Systems Thinking in Data Security: Relating the STAMP Method to Data Security Systems Creation and Incident Evaluation

Or: STAMP, for Dummies

By Matthew Russell

Abstract

Since the turn of the century, computer and internet use has provided businesses and customers with unprecedented methods of connectivity and availability; the digital transfer and storage of sensitive information are now commonplace. These methods, while extremely useful, also introduce immense complexity in what heretofore have been far more basic transactions. Such complexity and availability provides opportunity for hackers with malicious intent; data breaches have become likewise commonplace. A brief glimpse into two data breaches of recent years underscores begs the question - will it ever be possible to analyze cybersecurity incidents and design systems responsible for sensitive data in a way that can actually address the root causes behind them? The STAMP (System-Theoretic Accident Model and Processes) method, championed by Nancy Levenson of MIT, is an accident analysis method designed for modern complex systems that relies on a systems-theory based approach; rather than focus on the usual event-chain model of accident analysis, the STAMP method harnesses a holistic technique seeking to elicit issues arising from “inadequate control or enforcement of safety-related constraints on the design, development, and operation of the system.”[1] The STAMP method is introduced, and general insights are offered into how the widespread use of STAMP can foster a safer cyber-world, both for those designing complex systems, as well as ‘the average coder.’

Table of Contents:

Introduction

Since the turn of the century, digital technology has become an increasingly large part of the fabric of modern society. Digital methods to transfer and store sensitive personal and financial information have become commonplace. And, although these methods are extremely useful, they are also inexorably linked to a dramatic increase in complexity among interactions between humans, humans and computers, and computers and computers, such that the resources need even to carry out the simplest of tasks (i.e. a credit card swipe) is mind-boggling; furthermore, each such increase in complexity of interactions likewise increases the difficulty of maintaining the security of users' private information. This development of complexity in the interactions and storage of sensitive data demands a paradigm-shifting approach to the design of systems responsible for such data, and in the analysis of incidents when things go awry. This paper introduces the Systems-Theoretic Accident Model and Processes (STAMP) method, one that has the potential to make our systems, and our data, much safer.

To The Community

The Systems-Theoretic Accident Model and Processes (STAMP) method, championed Nancy Levenson, a dual professor at MIT in the Department of Aeronautics and Astronautics and the Engineering Systems Division, is the perfect remedy for the challenge of our modern cybersecurity woes. The STAMP method has successfully been applied to a variety of problem-areas ranging from water contamination investigation to assessment of risk of inadvertent launch of the US ballistic missile defense system[1], yet its potential has not been fully applied to cyber and data security.

The crux of the STAMP method is a reformulation of our modern-day approach to safety in systems - one that shifts from the idea of “reliable components” to "a hierarchy of levels of organization, each more complex than the one below, where a level is characterized by having emergent properties.”[2] In such a system, in addition to the notion that the various components are ‘reliable,’ the hierarchy of organization is paramount, with the control structure among components of the hierarchy being of primary importance. It is the attempt of this paper to elucidate the relevant application of the STAMP method to the design of data-security systems and evaluation of data-security incidents in the hopes that, to the dismay of malicious hackers everywhere, it may become more commonly employed.

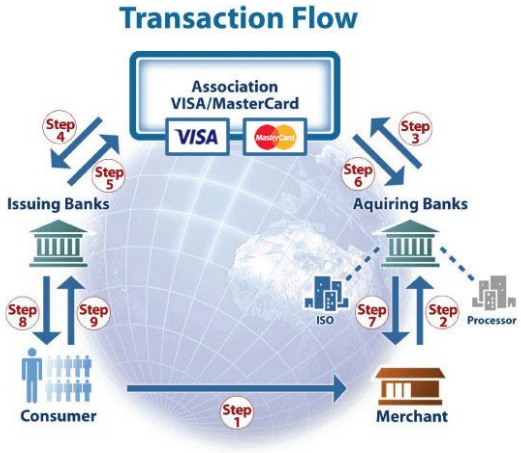
The Costs of Data Security

Intrusions by malicious actors to steal users’ personal information is serious business: here is a small list of companies who have been successful targets of malicious hackers in the past 10 years: Uber; Pizza Hut; Deloitte; Yahoo; Equifax; Adobe; Home Depot; Verisign; Sony Playstation Network; US Office of Management; JP Morgan Chase; Target; Ebay. The costs of such breaches for companies - not to mention to the people whose data is stolen - can be catastrophically high; an IBM whitepaper on the cost of data breaches, coauthored by The Ponemon Institute, explains that even with fast response times (identification < 100 days), a data breach costs an average of $2.8 million; with slower response times (>100 days), average costs average $3.83 million.[3] In addition, the paper reveals that United States organizations

paid $4.13 million in total for losing customers (the highest among all countries surveyed).”[3] Despite these statistics, corporate data breaches still seem to remain commonplace. Why do such breaches happen so often? How can the broader cybersecurity community construct an intelligent approach to solving the significant problem of keeping users’ personal information safe?

Thoroughly Understanding the Problem: Complexity in Data Security

In order to understand how to keep users’ information safe, we must first thoroughly understand the exact nature of the problem. We will thus explore a few examples of how breaches can and do occur. First, let us examine what seems to be an easy job: processing a credit-card transaction. What for end-users is as simple as swipe and sign, is in actuality a series of data transfers through an intricate network of users, banks, and credit card companies, all done through a variety of software and hardware: to put it bluntly, what seems easy, is actually quite difficult. **Figure 1** provides a cursory depiction of the credit-card transaction process and should suffice for the average reader to understand the basic picture; please go to the link cited in the references section if you would like find out more about credit card processing



.

Figure 1: The 'simple' activity of processing a credit card transaction.[4]

Compare the activities required in credit card processing with the transfer of cash from one party to another. For a hacker with malicious intent, the ballooning of complexity associated with financial transactions of modern times presents a variety of opportunities to thieve valuable information. And, as we know from the above list of recently hacked companies, the malicious actors are often successful. According to a SANS Institute Whitepaper on the 2013 Target breach of 40 million credit cards, one failure in that instance was point of sale terminals being compromised with malicious software[5] - an exploit in step 1 of the above listed process. Now, take a moment to reflect on the fact that the processing of credit card information is only one piece of the much larger puzzle of keeping users' information secure.

The Chief Digital Officer of Stanford's Business School has recently had to step down following a massive data breach revealing "14 terabytes of confidential student data from financial aid applications,"[6] in addition to the "[potential exposure] of the personal information of nearly 10,000 non-teaching staff who were employed throughout

the university in August 2008."[6] The reason for this breach was a simple matter of folder permissions being incorrectly set to allow access to a broad range of users.[7] Interestingly, matters such as folder permissions are actually very easy to fix - as easy as one command in a terminal window - why, then, did this happen? Was it one person's neglect? Similarly, was the Target breach as simple as improving protocols for point-of-sale terminal security?

Although the impetus to assign blame to one individual or problem-point may be tempting, particularly given the ease with which certain problems may be or have been solved - we must acknowledge the complexity of the interactions surrounding particular problem-points. For instance, once we put the relatively simple situation regarding folder permissions at Stanford in the context that "most of the files...were managed by six different campus offices,"[7] the complexity undergirding the situation - the connection and communication between the six offices - reveals itself. As for Target, it turns out that hackers' first step was to send malware to Fazio Mechanical, one of the refrigerator vendors that supplied Target - this malware then granted hackers the access they needed to Target's vendor portal, from which they continued their elaborate attack. So, simply 'fixing their point of sale terminals' certainly would not have solved their problem.

Beginning to Understand The Solution: Introduction to The STAMP Method

With such a wide variety of possible attack vectors, in addition to the immense complexity among systems responsible for the safeguarding of sensitive data, how can we keep attackers from accessing sensitive data? The task is daunting, but certainly not

impossible. In order to approach it, through, a paradigm-shift in incident analysis and system design is necessary - rather than address individual problem-points, we must address problematic systems. Nancy Levenson of MIT has championed the STAMP method for incident analysis for just such a purpose. She writes:

> The most common accident causality models assume that accidents are caused by component failure and that making system components highly reliable or planning for their failure will prevent accidents. While this assumption is true in the relatively simple electro-mechanical systems of the past, it is no longer true for they types of complex sociotechnical systems we are building today. A new, extended model of accident causation is needed to underlie more effective engineering approaches to improving safety and better managing risk.[2]

The STAMP method, developed with inspiration from systems analysis, acknowledges that in the development and evaluation of complex systems and security/safety incidents relating to such systems, it is not sufficient to demand that each particular component is 'reliable.' What is the alternative? Rather than focus on 'reliable components,' STAMP focuses on the "inadequate control or enforcement of safety-related constraints on the design, development, and operation of the system."[8] At both Target and Stanford, the systems 'worked,' but in both cases, those systems were exploited because of incomplete constraints imposed on the interactions among components within the systems. So, at Stanford, if the network were being built with the STAMP method in mind, the question would be asked: "what constraints do we need to impose on departmental interactions?" Therefore, rather than addressing the issue of folder permissions explicitly, the problem of folder permissions becomes 'automatically solved' by considering the higher level of relationships and interactions. The Target breach, similarly, may have been preempted by deeper consideration of the constraints

necessary in interactions with suppliers. Significantly, both of these focal points are intimately connected with the broader social interactions that occur, rather than simply focusing on the technical coding elements involved. Fortunately, the STAMP method considers these aspects with equal importance.

Action Items

Levenson writes: "With software, the limits of what is possible to accomplish are different than the limits of what can be accomplished successfully and safely - the limiting factors change from the structural integrity and physical constraints of our materials to limits on our intellectual capabilities."[2] Indeed, in addition to the two examples of Target and Stanford, if we consider the top two items on the OWASP Top 10 list: Code Injection and Cross-Site Scripting [9], they are both examples of software being exploited to accomplish exactly what is *not* the intention of the authors (or, perhaps more accurately, not within the limit of the authors' imaginations) - indeed, nearly all of the OWASP Top 10 fall under this categorization; as computer scientists, **if in the development of code, and the development of systems based on code, we consider the constraints necessary to impose on the hierarchy of relationships among and surrounding the code, then our data, and indeed our cyber-world, will become a much safer place.**[1]

[1] It is worth mentioning that this sentiment may seem irrelevant to current computer science students, given the incredibly small systems of interaction (usually, just testing code) that surrounds their code. The author finds this abhorrent.

Conclusion

Tackling the serious problem of keeping sensitive data safe is one that we can accomplish, but it will require designing software and systems within the context of the complex socio-technical spaces in which they are deployed. The STAMP method should be of particular interest to those currently developing such complex systems tasked with safeguarding sensitive data, or in evaluating incidents relating to such systems. The examples presented in this text have been very briefly overviewed with the hope of convincing readers of the applicability of the STAMP method to data analysis issues, but for an in-depth analysis of the application of STAMP to modern cybersecurity issues, please consult Hamid M. Salim's MIT Master's Thesis[10]; for further information regarding the STAMP method, please consult Nancy Levenson's book, *Engineering a Safer World*, available for free online through MIT Press[2].

References

[1] Nancy Levenson. Personal Webpage: http://sunnyday.mit.edu/

[2] Nancy Levenson. 2012. *Engineering a Safer World* (Online PDF). The MIT Press, Cambridge, MA.
[3] Ponemon Institute. 2017. *2017 Cost of Data Breach Study*. Sponsored by IBM Security.

[4] Fidelity Payment Services. 2017.
from: http://www.fidelitypayment.com/resources/what_are_merchant_services

[5] Teri Radichel. 2014. *Case Study: Critical Controls that Could Have Prevented Target Breach*. Sans Institute.
from:https://www.sans.org/reading-room/whitepapers/casestudies/case-study-critical-controls-prevented-target-breach-35412

[6] Bay Area News Group. 2017. *Stanford business school's exec steps down after data breach revealed*. (December 2017). Mercury News.
from:http://www.mercurynews.com/2017/12/05/stanford-business-schools-exec-steps-down-after-data-breach-revealed/

[7] Lisa Lapin. 2017. *Confidential Data Exposure*. (December 2017). Stanford News.
from: https://news.stanford.edu/2017/12/01/confidential-data-exposure/

[8] Nancy Levenson. 2004. *A New Accident Model for Engineering Safer Systems*. Safety Science, Vol. 43, No 4.
from: http://sunnyday.mit.edu/papers/incose-04.pdf

[9] OWASP Top 10. 2013 Edition.
from: https://www.owasp.org/index.php/Top_10_2013-Top_10

[10] Hamid M. Salim. 2014. *Cyber safety: a systems thinking and systems theory approach to managing cyber security risks*. Master's thesis. MIT, Cambridge, MA.
from: http://hdl.handle.net/1721.1/90804