Assignment 6

Forensics

Computer Forensics (Motivation)

- search for evidence of computer activities in support of an investigation
- computing activity (criminal or otherwise) leaves lasting evidence
- users may hide, destroy, or protect this evidence
- forensic analysts discover and collect evidence, avoiding defense mechanisms and accidental data loss
- For this assignment, we will be simulating this analysis by providing you with a “seized” virtual image of a suspect’s computer; you must carefully interact with it to find and report evidence related to a fictional murder case.

Computer Forensics (Your Tasks)

- load a suspect virtual appliance to your environment
- analyze it using the tools, techniques, and concepts we present here
- discover “tokens” hidden in digital evidence
- submit the tokens alongside an explanation of how you found them and a directory containing the files in which you found them

Disclaimers

Everything for this assignment should be done on your local/virtual machine.

Do not access any machines or images without express permission (certain privacy laws protect people against unauthorized access to computers/devices).

Do not perform any network or remote analysis; it is out of scope and unnecessary for this assignment.

Assignment Submission

At Least **1** out of 4 **Password**

- passwords used for encryption or login
- all passwords are exactly one word from the top 100k English words

6 of 9 Total **Two-Word Tokens**

- tokens are two English words separated by a single character
- check places like the end of text files, file names, interesting strings

3 of 4 Total **Three-Word Tokens**

- three English words with a single character between each

2 of 4 Total **Four-Word Tokens** or **Transaction Identifier (Limit 1)**

- four English words with a single character between each
- one of your submitted tokens may be a relevant transaction identifier

Dead and Live Analysis

Dead Analysis

- Investigator examines device resources without running it
- User account password is not required for analysis
- Can avoid suspect counter-measures
- Does not alter evidence state when done correctly

Live Analysis

- Investigator examines running device/image
- User account password will be required to log in
- Can see the environment as the suspect used it.
- Almost always risks loss of evidence.

Dead and Live Analysis (trite examples)

Dead Analysis

- Remove hardware and insert into another machine
- Specialized measurement tools for hardware resources
- Read raw bytes from memory and storage devices
- **Mount partitions to another (potentially virtual) machine**

Live Analysis

- Boot and interact with a suspect's device
- Gain remote access to a running suspect device
- Install specialized software to extract information
- **Run a copied image of a real machine in a virtual machine**

A note on Virtual Traps

Imagine you are protecting incriminating evidence on your machine. What might you do to help prevent others from accessing it?

- Encryption
- Dead-man switch
- Destruction on incorrect passwords
- Hidden or disguised files
- Corrupt your own tools
- Create false interfaces to mislead investigators

Partitions

Partitions separate a physical storage medium into logically isolated sections

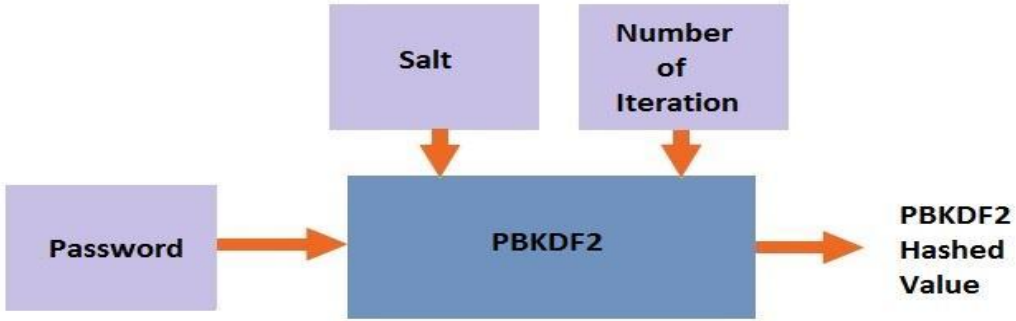
- Each partition has a different file system
- Some partitions can be “bootable” - defining how to boot an operating system
- Partitions can be mounted to a running file system (see install instructions)
- Individual partitions can be encrypted to protect the data stored there

Linux Unified Key Setup (LUKS) - Disk Encryption

- Cryptographically secure method of encrypting entire partitions of data
- Must provide passphrase/key to allow OS to decrypt data read from disk
- Protects data regardless of OS/filesystem
- Vulnerable only with weak passwords (e.g., bruteforce/dictionary attacks)*
- PBKDF can mitigate weak password vulnerabilities

Password-based Key Derivation Functions

- Derive a key from a password or passphrase using a PRF
- Repeatedly (*n* iterations) chain PRF outputs together starting with the password/passphrase to generate the key
- Use the key to perform encryption (e.g. using AES)



Unix File System

General locations:

Location	Description
/	Root directory
/dev	Attached devices and pseudo-devices
/home/mona	Mona’s home directory
/mnt, /media	For mounting, managing, attaching removable devices, filesystems etc
/var/log	System, process log files
/tmp	Temporary files
/etc	Contains system-wide configuration files

Unix File System

`ls -l` without the *-a* flag [hidden files and folders not shown]

Unix File System

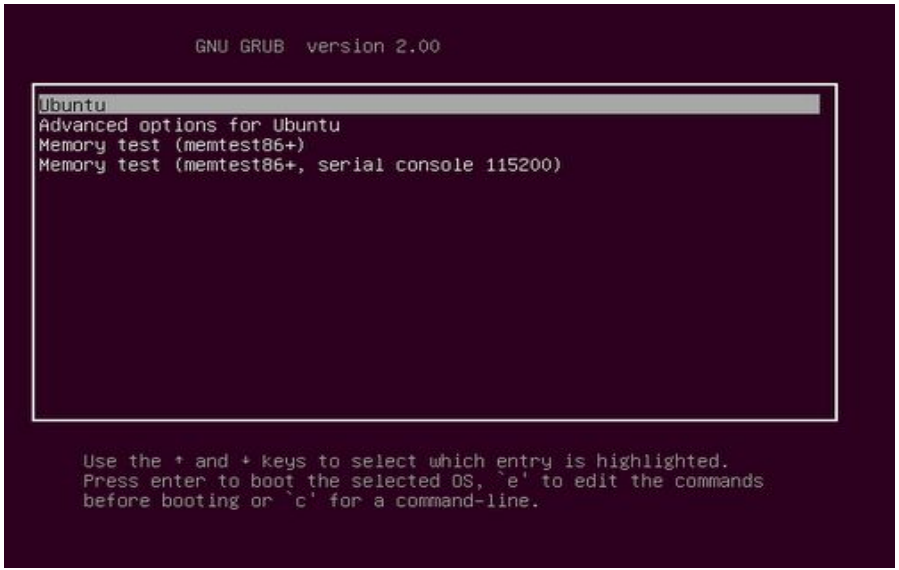
`ls -la` with the *-a* flag [hidden files and folders shown]

```
total 136
drwx------+   3 aruneshmathur  staff    96B Sep  8  2017 Pictures/
drwx------+   3 aruneshmathur  staff    96B Sep  8  2017 Movies/
drwxr-xr-x+   5 aruneshmathur  staff   160B Sep  8  2017 Public/
-r--------    1 aruneshmathur  staff     7B Sep  8  2017 .CFUserTextEncoding
drwx------+   4 aruneshmathur  staff   128B Sep 15  2017 Music/
drwxr-xr-x    5 aruneshmathur  staff   160B Oct 13  2017 .ipython/
drwxr-xr-x    3 aruneshmathur  staff    96B Oct 13  2017 nltk_data/
drwxr-xr-x    3 aruneshmathur  staff    96B Oct 14  2017 .jupyter/
-rw-r--r--@   1 aruneshmathur  staff   243B Oct  1 17:08 .gitconfig
drwxr-xr-x    5 root           admin   160B Nov  8 22:52 ../
drwx------+  15 aruneshmathur  staff   480B Nov 10 15:54 Documents/
drwxr-xr-x    6 aruneshmathur  staff   192B Nov 10 16:25 .subversion/
-rw-r--r--    1 aruneshmathur  staff     0B Nov 10 16:26 .Rhistory
drwxr-xr-x   15 aruneshmathur  staff   480B Nov 10 20:28 .atom/
drwx------@   3 aruneshmathur  staff    96B Nov 11 16:13 Applications/
drwxr-xr-x@  18 aruneshmathur  staff   576B Nov 14 11:04 Website/
drwx------@   5 aruneshmathur  staff   160B Nov 14 20:24 .ssh/
drwxr-xr-x    5 aruneshmathur  staff   160B Nov 15 21:57 .matplotlib/
-rw-r--r--    1 aruneshmathur  staff   199B Nov 18 13:01 .bash_profile
drwx------@  66 aruneshmathur  staff   2.1K Nov 20 13:20 Library/
drwx------    5 aruneshmathur  staff   160B Nov 20 13:58 VirtualBox VMs/
-rw-------    1 aruneshmathur  staff   6.7K Nov 20 22:44 .sqlite_history
-rw-r--r--@   1 aruneshmathur  staff    14K Nov 26 17:06 .DS_Store
drwxr-xr-x   21 aruneshmathur  staff   672B Nov 28 11:33 .rstudio-desktop/
drwx------+  15 aruneshmathur  staff   480B Nov 29 11:03 Desktop/
drwx------    2 aruneshmathur  staff    64B Nov 29 11:51 .Trash/
drwx------+   7 aruneshmathur  staff   224B Nov 29 12:20 Downloads/
-rw-------    1 aruneshmathur  staff    21K Nov 29 15:20 .viminfo
drwxr-xr-x+  31 aruneshmathur  staff   992B Nov 29 15:20 ./
-rw-------    1 aruneshmathur  staff   6.1K Nov 29 22:59 .bash_history
drwx------  110 aruneshmathur  staff   3.4K Nov 29 23:00 .bash_sessions/
```

GNU Grub

Bootloader program used by most Linux operating systems

Presents a list of the operating systems available on disk



Access by hitting ‘Esc’ during boot

GNU Grub

/etc/rc.local is a script that is run when the operating system first loads

- Can be used to start custom and startup services services etc.
- Often used by system administrators to perform maintenance tasks

Linux User Account Passwords

Background: Linux Accounts and Access Control

Account Security is enforced by the operating system

- The OS keeps state of which user owns a process
- It denies access (read/write/execute) based on the user and file permissions
- If the OS isn't running (dead analysis), it can't check permissions

Passwords are stored as hashes in a permissions-protected /etc/shadow

- Changing a password requires only changing this file
- You need to know the root password to change this file, since it is readable/writable only by the root account, or do you…

Communication Traces

Look for traces of various digital communication:

- Internet Relay Chat (IRC)
- Local and web-based email clients
- Browser-stored passwords
- Sites with lingering sessions or “stay signed in” features
- General system log files to see what applications are used for communication

PGP/GPG

"Pretty Good Privacy" / "GNU Privacy Guard"

- Very similar implementation (GPG <= open source re-implementation of PGP)
- Uses RSA techniques to provide asymmetric encryption for end users
- All of the RSA features still apply to a user's GPG private key and public key
  - Public key can be used to encrypt files, archives, strings, etc. such that only the private key can decrypt the output
  - Private key can be used to sign arbitrary strings or files such that a public key can verify that the signature was generated by the corresponding private key
- Ubuntu ships with a GPG binary that can perform all the operations necessary for asymmetric encryption (e.g., generate/import keys, sign, encrypt, verify, decrypt)

Bitcoin tools

Wallet Key Pair (not a PGP/GPG key pair)

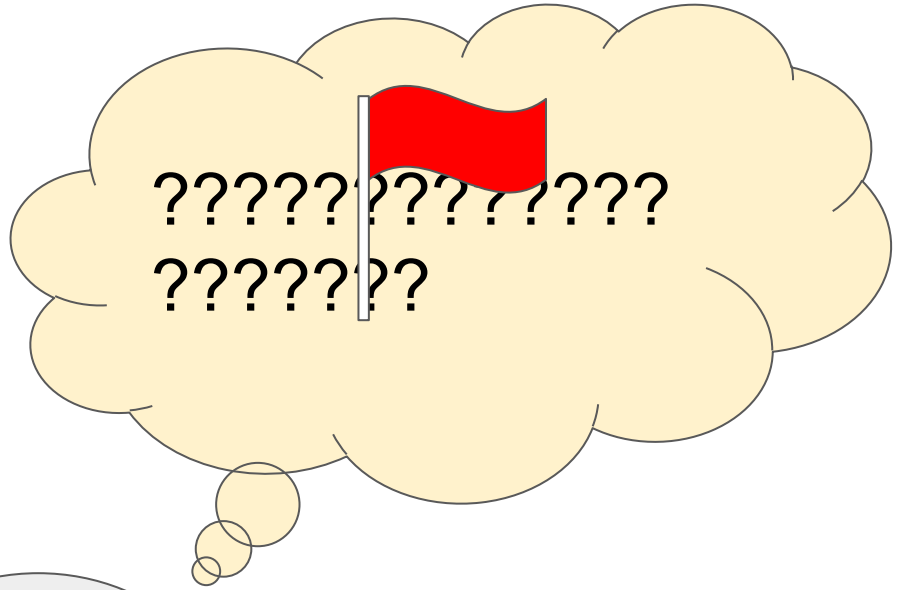
- Bitcoin wallets use asymmetric encryption where each address has a key pair
- Bitcoin key pairs can perform normal encryption/signature operations
- The Bitcoin specification defines how to derive addresses from the keys
- Given the address, anyone can search the blockchain for it

Wallet Software

- Existing tools can perform all the arithmetic associated with key derivation
- Resources like Blockchain Explorer allow basic analysis of the blockchain

Questions?

the slides after this point have nothing to do with the assignment



other common types of CTF “puzzles”

- **“pwn”** -> learn how server works to break it
- **web exploitation** -> learn how website works to break it
- **reversing binaries** -> learn how program works to break it
- **cryptography** -> math, probably

- lots of others!

Beginner ctfs & learning resources

- general/pwn: https://overthewire.org/wargames/bandit/
- web: https://overthewire.org/wargames/natas/ (a bit outdated but still good)
- reversing: https://ropemporium.com/ https://crackmes.one
- cryptography: https://cryptopals.com/

- picoCTF: https://picoctf.com/
- Google Beginner's Quest: https://capturetheflag.withgoogle.com/beginners-quest