SANS The most trusted source for cybersecurity training, certifications, degrees, and research

GIAC CERTIFICATIONS

SEC760: **Advanced Exploit Development for Penetration Testers**

6 Day Program	46 CPEs	Laptop Required

You Will Be Able To

- Discover zero-day vulnerabilities in programs running on fully patched modern operating systems
- Use the advanced features of IDA Pro and write your own 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
- Perform Windows kernel debugging up through Windows 10 64-bit Build 1903
- Perform Windows driver and kernel exploitation

What You Will Receive

- A four-month license to IDA Pro, which is provided by Hex-Rays, is included in this course. In order to obtain the license, you must agree to the terms, including providing your name and an e-mail address, so that Hex-Rays may assign the license. After the course ends, students may choose to extend the license at a discounted rate by contacting Hex-Rays. (If you choose to opt-out, then you must bring a copy of IDA Pro 7.4 advanced or later.)
- Various preconfigured virtual machines, such as Windows 10
- Various tools on a course USB that are required for use in class
- Access to the in-class Virtual Training Lab with many in-depth labs
- Access to recorded course audio to help hammer home important network penetration testing lessons

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 skill set 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 skill set regardless of the increased complexity. SEC760: Advanced Exploit Development for Penetration Testers, the SANS Institute's only 700-level course, teaches the skills required to reverse-engineer 32- 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 7/8/10 operating systems
- How to perform complex attacks such as use-after-free, kernel and driver exploitation, one-day exploitation through patch analysis, and other advanced attacks
- 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

Authors' Statement

"As a perpetual student of information security, I am excited to offer SEC760: Advanced Exploit Writing for Penetration Testers. Exploit development is a hot topic as of late and will continue to increase in importance moving forward. With all of the modern exploit mitigation controls offered by operating systems such as Windows 7 and 8, the number of experts with the skills to produce working exploits is highly limited. More and more companies are looking to hire professionals with the ability to conduct a Secure-SDLC process, perform threat modeling, determine if vulnerabilities are exploitable, and carry out security research. This course was written to help you get into these highly sought-after positions and to teach you cutting-edge tricks to thoroughly evaluate a target, providing you with the skills to improve your exploit development."
—Stephen Sims

"Teaching and helping author SEC760: Advanced Exploit Writing for Penetration Testers has given me the opportunity to distill my past experiences in exploit writing and technical systems knowledge into a format worth sharing. This course is meant to give you a look into a number of different exploitation techniques and serves as an amazing jumping-off point for exploitation of any modern application or system. Even if you don't plan on having a career in exploit writing or vulnerability research, this course will be valuable in understanding the thought process that goes into constructing an exploit and what technologies exist to stop an exploit writer from being successful."
—Jaime Geiger

Section Descriptions

SECTION 1: Exploit Mitigations and Reversing with IDA

The course starts with a deep dive into both mature and modern exploit mitigations. It is rare today to come across an application or operating system that doesn't use a combination of mitigations to thwart the exploitation of a vulnerability. Outdated operating systems and applications do exist, such as in the industrial control system and Internet of Things space, but that is not the focus of this course. We address the effectiveness and technical details behind each control, such as those implemented in Windows Defender Exploit Guard. We then spend the remainder of the section using IDA Pro, which comes bundled with the course. We quickly ramp up on the basics of IDA Pro as a disassembler and then move into remote debugging with the tool. We finish up Section 1 utilizing IDA FLIRT and FLAIR and writing IDAPython scripts to help with bug hunting and analysis.

TOPICS: Exploit Mitigations; Windows Defender Exploit Guard; Introduction to IDA Pro; Debugging with IDA Pro; FLIRT & FLAIR; Scripting with IDAPython and Python 3

SECTION 2: Advanced Linux Exploitation

The ability to progress into more advanced reversing and exploitation requires an expert-level understanding of basic software vulnerabilities, such as those covered in SANS' SEC660 course. 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, which is necessary to continue further with the course. Linux can sometimes be an easier operating system to learn these techniques, serving as a productive gateway into Windows. Most courses on exploit development focus purely on the Windows OS, and it's important to have an understanding of vulnerability research on the Linux OS as well.

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; Using Format String Bugs for ASLR Bypass

SECTION 3: Patch Diffing, One-Day Exploits, and Return-Oriented Shellcode

Attackers often download patches as soon as they are distributed by vendors such as Microsoft in order to find newly patched vulnerabilities. Vulnerabilities are usually 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 aware of this and quickly work to find the patched vulnerability in order to take control of unpatched systems, as many organizations struggle with getting patches out quickly. Binary diffing and patch diffing is also performed by incident handlers, IDS administrators and vendors, vulnerability and penetration testing framework companies, government entities, and others. You will use the material covered on this day to identify bugs patched by Microsoft, taking some of them through to exploitation. We will also focus on using Return Oriented Programming (ROP) to string together gadgets that emulate shellcode.

TOPICS: The Microsoft Patch Management Process and Patch Tuesday; Obtaining Patches and Patch Extraction; Binary Diffing with BinDiff 5; Visualizing Code Changes and Identifying Fixes; Reversing 32-Bit and 64-Bit Applications and Modules; Triggering Patched Vulnerabilities; Writing One-Day Exploits; Using ROP to Compiled Shellcode on the Fly (Return-Oriented Shellcode)

SECTION 4: Windows Kernel Debugging and Exploitation

The Windows kernel is complex and intimidating, so this day aims to help you understand the Windows kernel and the various exploit mitigations added into recent versions. You will learn how the kernel works with drivers to talk to devices and how some functionality can be exposed to user-mode, sometimes insecurely! You will perform kernel debugging on Windows 10 and learn to deal with its inherent complexities. Exercises will be performed to analyze Ring 0 driver vulnerabilities, look at exploitation techniques, and get working exploits.

TOPICS: Understanding the Windows Kernel; Navigating the Windows Kernel; Modern Kernel Protections; Debugging the Windows 10 Kernels and Drivers; WinDbg; Analyzing Kernel Vulnerabilities and Vulnerability Types; Kernel Exploitation Techniques; Token Stealing and Information Disclosure Vulnerabilities

SECTION 6: Capture-the-Flag Challenge

Section 6 will feature a Capture-the-Flag event employing different types of challenges from material taught throughout the week. Test your reverse-engineering, bug discovery, and exploit-writing skills in a full day of Capture-the-Flag exercises!

SECTION 5: Advanced Windows Exploitation

The focus of this day is on the advanced exploitation of applications running on the Windows OS. For many years now memory corruption bugs have been the de facto standard regarding exploiting Windows applications. Examples include Use After Free (UAF) and Type Confusion bugs. Many of these vulnerabilities exist due to complexities with large C++ applications such as object tracking and dynamic memory management. In this section we focus on these types of application vulnerabilities on the Windows 7, 8, and 10 operating systems.

TOPICS: Windows Heap Management, Constructs, and Environment; Understanding the Low Fragmentation Heap; Browser-Based and Client-Side Exploitation; Remedial Heap Spraying; Understanding C++ vftable/vtable Behavior; Use-After-Free Attacks and Dangling Pointers; Avoiding Protections such as MemGC and Isolated Heap; Dealing with ASLR, DEP, and Other Common Exploit Mitigation Controls

Who Should Attend

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

> **"SEC760 is a kind of training we could not get anywhere else. It is not a theory, we got to implement and to exploit everything we learned."**
>
> — Jenny Kitaichit, **Intel**

> **"SEC760 is the challenge I am looking for. It will be overwhelming, but well worth it."**
>
> — William Stott, **Raytheon**