Bsides DFIR Memory forensic challenge

Challenge overview

One of the systems compromised by the attacker was a workstation. The incidence response team extracted the Linux system for memory analysis.

Files Provided:

A memory-dump.bin

Expected tools to solve the challenge:

*Volatility

What is expected of the challenge player:

*Create a linux volatility profile to analyse the memory image with volatility.

*Answer questions as aided by Volatility

Questions:

Prior to building a profile, one is required to know the following details:

- linux distribution and version
- kernel version

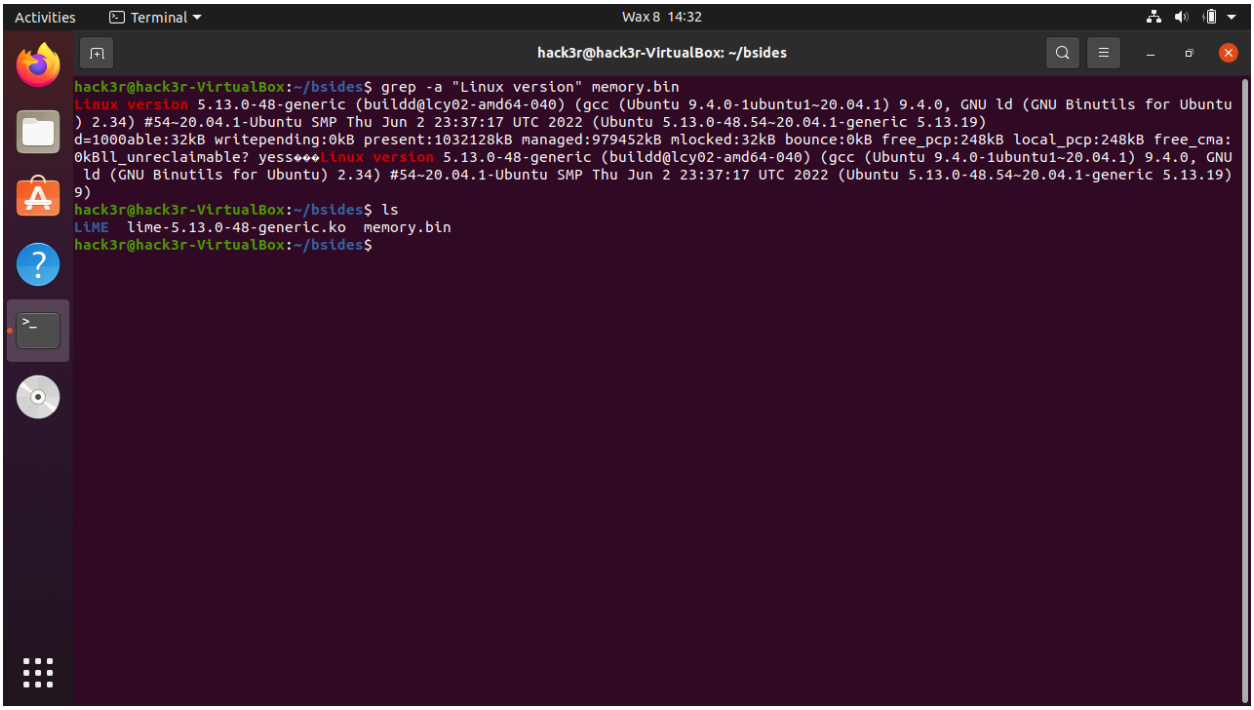
Question

*Identify the linux distro and version (ubuntu-20.04.1)

*Identify the kernel version = (5.13.0-48-generic)

This can be found by running:

grep -a “Linux version” memorydump.bin



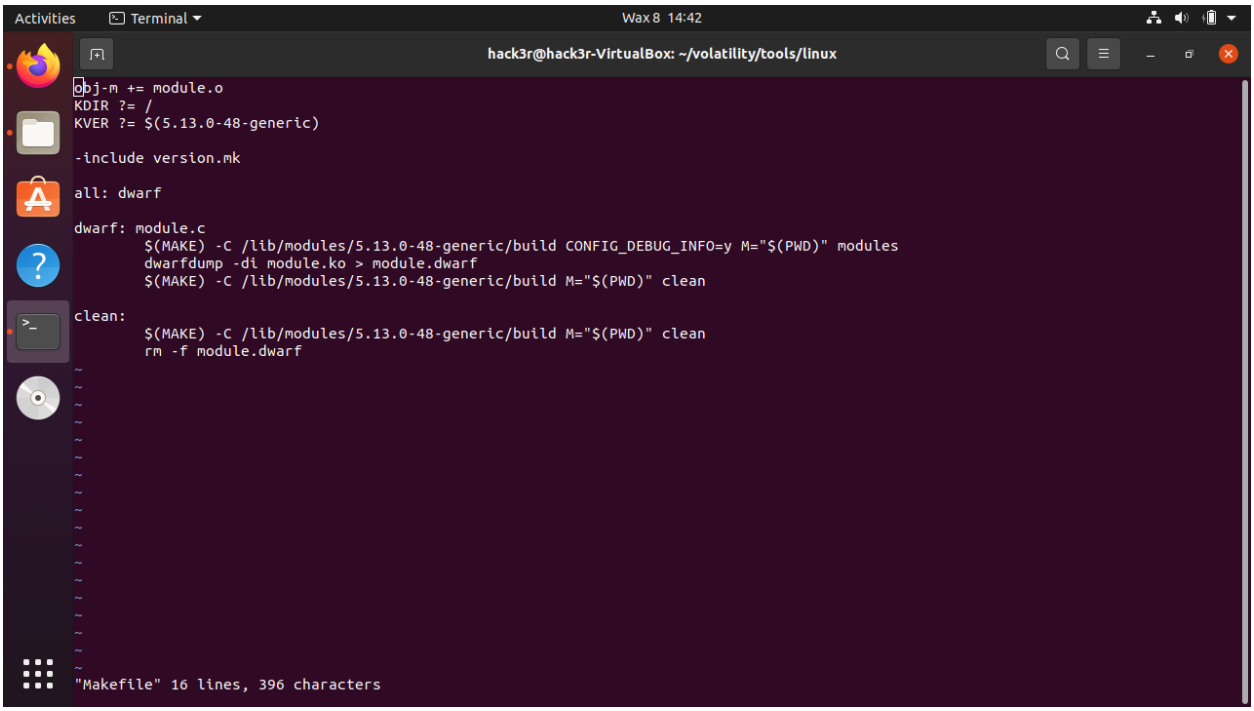
creating the volatility profile steps:

Fetch volatility from repo:

$ git clone https://github.com.volatilityfoundation/volatility

$cd volatility/tools/linux

change the kernel detection value in the Makefile to match the kernel version = 5.13.0-48-generic



When creating a linux volatility profile,the only components of the system we need are Linux headers and a system map which can be replicated from an ubuntu docker container.

Setup a container;

docker run -it —rm -v $PWD:/volatility ubuntu:20.04 bin/bash

Install the necessary packages to aid in the profile creation: run the following commands:

#cd volatility

#apt update

apt install build-essential linux-headers-4.15.0-112-generic dwarfdump make

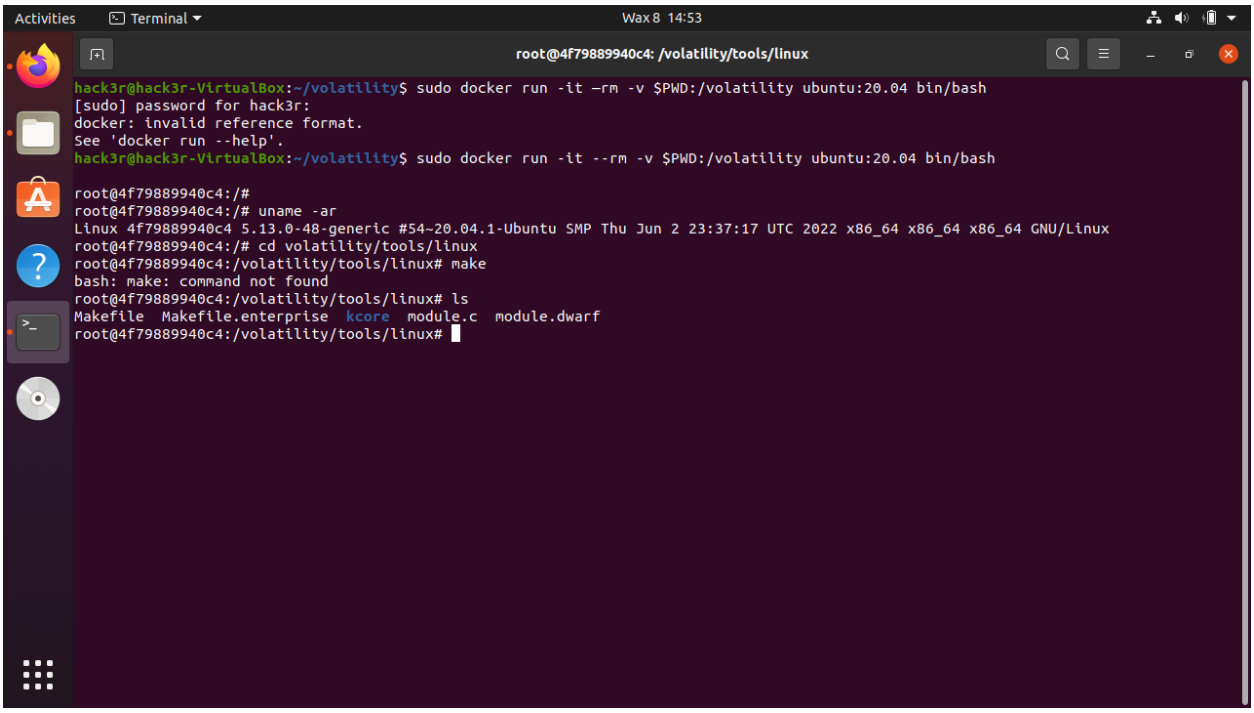
zip linux-image-4.15.0-112-generic

cd /volatility/tools/linux

create dwarf file:

#Make

There should be a dwarf file module.dwarf created



zip the dwarf file and system map

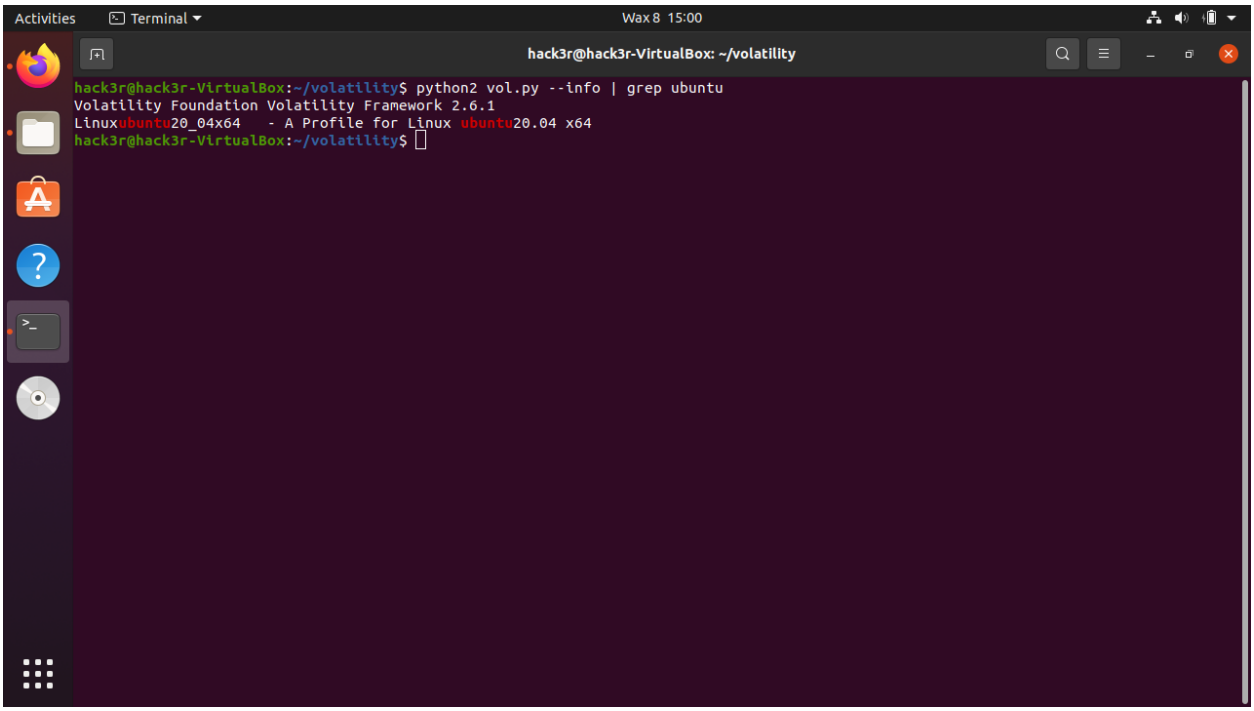
#zip ubuntu20.04.zip module.dwarf /boot/system.map-5.13.0.48-generic

exit the docker

#mv the profile.zip to volatility/plugins/overlays/linux

Volatility

Confirm that the profile can be detected



Questions

The volatility profile will be used to answer questions such as:

1. What malicious process was running on the workstation - use plugin linux_pslist and linux_pidhashtable

(This will be a netcat backdoor)

*PID of process

*Process name

*Time launched

2. What commands was executed by the attacker: - git command to download malware from repo

git clone <git url to python script that acts as a stager to download malware from pastebin>

3. What was the attackers entrypoint?

- The python command to run malware.py executed by attacker to download malware from pastebin.

4. We secured the workstation to keep the attacker out but there seems to be another way in, find out how?

- dump memory from bash process spawned by python command, command to add an ssh key to authorized_keys which will allow him to login without the system’s password

5. There is another backdoor to the system through a rootkit, find out the following info

*backdoor name - malicious module(**syshookmal.ko)**

*which syscall was hooked - **symlink(88)**

- use plugin linux_check_syscall to find which sys call was hooked. - the syscall will be symlink - 88 hooked by a malicious module syshookmal.ko(illegitimate module)