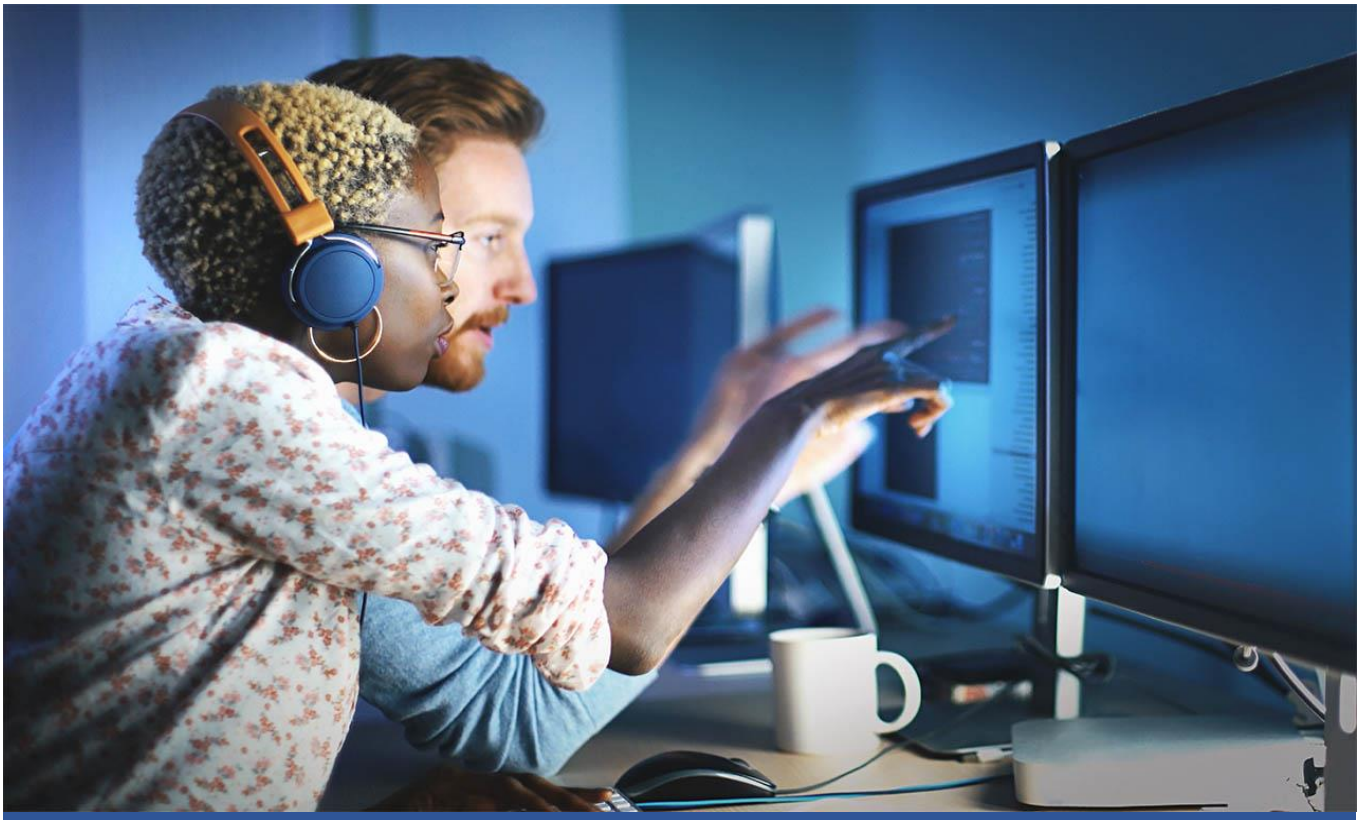
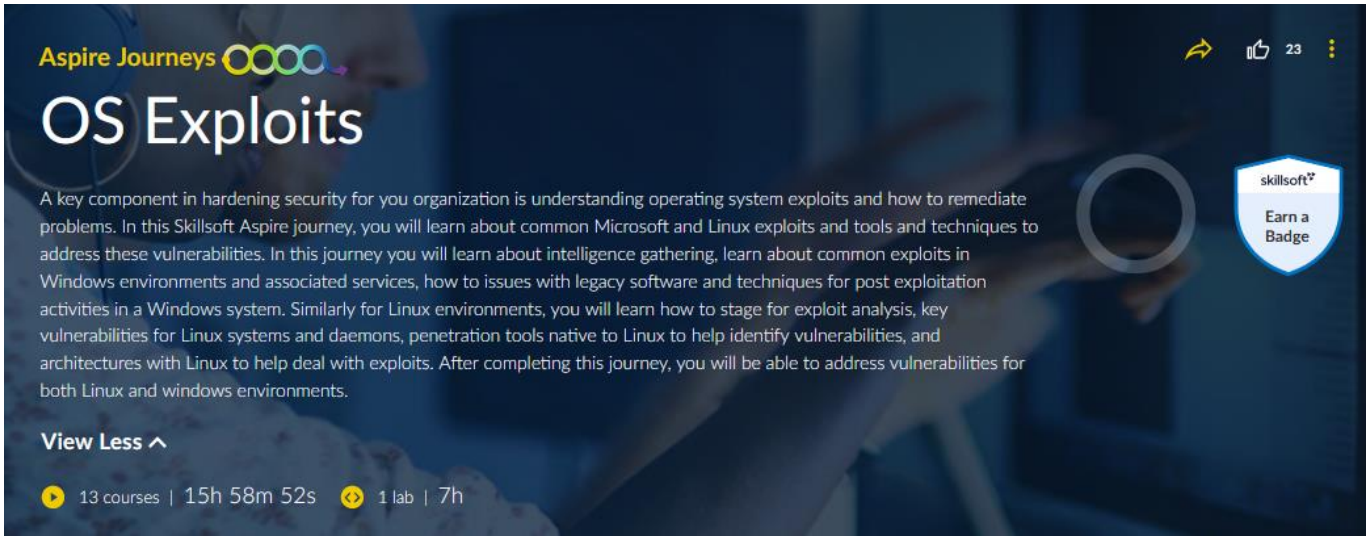


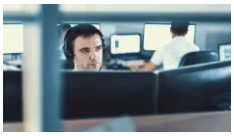
OS Exploits

Skillsoft Aspire Journey

skillsoft

A key component in hardening security for you organization is understanding operating system exploits and how to remediate problems. In this Skillsoft Aspire journey, you will learn about common Microsoft and Linux exploits and tools and techniques to address these vulnerabilities. In this journey you will learn about intelligence gathering, learn about common exploits in Windows environments and associated services, how to issues with legacy software and techniques for post exploitation activities in a Windows system. Similarly for Linux environments, you will learn how to stage for exploit analysis, key vulnerabilities for Linux systems and daemons, penetration tools native to Linux to help identify vulnerabilities, and architectures with Linux to help deal with exploits. After completing this journey, you will be able to address vulnerabilities for both Linux and windows environments.





Track 1: OS Exploits

12 courses | 14h 24m 13s

PREREQUISITES

In order to fully profit from the potential of this Aspire Journey, we recommend the following prerequisite skills:

- Be familiar with Windows OS
- Be familiar with Linux OS

Track 1: OS Exploits (duration: 14h 24m 13s)

Windows Exploits and Forensics: Intelligence Gathering

Objectives

- identify open source intelligence gathering techniques and sources
- conduct an OSINT investigation on a public document
- identify what to look for using social media and other tools when finding targets for social engineering exercises
- outline how to scan a network for open ports
- conduct an Nmap scan of a Windows-based network
- identify common Windows services and their ports
- outline how to scan a system and name tools used to conduct basic enumeration
- conduct a scan of a Windows-based system and enumerate data
- identify basic tools used within the Kali hacking environment
- use basic commands and recognize common issues within Metasploitable
- recognize common locations to find Windows exploits

Windows Environments

Objectives

- recognize the standard security features and controls placed on Windows hosts
- identify different Windows Server operating systems and their various uses within the environment
- recognize the role of intrusion detection systems (IDS) and intrusion prevention systems (IPS) within a Windows environment
- outline the MITRE ATT&CK framework and how it relates to Windows intrusions
- identify the location of command Windows-based logs and the event viewer
- view Windows event logging in action
- name the various user and service accounts within a Windows Active Directory environment
- use basic Windows and PowerShell commands
- outline how NTFS and Active Directory permissions work and some of their common misconfigurations
- describe the hashing algorithm used to store Windows passwords
- crack an NTLM hash value using several tools
- use the Windows Registry and recognize the different artifacts contained within
- list and describe various artifacts created within the Windows operating system
- outline how Kerberos works and some common Active directory misconfigurations

Windows Exploits and Forensics: SMB & PsExec

Objectives

- outline how SMB works and how permissions are set
- list various tools and techniques used to enumerate SMB
- enumerate SMB information from an active machine
- outline how to identify potential vulnerabilities in SMB
- outline various methods of attacking SMB
- conduct a brute force attack against an SMB service
- conduct a denial of service attack on the SMB service
- exploit a system to gain a reverse shell on a Windows machine
- define what PsExec is and describe how it works
- use PsExec to execute commands on a remote machine
- use Mimikatz to "pass the hash" and steal logon credentials
- describe the background of the EternalBlue exploit and outline how it works on Windows systems
- conduct an attack on a system using EternalBlue

Windows Exploits and Forensics: FTP, RDP, & Other Services

Objectives

- recognize how to exploit common Windows services, such as FTP, RDP, and others
- enumerate data from an FTP
- outline the various methods of attacking FTP services
- conduct a brute force attack against an FTP server
- discover IIS and how it relates to Windows and FTP Clients
- use ASP to gain a reverse shell on a Windows machine
- outline what RDP is and how it works within a Windows environment
- state various methods of attacking the Windows RDP service
- enumerate a Windows machine using the RDP service
- exploit an RDP system using the BlueKeep vulnerability
- describe the features of WMI and how it works
- exploit WMI on a Windows-based system

Windows Exploits and Forensics: Legacy Systems & Third Party Applications

Objectives

- identify common attacks against legacy Windows host-based machines
- identify common attacks against legacy Windows Server-based machines
- scan a Windows Server 2008 environment for potential vulnerabilities
- enumerate data from services running on Windows Server 2008 hosts
- run an exploit on a Windows Server 2008-based machine to gain user credentials
- run an exploit on a Windows-based machine to gain a reverse shell
- list common third-party applications used in Windows environments
- outline how to find vulnerabilities for third-party applications
- exploit a third-party application and gain access to a system
- recognize a honeypot and how to avoid falling into their trap

Windows Exploits and Forensics: Post Exploitation

Objectives

- recognize various user levels and methods of privilege escalation within Windows
- conduct a basic privilege escalation on a Windows machine
- use a DLL injection to escalate user privileges on a Windows machine
- describe the concept of pivoting within a Windows environment and typical end goals
- use CrackMapExec to steal user credentials from a Windows machine
- use PowerView to enumerate information from an exploited Windows machine in order to pivot the attack
- use BloodHound to 'walk the dog', identifying Active Directory security issues and gaining domain admin privileges
- recognize cleanup methods used post exploitation to hide your tracks
- perform post attack cleanup tasks
- recognize what an advanced persistent threat (APT) is and methods used to configure them
- configure an APT on a system after exploitation
- use a ransomware attack as a quick method to clean up post attack

Linux Exploits & Mitigation: Staging for Exploit Analysis

Objectives

- establish an approach to using virtual environments to stage exploits
- set up QEMU and its dependencies for machine emulation and virtualization
- launch an instance of Alpine Linux within a QEMU environment
- mount the QEMU virtual drive to copy files into and out of a QEMU virtual machine
- compile a version of the Linux kernel
- configure networking options in a QEMU virtual environment
- describe architectural considerations based on the targeted platform
- emulate ARM in QEMU to emulate the Alpine Linux ARM variant
- take and restore snapshots of virtual machines using QEMU Monitor
- monitor system information from a staging environment using QEMU Monitor
- recognize escape vulnerabilities from virtual machines to hosts
- describe safeguards and considerations when running insecure programs in virtual environments

Linux Exploits & Mitigation: Program Essentials

Objectives

- describe a program's structure in memory in terms of address space layout
- run gdb to step through and trace debug a C program
- run gdb to disassemble a program into its assembly code
- run objdump and readelf to disassemble and inspect a Linux program
- describe how data and functionality are protected by separating computing resources
- discuss how data and functionality are protected within the Linux operating system by kernel and userland separation
- describe the GNU C Library (glibc) and how it integrates with the Linux kernel
- interface with the Linux kernel through system calls in C
- interface with the Linux kernel through system calls in Assembly
- describe the main components of the Linux system call table
- query system calls available in your installed version of Linux
- analyze simple Linux program system calls using strace
- explore how programs are segmented between their text, data, and BSS segments

Linux Exploits & Mitigation: String Vulnerability Analysis

Objectives

- describe how strings are exploited in computer programs
- illustrate the weaknesses caused by string formatting methods
- perform a string buffer overflow in a C program
- apply flags to the gcc compiler to catch string weaknesses by converting warnings into errors
- recognize and correct weaknesses introduced by poorly implemented string copies
- recognize and correct common input string vulnerabilities
- explore how generating command line string inputs can exploit insecure string methods
- check input strings for validity and safety
- perform loops over characters in a string in a safe manner
- run programs that fail due to unsafe strings
- describe how strings executed dynamically can lead to vulnerabilities
- recognize safe and unsafe methods of returning strings in C

Linux Exploits & Mitigation: Memory and Pointer Vulnerabilities

Objectives

- describe methods and goals for allocating memory
- investigate what it means to overflow the heap
- recognize and avoid dangling pointers in a C program
- recognize and avoid null dereferences in a C program
- investigate what it means to exploit the heap
- illustrate use-after-free (UAF) vulnerabilities
- recognize and avoid stack buffer overflows
- describe the nature of out-of-bounds write vulnerabilities and their impact
- recognize and avoid looping off-by-one in a C program
- describe how coding errors and vulnerabilities lead to corrupting memory
- illustrate how to execute arbitrary code introduced by coding errors
- illustrate how out-of-bounds errors are exploited

Linux Exploits & Mitigation: Penetration Tools

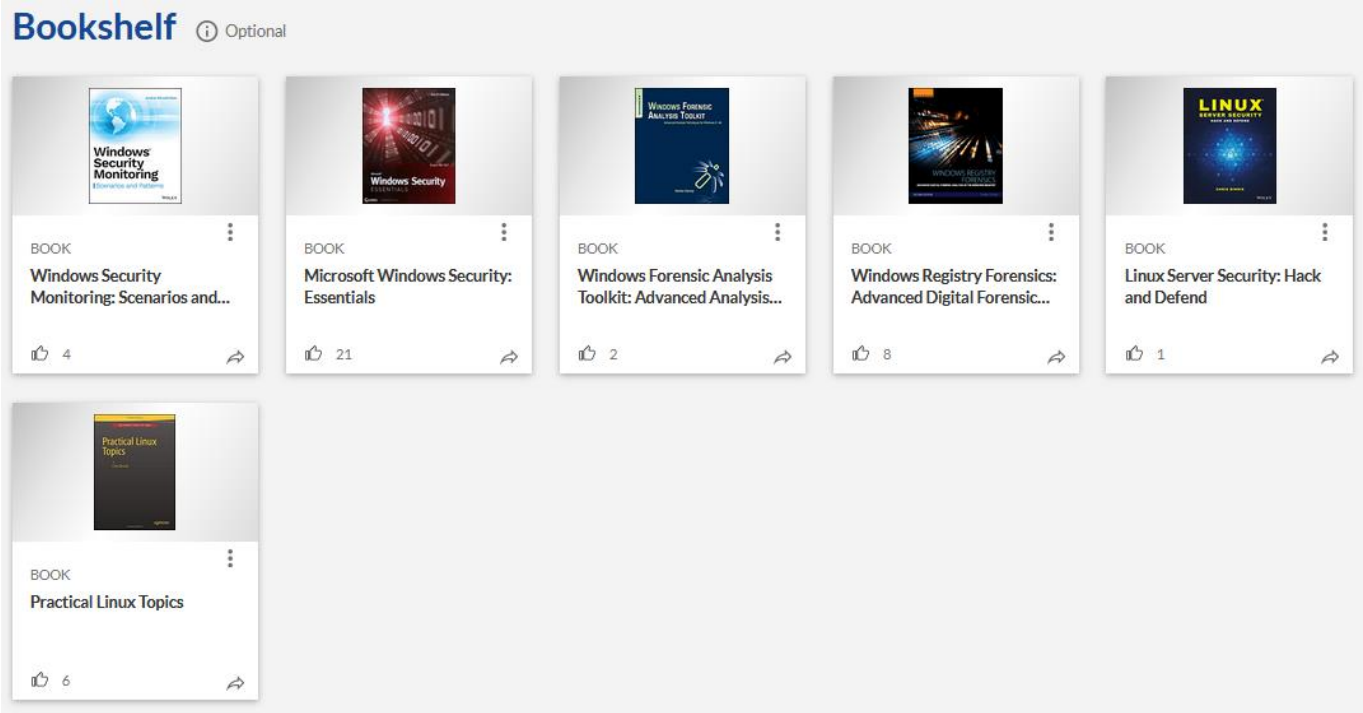
Objectives

- navigate the basic commands used to prepare exploit tests using Metasploit
- run the Metasploitable vulnerable virtual machine
- prepare a Metasploit command that exploits a vulnerable web service
- modify options used to vary the operation of a Metasploit command
- exploit an outdated Samba file sharing service to gain root access using Metasploit
- test for command injection vulnerabilities in web-based applications using Commix
- search for exploits and shellcodes using Exploit Database
- detect Linux security weaknesses using the Linux Exploit Suggester utility
- scan for potential router vulnerabilities using RouterSploit
- resolve opcodes and syscall numbers from binary executables using ShellNoob
- convert shellcode between binary and text formats using ShellNoob
- explore the use of SQL injection attacks and protections against them using SQLMap

Linux Exploits & Mitigation: Linux Exploit Architecture

Objectives

- describe race conditions, their potential for vulnerabilities, and approaches to avoiding race conditions
- disable compiler protections to construct and execute shellcode in C
- describe out-of-order execution and related processor concepts and vulnerabilities
- describe common weaknesses and errors made when working with integers and how to prevent them
- explore compiler warnings that are vital to security and program stability
- explore how stack smashing vulnerabilities occur and how they are mitigated
- describe use-after-free vulnerabilities, how they occur, and typical target examples to keep in mind
- describe the impact and mitigations in place to avoid and mitigate the Spectre and Meltdown vulnerabilities
- describe the Write XOR Execute (W^X) feature and its impact on memory security
- various processor and operating system considerations that need to be taken into account when developing mitigations to vulnerabilities and exploits
- targets for privilege escalation exploits and common privilege control mechanisms
- targets for exploiting processes and tasks of a running Linux system



FOLLOW US ON:

www.skilltech.pl

email: biuro@skilltech.pl

tel. +48 22 44 88 827

SkillTech
Technology hired for excellence