CTF Basics

(aka “How do I capture that flag?”)

Environment

To start, working with Windows will be extremely difficult - not only because it will be more difficult to set up the necessary environments and tools, but also because some tools do not even support Windows. That is why it is far better to use a Linux-based operating system in a virtual machine (VM). A good choice is Kali Linux, which is specifically tailored for penetration testing and contains many of the tools you might need to test the various vulnerabilities. Similarly, even if you use Linux as your daily driver, it might be more convenient to install a Kali VM, since then not only will you have the necessary tools on-hand, but also will not have to spend precious time installing and setting up tools for your specific distro and environment.

Setting up a VM is relatively simple - to begin, you will need a VM host, this is the program that will be running and managing your VM. Good choices are Virtualbox (Free) and VMWare (Player is free, contains a basic feature set but it is more than sufficient), both of which are supported on Windows, Mac and Linux. On Windows, you also have the choice of using Hyper-V, which is built into Windows. On Linux, you can also use GNOME Boxes. The differences between the programs are minor, but if a specific program does not work, it might be helpful to check out alternatives.

If you need more detailed tutorial here is the simplest tutorial that we could find - https://www.wikihow.com/Install-Ubuntu-on-VirtualBox

Many cybersecurity and CTF veterans use Kali Linux because by default it contains many useful tools[1]. Kali Linux (https://kali.org) offers both installation ISOs (which you can use to install Kali yourself) or VM images, which come with Kali pre-installed, so the only thing you have to do is load the image into your VM host of choice. If you choose to use a VM image, pay special attention that you download the image format which is correct for your VM host of choice, otherwise you will be unable to import the image into the VM host. By default, for Kali VM images the user is kali:kali (username:password), which you will most likely have to enter when booting up the VM.

[1] Also many say that it is confusing and it is easier to use ParrotOS security edition because it contains almost as many tools but is more friendly to beginners

Tools

As a general note, most of these tools have help pages which can be accessed by appending `-h` or `--help` to the arguments, which will provide the general usage and overall arguments which the command supports. Similarly, commands usually are installed with `man` pages, which can be accessed via `man <command name>`, which also provide information on what a command is, what it does and how its behaviour can be changed with arguments.

It also helps to have basic bash skills, such as being able to pipe one command's output into a different command as input or run one command after the other. Command output can be piped using the `|` operator, while the && operator will instruct bash to run one command after the other. Bash scripting also supports more advanced behaviours, such as variables, for loops, etc., for which you can find a cheat sheet here: https://devhints.io/bash

Finally, some challenges will require you to connect to a machine via `openvpn`. You will be given a .ovpn file through the CTFd platform, which you will be able to use to connect to a simulated network that will contain the machine on it. Simply run `openvpn <.ovpn file>` (you might need to run the command with sudo) and it will automatically connect you to the network, after which you will be able to freely access the machine.

Steganography

- ➔ https://stegonline.georgeom.net - online tool with which you can see information about an image, for example, view EXIF metadata, output of the strings command. Additionally, you can easily see colour information of the image to check whether individual colour layers do not have anything hidden in them.
- ➔ `strings` - very useful Linux command line tool that prints out any readable strings contained within a file. Useful arguments: change minimal length (`-n <length>`), change the character width (`-e {s,S,b,l,B,L}`)
- ➔ `binwalk` - command line tool which analyses a file for any files that might be included within it, for example, a PNG image which might have another PNG image embedded inside it. Useful args: automatically extract found files (`-e`)
- ➔ `xxd` - command line tool which outputs the bytes of a file in hexadecimal, which might help shed light on what the file is and what could be found in it.

Reverse Engineering

- ➔ `strings` - also useful for reverse engineering, because often string literals used in code tend to not be hidden, which might help shed light on what the program is and what it is doing.
- ➔ `ltrace` - command line tool which shows any library calls a program makes, which can be used to form an understanding of what the program might be doing and how to exploit it.
- ➔ `strace` - command line tool with which you can analyse system calls (for example, input, output, opening files) the program makes.
- ➔ `gdb` - command line tool suite with which you can decompile, debug and analyse programs. For example, with `disass main` you can decompile and see the assembly instructions describing the main function of a program.
- ➔ `ghidra` - GUI tool suite for general reverse engineering of programs. Allows you to view the full assembly of a program, translate assembly instructions into C and analyse it.
- ➔ `netcat` - command line tool for making and receiving connections and outputting them to the terminal. Useful for listening for reverse shells and sending payloads over connections.
- ➔ `telnet` - command line tool for making connections to servers. Useful when you are given an IP and a port with no clue as to what is actually running on it.
- ➔ `printf` - a command that simulates the behaviour of the printf function in C. This can be useful to pass specific bytes to a program in order to exploit a vulnerability within it.

Cryptography

- ➔ https://cyberchef.org/ - web tool that allows you to perform and chain various cryptographic and data transformations with text or data.
- ➔ `john` (John the Ripper) - tool for brute forcing hashes to determine their source string using a wordlist or pure brute forcing.
- ➔ `hashcat` - tool for analysing hashes (determining their type) and brute forcing them using wordlists or simple guessing, similar to John the Ripper.

- ➔ https://www.dcode.fr/en - website that contains many ciphers and tools to analyse and crack them

Web

- ➔ `gobuster` - command line tool with which you can enumerate directories or subdomains to find hidden or unknown directories that might contain information.
- ➔ `dirbuster` - an alternative to gobuster which also enumerates directories or subdomains in order to find any hidden or unknown directories that might contain information.
- ➔ `sqlmap` - command line tool which allows you to test forms for SQL injections and pull data from the underlying database if a SQL injection is found.
- ➔ Burp Suite - GUI tool for analysing the security of websites, for example, capture and analyse requests a website makes (Scanner), repeat requests with modified fields (Repeater) as well as brute forcing values of certain fields (Intruder).
- ➔ Wappalyzer - browser extension that allows viewing the tech stack of a website, for example, see what version of Wordpress a website uses. This can be useful to determine what vulnerabilities a website might have.
- ➔ OWASP Zap - a GUI tool for analysing web applications for any potential vulnerabilities. https://www.zaproxy.org/

Forensics

- ➔ `volatility` - command line tool which allows you to view and analyse memory dumps in order to see processes and opened files that were contained in memory at the time.
- ➔ Wireshark - GUI tool with which you can analyse network packets and their content, as well as viewing "conversations" (for example, an HTTP session) and see the full content sent within it. Wireshark also allows the loading of previously recorded conversations through .pcap files.
- ➔ Audacity (or any of its forks) - general audio editing tool, useful for analysing audio (both waveforms and spectrograms) to see whether there is any information hidden within an audio file.

- ➔ exiftool - dumps metadata of various filetypes (such as PNGs), useful for finding information that might be hidden in the metadata of the image.

Wordlists

- ➔ rockyou.txt - a wordlist containing the most frequently used passwords, which can be used in other tools (such as hashcat or John the Ripper) to crack passwords.
- ➔ https://github.com/danielmiessler/SecLists - a repository of wordlists for a plentitude of scenarios, for example, frequently used directory names, which could be used in a tool such as `gobuster` to effectively iterate through directories and find anything useful.
- ➔ https://github.com/swisskyrepo/PayloadsAllTheThings - a repository of a plentitude of useful injections that could be used to check and determine possible vulnerabilities.