Lab 1

Dr. Darryl J D
Mob No: 9986382162

Motivation

> You are leaving a trail, albeit a digital one;
> it's a trail nonetheless.
>
> John Sammons

REvil Ransomware Group gets arrested in Russia

Author of Raccoon Stealer gets arrested in Netherlands

vx-underground
@vxunderground

Mark Sokolovsky, a Ukrainian and the alleged author of Raccoon Stealer, was arrested in the Netherlands in March, 2022 with the accidental help of his girlfriend

They fled Ukraine together. She documented everything on Instagram

She posted this photo 2 days prior to his arrest

vx-underground
@vxunderground

tl;dr if you're an internationally wanted cyber criminal, avoiding a mandatory draft, and illegally immigrating into NATO territory, do not document everything on social media.

9:46 AM · Nov 1, 2022

Link to the actual tweet: https://twitter.com/vxunderground/status/1587304651426332673

How A Floppy Disk Brought The BTK Killer Down (Yes, Really)

https://www.refinery29.com/en-us/2019/08/240899/btk-killer-caught-when-how-floppy-disk-dennis-rader

Linux command line which are essential for digital forensics

ls — used to list the files and directories in a directory

The ls command lets you see a list of all the files and folders in a specific folder

$ ls Desktop Documents Downloads Music Pictures Public Videos

cat — used to display the contents of a file

The cat command (short for "concatenate") lets you print the contents of a file.

$ cat file.txt

Hello World!

grep — used to search for a specific string or a pattern in a file or multiple files

The grep command is extremely useful for searching a string in large files, such as log files. It can speed up investigations dramatically by letting you search for patterns like URLs, E-mail addresses, MD5 hashes, and more.

$ grep "Hello" file.txt
Hello World!

ps — used to list the process running on the system

The ps command lets you see a list of all the processes that are currently running on your computer. The information it provides includes the process ID, user, state, and command that started the process.

$ ps

PID	TTY	TIME	CMD
1824	pts/0	00:00:09	zsh
2879	pts/0	00:00:05	sublime_text
2924	pts/0	00:00:00	plugin_host-3.3
2927	pts/0	00:00:00	plugin_host-3.8
22666	pts/0	00:00:00	ps

strings — used to display the printable strings in a file

The strings command is used in Linux to extract printable strings from binary or non-text files. It scans the files to search for sequences of readable characters. We use it to examine executable files and libraries to fetch readable function and variable names, error messages, and embedded strings.

Most of the bytes within a binary file are not human readable and cannot be printed to the terminal window in a way that makes any sense. There are no characters or standard symbols to represent binary values that do not correspond to alphanumeric characters, punctuation, or whitespace. Collectively, these are known as "printable" characters. The rest are "non-printable" characters.

So, trying to view or search through a binary or data file for text strings is a problem. And that's where strings comes in. It extracts strings of printable characters from files so that other commands can use the strings without having to contend with non-printable characters.

strings — used to display the printable strings in a file

A natural question that might come to your mind is how are strings different from the cat command. The strings command is primarily used to extract readable strings from binary files, while the cat command is used to display the contents of files.

The string command by default prints sequences of characters that are at least 4 characters long, unless you adjust the minimum string length with the -n option

Syntax:

strings [options] filename

Where options can change the behavior of strings command and the filename should be the name of the file to search for printable strings.

strings — used to display the printable strings in a file

The strings command allows you to see human-readable strings of characters inside a file which is helpful in identifying any suspicious strings.

$ strings file.txt

Hello World!

$ strings /bin/bash

/lib64/ld-linux-x86-64.so.2

$DJ

CDDB

E`%

`0

"BB1

B8:

0D@kB

) 9E4

etc

strings — used to display the printable strings in a file

For You to try:

1. Check the content of openssl file in the bin dir using cat command.
2. Check the same file using strings command.

Options of Strings Command

For You to try:

Use the binary file whoami in /bin directory.

-n or --bytes :

By default, the string command scans for sequences of characters that are at least 4 characters long. But we can change this minimum length by using the -n command option.

Example:

For example to extract and display strings that are at least 2 characters long from a file named whoami.

strings -n 2 whoami

Options of Strings Command

For You to try:

Use the binary file whoami in /bin directory.

-f or --print-file-name :

The -f option is used to display the name of the file before each string. This comes helpful when you search in multiple files.

Example:

strings -f openssl whoami

Options of Strings Command

For You to try:

-e:

By default strings command searches for ASCII and UTF-8 text. You may use -e option to tell strings command to support additional character encoding. This comes useful when files are not primarily in ASCII.

Syntax:

strings -e encoding file-name

Supported encodings are:

s for single-7-bit-byte characters.

S for single-8-bit-byte characters.

b for 16-bit little-endian.

l for 32-bit little-endian.

B for 16-bit big-endian.

L for 32-bit big-endian.

Example:

strings -e S whoami

What is a Binary File?

A binary file is a type of computer file that is stored in a binary format, which means that it is composed of a series of 0s and 1s that represent the data stored in the file. Binary files are often used to store data in a form that is more efficient or more compact than a text file.

Binary files are used for a wide range of purposes, including storing executable programs, images, audio and video files, and data files. The most common type of binary file is an executable file, which is a program that can be run on a computer. Other common types of binary files include image files (such as JPEG, PNG, and GIF), audio and video files (such as MP3, AVI, and MOV), and data files (such as database files and spreadsheet files).

Characteristics of a Binary file

One of the main characteristics of a binary file is that it cannot be easily read or edited by humans. This is because the data in a binary file is stored in a series of 0s and 1s, which are not easily understandable by humans. In order to read or edit a binary file, you typically need to use a specialized software tool or program.
Another characteristic of a binary file is that it is typically much smaller in size than a text file containing the same information. This is because binary files are more efficient at storing data, and do not require any additional formatting or structure.

Creating a Binary file

There are several ways to create a binary file. One common method is to use a compiler, which is a program that translates source code (written in a programming language such as C or Java) into a binary file that can be executed by a computer. Binary files can also be created using specialized software tools or by manually converting a text file into a binary format using a hex editor or other type of binary editor.

Here is the most common type of binary file everyone used daily:

Executable files: Executable files, also known as binaries, are programs that can be run on a computer. They are typically stored in a binary format and are used to perform a wide range of tasks, including running applications, installing software, and performing system tasks. Examples of executable file types include .exe, .bin, and .com.

Image files: Image files are used to store digital images in a variety of formats, including JPEG, PNG, and GIF. They are typically stored in a binary format and can be viewed using image viewing software or a web browser.

Audio and video files: Audio and video files are used to store audio and video content, respectively. They are typically stored in a binary format and can be played using media player software or a web browser. Examples of audio file types include .mp3, .wav, and .aac, and examples of video file types include .mp4, .avi, and .mov.

Practical Use Cases of strings command

Dr. Darryl J D

Finding Specific Strings in Binary Files

1.The script asks the user to provide the keyword as input.
2.It stores the user response in the variable named "keyword" and displays the response on the terminal.
3.It begins the search of the keyword in all the binary files one by one using a for loop.
4.The for loop iterator is named as "file". It begins searching for the keyword in the files with extension bin, one file at a time.
5.The script uses the strings command on every file to fetch the readable text and then looks for the keyword using the grep command.
6.We store this result in the "result" variable.
7.We then use an if block with -z option to check if the result variable is not null (empty).
8.For not null results, we print the success message that the keyword has been found along with the file name.
9.If null, we print that we couldn't find the keyword in the current file and we move to the next one.

```
#!/bin/bash

echo "Enter the keyword to search in the binary file: "
read keyword
echo "Keyword provided is $keyword"

for file in *.bin
do

    echo "Searching for $keyword in $file"
    result=$(strings "$file" | grep "$keyword")

    if [ ! -z "$result" ]
    then
        echo "$keyword found in $file"
    else
        echo "$keyword not found in $file. Moving to next"
    fi
done
```

Find Command in Linux

The find command in Linux is a dynamic utility designed for comprehensive file and directory searches within a hierarchical structure. Its adaptability allows users to search by name, size, modification time, or content, providing a flexible and potent solution.

As a pivotal component of the Linux command-line toolkit, the find command caters to the nuanced needs of users, ensuring precision in file exploration and retrieval. Discover the diverse functionalities of the find command and enhance your file management efficiency on the Linux platform.

Syntax of Find Command in Linux :

find [path] [options] [expression]

Here,

path: Starting directory for the search.

Example: find /path/to/search

options: Additional settings or conditions for the search.

Example: find /path/to/search -type f -name "*.txt"

expression: Criteria for filtering and locating files.

Example: find /path/to/search -type d -name "docs"

This syntax allows you to customize your file search by specifying the path, adding options, and defining search criteria using expressions.

Search Start Point

The search point tells the find command where to start the **recursive** search.
Start point as Current Directory: By default find command uses the current directory as the start point of the search. Or you can explicitly mention using a dot ('.').
Example:
find -name myfile.txt

Start point as Specific Directory: You can specify the absolute or relative path as the location of the search start search. Example of the absolute path "/home/ubuntu". Example of the relative path - relative to the current directory "./subdir1".
Example:
find /home/msis/test -name file.txt

Start point as Root Directory: Use / to tell find to search from root directory.
Example:
find / -name myfile.txt
This command search myfile.txt from / and all its subdirectories. That means it searches the entire filesystem. You may see in the output permission denied messages, this mostly happens due to the permission issue.

Search Criteria

Searching by File Name

One of the most common uses of find is to search for files by its name. It's done using the -name test.

Examples:

To find a file named myproject.txt in /home/ubuntu and its subdirectories:

```
find /home/ubuntu -name projectdata.txt
```

To find all files with extension .txt in the current directory and its subdirectories:

```
find . -name "*.txt"
```

To find all files that start with the name "project" in the entire filesystem, use:

```
find / -name "project*"
```

Search multiple directories: Provide each directory path separated by space.

Example:

```
find . dir1 /path/to/dir2/ /path/to/dir3 -name file1.txt
```

How to Find a File in Linux from the Command Line

For You to try:

This query is designed to pinpoint a file within a designated directory. In the provided example, it seeks a file named “sample.txt” within the “GFG” directory.

find ./GFG -name sample.txt

The find command traverses the specified directory (./GFG) and looks for a file named “sample.txt.” If found, it displays the path to the file.

How to Search Files with a Pattern Using `find` Command in Linux

For You to try:

This command is tailored for discovering files within a directory that adhere to a specific naming pattern. In this case, it identifies files ending with ‘.txt’ within the “GFG” directory.

find ./GFG -name *.txt

The command looks for files with names ending in ‘.txt’ within the “GFG” directory, presenting a list of matching files.

How to Find a File in Linux from the Command Line

For You to try:

Searching by File Type

The find command can search files based on their type using -type test. Some of the file types are follows:

f for regular files

d for directories

l for symbolic links

c for character devices

b for block devices

s for local sockets

p for named pipes (FIFOs)

```
find . -type d
```

How to Find a File in Linux from the Command Line

For You to try:

Searching by File Size

he find command, use -size test to search files by size to match.

You can precede the size by + or - to find files greater than or less than the specified size. Without it, you can search for specific sizes.

Size units:

c for bytes

k for Kilobytes

M for Megabytes

G for Gigabytes

b for 512-byte blocks (this is the default if no suffix is used)

find all files in /home/ubuntu with an exact size of 50 Kilobytes in size, type

find /home/ubuntu -size 50k

To find all files in the entire filesystem greater than 1 Megabyte in size, use:

find / -size +1M

How to Find a File in Linux from the Command Line

For You to try:

Find Files by When They Were Modified Using `find` Command in Linux

The find uses a few tests to search files by modification time such as -mtime and -mmin.

-mtime n : Used to search for files that were modified n x 24 hours ago.

Where n for the exactly n, +n more than n and -n less than n.

-mmin n : For more precision, you can use this, which specifies time in minutes instead of days.

Examples:

To find files modified greater than 24 hours ago:

find . -mtime +0

This command will print a list of files in your current directory and its subdirectories that were modified more than 24 hours ago. It's important to note that, because of rounding, find -mtime +0 will not find files modified within the same day.

How to Find a File in Linux from the Command Line

For You to try:

Find Files by When They Were Modified Using `find` Command in Linux

To find files modified between now and 1 day ago

find . -mtime 0
This searches for files in the current directory (and its subdirectories) that have been modified from 0 to 24 hours ago (ie within the past day).

To find files modified less than 1 day ago

find . -mtime -1
This is the same as -mtime 0.

To find files modified between exactly 1 day ago, but less than 2 days ago:

find . -mtime 1
searches for files in the current directory (and its subdirectories) that have been modified between 24 and 48 hours ago.

To find files modified within the last 7 days, you can use:
find /path/to/search -mtime -7

Use Grep to Find Files Based on Content Using `find` Command in Linux

For You to try:

Combining the find command with grep allows you to search for files based on their content. For example, to find files containing the word “pattern” in the current directory and its subdirectories, you can use:

find . -type f -exec grep -l "pattern" {} \;

find .: Initiates the find command, starting the search from the current directory (.).

-type f: Specifies that the search should only consider regular files (not directories or other types of files).

-exec: This option allows you to execute a command for each file found.

grep -l "pattern" {}: The grep command is used to search for the specified pattern within each file. The -l option tells grep to print only the names of files that contain the pattern. The {} is a placeholder for the current file being processed by find, and \; marks the end of the -exec command.

md5sum, sha1sum — used to compute the MD5 and SHA1 hashes of a file

Both of these commands take an input and generate a fixed-length string, also known as a hash or a checksum. If the contents of the file change, even slightly, its hash will be different. This can be useful for detecting if a file has been modified or tampered with.

```
$ md5sum file.txt
8ddd8be4b179a529afa5f2ffae4b9858  file.txt

$ sha1sum file.txt
a0b65939670bc2c010f4d5d6a0b3e4e4590fb92b  file.txt
```

Importance :

Suppose, anyone wants to install an operating system , so to verify if it's correct CD, it's always a good idea to verify .iso file using MD5 checksum, so that you don't end up installing wrong software (some sort of virus which can corrupt your filesystem).

md5sum, sha1sum — used to compute the MD5 and SHA1 hashes of a file

Options :
-b : read in binary mode
-c : read MD5 from files and check them
–tag : create a BSD-style checksum
-t : read in text mode(it's by default)

The options which are useful when verifying checksum :
–ignore-missing : don't report status for missing files
–quiet : don't print OK for each successfully verified file
–status : don't output anything, status code shows success
–strict : exit non-zero for improperly formatted checksum files
-w : warn about improperly formatted checksum files

For You to try:

Example 1: Store the MD5 checksum in file and then verify it.

```
# md5sum /home/djd/test/test.cpp > checkmd5.md5
```

It will store the MD5 checksum for test.cpp in file checkmd5.md5

```
# md5sum -c checkmd5.md5
```

It will verify the contents of file

Output :

```
/home/mandeep/test/test.cpp: OK
```

After changing the contents of file checkmd5.md5, the output will be :

```
/home/mandeep/test/test.cpp: FAILED md5sum: WARNING: 1
computed checksum did NOT match
```

Sha1sum Examples - https://shapeshed.com/unix-sha1sum/

file — used to determine the type of a file based on its contents

The file command can be used to identify files such as text, image, audio, video, and executable files. It can also be used to identify unknown files that may potentially be malicious.

$ file /etc/passwd
/etc/passwd: ASCII text

$ file image.png
image.png: PNG image data, 562 x 424, 8-bit/color RGB, non-interlaced

For You to try: jpg, mkv, gif etc

xxd — used to print hex dump of a given file

The xxd command is useful to print hex dump of a given file or standard input. It can also convert a hex dump back to its original binary form.

```
$ xxd image.jpg
00000000: ffd8 ffe0 0010 4a46 4946 0001 0101 0048  ......JFIF.....H
00000010: 0048 0000 ffdb 0043 0005 0304 0404 0305  .H.....C........
00000020: 0404 0405 0505 0607 0c08 0707 0707 0f0b  ................
00000030: 0b09 0c11 0f12 1211 0f11 1113 161c 1713  ................
00000040: 141a 1511 1118 2118 1a1d 1d1f 1f1f 1317  ......!.........
00000050: 2224 221e 241c 1e1f 1eff db00 4301 0505  "$".$.......C...
```

ps — used to list the process running on the system

The ps command lets you see a list of all the processes that are currently running on your computer. The information it provides includes the process ID, user, state, and command that started the process.

```
$ ps
  PID TTY          TIME CMD
 1824 pts/0    00:00:09 zsh
 2879 pts/0    00:00:05 sublime_text
 2924 pts/0    00:00:00 plugin_host-3.3
 2927 pts/0    00:00:00 plugin_host-3.8
22666 pts/0    00:00:00 ps
```

netstat — used to display information about the network connections on a system

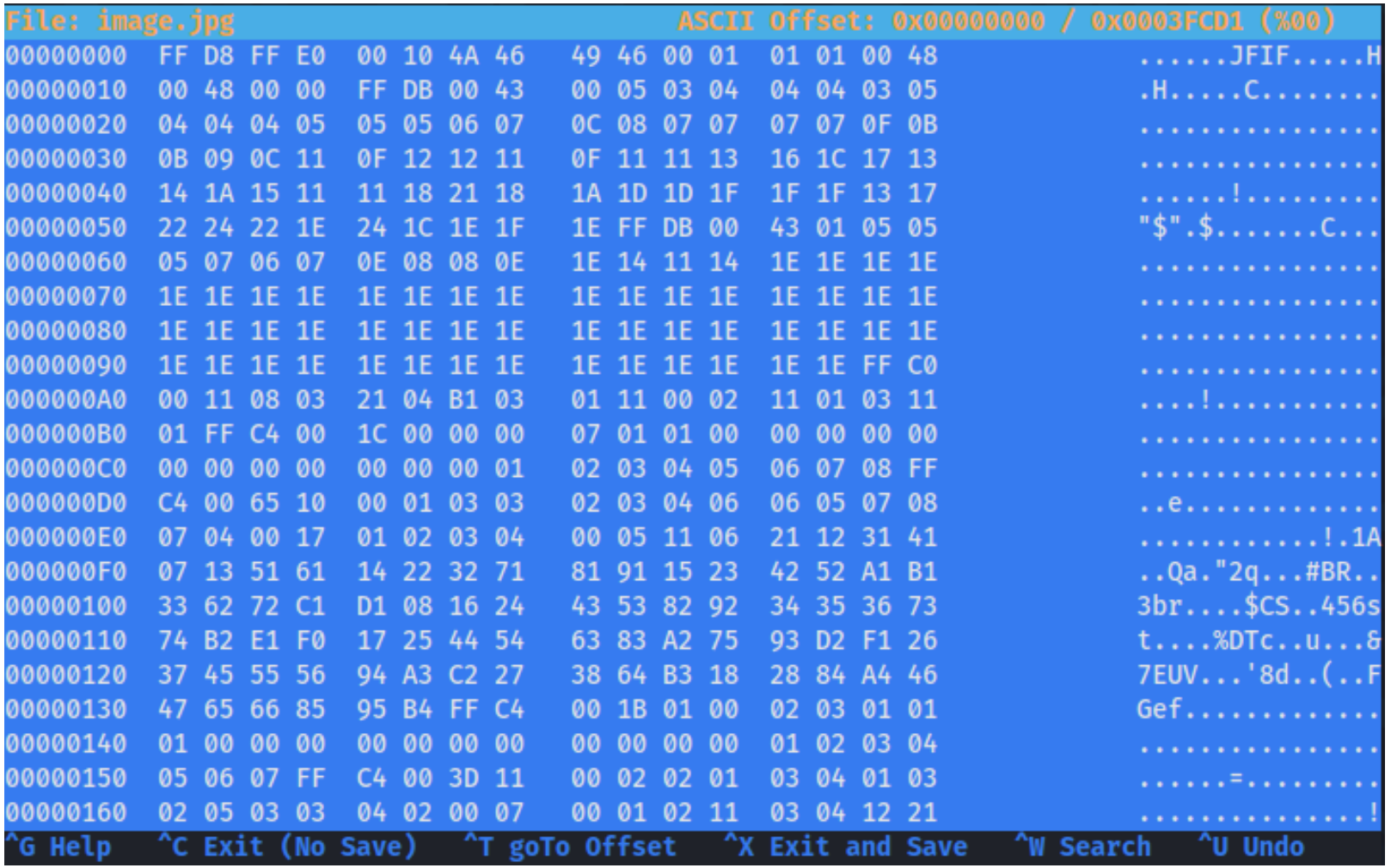
This tool provides useful information about active connections on a system. The information displayed by this tool includes local and remote addresses and ports of active connections.

```
$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 192.168.0.106:44300     239.237.117.34.bc:https ESTABLISHED
tcp        0      0 192.168.0.106:39884     93.184.220.29:http      ESTABLISHED
tcp        0      0 192.168.0.106:58308     ec2-52-39-122-167:https ESTABLISHED
tcp        0      0 192.168.0.106:56910     static-48-7-129-15:http ESTABLISHED
udp        0      0 192.168.0.106:bootpc    192.168.0.1:bootps      ESTABLISHED
```

hexedit — used to edit files in hexadecimal format

The hexedit tool lets you edit raw bytes of a file in an interactive way. It is often used for repairing corrupted files.

$ hexedit image.jpg



Exercises

Provide the complete commands for all the exercises where asked for the command, and provide a descriptive answer where asked for an explanation. There may be multiple answers/commands for these exercises, so feel free to submit the answer you feel most comfortable with.

Questions

1. If we wanted to list all the .txt files in the current directory, what command would we want to use?

2. What command can we use to read the contents of the file /etc/passwd?

3. If we wanted to search for the string Error in all files in the /var/log directory, what would our command be?

4. What would be the commands to calculate MD5 and SHA1 hashes of the file /etc/passwd?

5. Use the file command to determine the type of the file /usr/bin/cat and explain the output in 2-3 sentences.

6. What command can we use to display all printable strings of length ≥ 8 in the file /bin/bash?

Questions

7. Given the following output of the file command, can you determine what's wrong with this file?

$ file image.jpg

image.jpg: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=3ab23bf566f9a955769e5096dd98093eca750431, for GNU/Linux 3.2.0, not stripped

8. If we wanted to look for files modified in the last 30 minutes in /home directory, what command would we want to use?

Hint: Explore how you can use find command to achieve this.

Questions

9. What command can we use to display information about all active TCP connections on the system?

10. Given 3 corrupted image file, can you find a way to recover and view its contents?

Hint 1: A quick google search for “magic bytes” might help.

Hint 2: Explore how hexedit can help you here.

The three corrupted files are named one, two and Three.