

COMPARATIVE ANALYSIS OF INTERFACE USABILITY FOR CYBERSECURITY
APPLICATIONS

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Wyly West Andrews

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science
Option:
Cybersecurity

August 2021

Fargo, North Dakota

North Dakota State University
Graduate School

Title

COMPARATIVE ANALYSIS OF INTERFACE USABILITY FOR
CYBERSECURITY APPLICATIONS

By

Wyly West Andrews

The Supervisory Committee certifies that this *disquisition* complies with
North Dakota State University's regulations and meets the accepted
standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Jeremy Straub

Chair

Stephanie Day

Kendall E. Nygard

Approved:

August 12, 2021

Date

Simone Ludwig

Department Chair

ABSTRACT

In cybersecurity, understanding the technologies and the best ways to interface with them is paramount for staying ahead of growing cyberthreats. Developers of cybersecurity software will benefit greatly from a greater understanding of how users prefer to interact with cybersecurity technology. In the modern world, two primary interface methods are currently used: the command-line interface (CLI) and the graphical user interface (GUI). This study is a survey and introspective into what benefits and drawbacks that each method has when in the hands of users who do not have a comprehensive background in cybersecurity. Untrained individuals showed proficiency when working with GUI systems, showing that developing modern cybersecurity systems with GUIs would improve ease of use for such individuals. Additionally, the CLI was favorable for more complex operations but was difficult for users who were not accustomed to the CLI.

ACKNOWLEDGEMENTS

I would like to thank so many people for helping me make this thesis a reality.

First, I would like to thank my family, for doing everything they can to support me and love me. You were there with me since the beginning. It's hard to stress how grateful I am for everything you've done. You have done so much for me, and I know that you would have done so much more to be there for me. I have the most amazing parents and the most amazing brother in the world.

I would also like to thank my girlfriend, for being there for me through many, many stressful nights. You provided me so much unconditional love and support. Knowing that you would be there for me every step of the way meant so much to me.

A huge thanks to my advisor, for pushing me to go further than I thought was possible and for the endless revisions and thesis changes that helped this thesis succeed.

Thank you, so much, for everyone that helped me get this far. Without your help, this would have never been possible. I am incredibly fortunate to have such amazing family and friends to support me.

TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGEMENTS.....	iv
LIST OF FIGURES	vii
LIST OF APPENDIX FIGURES.....	viii
1. INTRODUCTION	1
2. BACKGROUND	5
2.1. Cybersecurity	5
2.2. Gamification.....	7
2.3. Interface Methods.....	9
3. METHOD	12
3.1. Participants.....	12
3.2. Demographics.....	13
3.3. Procedure.....	13
3.4. Implementation.....	15
3.4.1. Kali Linux.....	15
3.4.2. Metasploit	16
3.4.3. Virtual Box	16
3.4.4. Metasploitable	17
3.4.5. Windows.....	17
3.4.6. Unity	17
3.4.7. Security Command	18
3.4.8. Fantasy Map Generator	19

3.5. Labs	20
3.6. Interview.....	21
3.6.1. Questions	21
3.6.2. Data Preparation for Analysis.....	21
4. ANALYSIS.....	22
4.1. Benefits and Drawbacks.....	23
4.2. Preferences on CLI and GUI.....	24
4.3. Unique Perspectives	25
5. CONCLUSION.....	28
6. FUTURE WORK.....	30
REFERENCES	33
APPENDIX A. FIGURES	38
APPENDIX B. USER INSTRUCTIONS.....	42
B.1. Lab One Instructions	42
B.2. Lab Two Instructions.....	43
APPENDIX C. PARTICIPANT INFORMATION SURVEY	44
APPENDIX D. INTERVIEW QUESTIONS	45

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. The Metasploit terminal with ASCII art of the Metasploit logo.....	16
2. Security Command showing various devices and designated target device to attack.	18
3. The example map used in Security Command.	20

LIST OF APPENDIX FIGURES

<u>Figure</u>	<u>Page</u>
A1. After running an Nmap scan on the target, the CLI shows a vulnerable port.....	38
A2. Executing a VSFTPD 2.3.4 backdoor opens a command shell on the target.	38
A3. Security Command showing the locations and affiliation of accessible devices.....	39
A4. Security Command screen showing a quick menu for immediate action support on a target device.	39
A5. A successful scan has been completed in Security Command.....	40
A6. A successful VSFTPD exploit is completed, and a shell is open on the target machine.	40
A7. A terminal can be opened through the Security Command application.	41

1. INTRODUCTION¹

Cybersecurity is a high-risk branch of computer science (Keskin et al., 2021). There is great risk when cybersecurity fails, whether the damages are financial, confidential, or political (King et al., 2018). As developers create new systems, databases, and algorithms, they must stay vigilant to keep these systems secure against malicious agents. Cybersecurity is an ever-increasing arms race against hactivists, script kiddies, cyber criminals, state-sponsored attackers, and even insiders looking to dismantle a system they had a hand in creating (Adams & Makramalla, 2015). Any application in the field of computer science has the potential to develop new vulnerabilities that need to be identified, understood, and resolved before malicious individuals can take advantage of them.

With an ever-present threat of attackers compromising valuable systems, cybersecurity professionals should consider taking precautions to improve the security of these systems. The development of informative and effective cybersecurity tools could benefit professionals in this regard. Cybersecurity professionals are asked to prepare secure systems, to act quickly to shut down ongoing attacks, and to mitigate damage of exploited vulnerabilities.

Building the powerful cybersecurity tools necessary to provide all the support and data needed by a cybersecurity specialist is no small undertaking. Improving cybersecurity tools to prepare an expert more effectively with all available information could provide safer cybersecurity systems by better preparing the expert to react to the situation if the need arises. If

¹ This thesis draws from and is based on a paper titled "Analysis of Interface Usability Enhancement to Enable Low Skill Staff Service in Critical Cybersecurity Positions" which is under preparation for submission to Technologies. The material in this chapter was co-authored by Wyly Andrews and Jeremy Straub. Both Wyly Andrews and Jeremy Straub were involved in the conceptualization of this project. Wyly Andrews the had primary responsibility for software development, and data collection. Wyly Andrews also wrote the initial draft of this chapter. Jeremy Straub provided feedback on and corrections to the chapter.

the expert has access to all available information, they can make informed decisions to secure the system and mitigate losses.

Now, the question is, how can developers improve cybersecurity tools? Similar studies have implemented techniques pulled from other fields in computer science. Virtual reality (VR) has been investigated for improving the interface experience by augmenting the user's view and interaction (Tipparach, 2019). With advancements in quantum computing, quantum computers may shape how cybersecurity specialists approach and solve cybersecurity problems. With their incredibly fast processing times, quantum computers can drastically improve decryption techniques (Wadhwa, 2015). These advanced technologies can potentially improve how to tackle cybersecurity problems, but a simpler method may exist in the way a user interacts with cybersecurity programs.

Throughout all of computing history, work has been put into developing better interface methods for users to interact with computers. Many changes have been implemented in user interface design since the inception of computer science. Even now, new methods are being researched to hopefully push the field further and better connect users to computing devices. Examples of this include virtual reality (Tipparach, 2019) or radical atoms, an interface where all information can be interacted with physically (Ishii et al., 2012). However, of the methods currently implemented and used, no one method is universally regarded as the best. (Afinogenov, 2003; Computer Hope, 2020)

This study aims to evaluate on the benefits and drawbacks of the command-line interface (CLI) method and the graphical user interface (GUI) method as they relate to cybersecurity for untrained users. These two methods see popular use in modern cybersecurity tools. By investigating the effectiveness of these methods when in the hands of untrained individuals,

researchers can learn what benefits and drawbacks each provides, as well as the unique perspectives of individuals working with these interfaces. A comparison investigation like this will help better gauge the potential of implementing these interface methods in live cybersecurity operations. Furthermore, this study can potentially drive greater interest for novice users into cybersecurity roles, by improving the experience of operating a cyberattack or defense. There is currently a cybersecurity skills gap (*Cyberseek*), which could be reduced by making cybersecurity tools more accessible.

By giving users with little history in cybersecurity first-hand experience with both CLI and GUI interface methods, this study seeks to reveal the opinions of this type of user and determine what potential benefits each method can provide. The overarching question this study hopes to answer is which method would better be suited for the development of cybersecurity applications to better cater to untrained individuals.

This study took student volunteers and asked them to complete a short cybersecurity attack in a CLI and then in a GUI. The Metasploit Framework, a popular open-source penetration testing tool, was used for the CLI. An application called Security Command was developed for this study to be used as the GUI example. Participants were asked to try each interface and compare the two for their benefits and drawbacks.

Afterwards, this study analyzed the participants' feedback for shared opinions on these interfaces. Neither interface showed to be completely superior over the other when used in the given scenario, but there was moderate support for the GUI being more user-friendly while the CLI showed to be more informative. This study discusses the results further in-depth.

This study hopes to push the field of computer science by investigating which interface inexperienced users prefer. By studying user interfaces, researchers can better ascertain how

users interact with computer systems, which will help computer science developers build better interfaces. Improving user interfaces will allow computer systems to be operated at their full potential, otherwise that potential would be wasted. Since an interface is the point of interaction for a user with a computer, studying user interfaces means studying users and how they respond to different interfaces.

2. BACKGROUND²

There are three main focus areas that this research draws from: cybersecurity, gamification, and interface methods. This study seeks to see how these three focuses interconnect to provide for better cybersecurity tool implementation. By implementing gamification into interface design and viewing these concepts under the lens of cybersecurity, hopefully better tools can be made to assist cybersecurity operatives more effectively in their work. Prior work in each of the three areas is now discussed.

2.1. Cybersecurity

Cybersecurity is a broad term that Craigen, et al. (2014) suggests, “is the organization and collection of resources, processes, and structures used to protect cyberspace and cyberspace-enabled systems.” Today, there are many varieties of threats to cybersecurity systems, malicious and otherwise. Understanding these threats prepares an expert to prevent disasters from happening and mitigate the damage when they do strike. A botnet or hactivist group could cause massive system outages by create a distributed denial of service (DDOS) attack. An insider could cause a data breach, compromising personal information or secrets not meant to be publicized (Adams & Makramalla, 2015). Natural disasters, too, could also damage a system physically, causing all sorts of data loss, power outages, or leave systems vulnerable for more malicious attacks (Fekete & Rhyner, 2020). Disasters come in all shapes and forms, so better preparing a cybersecurity professional to tackle these problems will make systems more safe and secure.

² This thesis draws from and is based on a paper titled "Analysis of Interface Usability Enhancement to Enable Low Skill Staff Service in Critical Cybersecurity Positions" which is under preparation for submission to Technologies. The material in this chapter was co-authored by Wyly Andrews and Jeremy Straub. Both Wyly Andrews and Jeremy Straub were involved in the conceptualization of this project. Wyly Andrews the had primary responsibility for software development, and data collection. Wyly Andrews also wrote the initial draft of this chapter. Jeremy Straub provided feedback on and corrections to the chapter.

In the 1980s, computer networks went global, beginning a new age of international cyber threats (Warner, 2012). As these global networks grew, threats could appear in more obscure corners of the world creating a growing danger of malicious individuals breaching sensitive systems. Prior to the 1990s, however, the public was not fully aware of cybersecurity and the dangers of global cyber growth (Warner, 2012). In the 1990s, internet governance started taking shape, leading to today's public cybersecurity infrastructure (Fidler, 2017).

Early on, the potential for cyberthreats was unknown. It was not clear what dangers could be caused by the obscure field of computing, which was still in its nascency. In 1997, the US Defense Department undertook a classified exercise to gauge the potential of danger in malicious cyberattacks. This exercise revealed great concerns about the US's nascent cybersecurity program (Martelle, 2018). ER97 Red Team Chief Targeting Officer Keith Abernethy expressed his grave concern about the ease in which the Red Team, or mock attacking team, had overwhelmed the Blue Team, or defending team. As he elaborates, "we had the Blue Team on the run by the third day of the actual exercise" (Martelle, 2018). This exercise helped outline the true danger that cybersecurity threats could pose in a world where everyone has access to a global network.

There are many different attack techniques in cybersecurity. Spamming, distributed denial of service (DDOS) attacks, and search poisoning all showcase powerful examples of executing cyberattacks through the web en masse (Mahmood & Afzal, 2013). Man in the middle (MitM), backdoor injection, and targeted brute force attacks are examples of cyberattacks that are much more targeted towards one system or communication (Singh et al., 2016).

Conversely, there are many ways to counter these attacks. Firewalls, proper corporate anti-phishing solutions, and content filters can cut down on a lot of incoming attacks (Mahmood

& Afzal, 2013). There is no single way to defend against all these attacks, but various methods can slow down, prevent, or detect these attacks when they occur (Singh et al., 2016).

2.2. Gamification

The definition of gamification ranges wildly, but one definition calls gamification “a social scientific, post-positivist subdiscipline of game science that explores the various design techniques, and related concerns, that can be used to add game elements to existing real-world processes” (Landers et al., 2018). Games are fun, entertaining, and engage users. Work is perceived as a societal obligation, a rigid job of task completion. However, by mixing work and play, gamification intends to improve the user experience in a piece of software or an environment by incorporating elements of games that engage users and improve retention.

The application of gamification predates formalized research on this topic and has been used in various forms over the past century (Growth Engineering, 2019). The Boy Scouts of America have been using badges and ranks to incentivize scouts since 1910 (Growth Engineering, 2019). Gamification began appearing academically, however, in the past 40 years (Growth Engineering, 2019). Although the term itself was not used until 2002 (Pelling, 2017), the application of game-like elements into non-game areas began being studied much earlier. Early academic papers related to this idea were written closer to 1980 (Townsend, 2019). One study by Malone (1981) hoped to answer two fundamental questions that would fit well into the realm of gamification today: “why are computer games so captivating” and “how can the features that make computer games captivating be used to make learning—especially learning with computers—interesting” (Malone, 1981). Understanding these questions will help better prepare someone to understand the intent of gamification’s role in modern software.

The term gamification itself can be traced back to Pelling in 2002. Pelling claimed founding the term in a 2011 blog post, describing gamification as a way of improving interfaces for commercial electronic devices (Pelling, 2017). The term remained obscure until 2010, when it received a spike in popularity after Mangalindan reintroduced the term in a Forbes article. Mangalindan reinvented the term to mean replicating game mechanics used to hook users for uses in other contexts, like business or healthcare (Mangalindan, 2010). Mangalindan's version of the term persists to be the modern-day version used academically and is the version used in this paper.

Gamification has already been implemented in many fields (Çeker & Özdamlı, 2017), including outside of computer science. In the medical field, gamification has been used as a tool for improving physical and mental health. For example, one study shows that gamification in smoking cessation mobile apps has improved motivation and engagement in users over similar, non-gamified apps (Rajani et al., 2019). Research into implementing gamification in crowdfunding has led to increase in participant engagement (Styles, 2018). Gamification has also appeared in education (Youssef, 2015), business, banking, and commerce sectors (Çeker & Özdamlı, 2017).

Gamification has showed benefits in the fields and industries where it has been implemented (Rajani et al., 2019; Styles, 2018). The current study is partially an investigation on implementing gamification into cybersecurity. If gamification can aide in user engagement and involvement in other contexts, then surely it is possible to extend gamification to the cybersecurity specialist's repertoire of tools. Gamification has already seen some use in the cybersecurity sector. In network security, gamification has been implemented to help reveal malicious attackers by deceiving them into believing they are being successful in their attacks

(Bellekens et al., 2019). Additionally, gamification has been used to improve cyberattack simulations to better prepare organizations in the event of a real attack (Adams & Makramalla, 2015).

2.3. Interface Methods

Humans interfacing with a computer has always been a challenge in software development (Fellmann & Kavakli, 2007). The key question of “what is the best way for a user to interact with a system” has had many answers. Many different methods of interaction have been developed since the beginning of this field. One early method is the command-line interface (CLI). The command-line interface is a completely text-based interface where all interactions between the computer and the user are done through text. The computer displays information through text to the user to read from a monitor, and the user can respond by inputting text commands through a keyboard (Hultstrand & Olofsson, 2015).

Later, the development of the direct manipulation interface method emerged, providing users with extended interaction using a mouse or joystick. This method, modernly known as the graphical user interface (GUI), was used in Apple Macintosh products in the 1980s (Friedman, 1997). The popularity of the GUI further skyrocketed into popularity through the “commercial of the decade” (*Advertising Age*). Apple’s 1984 Super Bowl ad referencing the book *1984* urged computer users to stop viewing computers as tools and start viewing them as empowering devices to combat conformity (Friedman, 1997). This advertisement boosted Macintosh’s success where the failed Lisa, the first of Apple’s mouse-implemented computers, could not. With the success of the Macintosh, the GUI saw mainstream popularity, while also introducing windows, icons, menus, and pointer (WIMP) concepts to the PC personal-use audience. Hazari

and Reaves (1994) showed that novices could adapt themselves to the GUI more easily than they could to a CLI.

An often-debated consideration between interaction methods is the user-friendliness of the interface. This term can cause frustration, because for an interface to be user-friendly, it must have other benefits that a user can appreciate. In this case, the term user-friendly is completely redundant. User-friendliness shifts from user to user and can even change in a single user as time goes on. Before the arrival of Macintosh, users would certainly not describe the GUI as user-friendly when most users were familiar with MS-DOS and its CLI implementation (Afinogenov, 2003). Today, the GUI has become a familiar interface method for many users, but forty years ago (Friedman, 1997) or forty years from now, that may not be the case. A careful lens should be used to examine an interface's user-friendliness, as not a quality of the interface itself, but how current society connects with it. In fact, a further step can be taken to examine any interaction between user and an interface as not separate entities, but how both user and interface influence each other.

In the future, there may be a shift in the predominant interface method. Virtual Reality is one such technology that has the capability to enhance the cybersecurity interface. Virtual reality has already entered the medical (Couperus et al., 2020) and entertainment fields. VR has shown promising potential in the application of a cybersecurity interface, especially in the hands of younger generations. A 2017 study by Protectwise showed that 77% of millennials and post-millennials said, "they would get more enjoyment from using VR-based tools than from using desktop-based tools," and that 74% said "the presence of VR tools would increase their likelihood of pursuing cybersecurity careers" (Technologies, 2017).

Other interface methods also exist, such as the tangible user interface (TUI). This method is characterized by implementing physical objects to ground interactions to reality, driving ease of use for beginners. The implementation of the mouse in modern computer design is an example of a TUI (Ishii et al., 2012). This implementation only partially implements digital data physically; even though the user is physically interacting with the mouse, most of the data is being displayed digitally through a GUI or CLI. The idea of a totally tangible interface, where all the data can be dynamically interacted with in the physical realm, is only hypothetical at this point, but has been called a “radical atoms” approach (Ishii et al., 2012). Neither virtual reality nor radical atoms is investigated in this study, but each has the potential to shape interface design in the future.

3. METHOD³

This study is a survey of untrained individuals for their opinions on CLI and GUI for use in cyberattack tasks. The participants were exposed to each method, were asked to complete a short cybersecurity task, and then were interviewed on their experiences. The following sections detail the methodology of how the participants were selected and the procedure they were asked to complete.

3.1. Participants

Fifteen participants participated in this study. The number of participants was driven by budgeting limitations as well as to allow the interviewer one-on-one time with each participant and to provide time for participants to experience both CLI and GUI labs under COVID-19 guidelines.

Participants were recruited via email to North Dakota State University's (NDSU) research participants mailing list, a listserv for research study recruitment. This listserv contacts students, staff, and faculty across all NDSU departments, including both undergraduate and graduate students. Participants, thus, were not limited to computer science or computer engineering students but included students from many different majors. Since more than fifteen students volunteered, participants were selected in the order they volunteered and were skipped over, as needed, if they did not respond to attempts to contact them. Participants were offered a \$15 gift card for their participation.

³ This thesis draws from and is based on a paper titled "Analysis of Interface Usability Enhancement to Enable Low Skill Staff Service in Critical Cybersecurity Positions" which is under preparation for submission to Technologies. The material in this chapter was co-authored by Wyly Andrews and Jeremy Straub. Both Wyly Andrews and Jeremy Straub were involved in the conceptualization of this project. Wyly Andrews had primary responsibility for software development, and data collection. Wyly Andrews also wrote the initial draft of this chapter. Jeremy Straub provided feedback on and corrections to the chapter.

3.2. Demographics

Nine of the fifteen participants were female, and six were male. All participants were students at NDSU. Three of these participants were pursuing a major in computer science or computer engineering. Only one participant had any prior experience in the military. On average, participants self-rated their computer technology experience levels at 4.6 out of 9, on a scale ranging from 1 – novice to 9 – expert. The lowest participant rated himself at 2, while the highest participant rated himself at 7. Nine participants were pursuing a minor, but none of these minors were computer science or computer engineering focuses.

Interestingly, of the sixty-one people who volunteered for this study, thirty-nine of them were female, and only twenty were male (two of the volunteers had unisex names). In 2019, only 27% of science, technology, engineering, and math (STEM) workers in the US are women (Martinez & Christnacht, 2021). It is, thus, surprising to see 64% of volunteers to be female, considering this employment gender gap. It was expected that the gender percentage of volunteers would be closer to the 27% number, matching the proportion of STEM workers. This statistic could indicate a disparity between the percentage of women that are interested in STEM and the percentage of women that end up in the field. Cybersecurity needs more women professionals, as only 11% of cybersecurity professionals are women (Poster, 2018). As this research attracted a high percentage of women, maybe further research into user interfaces can drive greater female interest in joining the cybersecurity workforce.

3.3. Procedure

For the study, participants were asked to complete two short, cybersecurity labs followed by an in-person, audio-recorded and transcribed interview. The first lab had the participants complete a basic Metasploit attack process through a terminal. This lab was setup to mimic a

traditional approach at a cyberattack. The participants were given a step-by-step process on how to enter the Metasploit Framework console, run an Nmap scan against the target machine (a Metasploitable virtual machine), execute a vsftpd 2.3.4 backdoor attack, and verify that access has been granted. Participants were encouraged to ask for help when needed, but almost all the participants were able to complete the exercise without intervention based on the instructions provided.

The second lab had the participants complete a very similar process using an in-house developed graphical user interface. The graphical user interface, called Security Command, was developed to contrast to the terminal-approach that the Metasploit Frame console implements. The participants were asked to follow a similar step-by-step process from the first lab. Instead of terminal commands to input, the participants were given icons, buttons, and drop-downs to interact with to complete the attack.

Both labs were setup to be as similar as possible to avoid bias. In both labs, participants were asked to complete the same attack. The steps to complete the lab were also setup to be as similar as possible (scan the target, setup the attack, and execute the exploit). Lab instructions can be viewed in Appendix B. The participants were asked to do the second lab immediately after the first lab in the same room. Since the GUI was not connected to a Metasploitable VM, like the CLI was, artificial pauses were put into the GUI to make the execute time similar between the two labs.

When the participants completed the labs, they were then interviewed. The interview consisted of eleven questions and averaged about five to ten minutes per participant. Each participant was asked to compare and contrast the two labs. Each participant was asked to identify which method of attack, command-line or graphical user interface, that they would

prefer to use if given the choice. Finally, participants were asked if they would feel comfortable using their preferred method for up to seven or eight hours per day. These questions were asked to better understand how different people with different levels of cybersecurity experience react to working in both environments. The interview script is included in Appendix D.

The study took place over a two-week period. Each participant was given a half hour session to complete both labs and the interview. The participants were asked to attend an NDSU lab in-person at their designated session time. This study was conducted in-person as a cybersecurity precaution. By conducting this study in-person, the labs were able to be disconnected from the internet and air-gapped from any other device on NDSU's campus network.

3.4. Implementation

The software used for this study is detailed below. This includes operating system software, development software, or cybersecurity software used to perform the labs.

3.4.1. Kali Linux

Lab one was run on a Kali Linux distribution. Kali Linux was chosen for its cybersecurity tools and offensive security capabilities (*Kali.org*, n.d.). Metasploit, a major focus of this study, comes pre-installed on the Kali Linux distribution. Kali Linux was chosen over Windows because the only functionality required for lab one was Metasploit and Metasploitable. While it is possible to run both Metasploit and Metasploitable on a Windows device, Windows Defender causes problems and generates error messages when detecting the vulnerabilities from the Metasploitable Virtual Machine. Once the Metasploitable machine was downloaded to the Kali Linux machine, the machine was taken off the internet for network security reasons.

Metasploit tool as a separate device, even though Metasploitable is being hosted on the same machine.

3.4.4. Metasploitable

Metasploitable is an intentionally vulnerable machine that was developed for use in practice cyberattacks (*Metasploitable 2 Exploitability Guide*, n.d.). As its name suggests, Metasploitable can be configured to test many of Metasploit's built-in cyberattacks. Metasploitable provides researchers and ethical hackers with a vulnerable target for testing and practicing attacks, without worrying about legal or other repercussions. For this lab, Metasploitable was left in its default configuration, which leaves it vulnerable for the vsftpd 2.3.4 backdoor exploit.

3.4.5. Windows

Lab two was run on a Windows 10 computer. Unlike lab one, lab two does not need any built-in cybersecurity tools. Instead of setting up Metasploitable, lab two had a mock backend where Metasploit and Metasploitable would go. Since the Security Command GUI that's the focus of this lab automatically converts the user's actions to Metasploit commands, setting up Metasploit and Metasploitable again on the lab two machine would be redundant.

3.4.6. Unity

Unity is, traditionally, a game development engine that sees popular use for many genres of videogames, from 2D platforms to 3D shooters (Unity Technologies, n.d.). Unity is written in C++ with a C# scripting API. Security Command was developed in Unity because Unity provides the tools for user interface development as well as backend scripting support through C#. This allows for the full suite of user interface tools commonly used for friendly user

interface design, while also providing for backend support that can be used for the more complicated sections of cybersecurity code.

3.4.7. Security Command

Security Command is a custom-built graphical user interface developed for this research project. Security Command is a cybersecurity application that simulates the steps necessary to execute cybersecurity attacks. The program is setup to mimic a possible real-world cybersecurity-operations application. It was developed with a focus on providing a user-friendly interface that can relay specific, important information to the user so the user can make immediate, informed decisions on conducting real-time cybersecurity operations. Currently, Security Command implements the basics for performing reconnaissance and executing exploits but does not implement any defensive cybersecurity operations. The Security Command user interface is depicted in Figure 2.

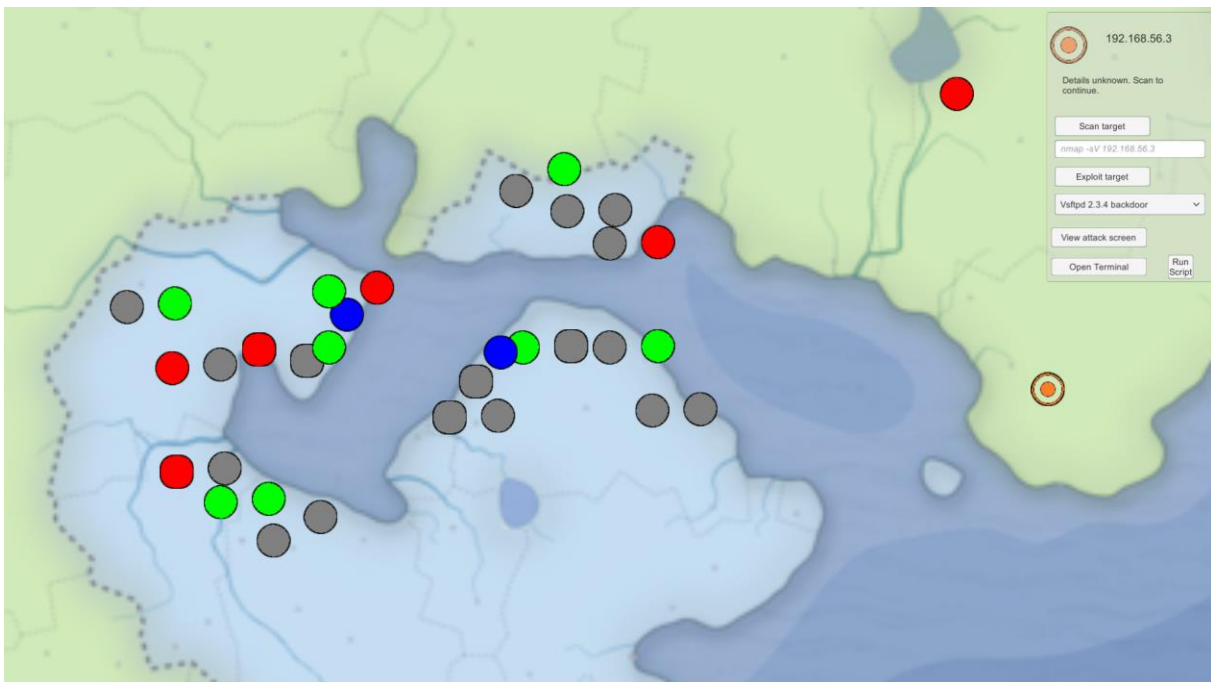


Figure 2. Security Command showing various devices and designated target device to attack.

The major focus of developing Security Command was making a graphical user interface that could serve as an alternative to the traditional command-line approach. By focusing on keeping a simple, clean interface with less clutter for the user to focus on, an introductory user into the realm of cybersecurity can better organize the information than that same user can garner from a command-line counter system. Great care must be taken in creating this sort of software, because efforts to reduce clutter can potentially reduce functionality, while trying to provide more information to the user might distract him/her from the important, pertinent information necessary to make decisions in a hurry.

Security Command was custom-built for the purpose of this study, so the software only has the functionality to execute a vsftpd 2.3.4 backdoor exploit. This could unfairly put the GUIs in a more positive light, as a more complete, feature-heavy application could add more clutter that could distract or confuse a user. However, developing Security Command in Unity provides for a chance to implement a more gamified user interface for beginner users to execute cybersecurity commands. Comparisons could also benefit from using Armitage, which is a GUI designed with Metasploit in mind.

3.4.8. Fantasy Map Generator

Fantasy Map Generator is an open-source, free-to-use map generator used to fabricate highly detailed, randomly generated maps (Azgaar, n.d.). This web application was developed on GitHub and is hosted publicly at <https://azgaar.github.io/Fantasy-Map-Generator>. Security Command uses a generated map from this application to simulate how the world view would look like, without using real-world examples.



Figure 3. The example map used in Security Command.

3.5. Labs

Before the interview, participants were asked to complete two labs. In both labs, participants were given step-by-step instructions on completing a vsftpd 2.3.4 backdoor exploit, a very simple exploit available in Metasploit. The first lab asked the participants to complete this exploit on a CLI, using Metasploit's attack terminal on a Kali Linux machine. The second lab asked the participants to complete the same exploit but on a GUI, using Security Command on a Windows machine. The instructions for each lab can be found in Appendix B.

Each lab was made to be as similar to each other, except for the interface method. In each lab, participants were asked to follow the instructions given to the best of their ability. If the participants ever got lost or ran into problems, they were encouraged to ask for assistance. Each lab would be considered completed when the participant successfully executed the exploit.

3.6. Interview

Participants were asked to complete an interview upon completing both labs. The interview was setup to gather participants' opinions on the labs and to see what benefits and drawbacks each method of interface had. The interview script is included in Appendix D.

3.6.1. Questions

The interview consisted of eleven questions. Participants were asked to take about fifteen minutes to complete the interview, but most participants were finished in five to ten minutes. In addition to the question prompts, follow up questions were also utilized. For example, if a participant indicated background or career plans in cybersecurity, they were asked follow-up questions to elaborate on their experiences. Participants were also asked to elaborate on why or why not they would feel comfortable using their preferred method for an extended period.

3.6.2. Data Preparation for Analysis

The interview was audio recorded and was then processed to produce automatically generated transcripts. The transcripts were compared against the original recordings to correct any mistakes that occurred when the transcripts were generated. Once the transcripts were verified as complete and accurate, the audio recordings of the interviews were destroyed. All analysis was conducted on the transcripts.

Qualitative analysis coding was applied to the participants' responses to get an understanding of what the participants think about using the command-line versus the graphical user interface. Participants were also asked to fill out a demographic information form which was used alongside their responses to find any trends. This demographic information form is included in Appendix C.

4. ANALYSIS⁴

The interview responses provided several different perspectives on the interface methods. All participants were able to complete both labs. However, a few participants requested help during the command-line interface lab. Issues typically arose from mistyping commands (like missing spaces) or getting confused as to which step of the process to be on (for example, trying to execute an exploit without setting a target). Both types of issues likely occurred because of the subjects' inexperience with working with command-line interfaces or because the participants were working with Metasploit for the first time. No similar requests were made using the GUI system.

It is important to note that most of the participants had little to no experience in cybersecurity. Out of the fifteen participants in the study, eleven of the participants rated themselves at a five or lower in experience level in computer technology (on a scale from one to nine, with one indicating novice level and nine indicating expert level). Only one participant out of all fifteen had any cybersecurity experience before participating in this study. This participant, participant eight, marked himself as having the most cybersecurity experience (at a level of seven out of nine). Thus, when analyzing the data, it was important to keep in mind that the collected data captures the opinions of people who are inexperienced in cybersecurity. People who have more cybersecurity experience may have different opinions about their preferred interface method. Creating cybersecurity systems for inexperienced users can still provide many

⁴ This thesis draws from and is based on a paper titled "Analysis of Interface Usability Enhancement to Enable Low Skill Staff Service in Critical Cybersecurity Positions" which is under preparation for submission to Technologies. The material in this chapter was co-authored by Wyly Andrews and Jeremy Straub. Both Wyly Andrews and Jeremy Straub were involved in the conceptualization of this project. Wyly Andrews had primary responsibility for software development, and data collection. Wyly Andrews also wrote the initial draft of this chapter. Jeremy Straub provided feedback on and corrections to the chapter.

benefits, as this allows a wider audience access to perform cyberattack or cyberdefense operations. Cybersecurity software could become more accessible to the general public if GUIs were used for more systems.

4.1. Benefits and Drawbacks

One focus of this study is discovering what benefits and drawbacks the participants recognized in both the CLI and the GUI approaches. When asked to compare the GUI and the CLI, the responses from the participants varied. However, there were very few contradictory opinions from the interviews. Almost all the participants agreed that the GUI was a simpler, easier method, and a few said that it provides a more beginner friendly experience. Fourteen of the fifteen participants either described the GUI as simpler or easier than the CLI. In fact, a couple of the participants remarked on how the GUI could be considered too easy to use; as participant eight puts it, “the term script kiddie also comes to mind. That [the GUI is] too easy for amateur hackers to take advantage of without fully understanding what they're doing, how it works, and the risks involved with using it.” Note that the participant refers to script kiddies, which are amateur users nefariously using hacking software developed by more proficient individuals.

This design allows the GUI to perform faster for the short tasks like those that the participants were asked to do. Eight of the participants indicated as such, indicating speed as one of the benefits that the GUI provides over the CLI. Participant fifteen had an interesting take on this matter: that the simplicity of the GUI lends to faster execution time, but only for beginner users. The CLI would, conversely, prove to be the faster method of the two for a skilled individual. “In the same way that the manual attack is benefited by the skill of the user, [the GUI] is - in the hands of a skilled user - would feel almost too slow, like, too limited.” It would

be interesting to test this idea in the future, by asking both skilled and unskilled individuals to perform both simple and complex cybersecurity tasks with the two interfaces in the future.

Even though the manual method was considered to be the more difficult interface method, the participants still recognized benefits from it. Nine of the fifteen participants recognized the CLI method to have more options or flexibility for the user to work with, especially for more experienced users. As participant fourteen put it, “I’d say the [CLI] attack offers more specificity in exactly what you want to do and knowing that what you want to do is going to go through because of how basic and down to earth the monitor is. Yeah, I’d say it’s pretty beneficial.” Participant seven expands on this: “[In the GUI], you get the flip of a coin of having able to do this done fast and being able to do this effectively, but on the off chance you lose a lot of information that you could use in case something is rerouted back to you as a result.”

4.2. Preferences on CLI and GUI

Many of the participants of this study indicated a low level of computer technology experience. After asking participants what benefits and drawbacks they see with each interface method, participants were asked to identify which method they would prefer, and whether they could recognize a situation where their non-preferred method would be preferred. Out of all fifteen participants, thirteen of them said their preferred method would be the GUI method. Even though they said they would prefer using the CLI method, the two other participants indicated that there were times when the GUI would be preferred: specifically for shorter, smaller tasks. As one of these two participants described, “if it was for just a one-time quick security scan, say working like doing a quick research thing, the graphical user interface is obviously a lot faster and a lot quicker to pick up on.” Conversely, out of the thirteen participants who preferred the GUI, only three participants said they would not use the CLI method as an alternative and would

only use the GUI. The two participants who preferred using the CLI method rated themselves more highly in having computer technology experience than the other participants in the study (having rated themselves as six and seven on the scale out of nine).

4.3. Unique Perspectives

Two of the participants related the GUI to a video game interface. As participant two said, “it kind of reminded me of a video game.” When asked about the benefits of the graphical user interface, participant five also remarked that “it felt like I was playing a video game instead of doing a difficult task.” This suggests that the participants relate the GUI method with interfaces that are designed to have more comfortable interaction. In fact, most of the participants remarked about the GUI’s ease of use. As participant two continues, “[The GUI is] almost impossible to mess up on. And, yeah, I was able to make it through very easy, and I have no cybersecurity experience.” If the GUI can significantly improve user interaction to the point where the user can rarely make errors, then users would not need to be trained to complete these tasks or need to be given step-by-step instructions to accomplish these attacks. Participant 15 had a similar opinion. “[The GUI is] just a more broad [*sic*], beginner-friendly program. That’s what I would say.”

This ease of use comes with a price. Since the GUI’s streamlined approach removes the terminal access point for direct control from the user, participants remarked about the loss of control and specificity that could be achieved by using the CLI. Participant 15 had this to say: “I’d say [the CLI] is more specialized. You can switch your operation on a dime... I would definitely say it’s more precise based on the user’s skill and understanding.”

Ideally, this amount of specificity could be achieved by building a more comprehensive GUI that can support all the features and provide all the information that the CLI can show.

Building such a tool would presumably require a longer and costlier development process. If such a comprehensive GUI could be created, the perception of benefits described by the participants could shift accordingly. With more options and information, ease of use and beginner-friendliness might suffer.

As thirteen of the participants prefer the GUI, and less users reported struggles with completing the GUI lab, the GUI might provide a lower barrier to entry than the CLI. Users of the GUI software would not need comprehensive training or instructions to figure out how to operate basic commands. As participant two describes, “I felt like I could repeat that [GUI] again without the instructions, and I would need the instructions and a bunch of weird codes for the other one.” Participant ten also says, “It's just so simple, like, too simple. Like, anybody could figure that out if given instructions. It took me, like, what a few seconds to do it.”

This matched up with what participants had to say about the CLI method. A common drawback recognized between the participants is its difficulty and how it causes confusion. As participant eight describes the concerns with the CLI method, “Learning curve for sure is the biggest thing. You gotta [*sic*] know what commands you're using and how and where everything is located.” Considering that the participants were asked to complete a relatively simple exploit, this problem would be amplified when in a more complex real-life situation.

However, for more complicated processes, the command-line interface tends to be the more requested method. For cyberattacks and defense, which can involve immediate dangers and provides many high-stakes, complex challenges, the command-line interface might be the recommended option for experienced users. Participant eight, who self-rated himself as having more computer experience than the other participants of this study, draws attention to the usefulness of CLI in time-sensitive scenarios. “If you know what you're doing going into it and

you have it set up, graphical's really great, because it goes nice and fast. But if you are working on something and you aren't fully sure what you're going to be running into, and you need to be able to have a full access of the tool as much as possible in a short amount of time, I think the command line is more suitable for that.”

5. CONCLUSION⁵

Among inexperienced users, the graphical user interface was shown to be easier to use than the command-line interface. Even more experienced users showed interest in using the GUI for quicker, simpler operations as opposed to the more powerful CLI tool. Yet the CLI tool was identified as being the more versatile option, as even though the participants involved rated themselves as having low computer technology experience, many of them were able to recognize situations where the CLI would perform better, often in scenarios when simplicity and inexperience in the software would be more detrimental than beneficial. Inexperienced users require less time and resources to train.

As society shifts and adapts, technology will grow and evolve. The GUI shows great promise as a beginner-friendly interface now, but if technology ever shifts away from GUIs for personal use, a more modern method or even CLI could even become the more user-friendly interface method. The background of the user should be taken into account, additionally. As most of the participants in this study were on the younger side, these participants most likely grew up around using GUIs. Forty years ago, a GUI would have been foreign and obscure. Individuals with prior experience in CLIs may still prefer them to GUIs. Even with all of these factors, a notable portion of the participants still preferred using the GUI over the CLI.

A major shift away from CLIs to GUIs occurred in the 1980s in personal computer usage (Friedman, 1997). This shift indicates an intrinsic benefit of GUIs over CLIs for untrained users

⁵ This thesis draws from and is based on a paper titled "Analysis of Interface Usability Enhancement to Enable Low Skill Staff Service in Critical Cybersecurity Positions" which is under preparation for submission to Technologies. The material in this chapter was co-authored by Wyly Andrews and Jeremy Straub. Both Wyly Andrews and Jeremy Straub were involved in the conceptualization of this project. Wyly Andrews had primary responsibility for software development, and data collection. Wyly Andrews also wrote the initial draft of this chapter. Jeremy Straub provided feedback on and corrections to the chapter.

to make the switch. This benefit appears to be in how a user can more quickly familiarize themselves with a GUI over a CLI.

Finally, there is possibility for room for both methods in modern cybersecurity. As the preferred interface method differs from person to person or situation to situation, a hybrid interface method that includes both interfaces could be implemented. Computers today already implement hybrid interface methods (the implementation of the computer mouse originated from tangible user interface ideas) (Lucignano et al., 2014). Thus, software that can implement both methods could allow users the option to switch methods on the fly as they see fit.

The user preferences show that the GUI is favorable for use in short cybersecurity operations. Without the need for complex commands, the GUI can provide a comfortable, uncluttered interface for a user to complete the task given. For the development of cybersecurity software that seeks for users to complete more complex tasks, having the option to access or switch to a CLI method may be preferable for some users. Even so, a comprehensively built GUI could potentially achieve a similar level of functionality to the CLI, which would be preferable to users who are adept at using the GUI. However, there currently exists a societal need for cybersecurity personnel. From April 2020 to March 2021, there were 131,000 workers in 144,700 information security analyst positions (*Cyberseek*). Improving user interfaces to better function for untrained users could help close this skill gap. Developing future cybersecurity software with GUIs could draw new workers into these cybersecurity roles that would previously be left unfilled.

6. FUTURE WORK⁶

One of the major drawbacks of creating an in-house graphical user interface for the purposes of this study is that the developed GUI is only a mock-up of the capabilities that a real-world cybersecurity application would have. Future studies would benefit from implementing a more complete interface that better showcases what a professional cybersecurity application would feel like. Armitage, a complement tool for Metasploit, implements a real-world example of executing cybersecurity attacks through a graphical user interface. Since Security Command only simulates the attack process, it does not have the full functionality that the command-line version has. A GUI with full functionality would be a better candidate for comparison against command-line.

Another idea would be to ask participants to perform more complex cybersecurity operations and to ask participants to do so without a script, based on a short training. In this study, the exact steps required to execute the exploits were given to the participants. This drastically improves their ability to perform the labs, especially for participants with weaker cybersecurity backgrounds. If participants were instead asked to attack a vulnerable device without explicitly telling them how to do so or perhaps not even telling them how the device is vulnerable, then the interfaces' capabilities would be more thoroughly tested.

This study included mostly participants with little to no cybersecurity backgrounds. This clearly impacted how comfortable the participants were with the two systems, as graphical user

⁶ This thesis draws from and is based on a paper titled "Analysis of Interface Usability Enhancement to Enable Low Skill Staff Service in Critical Cybersecurity Positions" which is under preparation for submission to Technologies. The material in this chapter was co-authored by Wyly Andrews and Jeremy Straub. Both Wyly Andrews and Jeremy Straub were involved in the conceptualization of this project. Wyly Andrews the had primary responsibility for software development, and data collection. Wyly Andrews also wrote the initial draft of this chapter. Jeremy Straub provided feedback on and corrections to the chapter.

interfaces have become more common than command-line applications for daily life. Further study into improving cybersecurity interface design for untrained users would further expand the userbase that can access these tools, as well as making the barrier of entry into cybersecurity less costly. Comparatively, people with more cybersecurity experience will typically have more time working in command-line applications. Repeated with more experienced cybersecurity professionals, the results may show higher preference towards the command-line application.

This study also includes a very limited sample size of participants. Since this study focuses more on garnering opinions of users and investigating these opinions more thoroughly, the study certainly lacks a broad view of preferences for these interfaces. A future study could investigate user preference with a much larger sample size to determine more accurate user preferences. A potential concern with this is that inexperienced users may not be familiar with the terms command-line interface and graphical user interface. This potential study would have to introduce the terms and/or provide example demonstrations to get accurate data.

Participants could also have been asked what prior experience they have had with GUIs and/or CLIs, to gauge how previous familiarity of the participants had in each of these interface methods impacted their answers. This could have helped analyze how the participants' opinions could have been affected by their knowledge of each system. A future study should consider including this detail as part of its analysis.

Additionally, one participant brought up a concern with the CLI method that could affect potential users. Participant twelve mentioned that they have had difficulty working with the CLI because of the text-based interface. “[The CLI is] pretty easy to mix up letters and numbers, especially if you have trouble with those.” Since the CLI displays information entirely based on text, this can cause problems for users with specific disorders, such as dyslexia, who could have

an easier time working with software that displays more graphically. A future study might want to investigate the potential connection between dyslexic users and performance in command-line interfaces.

REFERENCES

- Adams, M., & Makramalla, M. (2015). Cybersecurity Skills Training: An Attacker-Centric Gamified Approach. *Technology Innovation Management Review*, 5(1), 5–14.
<https://doi.org/10.22215/timreview861>
- Afinogenov, G. (2003). *GUI vs . CLI : A Qualitative Comparison David Lanham : Inside*. 1–25.
- Azgaar's Fantasy Map Generator v1.64*. (n.d.). Retrieved July 26, 2021, from
<https://azgaar.github.io/Fantasy-Map-Generator/>
- Bellekens, X., Jayasekara, G., Hindy, H., Bures, M., Brosset, D., Tachtatzis, C., & Atkinson, R. (2019). From Cyber-Security Deception to Manipulation and Gratification Through Gamification. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11594 LNCS, 99–114.
https://doi.org/10.1007/978-3-030-22351-9_7
- Çeker, E., & Özdamli, F. (2017). What “gamification” is and what it’s not. *European Journal of Contemporary Education*, 6(2), 221–228. <https://doi.org/10.13187/ejced.2017.2.221>
- Computer Hope. (2020). *Command line vs. GUI*.
<https://www.computerhope.com/issues/ch000619.htm>
- Couperus, K., Young, S., Walsh, R., Kang, C., Skinner, C., Essendrop, R., Fiala, K., Phelps, J. F., Sletten, Z., Esposito, M. T., Bothwell, J., & Gorbatkin, C. (2020). Immersive Virtual Reality Medical Simulation: Autonomous Trauma Training Simulator. *Cureus*, 12(5).
<https://doi.org/10.7759/cureus.8062>
- Craigen, D., Diakun-Thibault, N., & Purse, R. (2014). Defining Cybersecurity. *Technology Innovation Management Review*, 4(10), 13–21. <https://doi.org/10.22215/timreview835>
- Cyberseek*. (n.d.). Retrieved August 9, 2021, from <https://www.cyberseek.org/>

- Fekete, A., & Rhyner, J. (2020). Sustainable digital transformation of disaster risk—integrating new types of digital social vulnerability and interdependencies with critical infrastructure. *Sustainability (Switzerland)*, *12*(22), 1–18. <https://doi.org/10.3390/su12229324>
- Fellmann, T., & Kavakli, M. (2007). A command line interface versus a graphical user interface in coding VR systems. *Proceedings of the 2nd IASTED International Conference on Human-Computer Interaction, HCI 2007, July*, 142–147.
- Fidler, B. (2017). Cybersecurity governance: a prehistory and its implications. *Digital Policy, Regulation and Governance*, *19*(6), 449–465. <https://doi.org/10.1108/DPRG-05-2017-0026>
- Friedman, T. (1997). *Apple 's 1984 : The Introduction of the Macintosh in the Cultural History of Personal Computers*. 1–7.
- Growth Engineering. (2019). *The History of Gamification (From the Very Beginning to Now)*.
- Hazari, S. I., & Reaves, R. R. (1994). Student preferences toward microcomputer user interfaces. *Computers and Education*, *22*(3), 225–229. [https://doi.org/10.1016/0360-1315\(94\)90003-5](https://doi.org/10.1016/0360-1315(94)90003-5)
- Hultstrand, S., & Olofsson, R. (2015). *Git - CLI or GUI : Which is most widely used and why?* 1–55. <http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A852747&dswid=4529>
- Ishii, H., Lakatos, D., Bonanni, L., & Labrune, J.-B. (2012). *Association for Computing Machinery Radical Atoms: Beyond Tangible Bits, Toward Transformable Materials*. *XIX*(february).
- Kali Linux / Penetration Testing and Ethical Hacking Linux Distribution*. (n.d.). Retrieved July 26, 2021, from <https://www.kali.org/>
- Keskin, O. F., Caramancion, K. M., Tatar, I., Raza, O., & Tatar, U. (2021). Cyber third-party risk management: A comparison of non-intrusive risk scoring reports. *Electronics (Switzerland)*, *10*(10). <https://doi.org/10.3390/electronics10101168>

- King, Z. M., Henshel, D. S., Flora, L., Cains, M. G., Hoffman, B., & Sample, C. (2018). Characterizing and measuring maliciousness for cybersecurity risk assessment. *Frontiers in Psychology, 9*(FEB), 1–19. <https://doi.org/10.3389/fpsyg.2018.00039>
- Landers, R. N., Auer, E. M., Collmus, A. B., & Armstrong, M. B. (2018). Gamification Science, Its History and Future: Definitions and a Research Agenda. *Simulation and Gaming, 49*(3), 315–337. <https://doi.org/10.1177/1046878118774385>
- Lucignano, L., Cuendet, S., Schwendimann, B. A., Shirvani Boroujeni, M., Dehler, J., & Dillenbourg, P. (2014). My hands or my mouse: Comparing a tangible and graphical user interface using eye-tracking data. *Fablearn 2014 (No. EPFL-CONF-204226)*, 1.
- Mahmood, T., & Afzal, U. (2013). Security Analytics: Big Data Analytics for Cybersecurity. *2013 2nd National Conference on Information Assurance (NCIA)*, 129–134. <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6725337>
- Malone, T. (1981). What Makes Things Fun to Learn? A Study of Intrinsically Motivating Computer Games. *Pipeline, 6*(2), 50.
- Mangalindan, J. (2010). Play to win : The game-based economy. In *October* (pp. 5–10).
- Martelle, M. (2018). *Eligible Receiver 97 : Seminal DOD Cyber Exercise Included Mock Terror Strikes and Hostage Simulations*. 1–12.
- Martinez, A., & Christnacht, C. (2021). Women Are Nearly Half of US Workforce but Only 27 % of STEM Workers. *Census.Gov*.
- Metasploit | Penetration Testing Software, Pen Testing Security | Metasploit*. (n.d.). Retrieved July 26, 2021, from <https://www.metasploit.com/>
- Metasploitable 2 Exploitability Guide | Metasploit Documentation*. (n.d.). Retrieved July 26, 2021, from <https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/>

- Oracle. (n.d.). *Oracle VM VirtualBox*. Retrieved July 26, 2021, from <https://www.virtualbox.org/>
- Pelling, N. (2017). *Funding Startups (& other impossibilities) Getting to " yes " in a world of " no "....* 2, 1–31.
- Poster, W. R. (2018). Cybersecurity needs women. *Nature* 2021 555:7698, 555(7698), 577–580. <https://doi.org/10.1038/d41586-018-03327-w>
- Rajani, N. B., Weth, D., Mastellos, N., & Filippidis, F. T. (2019). Use of gamification strategies and tactics in mobile applications for smoking cessation: A review of the UK mobile app market. *BMJ Open*, 9(6). <https://doi.org/10.1136/bmjopen-2018-027883>
- Singh, J., Kaur, S., Kaur, G., & Kaur, G. (2016). A Detailed Survey and Classification of Commonly Recurring Cyber Attacks. *International Journal of Computer Applications*, 141(10), 15–19. <https://doi.org/10.5120/ijca2016909811>
- Styles, N. C. (2018). The Use of Gamification and Its Impact on Crowdfunding Participation: A Participatory Action Research. *ProQuest Dissertations and Theses*, 130. https://ir.stthomas.edu/caps_ed_lead_docdiss/105%0Ahttps://search.proquest.com/docview/2056571290?accountid=13552%0Ahttp://primo-direct-apac.hosted.exlibrisgroup.com/openurl/RMITU/RMIT_SERVICES_PAGE?url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:dis
- Technologies, I. (2017). *Virtual Reality and Immersive Technology are Game Changers for Cybersecurity Job Growth*.
- Tipparach, S. (2019). *The Design of Virtual Reality Based Data Visualization and User Interface in a Semi-Automated Cyber-Security Research Application*.
- Townsend, C. (2019). *A Brief and Incomplete History of Cybersecurity | United States Cybersecurity Magazine*. 1–5. <https://www.uscybersecurity.net/history/>

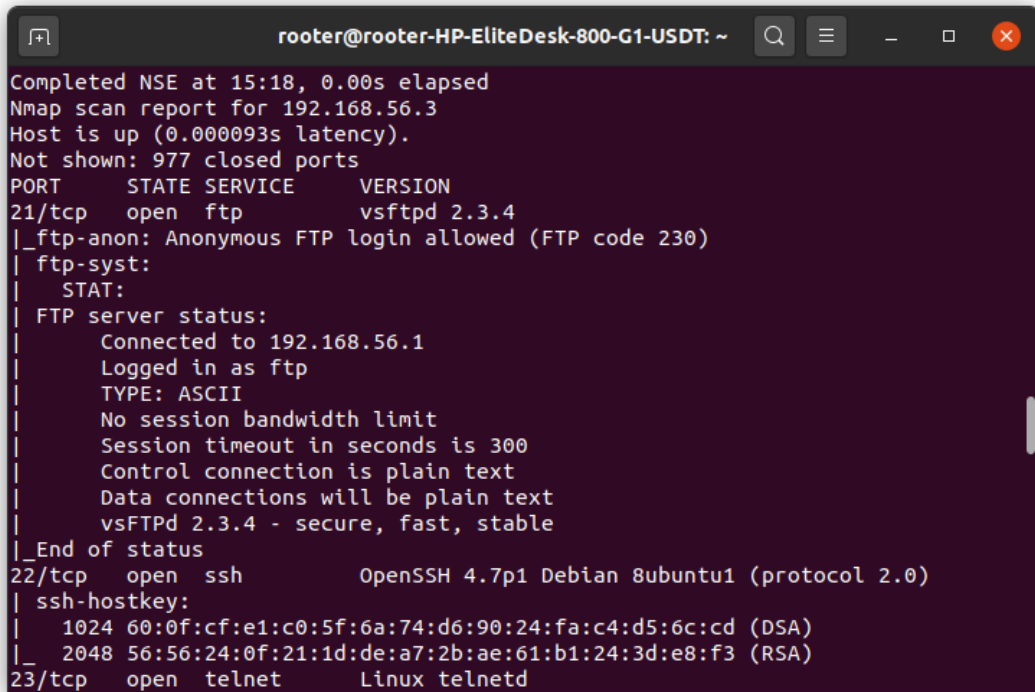
Unity Real-Time Development Platform | 3D, 2D VR & AR Engine. (n.d.). Retrieved July 26, 2021, from <https://unity.com/>

Wadhwa, V. (2015). Quantum Computing Is about to Overturn Cybersecurity's Balance of Power. *The Washington Post*, 1–4.

Warner, M. (2012). *Cybersecurity : A Pre-history*. 27(5), 1–5.

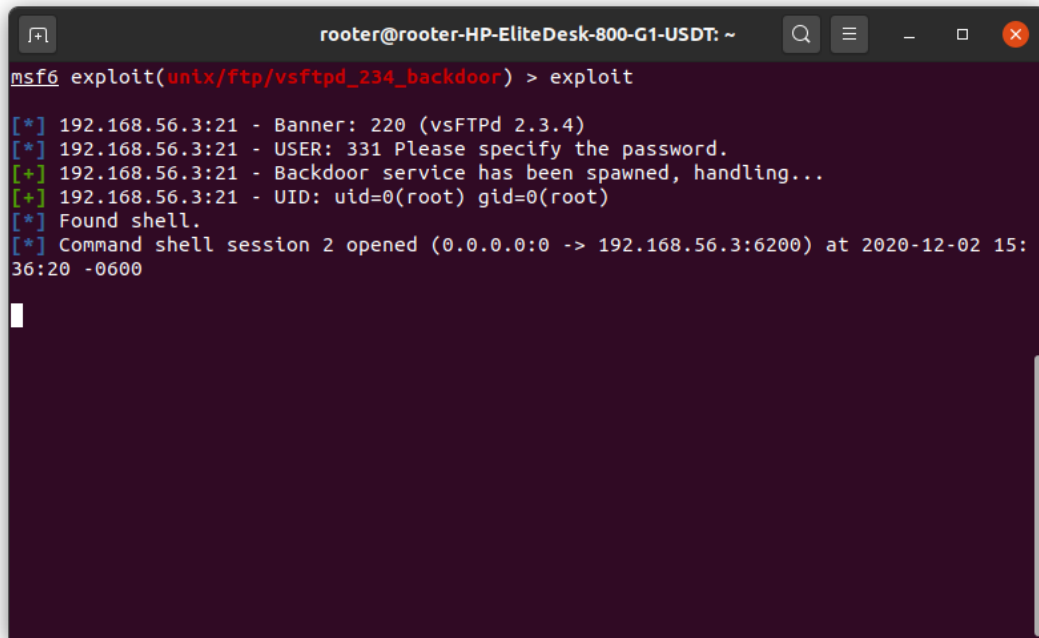
Youssef, Y. (2015). *Gamification in E Learning*. February 2015.
<https://doi.org/10.13140/RG.2.1.4613.4162>

APPENDIX A. FIGURES



```
rooter@rooter-HP-EliteDesk-800-G1-USDT: ~  
Completed NSE at 15:18, 0.00s elapsed  
Nmap scan report for 192.168.56.3  
Host is up (0.000093s latency).  
Not shown: 977 closed ports  
PORT      STATE SERVICE      VERSION  
21/tcp    open  ftp          vsftpd 2.3.4  
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)  
| ftp-syst:  
|   STAT:  
|   FTP server status:  
|     Connected to 192.168.56.1  
|     Logged in as ftp  
|     TYPE: ASCII  
|     No session bandwidth limit  
|     Session timeout in seconds is 300  
|     Control connection is plain text  
|     Data connections will be plain text  
|     vsFTPD 2.3.4 - secure, fast, stable  
|_End of status  
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)  
| ssh-hostkey:  
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)  
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)  
23/tcp    open  telnet      Linux telnetd
```

Figure A1. After running an Nmap scan on the target, the CLI shows a vulnerable port.



```
rooter@rooter-HP-EliteDesk-800-G1-USDT: ~  
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit  
[*] 192.168.56.3:21 - Banner: 220 (vsFTPd 2.3.4)  
[*] 192.168.56.3:21 - USER: 331 Please specify the password.  
[+] 192.168.56.3:21 - Backdoor service has been spawned, handling...  
[+] 192.168.56.3:21 - UID: uid=0(root) gid=0(root)  
[*] Found shell.  
[*] Command shell session 2 opened (0.0.0.0 -> 192.168.56.3:6200) at 2020-12-02 15:  
36:20 -0600
```

Figure A2. Executing a VSFTPD 2.3.4 backdoor opens a command shell on the target.

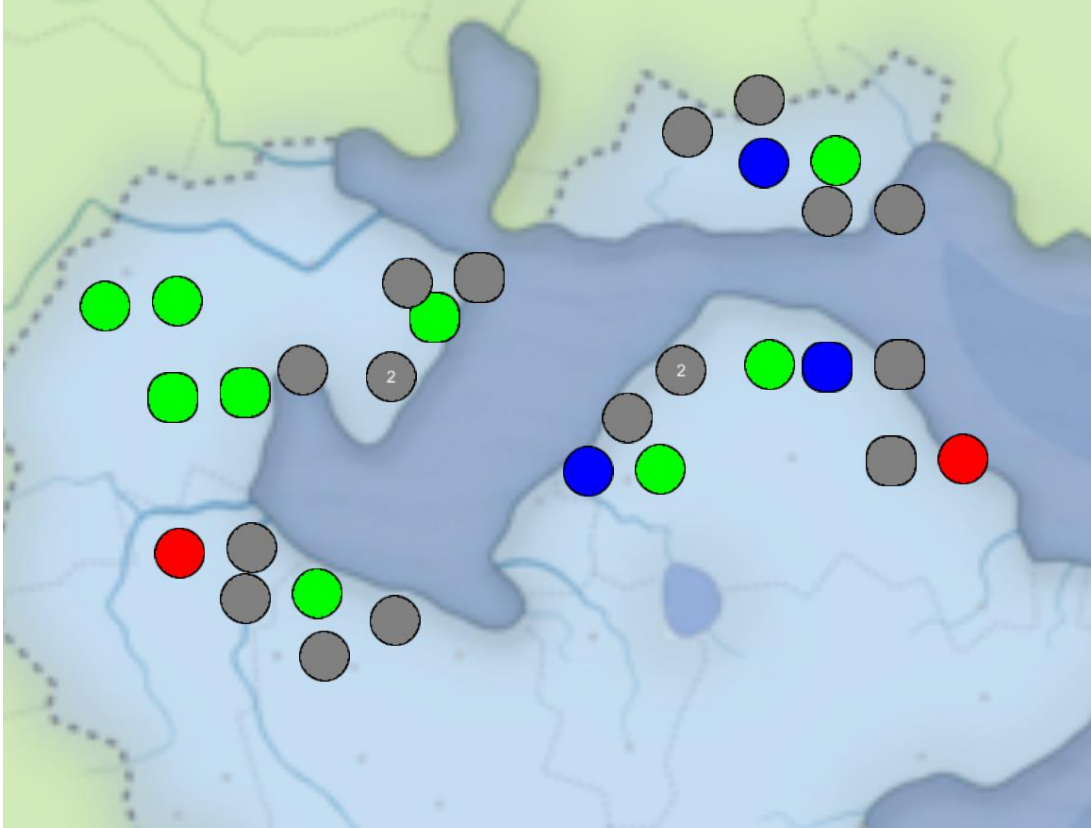


Figure A3. Security Command showing the locations and affiliation of accessible devices.

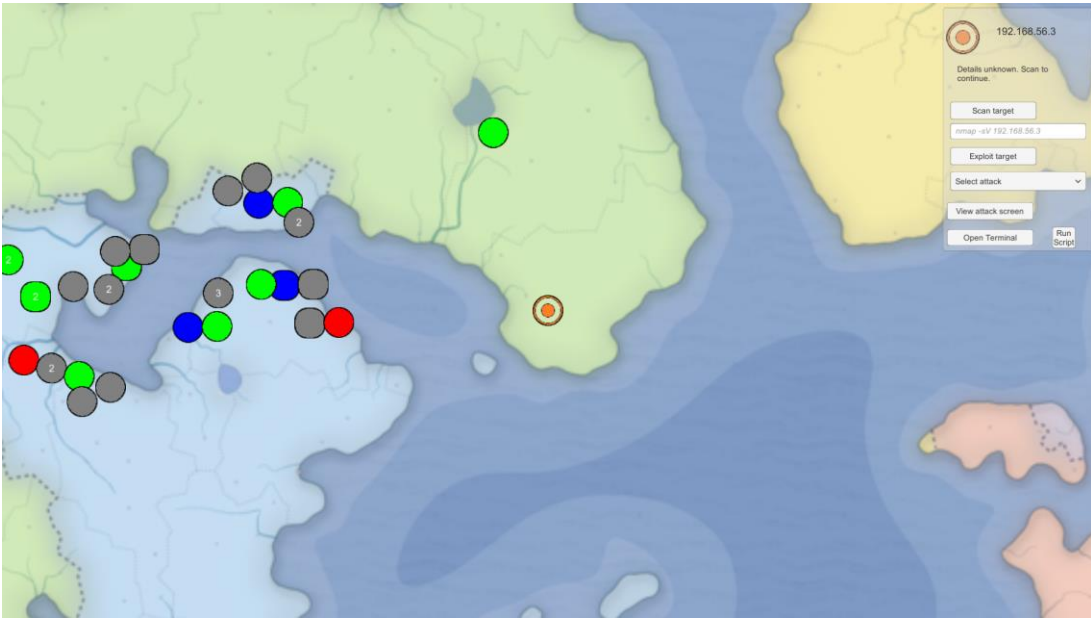


Figure A4. Security Command screen showing a quick menu for immediate action support on a target device.



Figure A5. A successful scan has been completed in Security Command.

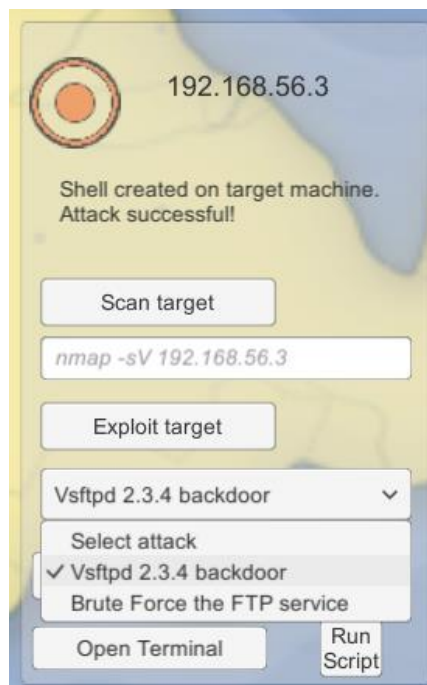


Figure A6. A successful VSFTPD exploit is completed, and a shell is open on the target machine.

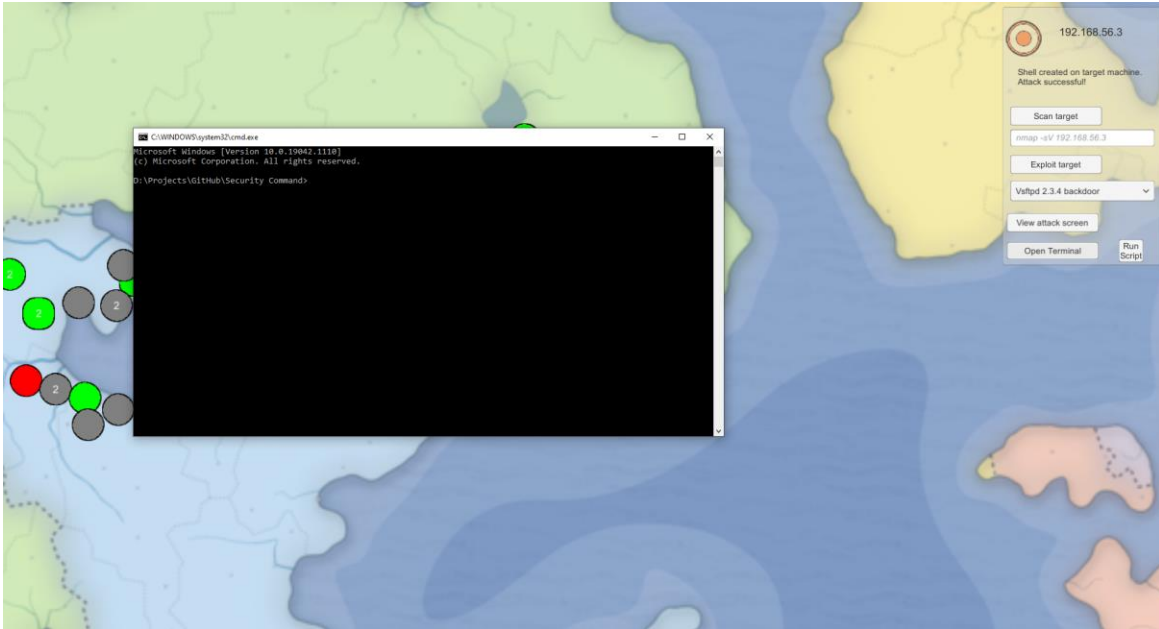


Figure A7. A terminal can be opened through the Security Command application.

APPENDIX B. USER INSTRUCTIONS

B.1. Lab One Instructions

We will be entering the Metasploit Framework Console to perform this attack.

1. You should be at a terminal window. In the terminal, type “msfconsole” and hit Enter.

You should be brought to the Metasploit Framework Console. When the console finishes loading, you should be able to execute Metasploit commands to scan and attack our target. Our target machine has the IP of “192.168.56.3”. Next, we are going to scan the target for weaknesses using an Nmap versioning scan.

2. Enter “nmap -sV 192.168.56.3” into the Metasploit Framework Console.

You should be shown a list of open ports and the versions running on each port. On the open port 21, the target machine is running vsftpd version 2.3.4. This is a vulnerable version for our attack.

3. Enter “use exploit/unix/ftp/vsftpd_234_backdoor” into the Metasploit Framework Console.

Before executing the attack, this attack requires the target machine as a parameter. Once you enter the target machine, you can begin the attack.

4. Enter “set RHOST 192.168.56.3”.
5. Enter “exploit” to begin the exploit on the target machine.

The exploit should begin. If the attack is successful, a command shell should open on the target machine.

6. In the command shell, type “ifconfig”. Verify that you are on the 192.168.56.3 machine.

If you were successful, then you should be able to verify that you are on the 192.168.56.3 machine by using ifconfig.

B.2. Lab Two Instructions

We will be entering the Security Command GUI program to perform this attack.

You should be inside the Security Command GUI program. If you are, you should be able to see a map of devices on a world map. You can use WASD or the arrow keys to scroll around the map. You can also use the mouse wheel to zoom in or out.

1. Find the target device. It should be marked as an orange target icon.
2. Left click on the target device to pull up the quick-attack menu.

You should see the quick-attack menu show up on the right side of the screen. The quick-attack menu has a selection of information and buttons that we will use to perform our attack.

3. Notice that we have not scanned the target. Scan the target now by pressing the “Scan target” button.

Now, the information updates to show that the target machine is vulnerable to a vsftpd 2.3.4 backdoor attack. Let us exploit this vulnerability to gain access to the machine.

4. In the “Select Attack” dropdown, select the attacked labelled “Vsftpd 2.3.4 backdoor”.
5. Once you select you attack, press the “Exploit target” button.

If you were successful, the quick-attack menu should tell you that a shell was created on the target machine.

APPENDIX C. PARTICIPANT INFORMATION SURVEY

Please provide the following information:

Age: _____

Gender: _____

Major: _____

Minor: _____

What is your level of computer technology experience? Please circle one.

1 2 3 4 5 6 7 8 9

Novice

Expert

Do you have any prior service in the military, reserve officer training (ROTC) or Junior ROTC?

Yes

No

APPENDIX D. INTERVIEW QUESTIONS

1. Do you have any education or career plans in cybersecurity? If so, what are they?

Ask for specifics of the type of prior education of not mentioned.

2. Please describe the process of conducting a manual attack.

3. Please describe the process of conducting an attack using the graphical user interface.

4. Please describe the benefits of conducting a manual attack.

5. Please describe the benefits of conducting an attack using the graphical user interface.

6. Please describe the drawbacks of conducting a manual attack.

7. Please describe the drawbacks of conducting an attack using the graphical user interface.

8. Which approach would you prefer to use and why?

9. Would there be times when you would prefer to use the other method?

10. Would you feel comfortable using your preferred method for a job where you would use the system for 7 to 8 hours per day?

11. Is there anything else you would like to share regarding your experience using the two systems?