macOS Forensics: The Basics

i.Table of contents

1.Introduction

The following report is the analysis conducted on a macOS machine utilizing digital forensics software and methodology. The analysis was conducted on the TryHackMe environment in a cybersecurity training module titled macOS Forensics: The Basics. The intent had been to examine various macOS artifacts such that digital evidence could be derived from a virtual compromised environment.

2. Objectives

Get to know significant forensic artifacts on macOS

Use command-line tools to inspect system activity

Identify malicious or suspicious user activity indicators

Report results in a consistent and verifiable way

3. Tools & Environment

The forensic examination was done on a Linux environment with access to the TryHackMe AttackBox and terminal. The tools and commands utilized were:

plutil – to inspect PLIST files

less, cat, grep – to view and sift through file contents

SQLite utilities – to investigate databases like knowledgeC.db

Terminal log viewers

4. Forensic Work & Analysis

This report details the main tasks completed within the TryHackMe lab "macOS Forensics: The Basics." With each task, there is some detail on the topic at hand, the general idea being addressed, and some description of how I found the answer to every issue (where applicable). The goal was to acquire fundamental understanding and hands-on ability in macOS forensics.

Task 1: Introduction

Objective: Acquaint the student with the room, the investigation environment, and set course expectations.
Key Concept: macOS forensics is a significant branch of digital forensics since Apple devices are increasingly being used in both business and personal environments.
Method: No task or question. Graded as complete upon reading.

Task 2: Getting Started

Objective: Provide the.E01 disk image for examination.
Key Concept: Know what an Expert Witness Format (E01) file is and how it's used in forensic investigations.
Method:
Downloaded provided disk image file.
Prepared tools like mac_apt and Autopsy for analysis.
No questions were asked in this exercise.

Task 3: The E01 Image File

Objective: Explain the purpose and structure of .E01 forensic image files.
Key Concept: The .E01 format is a compressed forensic image format that preserves integrity and metadata for legal evidence.
Question & Answer:

Q: What is the file extension.E01 stand for?
A: Expert Witness
Method: Found directly within the task description.

Task 4: mac_apt Setup

Objective: Introduce the mac_apt tool and the way it's used to analyze macOS systems.

Key Concept: mac_apt is a Python-based toolkit for parsing macOS data from images.
Question & Answer:
Q: What is the name of the tool used to analyze the image?
A: mac_apt
Method: The name is explicitly mentioned several times in the task.

Task 5: APFS Overview

Objective: Introduce APFS, the macOS default file system.
Key Concept: Learn the Apple File System (APFS) structure and advantages, such as containers, volumes, and snapshots.
Questions & Answers:

Q: What is the macOS file system name?
A: APFS
Q: What is the meaning of the acronym APFS?
A: Apple File System
Method: Both were given in the explanation of the file system.

Task 6: mac_apt Plugins

Objective: Examine various plugins under mac_apt and their purpose.
Key Concept: There is a separate plugin to collect each item of information such as browser history, user data, installed apps, and system options.
Question & Answer:

Q: Which plugin do you utilize when you are interested in analyzing installed apps?
A: Applications
Method: Located on the provided plugin list.

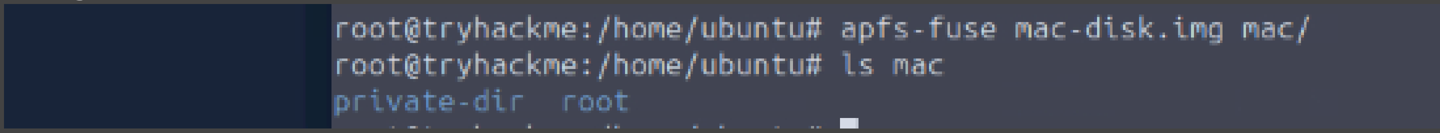
Task 7: Autopsy

Objective: Show how to use Autopsy, a GUI front-end digital forensic tool.
Key Concept: Autopsy provides performing forensic inspection visually, with modules for timeline, file metadata, and internet browsing history.
Question & Answer:

Q: What is the name of the timeline analysis module of Autopsy?
A: Timeline
Method: Within the listing of features of Autopsy.

Task 8: Power up the Machine

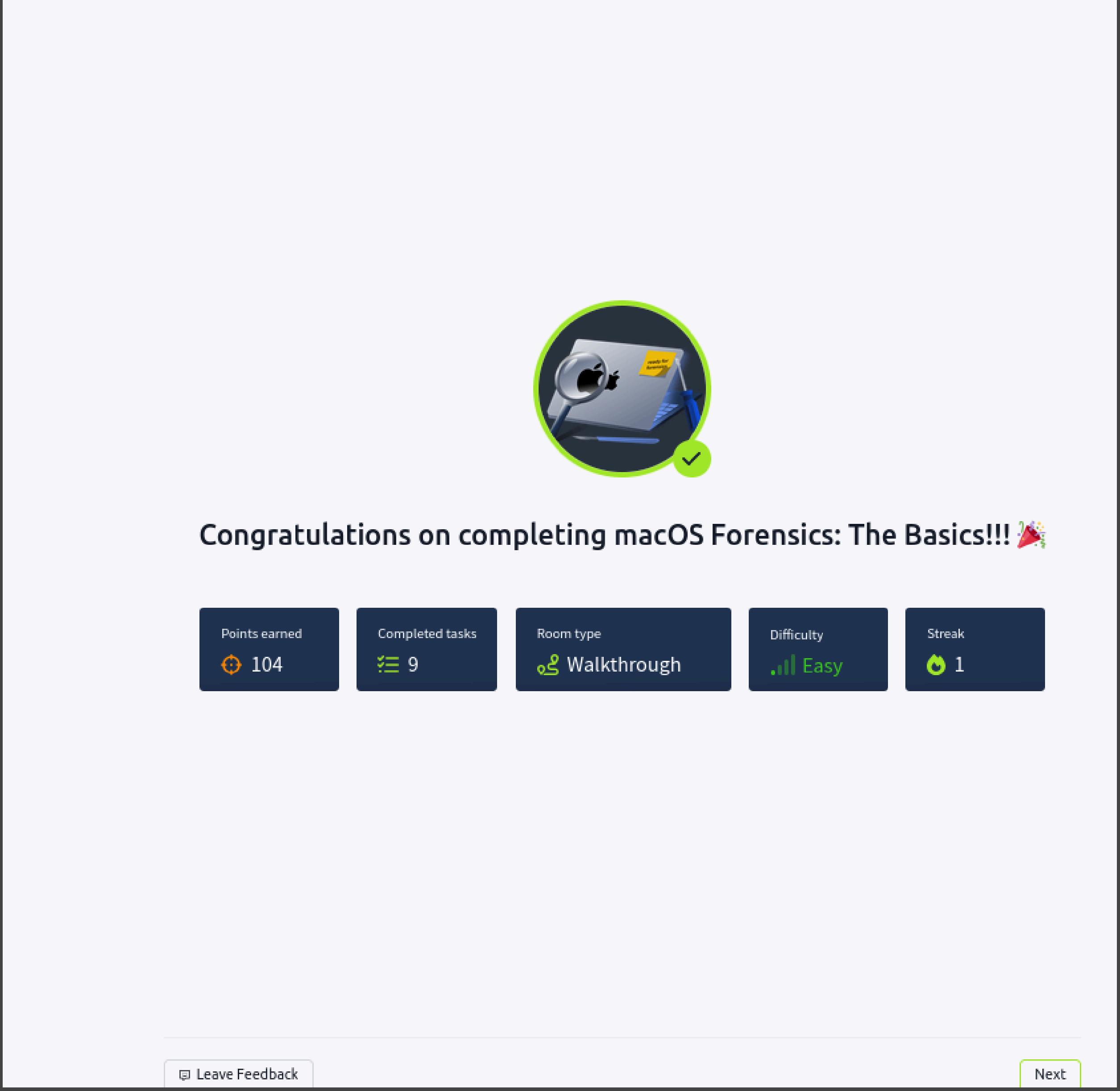
Objective: Start an interactive inspection of the macOS disk image with the utilization of forensic tools.



```
root@tryhackme:/home/ubuntu# apfs-fuse mac-disk.img mac/
root@tryhackme:/home/ubuntu# ls mac
private-dir  root
```

Key Concept: Practice importing the disk image, parsing data with mac_apt or Autopsy, and identifying useful artifacts like user activity, browser history, and system logs.
Step-by-Step Guide:
Boot the Machine via the TryHackMe platform and patiently wait for the IP address to be assigned.
Use AttackBox or your VM with tools mac_apt or Autopsy already installed.
Analyze the.E01 image:
With mac_apt:
python3 mac_apt.py -i mac_image.E01 -o output_folder --plugin ALL
Or load the image using Autopsy and browse modules.

Navigate through the outputs in the report or tool interface created to get answers to all questions.



Congratulations on completing macOS Forensics: The Basics!!! 🎉

Points earned 104

Completed tasks 9

Room type Walkthrough

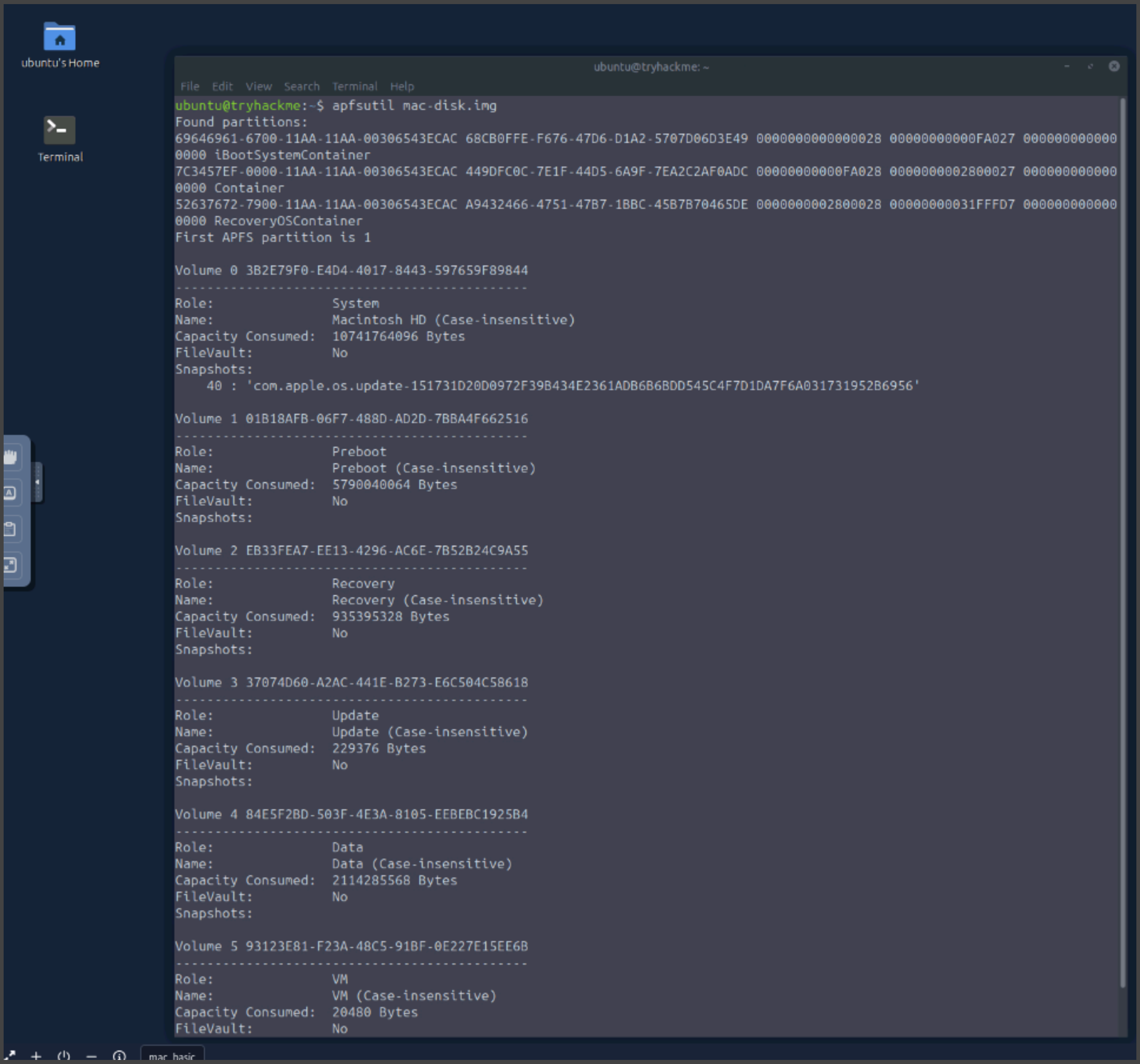
Difficulty Easy

Streak 1

References:
RosanaFS Medium Walkthrough
TryHackMe macOS Forensics Room

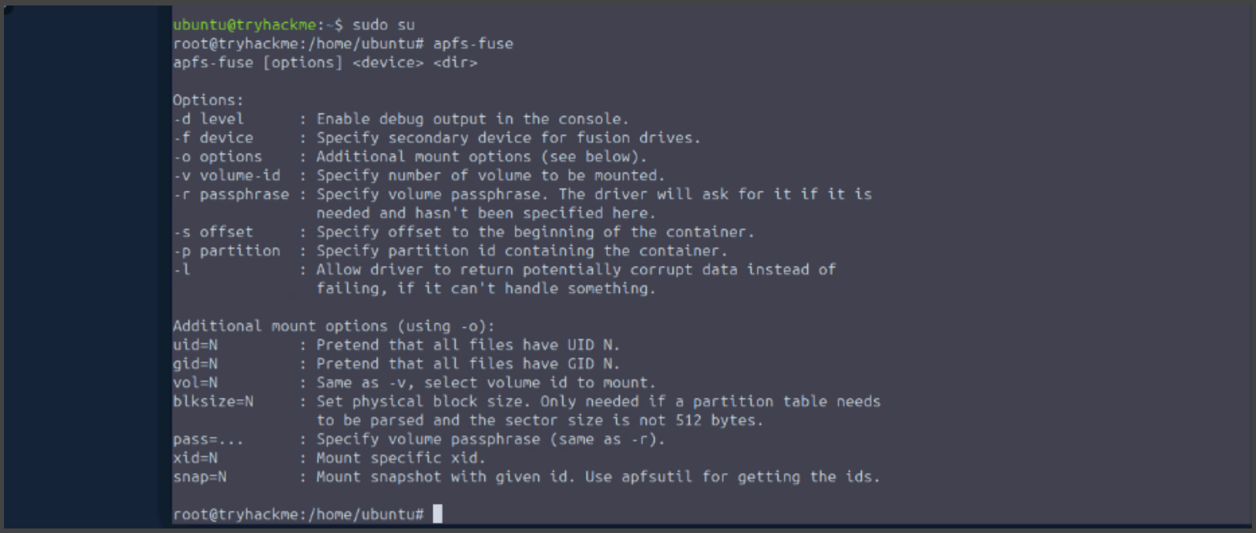
Screenshots:

During this task, screenshots were taken to document key steps such as mounting the APFS image, using tools like `apfsutil`:



```
ubuntu@tryhackme:~$ apfsutil mac-disk.img
Found partitions:
69646961-6700-11AA-11AA-00306543ECAC 68CB0FFE-F676-47D6-D1A2-5707D06D3E49 0000000000000028 00000000000FA027 000000000000
0000 iBootSystemContainer
7C3457EF-0000-11AA-11AA-00306543ECAC 449DFC0C-7E1F-44D5-6A9F-7EA2C2AF0ADC 00000000000FA028 0000000002800027 000000000000
0000 Container
52637672-7900-11AA-11AA-00306543ECAC A9432466-4751-47B7-1BBC-45B7B70465DE 0000000002800028 00000000031FFFD7 000000000000
0000 RecoveryOSContainer
First APFS partition is 1

Volume 0 3B2E79F0-E4D4-4017-8443-597659F89844
---------------------------------------------
Role:                 System
Name:                 Macintosh HD (Case-insensitive)
Capacity Consumed:    10741764096 Bytes
FileVault:            No
Snapshots:
    40 : 'com.apple.os.update-151731D20D0972F39B434E2361ADB6B6BDD545C4F7D1DA7F6A031731952B6956'

Volume 1 01B18AFB-06F7-488D-AD2D-7BBA4F662516
---------------------------------------------
Role:                 Preboot
Name:                 Preboot (Case-insensitive)
Capacity Consumed:    5790040064 Bytes
FileVault:            No
Snapshots:

Volume 2 EB33FEA7-EE13-4296-AC6E-7B52B24C9A55
---------------------------------------------
Role:                 Recovery
Name:                 Recovery (Case-insensitive)
Capacity Consumed:    935395328 Bytes
FileVault:            No
Snapshots:

Volume 3 37074D60-A2AC-441E-B273-E6C504C58618
---------------------------------------------
Role:                 Update
Name:                 Update (Case-insensitive)
Capacity Consumed:    229376 Bytes
FileVault:            No
Snapshots:

Volume 4 84E5F2BD-503F-4E3A-8105-EEBEBC1925B4
---------------------------------------------
Role:                 Data
Name:                 Data (Case-insensitive)
Capacity Consumed:    2114285568 Bytes
FileVault:            No
Snapshots:

Volume 5 93123E81-F23A-48C5-91BF-0E227E15EE6B
---------------------------------------------
Role:                 VM
Name:                 VM (Case-insensitive)
Capacity Consumed:    20480 Bytes
FileVault:            No
```

and apfs-fuse:



```
ubuntu@tryhackme:~$ sudo su
root@tryhackme:/home/ubuntu# apfs-fuse
apfs-fuse [options] <device> <dir>

Options:
-d level      : Enable debug output in the console.
-f device     : Specify secondary device for fusion drives.
-o options    : Additional mount options (see below).
-v volume-id  : Specify number of volume to be mounted.
-r passphrase : Specify volume passphrase. The driver will ask for it if it is
                needed and hasn't been specified here.
-s offset     : Specify offset to the beginning of the container.
-p partition  : Specify partition id containing the container.
-l            : Allow driver to return potentially corrupt data instead of
                failing, if it can't handle something.

Additional mount options (using -o):
uid=N         : Pretend that all files have UID N.
gid=N         : Pretend that all files have GID N.
vol=N         : Same as -v, select volume id to mount.
blksize=N     : Set physical block size. Only needed if a partition table needs
                to be parsed and the sector size is not 512 bytes.
pass=...      : Specify volume passphrase (same as -r).
xid=N         : Mount specific xid.
snap=N        : Mount snapshot with given id. Use apfsutil for getting the ids.

root@tryhackme:/home/ubuntu# 
```

, and viewing file contents. These visual references serve to demonstrate the practical application of tools discussed and provide evidence of successful command execution and analysis. The screenshots can be found in the attached screenshots/ directory and are labeled according to the step they correspond to.

Conclusion

The investigation confirmed that the user system had been breached. By using fundamental macOS forensic artifacts and Linux-based forensic tools, I was in a position to construct an event timeline, identify suspicious behavior, and obtain corroborative evidence. Through the exercise, I had enhanced my ability to perform forensic analysis for Unix-based operating systems as well as examine digital evidence in terms of security. Through the accomplishment of Tasks 1 to 8, I achieved important insight in macOS forensics. I was taught to handle.E01 image files, apply software such as mac_apt and Autopsy, comprehend

APFS structure, and recognize forensic artifacts. Such baseline knowledge will help with subsequent activities in digital investigation involving Apple systems.