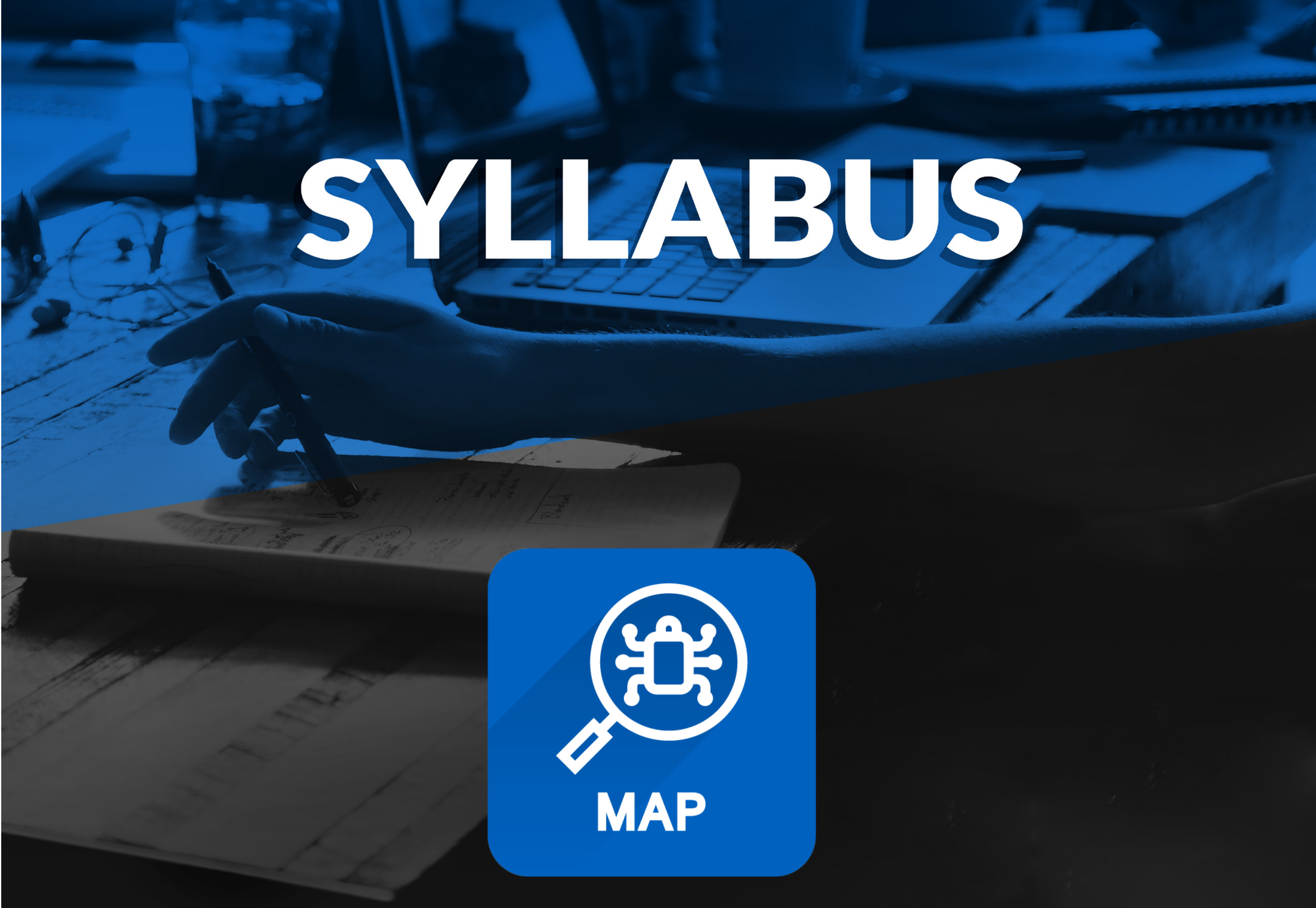


SYLLABUS

MAP

MALWARE ANALYSIS PROFESSIONAL
VERSION 1

eLearnSecurity AN INE COMPANY

eLearnSecurity has been chosen by students in 140 countries in the world and by leading organizations such as:

COURSE DESCRIPTION

Malware Analysis Professional (MAP) is an online, self-paced training course that teaches students the knowledge and skills necessary to dissect malicious software in order to understand its mechanics and purpose.

In this course, you will be able to:

- Work with realistic malware samples created to prepare you for real-world samples
- Analyze real-world samples: ransomware, botnets, rats, etc.
- Explore an entire module dedicated to x64 bit assembly
- Dive into the TLS method
- Understand how malware uses Windows APIs to achieve their malicious activity
- Debug samples using different debuggers

MAP provides a holistic approach to dissecting malware. You will also learn more about Reverse Engineering and add an additional skill to your arsenal, allowing you to dissect a product to understand its blueprint, how it was made, and:

- Understand and bypass Anti-Reversing techniques
- Learn about IA-32 CPU Architecture
- Perform full manual unpacking on packed executables
- Learn different methods to locate the important algorithms

MAP is a self-paced course that comes with 36 labs so you can develop your knowledge and test your skills through hands-on dissection and analysis of malicious software. In the Reverse Engineering portion of MAP, there are 10 downloadable, offline labs (executables) that provide practical reverse engineering experience. Additionally, this content comes with videos that provide step-by-step guidelines, providing an in-depth explanation of every technique.

PREREQUISITES

Basic Knowledge and understanding of:

- Networking and Network Protocols: TCP, UDP, ARP, ICMP, etc
- Operating Systems and Computer Architecture Concepts
- Programming Languages: x86 Assembly, C, C++, and Python
- Information Security: Cyber Attacks, Malicious Content, Exploitation, Shellcodes and Digital Forensic Investigations

WHO SHOULD TAKE THIS COURSE?

The target audience of this course are:
- Incident Responders
- Digital Forensic Examiners
- Malware Analysts
- Penetration Testers who want to adapt Malware methods for their PT
- Reverse Engineers with 0 - 2 yrs of experience
- Cybersecurity Researchers and Students

WILL I GET A CERTIFICATE?

eLearnSecurity's eCMAP (Certified Malware Analysis Professional) certification is the most practical and professionally oriented certification you can obtain in malware analysis. Instead of putting you through a series of multiple-choice questions, you are expected to perform a full analysis on a given malware sample, show proof of what the malware does, and finally write a signature that could be used to detect the malware sample on other systems or networks. This will be done by applying all or part of the skills acquired from the course and research.

ORGANIZATION OF CONTENTS

The student is provided with a suggested learning path to ensure the maximum success rate and its minimum effect.

MODULE 1: INTRODUCTION TO MALWARE ANALYSIS

Module 1 serves as a basic introduction to Malware, the different types of Malware, what they are and how to analyze them. This module also covers malware analysis techniques that are used in order to analyze malware and why we need different methods and techniques. Finally, the module also covers many of the tools that could be used to acquire evidence, including malware.

MODULE 2: STATIC ANALYSIS TECHNIQUES

This module introduces the basic static methods used to identify malware and run an initial assessment. The module covers the different file types, generating different types of hashes and how they can be helpful, how to extract strings and use them as starting points in your analysis, and when and why you might consider using an online scanner or a sandbox. The module also goes in-depth on the Portable Executable (PE) file format and how to analyze its structure and understand all the main values in it. Finally, this module explains how to use all of this information to build an indicator of compromise (IOC) and use it to find malware samples on other systems using YARA.

MODULE 3: ASSEMBLY CRASH COURSE

Since most systems used today are 64-bit, this module is considered a crash course to x64 assembly language. How x64 is different from x86, why malware analysts should be skilled at analyzing 64-bit code and even how to write some x64 assembly code, will all be covered in this module.

MODULE 4: BEHAVIOR ANALYSIS

Module 4 begins by explaining Windows processes, threads and the different objects available to a process. This is where malware is run to understand its behavior and how it affects a victim's system. The module goes through the different injection methods used by malware and how they achieve persistence on systems, so they can come back or have a foothold on the victim's system or network. The module also covers the different tools to use and how to use them effectively to analyze malware, whether they were EXEs or DLLs. Finally, the module also explains how to use a sandbox to automate your dynamic analysis and get faster results that could aid your further investigations.

4.1 Introduction
4.2 Dynamic Analysis
4.3 Windows Processes
4.4 Sysinternals Tools
4.5 System Processes and Services
4.6 Injection Techniques
4.7 Persistent Methods
4.8 Tools and Automation
4.9 Windows APIs

MODULE 5: DEBUGGING AND DISASSEMBLY TECHNIQUES

This module covers two of the most advanced malware analysis methods: debugging and disassembling. Why are these methods needed and when to use them, their pros and cons, and what tools are available, are all core concepts covered in this module. The different debugging methods, breakpoints and controls will be covered and how use them to run and analyze a malware sample. Finally, this module covers disassembly and reverse engineering in greater detail with the focus on recognizing common malware characteristics at the Windows API level.

5.1 Introduction
5.2 Debugging and Debuggers
5.3 Disassembly and IDA Pro
5.4 Other Tools

MODULE 6: OBFUSCATION TECHNIQUES

One of the goals malware developers try to achieve is to hinder analyzing their malware code. Therefore, to achieve that, they would use many different methods and techniques. This module covers the most common obfuscation techniques used by malware developers ranging from decoding (Base64, XOR, etc), using anti-debugging and anti-reverse engineering techniques. This module goes over what packing is in detail and how to apply different unpacking techniques to unpack different malware samples. Finally, the module covers shellcode (mainly 64-bit shellcode) and how to locate, extract and analyze it.

MODULE 1: THE NECESSARY THEORY PART 1

The first three modules aim to cover all the necessary theory as well as the concepts on which the practical part of this course is based. We will start with a short description about what Reverse Engineering is and the reasons why someone might need it, and then proceed with more technical concepts. During the first three chapters we will discuss the basics behind the Intel IA-32 CPU architecture (x86), the stack, the heaps, as well as exceptions, Windows APIs with some Windows Internals, and the most common types of reversing tools used these days.

MODULE 2: THE NECESSARY THEORY PART 2

So here we are in the second module, which is also dedicated to the theoretical knowledge necessary for this course. One thing to keep in mind is that 'theoretical' doesn't actually mean that you might need it…or not. In fact, the theory discussed during these first three modules covers all the fundamental knowledge and the concepts that you will need, not just for this course and its technical assignments, but for the rest of your time as a reverser.

MODULE 3: THE NECESSARY THEORY PART 3

The third module of this course aims to offer some extra theoretical knowledge necessary for the rest of the course. During this module we will briefly touch on the concept of heaps, we will discuss handles, exceptions, some basic Windows Ring3 Internal structures, and Windows APIs. Finally, we'll go through the most common types of reversing tools used today for software reverse engineering.

MODULE 4: VA/RVA/OFFSET AND PE FILE FORMAT

In this module we will discuss virtual addresses, relative virtual addresses, offsets, as well as some basic information regarding the Portable Executable File Format which describes the basic structure of all Windows executable files.

MODULE 5: STRING REFERENCES AND BASIC PATCHING

This module is dedicated to 'String References' as well as Basic Memory and File Patching. We demonstrate the use of data strings in order to locate the algorithm we are interested in and then we reverse its logic. Finally, we explain how we can manually calculate the offset of a byte inside the physical file by knowing its virtual address in memory.

MODULE 6: EXPLORING THE STACK

This module focuses on exploring the data that we can retrieve from the stack in order to trace back an algorithm. A very important technique when we have to deal with on-the-fly encryption and decryption of data.

MODULE 7: ALGORITHM REVERSING

During this module, we dig deep into Reverse Engineering by analyzing in detail all the important algorithms of the executable which include the data encryption/decryption algorithm as well as the input data validation algorithm.

MODULE 8: WINDOWS REGISTRY MANIPULATION

This module is dedicated to Windows Registry. We start with an overview of this important Windows component and then we proceed with the detailed analysis of an executable that attempts to read data from the registry and validate it according to a custom algorithm which we finally Reverse Engineer. During this module we also make use of Hardware Breakpoints and we demonstrate their importance.

MODULE 9: FILE MANIPULATION

During this module we Reverse Engineer an executable that attempts to locate a specific file in the system and read data from it. In addition, we once more analyze in detail the custom algorithm used to validate that data in order to extend our skills in Reverse Engineering custom algorithms.

MODULE 10: ANTI-REVERSING TRICKS PART 1

This is the first module dedicated to Anti-Reversing tricks which includes some basic direct and indirect ways to detect a Ring3 debugger.

MODULE 11: ANTI-REVERSING TRICKS PART 2

In this module we continue talking about Anti-Reversing tricks regarding debuggers and reversing tools detection methods.

MODULE 12: ANTI-REVERSING TRICKS PART 3

This module is again focused on Anti-Reversing tricks. In this case we discuss differences between SW and HW breakpoints and how they can be detected. We also talk about more advanced tricks that involve the use of exceptions, and finally we talk about some well-known methods for detecting a few popular VM environments.

MODULE 13: CODE OBFUSCATION

In this module we discuss different types of native code obfuscation methods. We explain how these are implemented, the obstacles that can be created and how we can analyze and cleanup obfuscated code.

MODULE 14: ANALYZING PACKERS AND MANUAL UNPACKING

This module focuses on executables packers and more specifically on different generic methods that we can use in order to successfully find the Original Entry Point of applications packed with common packers. We give practical examples and we unpack them together for fun and knowledge.

MODULE 15: DEBUGGING MULTI-THREAD APPLICATIONS

In this module we discuss debugging and the analysis of multi-thread applications, or applications that are able to execute various blocks of code via different threads. Reverse Engineering multi-thread applications can sometimes be quite frustrating, especially for beginners.

The MAP course is a practice-based curriculum. Being integrated with Hera Lab, the most sophisticated virtual lab in IT Security, it offers an unmatched practical learning experience. Hera is the only virtual lab that provides fully isolated per-student access to each of the real-world scenarios available on the platform. Students can access Hera Lab from anywhere through VPN.

Modules will be accompanied by 26 hands-on malware analysis labs, with an additional 10 Win32 applications to reverse engineer.

SECTION 1 MALWARE ANALYSIS LABS

MODULE 1

- Lab 1: Evidence Acquisition using KAPE

MODULE 2

- Lab 2: File Identification
- Lab 3: Analyzing PE File Structures
- Lab 4: Packed Malware Identification And Basic Analysis
- Lab 5: From IOCs to YARA Rules

MODULE 3 LABS

- Lab 6: Writing and Debugging Assembly x64 Code

MODULE 4 LABS

- Lab 7: Working with Windows Processes
- Lab 8: Analyzing a Custom Downloader
- Lab 9: Working with DLLs and DLL Injection
- Lab 10: Dynamically Analyzing a Custom Backdoor
- Lab 11: Dynamically Analyzing a KeyLogger

MODULE 5 LABS

- Lab 12: Reverse Engineering a 64-bit Downloader Using x64dbg
- Lab 13: Debugging a 64-bit Downloader Using x64dbg
- Lab 14: Debugging a 64-bit Dropper
- Lab 15: Reverse Engineering a Keylogger using IDA Pro
- Lab 16: Reverse Engineering a Bot Using IDA Pro
- Lab 17: Analyzing the WannaCry Ransomware
- Lab 18: Reverse Engineering a Custom Backdoor using IDA Pro (64-bit)

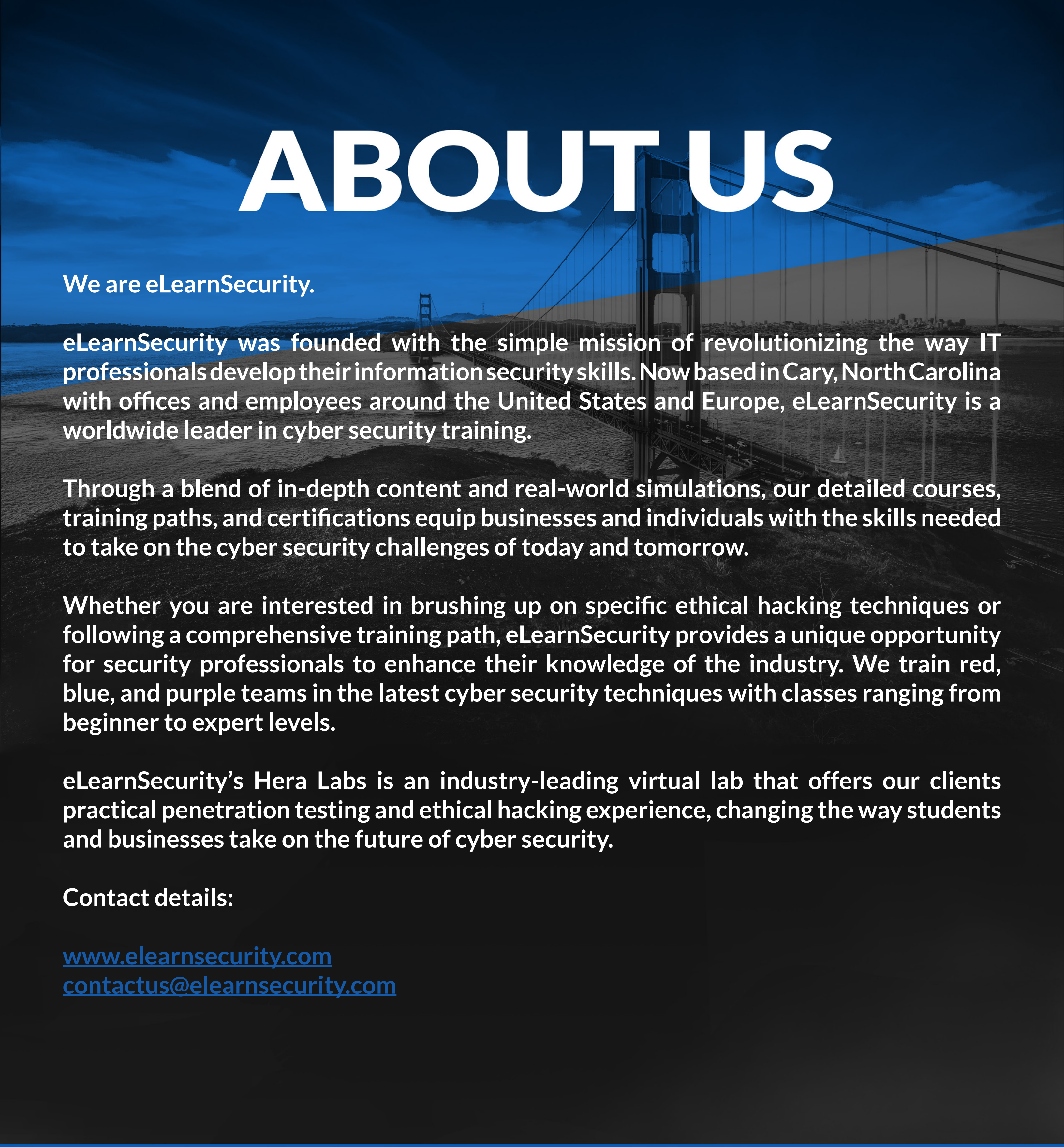
MODULE 6 LABS

- Lab 19: Manually Unpacking a Malware Using x64dbg
- Lab 20: Manually Unpacking UPX using x64dbg
- Lab 21: Manually Unpacking Real-Life Sample (Redaman)
- Lab 22: Manual Unpacking Real-Life Sample (Locky
- Lab 23: Binary Patching KillemAll Malware
- Lab 24: Debugging Obfuscated Downloader
- Lab 25: Debugging Process Hollowing (RunPE)
- Lab 26: Debugging Process Hollowing with TLS Callbacks

SECTION 2 REVERSE ENGINEERING LABS

MODULES 5 - 15 LABS

- Lab 27: String References & Basic Patching
- Lab 28: Exploring the Stack
- Lab 29: Algorithm Reversing
- Lab 30: Windows Registry Manipulation
- Lab 31: File Manipulation
- Lab 32: Anti Reversing Tricks I
- Lab 33: Anti Reversing Tricks II
- Lab 34: Anti Reversing Tricks III
- Lab 35: Code Obfuscation
- Lab 36: Analyzing Packers & Manual Unpacking



ABOUT US

We are eLearnSecurity.

eLearnSecurity was founded with the simple mission of revolutionizing the way IT professionals develop their information security skills. Now based in Cary, North Carolina with offices and employees around the United States and Europe, eLearnSecurity is a worldwide leader in cyber security training.

Through a blend of in-depth content and real-world simulations, our detailed courses, training paths, and certifications equip businesses and individuals with the skills needed to take on the cyber security challenges of today and tomorrow.

Whether you are interested in brushing up on specific ethical hacking techniques or following a comprehensive training path, eLearnSecurity provides a unique opportunity for security professionals to enhance their knowledge of the industry. We train red, blue, and purple teams in the latest cyber security techniques with classes ranging from beginner to expert levels.

eLearnSecurity's Hera Labs is an industry-leading virtual lab that offers our clients practical penetration testing and ethical hacking experience, changing the way students and businesses take on the future of cyber security.

Contact details:

www.elearnsecurity.com
contactus@elearnsecurity.com