Arvest Bank Cyber Security Tabletop Exercise Proposal

Presented By: Zachary Chapman, Jacob Smith, Matthew Mclendon, Ashwin Prakash, and Rudy Ramirez

Problem

- Today Arvest is faced with the potential of cyber attacks
- Arvest is a bank that has sensitive information in their systems
  - The information could be used for malicious monetary gain
- Need cyber exercises to assit them in attack prevention and response scenarios
  - Want to make sure the consumer does not suspect a breach
  - They will be more confident when something does arise

Objective

- Create a web application that can be used by Arvest Bank for their team to generate cyber exercises to work on and complete
- The web application will be created with knowledge provided by Arvest Bank
- The web application will focus on the following categories: threat/incidents, targets/assets, vulnerabilities, impacts, and injects/persistence
- End goal is for the application to randomly generate scenarios based on the categories

Key Concepts

- **Exercises**
  - Companies have teams to fix problems that occur with websites, databases, etc.
  - These teams need exercises to work through to prepare for future problems
  - These exercises maintain skills and improve experience and preparation
- **Threats/Incidents**
  - Circumstances that may cause loss or harm to a system
  - Cyberattacks use these to exploit vulnerabilities
    - Not checking user input (SQL injection attacks)
    - Storing passwords using an unsecure format/ within a dangerous environment
- **Targets/Assets**
  - What proper users are defending against malicious users
  - Classified or personal information such as a person's addresses, passwords, social security number, etc.

Key Concepts

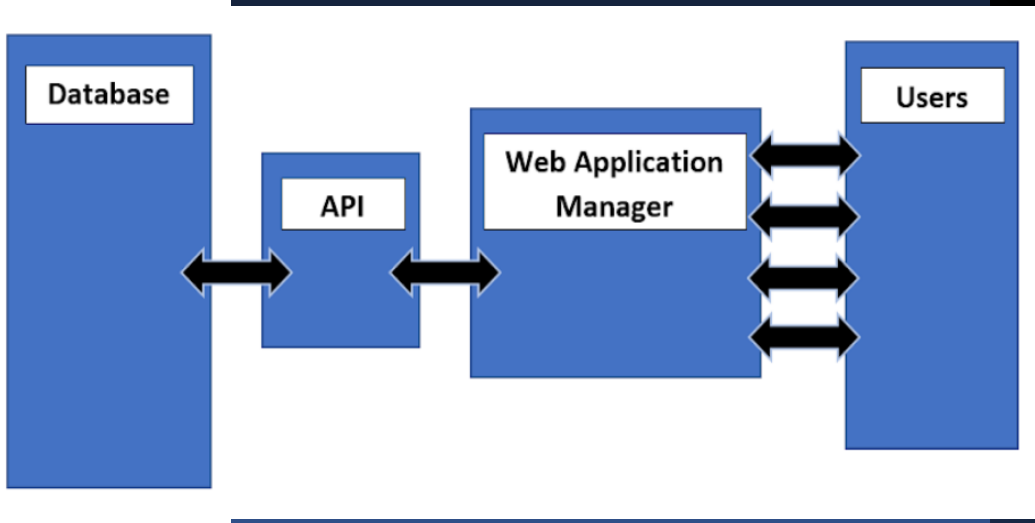
- **Vulnerabilities**
  - Weaknesses in a system that can be exploited
  - How malicious users gain access to a system
  - Usually occur when a user does something with an application that isn't its intended purpose
- **Impact**
  - Day-to-day issues an attack would affect
  - 5 impacts (for us): legal/compliance, reputational, financial, health and safety, and operational
- **Injects/Persistence**
  - Real-world events that can affect how a cyber-attack occurs
    - For example – a worm. Even if it isn't harming anything now, it still needs to be removed to preserve customer relations, etc.

Requirements and Design Goals

- Create an application that will generate a random cyber exercise for the cyber response team at Arvest Bank.
- Need to have different user roles.
  - One to generate exercises
  - One to manage the application and its users
- The goal of this project is the make the Application function and implement some security measures to make the integrity of the system stronger.
  - Security functions are a goal for us as it will help us learn and add more depth to the project however due to it being for internal use it's not a requirement.

High-Level Architecture

- 7 Database Tables
  - User information, User history, Threats, Targets, Vulnerabilities, Impacts, and Injects
- The Database will be built using SQL and mySQL.
- The API will be written in Python. This is where the program will create SQL queries that it can send to the database to either gain or update information.
- The web application manager will be written in both PHP and Python and will communicate with the API to show database information in a user-friendly manner.
- The web application layer will also take care of the security aspects of the application.
- Users will have 2 different roles either database manager, or exercise creator.



Risk

- **User access control**
  - Implement a role-based access control system to protect the integrity and confidentiality of the systems.
- **SQL Injection Attacks**
  - Filter user input to reduce the ability to perform SQL injection attacks on the database.
- **Account Security**
  - Require a username and password from the user before entering the application.
  - Make sure all passwords are encrypted and stored in the database.
- **Confidentiality**
  - Require a valid user log-in to access the site to keep records from Arvest Bank safe.

Tasks

- Create an exercise observation report.
- Create a Final Proposal Document and share it with the Arvest Bank Team.
- Design, Implement, and Test database schema with options for the five categories: Actor, Vulnerability, Target, Consequences, and Impact.
- Design, Implement, and Test Database (DB) API
  - Get a random attribute from the five aspects of a cyber exercise.
  - Update information in the database.
  - Implement security aspects like user credentials and roles.

Tasks

- Design, Implement, and Test a Web Application Manager that retrieves specific information from each of the five categories in the database.
  - Show the random attribute that was obtained.
  - Get user information and pass it to the DB API.
  - Check user login information will database and encrypt all passwords.
- Merge the DB API with the Web Application Manager
- Test final product
- Create a Final Report Document that describes the completed design by giving details on the architecture, technologies involved, interface design, and software implementation.