Lock_Out

A Cybersecurity MQP and Game

By

Rosemary Lindsay

Kyle Stack

Alex Hebert

Chandler Reynolds

Abstract

Teaching games can be an extremely effective strategy for learning about a new subject. The issue is that it requires a delicate balance of real facts, understandable gameplay, and playability. While we were able to find several cybersecurity teaching games, none of them had all three balanced in a way that seemed viable to us as game designers. Therefore, we made a game that would balance the three.

In this paper, we will describe the game we have made over the course of this year and the processes we went through to create it. We will begin by explaining our reasoning for making the game, followed by a description of the game in its final form. We will then describe our process in making it. We will finish with a reflection of our work throughout the year, as well as potential future paths this project could take.

Table of Contents

Table of Figures

Chapter 1: Project Overview

In this modern world, consumers rely more and more on technology to help them manage their lives. This can be seen in schools, workplaces, personal lives, and communications. Even shopping can now be done through the medium of a computer or phone. But the more we allow technology to assist us, the more vulnerable we are to attacks through that technology.

As long as there have been networks, there have been attempts to abuse those networks. Even as early as the 1980s digital vulnerabilities have been enough of a threat to risk jail if caught exploiting them (Cornell University Law School, n.d.). As the networks themselves became more advanced, so did the attempts to use them for illegal personal gain. While some people are specifically trained to patch these vulnerabilities, others find themselves lost in the sheer number and complexity of potential attacks.

This is why teaching about this field, Cybersecurity, is so important. Though some issues require advanced programming or a large store of knowledge, some are as simple as not clicking a suspicious link. By making users aware of simple things they can do to make their lives and workplaces more secure, we can reduce the number of vulnerabilities in a system. Additionally, by alerting them of the existence of other threats and explaining why they are issues, users will gain a greater understanding of how to react to threats or how to research digital security.

Our initial research found that we were not the only ones with this idea. Several other groups, including PBS.org's Nova Labs and the somewhat older cyberCIEGE have created games for teaching cybersecurity topics. Although these games adequately covered many types of attacks, there were times when irregular gameplay or over-realism distracted from the learning. Additionally, in the ever-changing field of cybersecurity, each new year brings new threats and solutions previously unheard of.

Creating an engaging game that is able to teach players the basics of cybersecurity would greatly help the overall population defend against common threats. To accomplish this, we developed a game whose narrative used elements of mystery and suspense to keep players engaged. Rather than focusing on technical details, aiming to generalize and abstract topics to allows understanding from people not already in the field. Among the topics included in our game are phishing, spyware, and the importance of antivirus software. These three cover some of the most common issues in cybersecurity so that the game appeals not only to professionals in government and corporations, but also to the general public.

Chapter 2: Game Description

2.1 Goals

The major demands of the project were to create a game that teaches players about cybersecurity and hold their interest while they play. However, it was also necessary to spark the players' continued interest in the subject matter even after the game was over. For the first, we looked at previous cybersecurity games and analyze what worked for them and what didn't. From there, we created the game based on both our findings and from our advisors' input. To be able to hold the player's interest while playing, we created an original storyline and characters that would be interesting enough to keep them hooked.

Our third goal was the hardest, as players' interest beyond the game could vary greatly. Eventually it became clear that the best way to achieve this was to ensure that throughout the game the player was reminded that this was just a fraction of what was out there, and then at the end give them more things to potentially explore.

2.2 Mechanics

Within the final version of Lock_Out the primary mechanic of the game is the email system. Each day when the players log into their computer within the game, there will be several emails that they will have to read through. For each email the player will choose whether to allow the email to go through the system or report it. If they choose incorrectly they may enter a mini-game as a light punishment for their incorrect decision. Additionally, there is a set of statistics that is affected by the choices the player makes on the emails. These values, found on the bottom left hand of the office computer screen, serve to inform the player of whether or not they made good choices with the emails they receive. This serves the purpose of informing the player of possible outcomes for their employer, in this case a university, based on the actions they take. The mini-games which the player enters at certain points of the game, along with the emails, are intended to vary the gameplay which might otherwise be repetitive. The first mini-game has to do with phishing attempts. If the player makes a wrong choice on a phishing email they will enter a mini-game where they must avoid lines of text as they attempt to grab the email out of the system, reinforcing the basic concept.

Additional mechanics include character interactions and dialogue choices where the player has conversations with their co-workers in the game. These interactions uncover clues about a mystery unfolding in the IT office concerning the disappearance of an office worker, Robert Finch, who had the very job now belonging to the player. Throughout these interactions, the player learns about the story and cybersecurity, at the same time, keeping them engaged and interested in what could happen in the future.

2.3 Story

The game begins by asking you to log in to your email account. You are given the option to do so or skip this process. If the player attempts to log in, the game, personified as the AI character Lock, stops you and decides you need to learn more about cybersecurity and safety. Either way, you are then placed in an office where the boss welcomes you as a new member to the IT department, where you're then taught the basics of catching phishing emails. As the day progresses, the player can leave their desk at designated times to talk to the other characters. While optional, doing so allows the player to either gain knowledge by asking questions, receiving tips and advice, or learning more about the story and characters. As the first day draws to a close, the player is pulled out of the game by Lock, and quizzed on what they learned. Lock gives the player a break from the office narrative with mini-games and questions.

The player is then returned to the office where they are informed about the threat of malicious software, such as viruses and other forms of malware. At some point during the day the player may begin to notice their computer acting strangely, and at the first break players can question their co-workers to find out the cause of the disturbance. Your fellow co-worker “accidentally” allowed a malware onto the system, which began causing problems. From here, you can attempt to remove the malware without your manager noticing, or you can tell your manager about your co-worker’s suspicious mess-up. Then once again, the AI pulls you out of the office for another quiz and minigame.

On the third day, the player must deal with the company asking them to change their account password. However, the email asking employees to do this turns out to be a phishing email: a ploy planned by someone who knew it was the day for everyone to change their passwords. The player also receives an email from Robert Finch, revealing that he is in a dire situation and his account information has been compromised.

In the final chapter, the AI becomes corrupted and the player has to shut down the game from the inside and escape. It tests what the player learned, and ends when the player ‘escapes’ from the game. At the end, Lock reveals that it faked becoming corrupted in order to test what the player learned, and evaluates them accordingly.

2.4 Art

Our game is intended for a corporate audience, so we used two distinctly different art styles for our game. The first style, which much of the UI and in-game communication uses, is intended to resemble professional dialogues and applications. The intent behind this is to make players recall the game when they see the real-world equivalents of the scenarios in our game, allowing them to more successfully use their gained knowledge.

The second art style was used for both character art and the in-game “world”, a highly pixelated sprite style. The pixelation removes most of the identifying features of the characters, allowing a larger audience to relate to the world than if we used detailed (and possibly recognizable) assets. It also serves to remind players that our in-game “attacks” are merely abstractions of reality.

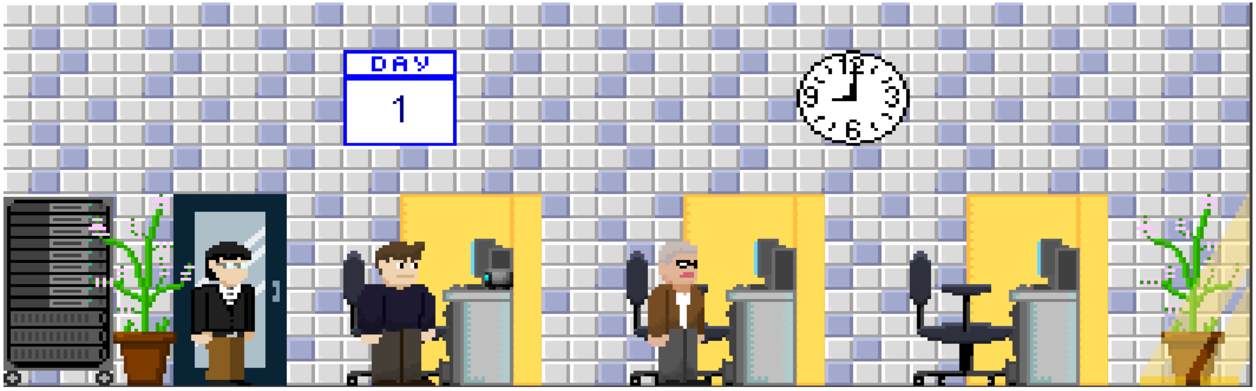


Figure 1: In-game office screenshot

Chapter 3: Development

3.1 Redesigning and Planning

The original design for our game revolved around the player working for a company, where they would have to manage and balance company resources, security, and user satisfaction. The game was going to be more of a workplace simulation where the player sorted through emails to help manage the company's security. While the player sorted through the emails they were given, they also had to pay attention to the various resources at their disposal. At the time we considered having a fourth resource, but eventually merged it with User Satisfaction. The player would use their resources to deal with various issues from the emails they received, from complaints and questions about technical assistance to phishing and scam emails. The game itself was more of a simulation than a game, where the player simply sorted through emails and learned almost nothing.

Halfway through the project it became clear that the scope of the story was too large for the mechanics previously developed, and we needed to rethink much of the design. The story that was written was for a pure simulation, rather than one for a serious game. As a result, not only was the game not enjoyable with less story, and the story didn't fit as well as it could. At this point, we adjusted much of our story and game mechanics in order to create something that retained the good aspects of our original plan while getting rid of those aspects that did not mesh as well with the overall theme.

3.2 Story

3.2.1 Original Narrative

Originally, the game was designed to involve an evil company attempting to steal a recent advancement in technology from WPI, which would be played out over the course of five chapters. The first chapter was more of a tutorial in which the player would be introduced to the department and the other characters. During this chapter, the player might see a few hints as they sorted through emails that foreshadowed events to come. The second chapter was very similar to the first, except that it would introduce scam emails that also could contain malicious programs in addition to needing to watch out for phishing emails. During this time, additional emails and conversations between the other characters would continue to create an air of mystery as rumors of a shady organization circulated.

The start of the third chapter was where the evil organization began to make its move, starting off with a grand announcement to the public regarding this revolutionary technology that was being kept secret from the rest of the world, and their goal to either spread it to the rest of humanity or to destroy it outright. The player would also have to deal with direct attacks through the net as the organization attempted to break in.

The fourth chapter involved a member of the organization infiltrating school grounds and placing a USB drive with a virus into one of the devices in the server room. This severity and outcome of this event are partially dependent upon decisions the player would have made earlier in the game, such as tightening security to the displeasure of students and staff.

In the fifth chapter, that the virus uploaded to the system allowed the organization to lock everyone out of their devices and hold them as hostages so that the school would hand over the technology and all of its research. The player would need to battle through puzzles and test their knowledge of what they had learned in order to contain and eventually suppress the virus.

We quickly learned, however, that the original story created was too broad, without enough substance to each part. Some important bits were too vague, and our writers were having trouble filling them in. Additionally, it wasn't something that worked well with our design for the game at the time. Because of this, the game was entirely re-scoped and the story re-written.

3.2.2 New Narrative

As we worked, our rate of progress made it clear that we had to cut back on the scope of both the game and the story. We'd keep some of the elements from the first iteration, but change the story enough, as to make it more comprehensive and fit the gameplay better. First, we shortened the game from five chapters to three, keeping the initial tutorial chapter, but removing the evil organization from the game entirely. Instead had the player interact with an Artificial Intelligence who begins to teach the player about various aspects of cybersecurity. By moving away from a real world scenario and to a teaching game, we found it much easier to not only develop an interesting story for the player, but also to complete our goal of teaching people about cybersecurity at the same time. In this new story, the AI or game became the main non-player character, Lock, while the player acts according to its instructions. At the same time, we simplified the email system to make it easier to understand how to properly work with the emails the player received at the terminal rather than having them serve as a way to manage the company's resources.

As events play out in the game, the game becomes corrupted due to a virus, and the AI that had been instructing you is suddenly incapacitated. At this point the player must use their accumulated knowledge to shut down the game and escape from the computer. When the player finally manages to 'escape' we find out that the AI hadn't been corrupted at all, and it was the game's final test for the player to see if they had truly learned what they needed to.

3.3 Art

3.3.1 World assets

In our original concept, there was no world outside the email screens, but as the game developed and grew, we included an office scene that would allow transition between mini-games and topics. Additionally, this would allow more character interaction, as "physical" characters are easier to connect to than characters hidden behind email chat. After discussion, we used a pixelated art style for the world, to broaden the relatability and match the character designs we originally came up with.

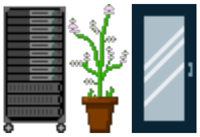


Figure 2: Pixelated assets from game; server, potted plant, door

Once that was decided, the next step was to figure out what items were necessary to indicate to viewers that this was an office without getting too cluttered. Some things were simple to decide on, such as walls, floors, desks, chairs and windows. These items could be found in most offices, and would not detract from the user experience.

The next few items were chosen because of the game specifically: computers, a printer, and a clock. The computers were an obvious choice--both facilitating the email segments and indicating that this office was based around technology. The printer was for a specific plot point. The clock was used to indicate the passing of time, facilitating the passing of "days" as topics are broached. While not every asset made it into the final game, the additional aspects could be used in further development of the game.

3.3.2 Character Assets

The character assets changed drastically from our initial design, possibly the most out of any part of our game. In our initial design, characters would only be seen as profile pics on emails, 256x256 pixel abstractions of people. Then, as we moved away from the original idea of a pure simulation, made a world outside the emails, which in turn gave us the option for character sprites. These character sprites each would be composed of 16x16 sprites.

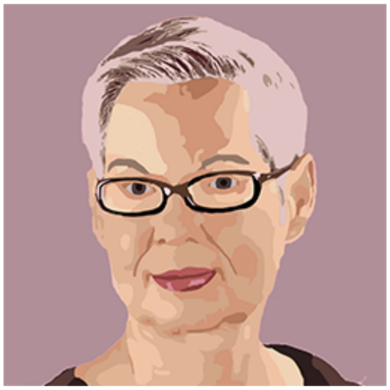


Figure 3:Original art concept for Janice

This changed when we restructured our entire game after our first build. The greatest issue was that most of the assets were 32x32 or 64x64 pixels, making the characters too small. This along with the fact that we lessened the number of characters made us believe a complete revamp of the character sprites was possible. Additionally, Kate Olguin, an artist experienced in sprite art, offered to assist us on our game and successfully created more unified sprite and profile assets for our remaining characters.

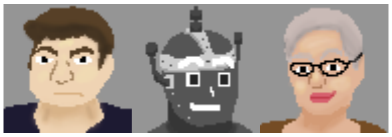


Figure 4: Final portraits for characters

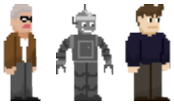
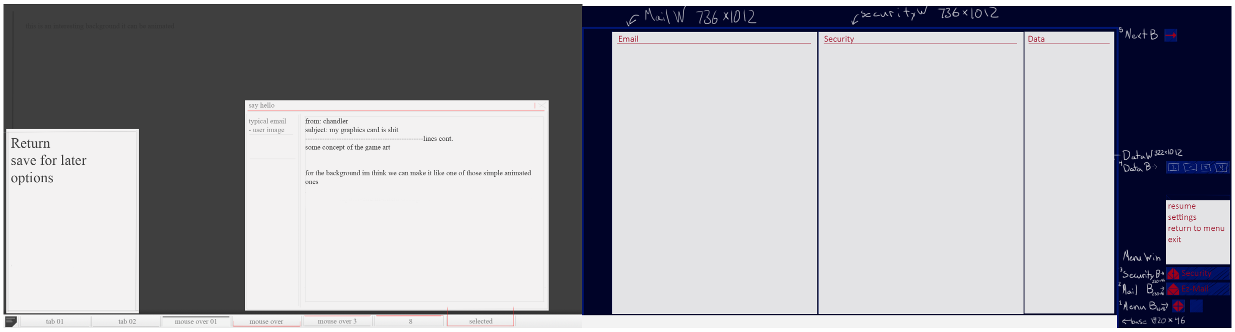


Figure 5: Final in-world assets for characters

3.3.3 UI

UI assets expanded a great deal from the start of development. They did not however undergo many design changes. In the first version of the game where the player was playing a simulation game there was only a single UI set. This UI was the ‘office UI’. The concept for this UI was a clean and simple style with a small color palette. Figure 6 shows the first designs for what the windows would look like within the game, along with various styles for the tabs.



from: chandler
subject: my graphics card is shit
---lines cont.
some concept of the game art

for the background im think we can make it like one of those simple animated ones

Figure 6: Original color and layout design (left) and window layout (right)

It was at this point the game underwent its overhaul. This saw the reduction of windows as well as the introduction of other UI. The next set of UI saw the loss of the Security window and the dominance of the email window as it was now a primary mechanic. The concept of a simple style remained, except for part of the background due to the odd window shapes (Figure 7).

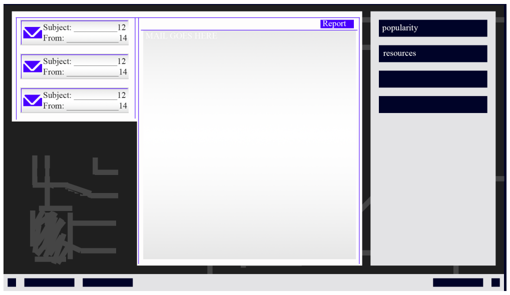


Figure 7: UI design progression

The next form of the UI, which is the final version used, returned to the original color palette and only used one window for the office UI, further simplifying what was on screen and allowing it to appear ‘professional’ (Figure 8).

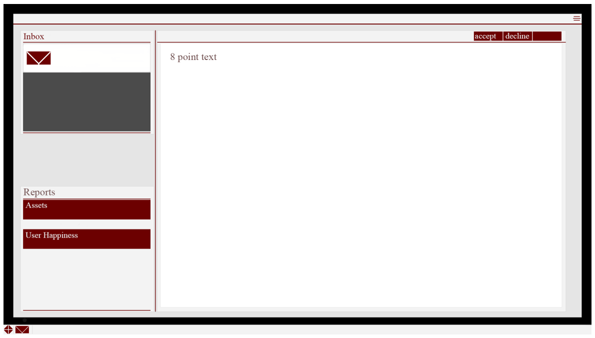


Figure 8: Final UI design

This design was also used for the other UI sets, leaving plenty of space for speech text. These areas included the AI area which was made to resemble a command prompt screen, the login screen, and the main menu. These assets were made faster, and only had one version. It may have been beneficial for these assets to have multiple versions as there could have been a better way to develop them, but they fit well enough. The one design point that affected all of these areas as well as the office was the in-game menu button.

3.4 Programming

The programming of Lock_Out was a very complicated process. Because we chose to work in the engine Game Maker Studio, it was very difficult to collaborate on the programming side of the project. Thus, only one person could be working on coding at a time, which became a bottleneck for the project later down the production line.

There were quite a few changes to our idea as to what we wanted the gameplay to be, which drastically affected the programming of the game. Specifically, the emailing system went through 3 drastic changes- from procedurally generating a narrative, to generating based on the type of email, to each email having defined content.

There was a constant reorganizing of assets, objects, and scripts. Old code frequently had to be entirely thrown out to begin programming new mechanics and narrative changes. Even some modular systems in the game, such as the initial email generation system, had to be scrapped, as later narrative choices required a different way of displaying the emails. This constant scrapping and re-writing of code wasted a large amount of time, and showed the importance of formulating a solid idea of what a game's core mechanics will be.

In the end, the game had quite a few modular systems coded in. From email generation, to character speaking, to dialogue options, much of the code was written to be as easily expandable as possible. This allowed for code to be rewritten when needed in a very reasonable amount of time, and new narrative content was also easy to add.

3.5 Audio

The audio for Lock_Out was difficult to prepare. The overall tones of scenes were variable due to the nature of the game's narrative, and thus it was difficult to find music that would fit all or most of the situations. The cutting out of music and playing of different tracks was utilized to affect the mood and tempo of the game as well, with cut-outs being used for both narrative and comedic effect.

The sound effects are all 8-bit sounds, mirroring the pixelated style of the game. This choice of audio also made it much simpler to gather the needed sound effects in a reasonable amount of time, making this part of production much smoother.

3.6 Playtesting and Polish

Playtesting revealed many things about the game and narrative that we would not have been able to catch without it. First, in terms of bugs, glitches, and narrative errors, playtesters were much more able to find these than the team working on the game, as they were able to think and try things that we would not even think of, such as seeing that reporting every email would have no bad repercussions.

We were able to gather many meaningful notes from playtesters that will help us improve the game, such as making it more obvious that players should talk to the NPCs in between email sessions, and making it clearer as to why an Allow or Report choice was correct or not. This could be achieved by the game character, Lock, popping up and explaining something about the email the players should look out for.

Playtesting has shown that, for the most part, the game delivers the intended message that we set out to convey. Players learn about phishing emails, good anti-virus practices, and when admin access should be revoked in certain situations. Most players were able to pass the finale, suggesting they had learned a good deal about Phishing through the game.

3.7 Additional Mini-games

In addition to the overall game, Lock_Out, two separate mini-games were created in an attempt to explore the potential of these games to contain less abstract attacks by using lower level abstractions. These two games, Piggyback! and B-Link show two methods for making more realistic games.

3.7.1 Piggyback!

The game Piggyback! is intended to show players one of discovering spyware or other, malicious software. In this mini-game, the player is faced with the scenario that their smartphone has some sort of suspicious program running on it, noticed by the irregularly draining battery. They are then shown several programs alongside their current drain in milliamperes (mA). Using a combination of logic, hints given on the screen, and their experience, they must decide which program is misbehaving.

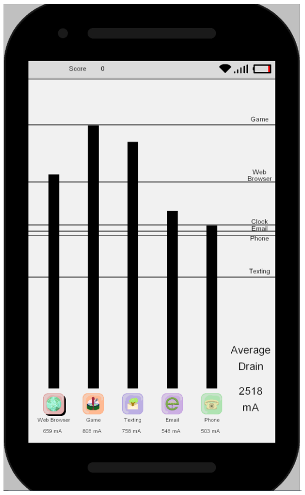


Figure 9: Piggyback! gameplay screenshot.

One of the primary methods used to recognize spyware is to compare its behavior to known programs (Mohamed Adel Sheta, 2016). This game uses data, taken from actual hardware, to show realistic battery usage for various apps (see Appendix 2). Although when the game starts the play screen merely depicts the battery drain of each app, the player can press "h" to show the average drain of each app normally, allowing them to notice irregularities. The game continues until the player fails to correctly guess which app is behaving oddly, and the score is based on how many they get right in a row. If added to Lock_Out as an integrated mini-game, either a timer or limited number of game instances could be implemented to help with story flow.

3.7.2 B-Link

The game B-Link is slightly different, being a game that teaches players about the Link Layer of a system connected to a network. The Link Layer establishes and maintains connections between network entities (Oxford, 2016). It receives data in "frames", translated by the physical layer from raw energy into readable data. The frame contains the information being sent, as well as directions for where the data is directed, and where it is from. The Link Layer also checks for errors in the received frames and forwards frames that are not intended for it.

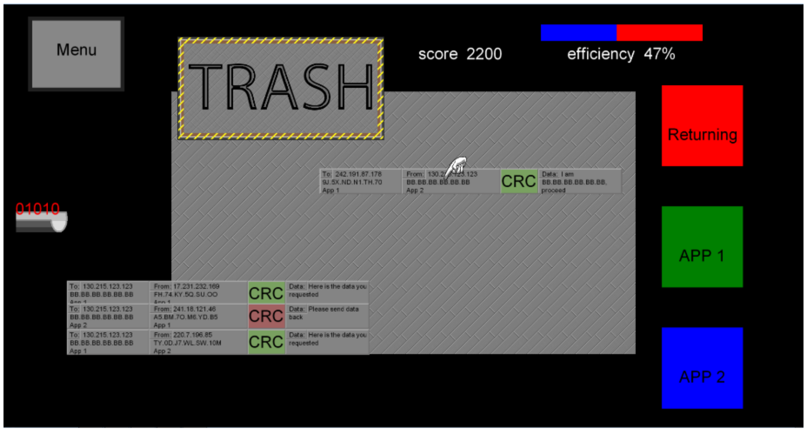
One of the most efficient methods for the Link Layer can use to check for errors is the Cyclic Redundancy Test, or CRC. The CRC works by appending a value onto a sent frame. The sender and receiver both possess a key that, when an uncorrupted frame with a CRC is divided by it, will return as 0. The sender uses this key to encode the CRC, and the receiver uses it to check for accuracy (Ying Wu, 2012).

ARP, or Address Resolution Protocol, is a method used in networks to connect two nodes. When a networked computer needs to send data to another, they broadcast an ARP request which is answered by the computer they are seeking. While real-world ARPs have a section of the frame dedicated to them, B-Link simplifies them down to just data. Additionally, ARPs that are not intended for the player are deleted, rather than stored (Talal Alharbi, 2016).

In B-Link, frames intended for other computers are not forwarded, but ignored. The only frames it receives that are intended for others are ARP requests, which are broadcast to all IPs. The IP address was chosen based on the majority of WPI IP addresses, which tended to follow the format 130.215.X.X.

MAC Addresses are assigned by vendors, and change based on the creator. For the purpose of this game, a MAC Address was chosen ignoring vendors. B-Link boundaries of the Link layer, bleeding into the transport layer slightly in order to create more interactive gameplay

B-Link is a fast-paced sorting mini-game in which the player takes the role of the Link Layer of a computer with the MAC Address BB:BB:BB:BB:BB:BB (MAC-BB). In the game, their job is to deal with incoming frames, making sure they arrive at their intended locations using the frame data given to them. There are five types of frames the player will deal with: incoming data, incoming requests for data, incoming ARP frames for them, incoming ARP frames for other computers, and outgoing data.



To: 242.191.87.178 9J.5X.ND.N1.TH.70 App 1	From: 130.2[illegible]123.123 BB.BB.BB.BB.BB.BB App 2	CRC	Data: I am BB.BB.BB.BB.BB.BB, proceed
To: 130.215.123.123 BB.BB.BB.BB.BB.BB App 1	From: 17.231.232.169 FH.74.KY.5Q.SU.OO App 1	CRC	Data: Here is the data you requested
To: 130.215.123.123 BB.BB.BB.BB.BB.BB App 2	From: 241.18.121.46 A5.BM.7O.M6.YD.B5 App 1	CRC	Data: Please send data back
To: 130.215.123.123 BB.BB.BB.BB.BB.BB App 1	From: 220.7.196.85 TY.0D.J7.WL.SW.10M App 2	CRC	Data: Here is the data you requested

Figure 10: B-Link gameplay screenshot

Incoming data is sent on to the higher levels of the computer, and on doing this the player receives points to their score. Requests for data and ARP frames for MAC-BB are dealt with similarly, but produce outgoing data as well. ARP frames for computers other than MAC-BB are deleted.

Outgoing data in the game is only produced in response to incoming data requesting it. When outgoing data is produced, it must be moved to the in/out ports where the frames are coming in. The player must then send out the data at a time when no data is being sent in. If successful, the player's score will increase. Otherwise, both ingoing and outgoing frames will be corrupted and overall efficiency will go down.

Any frame in the game has a small chance of being corrupted. Additionally, failure to send outgoing frames without interrupting incoming ones corrupts both frames involved. Corrupted frames can be recognized by their red CRC. Corrupted frames must be deleted, as they contain inaccurate data. If a corrupted frame is not deleted, or if any other frames are dealt with incorrectly, the player's "efficiency" is decreased. If efficiency ever reaches 0, the player loses. Efficiency can be increased by properly dealing with frames.

Chapter 4: Review

4.1 Successful Goals

The goal of our project was to create a game that teaches players about cybersecurity and the impact it can have on both individuals and businesses when not handled properly. In addition, we wanted a game that had an actual story, and that was interesting and fun to play. We feel that we've successfully met our expectations, even if we had to cut back on the original scope.

Based on feedback that we received in the playtesting stage, players replied that they felt they had a better understanding of cybersecurity after playing through the game. When it came to implementing an interesting story that also worked with the gameplay, we found that creating a mystery within the game helped keep players interested in the events taking place, especially from feedback where people asked questions about the characters and events that took place within the story.

Finally, we had to make sure the game was fun to play. Due to Lock_Out being a serious game, we had to design it differently from a regular game in a popular franchise. By having several mini-games at various points, we managed to make the game more enjoyable to play than just quizzing players after presenting information. By tying the mini-games into the story and topics we were trying to teach, we were able to create a cybersecurity game that not only teaches, but also is interesting to play.

4.2 Unsuccessful Goals

Throughout the development of Lock_Out, we were unable to achieve several goals we hoped to achieve. The most notable of these were creating a realistic simulation and accurate depictions of attacks. These goals were relevant because having a full understanding of how and why these attacks happen is important for those who have the potential to be attacked. While abstractions of these attacks can inform people to the importance of defense, the idea of how to do so, and what happens if you don't, further understanding can be important.

These goals were not achieved due to the amount of time available for production. Building a realistic simulation of hacking events would have been time consuming and may not have been accurate. Another issue that prevented the creation of such a game was target audience. The simulation would have targeted a smaller, more tech-savvy audience. However, we hoped the final design would reach a broader, more entry-level audience. Accurate depiction of attacks was also limited by time, and by the fact that attacks themselves are constantly changing.

Similarly, we did not fully explore the idea of cost on a company scale. While it does explain the issues with data or security being compromised, Lock_Out does not fully describe the severity of large companies being hacked.

4.3 Important Lessons

Throughout the process of designing, writing, coding, and creating this project, we learned many things about production and the game industry. One major lesson learned is that designing serious or educational games with interesting and engaging story and mechanics is very difficult. One must go to great lengths to balance engagement of the player with portraying the lesson one wants to teach through the characters' dialogue or the game world. Adding too much immersion may take away from the game's ability to teach a lesson. However, focusing too much on teaching the lesson may take away from the player's immersion in the story, making them uninterested in the game and therefore failing to properly teach the lesson.

We have also learned a great deal about the production of games in general throughout this project. Planning out the major designs, mechanics, and narrative is critical to being able to complete a game's production in a timely manner. Pre-Alphas and Alphas are important to start as early as possible so that one can test the base design, mechanics, and narrative of the game and see if it successfully achieves its experience goals.

Another lesson was how difficult it is to design and create a game in a limited amount of time while also having an interesting story that keeps players engaged. When creating games with no set deadline, it's much easier to be able to work at your own pace and polish everything to the level of quality you desire, whereas with deadlines you often need to cut back on areas that you wish you could've fleshed out further. In addition, we found that while we had some fantastic ideas, many of them just weren't cut out for use in a serious game. Instead, we had to completely redo our story at one point to make it match our reworked game concept.

Until this point, we had never worked on any serious games, only games that were designed to be fun to play and that people would keep coming back to. As a result, we learned just how difficult it is to make a game that is fun and has a good story, but is also a teaching tool. Making a fun game is straightforward, but making a teaching game is extremely hard.

4.4 Issues

Although we completed the majority of our goals, we did run into a few major setbacks along the way, both with the game itself and with each other.

The first and largest of these was when, late B-term, we realized that our scope was too broad and vague. The current design had a large storyline, which we had not even managed to get one third of the way through. Once we realized the issue, we quickly set about rescoping. We discussed what we managed so far, and based on that decided to intensely focus each chapter and reduce the number of characters. By doing this, we were able to make the story smaller, concentrating on the important details and people. This reduced the amount of player interaction with and influence on the storyline, but allowed us to concentrate on making what we had more interesting and enjoyable instead.

The second issue occurred when one of our members, our lead character designer, was unable to continue on this project due to personal reasons. While we already had the personalities and approximate character interactions completed, we had very little character art finished. Fortunately, our rescoping reduced the number of characters needed and with help from our advisor, we were able to find outside help for character art.

The third issue stemmed from our rescoping, but did not crop up for some time. When we were rescoping, there were some miscommunications about how some chapters were going to be split up that were lost in the rush to recover lost time. By the time this was noticed, some parts were already written that contradicted others. Though it took some time, these parts were able to be integrated throughout the story, and even increased interconnectedness.

Our final major issue came with designing the finale. While writing the finale, our writers realized that nobody had planned a climax for the game. While one could be written in, it would not have the proper emotional impact, or remind players of what they have learned without additional input. However, we did not want to put too many new features into the game after so long. After some debate with our programmer, it was decided that we would implement callbacks to the assorted minigames previously implemented in the story to both test their knowledge and to create an obstacle to enhance the tension.

Chapter 5: Future Potential

Based on our experience in this MQP, we can see three major ways of improving or building off of this game. While there are many paths this project could take in the future, these are the most useful and unique possibilities.

5.1 Content

The first and simplest path to further this project would be to increase the scope of the game. This project was conceived, designed, and implemented within a year, and as such had a limited scope. Another group working from the base that our project creates could expand the game, making it a more overarching product.

5.1.1 Story

One way they could do this is by expanding the story. While our current story flows well and fits our mechanics, the smaller scope of our project limited the amount of meaningful choices we were able to implement. Even those that we have are primarily influential within the structure that our story creates. Now that the art and game structure are implemented, a group that concentrates on creating a branching story could build off of them to make a game whose structure itself is influenced by player choices.

For example, throughout the writing we toyed with the idea that, if the player noticed certain key flags during the course of the game, they could notice the "glitch" that causes the final challenge and possibly skip it altogether. This would reward them for noticing small issues as well as giving a little more replayability to the game as a whole. Eventually, we scrapped this idea in order to have a consistent climax to the game. A group who could concentrate solely on the story would be able to implement alternate story paths and alternate climaxes.

Alternatively, by adding small amounts of randomization to key pieces of the story, replayability could be increased by a large amount, as key events could happen in different orders, or not happen at all. Each game experience would be completely different. Especially in areas such as the email responses, repeat players could merely remember the proper answers, allowing them to skip through sections easily. Randomization would limit this, using a larger pool of emails for each section to give new experiences even to repeat players.

Finally, new chapters could be added to the game about new subjects, expanding what topics the game introduces players to. Currently we have focused our game on the more well known and common attack types, in an effort to introduce newcomers to the field of cybersecurity. By creating new chapters that gradually increase the sophistication and subtlety of the attacks, even more experienced players may learn something new.

5.1.2 Mini-games

Another way to expand the game would be to modularize the mini-games, allowing new mini-games about other topics to be added easily. Although adding new mini-games to the game proper would require alterations to the story (see section 5.1.1), even without story support mini-games could be made accessible through the main menu.

In order to teach players about the subjects these mini-games reference without support from the story, we would recommend the following: a description of what the mini-game is an abstraction of, how the abstraction represents the actual threat or solution, and a brief explanation of why this abstraction is relevant or important. The following are some potential additional subjects mini-games that could be created around, and some potential limitations to the abstractions.

Man in the Middle Attacks

Man in the middle (MIM) attacks occur when data is being transmitted between computers or systems. A simple example would be as follows: An attacker intercepts the public keys of two computers wanting to communicate, and sends both of these computers its own public key. From there, it uses these public keys to encrypt messages it receives from one to send to the other. Because the two computers are receiving and sending the correct data, they may believe that their communications are secure, not realizing that the attacker is able to view, delay, or even change the data sent (Mauro Conti, 2016).

Within a wireless Local Area Network, it is possible to limit the ability of MIM attacks. A program can be created that searches transmissions across the network for the characteristics of MIM attacks, such as the reassignment of IP/MAC pairs. Upon noting potentially suspicious behaviours, it can notify an administrator of the potential vulnerability (HUAN-RONG TANG, 2009) (Vikas Kumar, 2012).

A highly abstracted mini-game based upon this could be a maze in which you are seeking a given exit. There would also be a fake "exit", which could be identified as fake through subtle visual clues. In this game, the maze would be the medium in which a signal is sent, the player character would be the data, and the exit would be the goal of where the data should be sent. The fake exit would represent the MIM attacker.

A slightly less abstract mini-game could be a representation of the finder program's job, seeking out potential intrusions. Given lists of information for a small network, players would watch out for suspicious behavior and unauthorized changes in the data. Given a chain of commands of things to do to the data they have, they could send suspicious commands to a "network administrator" to deal with.

Spyware

Spyware is a broad label for a variety of programs that can discover and transmit sensitive data without users being aware of it. Some examples include keyloggers or apps that collect private information in the background (Hui Xu, 2012). Spyware is often difficult to detect because of its constant evolution. Four common detection methods are as follows: Comparing potential spyware to a database of previous spywares; analyzing the behavior of a program and comparing it to that of spywares; comparing the behavior of a suspicious program to that of what it claims to be; and data mining (Mohamed Adel Sheta, 2016).

Potential mini-games for spyware include a "guess who" style game and a memory style game. In the "guess who" game, the player would use behavioral patterns of the program to discover which part is the spyware and deal with it. The memory game would focus on recognizing bad programs based on previous bad programs.

Firewalls

Firewalls are primarily used to restrict access to certain areas for unauthorized users. In order to improve effectiveness, they are often broken up into pieces, each of which deal with specific areas or programs (Sukhveer Kuar, 2016).

A highly abstracted mini-game could contain a top-down shooter or invaders style game, where "bad" data must be gotten rid of, but "good" data must be kept. In this instance, the player would be the firewall, using visual cues such as color or behavior to identify authorized or unauthorized access.

Another potential mini-game concerning firewalls could be based on sorting. The player would be given "data packets" with specific physical characteristics, as well as whether or not they should be allowed through. They would then have filters they could use to sort the "packets" based on size, shape, color, or some other aspect. This could represent the firewall checking to see if access is authorized (or in this case the correct size, shape, etc.) before allowing it through.

5.2 Realism

Another way to proceed with this project is to make the mini-games and story elements more realistic. Currently, our game takes real issues, and abstracts them in order to make them more relatable and understandable to players that are not particularly knowledgeable. While this is useful for teaching, it may give players incorrect assumptions about how the attacks work and how to deal with them, especially if they take the abstractions literally. While the amount of realism in the game would depend on the intended audience, if this audience contains people with a greater knowledge of cybersecurity, more accurate details would likely make the game more enjoyable and edifying.

The most obvious way to improve realism would be to have one or more members of the project group majoring in cryptology or security. While our advisor, Lane Harrison, was a valuable source of knowledge about cybersecurity, having a member whose primary objective was bridging reality and game would allow better regulation of the realism. Additionally, if this member was able to simulate controlled attacks, more accurate data could be used.

Another method would be explaining more about how the abstractions represent the attacks, which would clarify the inaccuracies inherent in making them abstractions. As a teaching game for the basics rather than high-level security, this project was not intended to be exactly representative of attacks. The abstractions teach the basic idea behind the attack, but more detailed and extensive explanations of our methodology would cause fewer misunderstandings.

As for implementing these more realistic implementations into the mini-games, it would be impractical to have them in real-time. The DDOS mini-game, for instance, would be unplayable. Some DDOS attacks occur over the course of hours or even days. One survey found the average time of surveyed DDOS attacks to be 8.7 hours (SANS Institute, 2014). A mini-game that takes hours would be inconvenient, especially for someone who has a full-time job. One way around would be to use the game character to our benefit. The "game" character could warn you that you are about to be attacked, then go into the mini-game. There would be no end in sight, but after a certain time it would pause, and the character could say, "that's enough for now". Upon the game finishing, they could then explain how they skipped the majority of the attack, making it a playable experience. This would allow the abstraction to remain, while showing a more realistic version and allowing greater understanding of the abstraction.

5.3 Targeting Specific Audiences

Since the beginning of this project, the target audience was both a professional/corporate adult audience as well as the average adult that desires additional cybersecurity knowledge. By having our art resemble elements from our target groups' environment, we hoped it would help the player connect more easily to the characters in the game, especially if we used terminology they would already be familiar with.

Future iterations of the game could be targeted at specific individual companies. By changing the setting and various aspects of the story, it is possible to make the game world match the player's own environment. Any specific attacks or threats that a sponsor desires to be included can be easily added due to the modular nature of the code and assets of the game. Additional art or specific branding would need to be created by external groups if desired.

Chapter 6: Conclusion

Over the course of the past year, our team has created Lock_Out to teach players the basics of cybersecurity. It is our intention that they will be able to use this knowledge to identify and possibly defend against threats to themselves and their workplaces. The game uses an engaging storyline and enjoyable characters to keep players interested in the gameplay. This means that even players who might normally be unwilling to complete the game, due to what some may consider boring subject matter, will be more likely to complete the game.

Although there were some setbacks, our game has developed into a full thirty minute teaching experience. The game covers phishing, virus protection, proper anti-virus practices, safe security practices related to administrative access, and everyday password security and safety. The narrative in the game is designed as a mystery to keep players engaged in the subject matter. The abstractions of real world topics into pixelated mini-games provide an exciting way to learn about cybersecurity.

Our game discusses several hacks that are relatively common. However, it does not go into the technical details behind each. While part of this is to allow a broader audience to understand, enjoy, and learn, part of it is also because of the nature of these attacks. Cyber threats are constantly developing. So much so that if we were to create a game specifying details of how to counter these attacks on a technical level, it would be obsolete within a short amount of time. In order to have continuing relevancy, we used more abstract concepts to teach the theory behind the attacks, so they will be relevant longer. Additionally due to Lock_Out's design, it can be specialized for individual clients or for specific hacks that funders wish to be informed about. This is because a large amount of its content is modular, allowing for the easy addition of new, specific topics. While slightly more difficult, the story can also be modified to incorporate new narrative arcs, characters, and additional content.

Chapter 7: Credits

Design

Lead - Alex Hebert
Rosemary Lindsay, Chandler Reynolds, Kyle Stack

Story

Lead - Kyle Stack
Rosemary Lindsay, Alex Herbert

Programming

Alex Hebert

Art Design and Assets

World Assets:

Lead -Rosemary

Alex Hebert, Kyle Stack

Character assets:

Design - Joy

Kate Mary Alice Olguin, Alex Hebert

UI and splash screen assets:

Chandler Reynolds

Sound Design and Assets

Alex Hebert

Editing and Production Support

Professor Sheldon Lee

Professor Lane Harrison

Additional Information

Professor Lane Harrison

Alexander Wyglinski

Bibliography

Cornell University Law School. (n.d.). *U.S. Code › Title 18 › Part I › Chapter 47 › § 1030*. Retrieved from law.cornell.edu: https://www.law.cornell.edu/uscode/text/18/1030#a_1

HUAN-RONG TANG, R.-L. S.-Q. (2009). *Wireless Intrusion Detection for Defending Against TCP SYN Flooding Attack and Man-In-The-Middle Attack.* Retrieved from IEEE Xplore: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5212317

Hui Xu, Y. Z. (2012). *SpyAware: Investigating the Privacy Leakage Signatures in App Execution Traces.* Retrieved from IEEE Xplore: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7381828

Mauro Conti, N. D. (2016). *A Survey of Man In The Middle Attacks.* Retrieved from IEEE Xplore: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7442758

Mohamed Adel Sheta, K. A. (2016). *Anti-spyware Security Design Patterns.* Retrieved from IEEE Xplore: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7774822&tag=1

Oxford. (2016). Seven-layer reference model. In *A Dictionary of Computer Science (7. ed)* (pp. http://www.oxfordreference.com/view/10.1093/acref/9780199688975.001.0001/acref-9780199688975-e-4757#). Oxford University Press. Retrieved from http://www.oxfordreference.com/view/10.1093/acref/9780199688975.001.0001/acref-9780199688975-e-4757#

SANS Institute. (2014). *DDoS Attacks Advancing and Enduring: A SANS Survey.* Retrieved from sans.org: https://www.sans.org/reading-room/whitepapers/analyst/ddos-attacks-advancing-enduring-survey-34700

Sukhveer Kuar, K. K. (2016). *Implementing openflow based distributed firewall.* Retrieved from IEEE Xplore: http://ieeexplore.ieee.org/document/7857611/

Talal Alharbi, M. P. (2016). *SProxy ARP - efficient ARP handling in SDN.* Retrieved from IEEE Xplore: http://ieeexplore.ieee.org/document/7878805/

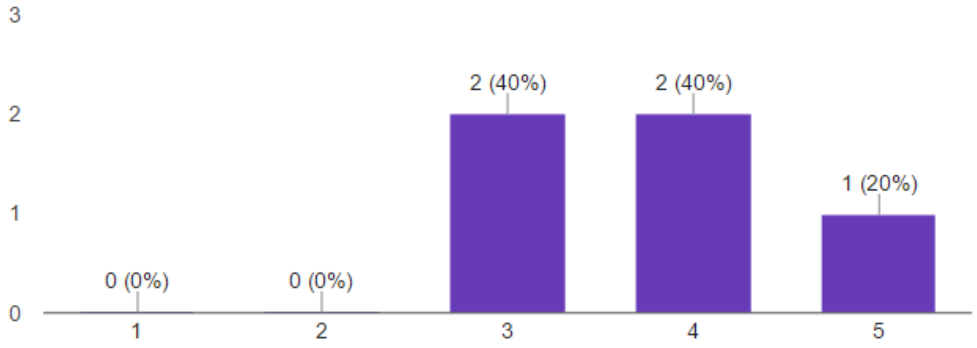
Vikas Kumar, S. C. (2012). *Detectino of Stealth Man-In-The-Middle Attack in Wireless LAN.* Retrieved from IEEE Xplore: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6449834

Ying Wu, Y. Q. (2012). *The 8-bit Parallel CRC-32 Research and Implementation in USB 3.0.* Retrieved from IEEE Xplore: http://ieeexplore.ieee.org/document/6394511/

Appendix 1: Playtesting Results

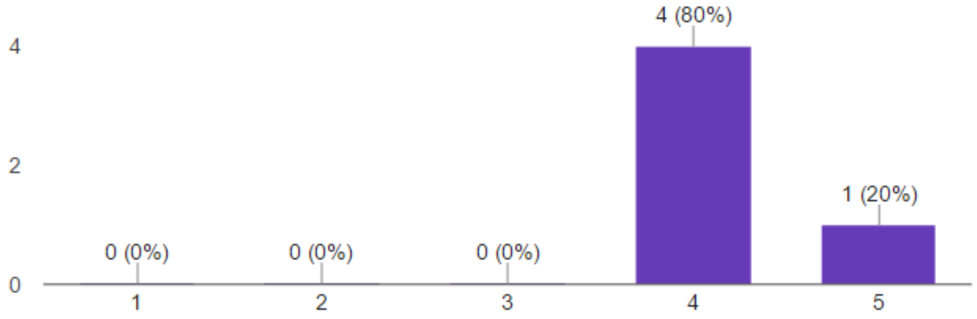
Before playing this game how would you rate your knowledge of cyber security?

(5 responses)

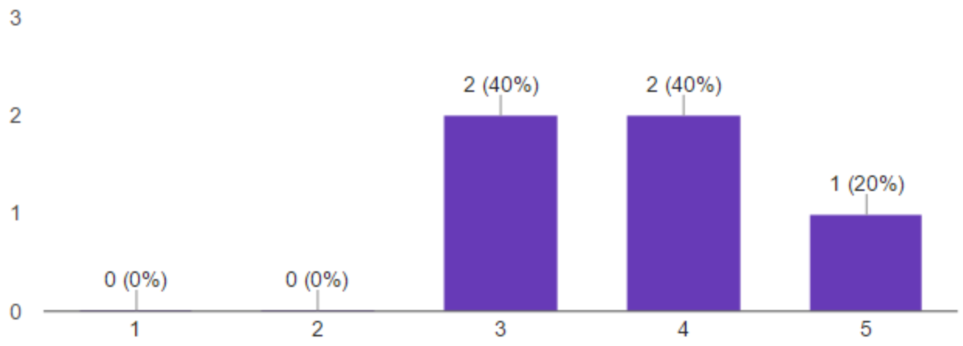


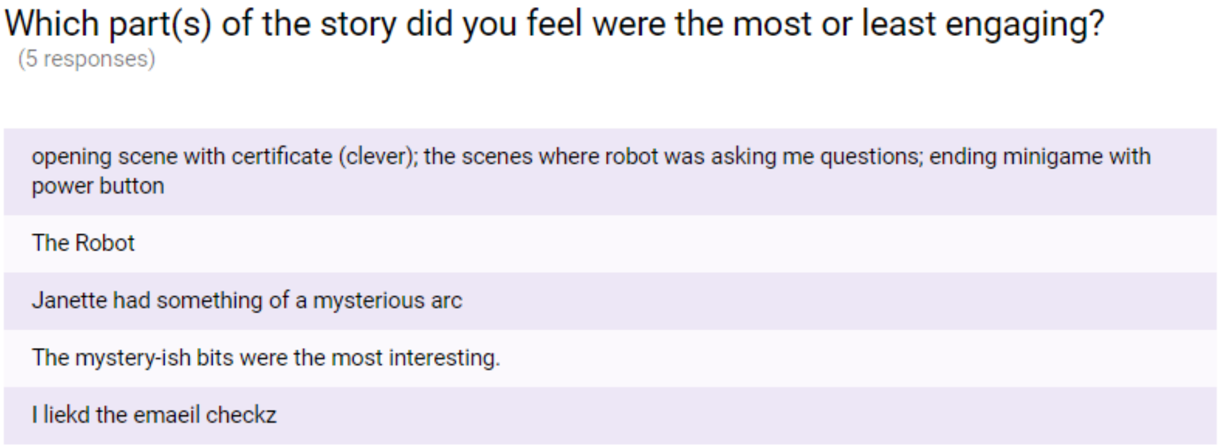
After playing this game how would you rate your knowledge of cyber security?

(5 responses)



How much did you enjoy this game? (5 responses)





Which part(s) of the story did you feel were the most or least engaging? (5 responses)

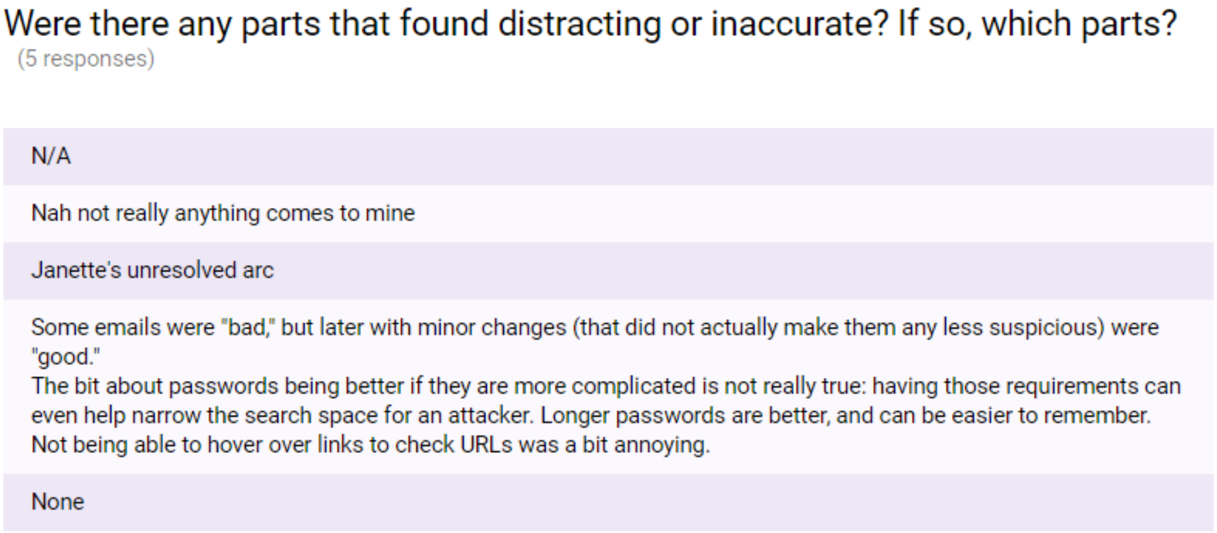
opening scene with certificate (clever); the scenes where robot was asking me questions; ending minigame with power button

The Robot

Janette had something of a mysterious arc

The mystery-ish bits were the most interesting.

I liekd the emaeil checkz



Were there any parts that found distracting or inaccurate? If so, which parts?

(5 responses)

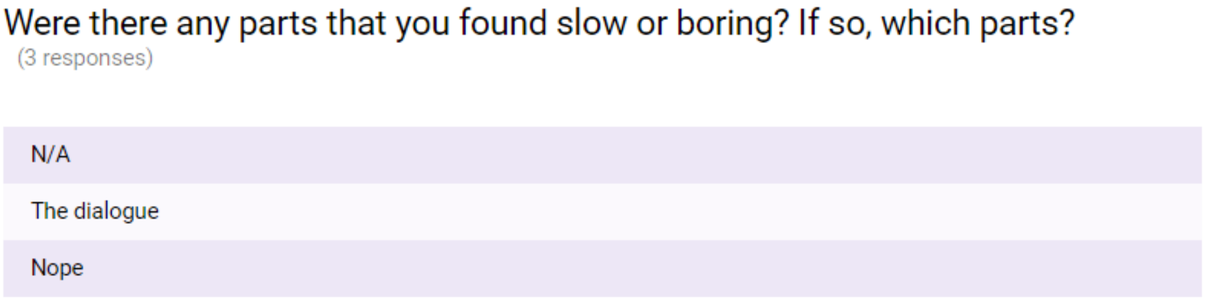
N/A

Nah not really anything comes to mine

Janette's unresolved arc

Some emails were "bad," but later with minor changes (that did not actually make them any less suspicious) were "good."
The bit about passwords being better if they are more complicated is not really true: having those requirements can even help narrow the search space for an attacker. Longer passwords are better, and can be easier to remember.
Not being able to hover over links to check URLs was a bit annoying.

None



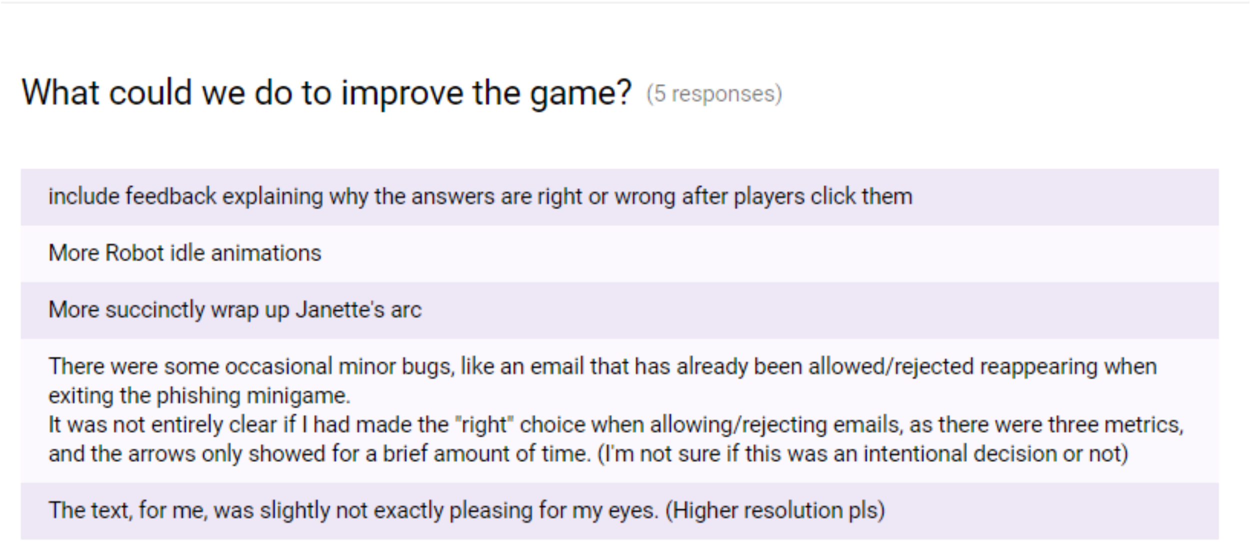
Were there any parts that you found slow or boring? If so, which parts?

(3 responses)

N/A

The dialogue

Nope



What could we do to improve the game? (5 responses)

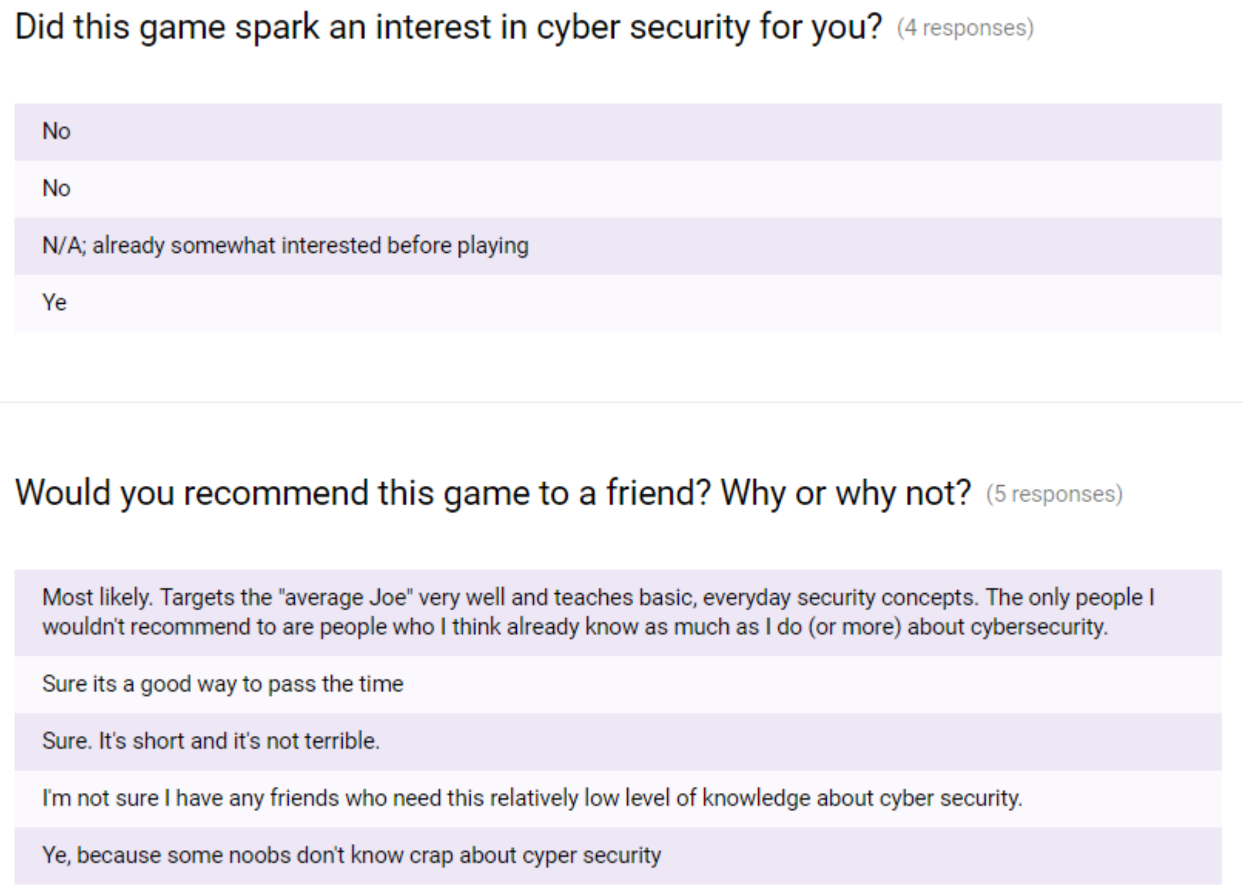
include feedback explaining why the answers are right or wrong after players click them

More Robot idle animations

More succinctly wrap up Janette's arc

There were some occasional minor bugs, like an email that has already been allowed/rejected reappearing when exiting the phishing minigame.
It was not entirely clear if I had made the "right" choice when allowing/rejecting emails, as there were three metrics, and the arrows only showed for a brief amount of time. (I'm not sure if this was an intentional decision or not)

The text, for me, was slightly not exactly pleasing for my eyes. (Higher resolution pls)



Did this game spark an interest in cyber security for you? (4 responses)

No

No

N/A; already somewhat interested before playing

Ye

Would you recommend this game to a friend? Why or why not? (5 responses)

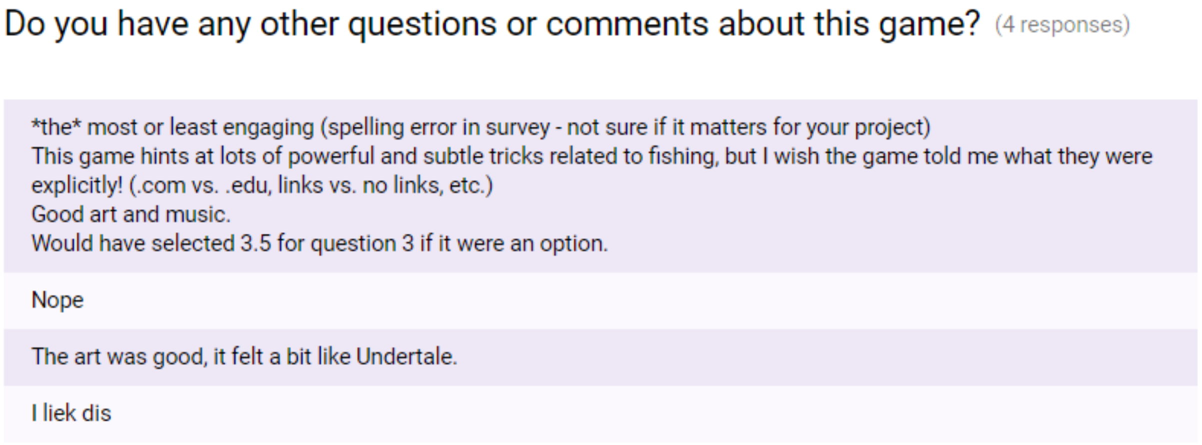
Most likely. Targets the "average Joe" very well and teaches basic, everyday security concepts. The only people I wouldn't recommend to are people who I think already know as much as I do (or more) about cybersecurity.

Sure its a good way to pass the time

Sure. It's short and it's not terrible.

I'm not sure I have any friends who need this relatively low level of knowledge about cyber security.

Ye, because some noobs don't know crap about cyper security



Do you have any other questions or comments about this game? (4 responses)

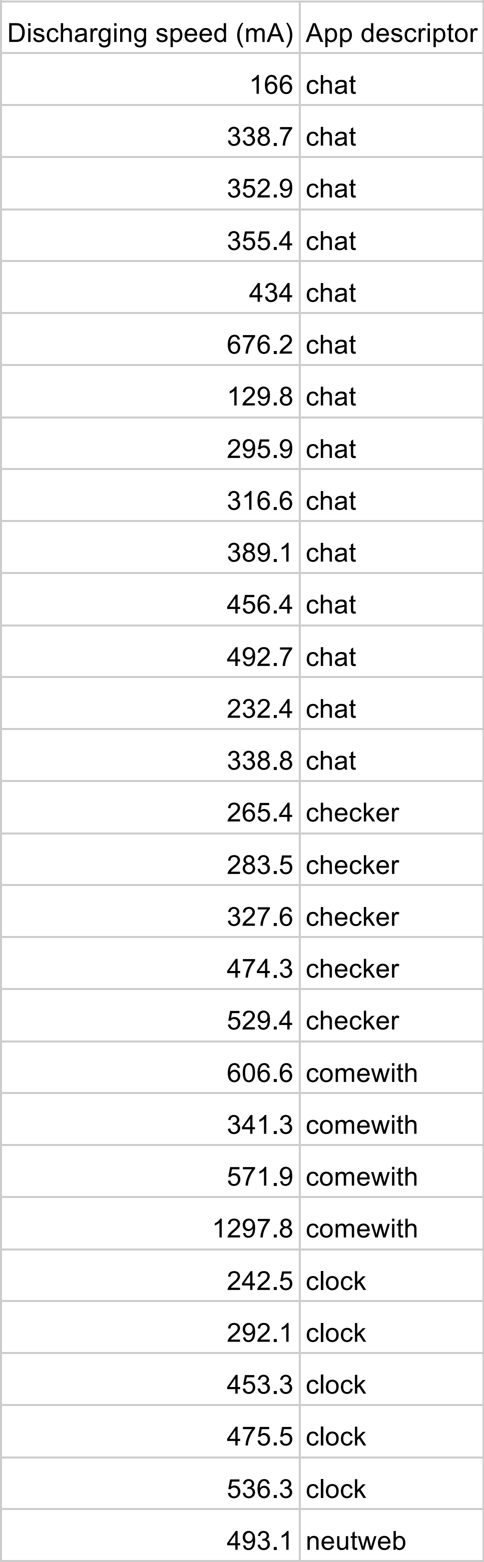
the most or least engaging (spelling error in survey - not sure if it matters for your project)
This game hints at lots of powerful and subtle tricks related to fishing, but I wish the game told me what they were explicitly! (.com vs. .edu, links vs. no links, etc.)
Good art and music.
Would have selected 3.5 for question 3 if it were an option.

Nope

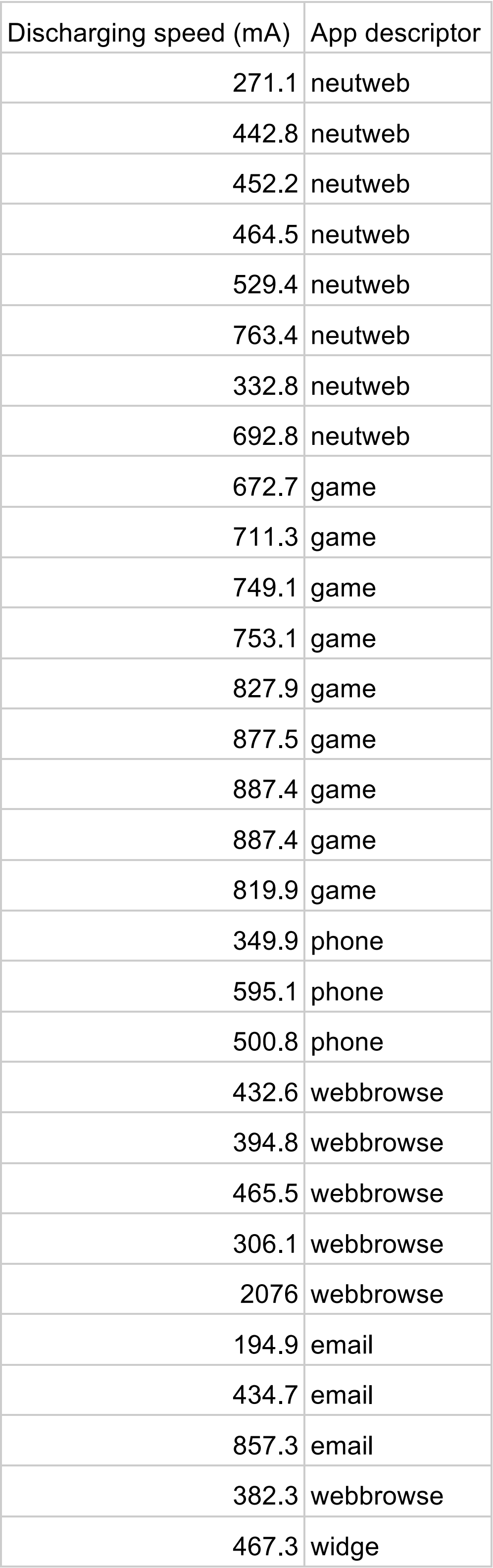
The art was good, it felt a bit like Undertale.

I liek dis

Appendix 2: Phone Discharge Data



Discharging speed (mA)	App descriptor
166	chat
338.7	chat
352.9	chat
355.4	chat
434	chat
676.2	chat
129.8	chat
295.9	chat
316.6	chat
389.1	chat
456.4	chat
492.7	chat
232.4	chat
338.8	chat
265.4	checker
283.5	checker
327.6	checker
474.3	checker
529.4	checker
606.6	comewith
341.3	comewith
571.9	comewith
1297.8	comewith
242.5	clock
292.1	clock
453.3	clock
475.5	clock
536.3	clock
493.1	neutweb



Discharging speed (mA)	App descriptor
271.1	neutweb
442.8	neutweb
452.2	neutweb
464.5	neutweb
529.4	neutweb
763.4	neutweb
332.8	neutweb
692.8	neutweb
672.7	game
711.3	game
749.1	game
753.1	game
827.9	game
877.5	game
887.4	game
887.4	game
819.9	game
349.9	phone
595.1	phone
500.8	phone
432.6	webbrowse
394.8	webbrowse
465.5	webbrowse
306.1	webbrowse
2076	webbrowse
194.9	email
434.7	email
857.3	email
382.3	webbrowse
467.3	widge