CTF categories

General Skills

1) Linux / Command Line
    a) Students will understand the uses of the command line.
    b) Students will be able to ssh to a server.
    c) Students will be able utilize nc to connect to a network service.
    d) Students will be able to utilize the following commands to change/create/delete files/directories:
        i) pwd
        ii) cd
        iii) rm
        iv) cp
        v) rm
        vi) Mkdir
    e) Students will be able to run C executables.
    f) Students will be able to run Python...
        i) … scripts in the command line
        ii) … scripts in a file
    g) Students will be able to open and edit files utilizing the following commands:
        i) cat
        ii) more
        iii) Shell redirection / shell piping
    h) Students will be able to utilize other commands in command line including:
        i) file
        ii) scp
        iii) whoami
    i) Students will be able make and edit files using a command line text editor (such as nano).
    j) Interfacing with the Command Line
        i) Students will understand the use of:
            (1) stdin / stdout / stderr
        ii) Students will be able to utilize pipes.
        iii) Students will be able to redirect stdin and stdout to be captured by other commands.
        iv) Students will learn how to use man pages.
    k) Students will be able to utilize the following commands for searching and manipulating files and directories:
        i) find
        ii) grep
        iii) strings

l) Students will be able to write basic bash scripts to interact with the command line.
m) Students will be able to understand the difference between relative and absolute paths.
n) Students will be able to interact with environment variables.
o) Students will be able to make and view hidden files.

2) Students will understand how data is represented on a computer:
    a) Different number systems (Binary, Octal, Base-64, Hexadecimal)
    b) ASCII (ANSI) text
    c) 2s Complement numbers
    d) Little/Big Endian

Web

1) Students will be able to view and understand the page source of a web page.
    a) Students will be able to identify and understand basic javascript code.
        i) Students will be able to identify static comparisons to values in a web page.
2) Students will understand HTTP as it is used today, and the two main methods of transmitting user data:
    a) Get
    b) Post
3) Students will be able to understand the use of cookies, including:
    a) editing/modifying
    b) User agent
4) Students will understand the need for the use of HTTPS.
5) Students will understand the difference between clientside and server side processing.
    a) Students will understand the basic functionality of PHP.
6) SQL/NoSQL
    a) Students will be able to perform basic SQL queries.
    b) Students will be able to perform basic SQL injection attacks.
    c) Students will be able to perform blind SQL injection attacks.
7) Students will understand how to gain Remote Code Execution on systems.
8) Students will be able to modify/add HTTP headers to outgoing requests
    a) GET
    b) POST

Forensics

1) Students will be able to perform file carving.
    a) Students will understand how headers and footers in files work.
    b) Students will understand how magic numbers function.
2) Students will understand the deleting process and recover deleted files.
3) Students will be able to analyze pcaps of captured data.
4) Students will be able to extract metadata from a file.
5) Students will understand the rationale of steganography.
    a) Students will be able to use specific programs to extract data.
    b) Students will be able to hide/extract using the least significant bit.
6) Students will be able to look through logs to find malicious activities.

Cryptography

1) Students will understand the issue with reusing a one-time pad.
2) Students will be able to apply known and chosen plaintext attacks on a ciphertext.
3) Students will understand how to leverage flaws in usage of pseudo-random number generators.
4) Students will be able to apply dictionary attacks to cracking passwords.
5) Students will be able to brute force a key with/without mistakes in implementation.
    a) Students will understand the role of entropy in modern cryptography.
    b) Students will be able to use frequency analysis to defeat some crypto systems.
6) Students will be able to break the following historical ciphers:
    a) Caesar Cipher
    b) Vigenere Cipher
7) Students will be introduced to the following crypto primitives:
    a) Symmetric Key
    b) Public Key Crypto (Asymmetric Key)
    c) Pseudo Random Number Generator
8) Students will be able to break some block ciphers including:
    a) CBC
    b) ECB
        i) Students will be able to apply padding to a message.
        ii) Students will be able to attack using oracle attacks.
9) Students will be able to utilize the following math rules as they apply to cryptography:
    a) Factoring Large Numbers
    b) Mod prime arithmetic
10) RSA
    a) Students will be able to utilize the following basic attacks on RSA:
        i) primes too small
        ii) wrong exponent released
        iii) n product of more than 2 primes
        iv) Students will be able to utilize e too small

Binaries

1) Students will be able to read C programs.
2) Students will understand built-in C types (e.g. word, double-word).
3) Students will be able to understand assembly, and its relationship to given the C code.
4) Students will be able to write basic python programs/script.
5) Students will understand file formats for binaries.
6) Students will learn about registers in assembly.
7) Students will be able to understand basic assembly operations
    a) Arithmetic Operations
    b) Control Flow Operations
    c) Logic Operations
8) Students will understand the difference between Big Endian and Little Endian.
9) Students will learn about the memory/section layout of a Binary.
10) Students will understand the layout of the stack in 32-bit programs.
11) Students will understand the x86/x86-64 calling conventions.
12) Students will understand how to locate important information in a binary
    a) function symbols
    b) Strings
    c) Global Offset Table
    d) Process Linkage Table

Reversing

1) Students will learn the basics of using a disassembler.
2) Students will learn the basics of using a debugger, including:
    a) Breakpointing
    b) Reading/Writing memory
3) Students will learn about optimizing code through dynamic programming (e.g. memoization)
4) Students will learn how to patch a binary/file.
    a) Patching bytes
    b) Recompiling
5) Students will gain experience working with foreign architecture assembly.
6) Students will understand how to find/generate a solution given a set of constraints.

Binary Exploitation

1) Students will learn how to debug a program using gdb.
    a) Running
    b) Stepping
2) Students will learn how to exploit buffer overflow vulnerabilities.
    a) Stack Buffer Overflow
    b) Heap Buffer Overflow
3) Students will learn how to control program execution.
    a) Overwriting return address (+ arguments)
    b) Overwriting GOT
    c) Use syscall to get a shell (execve)
    d) Use syscall to execute arbitrary shellcode (mprotect/mmap)
    e) Sigreturn
    f) Writing shellcode given buffer/character constraints
4) Students will learn about format string attacks and how to avoid them.
    a) Understand the risk of a user inputted format string
    b) Leak addresses with a format string
    c) Write arbitrary data to arbitrary addresses using %n
5) Students will learn about signed/unsigned overflow vulnerabilities.
6) Students will learn about techniques to squander exploitation and how to bypass them
    a) Canaries
        i) Spot where canaries are
        ii) Leak canary from stack/heap
    b) Stack randomization - ASLR
        i) Nop sled
        ii) Return to libc
        iii) Partial address overwrite
        iv) Brute-force (32-bit)
    c) NX-bit
        i) Retn-libc
        ii) ROP
        iii) Control Flow Integrity (CFI)
    d) Full Stack RELRO (Relocation Read-Only)
        i) malloc hook, printf hook
        ii) One gadget
7) Students will understand the basics of the C memory allocator (glibc malloc).
    a) Dangling pointer/Use after free
    b) Double-free
8) Students will learn how to manipulate data to leak addresses and write to memory.
9) Students will understand the Memory layout of Linux x86-64 userspace programs
    a) code/data/bss
    b) Heap
    c) Stack

10) Students will gain familiarity with the C standard libraryd.