NEW

SEC760: Advanced Exploit Development for Penetration Testers

Vulnerabilities in modern operating systems such as Microsoft Windows 7/8, Server 2012, and the latest Linux distributions are often very complex and subtle. Yet these vulnerabilities could expose organizations to significant attacks, undermining their defenses when attacked by very skilled adversaries. Few security professionals have the skillset to discover let alone even understand at a fundamental level why the vulnerability exists and how to write an exploit to compromise it. Conversely, attackers must maintain this skillset regardless of the increased complexity. **SEC760: Advanced Exploit Development for Penetration Testers** teaches the skills required to reverse-engineer 32-bit and 64-bit applications, perform remote user application and kernel debugging, analyze patches for one-day exploits, and write complex exploits, such as use-after-free attacks, against modern software and operating systems.

Some of the skills you will learn in SEC760 include:

- **How to write modern exploits against the Windows 7 and 8 operating systems**
- **How to perform complex attacks such as use-after-free, Kernel exploit techniques, one-day exploitation through patch analysis, and other advanced topics**
- **The importance of utilizing a Security Development Lifecycle (SDL) or Secure SDLC, along with Threat Modeling**
- **How to effectively utilize various debuggers and plug-ins to improve vulnerability research and speed**
- **How to deal with modern exploit mitigation controls aimed at thwarting success and defeating determination**

Who Should Attend

- Senior network and system penetration testers
- Secure application developers (C & C++)
- Reverse-engineering professionals
- Senior incident handlers
- Senior threat analysts
- Vulnerability researchers
- Security researchers

You Will Be Able To

- Discover zero-day vulnerabilities in programs running on fully-patched modern operating systems
- Create exploits to take advantage of vulnerabilities through a detailed penetration testing process
- Use the advanced features of IDA Pro and write your own IDC and IDA Python scripts
- Perform remote debugging of Linux and Windows applications
- Understand and exploit Linux heap overflows
- Write Return Oriented Shellcode
- Perform patch diffing against programs, libraries, and drivers to find patched vulnerabilities
- Perform Windows heap overflows and use-after-free attacks
- Use precision heap sprays to improve exploitability
- Perform Windows Kernel debugging up through Windows 8 64-bit
- Jump into Windows kernel exploitation

What You Will Receive

- Various preconfigured *NIX virtual machines
- A course DVD with various tools that are required for use in class

760.1 HANDS ON: Threat Modeling, Reversing and Debugging with IDA

Many penetration testers, incident handlers, developers, and other related professionals lack reverse-engineering and debugging skills. This is a different skill than reverse-engineering malicious software. As part of the Security Development Lifecycle (SDL) and Secure-SDLC, developers and exploit writers should have experience using IDA Pro to debug and reverse their code when finding bugs or when identifying potential risks after static code analysis or fuzzing.

Topics: Security Development Lifecycle (SDL); Threat Modeling; Why IDA Is the #1 Tool for Reverse Engineering; IDA Navigation; IDA Python and the IDA IDC; IDA Plug-ins and Extensibility; Local Application Debugging with IDA; Remote Application Debugging with IDA

760.2 HANDS ON: Advanced Linux Exploitaiton

The ability to progress into more advanced reversing and exploitation requires an expert-level understanding of basic software vulnerabilities, such as those covered in SEC660. Heap overflows serve as a rite of passage into modern exploitation techniques. This day is aimed at bridging this gap of knowledge in order to inspire thinking in a more abstract manner, necessary for continuing further with the course. Linux can sometimes be an easier operating system to learn these techniques, serving as a productive gateway into Windows.

Topics: Linux Heap Management, Constructs, and Environment; Navigating the Heap; Abusing Macros such as unlink() and frontlink(); Function Pointer Overwrites; Format String Exploitation; Abusing Custom Doubly-Linked Lists; Defeating Linux Exploit Mitigation Controls; Using IDA for Linux Application Exploitation

760.3 HANDS ON: Patch Diffing, One-Day Exploits, and Return-Oriented Shellcode

Attackers generally download patches as soon as they are distributed by vendors such as Microsoft in order to find newly patched vulnerabilities. Often, vulnerabilities are disclosed privately, or even discovered in-house, allowing the vendor to more silently patch the vulnerability. This also allows the vendor to release limited or even no details at all about a patched vulnerability. Attackers are well aware of this and quickly work to find the patched vulnerability in order to take control of unpatched systems. This technique is also performed by incident handlers, IDS administrators and vendors, vulnerability and penetration testing framework companies, government entities, and others.

Topics: The Microsoft Patch Management Process and Patch Tuesday; Obtaining Patches and Patch Extraction; Binary Diffing with BinDiff, patchdiff2, turbodiff, and darungrim3; Visualizing Code Changes and Identifying Fixes; Reversing 32-bit and 64-bit Applications and Modules; Triggering Patched Vulnerabilities; Writing One-Day Exploits; Handling Modern Exploit Mitigation Controls

760.4 HANDS ON: Windows Kernel Debugging and Exploitation

The Windows Kernel is very complex and intimidating. This day aims to help you understand the Windows kernel and the various exploit mitigations added into recent versions. You will perform Kernel debugging on various versions of the Windows OS, such as Windows 7 and 8, and learn to deal with its inherent complexities. Exercises will be performed to analyze vulnerabilities, look at exploitation techniques, and get a working exploit.

Topics: Understanding the Windows Kernel; Navigating the Windows Kernel; Modern Kernel Protections; Debugging the Windows Kernel; WinDbg; Analyzing Kernel Vulnerabilities and Kernel Vulnerability Types; Kernel Exploitation Techniques

SEC760 COIN

SEC760 Training Formats

(subject to change)

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

OnSite
sans.org/onsite

vLive Events
sans.org/vlive

Simulcast
sans.org/simulcast

SelStudy
sans.org/selfstudy

760.5 HANDS ON: Windows Heap Overflows and Client-Side Exploitation

The focus of this section is primarily on Windows browser and client-side exploitation. You will learn to analyze C++ vftable overflows, one of the most common mechanisms used to compromise a modern Windows system. Many of these vulnerabilities are discovered in the browser, so browser techniques will also be taught, including modern heap spraying to deal with IE 8/9/10 and other browsers such as FireFox and Chrome. You will work towards writing exploits in the Use-After-Free/Dangling Pointer vulnerability class.

Topics: Windows Heap Management, Constructs, and Environment; Browser-Based and Client-Side Exploitation; Remedial Heap Spraying; Understanding C++ vftable/vtable Behavior; Modern Heap Spraying to Determine Address Predictability; Use-After-Free Attacks and Dangling Pointers; Determining Exploitability; Defeating ASLR, DEP, and Other Common Exploit Mitigation Controls

760.6 HANDS ON: Capture the Flag

Day 6 will serve as a capture the flag day with different types of challenges taken from material taught throughout the week.