Course Title: Initial Access Operations

Description: Most red team classes cover a wide range of topics such as reconnaissance, initial access, post-exploitation, and more. However, the volume of material covered within each step often prohibits students from conducting a deep dive on any individual topic. We're changing that narrative with a course fully dedicated to Initial Access Operations.

Initial Access Operations is designed to immerse you in multiple techniques that attackers (and red teams) use to gain initial access into the environment they are targeting.

We'll look at credential harvesting techniques attackers commonly use when trying to entice victims to authenticate into a malicious (web) application. We'll also review the best ways to weaponize office documents, which are still widely employed by attackers and red teams because of the high success rates. Additionally, we'll learn about browser-based attacks, which can provide unique opportunities for attackers to remain largely in memory. Finally, we'll discuss different ways to protect malicious code by only allowing it to run on the exact system(s) you are targeting.

At the end of this course, you'll understand several different methods attackers use to compromise targets as well as have built your own malware.

There are no prerequisites for this course; however, we recommend students have an intermediate level of programming knowledge. The course is very hands-on and students will be authoring their own malware (but don't worry we'll provide you with plenty of templates to get started).

**Our content is updated on an ongoing basis. Not only do we provide students with the fundamental knowledge to create their own initial access malware, but we share the newest tactics that our red team uses on actual assessments.

Outline:

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 in to 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. Finally, you'll build out capabilities to alert you each time you've captured new user credentials.
Weaponized Word Documents	Malicious Word documents are a tried and true method that still produce 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 Execution (Part 1)	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 (Part 2)	Now that you have an understanding of how stagers work, and the API calls that you used to “stage” your malware, we’re going to look at new ways to accomplish the same tasks! We’re going to cover DotNetToJScript and dive into how it has completely changed phishing malware design. You’ll expand beyond this and start a deep dive into multiple routines to inject shellcode into your victim’s system beyond the standard CreateRemoteThread injection routines. You will be busy writing your own proof of concepts that utilize the covered techniques.
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.