University of Arkansas – CSCE Department
Capstone I – Final Proposal – Fall 2021

Arvest Cyber Exercise

Zachary Chapman, Jacob Smith, Matthew Mclendon, Ashwin Prakash, Rudy Ramirez

Abstract

For this project, Arvest is wanting a web application that will use a database to generate cyber exercises. Cyber exercises are a great way to keep their cyber defense team up to date with the current attacks that are happening and with refreshers of older attacks that are still used today. A cyber exercise is broken down into five parts that are the threat/incident, the target/assets, the vulnerabilities, the impacts, and the injections/persistence. All are crucial to the exercise and to make sure their team knows how to handle difficult situations when they appear.

Skills are something that you must practice over and over again to truly master and that includes defensive skills. So running different exercises with different scenarios that have you utilize the same skill set is a great way to master these skills. This is our goal with this project to create a website that will generate different scenarios for Arvest to use in order to perfect their cyber defense skills. This way they will be prepared to face any threat they might come across.

1.0 Problem

Today, Arvest Bank is faced with the potential of cyber-attacks that would affect day-to-day operations for their business. Since they are a bank, they have sensitive information and personal information about their users that they need to protect. Banks are very susceptible to cyber-attacks because of the information they hold. The information can be used by malicious users for monetary gain. To better prepare themselves, they practice cyber exercises routinely so they can understand how to respond in the best way possible. When the company knows what to look for and how to respond they can fix a problem internally before their consumers even know about it.

Making sure that Arvest has experience with solving problems related to cyber-attacks that they might encounter is the best way to prepare them for true cyber-attacks from malicious users. This will give them more confidence to deal with problems when they arise and the ability to deal with them quickly. It is important for Arvest to be knowledgeable in how to counter these attacks to keep their customer's information safe. It is also important to be knowledgeable about how to

delay and defend against these cyber-attacks to make sure day-to-day operations don't get affected and users don't begin to worry about the integrity of Arvest.

2.0 Objective

The objective of this project is to create a web application that can be used by Arvest bank to generate cyber exercises for their team to work on and complete. This will be some application that Arvest can use to generate a cyber exercise for their cyber response team to work on in order to better develop their response skills to a cyber-attack. This application will be created using the knowledge that Arvest Bank has provided and will focus on the following categories: threat/incidents, targets/assets, vulnerabilities, impacts, and injects/persistence. The end goal is for the application to automatically generate these exercises by pulling one of each category and combining them into a scenario for the cyber response team to react to.

3.0 Background

3.1 Key Concepts

Companies usually have teams of people that are responsible for fixing the problems that occur with their website, database, or applications. These people are not always working on a problem, so they need exercises to do when the workflow is slow. These exercises are a great way to keep their skills sharp as it flexes their brain over topics that they maybe haven't used in a long time or possibly don't have much experience with. This makes sure that they are prepared for any and all possible issues that arise from the website, database, or application.

A threat/incident is a set of circumstances that has the potential to cause loss or harm to a system. This is where cyberattacks start. The threat or incident is used to exploit a vulnerability in the system. Some examples would be not checking user input to secure the database from SQL injection attacks or storing passwords within a risky format/environment. Threats are the baseline of a cyberattack and where they start.

The target/assets are what we are trying to defend from malicious users. Usually, this is important or sensitive information such as classified documents or personal information. Personal information specifically is important due to it being one of the most common targets of a cyber attack. Personal information could include a personnel's address, password, or even their social security number; all things a malicious user could use against you.

Vulnerability is a weakness in the system that can be exploited. This is how malicious users gain access to a system. This is one of the more difficult parts of a system to design because it is hard to recognize problems in a system that you design. Problems usually occur from a user doing something with the application that isn't its intended purpose. That is why it is always best to have someone test your applications because they might try to perform actions in the wrong order or might try to do something that you as a developer didn't think of. The end goal is to design a system with no vulnerabilities but this is only something we can strive to achieve because it is impossible to make a perfect system.

Impacts are the day-to-day issues that the attack affects. In our case, we will have 5 impacts with those being legal/compliance, reputational, financial, health and safety, and operational. These are all things that could be hurt for the duration of the attack or something that could outlive the length of the attack. A damaged reputation is something that takes months if not years to fix.

If a user can't trust you with their private information then they won't use your company. Varonis found that less than 50% of Americans are likely to go back to a company after a data breach. Losing more than half of your customers is going to severely hurt your company so reputation is a huge impact.

Injects/persistence are real-world events that could affect how a cyber attack is played out. These events are things that may or may not directly affect the application but will definitely affect how the exercise is handled. For example, if a worm gets in the system, but it isn't proving to be doing anything extremely harmful yet, then you still need to work to remove the worm because if it got out to the public that a worm has been dormant in a company's system, it could hurt their customer relations, since some people may feel that the company is not trustworthy anymore.

3.2 Related Work

Prior to our involvement, members of Arvest's security team would develop and present individual exercises tailored to specific scenarios to better their security and inform other employees about how to handle certain security risks. One such exercise is known as the Garmin exercise. In 2020 the smartwatch company Garmin experienced a ransomware attack for a period of five days. Arvest's exercise for this attack included identifying the ransomware used, how the hackers specifically were able to attack them, what Arvest policies were in place at the time to prevent a similar attack against Arvest, and identifying what actions they could take to improve the security of their company.

Another related work to note is the tabletop exercise packages created by the Cybersecurity and Infrastructure Security Agency. These packages are resources that help stakeholders or other faculty in the company conduct their own security exercises. This would allow them to create discussion amongst others within the organization about various cyber-attack scenarios and how to implement a real solution into their company. The great thing about these packages is that they are customizable. This implies that there exists over one-hundred tabletop exercises to choose from that will allow other faculty to find a package that most closely matches their needs. These packages include a variety of interesting features ranging from multiple scenarios to discussion questions. These scenarios include multiple physical and cybersecurity topics, such as civil disturbances, pandemics, industrial control systems, election security, ransomware, and vehicle ramming. A final thing to note about these packages is that they have questions that will be used to discuss pre-incident info, intelligence sharing, incident response, and post-incident recovery.

The goal of our involvement with Arvest is to develop a generator tool to create such exercises. This will improve the efficiency of creating such exercises by allowing Arvest to use previously created software to develop the exercises rather than constructing them by hand essentially from scratch.

4.0 Design

4.1 Requirements and/or Use Cases and/or Design Goals

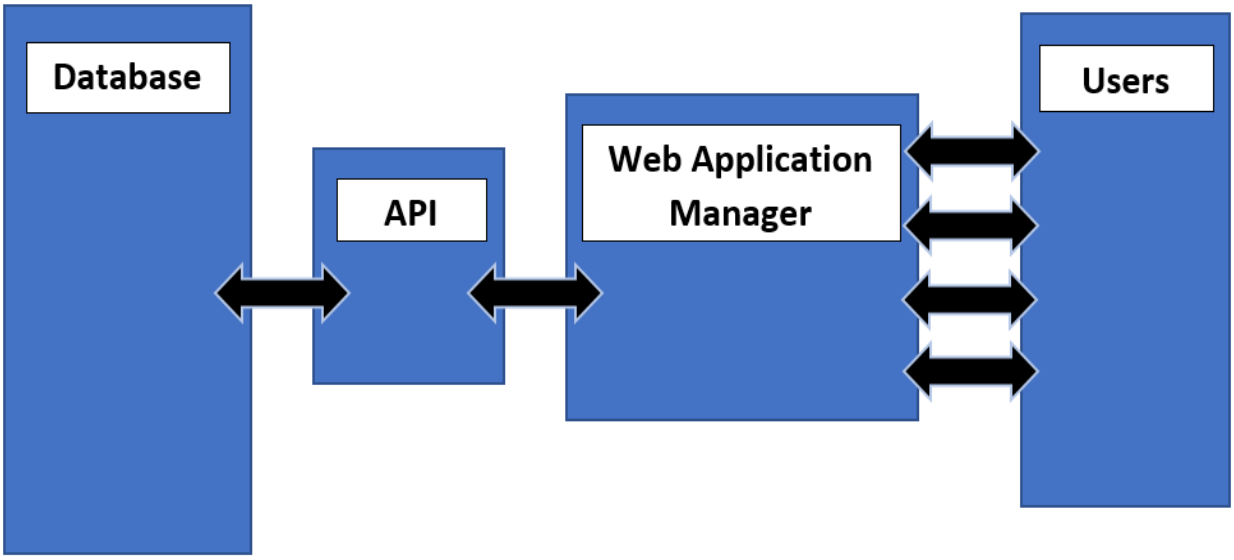
Our requirements for this project are to create an application that will generate a random cyber exercise for the cyber response team at Arvest Bank to respond to. This application will need to have user accounts to differentiate between an exercise creator and application manager. The exercise creator should be able to log in and randomly generate an exercise that can then be given to the cyber response team. The application manager will have the ability to add new options to

each category, so if more threats or vulnerabilities need to be added it can be done from the application.

This application is going to only be used by Arvest Bank and the scenarios it will generate may range from broad cyber-attacks that can happen to any company to more specific ones that can only happen to Arvest Bank. The scenarios will be created from the following categories: Actor, Vulnerability, Target, Consequences, and Impact. This will make it more specific and be more useful to Arvest Bank.

Our goal for this project is to make the application function as Arvest Bank requires and hit all of its requirements along with making the application security to make sure that no one can tamper with the exercises once they are generated to keep the integrity of the exercise

4.2 High-Level Architecture



The database schema will be composed of 7 tables one for user information, one for user's exercise history, and the other five for threat/incidents, target/assets, vulnerabilities, impacts, and injects/persistence. These tables will be created using SQL and a mySQL database. This database won't be a relational database as all tables will be separated and combined in the application layer.

The goal is to create python programs that will pull a random attribute from each table of the database and combine it all together to create the scenario. This will be done by creating the API that will be python programs that will communicate with the database and handle all query generation and SQL function calls. This will be where we would have functions that could get a random attribute or functions that will generate SQL queries that will update the database to add new information.

Then these python programs can interact with PHP programs and other python programs that will handle the web application and the combination of all attributes gathered from the API. This will be the code of the website itself and where we will be getting the information from the user. This will also be the layer with security aspects like user verification. We will make sure the users have entered the correct login information and that they are given the proper web pages

that correspond to their roles. This is to make sure that exercise creators and database managers are given different web pages so that they can fulfill their objectives with the application.

The users will need to request access by getting a database manager to add them to the system. The database manager will be able to add new users into the system and edit the database. This is to allow database managers to add new threats, targets, vulnerabilities, and injects as they see fit. This is an important feature because cyber-attacks keep adapting and Arvest Bank's cyber response team needs to be well equipped for all types of cyber-attacks both new and old.

If a user is not a database manager then they are an exercise creator. Exercise creators will be able to go into the application and generate new cyber exercises and it will save it to make sure that it will always generate a new exercise each time. This is important because we want to have Arvest Bank's cyber response team be well-rounded and not just specialized in one area of cyber defense.

4.3 Risks

Risk	Risk Reduction
User Access Control	Implement an access control system to protect the integrity and confidentiality of the system
SQL injection attacks	Filter user input to reduce the ability to perform SQL injection attacks on the database.
Account Security	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.

4.4 Tasks

1. Create an exercise observation report.
2. Create a Final Proposal Document and share it with the Arvest Bank Team.
3. Design a database schema with options for the five categories: Actor, Vulnerability, Target, Consequences, and Impact.
4. Implement the API for the database.
5. Test the API for the database to make sure that it works correctly.
6. Design a Web Application Manager to implement the specific information for each of the five categories in the database.
7. Test the Web Application Manager
8. Create a Final Report Document that describes the completed design by giving details on the architecture, technologies involved, interface design, and software implementation.

4.5 Schedule

Week Number	Dates	Active Tasks	Members Involved
Week 0.1	Nov. 14 - Nov. 20	● Exercise Observation Report	● All Members
Week 0.2	Nov. 21 - Nov. 27	● Create Final Proposal	● All Members
Week 0.3	Nov. 28 - Dec. 4	● Edit/Present Final Proposal	● All Members
Week 0.4	Dec. 5 - Dec. 11	● Edit/Present Final Proposal	● All Members
Week 0.5	Dec. 12 - Dec. 18	● Share Final Proposal with Arvest Team	● All Members
Winter Break	Dec. 19 - Jan. 15	● Brainstorm Database Schema ● Brainstorm DB API ● Brainstorm Web Application Manager	● Zachary C., ● Ashwin P., Rudy R. ● Jacob S., Matthew M.
Week 1	Jan. 16 - Jan. 22	● Create/Design Database Schema ● Capstone team project Website	● Zachary C., Ashwin P., Jacob S. ● Matthew M., Rudy R.
Week 2	Jan 23 - Jan. 29	● Build/Test Database Schema ● Finish up team website	● Zachary C., Ashwin P., Jacob S. ● Matthew M., Rudy R.

Week 3	Jan. 30 - Feb. 5	● Create DB API ● Create Web Application Manager ○ To show the random attribute that was obtained	● Zachary C., Ashwin P., Rudy R. ● Matthew M.
Week 4	Feb. 6 - Feb. 12	● Implement/Test DB API ○ To get a random attribute from the five aspects of a cyber exercise ● Implement/Test Web Application Manager	● Zachary C., Ashwin P., Rudy R. ● Jacob S., Matthew M.
Week 5	Feb. 13 - Feb. 19	● Implement/Test DB API ○ To get a random attribute from the five aspects of a cyber exercise ● Implement/Test Web Application Manager ○ To show the random attribute that was obtained	● Zachary C., Ashwin P., Rudy R. ● Jacob S., Matthew M.
Week 6	Feb. 20 - Feb. 26	● Implement/Test DB API ○ To update information in the database ● Implement/Test Web Application Manager ○ To get user information and pass it to the DB API	● Zachary C., Ashwin P., Rudy R. ● Jacob S., Matthew M.
Week 7	Feb. 27 - Mar. 5	● Implement/Test DB API	● Zachary C., Ashwin P. Rudy R.,

		○ Security aspects like user credentials and roles ● Implement/Test Web Application Manager ○ Check user login information will database and encrypt all passwords	● Jacob S., Matthew M.
Week 8	Mar. 6 - Mar. 12	● Implement/Test DB API ○ Security aspects like user credentials and roles ● Implement/Test Web Application Manager ○ Check user login information will database and encrypt all passwords	● Zachary C., Ashwin P., Rudy R. ● Jacob S., Matthew M.
Week 9	Mar. 13 - Mar. 19	● Merge the DB API with the Web Application Manager	● All Members
Week 10	Mar. 20 - Mar. 26	● Merge the DB API with the Web Application Manager	● All Members
Week 11	Mar. 27 - Apr. 2	● Merge the DB API with the Web Application Manager	● All Members
Week 12	Apr. 3 - Apr. 9	● Test the final product for any bugs	● All Members
Week 13	Apr. 10 - Apr. 16	● Create Final Report	● All Members
Week 14	Apr. 17 - Apr. 23	● Edit Final Report	● All Members
Week 15	Apr. 24 - Apr. 30	● Edit Final Report	● All Members
Week 16	May 1 - May 5	● Submit Final Report	● All Members

4.6 Deliverables

- Exercise Observation Report
- Final Proposal Document: The final proposal document will provide an overview of the organization, project objective, potential impact of the project, topic difficulty, related work, approach, tasks, schedule, high-level design, and how well we understood the material.
- Database schema: The database schema is for containing the data for the five attributes: Actor, Vulnerability, Target, Consequences, and Impact.
- Web Application: The web application is for generating a scenario based on the options for the five categories in the database.
- Final Report Document: The final report document will give details on the completed design. This will include a lot of details, which will be descriptions on the high-level architecture, the interface, the implementation of the software, which includes the database, API, and web application, the lessons learned, and future impacts. This final report will also include documentation on how to use the tool. We want to include some tutorial for new users to understand how the tool works.

5.0 Key Personnel

Jacob Smith - Smith is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Programming Paradigms, Database Management Systems, Computer Networks, and Software Engineering. He is responsible for completing the Garmin exercise observation report and the final proposal deliverable. Other tasks will be delegated by December 2021.

Zachary Chapman - Chapman is a senior Computer Science and Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Database Management Systems, Programming Paradigms, and Software Engineering. He is responsible for completing the Garmin exercise observation report and the final proposal deliverable. Other tasks will be delegated by December 2021.

Ashwin Prakash - Prakash is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Programming Paradigms, Software Engineering, Database Management Systems, and Computer Networks. He is responsible for completing the Garmin exercise observation report and the final proposal deliverable. Other tasks will be delegated by December 2021.

Rudy Ramirez - Ramirez is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Programming Paradigms and Software Engineering. He is responsible for completing the Garmin exercise observation report and the final proposal deliverable. Other tasks will be delegated by December 2021.

Robert Mclendon - Mclendon is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Programming Paradigms and Software Engineering. He is responsible for completing the Garmin exercise observation report and the final proposal deliverable. Other tasks will be delegated by December 2021.

Beth Rye is the Security Awareness and Education Coordinator at Arvest Bank. She has been working with Arvest Bank for a little over 15 years. She is the team lead on the Arvest side and leads all meetings.

Hayley Clark is a Business Analyst at Arvest Bank. She has been working with Arvest Bank for 2 years now. She will be our main point of contact for this project.

Juan Osorio Alonso is the IT Manager at Arvest Bank. He will be answering most of our technical questions related to the project. He has been working for Arvest Bank for almost 3 years and worked at Walmart prior.

Kourtney Connel is the IT Director at Arvest Bank. She will be helping on the technical side of this project when she can. She has been working with Arvest Bank for the last 2 years and worked at Walmart prior.

Tina Owens is a Business Continuity Analyst at Arvest Bank. She will be sitting in on the meetings and answering any questions she can. She has been with Arvest Bank for a little over 5 years.

6.0 Facilities and Equipment

For this project, we will require access to a server like Turing for testing purposes and keeping the web application running. The students will only need computers and the ability to access Turing. At this time, no other facilities or equipment will be necessary.

7.0 References

[1] Varonis, https://www.varonis.com/blog/company-reputation-after-a-data-breach/

[2] Cybersecurity & Infrastructure Security Agency, https://www.cisa.gov/cisa-tabletop-exercises-packages