# ELEC0138: Security and Privacy Coursework 1

## 1. Introduction and Objectives

In today's ever-evolving digital landscape, internet technology continues to advance rapidly. Supermarkets, serving as the primary source for everyday groceries, hold an indispensable position in people's lives. While the adoption of web technologies enhances convenience for customers, it also introduces vulnerabilities that cannot be overlooked, particularly concerning the stability and security of online platforms.

This document endeavors to outline a comprehensive threat model, identifying potential security and privacy breaches associated with malicious attackers, insider threats posed by employees, and the risk of payment information theft from online databases within marketing scenarios.

## 2. Asset Identification

## Customer Data: This includes personal information such as names, addresses, contact details, and potentially sensitive data like payment information (credit card numbers, bank account details), purchase history, and preferences.

## Online Platforms: The digital infrastructure and platforms used by supermarkets to facilitate online shopping. This encompasses websites, mobile applications, and any associated databases or servers.3. Threat Model

### Threat: Malicious Attacks

- Vulnerabilities: Exploiting vulnerabilities in the database access service.

- Likelihood: Low, but significant impact if successful.

- Impact: Loss of service /poisoned data/model, leading to the loss of critical functions, sensitive info. Exposure, leading to degraded reputation.

- Protection: Implement measures such as network isolation, application-layer threat prevention, identity and access management (IAM), and regular security updates to mitigate risks.

### Threat: Employees Misperformance

- Vulnerabilities: Internal database access service vulnerabilities.

- Likelihood: Relatively Low.

- Impact: Loss of service or leakage of sensitive data.

- Protection: Utilize Role-Based Access Control (RBAC), enforce the Least Privilege Principle, and employ data encryption to restrict unauthorized access.

### Threat: PII-contained data exposure during network transporting

- Vulnerabilities: network transportation & encryption vulnerabilities

- Likelihood: Moderate

- Impact: Data breaches, financial loss, reputational damage.

- Protection: Implement robust encryption algorithms, intrusion detection systems (IDS), and stringent access control mechanisms to safeguard payment information.

### Threat: Mis-usage of data … …

## 3. Data Sources and Attacks Set-Up

To find all implementation of the code, please visit <https://github.com/LagrKevin6/ELEC0138_GrpQ_23_24.git>

### Brute force attack implementation

In the scenario of a brute force attack, we encounter a situation where a careless employee has inadvertently set a simplistic password for their account. Perhaps due to oversight or a lack of awareness regarding cybersecurity best practices, the employee opted for a password that falls within easily guessable or common patterns.

Adding to the vulnerability, network traffic is being sniffed, meaning an attacker has gained access to the data transmitted over the network, which isimplemented through packet sniffing tools Wireshark. This grants the attacker knowledge of the unencrypted username associated with the targeted account. This information serves as a crucial starting point for the brute force attack, eliminating the need to separately guess or discover the username. Armed with this knowledge, the attacker can concentrate efforts solely on cracking the password linked to the known username.

Given these circumstances, the brute force attack aims to exploit the weaknesses inherent in the chosen password by systematically testing a range of possible combinations. The attack strategies provided cater to different password composition patterns, allowing the attacker to adapt their approach based on the nature of the target password.

The brute force function offers a versatile approach to cracking passwords, encompassing five distinct strategies tailored to different scenarios. When invoking the `bruteforce` function, users can specify the desired strategy to employ.

The default strategy, known as "common\_pass," leverages a repository of frequently used passwords sourced from GitHub. This strategy capitalizes on users' tendency to select passwords from a limited pool of predictable choices, with these common passwords serving as the initial targets for the brute force attack.

Alternatively, users may opt for the "common\_name" strategy, which focuses on popular first names. By testing against a list of common names, the algorithm aims to exploit human tendencies in password selection, especially those inclined towards personal or familiar terms.

For scenarios where passwords consist solely of numerical digits, the "digit" strategy comes into play. This approach systematically generates all possible combinations of digits within the specified maximum character limit, exhaustively exploring the numeric domain.

Expanding the scope to include lowercase alphabets alongside digits, the "digitAndLowercase" strategy caters to scenarios where passwords may combine letters and numbers. By iterating through permutations of digits and lowercase letters, this strategy broadens the search space while remaining focused on plausible combinations.

In cases where passwords exhibit greater complexity, encompassing a mix of digits, lowercase and uppercase letters, as well as special characters, the "fullCase" strategy offers a comprehensive solution. Here, the algorithm explores the entirety of ASCII characters within the specified character limit, encompassing both common and uncommon combinations.

Each strategy within the brute force attack implementation is designed to cater to specific password composition patterns, providing a flexible and adaptable approach to password cracking. By tailoring the attack methodology to the characteristics of the target passwords, the implementation maximizes efficiency and effectiveness in uncovering vulnerabilities.

Dos attack implementation

To assess and simulate potential attacks, we will utilize various data sources including network traffic logs, system logs, application logs, and data flow diagrams.

Data could be accessed via online databases (Kaggle etc.) or simulated if possible (de-private);

Set up a Linux vm for testing & performing attacks, reusing code snippets given in lectures.

-tcpreplay, pcap(s), python files (for later processing)

Possible attacks as listed:

Ddos (tcp,dos, http…)

Man-in-the-middle

## 4. Assess Impact and Prioritize Threats

### Assessment Criteria:

- Impact: Evaluate the severity of each threat based on potential consequences such as data loss, financial loss, legal implications, and reputational damage.

- Likelihood: Assess the probability of each threat materializing based on historical data, industry trends, and internal vulnerabilities.

- Feasibility: Consider the technical feasibility and resources required for attackers to exploit vulnerabilities.

- Mitigation Cost: Estimate the cost associated with implementing security measures to mitigate each threat.

### Prioritization:

1. Malicious Attacker: High impact, low likelihood, but critical to mitigate due to potential loss of service.

2. PII-contained data exposure during network transporting: relatively high impact, moderate likelihood, remedies may take time (update issues)

3. Employees: Low impact, low likelihood, but essential to prevent insider threats and data leakage.

By prioritizing threats based on impact, likelihood, and feasibility, organizations can allocate resources effectively to mitigate risks and enhance security posture. Regular risk assessments and proactive security measures are crucial to adapt to evolving threats and protect sensitive data effectively.