“Gamification” of the Cyber Tabletop

SEACAT Team
Saxman One

August 16th, 2024

Science & Engineering Apprenticeship Program

Abstract

This paper reports on the SEACAT project conducted by the SEAP interns during the course of the summer internship. During their 8-week internship at Naval Base Keyport, they were assigned a 7-week project revolving around the gamification of the Cyber Tabletop exercise. The CTT is described by the DoD as an intellectual war-like game; however, they discovered several opportunities to introduce further game mechanics. This paper describes the development process, mechanics, and future direction of their product.

Background

The project described in this document revolves highly around the concept and purpose of the Cyber Tabletop exercise. For simplicity, the Cyber Tabletop will be referred to as the "CTT" throughout this paper. The CTT exercise is a mission-based cyber risk assessment that can be utilized to test the efficacy of a system's cyber defenses in preventing potential threats and attacks [1]. This exercise consists of a roughly 3-18 month completion timeline, divided into 4 separate sections: exercise preparation, exercise execution, post-exercise analysis, and reporting [2]. During the preparation period, all participants will be divided into three groups: the control team, the operational team, and the OPFOR team. The control team possesses the authority and responsibility for the exercise and provides administrative support throughout the CTT. The operational team are the defenders of the system and are expected to have a thorough understanding of the defenses being tested. The OPFOR team are the adversaries attempting to breach the system and must have a meticulous understanding of cyber warfare. After separating into teams, all preliminary preparation—such as the creation of a mission timeline—will take place before the exercise execution. Following the preparations, the exercise execution will commence, in which the mission timeline will be initiated and the OPFOR team will trial various potential attacks against the operational mission. Data will be collected throughout this period and will be used for the post-exercise analysis. During the post-exercise analysis, several factors will be evaluated to determine the risk of a cyber attack, for example, the likelihood and impact assessments. As a result of this analysis stage, cyber defenders are able to further their understanding of a system's defenses and form improvements where they are found necessary. The final stage of the CTT is the reporting of the information gathered under the technical and executive brief, thus concluding the exercise.

Introduction

As described in the "DoD Cyber TableTop Guide," the CTT is an intellectual wargame-like exercise [2]. However, upon closer inspection of the mechanics of the CTT, we—the SEAP interns—noticed various opportunities for the further gamification of the exercise. For instance, there were multiple occasions where randomization could have been supplemented into the CTT without compromising the integrity of the exercise analysis.

Therefore, over the-8 week course of our internship, we were assigned the gamification of the CTT under the project name Science Engineering Apprenticeship Cybersecurity Augmented Tabletop (SEACAT). Our efforts were set to fully gamify the CTT and create a resulting product with fully fleshed game mechanics, visual designs, and a digital user interface that can be provided as an alternative to the CTT while maintaining the integrity of the exercise. This document will describe the process we underwent during the production of this project, the result, and the future direction of the product.

Development Process

The SEACAT project had a roughly 7-week completion timeline in which our team had to propose, draft, revise, and publish our product. To effectively accomplish our objective within the given timeframe, the scrum organization methodology was applied. Scrum is a method of team organization and scheduling that maximizes the productivity of a team by focusing on specific objectives each week named "sprints." Therefore, scrum allowed our team to optimize our limited time frame efficiently to accomplish all objectives that needed to be completed. To describe a brief overview of our sprints: week 1, basic planning; week 2, project proposal; weeks 3, 4, 5, & 6 draft; week 7, project revision and report. In addition to effective planning, the distribution of human resources was also a crucial component in the success of this project. The SEACAT project had certain components that needed to be met for the product to be successful. To ensure that these project requirements were met, we found the most success by dividing our project group into separate teams to have deeper engagement on each factor that needed to be developed. These teams consisted of the game development team, whose focal point were the concepts and mechanics of the game; the visual design team, whose focus was on the creation of the graphics for the GUI and other design elements; and the software development team, whose main focus was the construction of a functioning computer software. As stated, this allowed us to have a deeper focus on certain aspects of the project, but nonetheless, a significant consequence of dividing into separate teams was the reduced communication between project members, which could lead to an end product that fails to reflect ideas from all members. To counteract this, we depended heavily on the project manager to guarantee that all ideas were clearly communicated between groups to create an end product that was consistent with what all teams envisioned. For example, during the beginning of the project, the game development team was making frequent changes to the concept of the project, which was crucial for the other groups to be informed about. More so than often, the project manager was communicating to keep the other groups aware of important changes.

Game Development

As specified prior, the game development team focused on the creation of the concept of the game and game mechanics, but most importantly, set the foundation for what all other groups will reference for this project. Due to this, we decided to accelerate the pace of our development to ensure that all other teams will have the maximum

amount of time to work on their components. Additionally, the task of creating game mechanics required far less technical work in comparison to the other groups, hence requiring far less time to complete. Consequently, the efforts of the game development team were completed relatively early, finishing around 3 weeks into the project, aside from testing and revisions. However, despite the expedited pace of our team, we continued to emphasize and maintain a schedule to ensure the efficiency of our team. To suit this compact timespan, we further distinguished our schedule by allocating our time based on a day-to-day schedule rather than by week. The first 1-3 days of the project primarily consisted of gaining a thorough understanding of the CTT, searching for inspiration, and creating a basic plan. This included creating a methodology for how we were going to create the game, which we decided would strongly rely upon the mechanics of the Cyber Tabletop, such as the concepts of an operating team or player input, to accomplish the analysis. Additionally, we also began looking at various mechanics from games such as DnD or Magic the Gathering to determine how the CTT can be gamified. Nevertheless, one major setback that will be relevant for the entirety of this project is the lack of experience or examples with the real Cyber Tabletop exercise due to issues with the classification of documents. Following the planning period, our group moved onto drafting a concept of how the game would work. The general notion behind our idea was to modify the CTT in a way that allowed for a turn-based system and included some randomization without risking the accuracy of the system analysis. This process took only 1 day, yet the rest of our 3-week timeline would be spent bringing this concept to paper and expanding on different concepts of the game. One of the most significant changes we brought to the CTT game were changes to how the analysis would be conducted during the game. As mentioned earlier, the analysis portion of the CTT is considered to be the most important outcome of the exercise, thus ensuring that players of the game will be able to effectively get an accurate outcome during the use of game mechanics was crucial. The CTT analysis has three different risk assessments that are used to determine the overall risk probability of an attack: Likelihood Assessment (Figure 1), Impact Assessment, and Risk Assessment. Regardless, we decided to gamify only the likelihood assessment, as it relied more heavily on quantitative factors, and gamifying the entirety of the analysis process would compromise the accuracy of the overall risk probability. As shown in Figure 1, the likelihood assessment is made up of two factors:

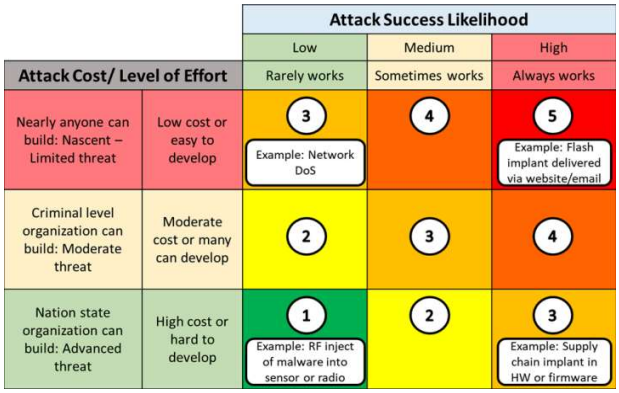


Figure 1. [2] CTT Likelihood Assessment

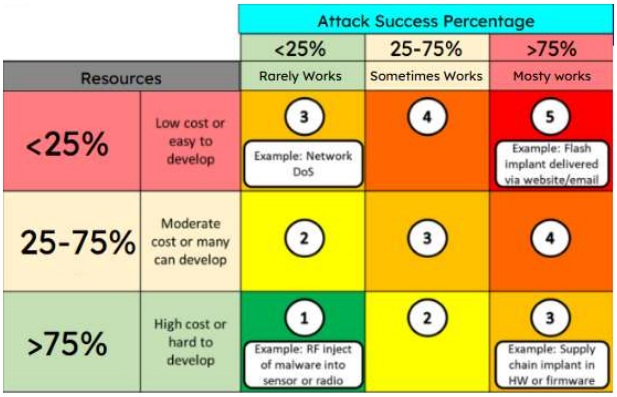
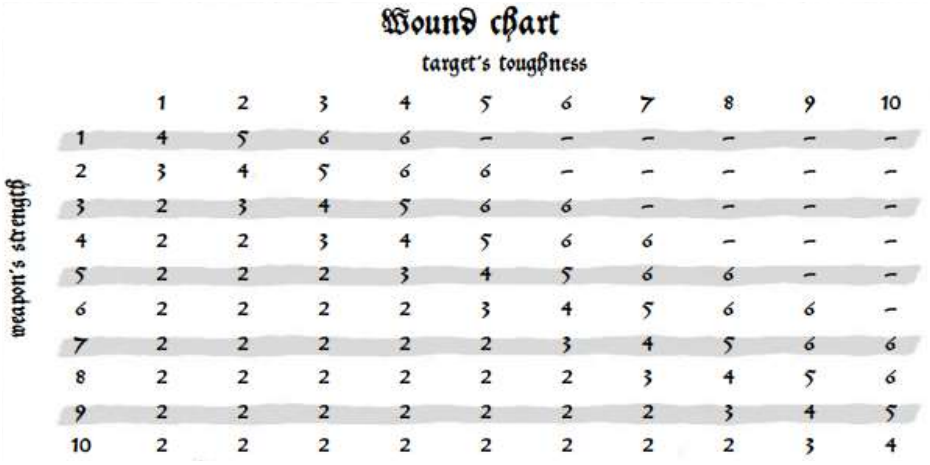


Figure 2. SEACAT Likelihood Assessment Chart

the attack success likelihood and the attack cost. We decided to take these two concepts and assign further numerical values to gamify these aspects. Beginning with the attack success likelihood, this component of the CTT is a determination of how successful a certain cyber attack is against a system. However, during the CTT, the outcome of the attack is heavily dependent on discussion by the people taking part in the exercise. Since the discussion can both provide justification and bias towards a certain success rate, we decided this would be a component that can easily implement a randomization factor. Rather than players deciding the attack success likelihood, they will decide all variables that can affect the success of an attack, such as the maturity of a system, and these factors will all play negatively or positively into the randomization factor. The randomization was also a highly debated decision within our group, as we had to balance both variability and keeping it grounded in reality. One of the most complicated systems was a similar randomization process to that of a "Wound Chart" from Warhammer Mordheim (Figure 3). To briefly describe this method, the strength of an attack and the maturity of a defense system would be evaluated in a staggered chart to output a value of 2-6. Depending on what the output value is, the attack success percentage will be calculated based on the statistics of how likely a 6-sided die can roll higher than or equal to the output value, and the success of an attack will be determined with an actual dice roll. It is important to distinguish that within the SEACAT game, there is a difference between attack success percentage—which is used for the analysis—and attack success—which is the actual determination of the success of an attack during the game. This method of randomization provided the opportunity for players to continue to discuss and account for variables and



Wound chart

target's toughness

weapon's strength	1	2	3	4	5	6	7	8	9	10
1	4	5	6	6	–	–	–	–	–	–
2	3	4	5	6	6	–	–	–	–	–
3	2	3	4	5	6	6	–	–	–	–
4	2	2	3	4	5	6	6	–	–	–
5	2	2	2	3	4	5	6	6	–	–
6	2	2	2	2	3	4	5	6	6	–
7	2	2	2	2	2	3	4	5	6	6
8	2	2	2	2	2	2	3	4	5	6
9	2	2	2	2	2	2	2	3	4	5
10	2	2	2	2	2	2	2	2	3	4

Figure 3. [3] Mordheim Wound Chart

add a randomization factor. Granted, this method had several drawbacks, the main disadvantages being complexity and variability. The complexity of a chart like this could discourage participants of SEACAT from editing the chart. The variability issue comes from the large change if the number required to get through increases by one; for example, 3+ has a 16.66% higher chance than 4+ of succeeding. Consequently, this led us to opt for an alternate method. A system using a 20-sided die was chosen as it gave an even distribution of percentage chance and a low enough change when an armor (maturity) value was decreased or increased. This method functioned similarly to the previous one; however, a change of 1 armor would result in a 5% chance change, with the values of attacker strength and defense maturity being on a scale of 1-20. The attack success would be determined by first taking the maturity value (after all variables are accounted for) and then determining the strength of the given attack (again after all variables are accounted for) in the form of the percentage. This percentage will then be used to find the base value of the attack. For example, a defense has a maximum maturity of 3, which will be an armor value of 12, and defender variables bring down this value by 16%, resulting in an armor value of roughly 10. An attack has a strength of 50% against the defense, giving it a base value of 5. Then, the 20-sided die will be rolled to add the factor of randomization. The attack is successful if the [base value + 20-sided die] is higher than the maturity level. The attack success percentage is then calculated by determining how statistically likely it is for the attack value to be higher than the maturity. In the previous example, the attacks would have to roll higher than a 5 on a 20-sided die, making the probability of success 75%. Overall, this simplification was made because any lower of a change could potentially make it seem more accurate than it is, and any higher would make a change too large. For the purpose of automating this process in the computer software, we also made several attempts to create a formula that could be run on an application. We first began by understanding what values were necessary to account for when calculating the probability. This included the independent or variable values, which were the attacker variables, defender variables, and maturity

level. As shown in figure 4, the "1" in front of each equation is the value of the whole percentage, equaling 100%. In both formulas, we multiply the maturity by 4 to make the value uniform to the 20-sided die. In the original formula, Section 1, is to calculate the effect of attacker variables on the maturity, with *V1* representing attacker variables. In Section 2, this is used to calculate the defender variable in correlation with the defender maturity and attacker variables. Once both calculations are complete, the difference is divided by 20 to get a fraction, which represents the percentage of times where the attack will not succeed. By subtracting this value from 1, we can get the decimal percentage of the attack success likelihood of an attack. However, this formula presented issues with evaluating more than two variables; in addition, "*V1*" has to be the attacker variable, if not, the formula outputs an incorrect value. To fix this issue, we simplified the formula to make it cleaner and multiplied all the variables at once to allow for more than two

Attack Success Percentage: 1 2

Original Formula *1-(((M*4)-((M*4)*(1-V1)))-(M*4)*(1-V1)*(1-V2))/20*

Final Formula *1-(M*4/20)*(1-V1)*(1-V2)...*(1-Vn)*

M=Maximum Maturity

V=Variable

Figure 4. Attack success percentage formula

variables to be evaluated. To give an example of this formula in use, the previous scenario of a maximum maturity level of 3, defense variables of 16%, and attack variables of 50% will be used. This would be .16 and .5—16% and 50%—and input the values for V1 and V2, respectively. And then, taking the maximum maturity level of 3, substitute that value for M. This would result in a value of roughly .75 or 75%, which is

1-(3*4/20)*(1-.5)*(1-.16)= .75

Figure 5. Example of Formula Use

an accurate calculation of how successful the attack will be. Following the establishment of the attack likelihood analysis, we focused on the creation of the resource system. In the CTT, the attack cost is a representation of how costly or developed an attack technique is; however, due to a lack of a numerical value, the analysis portion depends heavily on discussion, which can make it difficult to evaluate a long chain of attacks. Instead, we decided to give attacks a set number of resources at the beginning of the game, depending on who the adversaries are representing. For example, a more organized team of adversaries may have access to more resources. These resources would then be used to "develop" attacks. Undeniably, one complication with this mechanic was determining the number of resources to assign. Due to the vast range of variability among adversaries and attacks, it was impractical to provide a numerical value for them. Therefore, we fell back on the concept of discussion, although we would allow the players to assign realistic

resource values that accurately represent the scenario. For the analysis portion, we used a simplistic method of calculating the percentage of total resources used during the game. There were a few other options that were considered, such as “more than 20 resource points” being the threshold for the lowest row; however, returning to the point from earlier, without a definite system for determining resource values, this proposed method would be impractical. Additionally, the simplistic method would be similar to that of the attack success analysis. After determining both the attack percentage and resource analysis system, this largely concluded the changes being brought to both the likelihood assessment and overall game analysis. However, we did consider some changes for the impact assessment that were never implemented due to several complications. As shown

Impact	Mission Impact	Data Loss (Mission Critical)	System Performance (Mission Essential Function)	Delay (Operational Mission)
1	Fully Mission Capable	No data compromised	System Performance Not Impacted	Less than 5 Minutes
2	Partial to Fully Mission Capable	Public Access Level	System Performance Marginally Impacted	Greater than 5 Minutes ad less than 30 Minutes
3	Partially Mission Capable	CUI	Partial Loss of Functionality	Greater than 30 Minutes and less than 1 hour
4	Non to Partially Mission Capable	CUI/New Technology	Major Loss of Functionality	Greater than 1 hour, less than 2.5 hours
5	Non-Mission Capable	Classified	System Performance Severely Impacted/ Total Loss of Functionality	Greater than 2.5 hours

Figure 6. [2] CTT Mission Impact Assessment

in Figure 6, the Impact Assessment incorporates a row-by-row structure that scales from 1 to 5, and each column contains a different category of impact. However, this structure limits the ability to evaluate each impact equally. For example, if the impact of data loss was 5, system performance was 2, and delay was 1, there is no definite value that can be output. After making this observation, we attempted to find a compromise to this issue. Despite this, one major constraint was keeping the output value as a number from 1 to 5. This was a necessity as the output would go directly to the overall risk assessment, which is confined to the 1-5 scale. Considering that there were 3 different variables that needed to be evaluated, making a system where each was equally considered would be too complicated for the duration of this project, thus we decided to leave it unchanged. Aside from the analysis, a progression system was also a concept that was gamified. During the CTT exercise, the operational mission is the baseline on which the exercise progresses. It provides a scenario, an objective, but most importantly, a mission timeline. This mission timeline is the theoretical boundary within which the adversaries have to complete their cyber attack. Originally, the idea was to have the game progress in a turn-based system, except there were a multitude of reasons why it did not get further implemented. Primarily, the idea of a turn-based system was too abstract to represent a real cyber attack

in comparison to using time. Additionally, a turn-based system requires both sides to partake in some form of play; however, in the CTT, the defense team plays a far more passive role in comparison to the attackers. Thus, this led us to believe that a timeline-based system would be far more accurate in representing real scenarios and would be conceptually appropriate for the game. To further illustrate the timeline system, time is a game mechanic similar to resources in that it is a limiting factor for the adversaries. The initial concept was to have attacks take up a certain percentage of the total mission timeline. For example, adversaries could spend 15% of the timeline executing a firewall breach. But a major limitation of this idea was the difficulty of accurately depicting real-life scenarios with a percentage system in comparison to time. Therefore, this led us to define our mission timeline without percentages and use increments of minutes instead. In addition, to bring further simplicity to our game, we changed the attacker progression. The Mitre network outlines a 14-step attack chain, and in the CTT manual, the progression of attacks follows a 4-step process: Initial Access, Pivot, Disrupt, and Exfiltrate. However, upon noticing further capability to categorize the attacks, we modified the attack chain to be: Initial Access, Post Exploitation & Pivot, and Impact & Exfiltration—creating a 3-step attack chain. In regards to prototype testing we had a few opportunities. Most notably, upon the fifth week of the project, UUVRON-1 staff offered to test our game, who used a brief mission timeline involving one of their UUV operations. Overall, the testing conducted by this team was beneficial for revising the project; however, it provided little feedback for the game mechanics.

Visual Design

The visual design team was tasked with creating the graphics for the art, logo, and overall theme to depict the concept of the game and for the use of the GUI. This included steps such as becoming familiar with each type of cyber attack and defense to create accurate artistic interpretations. Consequently, due to the vast range of graphics that needed to be created, scheduling effectively was the foundation for the success of our team. As mentioned prior, scrum was predominantly used amongst our project members; however, the visual design team emphasized it significantly. We deliberately laid out our schedule to focus on different design aspects each week. Week 1 was spent creating a logo and prototyping card templates. Week 2-6's main focus consisted of creating designs for cards, computer graphics, and the game manual, with specific topics for each week. And week 7 was reserved for final edits and revisions. Following the establishment of our weekly sprints, one of the choices made initially into the project was deciding the theme of the game. With consideration for the topic of cyber security, we decided that a retro theme would strongly symbolize the concept of the digital world. Additionally, pixel art is a technique often tied closely to the retro world and has a simpler creation process, thus leading our team to utilize pixel art as our main medium. Later on, medieval themes would be incorporated due to the wide range of battle tactics that are reconcilable with

different attack and defense techniques. Following the decision of the theme, we had to find a software that would allow us to create pixel art that was both cost- and time-efficient. We came to a unanimous decision of using Pixilart, an online drawing platform. This was largely due to it being a free-to-use platform, with a reasonable learning curve. One of the first graphics we made in the software was the game and team logo. This provided us with something to make during the first week of the project and helped us gain an understanding of how to use the software. With deliberate inspiration from the name SEACAT, we created the anthropomorphic design of a cat surfing mixed with the 80's retro pixel medium and medieval theme. However, the concept of a cat in the logo eventually made its way into the game itself, inspiring many designs that integrate cats. The logo creation process was our first initial usage of the Pixilart

Figure 7. SEACAT Logo

Figure 8. Cat Inspired Card

software. As a result, we had to go through a learning process for all the tools Pixilart offers. However, by collaborating with each other, we were able to clear the learning curve and demonstrate a clear understanding of most of the tools. Some particularly useful features that Pixelart offers is the ability to import, scale, and layer past designs. This allowed us to reuse designs such as items or characters, and instead of redrawing them, the designs can be scaled to the proportions that are needed. Aside from the logo, one of the larger design tasks was creating cards that represent the different techniques within the cyber world. We did this by familiarizing ourselves closely with the types of cyber attack and defenses, as specified earlier. The main method by which we collected information on cyber techniques was through the Mitre infobase matrix. Mitre is a non-profit knowledge database of amassed cybersecurity knowledge that has been utilized by the DoD in the past. Mitre contains information on both attack and defense techniques under the web page names "Mitre ATT&CK" and "Mitre D3FEND." Through this network, we were able to gain information on cyber techniques to base our designs on. For example, in Figure 6, the card depicts the cyber attack "Internal Spearphishing."

Mitre describes this as a “multi-staged campaign where a legitimate account is initially compromised either by controlling the user's device or by compromising the account credentials of the user” [4]. We depict this by showing how the compromised account (blue) is giving valuable information to the adversaries (red), representing how the adversaries are controlling the account. Therefore, this example demonstrates how valuable Mitre was as a resource for our project. Overall, we have created over 300 unique cards for the SEACAT game, each depicting a specific defense or attack within the cyberworld, with both digital and physical productions of the cards. However, aside from the research aspect, there was an elaborate process involved in the development of these cards. Initially, the intention behind the cards was to depict each section of technique within Mitre. For example, Mitre ATT&CK has a section called reconnaissance. In this section, all techniques that revolve around gathering preliminary information, such as active scanning, occupy this section. By creating cards for each section, it would be far less time-consuming, but nonetheless, the limitations of this option resulted in us taking a different route. Some of these limitations included the point that by creating cards for each section, the design would have to be very broad to incorporate all techniques within this section, which would prevent the designs from going into specific details. Furthermore, having cards for each section was far less practical from the viewpoint of the player. During a CTT exercise, it is unusual for participants to describe a cyber attack by its technique section. If this were to happen, evaluating and analyzing the scenario would be a much greater challenge. Rather, adversaries would describe their attacks by their specific techniques. This would allow the analysis to be more thorough and, overall, provide more clarity during the exercise. Hence, although providing significantly more work we decided to create cards for each technique, while categorizing the cards into the 3 attack categories (Initial Access, Post Exploitation & Pivot, Impact & Extrifilation) described earlier. However, some techniques presented themselves with a conceptualization that contained more abstract tactics, making it harder to understand and depict the technique. To overcome this, we communicated with professionals who frequently worked in the cyber space to help us understand the concepts behind these attacks and allow us to have an easier time depicting these techniques. Another design concept that was intricately created was the color palette. Most artists make use of a color palette to ensure that their colors stay uniform throughout their pieces. Therefore, we followed a similar process, as our work consisted of creating hundreds of different designs, making it crucial that we have a selection of colors we adamantly use. As pointed out earlier, the adversaries are represented by red, and the defenders are represented by blue. We decided to follow the generic color symbolism of red representing evil and blue representing good to easily distinguish between the two sides. Aside from red and blue, we also had other colors included in our palette. As shown in Figure 7, we had a color palette of 8 colors in total.

Figure 9. Design Team Color Pallet

For each color, there was a selection of four hues—from dark to light—to further expand the variety of our options. To keep a close account of each color, a document was created with each hex color code. We frequently referred to this document to assure that the colors we were using were correct. Following the completion of the digital designs, we began moving towards the development of the physical game objects. The initial idea behind the creation of the physical cards was to find an online vendor that would allow us to cost-effectively and simply make the physical rendition of our digital graphics. Options for vendors had to be from the US and had to be cost-efficient, which put constraints on our possibilities. Due to this, a government contracted company, FedWriters, aided us in the production of our physical cards and other objects. Among

Figure 10. Card templates before & after

the most difficult parts of this process was resizing the templates for the cards. Originally, the digital design was 165 by 267 pixels—as shown in figure 8—however, for the

purpose of printing, the cards needed to be resized to fit the template of 165 by 231 pixels. This inconvenience required us to process each individual card and adjust it to fit the template. Aside from the cards, we were frequently reformatting other designs as well. Due to all three teams beginning their work simultaneously, other teams' work often required us to reformat or make additions to our work. For example, throughout the design process, we were tasked with making additions to the cards that allowed for new game mechanics to be displayed, such as time or resources. Another example includes when the software team wanted a visual timeline that could be placed at the top of the screen to represent the progress of a mission. We decided to make 3 different themes (sea, sky, and land) for the progress bar, with each having some extent of an animation. Instances such as these show how important communication was between our groups and the efforts we went through to meet the requests of the other teams. Overall, upon completion of the physical designs, revisions were infrequently made, and prototype testing had minimal feedback.

Software Development Group

The software development team worked on all aspects revolving around the computer application of this project, which included the task of selecting a programming language, drafting a game, and debugging. Aside from the responsibility of making a game from scratch, we also had to take into consideration the products and mechanics of the other groups if we were to make a computer application that was in agreement with the concepts of game development and the designs of the graphics team. Overall, this amassed a large amount of work for our team, which made scheduling an important aspect of our success. Our week-by-week schedule followed closely with that of the design team, as the majority of their graphics had to be included in our GUI. Week 1 was spent familiarizing with GDScript, the computer language we chose, and creating the bare features of the game. Weeks 2 and 3 were spent further elaborating and programming the software. Week 4 was spent implementing graphics for things such as the background or animation into the software. The rest of the project timeline, weeks 5-7, was spent testing, debugging, and quality-checking the product. As mentioned earlier, GDScript—the scripting language of the GoDot game engine—was the programming software of our choice. Godot is an open-source, free game engine that is far more secure and straightforward than engines such as Unity. This prompted us to choose Godot as the engine to create our game. Learning GDScript occupied most of the first week of the project. It would have taken longer; however, since it shared similarities with Python, a far more common programming language, we were able to learn the language far more efficiently by estimating what the approximation of a line of code in Python would look like in Godot. Additionally, one of our project members had past experience with Godot, allowing them to aid with the learning process. After learning how to use Godot and GDScript, the production of the game software began with the

development of the game's functions. One of the most useful tools we used throughout this process was the Godot documentation, which provides code for various functions, making it easier to implement certain features. For example, the file access system that we included in the software was largely made through the use of the Godot documentation. If we did not have access to this, it would have been significantly more of

```
func import_resources_data():
    var file= FileAccess.open("res://data/ATT&CK_database.txt", FileAccess.READ)
    while !file.eof_reached():
        var attack_data_set = Array(file.get_csv_line())
        attack_dict[attack_dict.size()] = attack_data_set
    file.close()

    var file2= FileAccess.open("res://data/D3fend_database.txt", FileAccess.READ)
    while !file2.eof_reached():
        var defense_data_set = Array(file2.get_csv_line())
        defend_dict[defend_dict.size()] = defense_data_set
    file2.close()

```

Figure 11. Software Code

a challenge to include some of these features. We approached the task with some ideas for a base outline, with the typical splash screens, setting controls, but we primarily created the computer interface by putting ourselves in the position of one of the players, starting where they would begin and expanding from there. Overall, the process of software development occurred with spontaneous ideas and the requests of other teams, with each of us layering levels of complexity into the process we were focusing on. One of the first features programmed into the software was the cards and card format. The game team communicated that a board or some system of visual display for the cards would be helpful for the player. Thus, we began with the format in which the players would be able to see their cards. As shown in Figure 10, the computer board was

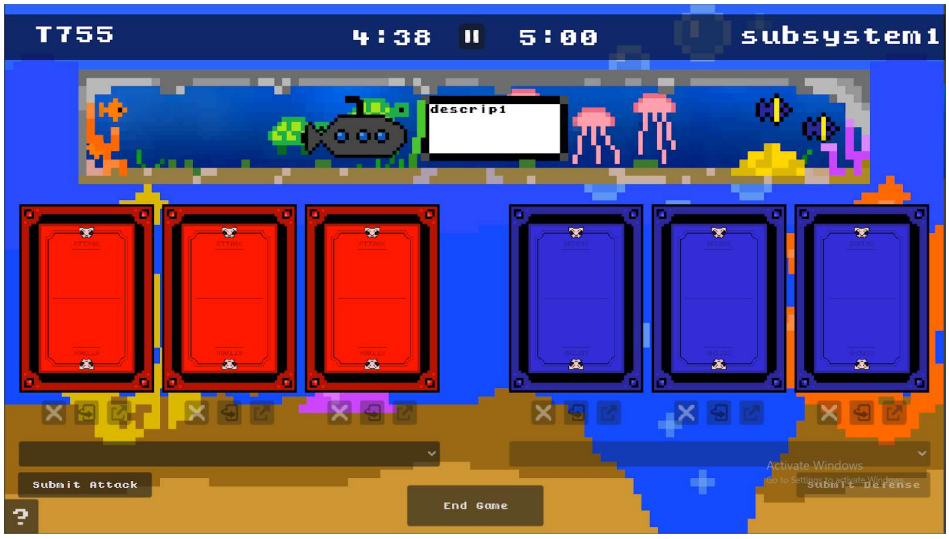


Figure 12. Software game board

formatted to have three cards on each side. By collaborating with the design team, we were able to make their graphic of the board interactable in the computer software, making the application seem increasingly like a game. However, creating the cards presented more of a challenge. Due to there being over 300 unique card designs, descriptions, and names, we had to import all the necessary data to recreate the cards in the software and then program them to be interactable. For instance, if the player were to hover their cursor over a card, the software would flip over the card to provide a brief description of the technique. We also had to program the cards to allow players to assign values for resources and time. To select cards, players are able to select their respective card through a drop-down menu in the software. Initially, we were also planning on including animation to the cards, involving the attack cards flying at the defense cards and exploding. However, complications with coding, design, and scheduling all led to this idea being abandoned, but this enthusiasm was brought to other parts of the software. Aside from the cards, another feature we implemented into the game was the progress bar, or mission timeline. The mission timeline is a crucial component to account for during a CTT, therefore we implemented it into the software to make it less complicated for the player. For the mission timeline, we once again collaborated with the graphics team to create this feature. As mentioned earlier, they designed three different progress bars: sea, sky, and land—with animation for each one. We then programmed the animation to run at 3-4 frames per second and for the timeline to advance as the mission progressed. However, arguably, the most important component we had to implement into the game was the inclusion of player input in subjective decision making. All throughout the game, it is necessary for players to make decisions with no definite answer. For example, the mission objective is a part of both the CTT and the game that is introduced purely by the player and can be completely objective. Therefore, it was fundamental that we ensured players maintained the ability to make these decisions. Other examples of this

include the resource or time system, which is again another completely objective decision that is determined by discussion between the players. As noted previously, the creation of the software was very impromptu, or “on the go.” Through this methodology, we were able to incorporate various features whenever we noticed something was missing or could greatly benefit from a new addition. An instance of this includes the game audio. To accurately replicate the atmosphere of a video game, it was recognized that game audio—sound effects and music—would have to be included. After browsing the web, Pixabay, a free online media library, was found to have audio effects and music that could be used in our game. To fit the retro/digital theme that the design team was after, all the music used in our game follows a similar theme of 8-bit audio. For sound effects, we included various audio queues, such as progressing through a timeline or clicking buttons. Alongside coding, debugging was a process that was frequently performed. In any project containing code, it is inevitable that there will be bugs within the program. During the production of the software, roughly half of our time was spent debugging. One of the most notable involved the board format and the cards. Due to an error in the code, when cards were selected, they would occasionally lock in place and cannot be replaced or opened. Clearly, this bug prevented the game from progressing and thus made it unplayable. Therefore, we focused on quickly fixing this error. One of the ways we did this was through the use of print statements. By reading specific chunks of the code, it became far more efficient to locate errors or bugs in the code. Because of this, we often made use of print statements to locate other errors we made in programming. Another example of debugging comes from the first prototype test we conducted with the UUVRON-1 staff. During this test run, the player inputted mission objective was overrun by a pre-existing file. However, unlike the example prior, print statements could not be used to find the error because it was unclear what section of the code it was in. Instead, we had to get back in contact with a project member whose internship had already ended. His understanding of the timeline allowed us to locate the file that was overrunning the program and fix the error. Nearing the end of the project, the software was approaching completion in terms of features and debugging. On account of that, we decided to automate parts of the analysis process described in the game development section. Specifically, we automated the attack success percentage formula. We originally hoped to fully automate the analysis process, however as stated in the game development section, the analysis contains a high level of subjectivity, thus we had to ensure that players had the ability to input specific values. The end result required players to input the variables and maturity level involving the attack, then the software calculates the attack success percentage of the attack.

Product Prototype

As noted in the introduction, the purpose of our project was to create a gamified version of the CTT that could be more interactive. This section will describe the mechanics of the

prototype SEACAT game, the major distinctions or similarities to the Cyber Tabletop exercise, and the limitations of the SEACAT prototype.

Gameplay

Upon opening the game manual, the player is directed to have a clear understanding of the CTT manual, as SEACAT relies heavily on concepts used during a Cyber Tabletop exercise. Next, the Pre-Game/Set-up provides explanations on how to prepare the game, which, similarly to the CTT, requires—to some extent—a preparation period. This section of the manual explains how to prepare the features of SEACAT that are not included in the CTT, such as cards. Additionally, players are also given an explanation of the computer interface, along with mechanics such as time or resources, which are important to determine beforehand. Following the SEACAT game set-up, players begin the gamified part of the exercise. In this section, it describes how players will progress through the game, explaining concepts such as counter-attacking or how to use multiple attacks or defenses during each step of the game. Most importantly, this section of the manual and game is where the majority of game mechanics, such as dice rolls, resources, and time, are used to determine how the attackers will progress and whether they are successful. In addition, players are also directed to begin the analysis process—which was described earlier in game development—consisting of large discussion periods. Following the completion of the game, the players will enter the post-game analysis. In this portion, players will use the data collected during the exercise to conduct the attack success percentage, resource usage, attack likelihood assessment, impact assessment, and finally complete the analysis with an output from the risk matrix. Finally, to note a feature of importance, the manual has a section at the end that describes several optional rules which could potentially be used to add further gamification or variability.

Distinctions & Similarities of SEACAT

As discussed earlier, the analysis portion of the CTT is one of the most crucial components that we gamified. By doing so, we introduced mechanics such as variables or resources, which significantly changed how the exercise is analyzed. However, some of these changes are yet to be explained in terms of what they represent in a real cyber scenario. The largest result of the changes we introduced was the inclusion of randomization. In the CTT, discussions are used to account for all possible scenarios, but this could lead to misrepresentation or bias towards a situation. Thus, we included a randomization factor through a dice roll to represent and account for variability that cannot be predicted. Although it is nonetheless undeniable that randomization in itself could potentially lead to a situation being misrepresented, due to it only being used for the progression of the game and not the analysis, the effects of it are marginalized. Variables are also a mechanic we conceptualized in the SEACAT game. In the CTT, as

stated earlier, it depends on discussion to depict a scenario accurately. Even though participants account for the variable factors that could affect the success of an attack, the CTT has no numerical system to determine the significance of a variable. Therefore, we created a system to provide more clarity when defining these factors, naming them attacker or defender variables. As addressed earlier in the "visual design" section, the SEACAT game has various accounts of physical and digital representation of the ongoing exercise. These representations include cards, physical models, a progress bar, and more. However, one aspect that was not greatly elaborated on was how customizable some of these visual aspects are. For example, the representation of resources and time on the cards within the computer interface is completely objective to the players' decisions. The physical cards also allow for customizability. This was made possible by placing an outer plastic sleeve on all cards, thus allowing dry erase markers to write. Additionally, another aspect that should be explained is the reasoning behind adding visuals and graphics for the majority of the concepts in the game. Aside from most games having visual models or interfaces, we noticed that the majority of the Cyber Tabletop happens in a theoretical sense, in which the bulk of processes are completed through discussion and thought and do not necessarily have any form of representation. Overall, the lack of visual representations in the CTT provided us with an opportunity to supplement this feature in the SEACAT game, making this one of the most significant differences between the two exercises. The computer software is also a feature that notably distinguishes SEACAT from the CTT. Although it was not stated prior, the purpose of developing a software—aside from depicting visuals—had two main rationales. Foremost, an online application of the game allows for virtual access. One major limitation of the CTT was the scheduling and location. Often, people partaking in the exercise had to fly out to a specific location to participate. However, having virtual software allows us to avoid complications such as travel. Secondly, computers are drastically more efficient than humans at certain tasks. On this basis, we decided to create a computer software with the second motivation of automating various parts of the game. Another considerable difference between the CTT and SEACAT is the time it takes to perform the exercise. Although shortening the timespan of the exercise was not our original intent, during our first prototype test with UUVRON-1 staff—with exceptions for preparations and analysis—the game took roughly a 1 hour period to complete. Compared to the 3-18 month interval of the Cyber Tabletop exercise, the SEACAT game is capable of providing several advantages due to the shorter duration of the exercise. Foremost, in addition to the other gamified elements that are introduced, a shorter exercise duration is able to provide a far more engaging experience for participants. Alternatively, as mentioned earlier, the intent of the game is to be played multiple times to supplement the potential loss of integrity in the analysis, and because of the shorter exercise duration, this is especially possible.

Limitations

Arguably, one of the largest limitations of the game is its dependency on many concepts on the Cyber Tabletop, which hence requires players to have a thorough understanding of the exercise to be capable of playing the game to its fullest extent. For example, players must understand how to evaluate the attack success likelihood in the CTT exercise if they are to calculate the attack success likelihood during a SEACAT game. This is subsequently a result of our game relying on players to understand and have the ability to assign realistic numerical values to a scenario. Additionally, as stated prior, the lack of access to any documentation of a CTT exercise resulted in many assumptions being made about a CTT, thus potentially leading to issues in the game. In regards to the computer software, the main limitation was its relative dependency on player input. Referring back to the statement from earlier, one of the motives behind the development of a computer software was to allow for the automation of the game. Although various components of the game were automated, there were also areas, specifically when players needed to input values like the analysis, that could not be automated as effectively. Finally, despite being a prototype, the game has not been tested or assessed sufficiently to come to a conclusive decision about the success or effectiveness of SEACAT as an alternative to the Cyber Tabletop exercise.

Future Direction

To ensure that the prototype of the game would be complete during our allotted 7-week project timeline, we—the SEACAT team—limited the scope of our work. However, this section will describe the potential for improvement and the future direction of this product. The analysis is a portion that was repeatedly emphasized throughout this paper due to the amount of attention it was given throughout this project. Despite this, the analysis continues to have several opportunities for improvement. Firstly, for convenience, a system could be implemented to determine the numerical effects of variables more readily. Currently, the game requires players to create an arbitrary value for variables that represent cyber scenarios as accurately as possible; however, if these variables could be elaborated as a concept, it may be possible to determine them beforehand. Additionally, this could further remove bias or the misrepresentation of a situation. Secondly, the impact methodology was mentioned earlier as an area that could potentially supplement gamification but was outside of our project timeline. Furthermore, if possible, the impact analysis should also be weighed to determine what results in the greatest consequences. To elaborate, a mechanic that we considered implementing was the evaluation of secondary impact. In most cyber attacks there will always be consequences, even if the primary mission of the adversaries is not completed. For example, during a cyber attack the adversaries need to bypass a network scanner as one of their steps for their mission, which they successfully do. However, they do not complete their primary mission. Even though the mission is not complete and there is no primary impact, there is a secondary impact from the loss of the integrity of some of the defense systems. Although secondary risks seem to be very relevant,

they are not accurately reflected within the CTT, which presents an opportunity for change. Lastly, referring back to the issue of player dependency in the computer, future production of SEACAT should emphasize the development of fully automating the analysis process within the computer software. In terms of visual design, a possible route for future development could be exploring other themes aside from retro. Although not mentioned during the limitations, a particular concern about the 8-bit graphics is determining whether the text is legible for all participants in the game. Therefore, a different graphics theme could benefit SEACAT.

Conclusion

Conclusively, the SEACAT prototype was completed over the 7-week project timeline, with all teams accomplishing their respective goals towards contributing to the overall development. As described, SEACAT comprehensively shortens the length of a CTT and provides an alternative method of analyzing a system's defenses without compromising the final result or analysis. The game is complemented with visual graphics that accurately depict cyber situations and a computer software that, in itself, is accompanied by graphics but also highly automates the process of the game. For the most part, we—the SEAP interns—acknowledge the SEACAT prototype to be complete and our contributions finished.

References

[1] Office of the Under Secretary of Defense. *Cyber Table Top Methodology in Test and Evaluation*. Cyber Table Top – DoD Research & Engineering, OUSD(R&E) (cto.mil)

[2] Department of Defense. *The Department of Defense Cyber Table Top Guide*, version 2.0, 2021. DoD-Cyber-Table-Top-Guide-v2.pdf (cto.mil)

[3] Pirinen, Tuomas et al,. *Mordheim*, Games Workshop, 1999.

[4] MalcomVetter, Tim & Swetha Prabakaran. *Internal Spearphishing*, The Mitre Corporation, 2019. Internal Spearphishing, Technique T1534 - Enterprise | MITRE ATT&CK®