**A black and white text with a green flag and a flag

Description automatically generated**

# What is a CTF?

Capture the Flags, or CTFs, are computer security competitions. Teams of competitors (or just individuals) are pitted against each other in various challenges across multiple security disciplines, competing to earn the most points.

## **Why play CTFs?**

Real-world vulnerabilities are featured in challenges, allowing you to flex your programming, problem solving, and teamwork skills! CTFs are often the beginning of one's cyber security career due to their team building nature and competitive aspect. In addition, there isn't a lot of commitment required beyond a weekend.

CTFs bring these vulnerabilities right to your machine in small, compartmentalized challenges, fostering collaboration and community building (with friendly competition of course!).

If you're looking to meet new people in this space, check out your local [CitySec](https://www.reddit.com/r/netsec/wiki/meetups/citysec/)!

## **Who can play in a CTF?**

Participants can work individually or in teams to solve challenges. Typically, an organization would feature multiple members playing for the same team, working together to solve challenges. If you're working alone, we encourage you to do some searching or friendly recruiting to have another mind to bounce ideas off of!

**Info**

For information about ongoing CTFs, check out [CTFTime](https://ctftime.org/).

## **Do I need special tools or computers?**

A terminal environment is essential to experiment and install tools in. Linux and MacOS systems should already have terminal emulators installed natively.

If you're on Windows, install Linux with [WSL](https://learn.microsoft.com/en-us/windows/wsl/install) or setup a VM ([virtual machine](https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-a-virtual-machine)). See our [recommended software](https://ctf101.org/faq/recommended-software/).

**Info**

Images like [Kali Linux](https://www.kali.org/get-kali/#kali-platforms) come prebuilt with tools for all your pentesting needs!

To learn more about getting a server or connecting to challenges, check out the [FAQ](https://ctf101.org/faq/connecting-to-services/)!

Got the hang of it? Move on to [CTF-basics](https://ctf101.org/intro/ctf-basics/)

# How to get started

First of all, make sure to check out our [recommended software](https://ctf101.org/faq/recommended-software/) section. It's handy to have these tools installed and ready as you get to solving some CTFs.

Ideally, you must have : - a decompiler like [Binja](https://binary.ninja/) - a debugger, [gdb](https://www.sourceware.org/gdb/) - a suite of web tools, [Burp](https://portswigger.net/burp/communitydownload), [sqlmap](https://sqlmap.org/), and [Wireshark](https://www.wireshark.org/download.html) are solid to begin with - the essential python package [pwntools](https://docs.pwntools.com/en/stable/install.html) to interact with processes easily

# How to get started

First of all, make sure to check out our [recommended software](https://ctf101.org/faq/recommended-software/) section. It's handy to have these tools installed and ready as you get to solving some CTFs.

Ideally, you must have : - a decompiler like [Binja](https://binary.ninja/) - a debugger, [gdb](https://www.sourceware.org/gdb/) - a suite of web tools, [Burp](https://portswigger.net/burp/communitydownload), [sqlmap](https://sqlmap.org/), and [Wireshark](https://www.wireshark.org/download.html) are solid to begin with - the essential python package [pwntools](https://docs.pwntools.com/en/stable/install.html) to interact with processes easily

**Cryptography**

Cryptography involves encrypting or decrypting a piece of data. The following are the tools used in cryptography:

* **CyberChef:** A web application that provides a suite of tools for data analysis and manipulation. It can be used for encryption, decryption, and many other purposes.
* **FeatherDuster:** A tool that can identify and exploit weaknesses in cryptographic implementations.
* **Hash Extender:** A tool for extending hash length attacks.
* **padding-oracle-attacker:** A tool for attacking padding oracle vulnerabilities in web applications.
* **PkCrack:** A tool for breaking PkZip encryption.
* **RSACTFTool:** A tool for attacking RSA encryption.
* **RSATool:** A tool for recovering the RSA private key from a given public key.
* **XORTool:** A tool for performing XOR encryption and decryption.
* **Cryptii:** A web application that provides a suite of tools for encryption, decryption, and encoding.
* **Keyboard Shift:** A tool for performing keyboard shift ciphers.

**Steganography**

Steganography is tasked with finding information hidden in files or images. The following are the tools used in steganography:

* **StegCracker:** A tool for cracking steganography-encoded files.
* **Steghide:** A tool for hiding data in files and images.
* **Openstego:** A tool for hiding data in images.
* **Stegsolve:** A tool for solving steganography challenges.
* **Online stego tool:** A web application that provides a suite of tools for steganography.

**Binary Exploitation/Pwn**

Binary exploitation involves exploiting a binary file and exploiting a server to find the flag. The following are the tools used in binary exploitation:

* **readelf:** A tool for analyzing ELF files.
* **formatStringExploiter:** A tool for exploiting format string vulnerabilities.
* **DLLInjector:** A tool for injecting DLLs into running processes.
* **libformatstr:** A library for exploiting format string vulnerabilities.

**Reverse Engineering**

Reverse Engineering in a CTF is typically the process of taking a compiled (machine code, bytecode) program and converting it back into a more human-readable format. The following are the tools used in reverse engineering:

* **ltrace:** A tool for tracing library calls.
* **Hopper:** A disassembler and decompiler for Mac OS X and Linux executables.
* **Binary Ninja:** A disassembler and reverse engineering platform.
* **gdb:** A debugger for C, C++, and Fortran.
* **IDA:** A disassembler and debugger for Windows, Mac OS X, and Linux.
* **radare2:** A disassembler and debugger.
* **Ghidra:** A software reverse engineering framework.
* **apktool:** A tool for reverse engineering Android APK files.
* **Androguard:** A tool for analyzing Android applications.

**Web**

The following are the tools used in web exploitation:

* **BurpSuite:** A web application security testing platform.
* **Commix:** A tool for command injection vulnerabilities.
* **Hackbar:** A tool for manual SQL injection attacks.
* **Raccoon:** A tool for scraping and enumerating endpoints in web applications.
* **SQLMap:** A tool for automating SQL injection attacks.
* **DirBuster:** A tool for finding hidden directories and files in web applications.
* **gobuster:** A tool for finding hidden directories and files in web applications.
* **nikto:** A vulnerability scanner for web servers.
* **wpscan:** A vulnerability scanner for WordPress installations.
* **CloudFlare Bypass:** A tool for bypassing CloudFlare protection.
* **Edit This Cookie:** A tool for editing cookies in web applications.
* **File or Directory(robots.txt, /.git/, /admin/):** A tool for finding hidden files and directories in web applications.

**Forensics**

Forensics challenges can include file format analysis, steganography, memory dump analysis, or network packet capture analysis. Any challenge to examine and process a hidden piece of information out of static data files (as opposed to executable programs or remote servers) could be considered a Forensics challenge. The following are the tools used in forensics:

* **split:** A tool for splitting files.
* **pdfinfo:** A tool for analyzing PDF files.
* **pdfimages:** A tool for extracting images from PDF files.
* **pdfcrack:** A tool for recovering PDF passwords.
* **pdfdetach:** A tool for extracting embedded files from PDF files.
* **Keepass:** A password manager.
* **Magic Numbers:** A tool for identifying file types based on their signatures.
* **foremost:** A tool for recovering files based on their headers, footers, and internal data structures.
* **binwalk:** A tool for analyzing and extracting firmware images.
* **Repair image online tool:** An online tool for repairing corrupt images.
* **photorec:** A tool for recovering lost files from hard disks and memory cards.
* **TestDisk:** A tool for recovering lost partitions and files.
* **pngcheck:** A tool for checking PNG files for errors.
* **pngcsum:** A tool for checking the integrity of PNG files.
* **Registry Dumper:** A tool for dumping the Windows registry.
* **Dnscat2:** A tool for tunneling data over DNS.
* **pefile:** A tool for analyzing Windows PE files.
* **Wireshark:** A network protocol analyzer.
* **Network Miner:** A network forensic analysis tool.
* **PCAPNG:** A tool for capturing and analyzing network traffic.
* **tcpflow:** A tool for capturing and analyzing network traffic.
* **PcapXray:** A tool for analyzing and visualizing network traffic.
* **qpdf:** A tool for manipulating PDF files.
* **Audacity:** A tool for analyzing audio files.
* **sonic visualiser:** A tool for analyzing audio files.
* **ffmpeg strings:** A tool for extracting strings from media files.
* **file:** A tool for identifying file types based on their content.
* **grep:** A tool for searching for patterns in files.
* **scalpel:** A tool for carving files from disk images.
* **bgrep:** A tool for searching for patterns in binary files.
* **hexdump:** A tool for dumping binary data in hexadecimal format.
* **xxd:** A tool for converting binary data to hexadecimal format.
* **base64:** A tool for encoding and decoding base64 data.
* **xplico framework:** A network forensic analysis tool.
* **zsteg:** A tool for detecting steganography in PNG and BMP files.
* **gimp:** A tool for editing images.
* **Memory dump — volatility:** A tool for analyzing memory dumps.
* **ethscan:** A tool for analyzing Ethereum blockchain transactions.

## Types of challenges

The CTF event consists of 8 challenges. Teams must attempt to complete as many of the tasks as possible, capturing ‘flags’ within each category. Capturing a flag consists of bitesize challenges ranging in difficulty, with the purpose of finding a ‘flag’ or a line of text hidden in software code, web application, or somewhere on the web.

The skill level varies, from beginner to advanced, which makes the challenge of the Business CTF accessible for all security professionals. It’s also a valuable training experience for junior and senior pen testers.

This year’s challenges included:

1. **Web** – identifying and exploiting vulnerabilities in web applications.
2. **Forensics** – investigating various types of data to understand what has happened, by using different recovery methods.
3. **Cloud** – detect and exploit cloud-related misconfigurations and vulnerable deployments.
4. **Pwn** – identifying and exploiting vulnerable software running on a given target.
5. **Fullpwn** – identifying and exploiting vulnerabilities against a given target, in order to obtain a foothold, and then attempt to escalate privileges.
6. **Reverse** – discover and exploit vulnerabilities against binary files by reverse engineering them.
7. **Crypto** – analyse the applied cryptographic functions/algorithms and understand how to decrypt and/or obtain the flag effectively.
8. **Hardware** – identifying and exploiting weaknesses in various hardware design and implementation flaws.

**Tools used:**

* [CyberChef](https://gchq.github.io/CyberChef/)
* [nmap](https://nmap.org/)
* [scilla](https://github.com/edoardottt/scilla)
* [cariddi](https://github.com/edoardottt/cariddi)
* [lit-bb-hack-tools](https://github.com/edoardottt/lit-bb-hack-tools)
* [pentestmonkey](https://github.com/pentestmonkey)
* [gobuster](https://github.com/OJ/gobuster)
* [Burpsuite](https://portswigger.net/burp)
* [metasploit](https://www.metasploit.com/)
* [sqlmap](http://sqlmap.org/)
* [zaproxy](https://www.zaproxy.org/)
* [wireshark](https://www.wireshark.org/)
* [whois](https://en.wikipedia.org/wiki/WHOIS)
* [dig](https://en.wikipedia.org/wiki/Dig_(command))
* [ping](https://en.wikipedia.org/wiki/Ping_(networking_utility))
* [traceroute](https://en.wikipedia.org/wiki/Traceroute)
* [exiftool](https://exiftool.org/)
* [hashcat](https://hashcat.net/hashcat/)
* [john the ripper](https://www.openwall.com/john/)
* [GTFObins](https://gtfobins.github.io/)
* [nikto](https://github.com/sullo/nikto)
* [hash-identifier](https://tools.kali.org/password-attacks/hash-identifier)
* [netcat](https://en.wikipedia.org/wiki/Netcat)
* [ftp](https://en.wikipedia.org/wiki/File_Transfer_Protocol)
* [binwalk](https://github.com/ReFirmLabs/binwalk)
* [steghide](http://steghide.sourceforge.net/)
* [curl](https://curl.se/)
* [PowerSploit](https://github.com/PowerShellMafia/PowerSploit)
* [hydra](https://github.com/vanhauser-thc/thc-hydra)
* [exploit-db](https://www.exploit-db.com/)
* [Nessus](https://www.tenable.com/products/nessus)
* [enum4linux](https://github.com/CiscoCXSecurity/enum4linux)
* [smbclient](https://www.samba.org/samba/docs/current/man-html/smbclient.1.html)
* [linEnum](https://github.com/rebootuser/LinEnum)
* [linpeas](https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/linPEAS)
* [What's my name Web](https://whatsmyname.app/)
* [Google Maps](https://www.google.com/maps)
* [Scylla.sh](https://scylla.sh/api)
* [Have I been Pwned](https://haveibeenpwned.com/)
* [Jeffrey's Image Metadata Viewer](http://exif.regex.info/)
* [radare2](https://github.com/radareorg/radare2)
* [ILSpy](https://github.com/icsharpcode/ILSpy)
* [PowerShell](https://en.wikipedia.org/wiki/PowerShell)
* [crackstation](https://crackstation.net/)
* [lxc](https://en.wikipedia.org/wiki/LXC)
* [shodan](https://www.shodan.io/)
* [gpg](https://gnupg.org/)
* [Dcode.fr](http://dcode.fr/)
* [stegsolve](https://en.kali.tools/all/?tool=1762)
* [Xor.pw](http://xor.pw/)
* [fcrackzip](https://github.com/hyc/fcrackzip)
* [Vim](https://www.vim.org/)
* [peepdf](https://github.com/jesparza/peepdf)
* [vmonkey](https://github.com/decalage2/ViperMonkey/blob/master/vipermonkey/vmonkey.py)
* [tplmap](https://github.com/epinna/tplmap)
* [xsrfprobe](https://github.com/0xInfection/XSRFProbe)
* [jwt.io](https://jwt.io/)
* [c-jwt-cracker](https://github.com/brendan-rius/c-jwt-cracker)
* [wfuzz](https://github.com/xmendez/wfuzz)
* [kerbrute](https://github.com/ropnop/kerbrute)
* [impacket](https://github.com/SecureAuthCorp/impacket)
* [Rubeus](https://github.com/GhostPack/Rubeus)
* [evil-winrm](https://github.com/Hackplayers/evil-winrm)
* [ffuf](https://github.com/ffuf/ffuf)
* [knock](https://github.com/grongor/knock) (Port Knocking)
* [knock](https://github.com/guelfoweb/knock) (Subdomains scanning)
* [Web Archive](https://web.archive.org/)
* [ViewDNS.info](https://viewdns.info/)
* [dirbuster](https://tools.kali.org/web-applications/dirbuster)
* [yarGen](https://github.com/Neo23x0/yarGen)

#### Tool Recommendations

Below you find just a few tools to start with. Of course this is not a complete list and there are always better tools for the job out there. Take small steps and get comfy with tools and techniques to develop and at last improve your unique approach on a system.

###### Information Gathering

* [Nmap](https://github.com/nmap/nmap)
* [masscan](https://github.com/robertdavidgraham/masscan)
* [RustScan](https://github.com/RustScan/RustScan)

###### Vulnerability Analysis

* [nikto](https://github.com/sullo/nikto)

###### Web Application Analysis

* [WhatWeb](https://github.com/urbanadventurer/WhatWeb)
* [Burp Suite](https://portswigger.net/burp/communitydownload)
* [Gobuster](https://github.com/OJ/gobuster)
* [dirsearch](https://github.com/maurosoria/dirsearch)
* [ffuf](https://github.com/ffuf/ffuf)
* [wfuzz](https://github.com/asciimoo/wuzz)
* [WPScan](https://github.com/wpscanteam/wpscan)

###### Database Assessment

* [sqlmap](https://github.com/sqlmapproject/sqlmap)

###### Password Attacks

* [thc-hydra](https://github.com/vanhauser-thc/thc-hydra)
* [hashcat](https://hashcat.net/hashcat)
* [NetExec](https://github.com/Pennyw0rth/NetExec)

###### Exploitation / Post Exploitation Tools

* [Metasploit](https://github.com/rapid7/metasploit-framework)
* [BloodHound](https://github.com/BloodHoundAD/BloodHound)
* [Impacket](https://github.com/fortra/impacket)
* [enum4Linux-ng](https://github.com/cddmp/enum4linux-ng)
* [PEASS-ng](https://github.com/carlospolop/PEASS-ng)
* [PSPY](https://github.com/DominicBreuker/pspy)
* [Evil-WinRM](https://github.com/Hackplayers/evil-winrm)

## Bundled docker vulnerable images

Docker vulnerable images are docker containers to running deliverately vulnerable services. The following vulnerable images are installed so that you can practice within your own Kali installation:

* [OWASP Juice Shop](https://owasp.org/www-project-juice-shop/)
* [OWASP WebGoat](https://owasp.org/www-project-webgoat/)

### Running the vulnerable docker images

#### OWASP Juice Shop

docker run --rm -d -p 3000:3000 --name juice-shop bkimminich/juice-shop

# use web browser to go to http://localhost:3000 in Kali host

#### OWASP WebGoat

docker run --rm -it -p 127.0.0.1:8080:8080 -p 127.0.0.1:9090:9090 -e TZ=UTC --name webgoat webgoat/webgoat

# use web browsser to go to http://127.0.0.1:8080/WebGoat in Kali host

## Additional Git Repos

Additional git repos cloned locally at these locations:

* [/opt/GitTools](https://github.com/internetwache/GitTools): A repository with 3 tools for pwn'ing websites with .git repositories available
* [/opt/zphisher](https://github.com/htr-tech/zphisher): An automated phishing tool with 30+ templates
* [/opt/vulhub](https://github.com/vulhub/vulhub): Pre-Built Vulnerable Environments Based on Docker-Compose
* [/opt/privesc-scripts/LinEnum](https://github.com/rebootuser/LinEnum): Scripted Local Linux Enumeration & Privilege Escalation Checks
* [/opt/privesc-scripts/linux-exploit-suggester](https://github.com/The-Z-Labs/linux-exploit-suggester): Linux privilege escalation auditing tool
* [/opt/privesc-scripts/linux-smart-enumeration](https://github.com/diego-treitos/linux-smart-enumeration): Linux enumeration tool for pentesting and CTFs with verbosity levels Topics
* [/opt/privesc-scripts/PEASS-ng](https://github.com/carlospolop/PEASS-ng): PEASS - Privilege Escalation Awesome Scripts SUITE (with colors)
* [/opt/WEF](https://github.com/D3Ext/WEF): Wi-Fi Exploitation Framework

## Kali Linux Useful Links

* [Kali Tools](https://www.kali.org/tools/)
* [Kali Linux Metapackages](https://www.kali.org/tools/kali-meta/)
* [Kali's Default Credentials](https://www.kali.org/docs/introduction/default-credentials/)

## Other Useful Links

* [**HackTricks**](https://book.hacktricks.xyz/) ⭐
* [**HackTricks Cloud**](https://cloud.hacktricks.xyz/) ⭐
* [OWASP Cheat Sheet Series](https://cheatsheetseries.owasp.org/)
* [OWASP Web Security Testing Guide](https://owasp.org/www-project-web-security-testing-guide/latest/)
* [Payloads All The Things](https://swisskyrepo.github.io/PayloadsAllTheThings/)
* [GTFOBins](https://gtfobins.github.io/)
* [CyberChef](https://gchq.github.io/CyberChef/) - Encode/decode data
* [CrackStation](https://crackstation.net/) - Hash Rainbow List
* [Reverse Shell Generator](https://www.revshells.com/)
* [CI/CD Goat](https://github.com/cider-security-research/cicd-goat)
* [h4cker](https://github.com/The-Art-of-Hacking/h4cker)
* [PortSwigger Web Security Academy](https://portswigger.net/web-security/all-materials)