COURSE PROFILE:

Mastering LLM Integration Security: Offensive & Defensive Tactics

2 days | **Intermediate** training

Version 4

claranet cyber security | NotSoSecure Training

Your Course

An immersive, intensive 2-day journey into the dynamic world of artificial intelligence. As LLMs increasingly becoming an integral part of various products and services, grasping their implementation nuances and securing these implementations is paramount for maintaining robust, efficient, and trustworthy systems.

Who is it for?

- **Security Professionals**
- **Back-End / Front-End Developers**
- **System Architects**
- **Product Managers**
- **Anyone directly involved in the integration and application of LLM technologies**

Audience

This course is designed for individuals with a beginner-to-intermediate understanding of artificial intelligence and cybersecurity. Whether you are a security consultant, developer, AI / LLM architect, or prompt engineer, you should have a foundational grasp of AI / LLM concepts and some experience with cybersecurity practices.

Student requirements

- **Basic Understanding of AI:** A foundational knowledge of AI and LLM principles and applications is essential.
- **Familiarity with Programming:** Some experience with coding, particularly in languages commonly used in AI development (e.g., Python), will be beneficial, though advanced proficiency is not required.
- **Understanding of cybersecurity concepts:** A basic understanding of cybersecurity threats and mitigation strategies will be advantageous.
- **Laptop:** AI labs are served in the cloud, access to python IDE is via Jupiter notebooks, the only hardware requirement is access to the latest version of Chrome or Firefox.

What students will be provided with

- **Comprehensive Course Materials:** Detailed handouts, slides, and digital resources covering all key concepts and techniques.
- **Interactive Labs:** Access to structured hands-on labs for practical experience in securing & hacking AI systems.
- **Case Studies:** documented examples of real-world AI breaches and security implementations.
- **Direct Support:** Access to the instructor for post-course questions, clarifications, and additional guidance to ensure you can apply what you have learnt effectively.

What you will learn

This course follows a practical “defense by offense” approach, anchored in real-world scenarios and hands-on labs rather than abstract theory. By the end of the course, you’ll be able to:

- Think and behave like a sophisticated attacker targeting LLM-based systems
- Understand how attackers discover and exploit prompt injections, insecure output handling, data poisoning, and other vulnerabilities in AI workflows
- Identify and exploit security weaknesses specific to LLM integrations
- Practice detecting and attacking common pitfalls (e.g., plugin misconfiguration, overreliance, and supply chain exposures) in real-world lab environments
- Implement effective prompt engineering and defensive measures
- Learn to craft prompts that minimize leakage, prevent injection, and ensure your LLM responds reliably within controlled security parameters
- Design LLM applications with minimal attack surface
- Explore best practices for restricting AI agent functionality (excessive agency), hardening plugin interfaces, and securing AI-driven workflows
- Apply forward-thinking strategies to protect training and inference data
- Develop robust security controls in real-world deployments
- Translate lab exercises into practical solutions by integrating logging, monitoring, and guardrails for continuous protection of LLM-based services

Why it is relevant

The rapid adoption of AI and, specifically, Large Language Models (LLMs), has opened new frontiers in innovation. And in attack surfaces... As companies rush to harness the power of LLMs in applications ranging from customer service to data analytics, they often overlook the emerging security gaps introduced by prompt injection, data poisoning, insecure plugin designs, and more.

Our course directly tackles these new challenges. Over two immersive days, you’ll not only uncover high-impact vulnerabilities that could already be at work within your systems but also learn how to patch them before they result in breaches or critical data leaks. In addition, we regularly update our modules and labs to incorporate the latest security breakthroughs, proof-of-concept exploits, and real-world incidents.

This focus on cutting-edge threats and solutions means that attendees can return year after year for fresh insights, continually refining their ability to secure AI-driven environments as new vulnerabilities emerge.

What is in the syllabus

Note: *Our syllabuses are subject to change based on new vulnerabilities found and exploits released. Claranet's comprehensive 2-day training course is designed to give you an in-depth understanding of the top security threats in Large Language Models (LLMs) and effective strategies to mitigate them. Across both days, we will cover critical topics essential for securing AI applications and combating potential threats.*

MODULES	WHAT YOU WILL LEARN
Prompt Engineering	This module introduces the fundamentals of what prompts are and how they function within the context of AI and LLMs. This module dives into the key aspects of prompt engineering • What makes a good prompt • How to write effective prompts • Including reference text in prompt • Few-Shot prompting • How to give AI time to think • Using Delimiters for Clarity and Security
Prompt Injection	This module covers the security risks associated with prompt injection vulnerabilities, which can lead to unintended behavior or the disclosing of sensitive data and provides strategies to address these issues. Understanding the nuances between direct and indirect prompt injections is vital for recognizing how attackers can exploit these vulnerabilities. By examining real-world examples, we can study the potential impacts and consequences • **Nature of Prompt Injection Vulnerabilities:** Explore how vulnerabilities arise from the manipulation of AI prompts. • **Direct vs. Indirect Injection:** Differentiate the methods attackers use to exploit prompt injection weaknesses. • **Real-World Exploits:** Analyze documented instances to understand the practical risks and execution of such attacks. • **Impact and Consequences:** Assess the potential severity of prompt injection, from misinformation to critical data leaks. • **Defense Strategies:** Learn about the latest techniques for detecting and thwarting prompt injection vulnerabilities.

	LAB ACTIVITIES: • **The Math Professor:** Users will perform direct prompt injection attacks to convince the professor the answer is always correct • **Indirect Prompt Injection:** Users will perform indirect prompt injection attacks via data which if fetched and supplied to the LLM during RAG.
ReACT LLM Agent Prompt Injection	The ReACT framework is designed to enrich Large Language Models (LLMs) with a structured approach to processing and generating tasks. Within this framework, AI agents are given a set of tools and follow a Reasoning-Action-Observation chain to interact with information and environment. However, vulnerabilities may arise from prompt injections, where malicious inputs disrupt normal operations • **Understanding ReACT:** Learn about the ReACT framework and its role in enhancing LLM tasks. • **Tools Purpose in ReACT:** Examine the functionalities of tools provided by the framework for AI agents. • **Tool Abuse in Frameworks:** Review how tools intended for productive use can be misused within frameworks, such as LangChain. • **RAO Chain Exploitation:** Analyze how the Reasoning-Action-Observation sequence can be corrupted through prompt injections and other methods. • **Prevention and Mitigation:** Gain insight into strategies to safeguard the integrity of systems utilizing the ReACT framework and similar structures. LAB ACTIVITIES: • **The Bank of NSS:** An imaginary bank built using LangChain, agents and GPT.3.5-turbo as the LLM. The bank assists users with queries related to balance and has access to rag systems to fetch data from various data stores. Users will perform prompt injection to fetch information from other users accounts.
Insecure Output Handing	This module focuses on the concept of insecure output handling in AI systems, providing a deep dive into the risks and examining the consequences through practical examples. • **Defining Insecure Output Handling:** Get familiar with what insecure output handling is and the risks it poses to AI system integrity. • **Recognizing Vulnerabilities:** Examine real-world scenarios where insecure output handling has led to system vulnerabilities. • **Simulated Attacks:** Participate in practical exercises designed to exploit insecure output handling in three AI applications, demonstrating the process of gaining unauthorized privileges. • **Impact of Weaknesses:** Understand the potential damage that can result from insecurely handled outputs in AI systems. • **Proactive Measures:** Introduce preventive measures and best practices to secure AI outputs against such vulnerabilities.

	LAB ACTIVITIES: • **Report summarization application:** Users will submit documents to be summarized by the application, the response is rendered on the front end. Users will battle with injection payloads inside of documents to try to coerce the LLM to return code which is in turn rendered on the front end. • **Network analysis agent:** Users will utilize the AI agent to perform network analysis on remote hosts, but what if it is possible to execute arbitrary code? • Stock Bot: An AI assistant designed to provide users with company stock market analysis. The agent works with live data which is fetched from external resources. But what if it is possible to fetch from an internal resource?
Training Data Poisoning	This module addresses the concept of training data poisoning, a technique where attackers deliberately manipulate the data that an AI model learns from, with the intent to compromise its performance, integrity or functionality. LAB ACTIVITIES: • **Adversarial Poisoning Attack Lab:** Simulate an attack that feeds misleading input to corrupt the model's learning process. • **Injecting Factual Information Lab:** Practice the technique of altering an LLM's output by injecting incorrect facts into its training dataset.
Supply Chain Vulnerabilities	This module addresses the vulnerabilities associated with the AI / LLM supply chain, examining the points in the supply process that can or might be exploited, and providing real-world examples of such attacks.
Sensitive Information Disclosure	This module covers the concept of Sensitive Information Disclosure within Large Language Models (LLMs). Learners will explore both theoretical concepts and practical risks, enhancing their understanding of how sensitive data can be inadvertently exposed by AI systems. KEY CONCEPTS: • Exploration of how LLMs may unknowingly reveal personal, proprietary, or confidential information embedded within their training data or through their interactions. • Discussion on common scenarios and mechanisms that lead to sensitive information disclosure. LAB ACTIVITIES: • **Incomplete Filtering lab:** Sensitive information is not properly filtered in training data. • **Overfitting / Memorization lab:** Sensitive data is memorized during the LLM training process. • **Misinterpretation:** LLM can misinterpret input and disclose sensitive information

Insecure Plugin Design	This module provides an in-depth look at the critical aspects of plugin design within AI applications, focusing on the security vulnerabilities that can arise. Students will learn about the common design flaws in and how these vulnerabilities might be exploited. LAB ACTIVITIES: • **Insecure tool usages:** Exploit a network analysis tool to archive code execution due to insecure implementation Langchain run method. File System Operations Security Lab: Evaluate an AI agent's capability to perform file system operations with sanitized paths and test the effectiveness of sanitization against exploitative insertions or confusion tactics post-sanitization.
Excessive Agency in LLM-Based Systems	This module covers the concept of excessive agency in LLM systems, this refers to the vulnerability that allows damaging actions to be performed in response to unexpected of ambiguous outputs. This can occur due to hallucinations, prompt injections, malicious plugins, poorly engineered prompts, or a poorly performing model. LAB ACTIVITIES: • **Excessive agency with excessive functionality:** A medical records-based agent designed to provide acute descriptions of diagnosed conditions. But perhaps more features exist? Users will attempt to modify medical records. • **Excessive agency with excessive permissions:** A file management AI agent, designed to read, list and summarize the contents of files, but once again more undocumented features exist. Users will locate the hidden functionality and used this to create, delete and perhaps even execute commands on the host operating system.
Overreliance in LLM's	This module covers the concept of overreliance in LLM systems, learners will get an in depth overview of what overreliance consists of, why it occurs and what legal repercussions can be faced.

What you will get

- **Certificate of completion**
- **30 days lab access post-course completion (with the opportunity to extend)**
- **8 Continuing Professional Education (CPE) credits awarded per day of training fulfilled**
- **Learning pack, including question & answer sheets, setup documents, and command cheat sheets**

Course highlights

What delegates love:

- **Our labs:** probably the biggest selling point for our courses. Not only will you spend most of the course hacking hands-on in a lifelike web environment, but you'll also have 30+ days access to practice your new skills afterwards.
- **Individual access:** you'll have your own infrastructure to play with, enabling you to hack at your own speed.
- **Real-world learning:** where many leading cybersecurity training courses are based on theory, our scenario-led, research-based approach ensures you learn how real threat actors think and act.
- **Specialist-led training:** you'll learn from highly skilled and experienced practicing penetration testers and red teamers.
- **Up-to-date content:** our syllabus remains so relevant, delegates come back year after year for more.
- **Remediations included:** you'll learn how to fix as well as find vulnerabilities.

Outcomes for budget holders

This course is designed to bring your in-house cloud security testing competency up to the industry standard, helping you: **Lower the likelihood of security incidents by identifying weaknesses in your cloud infrastructure**

- **Improve your understanding of the organization's risk posture based on the frequency and severity of weaknesses identified**
- **Improve the organization's approach to access control management**
- **Create a stronger case for securing software development, cloud deployment, and governance practices**
- **Develop a secure cloud roadmap that balances growth and risk**
- **Implement cloud-based attack detection and response tactics**
- **Build a closer relationship between development and security teams**
- **Internally pentest new tools and systems before making an investment**
- **Nurture and retain passionate, highly skilled, and security conscious employees**
- **Demonstrate commitment to security through training, compliance, and change management**
- **Develop the organization's competitive advantage for security-conscious customers**

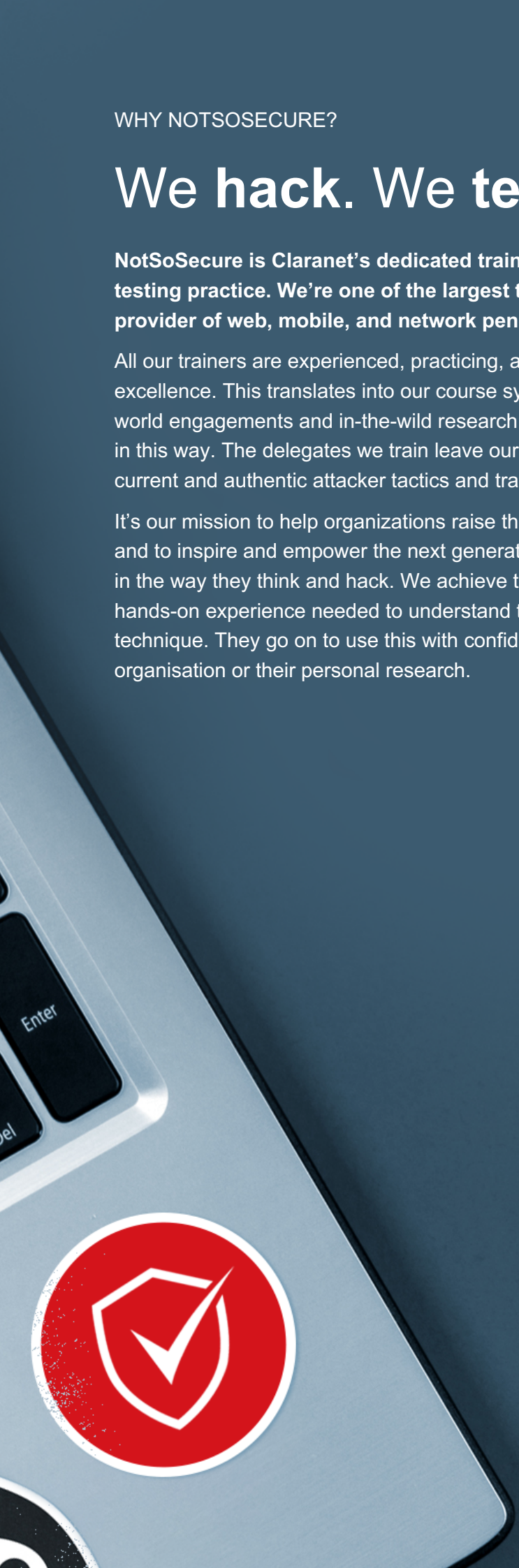
WHY NOTSOSECURE?

We **hack**. We **teach**.

NotSoSecure is Claranet's dedicated training division and part of its global penetration testing practice. We're one of the largest training partners at Black Hat and a respected provider of web, mobile, and network penetration testing.

All our trainers are experienced, practicing, accredited penetration testers with their own field of excellence. This translates into our course syllabuses, where each module is designed around real-world engagements and in-the-wild research. No other provider of cybersecurity training is modelled in this way. The delegates we train leave our courses armed with knowledge and skills based on current and authentic attacker tactics and tradecraft, not theory alone.

It's our mission to help organizations raise the bar when it comes to their cybersecurity, and to inspire and empower the next generation of IT and security professionals to remain relevant in the way they think and hack. We achieve this by delivering practical content, giving delegates the hands-on experience needed to understand the context behind each offensive and defensive technique. They go on to use this with confidence in their own work, be that within an organisation or their personal research.



WE HACK.
WE TEACH.

claranet cyber security