Unlock Your Potential

The purpose of this project is to empower you to self-learn advanced, possibly uncovered in the class, topics in the field of cybersecurity. Below is a list of advanced topics from which you are required to select one to develop.

Instructions:

- Each team can consist of **at most 3 students**.
- All team members are expected to contribute equally to the project.
- Every team member must fully understand every feature and component of the project.
- Each team will present their project in a **15-minute** presentation.
    1. **Practice your presentation** to ensure it fits within the 10-minute time frame.
    2. The remaining **5 minutes** will be reserved for questions.
    3. If your presentation exceeds 15 minutes, it will be **interrupted**.

Web security (3T ✯✯✯):

Solve 6 challenges from the OWASP Juice shop.

Limit: 3 teams.

Instructions:

1. Do NOT repeat any challenges we have covered in the class.
2. Your presentation must explain each vulnerability, the attack process, a demo on exploiting each vulnerability.

Programming (6T):

Select one of the following projects and develop it.

Limit: 6 teams

Suggested projects:

1. **Password Strength Analyzer** ✯✯✯: Create a tool that evaluates the strength of passwords based on various factors, such as length, complexity, entropy, and common patterns (dictionary words, sequential characters, etc.). You could also incorporate recommendations for creating stronger passwords.
2. **Secure Messaging System with End-to-End Encryption** ✯: Build a secure messaging platform that allows users to send encrypted messages to one another. Implement end-to-end encryption (e.g., using RSA, AES, or elliptic curve cryptography) to ensure that only the sender and receiver can read the messages.

3. **Two-Factor Authentication (2FA) System** ✯✯: Design and implement a simulated 2FA system that adds an extra layer of security to user logins. It could integrate with SMS, email, or app-based authentication (like Google Authenticator).
4. **Implement the AES algorithm from scratch.**
5. **Implement Simon & Speck cipher from scratch.**
6. **Color Image Steganography ✯:** Implement color image steganography technique that hides **encrypted** secret data in it. Your program should have some metrics (e.g., PSNR) to evaluate the strength of your technique.

Instructions:

1. Your program should function according to the specifications of the task.
2. Your presentation must explain the idea/theory behind your project.
3. You must address any difficulties or challenges you encountered during the implementation of the program.
4. You must include a demo of your project.

Networking (2T ✯✯)

Limit: 2 teams

1. **Desing and implement a simple VPN:** walkthrough the tutorial here https://seedsecuritylabs.org/Labs_20.04/Networking/VPN/
2. **Setting up a simple firewall**: https://seedsecuritylabs.org/Labs_20.04/Networking/Firewall/

Instructions:

1. You must explain what the tutorial is and what its objectives are.
2. You may NOT be able to repeat the design or setting up steps during the presentation. In this case, it is sufficient to only demonstrate how you implemented it and what tools/frameworks you used.
3. You must give a demo of the VPN/firewall works.

Advanced crypt (4T ✯✯✯✯):

Explore one these advanced cryptography subfields.

Limit: 4 teams, 2 teams per topic.

1. **Fully Homomorphic Encryption (FHE) (2T)**: Explore existing FHE schemes and libraries, and implement one of the following:
    a. A basic image processing tool: Your program should process a small, encrypted image (128x128 or smaller) and apply basic image processing techniques, such as adjusting the brightness and changing the contrast.
    b. A basic calculator that performs additions, subtractions, and multiplications only. Additionally, implement functionality to compute small matrix addition and multiplication (4x4).

2. **Secure Multiparty Computation (MPC) (2T)**: Solve at least two of the open issues related to secure multiparty computations using the Nada framework, available here: here https://github.com/NillionNetwork/nada-by-example/issues
    a. Choose any two open issues that no one else is working on.
    b. You should fork the repo to your profile and work on it.

Instructions:

1. In your presentation, you must explain what the topic is.
2. Explain what you are implementing and which scheme/library you are using.
3. You will need to provide a demo of your project: demonstrate how it works and explain key code segments.

Analysis (1T ✯✯✯):

Write simple malware analysis using YARA. YARA is a framework for malware analysis and detection.

Limit: 1 team.

Instructions:

1. Explain what malware is, including its common types, how they work, and the goal of each malware.
2. Explain what YARA rules are.
3. Demonstrate your analyzer and show how it works.

Passwords/cracking (1T ✯✯✯):

This project includes two tasks:

1. Explain the setup of the PGP system on Linux, how to use it securely manage keys and passwords on your system.
2. Use JohnTheRipper to crack weak passwords and hashes.

Limit: 1 team.

Instructions:

1. Explain what the PGP suite is and what its purpose is.
2. You may not be able to repeat the installation steps; in this case, you should demonstrate how to securely generate and store keys.
3. Explain what John the Ripper is and how it works.
4. Provide a demonstration of cracking a password and a hash.

Do you have another great idea? Discuss it with me.