure:**
  The file structure includes information about the Word document, such as version compatibility, CMG, DPB, GC, and Host Extender Info.
- **Version:**
  The Word version compatibility is set to "393222000" (probably indicating compatibility with Word 2003).
- **Other Information:**
  The document mentions a host extender with a specific ID.

Perform Anti-Virus Scan:

- I'll be running an anti-virus scan on the malware file using a command-line antivirus scanner like "clamscan".
- Run the following command.



Dynamic Analysis

Here are the steps I'll be following throughout this Dynamic Analysis.

- ☐ Set up a Sandbox Environment
- ☐ Monitor System Activities
- ☐ Analyze Process Activities
- ☐ Monitor File System Changes
- ☐ Capture System Memory

Set up a Sandbox Environment:

- Let's create a virtual machine to isolate the malware and protect the host system.

Monitor System Activities:

- I'll be using tools like "strace" or "ltrace" to monitor system calls and library calls made by the malware.
- Run the following command.

soffice.bin

```
strace /usr/lib/libreoffice/program/soffice.bin > strace_soffice.log
```

The output is much larger.

The summary of this log file is:

1. The malware attempts to connect to a UNIX socket at "/var/run/nscd/socket", but it fails due to "No such file or directory" (ENOENT).
2. It reads the "/etc/nsswitch.conf" file and "/etc/passwd" file, performing file operations like opening, reading, and closing.
3. It checks the existence of directories such as "/home/user/.config/libreoffice/4" and "/tmp".
4. It tries to launch "javaldx", but there's a warning that if failed, indicating a potential issue with java functionality.
5. The malware makes use of pipes and clones processes.

To monitor the system calls and library calls in real-time, use the following "strace" command.

```
strace -o output.txt -e trace=all -f -p 14845
```

Here, I've given the <PID> of soffice.bin process.

oosplash

1. **File Operations:**
    - Opened files using '**openat**' and checked file status using '**newfstatat**'.
    - Read content from files using '**read**'.
    - Closed files with '**close**'.
    - Accessed user-related information in "/etc/passwd".
2. **File System Operations:**
    - Created directories with '**mkdir**'.
    - Checked file existence and permissions with '**access**'.
    - Investigated symbolic links with '**readlink**'.
3. **Socket Operations:**
    - Created a Unix domain socket using '**socket**'.

- Attempted to connect to a socket file in "/tmp", but failed ('**connect**').

4. **Process and Thread Operations:**
   - Used '**clone3**' to create new process/thread.
   - Various operations related to process signals ('**rt_sigaction**', '**rt_sigprocmask**', '**futex**', '**exit_group**').
5. **Memory Operations:**
   - Allocated memory using '**mmap**'.
   - Set memory protection with '**mprotect**'.
6. **Miscellaneous:**
   - Checked user ID using '**getuid**'.
   - Checked and manipulated the current working directory with '**getcwd**'.

To monitor the system calls in real-time, we can use the following strace command.

```
strace -f -o strace_oosplash.log -p 14828
```

Here, I've given the <PID> of oosplash process.

Analyze Process Activities:

- Monitor the process activities of the malware using tools like "ps" or "top".
- Run the following command.

```
top
```

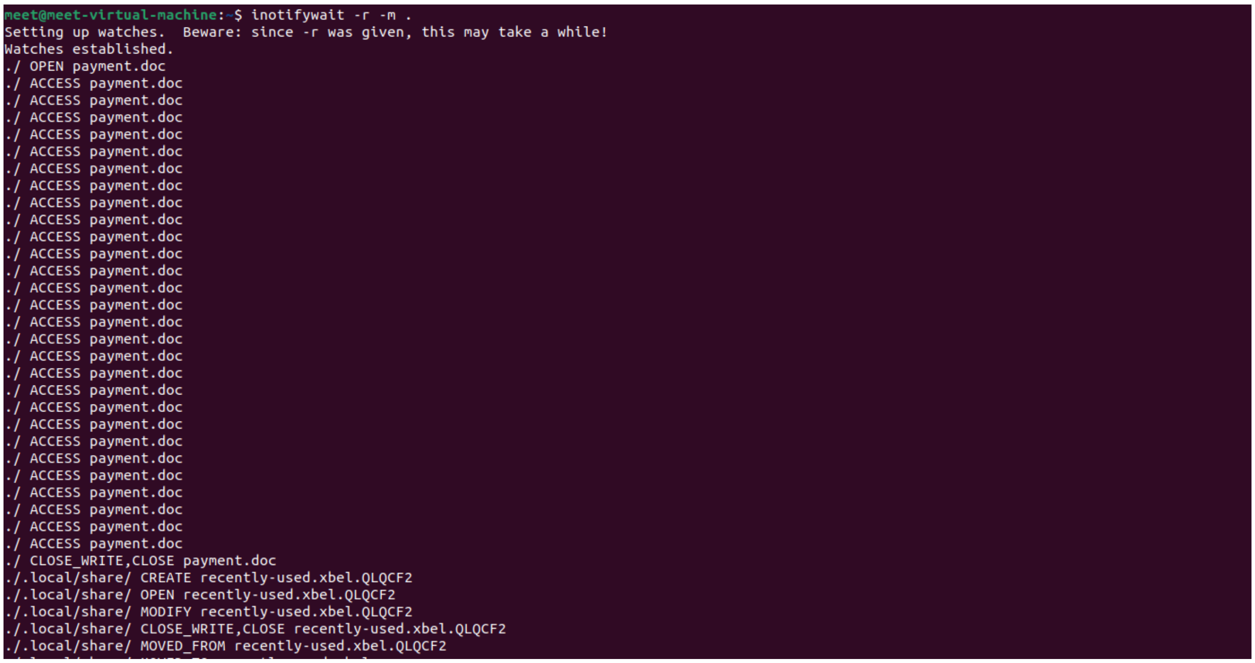
```
top - 23:57:23 up  2:11,  1 user,  load average: 0.19, 0.28, 0.42
Tasks: 297 total,   1 running, 295 sleeping,   0 stopped,   1 zombie
%Cpu(s): 21.0 us,  9.9 sy,  0.0 ni, 68.5 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
MiB Mem :   3871.6 total,   1137.9 free,   1612.0 used,   1121.7 buff/cache
MiB Swap:   3220.0 total,   3096.2 free,    123.8 used.   1967.9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  14845 meet      20   0 1034428 265044 140556 S  43.5   6.7   0:02.54 soffice.bin
   1246 meet      20   0 4514176 220808  88092 S  18.9   5.6   3:29.16 gnome-shell
   1108 meet       9 -11 1693072  19040  13664 S   1.0   0.5   0:31.82 pulseaudio
   1539 meet      20   0  323736  10668   5760 S   0.7   0.3   0:04.92 ibus-daemon
   1615 meet      20   0  356464  22928  12928 S   0.7   0.6   0:02.19 ibus-extension-
    352 root     -51   0       0      0      0 S   0.3   0.0   0:03.89 irq/16-vmwgfx
   1632 meet      20   0  297812  26584  17368 S   0.3   0.7   0:21.04 vmtoolsd
   9151 meet      20   0  576804  46516  31068 S   0.3   1.2   0:24.07 gnome-terminal-
  14173 root      20   0       0      0      0 I   0.3   0.0   0:01.40 kworker/0:3-events
  14812 meet      20   0   21884   4224   3328 R   0.3   0.1   0:00.39 top
      1 root      20   0  168128  11208   6216 S   0.0   0.3   0:04.66 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.05 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      8 root       0 -20       0      0      0 I   0.0   0.0   0:02.11 kworker/0:0H-kblockd
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     11 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_kthread
     12 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
     13 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
     14 root      20   0       0      0      0 S   0.0   0.0   0:02.65 ksoftirqd/0
     15 root      20   0       0      0      0 I   0.0   0.0   0:08.85 rcu_preempt
     16 root      rt   0       0      0      0 S   0.0   0.0   0:00.07 migration/0
     17 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
     19 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     20 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     21 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/1
     22 root      rt   0       0      0      0 S   0.0   0.0   0:00.87 migration/1
```

```
ps -ef
```

```
meet        1952    1094  0 21:46 ?        00:00:00 /usr/libexec/ibus-x11
root        2001       1  0 21:46 ?        00:00:07 /usr/libexec/fwupd/fwupd
meet        2673    1094  0 21:46 ?        00:00:00 /usr/bin/snap userd
meet        3312    1220  0 21:46 ?        00:00:00 update-notifier
root        4183       2  0 22:00 ?        00:00:01 [kworker/1:2H-kblockd]
meet        6426    1246  0 22:54 ?        00:00:03 gjs /usr/share/gnome-shell/extensions/ding@rastersoft.com/ding.js -E -P /usr/share/gnome-shell/ext
root        6673       1  0 23:03 ?        00:00:00 /usr/sbin/cupsd -l
root        6675       1  0 23:03 ?        00:00:00 /usr/sbin/cups-browsed
meet        8227    1094  0 23:04 ?        00:00:00 /usr/bin/gnome-calendar --gapplication-service
meet        8228    1094  0 23:04 ?        00:00:00 /usr/bin/seahorse --gapplication-service
meet        9151    1094  0 23:07 ?        00:00:23 /usr/libexec/gnome-terminal-server
meet        9774    1094  0 23:12 ?        00:00:26 /usr/bin/nautilus --gapplication-service
root       10873       2  0 23:18 ?        00:00:01 [kworker/1:2-events]
meet       11535    1094  0 23:22 ?        00:00:01 /usr/bin/speech-dispatcher -s -t 0
meet       11547   11535  0 23:22 ?        00:00:00 [sd_espeak-ng-mb] <defunct>
meet       11550   11535  0 23:22 ?        00:00:00 /usr/lib/speech-dispatcher-modules/sd_espeak-ng /etc/speech-dispatcher/modules/espeak-ng.conf
meet       11554   11535  0 23:22 ?        00:00:00 /usr/lib/speech-dispatcher-modules/sd_dummy /etc/speech-dispatcher/modules/dummy.conf
root       11964       2  0 23:24 ?        00:00:01 [kworker/u256:5-events_unbound]
meet       12047    9151  0 23:26 pts/1    00:00:00 bash
root       12081       2  0 23:26 ?        00:00:00 [tls-strp]
root       13214       2  0 23:30 ?        00:00:00 [kworker/1:0-cgroup_destroy]
root       13225       2  0 23:31 ?        00:00:01 [kworker/u256:0-flush-8:0]
root       13590       2  0 23:43 ?        00:00:00 [kworker/1:0H]
meet       13783    1246  0 23:43 ?        00:00:01 wireshark
root       13799       2  0 23:43 ?        00:00:00 [cfg80211]
root       14173       2  0 23:45 ?        00:00:01 [kworker/0:3-events]
clamav     14427       1  1 23:46 ?        00:00:11 /usr/bin/freshclam -d --foreground=true
root       14659       2  0 23:47 ?        00:00:00 [kworker/u256:2-events_unbound]
root       14706       2  0 23:50 ?        00:00:00 [kworker/0:0-events]
meet       14812   12047  0 23:55 pts/1    00:00:00 top
root       14819       2  0 23:56 ?        00:00:00 [kworker/0:1-events]
meet       14828    1094  0 23:56 ?        00:00:00 /usr/lib/libreoffice/program/oosplash --writer file:///home/meet/payment.doc
meet       14845   14828  4 23:56 ?        00:00:01 /usr/lib/libreoffice/program/soffice.bin --writer file:///home/meet/payment.doc
meet       14864    9151  0 23:56 pts/0    00:00:00 bash
meet       14871   14864  0 23:56 pts/0    00:00:00 ps -ef
meet@meet-virtual-machine:~$
```

Monitor File System Changes:

- Observe any changes made to the file system by the malware using tools like "inotifywait" or "auditctl".
- Run the following command.

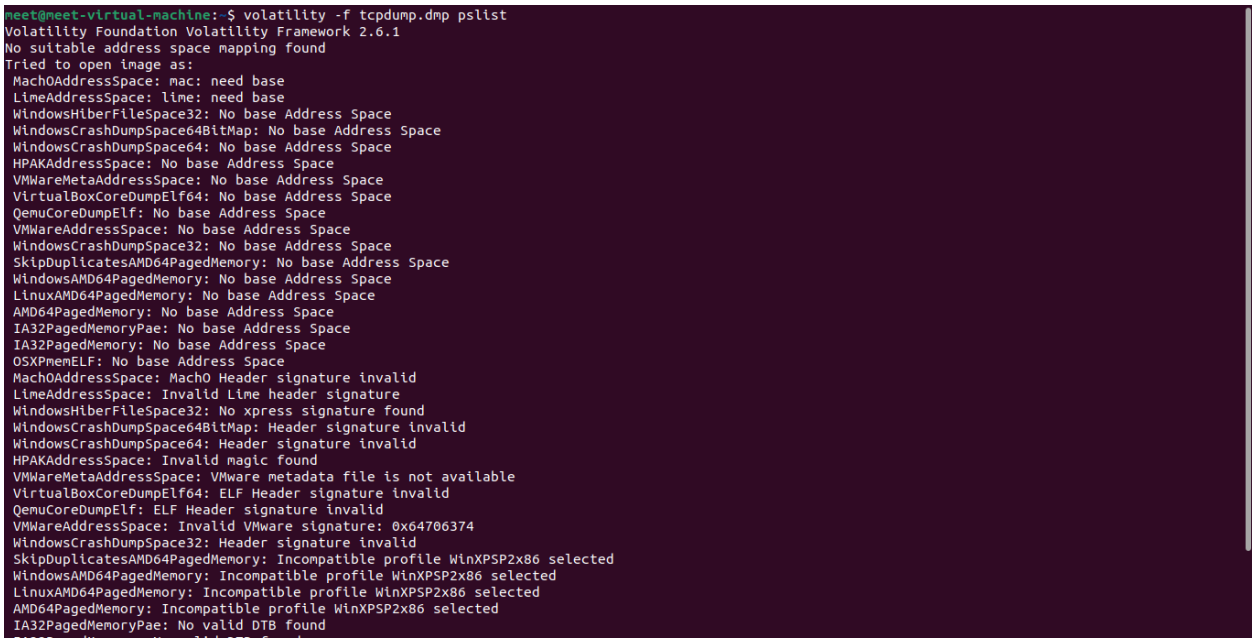


```
meet@meet-virtual-machine:~$ inotifywait -r -m .
Setting up watches.  Beware: since -r was given, this may take a while!
Watches established.
./ OPEN payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ ACCESS payment.doc
./ CLOSE_WRITE,CLOSE payment.doc
./.local/share/ CREATE recently-used.xbel.QLQCF2
./.local/share/ OPEN recently-used.xbel.QLQCF2
./.local/share/ MODIFY recently-used.xbel.QLQCF2
./.local/share/ CLOSE_WRITE,CLOSE recently-used.xbel.QLQCF2
./.local/share/ MOVED_FROM recently-used.xbel.QLQCF2
```

1. **Repetitive Changes:** The file "./ ACCESS payment.doc" is repeatedly accessed, suggesting potential modification or access.
2. **LibreOffice Configuration Changes:** There are multiple entries related to changes in LibreOffice configurations, such as modifications to toolbar, menubar, popupmenu, and statusbar.
3. **Recent Files:** The log shows activities related to recently-used.xbel, indicating modifications, movements, and attribute changes.
4. **User Configuration Changes:** Entries related to user configurations, like the creation and modification of files in the "./.config/libreoffice/4/user/config/" directory.

Capture System Memory:

- Capture and analyze the memory of the running malware using tools like "volatility" or "gdb".
- Run the following command.



Cyber Defense Techniques

Both in Windows & Linux Operating Systems

For Linux:

- ➢ Update your system regularly to ensure you have the latest security patches and updates.
  Command: `sudo apt update && sudo apt upgrade`
- ➢ Install and configure a firewall to monitor and control incoming and outgoing network traffic.
  Command: `sudo apt install ufw` (for Ubuntu-based systems)
  Command: `sudo ufw enable`
- ➢ Use strong passwords and enforce password policies to prevent unauthorized access.
  Command: `sudo passwd <username>` (to change user password)
  You can also use apg tool in Linux.
  The apg tool in Linux is a password generator that is used to generate strong and secure passwords. It stands for "**Automated Password Generator**." The apg tool uses a combination of various algorithms to generate random and complex passwords that are difficult to guess. It takes into consideration factors like password length, character set, and randomness to generate secure passwords. The generated passwords can be used for various purposes, including creating strong passwords for user accounts, securing network devices, or generating random encryption keys.
- ➢ Disable unnecessary services and remove any unnecessary software or packages.
  Command: `sudo systemctl disable <service>` (to disable a service)
- ➢ Regularly scan your system for malware using antivirus software.
  Command: `sudo apt install clamav` (for installing ClamAV)
  Command: `sudo clamscan -r /` (to scan the entire system)
- ➢ Monitor system logs for any suspicious activities or anomalies.

Command: `sudo tail -f /var/log/syslog` (to monitor system logs)

For Windows:

- ➢ Keep your operating system and applications up to date with the latest patches.
  Command: `wuauclt /detectnow` (to check for Windows updates)
- ➢ Install and configure a reputable antivirus software and keep it updated.
  Command: `Get-MpComputerStatus` (to check Windows Defender status)
- ➢ Enable and configure Windows Firewall to control inbound and outbound network traffic.
  Command: `Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled True`
- ➢ Regularly backup important files and data to prevent loss in case of an infection.
  Command: `robocopy <source> <destination> /MIR` (to create a mirror backup)
- ➢ Use strong and unique passwords for user accounts and enable multi-factor authentication where possible.
  Command: `net user <username> *` (to change user password)
- ➢ Enable and monitor Windows Event Logs for any suspicious activities or security events.
  Command: `Get-WinEvent -LogName Security` (to view security events)

Remember that these techniques should be implemented as part of a comprehensive cybersecurity strategy, and it's advisable to consult with a cybersecurity professional for a more tailored approach.