SCED

Certified Exploit Developer

Synapse Certified Exploit Developer

Course objective:

The Aim of the Synapse's Certified Exploit Developer Course is to teach their Students the art of developing hardcore exploits in a fully hands on training.

Course Requirements:

- A Laptop running windows XP service pack 3 (or on VM)
- A Laptop running windows Vista/Seven (or on VM)

Suitable Requirements:

- Basic understanding of assembler language
- Basic programming skills

Course Duration:

Course Name	Duration	Instructors	Delivery Method	Delivery Date
SCED Basic	One weeks (05 Days)	1/2	Onsite	
SCED Advanced	One weeks (05 Days)	1/2	Onsite	

Contacts:

- Email: info@synapse-labs.com
- website: http://www.synapse-labs.com

Course Schedule:

SCED 1 (Basic)

- Day 1: 9am-4pm
- Day 2: 9am-4pm
- Day 3: 9am-4pm
- Day 4: 9am-4pm
- Day 5: 9am-4pm

SCED 2 (Advanced)

- Day 1: 9am-4pm
- Day 2: 9am-4pm
- Day 3: 9am-4pm
- Day 4: 9am-4pm
- Day 5: 9am-4pm

SCED 1 (Basic)

Day 1

Chapter 1 : **Assembly primer (3 hour)**

- A crash course in assembly programming

Chapter 2 : **Inside Windows memory and PE structure (3 hours)**

- Windows memory layout
- Stack/Memory layout and mechanism
- The PE structure

Day 2

Chapter 3: **Inside windows debuggers (1 hours)**

- Introduction to windows debuggers
- Advanced debugging overview

Chapter 4: **Python programming (2 hours)**

- A crash course into python programming
- Python API for exploitation

Chapter 5: **Exploiting debuggers for fun (3 hour)**

- Debuggers API for scripting
- Python scripting for API
- Building you first automation scripts

Day 3

Chapter 6: **Introduction to vulnerabilities (1 hours)**

- what are vulnerabilities
- Origin of vulnerabilities

Chapter 7: **Finding vulnerabilities (5 hours)**

- Where to search
- Fuzzing
- Reverse engineering for vulnerabilities finding
- Debugger plug-ins, Automated scripts and tools
- Exploitation concepts

Day 4

Chapter 8: **Shellcodes (3 hours)**

- Introduction to shellcodes
- Shellcode vs Executables
- Writing your first shellcode
- Writing basic shellcodes
- API calling and complex shellcodes

Chapter 9: **Shellcode encoding (3 hours)**

- Shellcodes Encoding
- Using Encoders
- Memory limits
- Analyzing Encoders
- writing custom encoders

Day 5

Chapter 10: **First exploit (3 hour)**

- Stack based buffer overflow
- stack layout
- How stack overflow occurs
- Basic Stack Overflow Example
- Building First Exploit : Controlling EIP
- Building second exploit : Controlling registers

Chapter 11: **Lab and challenges (3 hour)**

SCED 2 (Advanced)

Day 1

Chapter 1 : **Stack protections (6 hours)**

- Memory Stack protections
- SEH Protection
- Inside Windows Exception Dispatching
- SEH Attacks
- GS cookie Protection and Attacks
- SafeSEH protection
- Bypassing SafeSEH protections
- SEHOP protections
- Bypassing SEHOP protections

Day 2

Chapter 2: **Fun with DEP & ASLR (3 hours)**

- DEP protections
- Bypassing DEP protections
- ASLR protection
- Bypassing ASLR protections

Chapter 3: **Python for debuggers (3 hours)**

- Automating search
- Building first exploitation tool

Day 3

Chapter 4: **Memory limitations & special shellcodes (6 hours)**

- What is Egghunter
- Dissecting different Egghunter Shellcodes
- Writing your first Egghunter
- Writing special egghunters
- Making the omelette
- Universal shellcodes

Day 4

Chapter 5: **ActiveX Browser based attacks (6 hours)**

- What is an ActiveX
- ActiveX properties
- Loading ActveX
- Heap Sprying techniques
- Heap Sprying protections
- Heap sprying through IE versions
- Heap sprying for Firefox
- ActiveX Lab

Day 5

Chapter 10: **Heap Overflow exploitation (6 hour)**

- What is the HEAP
- Inside Windows HEAP layout & structure
- Detecting Heap Overflows
- Heap Overflow exploitation techniques
- Heap Lab