COMMIT MALWARE ANALYZER: DETECTING MALICIOUS SOFTWARE THROUGH SEMANTIC ANALYSIS OF MALICIOUS SOFTWARE DEVELOPERS' BEHAVIORS, DESCRIPTIVE ATTRIBUTES, AND CODE PUBLICATIONS

DEFINING ETHICAL IMPLICATIONS IN MALWARE INTERACTION WITHIN THE CYBERSECURITY PROFESSION

A Thesis Prospectus
In STS 4500
Presented to
The Faculty of the
School of Engineering and Applied Science
University of Virginia
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Computer Science

By
Vanessa Barlow

November 2, 2020

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

ADVISORS
Catherine D. Baritaud, Department of Engineering and Society
Yuan Tian, Department of Computer Science

The emergence of increased computing power and technology has led to crucial benefits in a vast number of industries, but it has also challenged cybersecurity experts in securing software infrastructure and protecting personal data. Between 2006 and 2016, the U.S. witnessed a 600% increase in federal cyber incidents, coining the attacks "cyber pearl harbor" to highlight the gravity of cyber incidents and the need for new cyber defenses (Karim, 2020, p. 1). In 2018, the Identity Theft Resource Center reported a 44.7% spike in U.S. data breaches while the Electronic Privacy Information Center concluded that 73% of U.S. businesses had experienced a breach, with Equifax, Yahoo, Uber, Target, and eBay having the most catastrophic effects (Kennerly, 2018, p. 123). Looking forward, assessments from hackers, cybersecurity researchers, and information security professionals suggest that 11 major industries are highly vulnerable to future cyber threats. The industries addressed include: implanted medical devices, telework, smart-home devices, autonomous vehicles, cities, trains, aviation technology, 5G networks, schools, hospitals, and energy grids (Kamping-Carder, 2020, Section 2, paras. 2-16).

Cyber threats predominantly arise from malware, "a set of instructions that run on your computer and make your system do something that an attacker wants it to do" (Skoudis, 2003, "Defining the Problem"). The cybersecurity community has explored malware threat and vulnerability minimization techniques by conducting malware detection research. However, malware detection research has been severely limited to the creation of tools that neglect to focus on the identify of a malicious software developer, commonly referred to as a "hacker", alongside the malware they transcribe.

The technical research aims to devise and build a malware detection tool that classifies malware on GitHub, an online software development community. The tool will classify malicious and benign GitHub users based on the software developer's behavior, descriptive

features, and published code. To derive the most meaningful features from the data, malware samples will need to be available to train, test, and validate the detection model. As malware is essential to the success of the technical project, the STS research will explore the ethics of cybersecurity experts who manage and access malware data. This exploration will define the potential moral dangers of easily accessing malware in professional and academic settings. The technical project and loosely coupled STS project will result in a malware detection tool and an in-depth analysis regarding malware accessibility. Figure 1 and Figure 2 display a Gantt chart that details the timeline for completing each major task in the technical project and STS project, respectively.

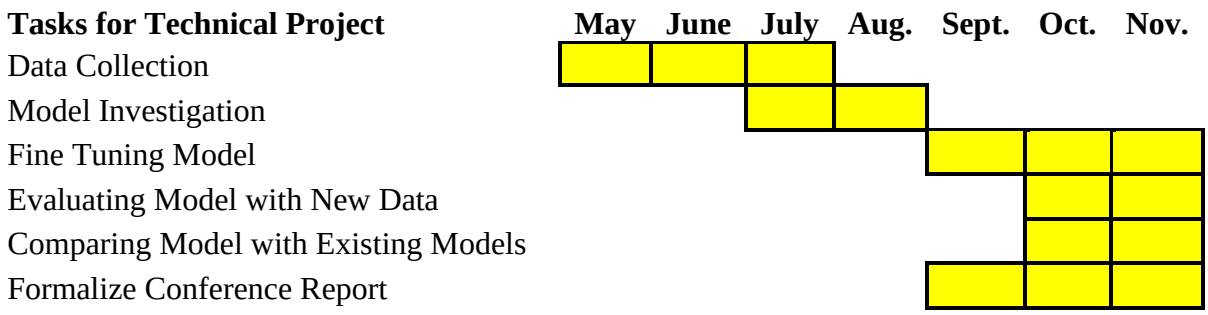

Figure 1: Gantt Chart for Technical Project. This figure visualizes the time frame planned to be spent on each major task in the technical project. (Barlow, 2020).

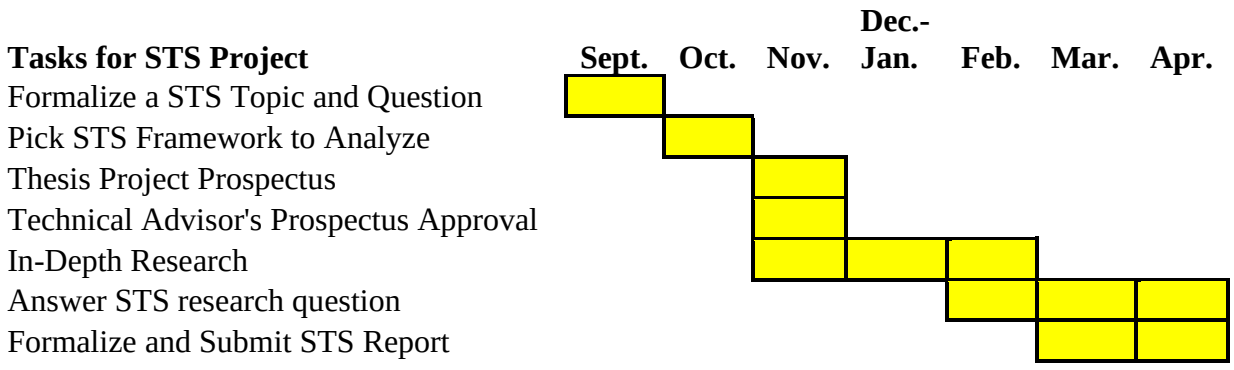

Figure 2: Gantt Chart for STS Project. This chart displays the time frame planned to be spent on each major task in the STS project. (Barlow, 2020)

COMMIT MALWARE ANALYZER

The technical project will explore a semantic analysis of GitHub users to construct a malware detection system which will be referred to as the "commit malware analyzer". Professor Yuan Tian, Department of Computer Science, will supervise the project which will consist of a UVA Department of Computer Science research team with the following 3 members apart from myself: graduate student, Faysal Hossain Shezan, master's student, Kamya Mehul Desai, and undergraduate student, Mahesh Menon. In addition to the UVA research team, the technical project will be in collaboration with Professor Yu Feng and graduate student, Yanju Chen, from the UCSB Department of Computer Science, while also in conjunction with Google personnel for technical guidance.

This system seeks to exceed accomplishments of past detection systems by combining two different techniques that are generally used individually: pattern-based code analysis and semantic classification of software developers' behaviors and characteristics. As illustrated in Figure 3, the general project description can be broken down into the two distinct paths.

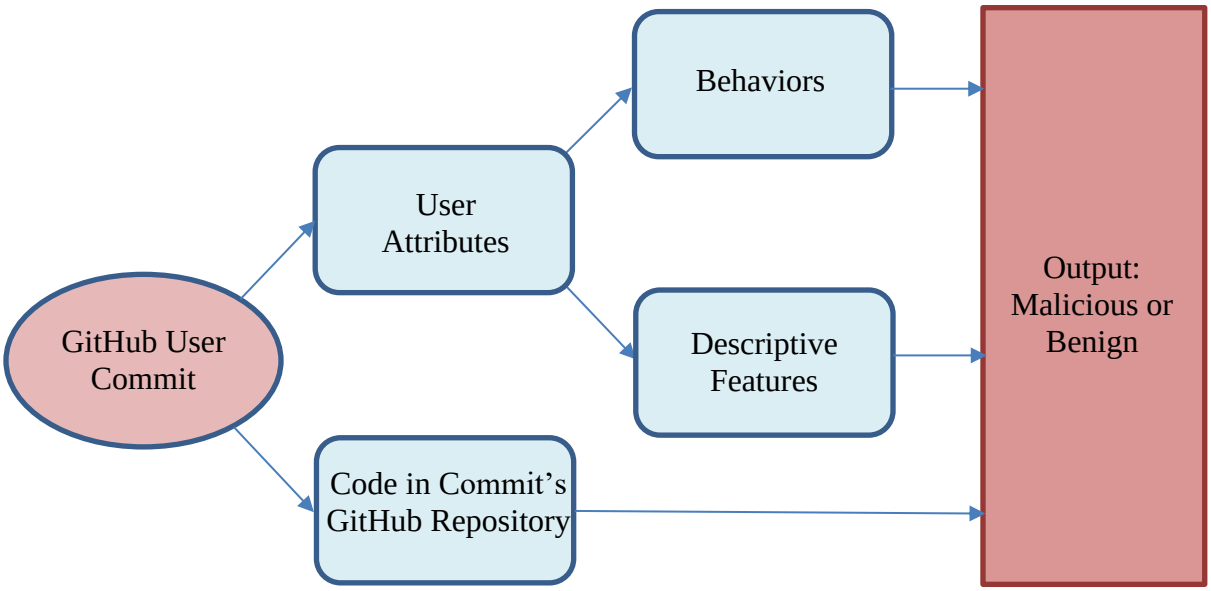

Figure 3: Commit Malware Analyzer Diagram. This figure illustrates the detection tool's input of a user commit, GitHub's term for published code, and the combination of two different types of

data that will be used to determine the final output of whether the commit is malicious or benign. (Barlow, 2020).

The commit malware analyzer receives an input of a GitHub user commit which is a sample of code, ranging from a small snippet to an entire library, that a GitHub user uploads to their working project. This working project is known as a repository. The analyzer will leverage information from the author of the commit by examining the author's actions performed on GitHub and the descriptions published on the author's GitHub page. Additionally, the commit malware analyzer will inspect the commit and the existing code in the repository that the commit is being uploaded to. Both data analyses combine to form an overall output of whether a user's commit is malicious or benign.

Previous tools have problematically relied on analyzing user commits based solely on the top path or bottom path visualized in Figure 3. For example, the detection tool, GitSec, "distinguishes malicious [GitHub] accounts from legitimate ones based on the account profiles as well as dynamic activity characteristics" (Gong et al., 2019, p. 1). GitSec concludes its classification through a data analysis that mimics the top path in Figure 3 since user attributes, such as username length, number of followers, and commit upload frequency, are analyzed to determine whether a GitHub user is malicious. Contrarily, VCCFinder, a code-metric malware detection tool, produces an output based on a similar process to the bottom path of Figure 3 since the tool analyzes and flags malware based on the pattern of code samples (Perl et al., 2015, p. 1). Both tools are limited as they either lack knowledge of the software developer or information from code samples. The commit malware analyzer aims to incorporate both paths in Figure 3 by including similar data and machine learning techniques proposed by GitSec and VCCFinder.

The final objective of this project is to expand on the types of malware that the detection system classifies. Current detection systems are flawed as they cannot generalize malware and

can only identify certain threats. For instance, VirusTotal, a popular research tool that identifies malicious URLs and files, fails to recognize phishing threats: a common cyber-attack where an attacker exploits a user's trust to access personal information (Peng, Yang, Song, & Wang, 2019 p. 1). The commit malware analyzer will improve upon this by training the model with a variety of malware to ensure that the tool can properly distinguish malicious code posing different threats.

To accomplish both objectives, an approach to this project will consist of two major tasks: data collection and machine learning modeling.

GITHUB DATA COLLECTION

To develop the commit malware analyzer, there must be sufficient commit and user data. The project relies on GitHub, an open source software developer community. GitHub offers public user data for research purposes. In the development of GitSec, the malicious user detection system proposed by Gong et al. (2019), the team collected public GitHub user data and formalized a labeling process to determine benign and malicious GitHub users. The data collection and labelling process provided in the GitSec conference paper was used to construct the initial dataset of malicious and benign GitHub users for the commit malware analyzer. The initial dataset comprised of GitHub users with dynamic behavior attributes provided by the GHArchive, a GitHub project that tracks the activities and actions of GitHub users, and basic user information that was accessed from GitHub users' webpages (p. 3). To further strengthen the dataset, the user behaviors, referred to as events, were studied to gain an understanding of the actions a user performed on GitHub. There are multiple types of events that imply that a user uploaded a commit to a distinct repository. The repository names were extracted from events that

implied a user uploaded a commit and queried through GitHub to access the code within the repository. The retrieved code was then added as a feature to the user data (GitHub, 2020).

Another dataset is necessary to ensure the commit malware analyzer does not heavily rely on user attributes. Relevant work completed by Russell et al. (2018) provides the idea of mapping code samples with major security flaws from the Common Vulnerabilities and Exposures (CVE) database to GitHub commits. This mapping allows for the observation of malicious and benign GitHub commits since GitHub commits can be labeled as malicious if a CVE is mapped to it (p. 3). An updated mapping capability between the CVE database and GitHub commits was implemented by UVA research partners, Desai and Shezan, to identify malicious and benign GitHub commit samples. These code samples can then train machine learning models to perform classifications based on pattern matching of malicious and benign GitHub commits.

MACHINE LEARNING MODELS

The assessment of various machine learning models will determine which algorithm can learn the data more efficiently and achieve a high accuracy in classifying GitHub commits. As the commit malware analyzer entails a supervised learning task, meaning that each input commit is labeled as malicious or benign, a support vector machine (SVM) model was considered since the malware detection tool, VCCFinder, took this approach (Perl et al., 2015, p. 1). However, preliminary results affirmed that the SVM model inaccurately accounted for user attributes. Research proposed by Zhan, Zhao, and LeCun (2015) suggested to examine a character-level deep learning model (ConvNet) as this model works “better for less-curated user-generated texts” which describes the essence of code samples (p. 7). The character-level ConvNet chosen for the commit malware analyzer was developed by Kim, Jernite, Sontag, and Rush (2016), a

team of researchers who implemented a ConvNet with an additional long short-term memory (LSTM) machine learning tool. The implementation of the LSTM allows the character-level ConvNet to account for previous classifications when generating new outputs (p. 1). This model will be adapted to fully analyze user attributes and code content.

PERFORMANCE METRICS OF ANALYZER

The commit malware analyzer will result in a reasonable performance level in comparison to existing detection tools. The performance metrics include: precision, recall, and F1-score. Precision refers to the proportion of code that the detection tool classifies as malicious that was indeed malicious, whereas recall denotes the percentage of correct classifications. F1-score is a measure that determines how well the detection tool performs overall with consideration of both recall and precision. The final evaluation and comparisons to existing tools will be presented in a conference style paper written by all authors involved.

MALWARE INTERACTION WITHIN CYBERSECURITY PROFESSIONS

To advance cybersecurity research and awareness, malware is often publicized and inspected for threat analysis and education purposes. Jiang and Zhou (2013) publicly released the Android Malware Genome Project, a dataset containing "1260 Android malware samples… in 49 different Android malware families" to engage cybersecurity researchers in observing and disassembling unique malware to build Android cyber defenses (p.1). However, evaluating ethics on publishing a dataset similar to the Android Malware Genome Project is controversial. On one hand, the cybersecurity community and software companies benefit from this disclosure since software can be patched and defense systems can be developed to mitigate the published malware. In contrast, hackers could use the Android malware samples jointly with their own malware to strengthen their attack.

More importantly, there is a grave danger of malware researchers and cybersecurity students learning and thinking like a malicious developer because their occupation challenges them "to develop the difficult skill of compartmentalizing their ability to think nefariously so that it does not overtake their ability to reason morally." (Sullins, 2014, p. 2). Maintaining a moral mentality when analyzing or learning about malware is essential since the cybersecurity field relies on computers and data, thus lacking the interaction of human participants. The lack of human participation weakens research ethics as decisions are occasionally made without assessing the risks to humans (Deibert & Crete-Nishihata, 2011, p. 535). This controversy poses a debate of whether managing and disclosing malware in the cybersecurity field is detrimental towards the morality of individuals in those professions.

The STS research tackles this question as there is minimal research on the effects of using malware in cybersecurity professions. Cybersecurity focuses predominantly on building improved cyber defense systems, reducing vulnerabilities, and detecting malicious code or users that involves the inspection, creation, and disassembly of malware. However, few cybersecurity researchers and experts acknowledge the threat of a researcher or professional employing the malicious techniques that they have studied. A principal threat researcher, Pompon (2018), addresses this issue of threat analysists taking advantage of the malicious content they evaluate and suggests that this issue is a "growing problem and one that not enough people are talking about" (para. 15). He details how cyber professionals who are regularly exposed to malware search the dark web, disassemble malware, and create malware to counteract hackers (para. 1). Pompon claims that these professionals tend to "find themselves using stealth, misdirection, and even outright deception" in their profession as well as in their personal life (para. 2).

ANALYZING THE IMPLICATIONS OF MALWARE

Pompon (2018) questions the consequences and misuse of malware within the cybersecurity field. To discuss this question, an in-depth analysis of malware and its involvement of various actors must occur. The Actor-Network Theory (ANT) proposed by Latour (1992), and Law and Callon (1988), can be used as a framework to outline the relevant actors in malware technology and to delve into the significant actors of cybersecurity professionals that are highly engaged with malware. Figure 4 illustrates the ANT framework, detailing the significant actors of malware.

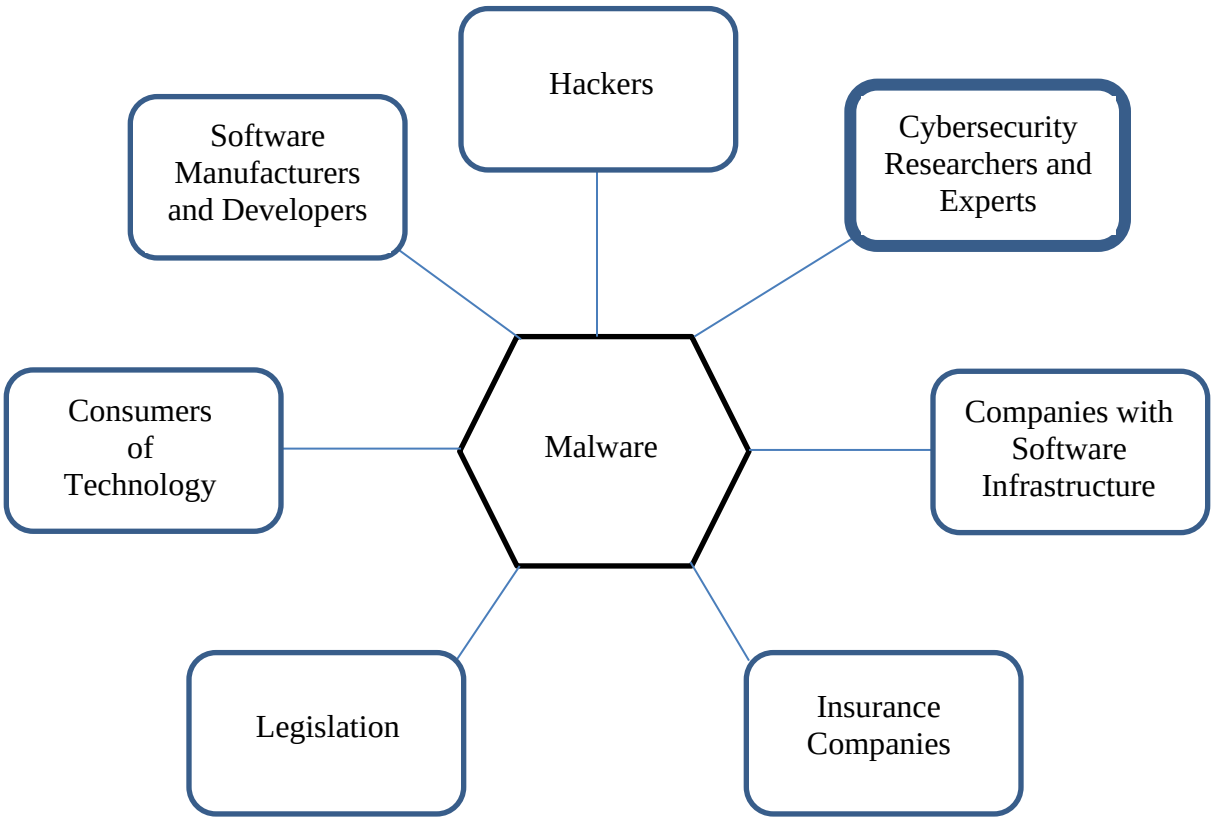

Figure 4: Actor-Network Theory Diagram for Malware. This diagram visualizes the 7 relevant actors of malware: hackers, cybersecurity researchers and experts, companies with software infrastructure, insurance companies, legislation, consumers of technology, and software manufacturers and developers. Cybersecurity researchers and experts are bolded since they are the main actor the STS research investigates. (Adapted by Barlow (2020) from Carlson, 2007).

As displayed in Figure 4, there are 7 dominant actors who contribute to malware: hackers, cybersecurity researchers and experts, companies with software infrastructure, insurance companies, legislation, consumers of technology, and software manufacturers and developers. Each actor has general knowledge regarding malware technology and most actors are interrogated based on ethical questions. For instance, insurance companies are known for providing resources to help those who experience ransomware or data breaches. However, the insurance industry has been challenged on the ethics of paying ransom to hackers as this could provoke future cyber-attacks. Furthermore, legislation is recognized as the driving force of strict or laissez-faire privacy and security regulations. However, the proposal of software security laws is heavily debated as it could limit cyber research and technological progression. Similar ethical confrontations exist within consumers of technology, companies with software infrastructure, and software manufacturers. Consumers and companies risk indulging in software technology that may not be entirely secure, whereas software manufacturers and developers put themselves at risk for creating and distributing vulnerable technology (Kamping-Carder, 2020). Lastly, hackers are often seen as pathological individuals that distribute malware with a criminal mindset and their reasonings for their illicit activities are questioned (Tim & Paul, 1998, p. 757). Differently from all of the actors addressed, cybersecurity professionals are less frequently challenged by ethical questions as they are seen as individuals whose sole purpose is to develop tools and strategies to protect individuals and systems from hackers. However, these professionals implement and disassemble malware daily to learn and reason like a hacker, which could lead to unethical actions.

The objective of the STS research entails an investigation of malware management and handling in the cybersecurity profession to provide a solution of whether malware is destructive

to the morals of individuals in cybersecurity occupations. To fully comprehend the effects of malware in cybersecurity professions, researchers should evaluate the relationship between cybersecurity professionals and their ability to maintain a code of ethics when working with malware. Figure 5 expresses three notable challenges the cybersecurity industry faces. The challenges include: defending systems and detecting malware as rapid as the rate of technological progression, readily alerting consumers and companies of security threats, and interacting with malware in a righteous context.

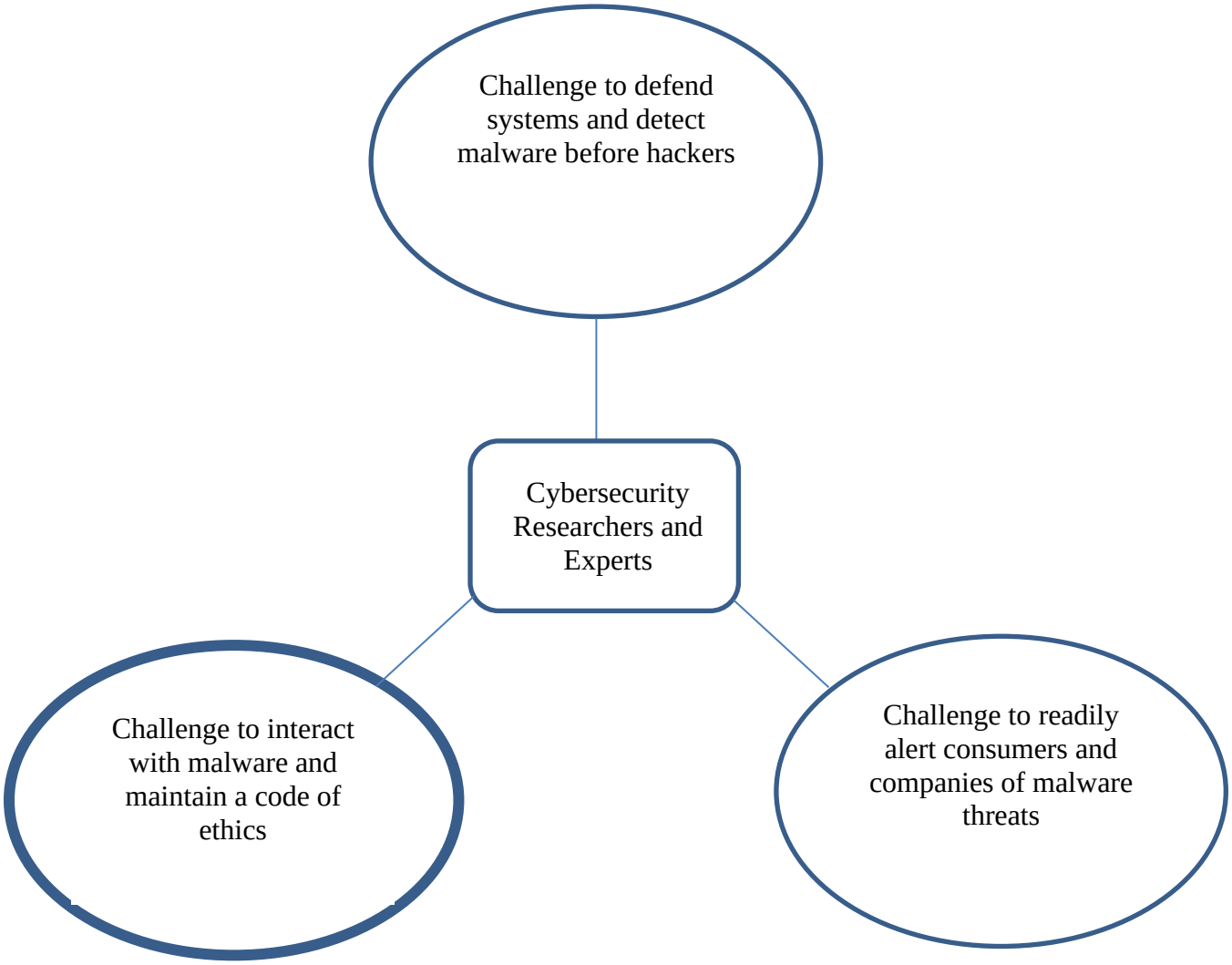

Figure 5: Relationship between Cybersecurity Researchers and Experts and their Perceived Problems. This visual presents 3 significant perceived challenges witnessed by cybersecurity

researchers and experts including the challenge to readily alert consumers and companies of malware threats, the challenge to interact with malware data and maintain a code of ethics, and the challenge to defend systems from and detect malware at a similar pace to the progression rate of technology. The bolded challenge will be the main problem that the STS research will inspect. (Adapted by Barlow (2020) from Carlson, 2007).

The STS research centers the discussion around the cybersecurity profession's challenge of interacting with malware data while maintaining moral and ethical values as emphasized in Figure 5. By examining this perceived problem, a review of the implications of malware availability and exposure in research and the workplace can be analyzed. To understand the extent of nefarious mentalities due to malware inspection and dissection, the two other perceived challenges in Figure 5 will be broadly inspected. This analysis will highlight specific ethical problems cybersecurity experts witness and how malware interaction affects their overall moral and ethical reasonings for their actions in the workplace. The STS deliverable will use the ANT framework to aid this analysis and craft a solution to understand the implications of malware interaction and minimize the threat to cybersecurity professionals' morality.

THE EVOLUTION OF THE DIGITAL AGE

As digital technology evolves, cyber threats increase in complexity. Malware research intends to manage and counteract these threats. The outcomes proposed by the technical research and loosely coupled STS research will seek to provide a new way of classifying malware while addressing the ethics involved in the handling and management of malware in the cybersecurity profession. With these improvements, cyber threats within and outside of the cybersecurity profession will be diminished.

REFERNCES

Barlow, V. (2020). *Gantt Chart for Technical Project.* [1]. *Prospectus* (Unpublished undergraduate thesis). School of Engineering and Applied Science, University of Virginia. Charlottesville, VA.

Barlow, V. (2020). *Gantt Chart for STS Project.* [2]. *Prospectus* (Unpublished undergraduate thesis). School of Engineering and Applied Science, University of Virginia. Charlottesville, VA.

Barlow, V. (2020). *Commit Malware Analyzer Diagram.* [3]. *Prospectus* (Unpublished undergraduate thesis). School of Engineering and Applied Science, University of Virginia. Charlottesville, VA.

Barlow, V. (2020). *Actor-Network Theory Diagram for Malware.* [4]. *Prospectus* (Unpublished undergraduate thesis). School of Engineering and Applied Science, University of Virginia. Charlottesville, VA.

Barlow, V. (2020). *Relationship between Cybersecurity Researchers and Experts and their Perceived Problems.* [5]. *Prospectus* (Unpublished undergraduate thesis). School of Engineering and Applied Science, University of Virginia. Charlottesville, VA.

Deibert, R., & Crete-Nishihata, M. (2011). Blurred boundaries: Probing the ethics of cyberspace research. *Review of Policy Research, 28*(5), 531-537. doi:10.1111/j.1541-1338.2011.00521.x

GitHub. (2020). *GHArchive* [Data set]. Retrieved from https://www.gharchive.org/

Gong, Q., Zhang, J., Chen, Y., Li, Q., Xiao, Y., Wang, X., & Hui, P. (2019). *Detecting malicious accounts in online developer communities using deep learning.* Retrieved from https://dl.acm .org/doi/proceedings/10.1145/3357384

Jiang, X., & Zhou, Y. (2013). Introduction. In: *Android malware*. Retrieved from https://link-springer-com.proxy01.its.virginia.edu/

Jordan, T., & Taylor, P. (1998). A sociology of hackers. *Sociological Review*, *46*(4), 757-780. doi: 10.1111/1467-954X.00139

Kamping-Carder, L. (2020, October, 9). The future of everything: The cybersecurity issue --- Hacking's next targets: Systems we use everyday may not be secure tomorrow. Here's what cybersecurity experts say could be a future focus for attacks. *The Wall Street Journal*, Retrieved from https://www.wsj.com/

Karaim, R. (2020). Cyberwarfare. *CQ Researcher, 30*(9), 1-55. Retrieved from http://library.cqpress.com/

Kennerly, E. (2018). Privacy and the internet. *CQ Researcher*, *28*(6), 121-144. Retrieved from http://library.cqpress.com/

Kim, Y., Jernite, Y., Sontag, D., & Rush M. A. (2016). *Character-Aware Neural Language Models.* Retrieved from https://www.aaai.org/ocs/index.php/AAAI/AAAI16

Latour, B. (1992). Where are the missing masses? The sociology of a few mundane artifacts. In W. Bijker & J. Law (Eds.), *Shaping technology, building society: Studies in sociotechnical change* (pp. 225-258). Cambridge, MA: MIT Press.

Law, J. & Callon, M. (1988). Engineering and sociology in a military aircraft project: A network analysis of technological change. *Social Problems*, *35*(3), 284-297. doi:10.2307/800623

Peng, P., Yang, L., Song, L., & Wang, G. (2019*). Opening the blackbox of VirusTotal: Analyzing online phishing scan engines.* Retrieved from https://dl.acm.org/doi/proceedings/10.1145/3355369

Perl, H., Dechand, S., Smith, M., Arp, D., Yamaguchi, F., Rieke, K., . . . Acar, Y. (2015)

VCCFinder: Finding potential vulnerabilities in open-source projects to assist code audits. Retrieved from https://dl.acm.org/doi/proceedings/10.1145/2810103

Pompon, R. (2018, May). The ethical and legal dilemmas of threat researchers. *(IN)SECUREMagazine.* Retrieved from https://www.helpnetsecurity.com/

Russell, R., Kim, L., Hamilton, L., Lazovich, T., Harer, J., Ozdemir, O., … McConley, M. (2018). *Automated Vulnerability Detection in Source Code Using Deep Representation Learning.* Retrieved from https://ieeexplore.ieee.org/xpl/conhome/8613701/proceeding

Skoudis, E., & Zeltser, L. (2003). *Malware: Fighting malicious code.* Retrieved from https://learning.oreilly.com/

Sullins, J. P. (2014). *A case study in malware research ethics education: When teaching bad is good.* Retrieved from https://ieeexplore.ieee.org/xpl/conhome/6954698/proceeding

Zhan, X., Zhao, J., & LeCun, Y. (2015). *Character-level convolutional networks for text Classification.* Retrieved from https://papers.nips.cc/book/advances-in-neural-information-processing-systems-28-2015