Exploitation

Course Introduction

Alexis Ahmed

Senior Penetration Tester @HackerSploit
Offensive Security Instructor @INE

aahmed@ine.com
@HackerSploit
@alexisahmed

Course Topic Overview

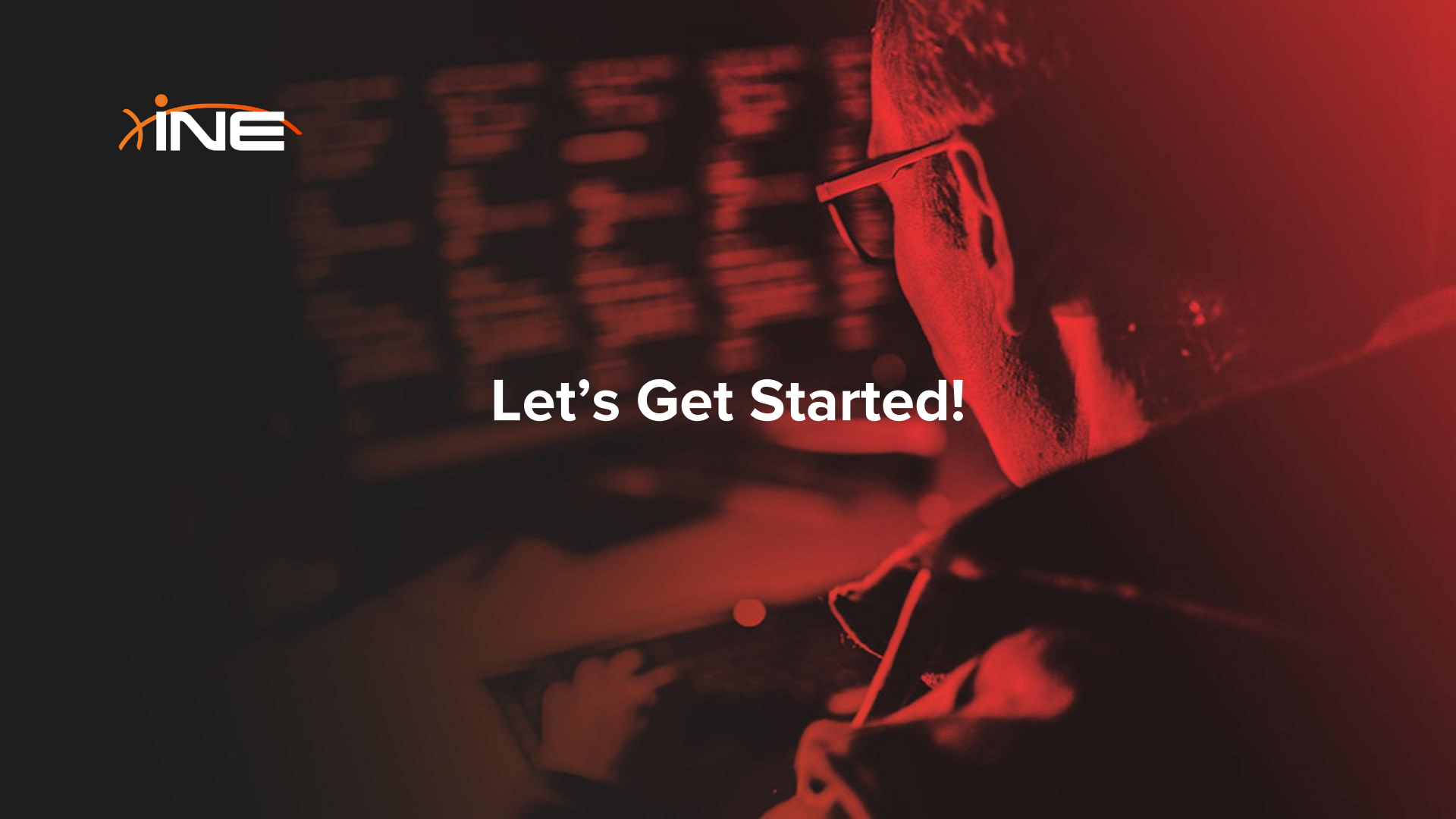
- Introduction To Exploitation
- Vulnerability Scanning
- Searching For Exploits
- Fixing Exploits
- Bind & Reverse Shells
- Exploitation Frameworks
- Windows Exploitation
- Linux Exploitation
- AV Evasion & Obfuscation

Prerequisites

- Basic familiarity with TCP & UDP
- Basic familiarity with Linux & Windows
- Basic familiarity with Metasploit

Learning Objectives:

- Students will get an introduction to the exploitation phase of a penetration test.
- Students will learn how to identify vulnerable services running on a target system.
- Students will learn how to search for, modify and compile publicly available exploit code.
- Students will get an understanding of how bind and reverse shells work .
- Students will get an understanding of the various exploitation frameworks available as well as how they can be used to streamline exploitation.
- Students will learn how to exploit both Windows & Linux systems in a simulated black box penetration test.
- Students will learn how to evade signature based AV solutions.


Let’s Get Started!

Introduction To Exploitation

Exploitation

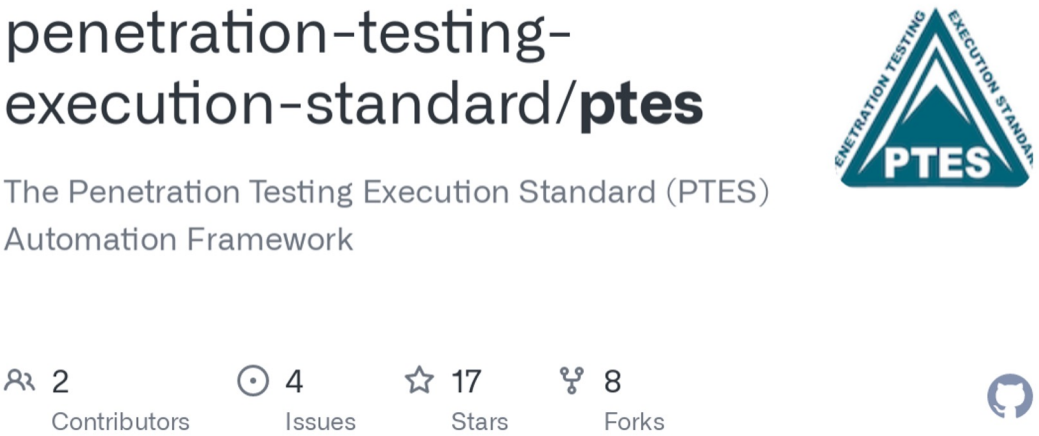
- Exploitation consists of techniques and tools used by adversaries/penetration testers to gain an initial foothold on a target system or network.

- Successful exploitation will heavily depend on the nature and quality of information gathering and service enumeration performed on the target.
  - We can only exploit a target if we know what is vulnerable - *Unknown*

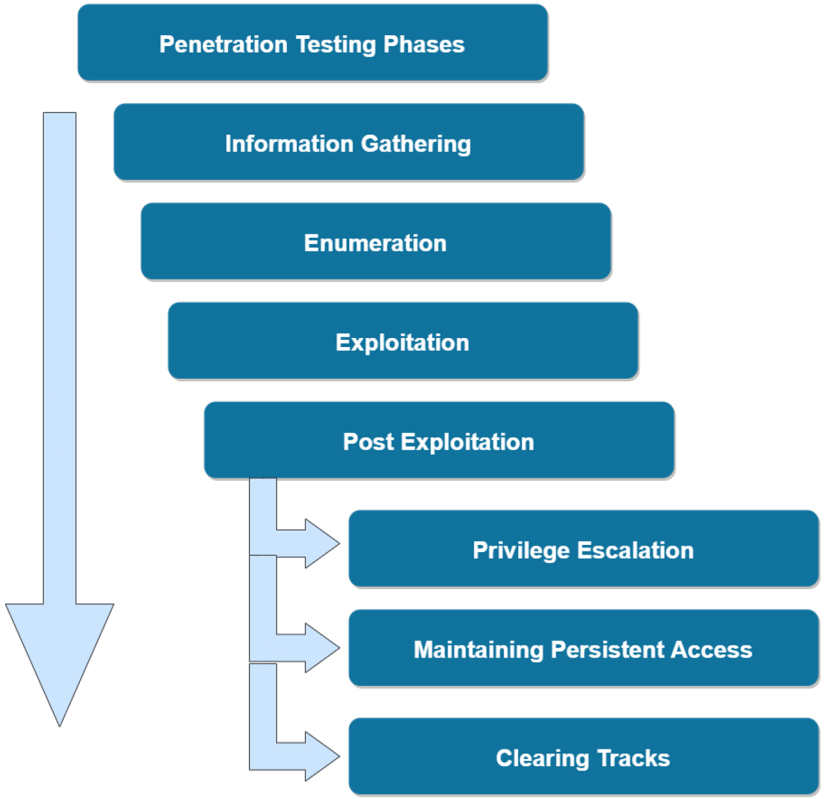
- So far, we have covered exploitation of Windows & Linux systems both manually and automatically, however, we still need to get a clearer picture of the exploitation methodology and the tools and techniques involved in the process.

Penetration Testing Execution Standard

The Penetration Testing Execution Standard (PTES) is a penetration testing methodology that was developed by a team of information security practitioners with the aim of addressing the need for a comprehensive and up-to-date standard for penetration testing.



Penetration Testing Phases



The following diagram outlines the various phases involved in a typical penetration test.

Exploitation Methodology:

- Identify Vulnerable Services
- Identify & Prepare Exploit Code
- Gaining Access
  - Automated - MSF
  - Manual
- Obtain remote access on target system.
- Bypass AV detection
- Pivot on to other systems

Banner Grabbing

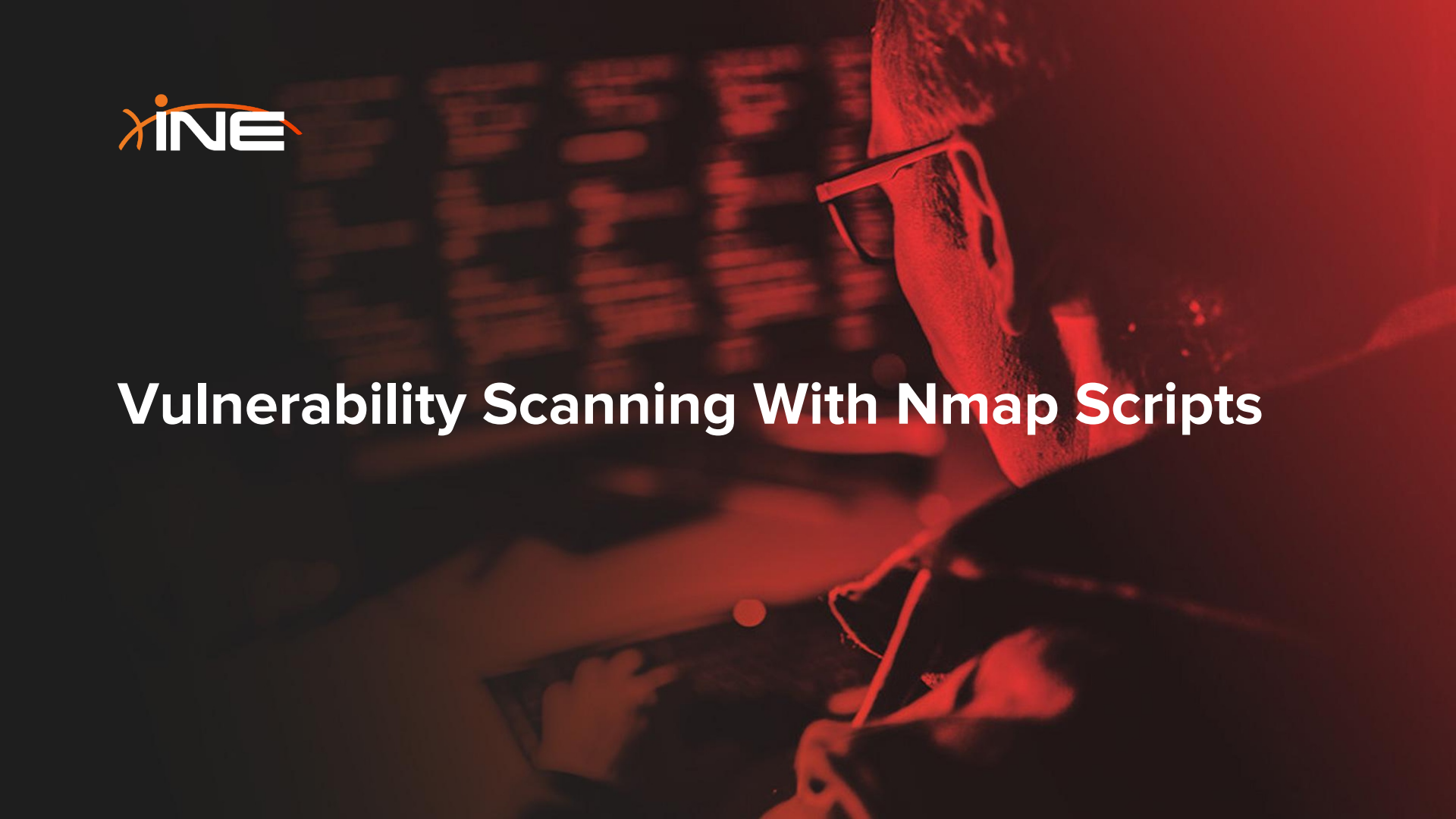
Banner Grabbing

- Banner grabbing is an information gathering technique used by penetration testers to enumerate information regarding the target operating system as well as the services that are running on its open ports.

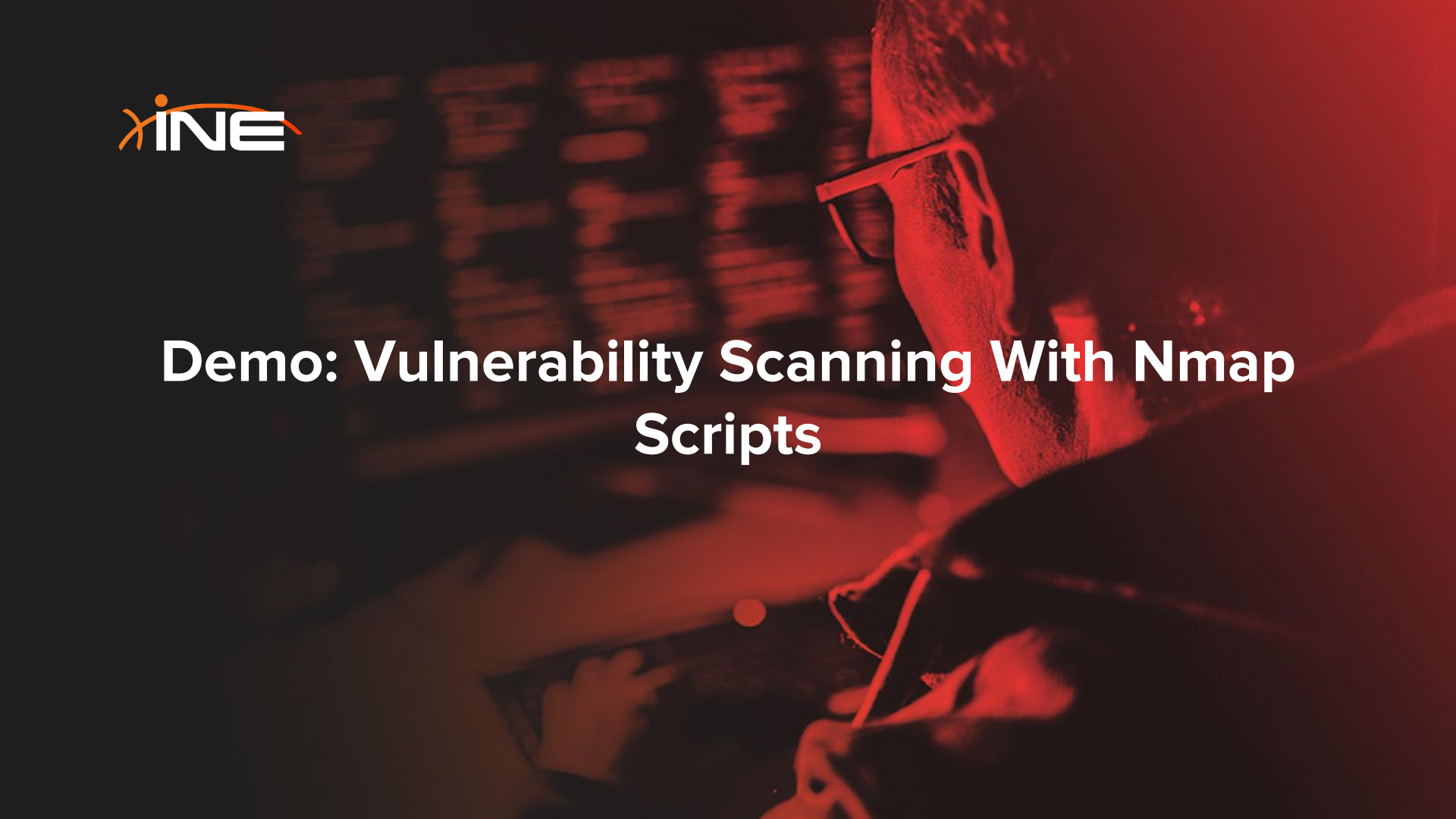
- The primary objective of banner grabbing is to identify the service running on a specific port as well as the service version.

- Banner grabbing can be performed through various techniques:
  - Performing a service version detection scan with Nmap.
  - Connecting to the open port with Netcat.
  - Authenticating with the service (If the service supports authentication), for example; SSH, FTP, Telnet etc.


Demo: Banner Grabbing



Vulnerability Scanning With Nmap Scripts



Demo: Vulnerability Scanning With Nmap Scripts

Vulnerability Scanning With Metasploit

Demo: Vulnerability Scanning With Metasploit

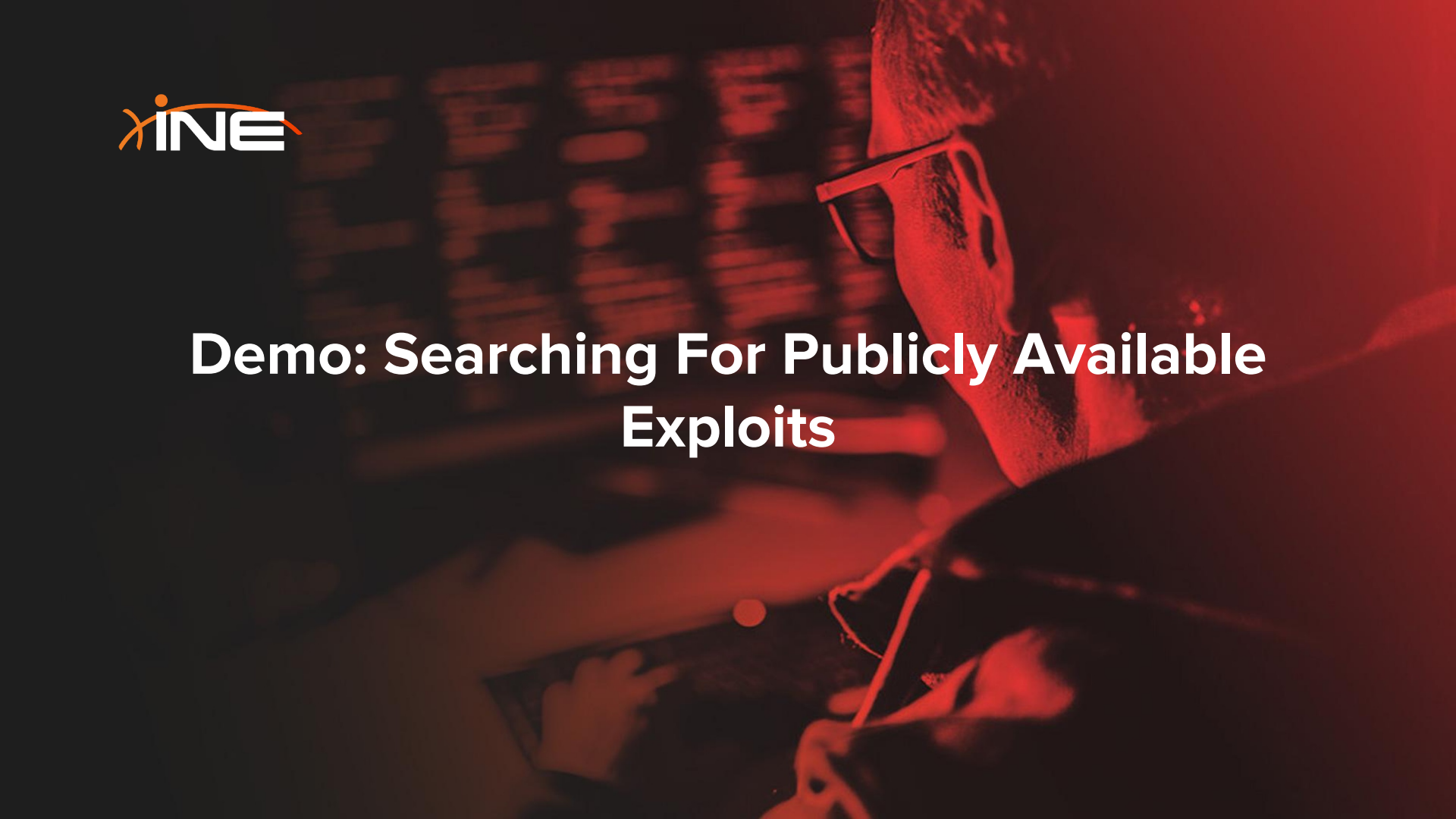
Searching For Publicly Available Exploits

Searching For Public Exploits

- After identifying a potential vulnerability within a target or a service running on a target, the next logical step will involve searching for exploit code that can be used to exploit the vulnerability.

- Exploit code can easily be found online, however, it is important to note that downloading and running exploit code against a target can be quite dangerous. It is therefore recommended to analyze the exploit code closely to ensure that it works as intended.

- There are a handful of legitimate and vetted exploit databases that you should use when searching for exploits online:
  - Exploit-db
  - Rapid7



Demo: Searching For Publicly Available Exploits

Searching For Exploits With SearchSploit

SearchSploit

- In certain cases, you may not have access to online exploits and as a result, you must be able to use the exploit sources available locally/offline.
- The entire Exploit-db database of exploits comes pre-packaged with Kali Linux, consequently providing you with all exploits locally.
- The Exploit-db offline database of exploits can be accessed and queried with a tool called SearchSploit.

Demo: Searching For Exploits With SearchSploit

Fixing Exploits

Demo: Fixing Exploits

Cross-Compiling Exploits

Cross-Compiling Exploits

- In certain cases, exploit code will be developed in C/C++/C#, as a result, you will need to compile the exploit code in to a PE (Portable Executable) or binary.
- Cross-Compiling is the process of compiling code for a platform other than the one performing the compilation.
- As a penetration tester, you will need to have the skills necessary to compile exploit code developed in C.

Demo: Cross-Compiling Exploits

Netcat Fundamentals

Netcat

- Netcat (Aka TCP/IP Swiss Army Knife) is a networking utility used to read and write data to network connections using TCP or UDP.
- Netcat is available for both *NIX and Windows operating systems, consequently making it extremely useful for cross-platform engagements.
- Netcat utilizes a client-server communication architecture with two modes:
  - Client mode - Netcat can be used in client mode to connect to any TCP/UDP port as well as a Netcat listener (server).
  - Server mode - Netcat can be used to listen for connections from clients on a specific port.
- Netcat can be used by penetration testers to perform the following functionality:
  - Banner Grabbing
  - Port Scanning
  - Transferring Files
  - Bind/Reverse Shells

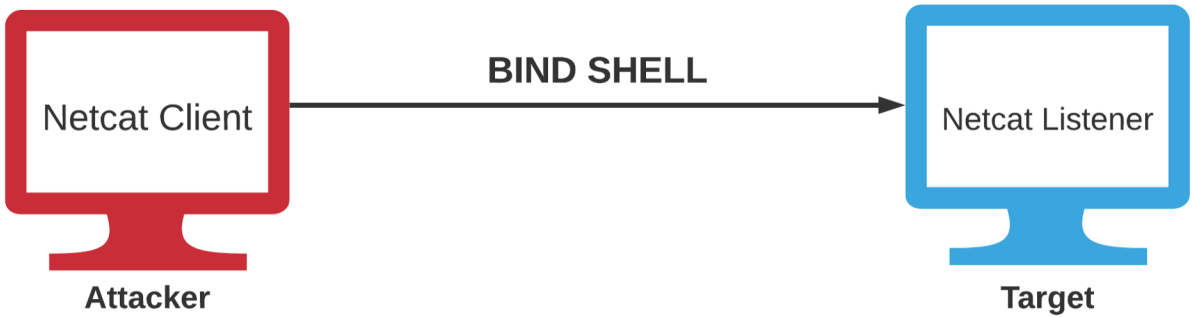
Demo: Netcat Fundamentals

Bind Shells

Bind Shells

- A bind shell is a type of remote shell where the attacker connects directly to a listener on the target system, consequently allowing for execution of commands on the target system.

- A Netcat listener can be setup to execute a specific executable like cmd.exe or /bin/bash when a client connects to the listener.



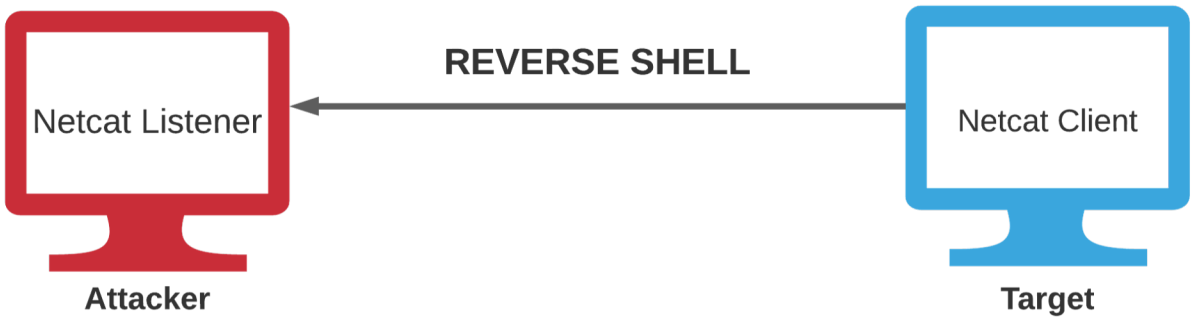
Attacker connects to Netcat listener on target

Demo: Bind Shells

Reverse Shells

Reverse Shells

- A reverse shell is a type of remote shell where the target connects directly to a listener on the attacker's system, consequently allowing for execution of commands on the target system.



Target connects to Netcat listener on Attacker system

Demo: Reverse Shells

Reverse Shell Cheatsheet

Demo: Reverse Shell Cheatsheet

The Metasploit Framework (MSF)

The Metasploit Framework (MSF)

- The Metasploit Framework (MSF) is an open-source, robust penetration testing and exploitation framework that is used by penetration testers and security researchers worldwide.
- It provides penetration testers with a robust infrastructure required to automate every stage of the penetration testing life cycle.
- It is also used to develop and test exploits and has one of the world's largest database of public, tested exploits.
- The Metasploit Framework is designed to be modular, allowing for new functionality to be implemented with ease.

Essential Terminology

- Interface – Methods of interacting with the Metasploit Framework.
- Module – Pieces of code that perform a particular task, an example of a module is an exploit.
- Vulnerability – Weakness or flaw in a computer system or network that can be exploited.
- Exploit – Piece of code/module that is used to take advantage a vulnerability within a system, service or application.
- Payload – Piece of code delivered to the target system by an exploit with the objective of executing arbitrary commands or providing remote access to an attacker.
- Listener – A utility that listens for an incoming connection from a target.

Metasploit Framework Console

- The Metasploit Framework Console (MSFconsole) is an easy-to-use all in one interface that provides you with access to all the functionality of the Metasploit Framework.

Penetration Testing With MSF

- The MSF can be used to perform and automate various tasks that fall under the penetration testing life cycle.
- In order to understand how we can leverage the MSF for penetration testing, we need to explore the various phases of a penetration test and their respective techniques and objectives.
- We can adopt the PTES (Penetration Testing Execution Standard) as a roadmap to understanding the various phases that make up a penetration test and how Metasploit can be integrated in to each phase.

Penetration Testing With MSF

Penetration Testing Phase	Metasploit Framework Implementation
Information Gathering & Enumeration	Auxiliary Modules
Vulnerability Scanning	Auxiliary Modules
Exploitation	Exploit Modules & Payloads
Post Exploitation	Meterpreter
Privilege Escalation	Post Exploitation Modules Meterpreter
Maintaining Persistent Access	Post Exploitation Modules Persistence Modules

Demo: The Metasploit Framework (MSF)

PowerShell-Empire

PowerShell-Empire

- PowerShell-Empire (Aka Empire) is a pure PowerShell exploitation/post-exploitation framework built on cryptological-secure communications and flexible architecture.

- Empire implements the ability to run PowerShell agents without needing powershell.exe, rapidly deployable post-exploitation modules ranging from keyloggers to Mimikatz, and adaptable communications to evade network detection, all wrapped up in a usability-focused framework.

- PowerShell Empire recently received an update and is now officially support and maintained by Kali Linux, more information regarding the update can be found here: https://www.kali.org/blog/empire-starkiller/

Starkiller

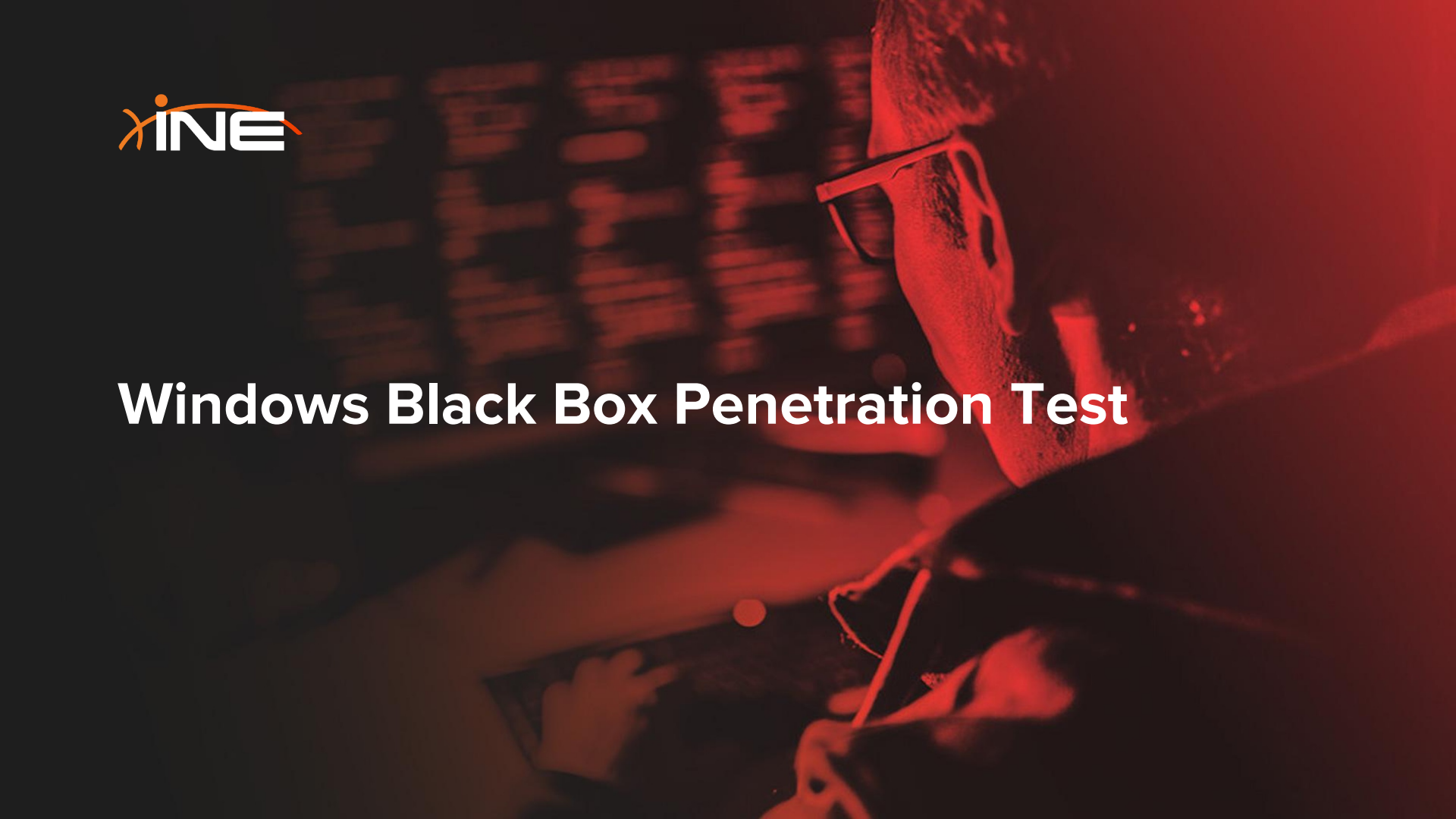
- In addition to being updated and modernized, BC Security, the company responsible for maintaining the Empire has also developed a companion to Empire called Starkiller.

- Starkiller is a GUI Frontend for the Powershell Empire. It is an Electron application written in VueJS and provides users with an intuitive way of interacting with Empire.

- In order to get an understanding of how Empire works and the components that make up the framework, I would recommend going through the official documentation which can be found here: https://www.powershellempire.com/

- PowerShell-Empire & Starkiller are both available as packages in the Kali Linux repositories.

Demo: PowerShell-Empire



Windows Black Box Penetration Test

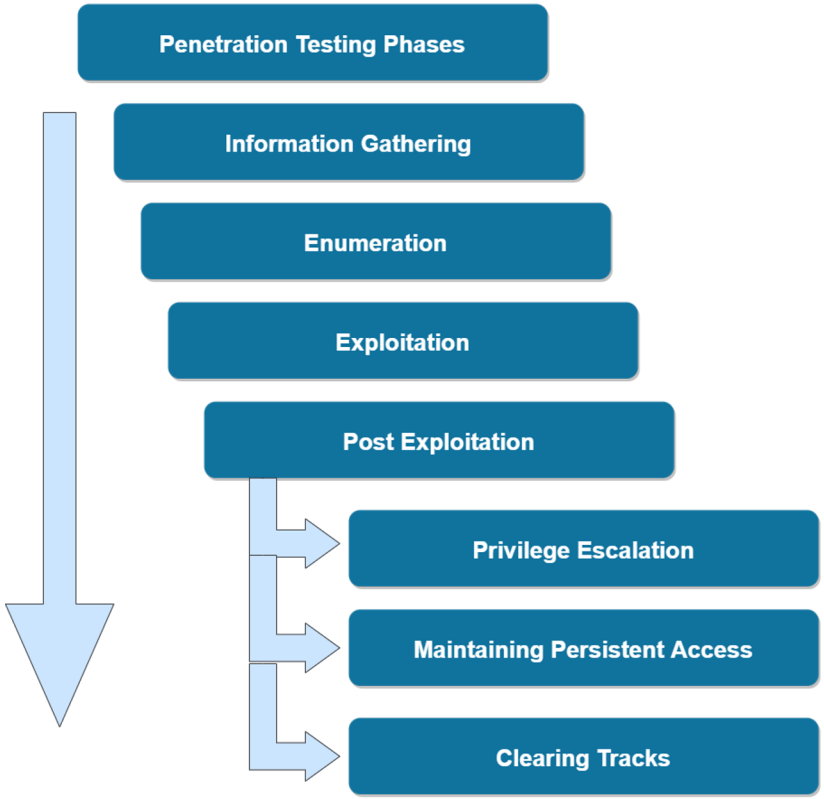
Black Box Pentest

- A Black box penetration test is a security assessment whereby the penetration tester is not provided with any information regarding the target system or network (No IP ranges, system information or default credentials are provided).

- The objective of a Black box penetration test is to accurately test the security of a system or network as an external unprivileged adversary.

- This approach is very useful as it demonstrates how an external attacker with no inside knowledge would compromise a company's systems or networks.

Penetration Testing Phases



The following diagram outlines the various phases involved in a typical penetration test.

Black Box Methodology:

- Host discovery
- Port scanning & enumeration
- Vulnerability detection/scanning
- Exploitation
  - Manual
  - Automated
- Post Exploitation
  - Privilege Escalation
  - Persistence
  - Dumping Hashes

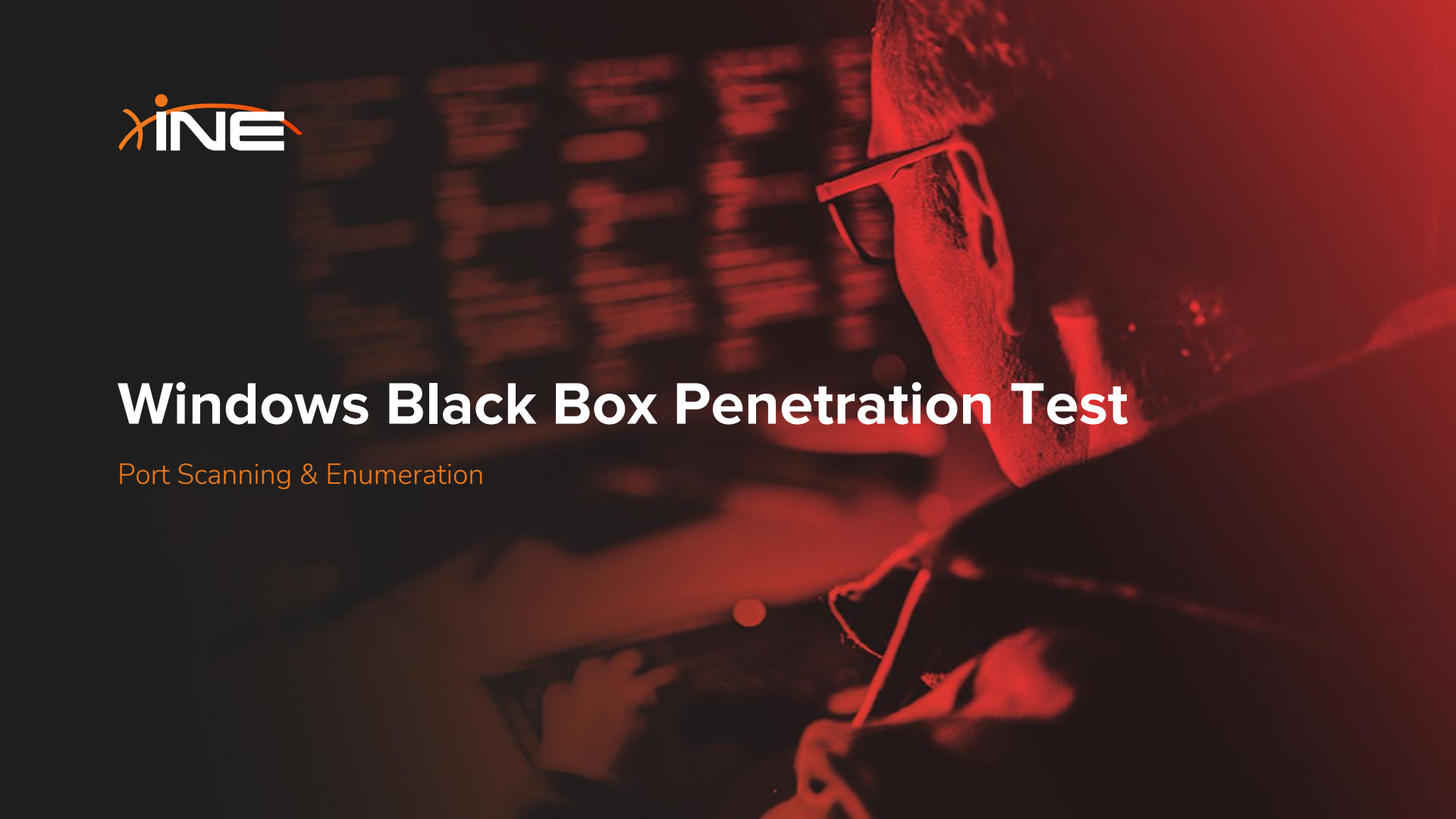
Scenario & Scope

- You have just begun your first job as a Junior Penetration Tester and have been assigned to assist in performing a penetration test on a client's network.

- The pentest lead has assigned you to gain access/exploit a host running Windows Server 2008.

- Your primary objectives are:
  - Identify services running on the target
  - Identify vulnerabilities within the services
  - Exploit these vulnerabilities to obtain an initial foothold

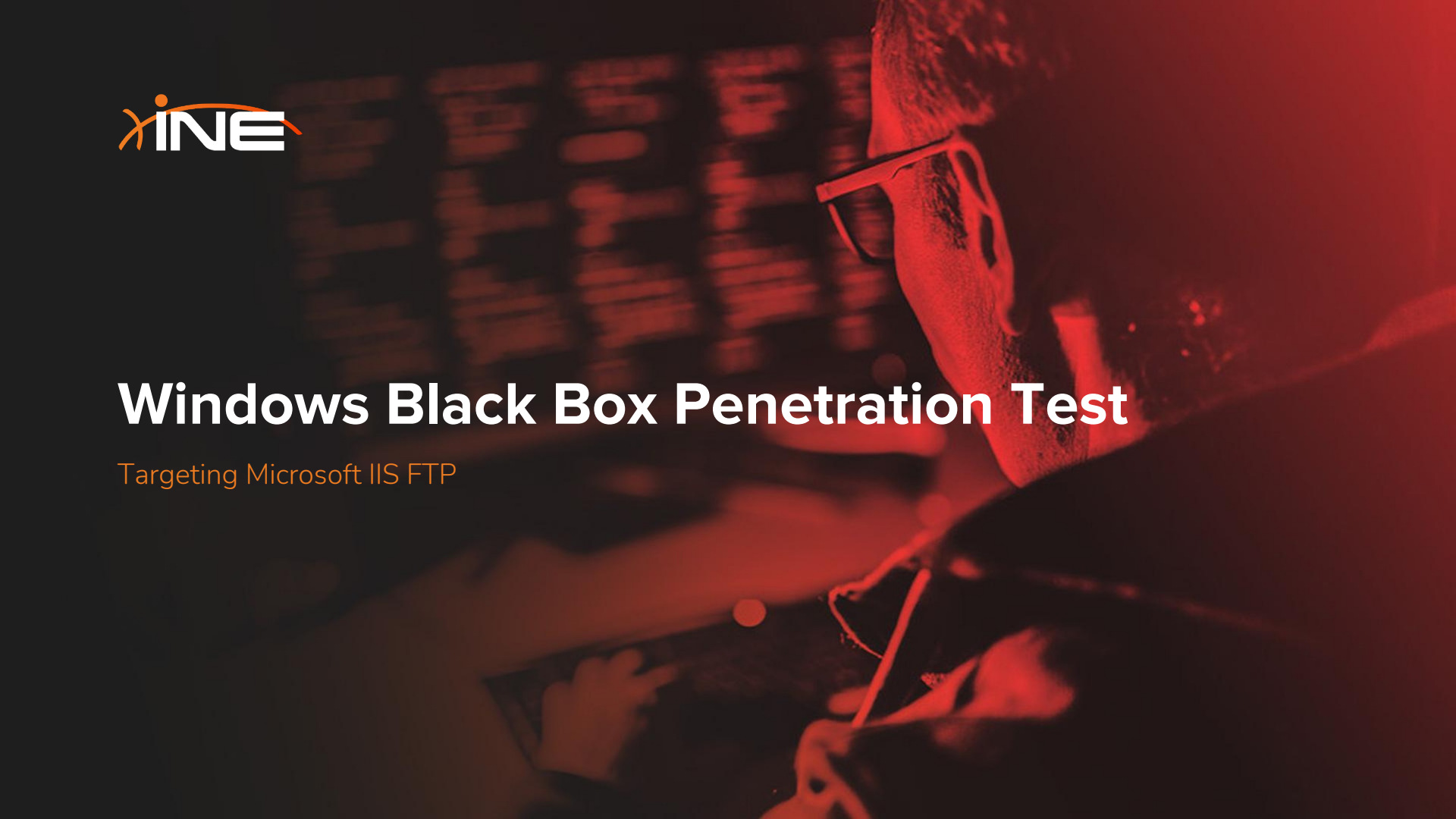
Note: You are permitted to use the Metasploit Framework



Windows Black Box Penetration Test

Port Scanning & Enumeration

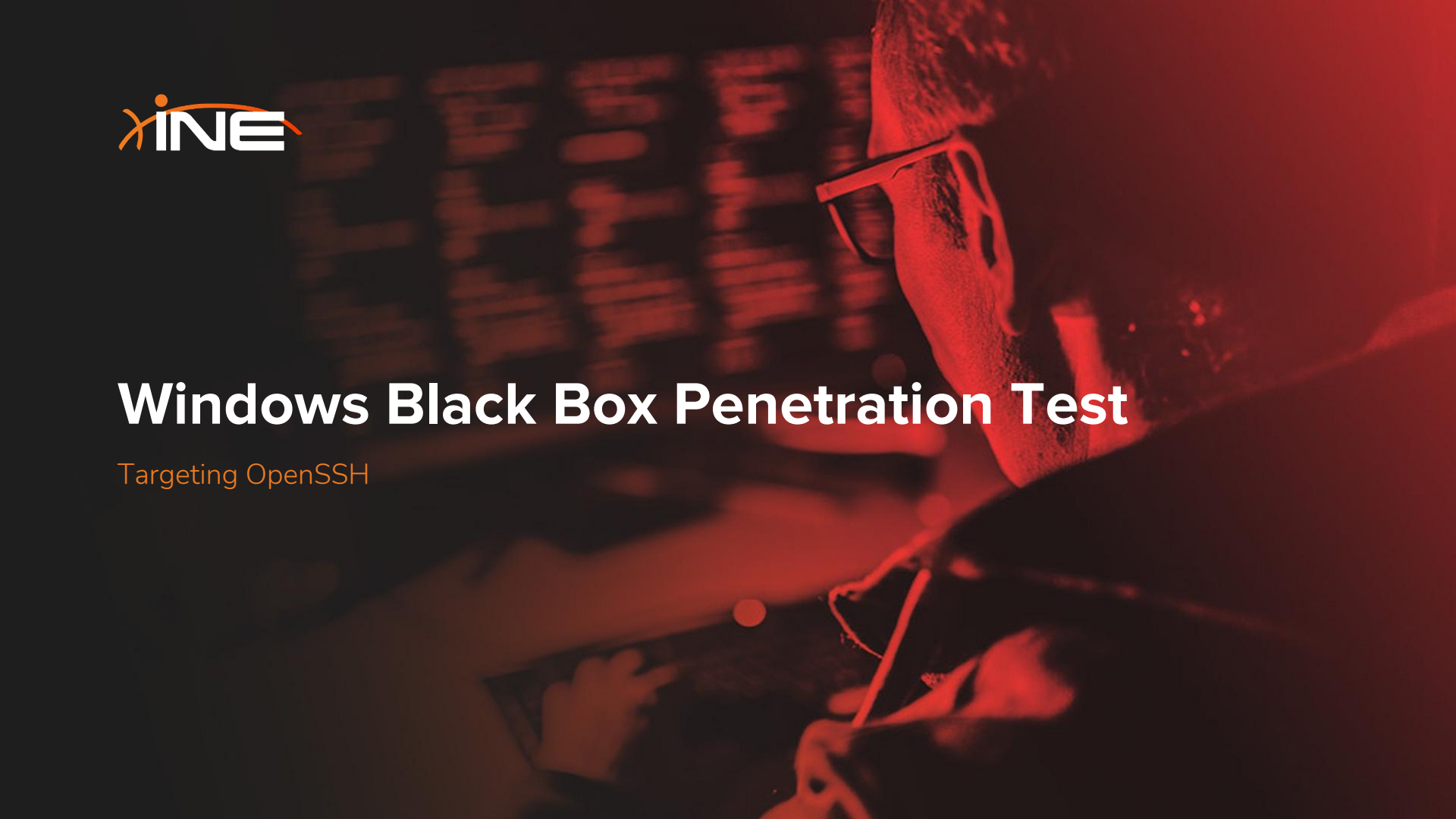
Demo: Port Scanning & Enumeration



Windows Black Box Penetration Test

Targeting Microsoft IIS FTP

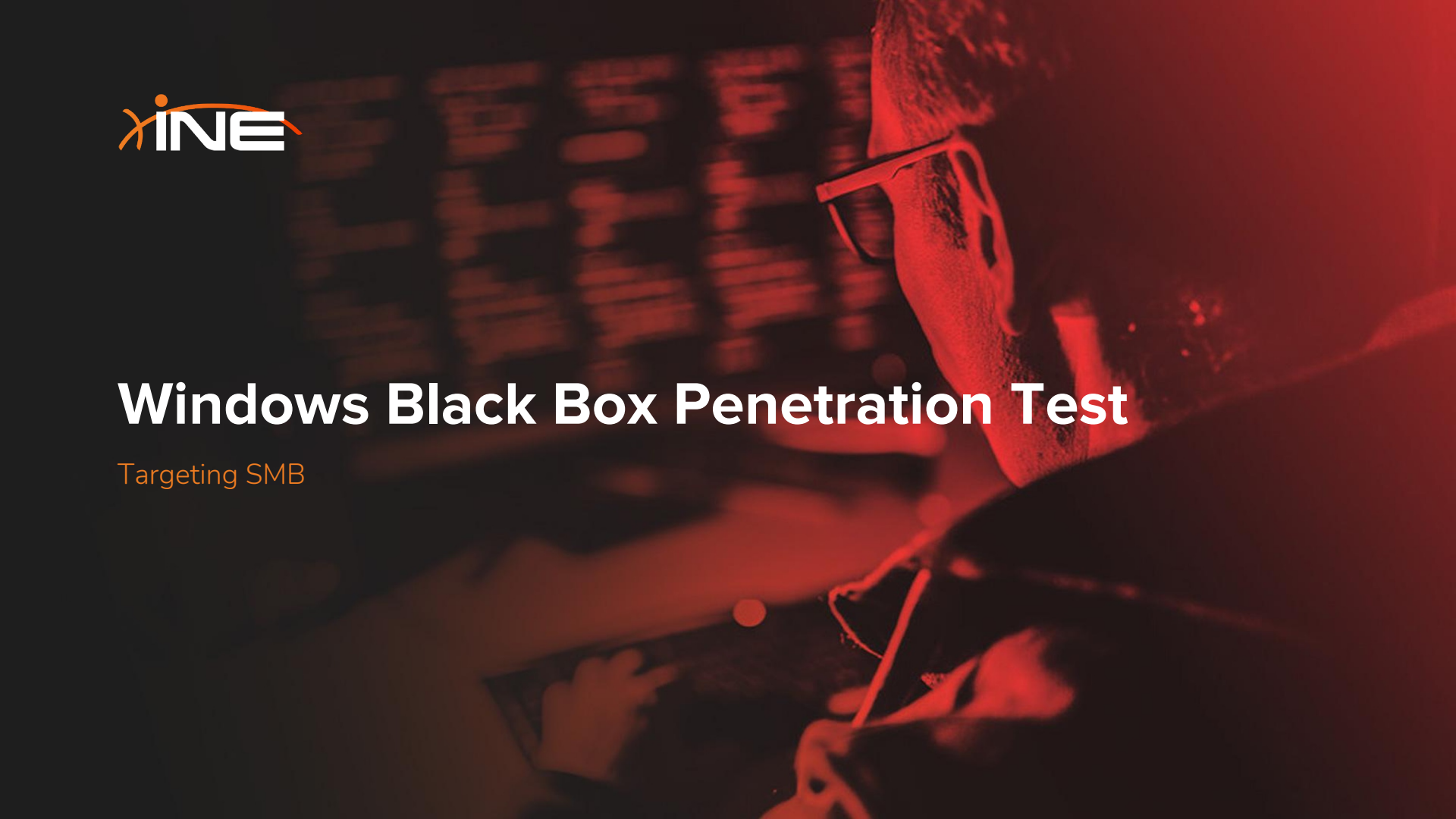
Demo: Targeting Microsoft IIS FTP



Windows Black Box Penetration Test

Targeting OpenSSH

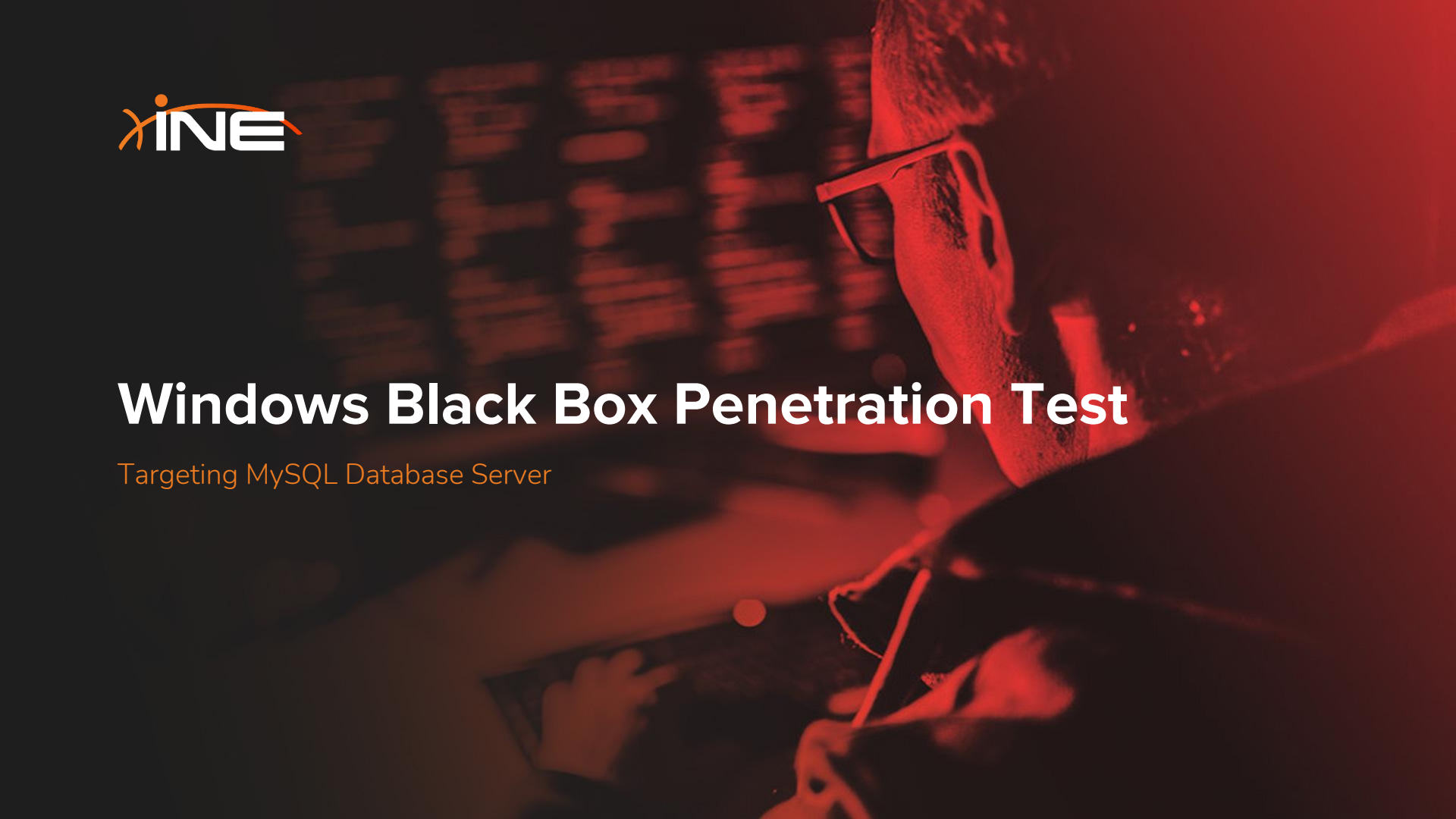

Demo: Targeting OpenSSH



Windows Black Box Penetration Test

Targeting SMB

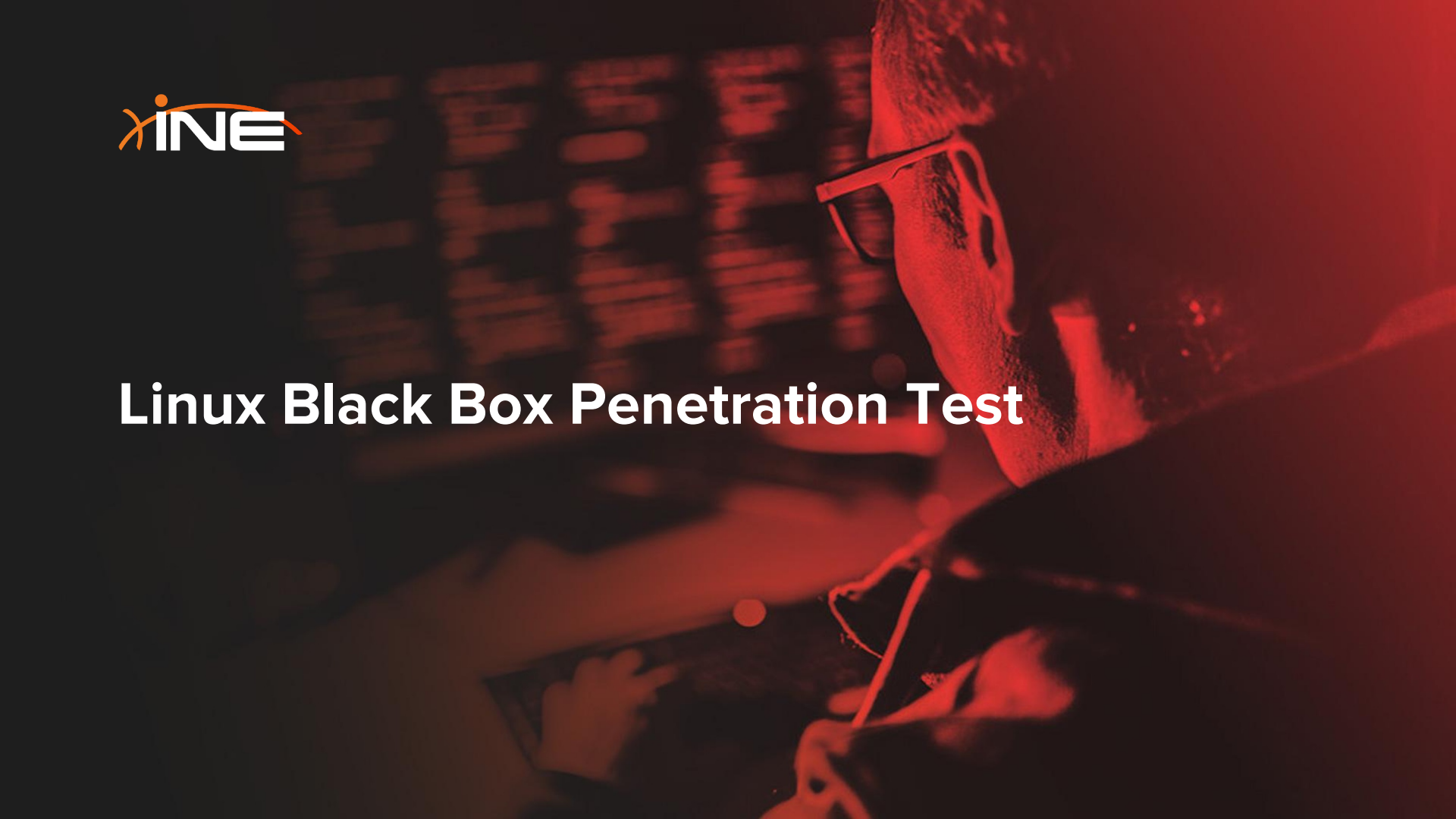

Demo: Targeting SMB



Windows Black Box Penetration Test

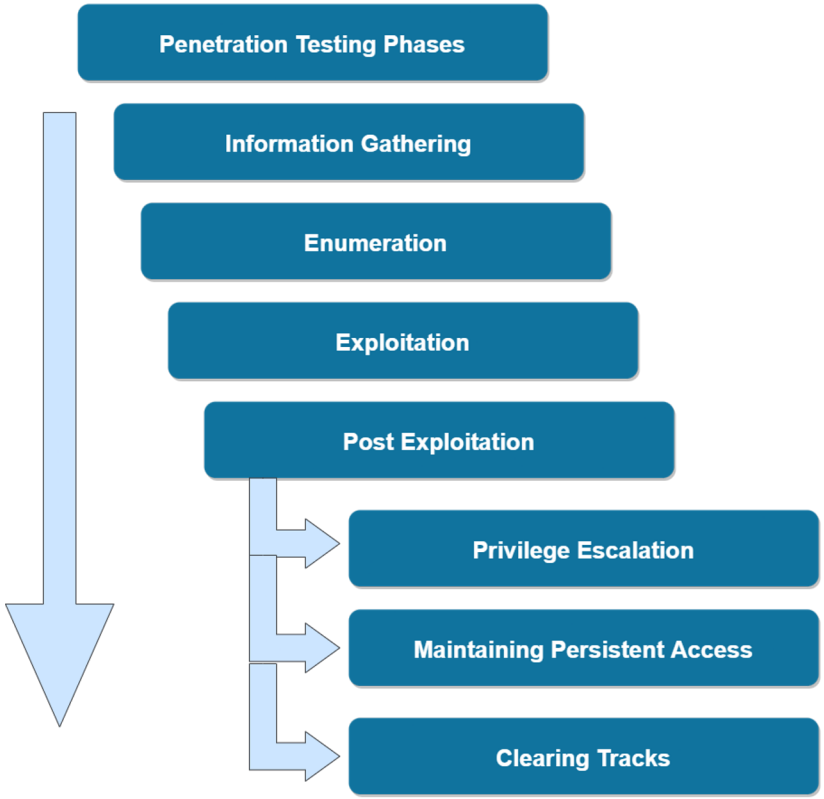
Targeting MySQL Database Server

Demo: Targeting MySQL Database Server



Linux Black Box Penetration Test

Penetration Testing Phases



The following diagram outlines the various phases involved in a typical penetration test.

Black Box Methodology:

- Host discovery
- Port scanning & enumeration
- Vulnerability detection/scanning
- Exploitation
  - Manual
  - Automated
- Post Exploitation
  - Privilege Escalation
  - Persistence
  - Dumping Hashes

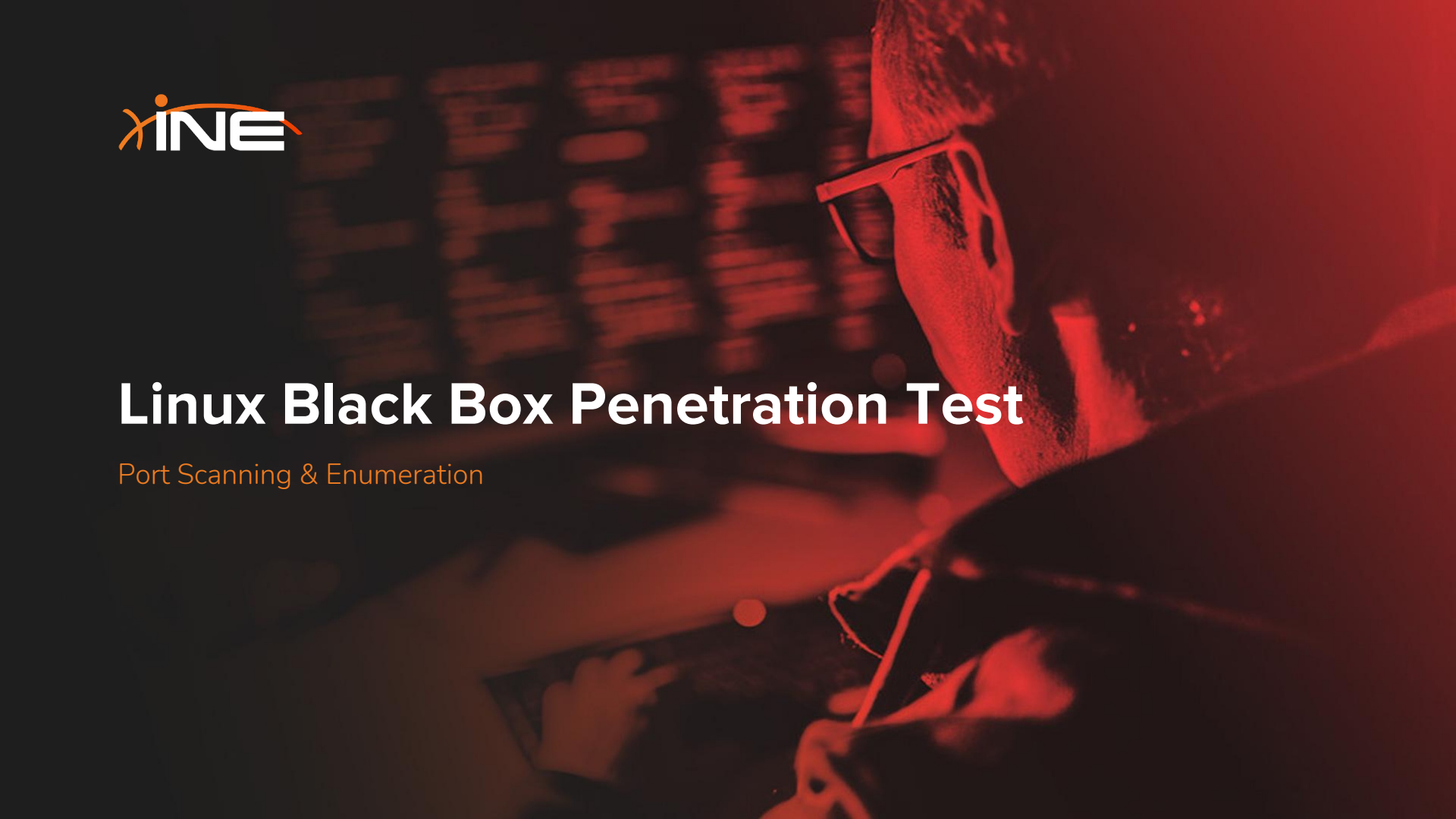
Scenario & Scope

- You have just begun your first job as a Junior Penetration Tester and have been assigned to assist in performing a penetration test on a client's network.

- The pentest lead was pleased with your ability to gain access to the Windows Server target and has assigned you to perform a pentest on a Linux server on the client's network.

- Your primary objectives are:
  - Identify services running on the target
  - Identify vulnerabilities within the services
  - Exploit these vulnerabilities to obtain an initial foothold

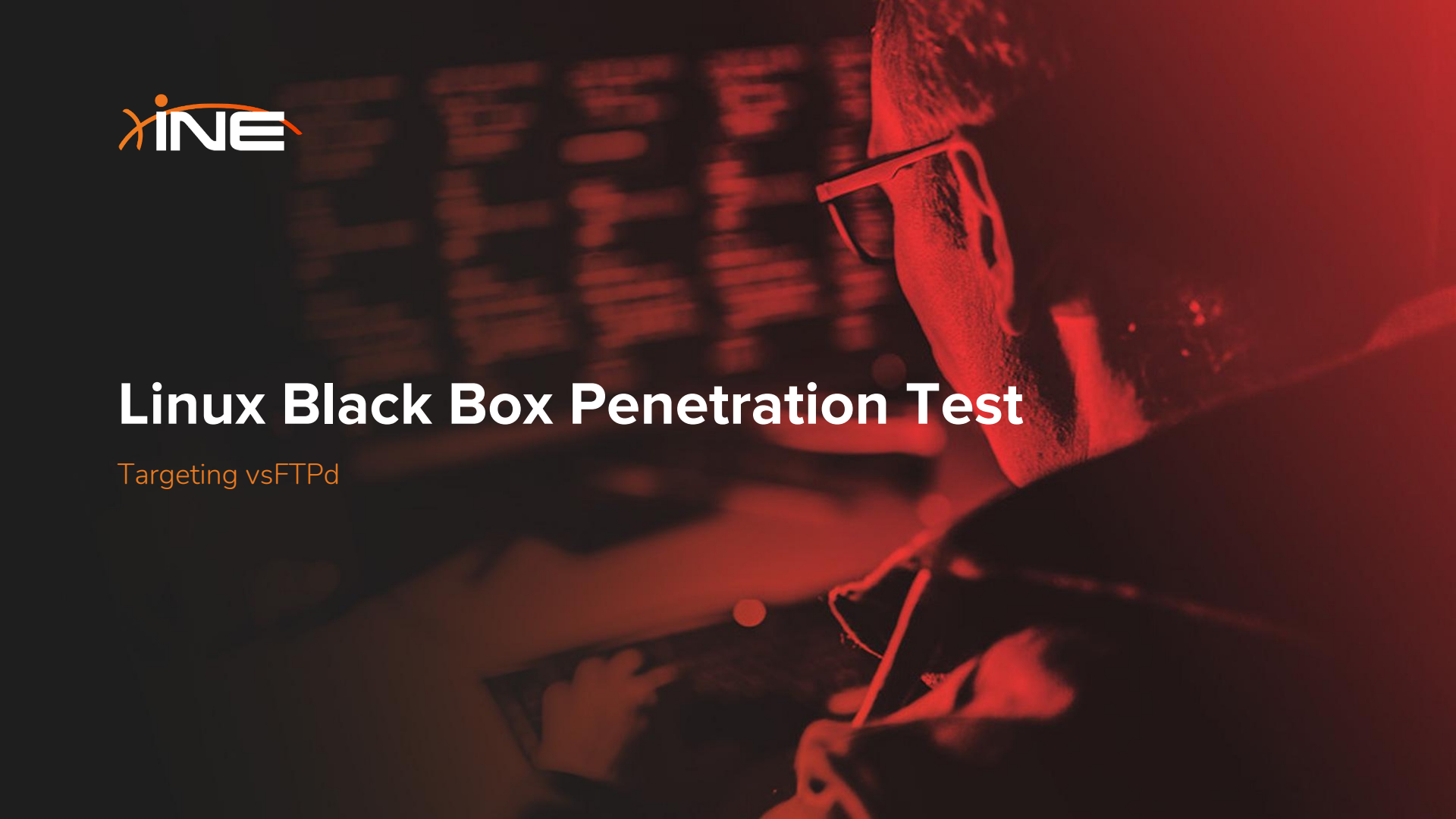
Note: You are permitted to use the Metasploit Framework



Linux Black Box Penetration Test

Port Scanning & Enumeration

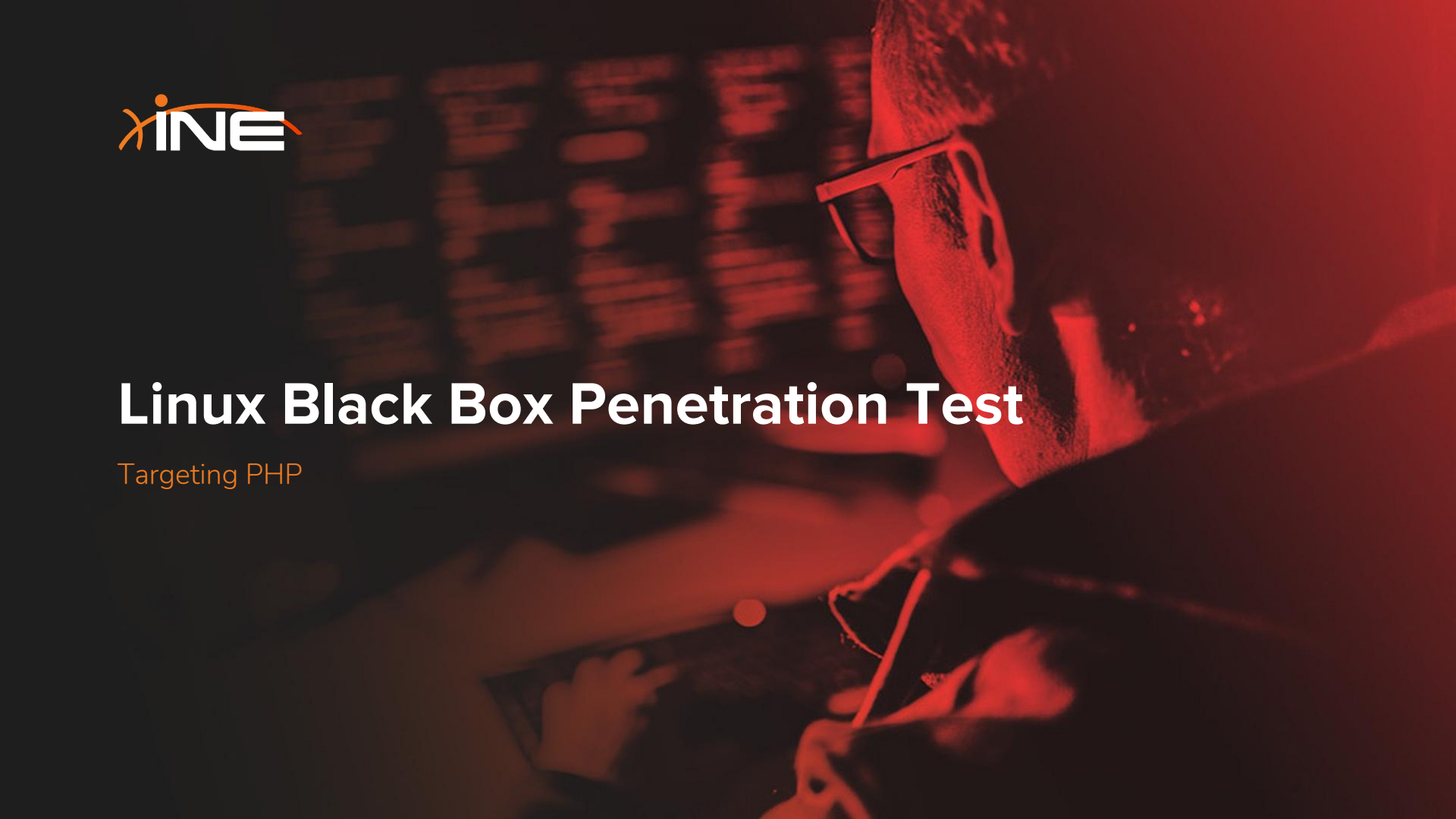
Demo: Port Scanning & Enumeration



Linux Black Box Penetration Test

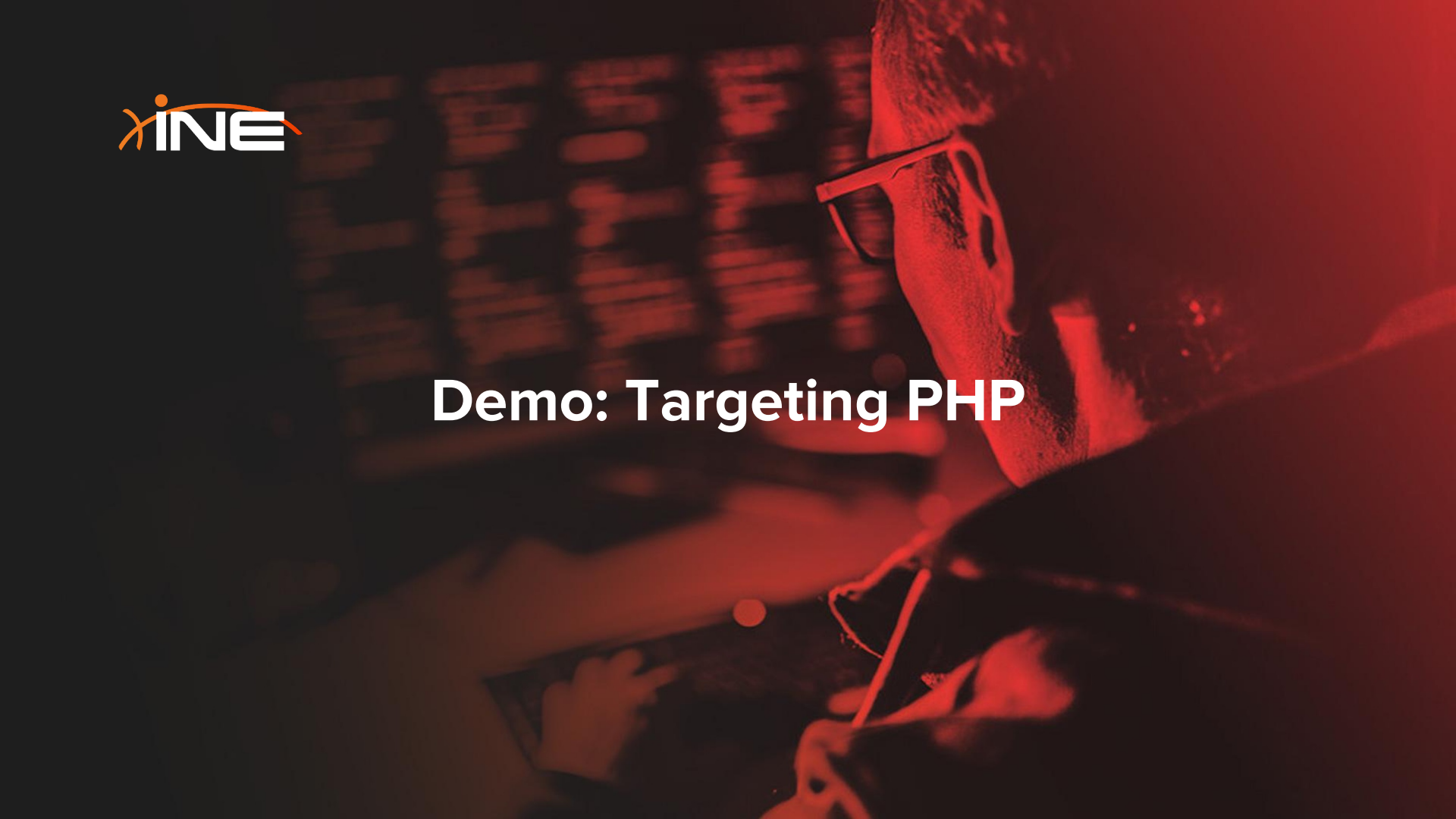
Targeting vsFTPd

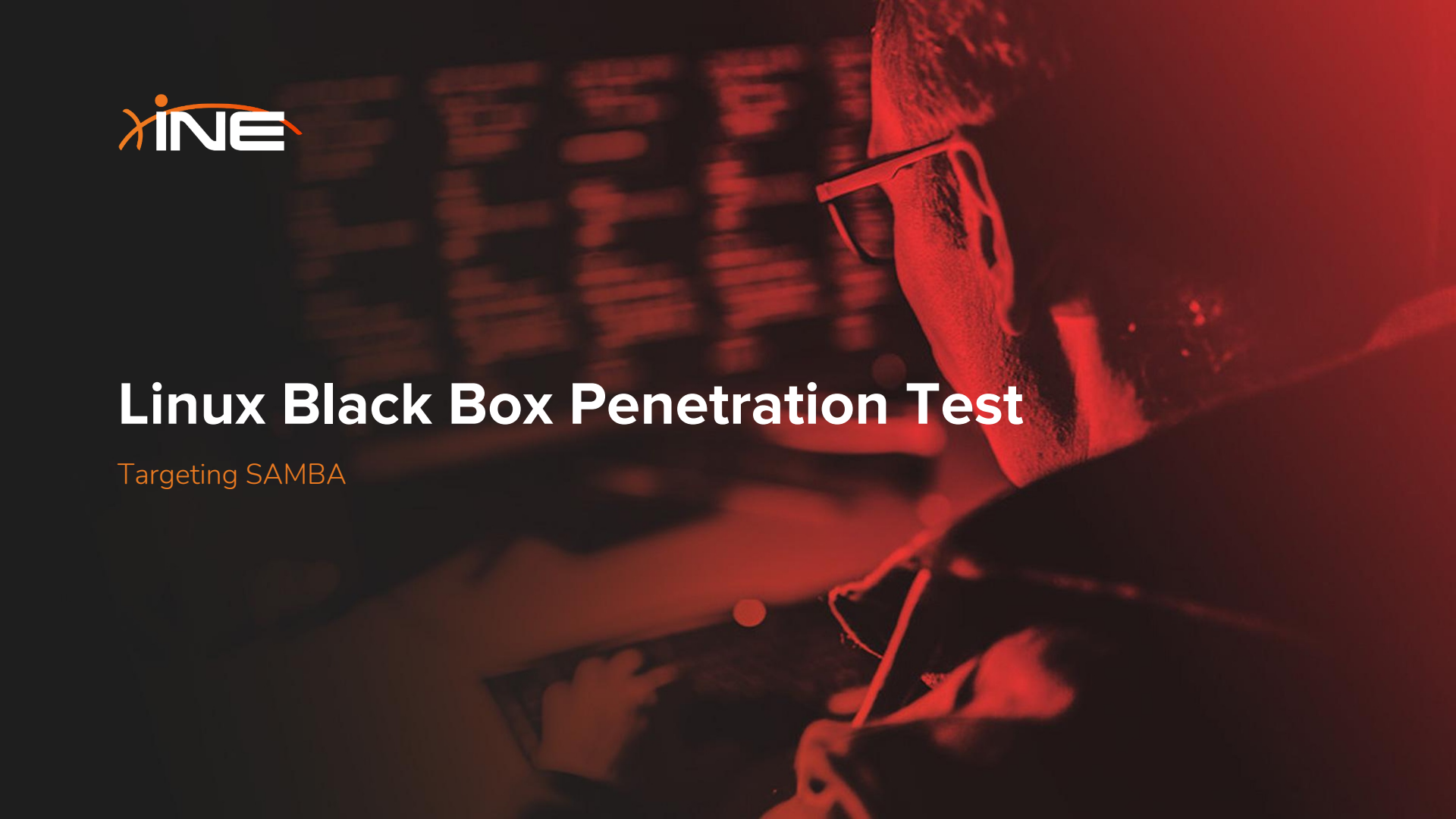

Demo: Targeting vsFTPd



Linux Black Box Penetration Test

Targeting PHP


Demo: Targeting PHP



Linux Black Box Penetration Test

Targeting SAMBA

Demo: Targeting SAMBA

AV Evasion With Shellter

Defense Evasion

Defense Evasion consists of techniques that adversaries use to avoid detection throughout their compromise. Techniques used for defense evasion include uninstalling/disabling security software or obfuscating/encrypting data and scripts. Adversaries also leverage and abuse trusted processes to hide and masquerade their malware. – MITRE

AV Detection Methods

AV software will typically utilize signature, heuristic and behaviour based detection.

1. Signature based detection - An AV signature is a unique sequence of bytes that uniquely identifies malware. As a result, you will have to ensure that your obfuscated exploit or payload doesn't match any known signature in the AV database.

We can bypass signature-based detection by modifying the malware's byte sequence, therefore changing the signature.

2. Heuristic-based detection - Relies on rules or decisions to determine whether a binary is malicious. It also looks for specific patterns within the code or program calls.

3. Behavior based detection - Relies on identifying malware by monitoring it’s behavior. (Used for newer strains of malware)

AV Evasion Techniques

On-disk Evasion Techniques

- Obfuscation - Obfuscation refers to the process of concealing something important, valuable, or critical. Obfuscation reorganizes code in order to make it harder to analyze or RE.

- Encoding - Encoding data is a process involving changing data into a new format using a scheme. Encoding is a reversible process; data can be encoded to a new format and decoded to its original format.

- Packing - Generate executable with new binary structure with a smaller size and therefore provides the payload with a new signature.

- Crypters - Encrypts code or payloads and decrypts the encrypted code in memory. The decryption key/function is usually stored in a stub.

AV Evasion Techniques

In-Memory Evasion Techniques

- Focuses on manipulation of memory and does not write files to disk.
- Injects payload into a process by leveraging various Windows APIs.
- Payload is then executed in memory in a separate thread.

Demo: AV Evasion With Shellter

Obfuscating PowerShell Code

Obfuscation

+ Obfuscation refers to the process of concealing something important, valuable, or critical. Obfuscation reorganizes code in order to make it harder to analyze or RE.

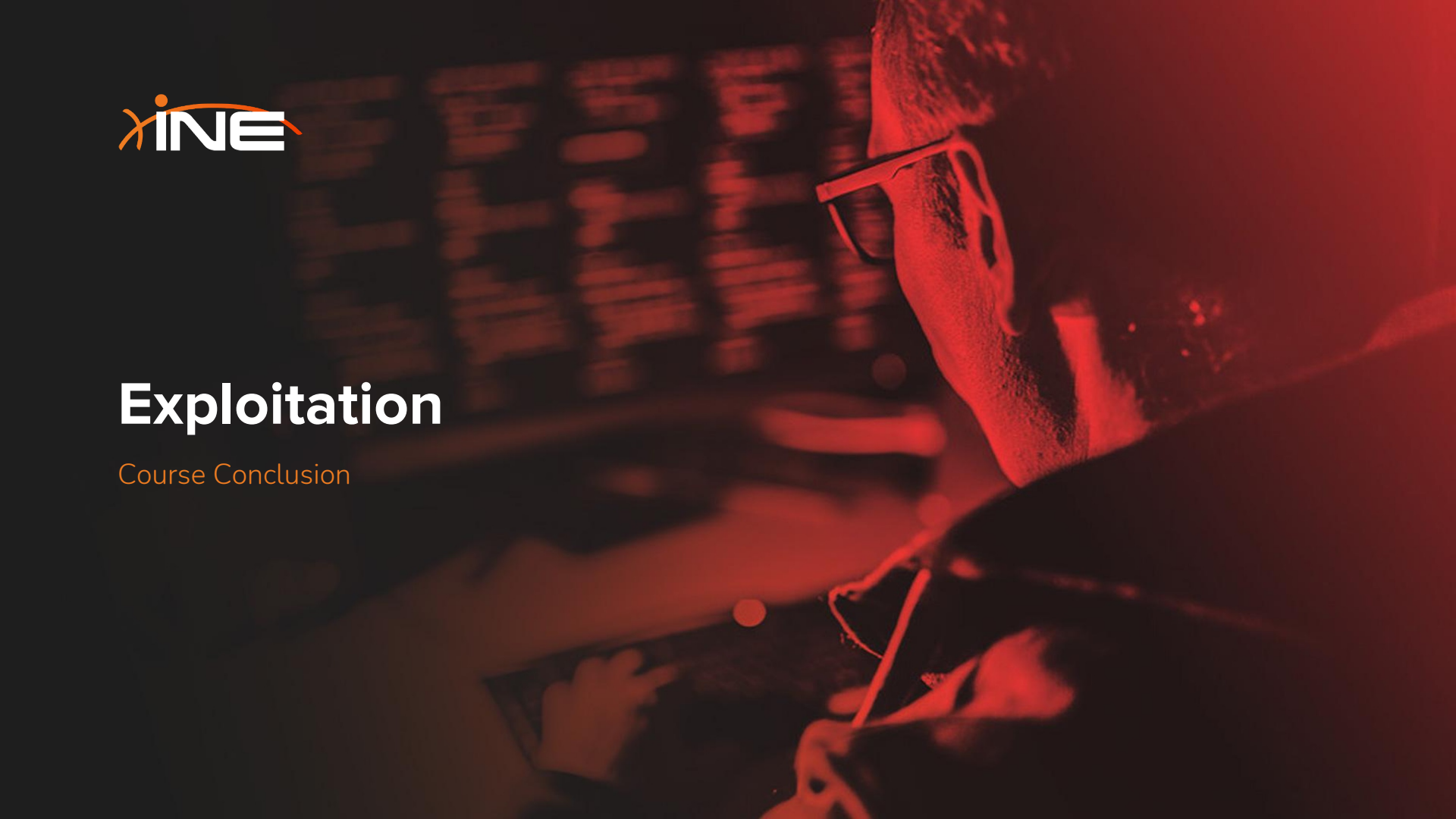
+ As a penetration tester, you will find yourself working with PowerShell code frequently. Most AV solutions will immediately flag malicious PowerShell code, as a result, you must be able to obfuscate/encode your PowerShell code and scripts in order to avoid detection.

Invoke-Obfuscation

- Invoke-Obfuscation is an open source PowerShell v2.0+ compatible PowerShell command and script obfuscator.

GitHub Repo: https://github.com/danielbohannon/Invoke-Obfuscation

Demo: Obfuscating PowerShell Code

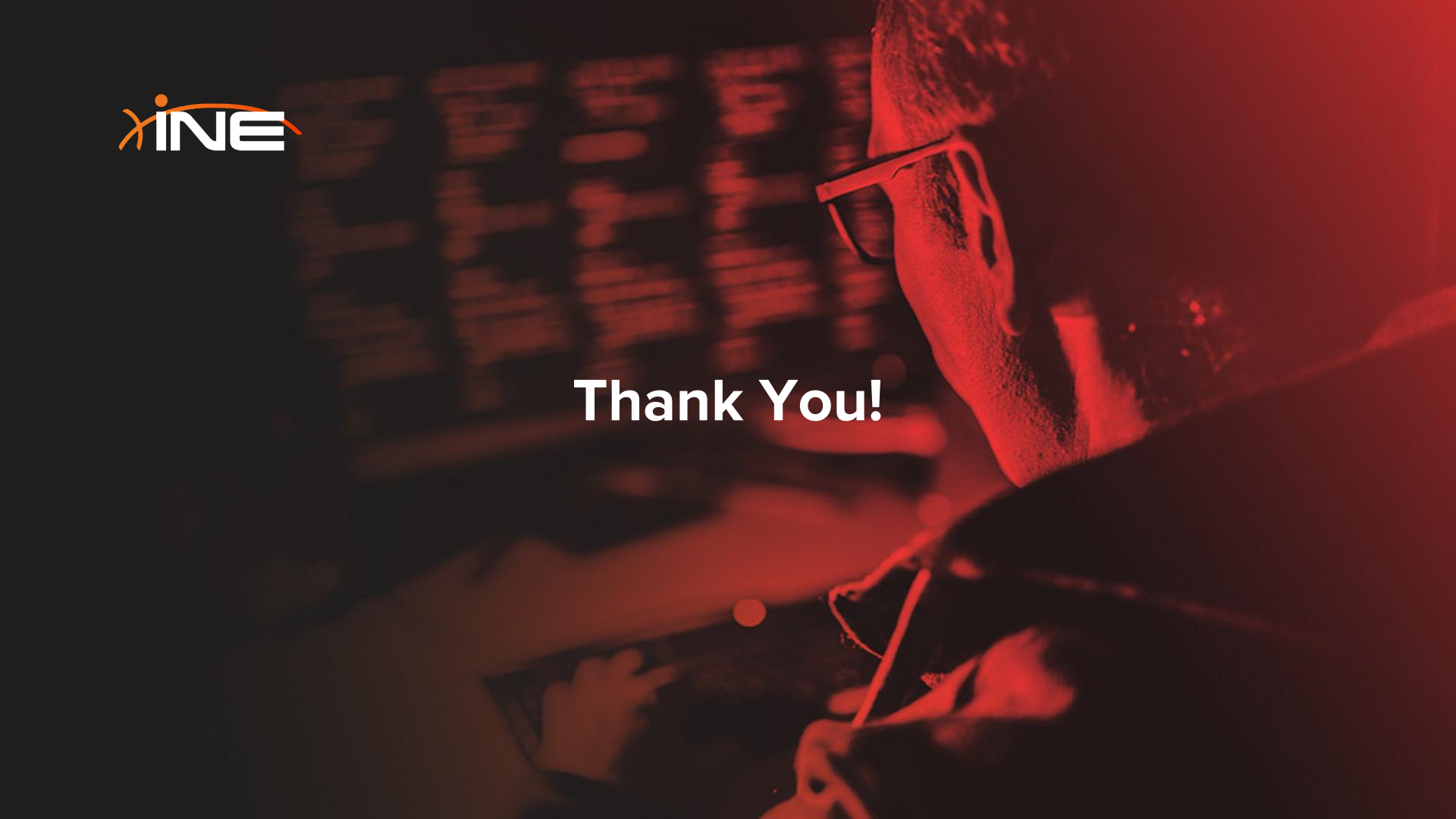


Exploitation

Course Conclusion

Learning Objectives:

- Students will get an introduction to the exploitation phase of a penetration test.
- Students will learn how to identify vulnerable services running on a target system.
- Students will learn how to search for, modify and compile publicly available exploit code.
- Students will get an understanding of how bind and reverse shells work .
- Students will get an understanding of the various exploitation frameworks available as well as how they can be used to streamline exploitation.
- Students will learn how to exploit both Windows & Linux systems in a simulated black box penetration test.
- Students will learn how to evade signature based AV solutions.


Thank You!