**Public URL:** [**https://sec.cybbh.io/public/security/latest/index.html**](https://sec.cybbh.io/public/security/latest/index.html)

Section 1: Understand x86\_64 Assembly

SKILL CCSE001: Understand x86\_64 ASM (Intel)

CCSE01.001 Explain datasizes, registers, and instructions

CCSE001.002 Describe the x86\_64 stack

SKILL CCOS002: Follow C Source Code

CCSE002.001 Demonstrate variables and functions in C programs

Section 2: Determining Binary Behavior

SKILL CCSE003: Binaries Debugging

CCSE003.001 Follow program flow utilizing common debugging tools

CCSE003.002 Implement program breakpoints

CCSE003.003 Modify flag values

SKILL CCOS04: Binary Disassembly

CCSE004.001 Disassemble binaries utilizing common tools

CCSE004.002 Follow disassembled program/binary flow

SKILL CCSE005: Determine execution behaivor using program flow

Section 3: Utilize flow of Program

SKILL CCSE006: Discover hardcoded, obscured binary data

SKILL CCSE007: Reverse mathematical formulas to determine binary data

CCSE007.001 Determine mathematical formulas necessary to enumerate data in binary

CCSE007.002 Develop script(s) to generate data based on mathematical formula

SKILL CCSE008: Determine obscured functionality of binary

CCSE008.001 Locate filesystem and OS internals modification(s)

Section 4: Defeating Binary Protections

SKILL CCSE009: Bypass Binary Protections

CCSE009.001 - Defeat binary obfuscation

CCSE009.002 - Circumnavigate anti-debugging techniques

Section 5: Exploit Development

SKILL CCSE010: Reverse Engineer a given binary

CCSE010.001 - Discover functionality of binary

CCSE010.002 - Find unbounded functionality within a binary

SKILL CCSE011: Discover exploitable memory errors

CCSE011.001 - Attempt binary exploitation techniques

CCSE011.002 - Utilize FOSS tools to find unprotected segments

SKILL CCSE012: Develop and deploy a method for using the exploit

CCSE012.001 - Determine environment to be used in

CCSE012.002 - Create script/program to utilize developed exploit in the given environment

CCSE012.003 - Deploy exploit to target environment

Section 6: Write a Penetration Test Report

SKILL CCSE013: Understand how to write a penetration test report

CCSE013.001 - Document all actions taken during penetration test

CCSE013.002 - Organize documented actions into detailed chronology

CCSE013.003 - Author executive summary of chronology

Section 7: Perform a Penetration Test

SKILL CCSE014: Define the phases of a Penetration Test

CCSE014.001 - Plan/Scope a penetration test

Section 7.1: Network Scanning and Reconnaissance

SKILL CCSE015: Perform network scanning and reconnaissance

CCSE0015.001 - Perform OSINT gathering

CCSE015.002 - Using NMAP NSE scripts

CCSE015.003 - Utilizing native tools

Section 7.2: Gaining Access

SKILL CCSE016: Perform exploitation methods

Section 7.2.1: Web Exploitation

SKILL CCSE017: Perform SQL Injection

CCSE017.001 - Identify unsanitized input fields

CCSE017.002 - Perform blind and Out-of-Band SQL injection

SKILL CCSE018: Upload Malicious files

CCSE018.001 - Gain shell access to a webserver

SKILL CCSE019: Perform URL injection

CCSE019.001 - Inject commands into URL strings

CCSE019.002 - Exploit directory traversal vulnerabilities

SKILL CCSE020: Perform XSS

CCSE020.001 - Perform stored XSS

CCSE020.002 - Perform reflected XSS

Section 7.2.2: Local Enumeration

SKILL CCSE021: Enumerate a local host

CCSE021.001 - Determine users of system

CCSE021.002 - Enumerate processes

CCSE021.003 - Enumerate running services

CCSE021.004 - Identify network connections

CCSE021.005 - Identify other actors'' pressence

CCSE021.006 - Identify malware on system

Section 7.2.3: Privilege Escalation

SKILL CCSE022: Perform Windows Privilege Escalation

CCSE022.001 - Bypass UAC via auto-elevated binary abuse

CCSE022.002 - DLL hijacking/vulnerable privileged software

CCSE022.003 - Escalation through services and scheduled tasks

CCSE022.004 - Unpatched Kernel Vulnerabilities

SKILL CCSE023: Perform Linux Privilege Escalation

CCSE023.001 - Weak sudo configurations

CCSE023.002 - Escape Shells with vi, awk and Python

CCSE023.003 - Vulnerable SUID/GUID executables

CCSE023.004 - Insecure file permissions (crons, directories, files, etc)

CCSE023.005 - Unpatched Kernel vulnerabilities

CCSE023.006 - Vulnerable privileged software

Section 7.3: Post exploitation

SKILL CCSE024: Define post-exploitation tasks

CCSE024.001 - Exfiltrate data

CCSE024.002 - Pivot throughout a networ

Section 7.4: Maintaining Persistence

SKILL CCSE025: Maintain Windows Persistence

CCSE025.001 - Add or hijack a user account

CCSE025.002 - Inject persistent registry entries

CCSE025.003 - Create scheduled tasks/jobs

CCSE025.004 - Hijack existing autorun process or service

SKILL CCSE026: Maintain Linux Persistence

CCSE026.001 - Add or hijack a user account

CCSE026.002 - Implement boot process persistence

CCSE026.003 - Cron jobs

CCSE026.004 - Kernel Modules

Section 7.5: Covering Tracks

SKILL CCSE027: Clean up forensically relevant artifacts

CCSE027.001 - Sanitize logs

CCSE027.002 - Modify timestamps

CCSE027.003 - Remove tools

CCSE027.004 - Obfuscate persistence

CCSE027.005 - Reduce activity from memory/swap/cach