Heartland Payment Systems: A Case Study in Unethical Behavior

STS Research Paper
Presented to the Faculty of the
School of Engineering and Applied Science
University of Virginia

By

Jacob Fishman

April 9, 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.

Signed: _____

Approved: _____ Date _____
Benjamin J. Laugelli, Assistant Professor, Department of Engineering and Society

Introduction

In 2009 Heartland Payment Systems announced that its database had been hacked and, over several months, hundreds of millions of customers' credit cards information had been stolen. The hackers used a strategy called Structured Query Language injection (SQLi). Once inside of the system, the hackers used a program called a sniffer program to steal the credit card data. The stolen data resulted in an estimated $300 million in damages (Vaas, 2018).

The Heartland Payment Systems breach is usually seen as a systemic failure on the part of regulating companies and taught in law school as an example case about the fallout of the breach (Sharkey, 2017), (Marcus, 2018). The previous literature on the topic fails to discuss the morality of the engineers who designed the Heartland Payment Systems database and web application. Many decisions the engineers made were ethically questionable. By not analyzing the morality of the engineer, students who look at the case have been denied the opportunity to learn what it means to design a product ethically.

Examining the actions of the programmer through virtue ethics will provide insight into the morality of the programmer's actions. I argue that the Heartland Payment Systems engineers acted unethically, according to virtue ethics, by designing a system that lacked two of Pritchard's Virtues for Morally Responsible Engineers – competence and seeing the "big picture" as well as the details of smaller domains - and lacked one virtue that Ibo van de Poel and Lambèrt Royakkers argue was a pre-requisite for being a morally responsible engineer: designing safe products (Pritchard,2001) (van de Poel and Royakkers, 2011). I will use virtue ethics to evaluate the morality of the designer's actions in designing the system.

Background

Hackers breached Heartland Payment Systems database in 2007 using structured query language (SQL) injections (Vijayan, 2009). SQL injections are a type of injection that allows attackers to execute harmful statements to a server (Acunetix). These injection attacks can bypass security measures giving attackers access to the entire database (Acunetix). To perform an SQL injection, attackers find a user input on a web page that is vulnerable to the attack (Acunetix). SQL injections are considered among the easiest and most simple form of hacking and are equally as simple to protect against. To do so, a designer must scrub all inputs before allowing the code to use it (Acunetix). There are applications that scrub websites for SQL injections and notify the designer if there is an SQL vulnerability (Acunetix). The hackers installed a sniffer program on the database, which allowed the hackers to steal the information under the radar (Vijayan, 2009). Sniffer programs accumulate all of the data that comes into and out of a system and give the user who deployed them access to the data (Symantec, 2002). Protection against these programs includes anti-sniffers (which search for sniffers and de-activate them), switched networks (which only allow access to certain people), and encryption (which makes the data unreadable except for the person who needs it) (Symantec, 2002).

Literature Review

Many research papers exist outlining the large-scale failure to secure data that resulted in the Heartland Payment Systems breach of many credit card information uncovered in 2009. These papers usually focus on the policy issues that allowed the attack to happen, such as the industry security standards and the contractual obligations between Heartland Payment Systems and the acquirer banks. The analyses avoid talking about the mistake Heartland made by not testing the websites for SQL injection vulnerabilities.

In Can data breach claims survive the economic loss rule? Catherine M. Sharkey (2017) provides details about how the contract was set up to include the Payment Card Industry Data Security Standards (PCI-DSS) (Sharkey, 2017). She focuses on the results of legal action taken against Heartland in the fallout of the breach, as the paper is a law review. She does not consider the design of the system as a reason for the consequences to occur. Her discussion of the PCI-DSS brings the regulations set on companies to the reader's attention, but Sharkey fails to place blame on the designer of the websites which allowed the data breach to occur (Sharkey, 2017).

Daniel J. Marcus (2018) discusses the fallout of the data breach, and actions taken against Heartland during the aftermath (Marcus,2018). Marcus states "shareholders brought a securities fraud class action against Heartland Payment Systems," in response to the company's stock dropping 80 percent because of the data breach (Marcus, 2018). He continues that shareholders claimed Heartland management lied, but the shareholders lost the case (Marcus, 2018). Marcus concludes that litigants who try to sue companies on securities fraud claims will have a hard time proving misinterpretations of companies security systems (Marcus, 2018). As this article is a law review, Marcus does not consider where to place the blame that allowed the breach to occur. He focuses on reactions to the breach through a legal lens, as does most of the literature that currently exists on the Heartland Payment Systems case.

There is much to be discussed about how the regulations failed the community of users, but the case should also be looked at through the lens of a failure to properly design a database to secure data. The research that exists currently states that the root cause of the problem is the policy that existed in place to create the environment that caused the data breach. This paper will use a virtue ethics framework in order to analyze the actions of programmers who designed a

system which, upon failure, incurred millions of dollars in damages through credit card data breaches.

Conceptual Framework

My analysis of the actions of the programmers of the Heartland Payment Systems application and the database of the credit card information draws on the framework of virtue ethics. Virtue Ethics is an ethical theory developed by Aristotle that "focuses on the nature of the acting person," (van de Poel & Royakkers, 2011). Each moral virtue is a mean between two extremes of evil (van de Poel & Royakkers, 2011). For example, modesty is the mean between shamelessness and shyness. Acting modest is the middle ground that is morally acceptable between shamelessness and shyness.

According to Aristotle, virtues can be practiced just like any other skill (van de Poel & Royakkers, 2011). Virtue ethics defines an action as morally acceptable if it is a decision that humans are meant to make (van de Poel & Royakkers, 2011). Aristotle believed that it takes moral skill to decide the virtue that is required depending on the situation (van de Poel and Royakkers, 2011). The types of virtues necessary for one's job differ for types of careers. Engineers have their own set of virtues, while businesspeople can have completely different virtues. Michael S. Pritchard created a list of "Virtues for Morally Responsible Engineers" which

1. Competence/Professionalism
2. Clear and informative communication
3. Ability to work in a team
4. Willingness to make compromises
5. Perseverance
6. Habit of documenting work thoroughly and clearly
7. Commitment to objectivity
8. Openness to correction
9. Striving for Quality
10. Being imaginative
11. Seeing the "big picture" as well as the details of smaller domains

Figure 1: Pritchard's 'Virtues for Morally Responsible Engineers'

I will use to analyze the actions of the programmers as Computer Science Engineers (Pritchard, 2001). The list consists of eleven virtues which are shown in Figure 1. Pritchard explains that to understand how these virtues help engineers they must be put in use in the explanation of engineering practice (Pritchard, 2001). An engineer could also have all of these virtues and still be connected to morally unacceptable projects. Thus, having the virtues does not make one a morally responsible engineer, but the lack of any of the virtue results in a morally ambiguous engineer (Pritchard,2001).

Van de Poel and Royakkers also discuss the engineer's responsibility for safety. They state that "in virtue ethics, care for users or, more in general, for people who suffer the consequences of your design is an important virtue" (van de Poel and Royakkers, 2011). To paraphrase them, designing a safe product is an important virtue. In virtue ethics, the engineer's responsibility is to make sure that the user is as safe as possible while using the product (van de Poel and Royakkers, 2011).

The next section of the paper will analyze whether the programmers acted virtuously. Analysis of the whether the programmers' actions will focus on whether they designed the system using Pritchard's list, as well as van de Poel and Royakkers' framework. To decide whether or not the virtue was used during the design of the system, it is helpful to have a definition of having a virtue. Alasdair MacIntyre describes virtues as having five features: being a desired characteristic, expressed in action, lasting and permanent, always present, and can be influenced by the individual (van de Poel and Royakkers, 2011). Using these five features, we can analyze whether the programmers of the Heartland Payment Systems designed the websites and databases using the virtues that Pritchard put forth.

Analysis

The Heartland Payment Systems engineers acted unethically by lacking three virtues that are necessary of morally sound engineers: competence, recognizing minute details in the system, and designing safe products for users. The designers made decisions that showed a deficiency in utilizing these virtues. This means that the system was designed unethically, as lacking just one of these virtues means that an engineer is not morally sound (Pritchard, 2001). Since the engineers are not morally sound, virtue ethics would say that their design of the system was unethical. The three subsequent sections will discuss each virtue, showing the actions and decisions the engineers made that resulted in a lack of each virtue.

Competence

The Heartland Payment Systems engineers acted unethically by lacking the virtue of competence in the system that they were designing. Many of the decisions that they made throughout the design of the website point to the fact that the engineers were not entirely competent in the design of a secure system and database.

In an article on cybersecurity website Comodo, the article states that the Heartland breach "started with an 'SQL Injection' attack in late 2007 that compromised their database," (Comodo, 2013). The aforementioned SQL injection is so simple that technology buffs "liken this kind of hack to simply turning the front doorknob to get into a house," (Buley, 2009). The fact that the Heartland database was susceptible to an SQLi attack shows that the engineers lacked the engineering virtue of competence in their expertise on the system.

Another piece of evidence that pointed at lack of competence was the fact that Heartland Payment Systems did not realize its database had been hacked for almost an entire year. Heartland Payment Systems did not even recognize the attack themselves (Comodo, 2013). Visa

and Mastercard notified Heartland of suspicious activity on its system, and Heartland found spyware program that was used to steal the data across several months (Comodo, 2013).

This would not have been an issue had the hackers been perfect but Lisa Vaas, a writer for Sophos, says “sloppiness played its part, both on the part of those vulnerabilities but also on the part of the hackers themselves” (Vaas, 2018). The fact that the hackers were sloppy, and still were able to slip through the cracks of Heartland Payment System’s database is another example of lack of competence on the part of the engineers.

I argued that the Heartland engineers showed a lack of competence by not preparing for SQL injection vulnerabilities. However, some people may argue that Heartland’s engineers may not have had knowledge about possible SQL injection vulnerabilities. These arguments are unsupported as SQL injections accounted for the largest number of data breaches of all the attack types in 2016, which is seven years after the data breach (Stockley, 2016). This shows that not only were they prevalent when the Heartland breach occurred, but they were still around seven years later as the industry’s laziness and lack of competence has stayed around (Stockley,2016). SQL injections should be the number one priority when designing websites with user input. It is also very simple to purchase a web application to check all webpages for potential vulnerabilities. The lack of competence comes from the engineers working on something that was outside of their expertise, which resulted in the hackers gaining access to the system through improper security on the website. The decisions that the engineers made demonstrated that competence in the system was not always present, expressed in their action, or lasting and permanent. Due to the reasons stated above, the engineers acted unethically in accordance with the virtue of competence.

Attention to Details

The Heartland engineers also acted unethically because they showed that they could see the big picture, but did not recognize the minute details (flaws) in their system. The smaller, less important aspects of their system were flawed, and this is what the hackers used to exploit the system. The webpage that was used to hack into the database was published eight years prior to the attack (Ritchey, 2015). This shows that even the tiniest flaws in a system can cause the biggest issue. The hackers used a flawed webpage from eight years prior to cause $300 million in damages (Vaas, 2018).

The PCI-DSS provides a standard for security and safety among payment systems. In an article on Forbes' website, the author quotes "'PCI specifically calls this out,' Roemer says. 'The way these guys got hacked there's no way they would have satisfied' those standards," (Buley, 2009). Roemer explains that the database the engineers designed did not follow the standards that they were supposed to follow. This act is not only illegal but also unethical. By not following the standards set out by the PCI, the engineers demonstrated that they did not have a recognition of the minute flaws in their systems that could potentially harm their users and the company that they work for.

CSO Online, a cybersecurity website, discusses that "companies using older versions of Microsoft's SQL Server database are especially vulnerable to SQL injection attacks," (Vijayan, 2009). CSO Online shows that Heartland's Microsoft SQL Server database was out of date, which caused the SQL vulnerabilities. The Microsoft SQL Server database is a tool that runs the database, as well as provides security for the database. Since Heartland's was out of date it allowed for the SQL vulnerabilities. By using out of date software, the engineers were focusing on the big picture of making the system that worked, but failed to

focus on the smaller details of making sure to keep the database secure. Leaving the database this vulnerable shows a lack of attention to the small detail and was a main cause of the database breach.

Mark Stockley, a writer for cybersecurity site Sophos, states that "if websites were properly coded then SQL injection and XSS attacks would have disappeared long ago," (Stockley, 2018). Stockley demonstrates that not only were Heartland Payment System's engineers at fault, but that the whole industry has an issue with coding to avoid SQL injection and cross-site scripting (XSS) attacks. Later in the same article, Stockley states that the attacks' "staying power is testament to the fact that making a site that isn't vulnerable takes a bit more time, effort and attention than making one that isn't," (Stockley, 2016). This quote displays that the attention to detail necessary in creating a website not vulnerable to SQL injections takes more work than one that is vulnerable. To that end, the designers of the Heartland system needed to put more effort into creating their system and pay more attention to the smaller details in their system.

The decisions that the engineers made above showed that attention to detail was not expressed in action, lasting and permanent, always present, and was not influenced by the individual. Due to the aforementioned reasons, the engineers acted unethically with respect to the virtue of attention to detail.

Safety of the User

The system designers also acted unethically by designing a system that was not safe for the users. A key virtue in the design of systems is making sure that the system is safe for its users. Safe is a very general term, but when dealing with credit cards and money, safe should mean secure. The system that the Heartland engineers designed was one that ended up not being

secure. The data breach caused 100 million cards to be leaked (Vaas, 2018). In 2009, when it occurred, this was the biggest criminal breach of card data.

The designers neglected to design a high-quality system that was safe for the user. The hackers were able to "exploit poorly coded Web application software" in order to enter the database and install a sniffer program (Vijayan, 2009). This gave the hackers access to any information that comes into or out of the database, allowing the hackers access to the credit card numbers of millions of people. The designers created a system that allowed the customers information to be exploited and was unsafe for the users to operate.

The designers should have been more knowledgeable and able to create a safer system than the one that was put out to the public. Taylor Buley, a contributor at Forbes, states that "securing the application layer is entry-level security stuff, which raises the question of why so many credit card handlers were vulnerable in the first place," (Buley, 2009). The application layer refers to anywhere on a website that the user can input information. This layer was unsecure, which caused the ability of the hackers to break into the database. The security of the system created an unsafe system that caused damage to the customer, as well as to Heartland as a company.

The Heartland engineers acted unethically in designing a system that was not safe for the user. The users of Heartland's systems ended up losing money, as well as confidence in the company. The engineers should always put the safety of their users first, and they lack the virtue of designing a safe system for their users first in this case.

Conclusion

I have argued that the engineers behind the design of the Heartland Payment Systems database and web application acted unethically in accordance with virtue ethics because they did not follow Pritchard's Virtues for Morally Responsible Engineers, as well as van de Poel and

Royakkers thought about virtue ethics. I argue that the engineers lacked the virtues of competence in the subject of design, seeing the minute details in the system, and designing a safe product for the user. The design and eventual release of the engineers' system was judged as immoral because the engineers do not act as a morally sound group would while designing a product.

Once a product is designed and distributed to the world, engineers impact on society is not over. The engineers are responsible for the user of the product's safety. Virtuous engineers are ones that always makes the choice that will result in the user's safety, as well as make the decisions that are one that a virtuous agent would make. Classifying decisions as right or wrong is often a difficult topic but one that must be done in order to hold the correct people accountable.

References:

Acunetix. (n.d.). What is SQL injection (SQLi) and how to prevent it. Retrieved February 28, 2020, from https://www.acunetix.com/websitesecurity/sql-injection/

Buley, T. (2012, July 11). Inside the year's biggest data breach. Retrieved February 28, 2020, from https://www.forbes.com/2009/08/18/hacker-sql-injection-technology-security-gonzalez.html#674edc51585d

Comodo. (2018, May 7). The Heartland breach: a cautionary tale for e-commerce. Retrieved February 28, 2020, from https://blog.comodo.com/e-commerce/the-heartland-breach-a-cautionary-tale-for-e-commerce/

Marcus, D. J. (2018). The data breach dilemma: proactive solutions for protecting consumers' personal information. *Duke Law Journal*, *68*(5), 559–593.

Pritchard, M. (2001). Responsible engineering: The importance of character and imagination. science and engineering ethics, 7(3), 391–402.

Ritchey, D. (2015, June 3). Data breach directions: what to do after an attack. Retrieved February 28, 2020, from https://www.securitymagazine.com/articles/86071-data-breach-directions-what-to-do-after-an-attack

Sharkey, C. M. (2017). Can data breach claims survive the economic loss rule? *New York University Law & Economics Research Paper Series*, *17*(30). Retrieved April 6, 2020 from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3013642

Stockley, M. (2016, June 15). The web attacks that refuse to die. Retrieved February 28, 2020, from https://nakedsecurity.sophos.com/2016/06/15/the-web-attacks-that-refuse-to-die/

Symantec. (n.d.). Sniffers: what they are and how to protect yourself: Symantec Connect. Retrieved February 28, 2020, from https://www.symantec.com/connect/articles/sniffers-what-they-are-and-how-protect-yourself

Vaas, L. (2018, February 19). Hackers sentenced for SQL injections that cost $300 million. Retrieved February 28, 2020, from https://nakedsecurity.sophos.com/2018/02/19/hackers-sentenced-for-sql-injections-that-cost-300-million/

van de Poel, I., & Royakkers, L. (2011). Ethics, technology, and engineering: An introduction. Hoboken, NJ: Blackwell Publishing Ltd.

Vijayan, J., & Gaudin, S. (2009, August 23). U.S. says SQL injection caused major breaches. Retrieved February 28, 2020, from https://www.csoonline.com/article/2124279/u-s--says-sql-injection-caused-major-breaches.html