Offensive Development Course

Overview

Dive deep into cutting edge techniques that bypass or neuter modern endpoint defenses. Learn how these solutions work to mitigate their utility and hide deep within code on the endpoint. The days of downloading that binary from the internet and pointing it at a remote machine are over. Today's defenses oftentimes call for multiple bypasses within a single piece of code.

This course is designed to take you deep into defensive and offensive tooling – an apex attacker must know the own indicators of compromise (IOCs) they're creating and the artifacts they're leaving behind.

Who Should Attend?

Anybody that is deeply passionate about red teaming and has a stron

Key Learning Objectives

Learn the IOCs and artifacts of using off-the-shelf tooling. Without understanding the defender's capabilities, an attacker brings little value to a red team engagement.

Prerequisite Knowledge

This is an intermediate level course – a background in C programming, Windows Internals, .NET programming, and how AV/EDR products work would be useful.

Lab Environment

Students will have access to their own contained lab environment within Snap Labs that consists of the following:

- Windows Server 2019 running Sophos Intercept X EDR
- Ubuntu Cobalt Strike Team Server
- Windows 10 Development Machine
- Kali Linux
- Admin Machine running Apache Guacamole
- Fully Patched Windows 10 Machine

Hardware/Software Requirement

- Ability to connect to the SnapLabs cyber range (must create an account)

Syllabus

Day 1 – Understanding Modern Defenses

- Hiding from the Import Address Table (IAT)
- Dynamically Building Your Strings
- Defeating string detection via encryption
- Finding EDR's DLL
- Unhooking EDR products
- .NET and Assembly.Load
- Obfuscating .NET assemblies and their IOCs
- AMSI bypass
- ETW bypass

Day 2 – Process Injection and Cobalt Strike

- Process Injection Variants
- Malleable C2 Profiles
- Beacon Object Files
- Cobalt Strike IOCs
- Attacking AV/EDR Products
- Dumping LSASS in 2022
- Making the final binary to bypass multiple EDR products