REDOPS
INFORMATION SECURITY

Live Training

Endpoint Security Insights: Shellcode Loaders and Evasion

Table of Contents

1 Introduction 3
2 Disclaimer 5
3 Key Learning Objectives 6
4 What this course isn't! 7
5 Who should participate? 8
6 Course Schedule and Time Zones 9
7 Participant Requirements 10
8 Hardware Requirements 11
9 Software Requirements 12
10 LAB 13
10.1 Important Information 13
11 Course Outline and Timetable 14
11.1 Day 1 15
11.2 Day 2 18
11.3 Day 3 20
11.4 Day 4 22
11.5 Course Content Disclaimer 24
12 Benefits 25
13 Workshop Pricing 26
14 NDA 27
15 Frequently Asked Questions (FAQs) 28
16 Copyright 30
17 Contact 31

1 Introduction

In recent years, the task of evading Endpoint Protection (EPP) and Endpoint Detection and Response (EDR) products has become increasingly challenging and is now a critical aspect of any red team engagement. Each EPP/EDR system has its own unique set of strengths and weaknesses, and there is no one-size-fits-all solution to EPP/EDR evasion. However, a solid understanding of the basics of EPP/EDR evasion and shellcode loader development is essential, even at the junior level.

Over the past six years, I have gained extensive knowledge of endpoint security evasion using a variety of different endpoint security products. I would like to share this knowledge in the context of shellcode loaders and endpoint security evasion in my new **4-day online** course "**Endpoint Security Insights: Shellcode Loaders and Evasion**".

A good part of the course time (about 50%-60%) will be spent on step-by-step development and debugging of various types of basic shellcode loaders in C (and partly in assembly), focusing on direct syscalls, indirect syscalls, unhooking encryption, metadata, entropy, etc. In general, most of the course time (about 70-80%) is hands-on time for the students. As the focus of this course is not entirely or only on malware development, students will not have to write code from scratch, but will have to solve different types of tasks in each module to complete different types of shellcode loader POCs. For me as a teacher, the most important thing is that each student can fully understand the functionality of each evasion technique and fully understand the functionality of each part of each shellcode loader.

Focusing on the Windows platform, the creation of the shellcode loaders will be done in Visual Studio 2022 and the debugging will be done with WinDbg and x64dbg.

The primary objective of this workshop is to provide an engaging workshop that focuses on teaching students to:

- Understand the necessary basics of Windows internals, such as Windows NT architecture, Win32 APIs, native APIs, system calls, etc.

- Understand the architecture of endpoint security products and the mechanisms they use for prevention or detection, such as AMSI, ETW, hooking, callbacks, etc.

- Understand different types of evasion techniques such as direct syscalls, indirect syscalls, module stomping, unhooking via software breakpoints, unhooking via hardware breakpoints, etc. that can be implemented in shellcode loaders to evade endpoint security products.

- Step-by-step creation of shellcode loaders using various evasion techniques mentioned above.

- Debug the crafted shellcode loaders to decipher their behavior and the potential Indicators of Compromise (IOCs) that EDRs use for detection.

- Evaluate the crafted shellcode loaders against an endpoint security product. For various reasons, only offline testing (no endpoint security cloud) combined with offline listeners (internal IP) is provided.

2 Disclaimer

The basic content for this training is freely available on the internet in various forms. I want to make it clear that I am not introducing any new tactics or techniques in my training. If you share my enthusiasm for self-taught learning, I encourage you to explore these resources for yourself and take advantage of the vast wealth of information available online.

However, if you are looking for high quality live training and a structured approach to learn about some essential concepts of endpoint security evasion and shellcode loaders, then my course "Endpoint Security Insights: Shellcode Loaders and Evasion" is a good choice for you.

3 Key Learning Objectives

By completing this course, students will:

- Improve your understanding of the fundamentals of Windows internals, enabling you to better understand endpoint security products and construct evasive shellcode loaders with greater proficiency.
- Develop a thorough understanding of endpoint security defence mechanisms, including user-mode hooking, event tracing for Windows, callbacks and more.
- Gain a thorough understanding of Win32 APIs, native APIs, system calls, direct syscalls, indirect syscalls etc.
- Improve your understanding of shellcode storage and encryption techniques.
- Gain a better understanding of how the entropy of a shellcode loader affects EDR evasion and how the entropy aspect of a shellcode loader can be improved.
- Gain the skills to systematically write your own indirect syscall shellcode loader or create advanced evasive shellcode loaders using Visual Studio.
- Gain the skills to systematically analyze, debug and understand your shellcode loaders through step-by-step guidance.

4 What this course isn't!

- A silver bullet for endpoint security evasion. The focus is on learning the functionality of evasion techniques such as indirect syscalls and how to implement and debug them in a shellcode loader.
- A complete developer course (students do not write code from scratch, but based on the relevant playbook, students must complete and debug the appropriate POC for each chapter)
- A full reverse engineering course
- A full Windows internals course

5 Who should participate?

The course will be of particular interest to penetration testers, junior red teamers, blue teamers, etc. who want to deepen their knowledge of EDRs on Windows, how they work in detail, and how to evade them. The course is also designed for those who want to learn more about Windows Internals to better understand the functionality of endpoint security products on Windows, learn creating, debugging, and understanding their own evasive shellcode loaders.

- Junior Penetration Tester and Junior Red Teamer
- Blue Teamer, Threat Hunter etc.
- General infosec professionals or infosec beginners with a strong desire to learn more about the basics from Windows Internals, EPPs/EDRs on Windows, malware development and debugging etc.

6 Course Schedule and Time Zones

The training is designed to be delivered as both online live training and live onsite training. If delivered as an online live training, the training will take place in Central European Time (CET).

The first run of this course as an online live training is scheduled for the **11th – 14th of November 2024**. Future dates will be published on https://redops.at/en.

7 Participant Requirements

The following requirements may be helpful, but they are not mandatory:

- Basics in C and x64 Assembly Language
- Basics with Visual Studio
- Basics with WinDbg and x64dbg

8 Hardware Requirements

To effectively engage in this course, please ensure to bring your own hardware which meets the following requirements:

- Notebook or PC with virtualization capable CPU(s)
- Minimum 16 GB of RAM (for running two guest VM)
- Minimum 120 GB of free disk space

9 Software Requirements

Ensure that the following software is locally installed and configured on the notebook you will be using to attend the workshop.

- Host operating system Windows 10 Professional 64-bit
- Microsoft Remote Desktop Client (to access the host)
- Zoom client (to join the workshop)
- Discord account (for written questions during the workshop)

10 LAB

Each student has access to their own dedicated lab environment consisting of the following virtual machines.

- Windows 10 Development
- Kali Linux
- Windows 10 with endpoint security in place (offline mode)

10.1 Important Information

Please note that for ethical and legal reasons we cannot provide commercial C2 frameworks, endpoint security products with internet or cloud connectivity in the provided lab environment.

Even if it is not ideal from an evasion point of view, only the **free version of the Metasploit framework** will be used **throughout the course**. Furthermore, if access to a Windows VM with Endpoint Security installed is provided, this will only be made available in offline mode.

Of course, every course participant is free to test the self-created shellcode loaders in their own company lab with commercial C2 frameworks and online EPP/EDR. However, you are welcome to use your favorite C2 framework outside the official LAB environment, but please note that I may not be familiar with your chosen C2 framework and will not be able to fully support you during the exercises.

11 Course Outline and Timetable

Endpoint Security Insights: Shellcode Loaders and Evasion is a **4-day interactive** (online) course designed for infosec professionals. My role as instructor is to present the essential theory and concepts for each module or chapter, which is about 20%-30% of the content. The remaining 70%-80% is devoted to practical exercises and answering your questions. The focus in relation to the shellcode loader part is not only to build the loader, furthermore we want to debug and understand each loader going in the direction of building an evasive shellcode loader.

11.1 Day 1

Module Name	Module Details	Time plan
Introduction to the Course	• Overview of topics to be covered. • Course objectives and expectations	8:00 am – 8:30 am
Windows Internals Fundamentals	• Understanding the Windows NT Architecture • Introduction to Win32-APIs	8:30 am – 9:00 am
Introduction to endpoint security	• Introduction to endpoint security • Differences between Anti-Virus (AV) and EDR • Introduction to relevant endpoint security mechanisms	9:00 am – 9:30 am
Hands-on: EDR evaluation or analysis	• Identify key components and detection mechanisms of endpoint security products	9:30 am – 10:00 am
	Break	10:00 am – 10:15 am
Introduction to Shellcode Loader	• Introduction to shellcode loaders • Understanding the concept of shellcode loaders • Different types of shellcode execution / process injection techniques • Different types of shellcode loaders • Key components and characteristics of an evasive shellcode loader	10:15 am – 11:00 am
Hands-on: Creating a Win32 API shellcode loader	• Learn the basics of shellcode loaders in the context of Win32 APIs	11:00 am – 11:45 am

	• Comprehensive walkthrough: build, debug and understand the inner workings of your Win32 API loader	
Shellcode/payload placement and staging	• Learn where and how to store your payload/shellcode in different types of locations. • Learn how to store shellcode outside of your loader and the differences, advantages and disadvantages compared to local storage.	11:45 am – 12:15 pm
	Lunch Break	12:15 pm – 13:00 pm
Hands-on: Payload placement and staging	• Play with placing your payload in your Win32 API loader in different types of locations like .text, .data etc. • Analyze the differences with debugging etc.	13:00 pm – 13:30 pm
Payload Obfuscation	• Learn how to obfuscate your payloads such as UUID, MAC, etc.	13:30 pm – 14:00 pm
Hands-on: Payload Obfuscation	• Play with different types of payload obfuscation. • For example, obfuscate your shellcode with UUID and implement it in your Win32 API shellcode loader.	14:00 pm – 14:30 pm
Payload Encryption	• Learn about how to encrypt your payload like XOR, RC4 etc.	14:30 pm – 15:00 pm

	Break	15:00 pm – 15:30 pm
Summary and Q&A for day 1	• Let us summarize day 1 and ask all your questions about the content and uncertainties of day 1.	15:30 pm – 16:30 pm

11.2 Day 2

Module Name	Module Details	Time plan
Windows Internals Fundamentals	• Introduction to NTDLL, Native APIs and system calls in Windows OS	8:00 am – 8:30 am
Hands-on: Native Functions	• Debugging native functions or native APIs	8:30 am – 9:00 am
Hands-on: Creating a Native API Loader	• Time to step down a level and move from Win32 APIs to NTAPIs. • Comprehensive walkthrough: build, debug and understand the inner workings of your NTAPI loader	9:00 am – 10:00 am
	Break	10:00 am – 10:15 am
Introduction to EDR user mode hooking and direct syscalls	• An insight into endpoint security and the importance of user mode hooking • What direct syscalls are and why they could (or did) help bypass endpoint security	10:15 am – 10:45 am
Hands-on: Creating a Direct Syscall Loader and hardcoded SSNs	• Again, it is time to move down a level and switch from NTAPIs to Direct Syscalls. • Comprehensive walkthrough: build, debug and understand the inner workings of your Direct Syscall shellcode loader	10:45 am – 12:00 pm
	Lunch Break	12:00 pm – 13:00 pm

Introduction to Event Tracing for Windows (ETW) and indirect syscalls	• Introduction to EDRs and their relationship to Event Tracing for Windows (ETW). • Theoretical foundation: Understanding indirect system calls.	13:00 pm – 13:30 pm
Hands-on: Creating an indirect syscall loader with hardcoded SSNs	• Transition: Move from direct syscalls to indirect syscalls. • Comprehensive walkthrough: Build, debug and understand the inner workings of your indirect syscall loader.	13:30 pm – 14:30 pm
Hands-on: Creating an Indirect Syscall Loader with Dynamic SSN Retrieval via APIs	• Improve your endpoint security evasion capabilities from your Indirect Syscall shellcode loader. • Transition from hardcoded SSN methods to dynamic SSN retrieval via APIs • Comprehensive walkthrough: Build, debug and understand your enhanced Indirect Syscall Loader.	14:30 pm – 15:30 pm
	Break	15:00 pm – 15:30 pm
Summary and Q&A for day 2	• Let us summarize day 2 and ask all your questions about the content and uncertainties of day 2.	15:30 pm – 16:30 pm

11.3 Day 3

Module Name	Module Details	Time plan
Windows Internals Fundamentals	• Introduction to the Process Environment Block (PEB)	8:00 am – 8:30 am
Hands-on: Creating an Indirect Syscall Loader with Dynamic SSN Retrieval via PEB walk	• Transition from dynamic SSN retrieval via APIs to dynamic SSN retrieval via PEB/EAT. • Comprehensive walkthrough: Build, debug, and understand the inner workings of your extended indirect syscall loader.	8:30 am – 9:30 am
	Break	9:30 am – 9:45 am
Hands-on: Indirect Syscall Loader and EDR Hooks Part 1	• Addressing the Chicken/Egg Dilemma: Indirect Syscalls and EDR's User Mode Hooking Part 1 → Halos Gate Approach • Continue to improve your indirect syscall shellcode loader and implement the Halos Gate approach.	9:45 am – 10:45 am
Hands-on: Indirect Syscall Loader and EDR Hooks Part 2	• Addressing the Chicken/Egg Dilemma: Indirect Syscalls and EDR's User Mode Hooking Part 2 → Tartarus Gate Approach • Continue to improve on your Indirect Syscall shellcode loader and implement the Tartarus Gate approach.	10:45 am – 11:45 am
Indirect syscalls vs user mode unhooking	• What are the differences, advantages, or disadvantages between (in)direct syscalls and unhooking in the context of a shellcode loader?	11:45 pm – 12:15 pm

	Lunch Break	12:15 pm – 13:00 pm
Introduction to user mod unhooking	• Different types of unhooking via software breakpoints • What are software breakpoints and what are hardware breakpoints? • What are the differences, advantages, and disadvantages between them?	13:00 pm – 13:30 pm
Hands-on: User Mode Unhooking and Event Tracing for Windows	• Continue to improve your indirect syscall shellcode loader and implement user-mode unhooking and ETW patching via software breakpoints.	13:30 pm – 15:00 pm
	Break	15:00 pm – 15:30 pm
Summary and Q&A for day 3	• Let us summarize day 3 and ask all your questions about the content and uncertainties of day 3.	15:30 pm – 16:30 pm

11.4 Day 4

Module Name	Module Details	Time plan
Introduction to DLL Sideloading	• What is DLL sideloading and how can it be used to bypass endpoint security products? • What are the differences, advantages, and disadvantages between loading shellcode directly through PE and loading shellcode indirectly through DLL sideloading?	8:30 am – 9:00 am
Hands-on: Shellcode Loader DLL Sideloading	• Learn how to find possible DLL sideloads. • Make the transition from your portable executable shellcode loader to shellcode execution via DLL sideloading	9:00 am – 10:00 am
	Break	10:00 am – 10:15 am
Hands-on: Shellcode Loader DLL Sideloading and Proxying	• Introduction to DLL proxying. • Improve DLL sideloading capabilities by making the transition from DLL sideloading to DLL sideloading/proxying	10:15 am – 11: 00 am
Introduction to Entropy and Metadata	• What is entropy in the context of shellcode loaders, and how does it affect endpoint security evasion? • What might be a useful effect of applying proper metadata to your shellcode in the context of endpoint security evasion?	11:00 am – 11:30 am
Hands-on: Shellcode Loader Entropy and Metadata	• Explore the principles of entropy and how it can be used to increase the stealth and effectiveness of your loader. • Learn about the principles of using useful metadata to increase	11:30 pm – 12:15 pm

	the stealth and effectiveness of your loader.	

	Lunch Break	12:15 pm – 13:00 pm
Hands-on: Shellcode Loader and Certificates	• Learn more about the principles of using useful certificates to increase the stealth and effectiveness of your loader.	13:00 pm – 14:00 pm
Hands-on: Malware/Compiling optimization	• Learn how to enhance or optimize the malware compilation process in Visual Studio as it relates to endpoint security evasion. • Improve your shellcode loader by using Optimized Compilation in Visual Studio	14:00 pm – 15:00 am
	Break	15:00 pm – 15:30 pm
Summary and Q&A for Day 4	• Let us summarize day 4 and ask all your questions about the content and uncertainties of day 4.	15:30 pm – 16:00 pm
Final Q&A	• Your time to clarify doubts, share insights, and get answers to your questions	16:00 pm – 17:00 pm

11.5 Course Content Disclaimer

Please note that the content of this course may be updated and revised. As the course progresses and new knowledge is gained, there may be additions, changes or improvements to the material. Please be assured that any changes will be made to enhance your learning experience and to ensure that the course remains up to date. We ask for your understanding of any changes to the content.

To ensure the exclusivity and interactivity of the course, **no recordings will be made**. We also ask all participants not to make any screen recordings or similar. Your commitment and attendance are therefore essential to maximize the benefits of this exceptional course.

12 Benefits

By attending the **Endpoint Security Insights** course, each student will receive:

- A dynamic **4-day online** training course with interactive learning (70%-80% hands-on)
- Access to a dedicated LAB for each participant with Endpoint Security in place (offline mode)
- Comprehensive slides and detailed playbooks for all modules
- Access to all workshop POCs
- A 60-day window to the dedicated workshop Discord channel
- A certificate of completion to recognize your achievement.

13 Workshop Pricing

Discover the pricing options for the workshop **Endpoint Security Insights**:

- Price: 2499 € (exclusive VAT)
- Limited to 8 seats

14 NDA

Please note that each student is required to sign a **Non-Disclosure Agreement** (NDA) before the course begins. This ensures that the student agrees not to disclose their personal course material to any third party, to use anything they learn only in an ethical context, and not to disclose any data/information about what they have learnt or the course itself to security product vendors.

Detailed information or the explicit NDA document will be sent to all students prior to the course.

15 Frequently Asked Questions (FAQs)

Is it mandatory to send the signed Non-Disclosure Agreement (NDA) to RedOps GmbH before the course starts?

Yes, it is mandatory to send the signed NDA separately for each course participant. Please send it to office@redops.at at least 1 week before the course starts.

Will I receive course materials such as slides and handouts?

When you register for the course, you will receive a watermarked PDF copy of the course materials. This PDF file, which will include your full name and email address, will be sent to you electronically before the course starts. It is strictly forbidden to pass this on to anyone else and will result in you being disqualified from future courses.

Can I contact you before the course starts so that I can prepare myself?

Yes, of course you can. If you have any questions or concerns, please contact office@redops.at. To speed up the process, please mention "Training Preparation" in the subject line of your email. In general, however, there is no need to prepare in advance as we cover all the necessary basics in the course.

What software do I need to take the course?

- Host operating system Windows 10 Professional 64-bit
- Microsoft Remote Desktop Client (to access the host)
- Zoom client (to join the workshop)
- Discord account (for written questions during the workshop)

How do I access the LABs associated with this course?

Access to the labs is via RDP (3389), which can be accessed via the Microsoft Remote Desktop Client from any unrestricted internet connection.

Will the labs be available online after the course?

Please note that the labs are only available for the duration of the course. At the end of the course, the labs will be deactivated. However, you will receive all the workshop POCs and can continue to use them in your own lab in your company or at home.

Can I use my favorite Command and Control (C2) framework for the lab exercises?

In the officially provided LAB environment, only the free version of the Metasploit framework is used. The provided shellcode loader POCs are basically designed to be used with different C2 frameworks. However, you are welcome to use your favourite C2 framework outside the official LAB environment, but please note that I may not be familiar with your chosen C2 framework and will not be able to fully support you during the exercises.

Do students receive a certificate at the end of the course?

All students who complete the course will receive a digital certificate of attendance.

Is there a minimum number of participants for a course to run?

Yes, we reserve the right to cancel the course if there are not enough participants. We will inform you as soon as possible and offer you a full refund or a place on a future course.

16 Copyright

17 Contact

If you or your company have any questions about the course, please do not hesitate to contact me. I will be happy to answer any questions you may have.

Website: https://redops.at/en

E-Mail: office@redops.at