U.S. DEPARTMENT OF ENERGY'S CYBERFORCE® PROGRAM

CyberForce® 101

CTFs

October 2023

cyberforcecompetition@anl.gov

CTFs 101

Overview

CTF (Capture the Flag) competitions consist of challenges of a variety of tasks ranging from a scavenger hunt to hacking into a server to steal data. CTF-type challenges are found in multiple CyberForce competitions, as either anomalies in the CyberForce competition or as challenges to break out of a room in Conquer the Hill: Reign.

In this walkthrough, we will introduce a variety of CTF challenges that you may see in other competitions. This walkthrough only serves as an introduction and not as a deep dive into each topic.

Cryptography

Cryptography can be generally divided into classic and modern. We provide a basic definition and overview below, but other walkthroughs provide a deeper look into these algorithms and encryption methods. All of these can be seen in CTF competitions.

Classic cryptography mainly includes the following aspects: monoalphabetic cipher, polyalphabetic cipher, and strange encryption. Classical cryptography mainly focuses on single-table substitution and multi-table substitution ciphers.

In single-table substitution ciphers, all encryption methods have almost one commonality: the plaintext one-to-one corresponded. There are generally two ways to crack: brute force and word frequency analysis. A good example of this is the Caesar cipher. When you use this cipher, each letter in plaintext is moved backward or forward by a fixed number as ciphertext according to its order in the alphabet.

In multi-table substitution ciphers, the encrypted letters almost no longer maintain the original frequency, requiring the use of the algorithm to achieve the corresponding weakness to crack. One example of these is the Vigenere code. It uses a series of Caesar ciphers to form a cipher alphabet, a simple form of multi-table ciphers.

Modern cryptography mainly includes the following aspects: symmetric cryptography (DES, AES, RC4), asymmetric cryptography (RSA), hash function (MD5, SHA-1, SHA-512), and digital signature (RSA, DSA). Symmetric encryption is mainly divided into two ways: block cipher and stream cipher.

Stream ciphers process information byte by byte or bit by bit. Typically the key length will be the same as the length of the plaintext, and the key is derived from a shorter key, with the derived algorithm being a pseudo-random number generation algorithm.

Block ciphers encrypt a piece of plaintext each time, and include IDEA encryption, DES encryption, and AES encryption. Two blocks are often needed for block encryption for padding and packet encryption mode.

For asymmetric cryptography, the keys used by the encrypter and decrypter are different. The RSA algorithm is based on the fact that finding the factors of a large composite number is difficult. It uses a discrete logarithm and uses a one-way function. More information on this method is described in the walkthrough **Introduction to RSA Encryption**.

A hash function compresses a message or data into a digest, making the amount of data smaller. We use hashes to ensure the integrity of a message and as a redundancy check. It's a one-way password file and allows for signature detection in intrusion detection and virus detection. More information on hashing can be found in the walkthrough **Introduction to Hashing**.

Digital signatures are mainly used to sign digital messages in case of impersonation or falsification of messages, and can also be used for identity authentication of both parties. It relies on asymmetric cryptography.

Tools

- `FeatherDuster` - automated, modular cryptanalysis tool
- `Hash Extender` - utility tool for performing hash length extension attacks
- `PkCrack` - tool for breaking PkZip-encryption
- `RSATool` - generate private key with knowledge of p and q
- `XORTool` - analyzes multi-byte xor cipher
- `Cyberchef` - online tool for decrypting ciphers

Web

These challenges focus on analyzing websites to gain information, through their source code, hierarchy of directories, and functioning ports. Topics in this class include SQL injection, XSS cross-site scripting, CSRF cross-site request forgery, file uploading, file inclusion, framework security, PHP common vulnerabilities, code auditing, and more. There are more walkthroughs that go more in-depth to these topics that can be found in the library.

SQL injections involve injecting SQL syntax into user-controllable parameters, thereby destroying the original SQL structure. XSS Cross-site scripting is when you insert malicious HTML code into the webpage. Command execution involves injecting a malicious system command into a normal command that is called by the application.

CSRF cross-site request forgery is an attack that causes a logged-in user to perform an action, typically stealing user data. SSRF server-side request forgery is constructed by an attacker to form a request initiated by a server, and the target of these attacks are usually an internal system that is inaccessible from the external network. File uploads that are allowed by the webpage could allow executable files and scripts to be uploaded to the server and cause it to fail.

These are just some examples of topics and exercises that can be found in the web exploitation category in CTFs.

Tools

- `BurpSuite` - graphical tool to test website security
- `Postman` - add on for chrome for debugging network requests
- `Raccoon` - high-performance offensive security tool for reconnaissance and vulnerability scanning
- `SQLMap` - automatic SQL injection and database takeover tool
- `W3af` - web application attack and audit framework

Reverse Engineering

The goal of reverse engineering is to collect new information and understanding of a technology through disassembling it to its base parts. You typically analyze software structure, process, algorithm, code, etc. You should be familiar with related knowledge (operating systems, assembly language, and encryption/decryption) and have experience in programming with a variety of high-level languages.

The typical reversing process involves collecting information using static analysis tools before studying the protection methods of the program (code obfuscation, protective shell, and anti-debugging techniques) and trying to break/bypass protection. Then, you should disassemble the target software, locating the key code for analysis.

You can also perform reverse engineering through dynamic analysis. This is where you locate the key code and verify its inference or understand the program function by outputting important information (register, memory change, and program output) during the running of the program. You typically do this through debugging, symbol execution, and stain analysis.

Tools

- `ApkTool` - Android decompiler
- `Barf` - binary analysis and reverse engineering framework
- `Binary Ninja` - binary analysis framework

- `BinWalk` - analyze, reverse engineer, and extract firmware images
- `Boomerang` - decompile x86 binaries to C
- `Frida` - dynamic code injection
- `GDB` - the GNU project debugger
- `GEF` - GDB plugin
- `IDA Pro` - most used reversing software
- `Jadx` - decompile Android files

Forensics

Forensics focuses on recovering the digital trail left on a computer. There are many different methods to find data that's been seemingly deleted, not stored, or covertly recorded. These challenges in a CTF often involve file format analysis, steganography, memory dump analysis, or network packet capture analysis. For these challenges, you usually want to have some basic knowledge of scripting, manipulating binary data, and recognizing formats, protocols, structures, and encodings.

Many of these challenges require an initial analysis for you to determine what to do next. This is usually where you search plain-text strings in a file, use `grep` to search for specific strings, and use `hexdump` which displays the content of a file in hexadecimal.

Occasionally, you can have a challenge that involves a full disk image (computer file containing the contents and structure of a disk volume or of an entire data storage device). You have to mount the disk image file before searching for the flag. For these challenges, knowledge of well-know file systems is beneficial (NTFS, FAT, EXT, and HFS).

Network traffic challenges involve Packet capture (PCAP) files using programs like `tcpdump` or `Wireshark`. With these PCAP files, sometimes you have to recover a transferred file or transmitted secret.

Tools

- `Audacity` - analyze sound files
- `Bkhive` and `Samdump2` - dump SYSTEM and SAM files
- `CFF Explorer` - PE Editor
- `Creddump` - dump windows credentials
- `Foremost` - extract particular kind of files using headers
- `NetworkMiner` - network forensic analysis tool
- `Shellbags` - investigate NT_USER.dat files
- `UsbForensics` - contains many tools for USB forensics
- `Volatility` - to investigate memory dumps

- `extundelete` - find deleted files in EXT3 and EXT4 file systems
- `TestDisk` - recover missing partition tables, fix corrupted ones, undelete files on FAT or NTFS, etc.

Steganography

These challenges usually involve finding hints or flags that have been hidden, usually within media files. It usually consists of a media file images or videos that appear to have nothing in them at first sight. To obtain the flag, you have to run the media through filters and algorithms.

Tools

- `Steghide` - hide data in various kind of images
- `Stegsolve` - apply various steganography techniques to images
- `Zsteg` - PNG/BMP analysis
- `Exiftool` - read and write meta information in files
- `Pngtools` - for various analysis related to PNGs

Pwn / Binary Exploitation

The goal of these challenges is to acquire access to a target system without the system administrator's permission. Target systems include personal computers, servers, websites, and networking devices or applications. Useful background knowledge of assembly language and C language would be helpful in these challenges. Exploits that you use can include things like buffer overflows, return oriented programming (ROP), global offset table (GOT), and format strings.

Often, pwn challenges will require you to gain access to a remote server and then exploit the binary on that server. This is done by connecting to the server (using something like `netcat`) before sending commands to the server. You're typically given an IP address and a port number to connect to. After connecting, you're going to want to figure out what the binary actually does before trying to find a vulnerability within it. You can then create a Python script with `pwntools` and obtain the flag.

Tools

- `Python` - programming language that can be used to automate tasks and interact with the server
- `pwntools` - python library for rapid exploit development
- `GDB` - a debugger for binaries
- `Ghidra` - reverse engineering tool for binaries

OSINT

Open Source Intelligence (OSINT) challenges focus on collecting and analyzing data gathered from open sources (overt and publicly available sources). Most of these problems can be solved in different ways.

These challenges vary but can involve tracking social media or collecting various metadata. You're often looking for flags that are contained in the information you're tracking down (e.g. looking for someone's personal information and finding a flag in their password). Below is a list of some tools, but there are a variety of ones you can use.

Tools

- `whois` - Linux command that retrieves various metadata about a given domain
- `Wayback Machine` - digital archive of the World Wide Web that stores archived copies of obsolete pages
- `curl` - command line tool that enables data transfer over various network protocols
- `sherlock` - allows finding social-media accounts by username
- `Maltego` - real-time data mining and information gathering from dispersed sources
- Reverse image search on Google

Practice

Below are multiple websites to practice and learn the skills commonly found in CTF competitions. Many of them host their own CTF competitions and also offer practice problems that you can solve at any time, regardless if there's a competition running or not.

- PicoCTF
- OverTheWire
- CTFLearn
- HackTheBox

Additional Resources

1. CTF Playbook

Sources

1. How to get started in CTF | Complete Beginner Guide
2. Your Intro to Capture The Flag (CTF)
3. Getting Started | CTF Wiki

4. Forensics | SANReN CSC
5. Approaching CTF OSINT Challenges - Learn by Example
6. An Introduction to PWN Challenges