

SEC660: Advanced Penetration Testing, Exploit Writing, and Ethical Hacking

This course is designed as a logical progression point for those who have completed **SEC560: Network Penetration Testing and Ethical Hacking**, or for those with existing penetration testing experience. Students with the prerequisite knowledge to take this course will walk through dozens of real-world attacks used by the most seasoned penetration testers. The methodology of a given attack is discussed, followed by exercises in a real-world lab environment to solidify advanced concepts and allow for the immediate application of techniques in the workplace. Each day includes a two-hour evening bootcamp to allow for additional mastery of the techniques discussed and even more hands-on exercises. A sample of topics covered includes weaponizing Python for penetration testers, attacks against network access control (NAC) and VLAN manipulation, network device exploitation, breaking out of Linux and Windows restricted environments, IPv6, Linux privilege escalation and exploit-writing, testing cryptographic implementations, fuzzing, defeating modern OS controls such as ASLR and DEP, Return Oriented Programming (ROP), Windows exploit-writing, and much more!

Attackers are becoming more clever and their attacks more complex. In order to keep up with the latest attack methods, one must have a strong desire to learn, the support of others, and the opportunity to practice and build experience. SEC660 engages attendees with in-depth knowledge of the most prominent and powerful attack vectors and an environment to perform these attacks in numerous hands-on scenarios. This course goes far beyond simple scanning for low-hanging fruit, and shows penetration testers how to model the abilities of an advanced attacker to find significant flaws in a target environment and demonstrate the business risk associated with these flaws.

SEC660 starts off by introducing advanced penetration concepts and providing an overview to help prepare students for what lies ahead. The focus of day one is on network attacks, an area often left untouched by testers. Topics include accessing, manipulating, and exploiting the network. Attacks are performed against NAC, VLANs, OSPF, 802.1X, CDP, IPv6, VOIP, SSL, ARP, SNMP, and others. Day two starts off with a technical module on performing penetration testing against various cryptographic implementations. The rest of the day is spent on network booting attacks, escaping Linux restricted environments such as chroot, and escaping Windows restricted desktop environments. Day three jumps into an introduction of Python for penetration testing, Scapy for packet crafting, product security testing, network and application fuzzing, and code coverage techniques. Days four and five are spent exploiting programs on the Linux and Windows operating systems. You will learn to identify privileged programs, redirect the execution of code, reverse-engineer programs to locate vulnerable code, obtain code execution for administrative shell access, and defeat modern operating system controls such as ASLR, canaries, and DEP using Return Oriented Programming (ROP) and other techniques. Local and remote exploits, as well as client-side exploitation techniques are covered. The final course day is dedicated to numerous penetration testing challenges requiring you to solve complex problems and capture flags.

Who Should Attend

- Network and systems penetration testers
- Incident handlers
- Application developers
- IDS engineers

You Will Be Able To

- Perform fuzz testing to enhance your company's SDL process
- Exploit network devices and assess network application protocols
- Escape from restricted environments on Linux and Windows
- Test cryptographic implementations
- Model the techniques used by attackers to perform 0-day vulnerability discovery and exploit development
- Develop more accurate quantitative and qualitative risk assessments through validation
- Demonstrate the needs and effects of leveraging modern exploit mitigation controls
- Reverse-engineer vulnerable code to write custom exploits

“The CTF with teams was awesome!!! I learned a lot more when working through some of the issues with my peers.”

-MIKE EVANS, ALASKA AIRLINES



giac.org



sans.org/cyber-guardian



sans.edu

660.1 HANDS ON: Network Attacks for Penetration Testers

Day one serves as an advanced network attack module, building on knowledge gained from **SEC560: Network Penetration Testing and Ethical Hacking**. The focus will be on obtaining access to the network; manipulating the network to gain an attack position for eavesdropping and attacks, and for exploiting network devices; leveraging weaknesses in network infrastructure; and taking advantage of client frailty.

Topics: Bypassing Network Admission Control; Impersonating Devices with Admission Control Policy Exceptions; Exploiting EAP-MD5 Authentication; IEEE 802.1X Authentication; Custom Network Protocol Manipulation with Ettercap and Custom Filters; Multiple Techniques for Gaining Man-in-the-Middle Network Access; Exploiting OSPF Authentication to Inject Malicious Routing Updates; Using Evilgrade to Attack Software Updates; Overcoming SSL Transport Encryption Security with Sslstrip; Remote Cisco Router Configuration File Retrieval

660.2 HANDS ON: Crypto, Network Booting Attacks, and Escaping Restricted Environments

Day two starts by taking a tactical look at techniques penetration testers can use to investigate and exploit common cryptography mistakes. We finish the module with lab exercises that allow you to practice your new-found crypto attack skill set against reproduced real-world application vulnerabilities.

Topics: Low Profile Enumeration of Large Windows Environments Without Heavy Scanning; Strategic Target Selection; Remote Desktop Protocol (RDP) and Man-in-the-Middle Attacks; Windows Network Authentication Attacks (e.g., MS-Kerberos, NTLMv2, NTLMv1, LM); Windows Network Authentication Downgrade; Discovering and Leveraging MS-SQL for Domain Compromise Without Knowing the sa Password; Metasploit Tricks to Attack Fully Patched Systems; Utilizing LSA Secrets and Service Accounts to Dominate Windows Targets; Dealing with Unguessable/Uncrackable Passwords; Leveraging Password Histories; Gaining Graphical Access; Expanding Influence to Non-Windows Systems

660.3 HANDS ON: Python, Scapy, and Fuzzing

Day three starts with a focus on how to leverage Python as a penetration tester. It is designed to help people unfamiliar with Python start modifying scripts to add their own functionality while helping seasoned Python scripters improve their skills. Once we leverage the Python skills in creative lab exercises, we move on to leveraging Scapy for custom network targeting and protocol manipulation. Using Scapy, we examine techniques for transmitting and receiving network traffic beyond what canned tools can accomplish, including IPv6.

Topics: Becoming Familiar with Python Types; Leveraging Python Modules for Real-World Pen Tester Tasks; Manipulating Stateful Protocols with Scapy; Using Scapy to Create a Custom Wireless Data Leakage Tool; Product Security Testing; Using Taof for Quick Protocol Mutation Fuzzing; IDAPro; Optimizing Your Fuzzing Time with Smart Target Selection; Automating Target Monitoring While Fuzzing with Sulley; Leveraging Microsoft Word Macros for Fuzzing .docx files; Block-Based Code Coverage Techniques Using Paimeir

660.4 HANDS ON: Exploiting Linux for Penetration Testers

Day four begins by walking through memory from an exploitation perspective as well as introducing x86 assembler and linking and loading. Processor registers are directly manipulated by testers and must be intimately understood. Disassembly is a critical piece of testing and will be used throughout the remainder of the course. We will take a look at the Linux OS from an exploitation perspective and discuss the topic of privilege escalation. We continue by describing how to look for SUID programs and other likely points of vulnerabilities and misconfigurations. The material will focus on techniques that are critical to performing penetration testing on Linux applications.

Topics: Stack and Dynamic Memory Management and Allocation on the Linux OS; Disassembling a Binary and Analyzing x86 Assembly Code; Performing Symbol Resolution on the Linux OS; Identifying Vulnerable Programs; Code Execution Redirection and Memory Leaks; Return Oriented Programming (ROP); Identifying and Analyzing Stack-Based Overflows on the Linux OS; Performing Return-to-libc (ret2libc) Attacks on the Stack; Defeating Stack Protection on the Linux OS; Defeating ASLR on the Linux OS

660.5 HANDS ON: Exploiting Windows for Penetration Testers

On day five we start off with covering the OS security features (ASLR, DEP, etc.) added to the Windows OS over the years, as well as Windows specific constructs, such as the process environment block (PEB), structured exception handling (SEH), thread information block (TIB), and the Windows API. Differences between Linux and Windows will be covered. These topics are critical in assessing Windows-based applications. We then focus on stack-based attacks against programs running on the Windows OS. We look at fuzzing skills, which are required to test remote services, such as TFTP and FTP, for faults. Once a fault is discovered, the student will work with Immunity Debugger to turn the fault into an opportunity for code execution and privilege escalation. Advanced stack-based attacks, such as disabling data execution prevention (DEP) and heap spraying for browser-based applications, are covered. Client-side exploitation will be introduced, as it is a highly common area of attack. The day will end with a look at shellcode and the differences between Linux and Windows.

Topics: The State of Windows OS Protections on XP, Vista, 7, Server 2003 and 2008; Understanding Common Windows Constructs; Stack Exploitation on Windows; Defeating OS Protections Added to Windows; Dynamic and Static Fuzzing on Windows Applications or Processes; Creating a Metasploit Module; Advanced Stack-Smashing on Windows; Return Oriented Programming (ROP); Windows 7 and Windows 8; Porting Metasploit Modules; Client-side Exploitation; Windows and Linux Shellcode

660.6 HANDS ON: Capture the Flag

This day will serve as a real-world challenge for students, requiring them to utilize skills obtained throughout the course, think outside the box, and solve simple-to-complex problems. In this offensive exercise, challenges range from local privilege escalation to remote exploitation on both Linux and Windows systems, as well as networking attacks and other challenges related to the course material.



SEC660 COIN

SEC660 Training Formats

(subject to change)



Live Training

sans.org/security-training/by-location/all



OnSite

sans.org/onsite



Simulcast

sans.org/simulcast



OnDemand

sans.org/ondemand



SelfStudy

sans.org/selfstudy