Course Title:

Initial Access Operations

INSTRUCTORS:

Principal Security Consultant Mike Saunders

Senior Security Consultant Corey Overstreet

Description: This course provides an in-depth exploration of the techniques used by attackers and red teams to gain initial entry into targeted environments. It covers key areas such as credential harvesting, the strategic use of office documents, browser-based attacks, and methods to safeguard malicious code to ensure it runs only on intended systems. By the end of this program, participants will not only grasp various attack strategies but also gain hands-on experience in crafting their own malware.

COURSE OUTLINE	
Introduction	The introduction will highlight the topics of the course, the course agenda, class requirements, class logistics, and how the next two days will work.

Development Environment and Goals	The development environment section will walk you through how to set up the systems you are going to be coding in to ensure you have the flexibility to write the code needed for your specific phishing scenarios. You'll learn about the different tools you can choose from (both open-source and commercial) that will help your development process and about properly setting up infrastructure for your actual phishing campaign.
Credential Harvesting	It's time to dive right into credential harvesting attacks! You'll learn how to configure your infrastructure, choose domain names, and pick service providers based on the target's web application you are trying to obtain credentials for. You're going to learn about tools that can aid you in cloning websites along with various techniques you can utilize to weaponize the cloned application.
Pretexts	When building out a successful social engineering campaign, a good pretext is key to help ensure the campaign's success. In this section, attendees will look at various examples of social engineering pretexts that have been used by real-world attackers. Everyone will build out a pretext campaign that could be used in a real-world test, and we'll end this section by

	discussing various tips and tricks that have worked for us on our assessments.
Process Injection	The process injection section is a very in-depth discussion on how process injection works when interacting with the Windows API. We'll start with discussing the various API calls that are used for injecting shellcode within your current process (such as allocating memory, creating a thread, etc.) along with how to define those API calls within your code. Once your code is finished, it's time to compile it and test it out! Once the foundation for injecting and executing shellcode is made, we'll introduce the many variations that exist for accomplishing the same tasks, along with some tricks that have been helpful for getting around endpoint protection software. Finally, everything that we've been discussing in this section has been for executing code within your current process (the process of your loader), but what if you want to inject shellcode into a remote process? This section will end with

	updating the appropriate API calls to allow remote process injection, along with also introducing alternative ways of doing so
Code Execution: HTA and CLICKONCE	The code execution section is a step in a different direction; rather than using credential harvesting web sites, or weaponized Word documents, you're going to start building browser-based attacks which allow you to compromise the underlying system via weaponized URLs. Part 1 will cover the use of HTAs (HTML Applications) and Click Once Applications, along with the required web resources to use them. You will also learn about stagers, what they accomplish, and understand the underlying code which will allow you to get your agent of choice up and running within your victim's PC. This is going to signal the beginning of writing .NET code and interacting with Windows functionality.
Code Execution: VBA	Malicious Word documents are a tried-and-true method that still produces great results. We'll discuss basic macro development and walk you through tools which can help you produce weaponized documents. Additionally, we're going to cover methods that allow you to remotely load weaponized documents to avoid ever

	sending a highly suspicious ".docm" file extension.
Code Protections	Finally, we're going to discuss methods to protect your code, also referred to as application guard rails. Why spend all that time writing your malware just to have someone open it on their home computer, on their phone, or in a sandbox? You'll learn about various checks you can build into your code to protect and prevent it from running outside of your target environment.