NTNU
Norwegian University of
Science and Technology

The economics of cybersecurity: Boomerang effects from misaligned incentives

Kostiantyn Lenchik

31-05-2016

Master's Thesis
Master of Science in Information Security
30 ECTS
Department of Computer Science and Media Technology
Norwegian University of Science and Technology, 2016

Supervisor : Prof. dr. techn., dr. rer. nat. Jose J. Gonzalez

Preface

This research paper is a Master's thesis in IMT at NTNU, carried out during the autumn semester of 2015 and spring semester of 2016. The idea of the project has resulted from the cooperation between me and my supervisor Jose J Gonzalez during the summer internship on system dynamics modeling. With the descriptive knowledge of cases when making the third parties liable for backfires over time, I have decided to build a theoretical model to simulate it using Vensim as a system dynamics tool. Jose J Gonzalez has a robust knowledge of the area, so the report would not have been complied without his guidance. He has contributed a lot with his new ideas and feedback to my work, yet, allowed me to follow my own ideas and directions.

Even though there are many system dynamics applications throughout the report, it does not require a vast prior knowledge of the subject. The author has tried to give a smooth introduction into the technique used and illustrated all milestones with figures and graphs. Hence, this report has no strict limitations for the reader's background.

31-05-2016

Acknowledgment

"***Progress is made by trial and failure; the failures are generally a hundred times more numerous than the successes ; yet they are usually left unchronicled.***" [1]

This thesis summarizes the scientific path I have pursued throughout obtaining my Bachelor and Master's degrees. The task to go along the path was not trivial, and there were many challenges on this way. Apart from a number of challenges, there were many uncertainties and trade offs, so this Master's thesis is a result of a learning curve interrupted by the decision of making milestones. In this short chapter, I would like to thank people who have helped me on the way.

Firstly, I would like to thank my supervisor - Jose Julio Gonzalez - for guidance. He has contributed greatly to this thesis in forms of guidelines, ideas and feedback on my work. His vast experience in scientific domains overlapping with my thesis has allowed me to perform a comprehensive research, while still focused on defined research questions and not wasting time off-road. Without him this thesis would not have reached the quality it has now.

Secondly, it would be unfair not to express gratefulness to my family for their great support and motivation. They have actually contributed in more ways than I can list here or even recall.

K. L.

[1] William Ramsay

Abstract

The paper under review is dedicated to the simulation of historical cases of poor information security decisions. Externalities like misaligned incentives that charge the third parties for bad information security are tough barriers to overcome. A number of proposals for regulatory options have been suggested. However, the claim that misaligned incentives have their impact on the third parties is not the whole truth. Security systems are complex not only in the sense of being composed of many interdependent parts. The most challenging part of their complexity resides in the propagation of effects, resulting in a highly unexpected, counterintuitive dynamic behavior. An interesting pattern that often recurs is "policy resistance": the "policy" (namely, the action or intervention) misfires or backfires, as the propagation of effects causes unintended consequences that compromise or even oppose the intended outcome of the policy. The research paper deals with a detailed analysis of Information Security cases where putting the responsibility for bad security on the third party does backfire. The preliminary literature review so far has identified 9 cases. The objective of the scientific work is to develop qualitative system dynamics diagrams for the identified scenarios first. To build diagrams I start with identifying all the instances of the problem. I proceed with arranging them in a closed feedback loop. Then, for all the relations in the feedback loop I define whether presiding instance increases or decreases the value of the following instance. As a result, I get a diagram that allows to see all the system components and understand their interdependencies, including indirect ones. Moreover, I have two feedback loops, the first one is standing for intended consequences, and the other loop - for unintended ones. After I obtain enough quantitative details about the occurred incidents I build quantitative system dynamics models. The benefit of these models is the possibility to simulate with any set of initial conditions and, thus, get a snapshot of the system state over time. As one of the purposes of the target work is education, I further implement them as online simulation models allowing students to test different strategies and get insights into the misaligned incentives and their impact on security. In the final part of my thesis, I provide conclusions towards proper security treatment. Namely, I show how important it is for each security decision to think not only about direct consequences, but also about side effects that due to time delays remain unknown for a long time. I provide clear evidence that the unintended consequence of shifting responsibility to the third party will strongly backfire over time.

Contents

List of Figures

List of Tables

1 Introduction

Misaligned incentives are responsible for bad cybersecurity to the extent that “security failures are caused at least as often by bad incentives as by bad design” [1] .

A frequent misaligned incentive occurs when the organization responsible for system security does not bear the full costs of its failure [2] . Based on such an observation a number of regulatory principles have been proposed to overcome barriers hindering good cybersecurity [3]. However, defenders pushing significant costs to the third parties can be hit, albeit with a time delay, quite severely themselves.

The perceived incentive from the defender’s side is doubly misaligned 1) because the third parties, by design, suffer from the resulting externality, and 2) since ultimately the chosen security strategy hits the defender as a boomerang with a revenge, owing to unanticipated side effects of a bad security solution.

1.1 Topic covered by the project

In this project, I start with the discussion of counterintuitive dynamic system behavior, resulting from feedback and time delays in complex systems. Then, I proceed with the overview of historical cases, where due to significant time delays, the presence of externalities and misaligned incentives security managers opts for decisions that backfire afterwards.

The first historical case exhibiting the described properties happened in 1993 with the European and American banks introducing different policies towards ATM fraud. In a survey on frauds against Automatic Telling Machines (ATMs) [4], Anderson found that fraud patterns depended on whether a bank’s customer or a bank itself had to provide evidence. In some countries, including the USA, if a customer disputed a transaction, the bank had the burden to prove that the customer was mistaken or lying; this gave the banks a motive to protect their systems properly. However, in European countries, including Britain, Norway and the Netherlands, the situation was opposite. The principle was the following: the bank is always right unless its customer can prove otherwise. Yet, for the customer this task appears nearly impossible. “Lucky” banks in these countries became complacent and careless. An unexpected result of this strategy was the avalanche of frauds, demolishing their complacency. On the contrary, the US and other countries’ banks that were forced to carry their burden of proof suffered from much fewer fraud cases. Moreover, they spent little money on security, compared to their European counterparts. Hence, better aligned incentives, when the defender suffered most if the security was bad, happened to be beneficial for both banks and their customers.

Having done the preliminary literature review I have managed to find 8 more cases from diverse industries showing a similar behavioral pattern. Through my project, I describe and model them by means of system dynamics. This is quite a challenge as it would require me to gather all the relative data and develop models that represent the reality as accurately as possible. My goal is to provide clear evidence that the unintended consequence of shifting the responsibility to the third party will strongly backfire over time. In Section 1.6 I give our insight into planned contributions.

1.2 Problem description

Security solutions involving technology and human factors are dynamically complex systems. There are two types of complexity: combinatorial and dynamic. Combinatorial complexity is an aggregate impact of a great number of system components; it can be efficiently dealt with by decomposing the system in subsystems, small enough to be easily handled.

Dynamic complexity reflects changes in the system state over time, which result from system components having non-linear relations with one another. It is hard to predict the behavior of such systems over time even when the system is small. Other complications that can be frequently observed in real systems are time delays. This means that if Component A has an influence on Component B, the result of this influence will show up with time delays. [5].

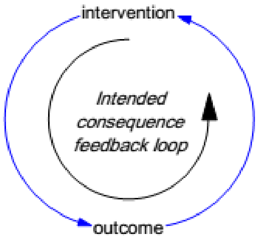


Figure 1: Intended consequence feedback loop.

The main consequence is that interventions in dynamically complex systems always have side effects. Let me consider the outcome intended by the decision maker first. The intervention must be applied over some period of time, and the outcome will be a certain time-dependent result that, in turn, will influence the dosage of the intervention (expressed by the influence arrows in Figure 1). The closed cause-and-effect loop describes the pattern of feedback occurring over a particular time interval. The feedback is shown symbolically by the loop labeled ‘Intended consequence feedback loop’.

Owing to the interdependent system components, the outcome will cause adverse effects. Unless the decision maker has done an excellent job on modeling the system so as to anticipate side effects, the system reaction will be unintended and quite unexpected. Again, one obtains the feedback acting over a certain interim. (labeled in Figure 2 as ‘Unintended consequence feedback loop’). The line labeled “system boundary” indicates that the unintended consequences are hidden from the view of a decision maker. In dynamically complex real systems the effects of interventions tend to show up far away from the origin of the intervention. In addition, the unintended consequences can appear with significant time delays as adverse effects, so that the causal connection between the intervention and the system reaction is not apparent. A significant time delay is shown in Figure 2 by the || on the influence arrow going from ‘outcome’ to ‘system reaction’.

Another important aspect is that, quite often, the dynamic complexity of a system makes the unintended consequences highly counterintuitive. Initially, in dynamic complex systems the intervention mostly achieves the intended outcome, but as soon as the system reaction evolves, the unintended consequence often compromises the intended outcome.

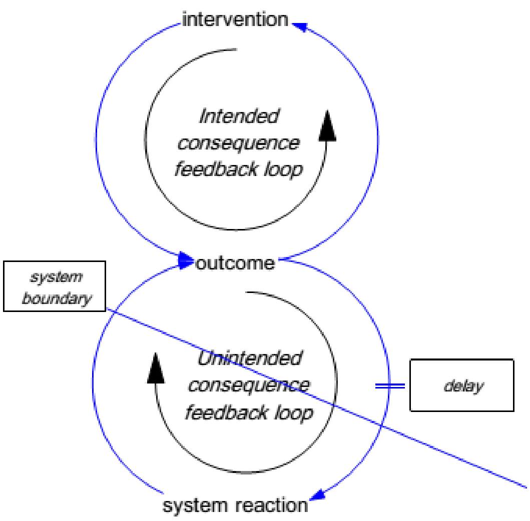

intervention
Intended consequence feedback loop
system boundary
outcome
Unintended consequence feedback loop
delay
system reaction

Figure 2: Intended consequence and unintended consequence.

In other words, in a dynamic complex system interventions tend to follow a better-before-worse behavioral pattern. This phenomenon is known as policy resistance [5].

The reverse course of events happens frequently: to achieve sustainable positive outcomes delayed long-term effects must be taken into account upfront to the extent that one has to accept a worse-before-better behavior. The initial phase (‘worse’) is typically caused by the need to invest resources in capacity that in the long run will produce the desired outcome.

1.3 Justification, Motivation and Benefits

The thesis will be beneficial to both industry and university. For university students it will provide mature case-study materials for learning to make counter-intuitive decisions. For industry the main lesson to be learned can prove that badly aligned incentives can be as a drastic treat as bad security design solutions. Furthermore, the target work provides a strong argument for a proactive approach and the application of System Dynamics modeling that can save from failures in future. From a different angle, the author views the System Dynamic as the best-suited tool for performing the target research. Further, in this paragraph I will explain why. To build the diagrams I start with identifying all the cause-effect relations of the problem. I proceed with arranging them in a closed feedback loop. After this, for all the relations in the feedback loop I define whether the cause increases or decreases the value of effect. As a result, I obtain a diagram that enables to see all the components of the system and understand their interdependencies, including indirect ones. Moreover, I set up two feedback loops: the first one stands for intended and other one - for unintended consequences.

To build quantitative system dynamics models, I need significant data review. Having it in place, I will be able to justify values implied in quantitative assessment. Such diagrams require vast usage of auxiliary instances. By auxiliary instances I mean variables that connect different parts of the diagram (e.g. “the effect of security level on the number of cumulated frauds” connect security level and the Number of cumulated frauds). In most cases such auxiliary instances are implemented as a lookup function, which is constructed according to available expert judgment.

The main benefit of quantitative models is the possibility to simulate with any set of initial conditions and, therefore, get snapshots of the system state over time. As one of the purposes of the target work is education, I further implement them as online simulation models allowing the student to test different strategies and get insights into misaligned incentives and their impact on security.

1.4 Feasibility study

First of all, this Master’s Thesis provided a valuable experience for the author in form of acquiring practical applications of knowledge gained during Master’s studies at NTNU Gjøvik. During the development of this Master’s Thesis the author’s proficiency was increased specifically in System Dynamics, Information Security Economics, Information Security Management.

Secondly, working on this Master’s thesis has improved the author’s research and scientific writing skills.

Thirdly, since the author had to work with detailed description CISO decisions and incentives, he also got valuable knowledge of industry approach towards information security.

1.5 Research questions

- What are the most important security cases concerning misaligned incentives?
- For each case: which kind of misaligned incentive occurred and which unintended consequences evolved?
- Have responsible people found the right solution? If not, what solution can I suggest?
- Which insights provide system dynamics models of the cases with misaligned incentives?
- How to implement online simulation models for learning about misaligned incentives in security?

To answer first three questions, I did literature review in order to pick up cases related to stated problem of misaligned incentives. After carefully analyzing them, I got clear understanding, as to what possible solutions can be proposed based on developed system dynamics models. In next chapters of Thesis I will examine one case in detail, including building a stock and flow model and online simulation. This will provide answer to the last research question.

1.6 Choice of methods

This thesis has both theoretical and practical aspects. I implement practical, or, in other words, the modeling part, by applying System Dynamics. To implement System Dynamics diagrams I will imply Vensim modeling environment.

Firstly, System Dynamics modeling was chosen due to the author's basic expertise in this field. During his Master's education, the author completed two security courses implying application of System Dynamics. Furthermore, the author completed summer internship which consisted of solving comprehensive exercises from both Vensim Modeling Guide and book "Business Dynamics: Systems Thinking and Modeling for a Complex World" by John D. Sterman. The book is considered to be the best introduction into the science of System Dynamics. During his internship the author had a beneficial interaction with his supervisor Jose Gonzalez, which led to obtaining good knowledge.

Secondly, addressing goal of a project to provide evidence that misaligned incentives do backfire, I had to choose a methodology that can model behaviour of complex system. Therefore I had to choose one of the available Imitation Modeling methodologies: Agent Modeling, Discrete-event simulation or System Dynamics. Agent based modeling suites best, when behaviour of system agents can differ a lot. Owing to the selected aggregation level, I decided that system agents will behave in almost the same way. Next, Discrete-even simulation is good for systems, that have a behaviour pattern relying heavily on weather certain events happened or not. Even though our work will have a lot of feedback loops, that introduce influence of past events on current state, I don't expect the model to rely heavily on particular discrete events. Most probable, its behavior will be based on variety of continuous feedback effects from past. Hence I opted for System Dynamics, that can model a behaviour of complex system with focus on system components and their interactions, rather than individuals or events.

1.7 Ethical and legal considerations

As for ethical and legal considerations, the following aspects were taken into account:

- citation and referencing procedure in my paper,
- confirmation of using scientists' thoughts and statements in my thesis,
- licensed software implementation for the model simulation.

Firstly, I understand a vital role of the ethical side while referring to the open sources as I have been doing it in all the other works of mine. So, I cited properly every idea or thought I used in my thesis.

Secondly, I asked for the approval to include other people's opinion as well as assessments to my project. Through the project my colleagues - supervising lecturer and experts - shared their ideas regarding the topic of my thesis, expressed their opinions and probable outcomes. Thirdly, I used free software only for the model simulation. The above-mentioned issues are the main ethical and legal considerations for the target project.

1.8 Contributions

In several application areas, it is shown that investing in simulation models for the strategy analysis totals to 0.005 of the cost of failures due to bad decisions. In our work by modeling an application I target at providing evidence for benefiting from proactive security treatment. Namely, I show how important it is for each security decision maker to think not only about direct consequences, but also about side effects that due to time delays remain unknown for a long time. I also target at providing clear evidence that unintended consequence of shifting responsibility to the third party can strongly backfire over time, hitting the defender himself who expected benefits from passing the liability to the third parties.

I expect my work to be of great interest for security officers in a broad variety of companies, who are seeking to develop the right approach towards security. Another purpose of the given work is education. Having said that, I further target at implementing developed system dynamics diagrams as online simulation models allowing students to test different strategies and get insights into misaligned incentives and their impact on security.

1.9 Outline

In Chapter 2 I identify and describe all the cases of interest. I select the most interesting cases that will be examined in more detail and argue for my choice. In Chapter 3 I provide a comprehensive literature review for selected cases and step by step build my stock and flow model. Chapter 4 is devoted to the simulating and testing of the designed model. Afterwards, I discuss the results of the simulation and provide its online version in Chapter 5. I summarize our work in Chapter 6 and give our plans for future work. The conclusion, Chapter 7 contains our main results and prospective studies in this field. The bibliography contains the list of sources I have used.

2 Identifying cases of interest

In this section I provide a short overview of 8 historical cases, where unintended consequences emerged due to misaligned security responsibilities. The cases provide proofs that putting the liability on the third parties ultimately backfires over time on the organization responsible for the security (hereafter called the agent). To make the presence of misaligned incentives more explicit I split the description of each case into sections describing an intended consequence, unintended consequence and, in some cases, a solution.

In the intended consequence section, I show that in order to have some gain the agent makes another party liable, and this is the outcome for the initial time period. The unintended outcome has no immediate influence due to time delays, but, eventually, results in losses for the agent (which can be related to economics, reputation, loss of goodwill, etc).

2.1 The ATMs in Europe and USA

The following description is derived from the study [1] concerning events that happened between 1982 and 1993. As ATM was a relatively new technology, it had various vulnerabilities and backdoors. In the context of this paper when I say "vulnerabiliity" I mean a certain feature that despite being implemented with good intents can be exploited by malicious actors. The term "backdoor" is used in a sense of a technological or software feature that was intentionally introduced. "Backdoors" were supposed to be removed, but due to various reasons were still active in the exploitation phase. I give particular examples of backdoors and vulnerabilities in Section 3.4.

A lot of customers suffered from fraud, bank employee‘s mistakes and technology failures, also known as “phantom withdrawals”. Many customers’ claims concerning withdrawals were not satisfied by the bank. Consequently, it set a legal precedent. In some countries, including the USA, the following regulation was implemented: if a customer disputed a transaction, the bank had the burden of proof that the customer was mistaken or lying; this gave banks a motive to protect their systems properly. But in several European countries (including Britain, Norway and the Netherlands), it was the customer who had the burden of proof: the bank was right unless the customer could prove it wrong – a nearly impossible task.

This description unfolds the scenario, how the European and US banks acted against ATM fraud.

2.1.1 Intended Consequence in US case

Due to imposed regulation by the US Federal Reserve "requires banks to refund all disputed transactions unless they can prove fraud by the customer"[6] [7]. The the U.S. banks had to refund all the abuses claimed by the customer, when they had no solid proof of the customer's guilt. If the customer disputes an ATM transaction, the burden of the proof is on the bank. Thus, the bank's intervention is 'Burden of proof on bank' in Figure 3. The intended outcome is to reduce the number of fraudulent transactions by the customer (represented by the variable 'Fraudulent transactions') to an acceptable target. Naturally, this led to the promotion of security technology development, including cryptology developments and video surveillance (expressed by 'Security spending') which affects fraudulent transactions with negative polarity. To the extent that fraudulent transactions occur, the burden of proof on the bank is exerted, closing the loop. As a result, one gets control as the intended consequence, resulting in a balancing feedback loop, labeled 'B: Bank is liable'.

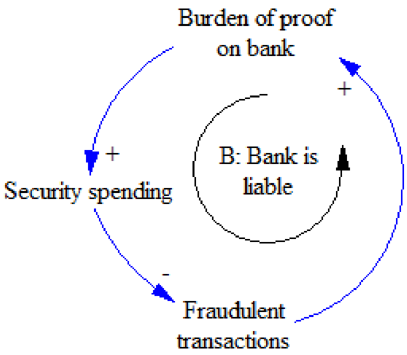


Figure 3: Burden of proof on bank.

2.1.2 Unintended Consequence in US case

Due to regulations some bank customers started exploiting the fact of the bank accepting losses and subsequently claiming to refund genuine withdrawals from ATM‘s, but it turned out to be a minor loss, on average not exceeding $15 000 per bank per year[8].

The major problem was linked to professional crooks who started to actively research zero-day vulnerabilities. With a time delay crooks will come up with ingenious 'Fraud schemes' in Figure 4 (positive polarity), which will increase the number of 'Fraudulent transactions' (positive polarity). The unintended outcome is a reinforcing loop ('R: Betting on the bank to accept the loss') [9].

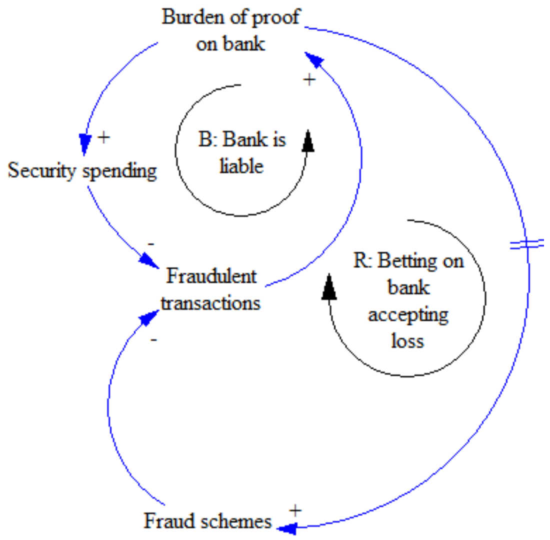


Figure 4: New fraud schemes emerge.

2.1.3 Intended Consequence in European case

The European banks did not meet such demanding regulations in 1980s as was applied to the US banks [10]. Naturally, they opted to put the burden of proof on a customer: the bank was right unless the customer could prove it wrong – a nearly impossible task. Thus, the intervention is 'Burden of proof on customers', see Figure 5.

The intended outcome is to reduce the number of fraudulent transactions by the customer (represented by the variable 'Fraudulent transactions') to the acceptable extent. Thus, one has as intended consequence a control strategy, expressed by the balancing feedback loop labelled 'B: Customer is liable' in Figure 5. The influence arrow from 'Burden of proof on customers' to 'Fraudulent transactions' has a minus sign – negative polarity – expressing that the two variables move in the opposite direction. That is, if the burden of proof on customers is increased, the outcome – fraudulent transactions – gets reduced (and vice versa).

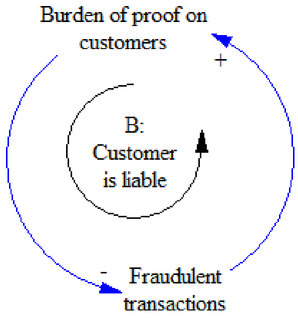


Figure 5: Burden of proof on customers.

2.1.4 Unintended Consequence in European case

In accordance with the facts[11], the unintended consequence of the bank making the customer liable is the increase in the bank's complacency – shown in Figure 6 by the influence arrow from 'Burden of proof in customers' to 'Bank's complacency'.

Let me note that this arrow has positive polarity, expressing that the variables move in the same direction. That is, the increase in the burden of proof exerted on customers increases the bank's complacency, whereas if the bank exerted less pressure on making the customer liable, the bank's complacency would decrease.

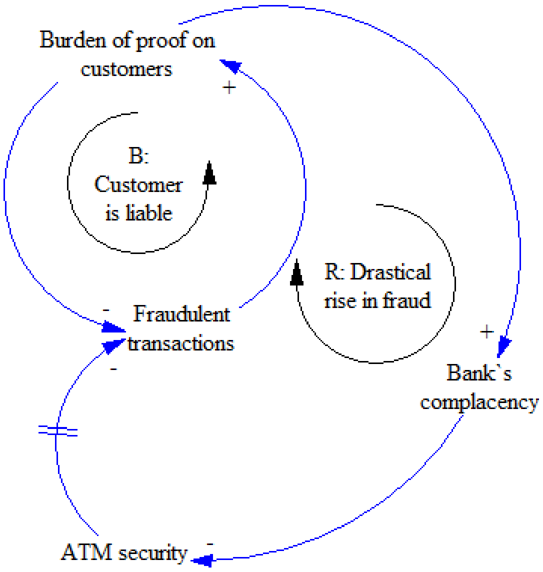


Figure 6: Bank's complacency.

In turn, the variable 'Bank's complacency' influences 'ATM security' with negative polarity: the increase in the bank's carelessness decreases the ATM security over time – with a time delay, indicated by | |, as too little is done to analyze the causes of fraud, discover vulnerabilities and exploits, and remedy them. Over time, again with some delay, 'ATM security' influences 'Fraudulent transactions' with negative polarity – expressing that the decrease in 'ATM security' increases the rate of fraudulent transactions – as more and more crooks discover the poor security in the ATMs along with the bank barking up the wrong tree.

Let me note that the influence arrow from fraudulent transactions to the burden of proof on the customer closes a second feedback loop. Walking along the influence links and considering their polarities, it can be recognized that this feedback look is reinforcing (R): if, e.g., the bank increases the burden of proof on customers, the chain of influences along the feedback loop 'R: ATM fraud epidemic', ultimately forces the bank to a further increase of the burden of proof on the customers.

The archetype in Figure 6 is an out-of-control archetype [4]. The balancing feedback loop 'B: Customer is liable' expresses the bank's intended consequence of its strategy, viz. to control fraud. The unintended consequence is expressed by the reinforcing feedback loop 'R: ATM fraud epidemic'. Reinforcing feedback loops can act viciously or virtuously, depending on whether they are triggered to increase or decrease unpleasant effects. In this case, the reinforcing feedback loop is vicious, indeed. Owing to the banks' refusal to recognize their prominent part in the bad ATM security[12] and time delays in the chain of influences, the crooks produced the avalanche of frauds that at long last caused major customers' dissatisfaction, loss of reputation and ultimately forced banks to improve the neglected ATM security [13] – at much higher costs than a well-designed proactive security would have required.

2.2 Online Identity theft

In this section, I present a case describing a situation when banks were promoting their own online services in order to decrease the costs of offline operation. Security level of these services was not strong enough[1], so crooks started actively abusing customers relying on the willingness of banks to save their public image rather than publicly acknowledge a pure security level of services and prosecute offenders.

The following description is the summary of trends on online identity thefts described in studies[1],[14] and [15].

2.2.1 Intended Consequence

One of the common ways to generate an unlawful profit in the online sector is to commit online identity theft. This option is becoming more and more attractive as business is moving to the online sector[3].

With the advent of the Internet it was discovered that moving business processes to the online sector could provide money savings, compared to offline operation[2]. One of the examples is the bank that created incentives for customers to use online services. As a consequence, customers adopted them (represented by variable "Customer adoption of online service" in Figure 7).

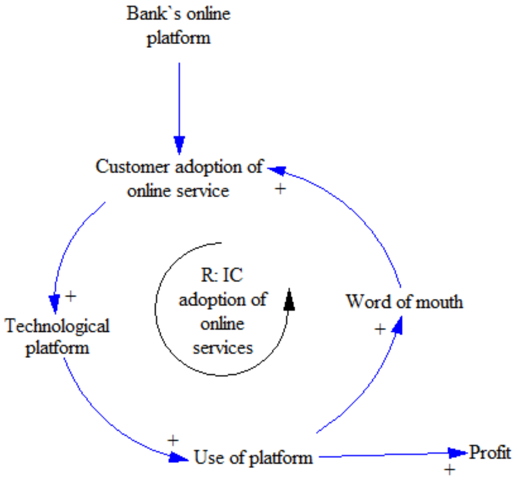


Figure 7: Customer adoption of online service.

The outcome of this action was an increasing portion of customers using online services (represented by variable "Use of platform") and increased profit to banks (represented by variable "Profit"). As more customers use platforms, they share their experience with other customers (represented by variable

“Word of mouth”), and, therefore, the augmentation in customers who currently use online services. As a corollary, I attain a reinforcing feedback loop, labeled “adoption of online services”.

2.2.2 Unintended Consequence

Meanwhile authentication procedures were not strong enough[1], which means many bank clients were insecure (expressed by variable “Insecure hosts” in Figure 8). As banks primarily rely on passwords, crooks started working out attacks, to take money from the customer (expressed by variable “hosts attacked by crooks”). For example, ‘keystroke loggers’ that can be installed on ATM or any PC that the bank‘s customer uses to perform online banking[6].

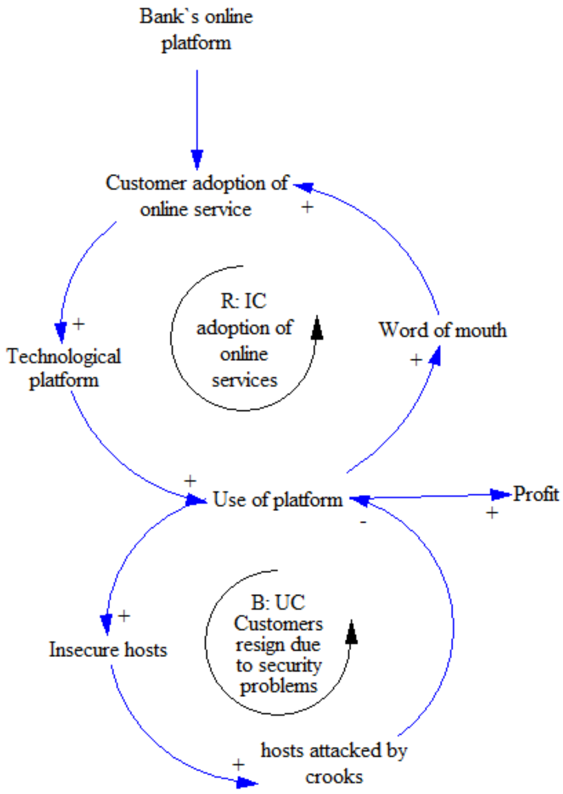


Figure 8: Customers resign due to security problems.

More sophisticated approaches resulted in ‘phishing’ attacks. An attack consisted of the following stages:

- Obtain banking credentials from the customer (i.e. by redirecting customer to fake web pages, that look like the bank‘s web page, where he\she is urged to type credentials).

- Sell bank details to the third party (known as a broker). The broker will sell credentials to specialist cashiers, who know how to launder the money[16].

In case the customer had approached the bank, claiming that he or she was the victim of fraud, the bank would not have accepted liability[6]. Depending on the situation, the bank either made the customer liable because his/her password was not strong enough, or proceeded with prosecution that might lead to the punishment of a cashier who laundered the money. In both scenarios customers lose trust into the online platform and cut down its usage (represented by the link with negative polarity between variables "hosts attacked by crooks" and "Use of platform"). Moreover, some customers managed to prove the bank's liability in court[17]. The unintended outcome is a balancing loop ("B: Customers resign due to security problems").

2.2.3 Solution

One of the prospective solutions is to "establish a program that regularly publishes the aggregated loss figures related to online banking and payment cards". Such a program should be established on the governmental level, but would succeed if promoted "from the bottom" (i.e. banks willing to comply)[1]. Having such a program in place will make publicly available precise and timely data on the online identity theft.

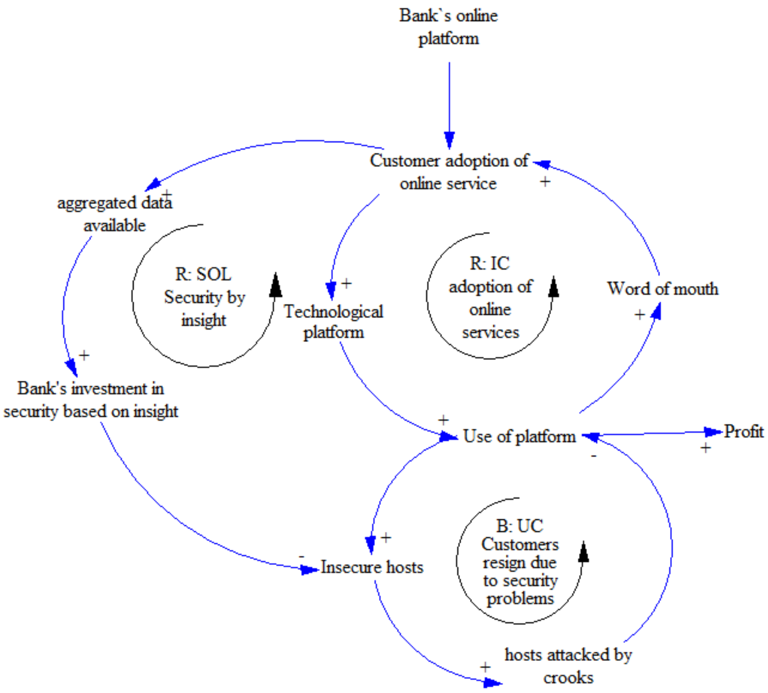

Bank`s online platform
Customer adoption of online service
aggregated data available
R: SOL Security by insight
Technological platform
R: IC adoption of online services
Word of mouth
Bank's investment in security based on insight
Use of platform
Profit
Insecure hosts
B: UC Customers resign due to security problems
hosts attacked by crooks
+
+
+
+
+
+
+
+
+
-
-

Figure 9: Security by insight approach.

Having good data (represented by variable “aggregated data available” in Figure 9) it will be possible to make better decisions on investments into security solutions or assessment of solutions that are already in place. Security research institutions would have the comprehensive vision of the problem. This would enable the development of good security solutions. By state regulations the bank would be obliged to deploy and comply with these security solutions (expressed by variable “Bank’s investment in security based on insight”), therefore, decreasing the number of potentially insecure customers (represented by the link with negative polarity between variables “Bank‘s investment in security due to regulations” and “Insecure hosts”)[1]. Thus, I get a balancing solution loop (“R: Security by insight”).

2.3 Patching large software

In this section, I present a case describing a generalized situation of patching large software products, that is according to[16] is in place in many large organizations, including Microsoft. The core problem is the imbalance between vast resources needed for protection and relatively small resources needed for attacking.

2.3.1 Intended Consequence

Because of marketing competition, Software vendors are forced to release their products already in beta-stage (expressed by “Released beta products” in Figure 10). This enables to quickly size the market and start getting the return on investment (expressed by “Profit”)[16]. Achieving this intended outcome makes the company trying to release even more products in beta stage (represented by the link with positive polarity between variables “Profit” and “Released beta products”). This results in reinforcing the feedback loop (“Beta products generate revenue”).

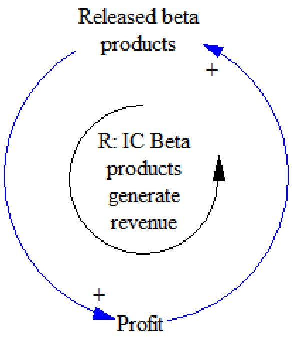


Figure 10: Released beta products.

2.3.2 Unintended Consequence

According to [16], an average software product released in beta stage has a high bug percentage (expressed by “Bugs” in Figure 11). Furthermore, the number of critical bugs, or in other words, exploits (expressed by “Fraction of critical bugs”), increases together with a cumulated number of bugs. Let us suppose that a huge Quality Assurance team was allocated to bug detection and patching, (its capacity 100 000 bugs/year), so every year the software improves significantly.

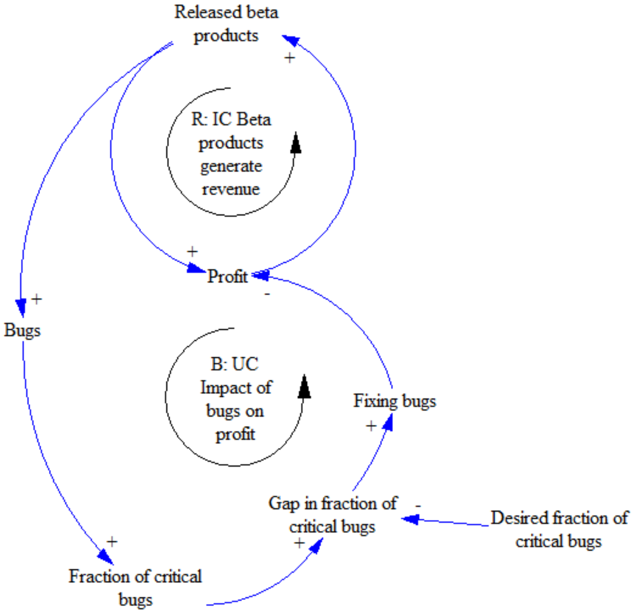

Released beta products
+
R: IC Beta products generate revenue
+
Profit
-
+
Bugs
B: UC Impact of bugs on profit
Fixing bugs
+
Gap in fraction of critical bugs
-
+
Desired fraction of critical bugs
+
Fraction of critical bugs

Figure 11: Presence of bugs affects profit from released products.

However, there is a hacker playing against Quality Assurance team. He has relatively low capacity (1 bug per year), but the probability that his discovered bug is not in the range of those 100 000 discovered by Quality Assurance is 90%[16]. This means that “desired fraction of critical bugs” for the company is extremely low (subsequently affecting variable “Gap in fraction of critical bugs” with negative polarity).

Therefore, a common approach is to make Quality Assurance team as big as possible, so that they can fix a large number of bugs (expressed by “Fixing bugs”). In [16] the author gives the example of employing 10 000 people to Quality Assurance department. However, it is economically impossible (and unreasonable) as it will decrease profits from product sales (described by a negative link between variables “Fixing bugs” and “Profit”). As a result, I get a balancing solution loop (“B: Impact of bugs on profit”), which mitigates the intended consequence of releasing products in beta stage.

2.3.3 Solution

The solution to this problem can be called a smart-selection approach (expressed by “smart selection approach” in Figure 12).

Firstly, not all the bugs are security-critical. A reasonable approach would be to separate security critical from non-critical. According to several researches, security-critical will total to only 1%. Concentrating on security-critical will provide drastic cost reductions (e.g. less staff needs to be employed)[16].

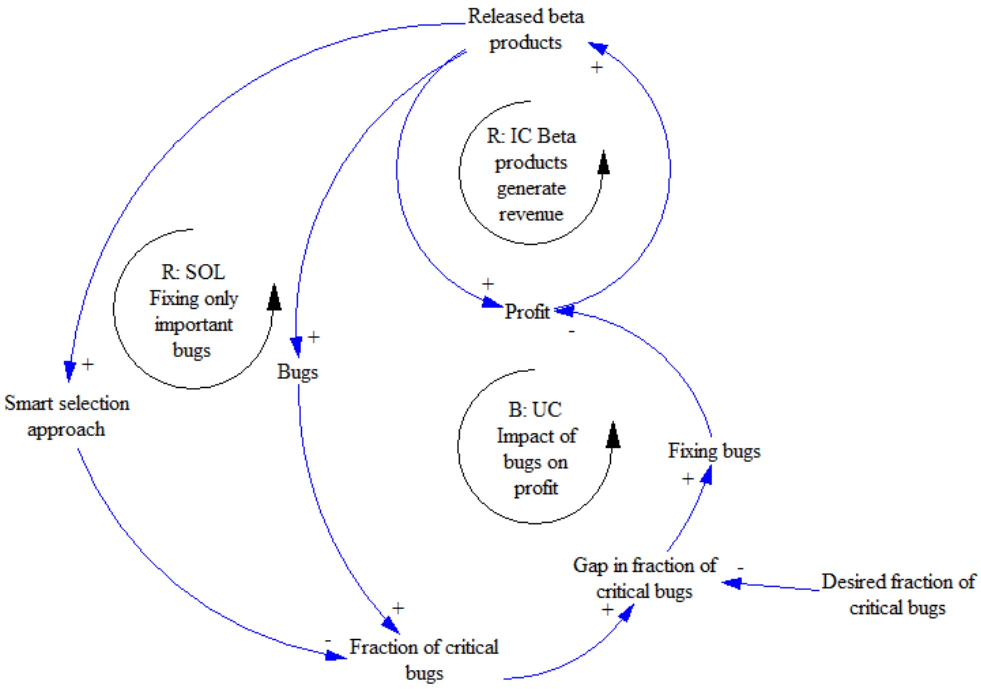

Released beta products
R: IC Beta products generate revenue
R: SOL Fixing only important bugs
Profit
Bugs
Smart selection approach
B: UC Impact of bugs on profit
Fixing bugs
Gap in fraction of critical bugs
Desired fraction of critical bugs
Fraction of critical bugs
+
-

Figure 12: Smart selection approach.

Secondly, it is important to understand that some vulnerabilities in implementation are based on respective vulnerabilities in technology standing behind this implementation. Therefore, if I fix vulnerability in technology, it will result in automatically removing a range of vulnerabilities in implementation. Hence, it is important to contribute into investigation of certain patterns in the range of vulnerabilities[16].

Given two facts will decrease “fraction of critical bugs” and form a reinforcing solution loop (“R: Fixing only important bugs”).

2.4 Software development process

In this section I present a case related to software development life cycle. The issue is to maintain balance between high quality products (for example highly secure) and its utility for customers (for example, good functionality due to innovative approaches and timely release to market). The vendors were supposed to carry responsibility for the security level of their products; however, they decided to shift this liability to the third party (software developers), by making them liable. In the long run this misalignment backfired on software vendors in terms of reduced software utility and, concomitantly, lower profit.

2.4.1 Intended Consequence

Because of marketing competition, software developers are forced to release their products already in beta-stage. This enables to quickly size the market and start getting return on investment, yet at the expense of having an enormous quantity of insecure software on the market[1]. In pursuit to improve an average software security level, software distribution companies make software developers liable of any consequences that will occur due to bugs in supplied software (expressed by variable "Burden on developers" in Figure 13)[1].In case a customer's complains are made due to vulnerabilities, the reseller's loss will be mitigated (represented by arrow with negative polarity between variables "Burden on developers" and "Reseller's loss due to vulnerabilities"). Essentially, it ensures that the desired profit from sold software is maintained (represented by the arrow with negative polarity between variables "Reseller's loss due to vulnerabilities" and "Reseller's Profit"). In the short term it gives the desirable result (intended consequence loop "B: IC Making developers liable").

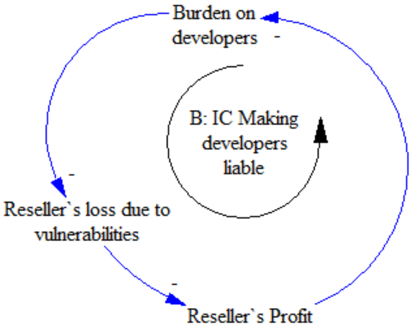

Burden on developers
-
B: IC Making developers liable
-
Reseller`s loss due to vulnerabilities
-
Reseller`s Profit

Figure 13: Burden on developers.

2.4.2 Unintended Consequence

In the long term, this brings significant changes to software development market in the form of reduction in the pace of innovation (represented by the arrow with negative polarity between variables "Burden on developers" and "Developers willing to innovate" in Figure 14). Naturally, if there is a threat of being sued for something, one will try to minimize the likelihood. Therefore, developers are no more willing to experiment, take risk and innovate, as no one will be happy to receive legal claim for some security failure, caused by his 5-year old code[1]. With a considerable time delay, this results in the decrease of Software utility for the end customer (represented by the arrow with positive polarity between variables "Developers willing to innovate" and "utility of Software for Customers"). As software becomes less useful, sales go down followed by the decreased profit for the reseller. As a result, I get a reinforcing intended consequence loop ("R:

UC Reduced pace of innovation").

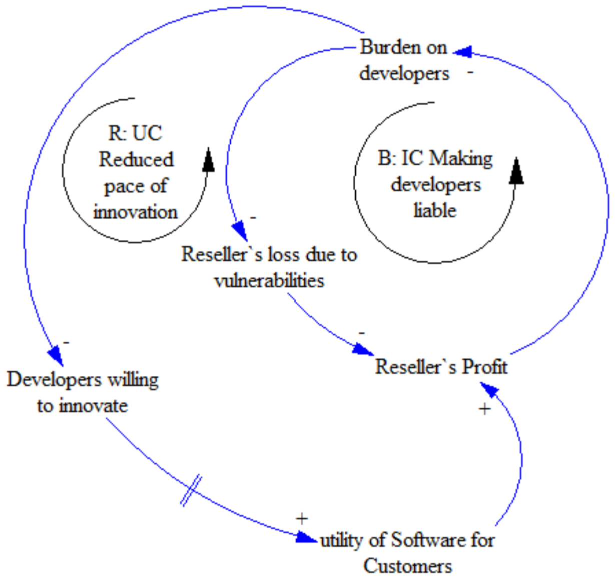

Burden on developers
-
R: UC Reduced pace of innovation
B: IC Making developers liable
-
Reseller`s loss due to vulnerabilities
-
Reseller`s Profit
-
Developers willing to innovate
+
+
utility of Software for Customers

Figure 14: Developers not willing to innovate.

To get insights into other unintended consequence, I will make a slight digression. George Akerlof got his Nobel prize for explaining insights on how markets with asymmetrical information operate [18]. In his work, he explored used car market, where cars for sale are in either good or poor state. As a car owner has more knowledge about his car quality than the buyer, this creates asymmetrical information. By practical data and theoretical models, Akerlof showed that equilibrium for such a situation is always the market flooded with bad-quality cars sold for overcharged price.

The same pattern can be traced in software development. Some companies are willing to develop highly-secured software, but market is already piled with badly-secured software sold more cheaply [1]. Customers are not willing to pay premium for protection and market comes to equilibrium, where only badly-secured software is available.

2.5 Sony Pictures hack 2014

In this section I present a case happened to Sony Pictures Entertainment Inc. Company (SPE), which is the American entertainment subsidiary of Japanese multinational technology and media conglomerate Sony. The hack was carried out in November, 2014, by a group of hackers named Guardians of Peace and resulted in leakage of several not yet released movie scenarios. One of the ground that made this hack possible was the misalignment of incentives: instead of balanced investments into core business objectives and security controls, security investments were almost neglected.

2.5.1 Intended Consequence

“The information security team is a relatively tiny one. On a company roster in the leaked files that lists nearly 7,000 employees at Sony Pictures Entertainment, there are just 11 people assigned to a top-heavy information security team. Three information security analysts are overseen by three managers, three directors, one executive director and one senior-vice president.”[19]

In the given quote, the approach towards security in Sony Pictures company is illustrated. Based on facts presented, the author of this work makes his assumption that the company had no clear incentive to invest into information security. The company was interested in investing (represented by variable “Investment” in Figure 15) into its main business - movie production. As a result, the investment capacity for movie production increased (represented by variable “Capacity”) and subsequently increased profit (represented by variable “Profit”). Thus, I get a reinforcing intended consequence loop (“R: IC Focus on primary business”). Due to the fact that security was not seen from the perspective of influencing net cash flow, the company‘s management had no incentive to invest in security[17].

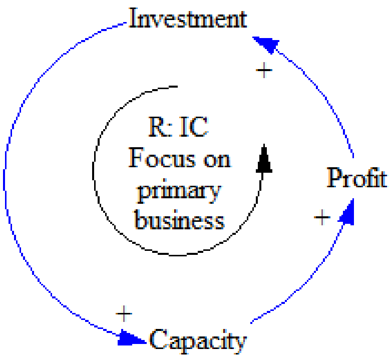


Figure 15: Investment into primary business needs.

2.5.2 Unintended Consequence

As a result, over time, the management and employees became complacent. The organization’s security level was decreasing, while its capacity to perform core business functions was growing (represented by the arrow with negative polarity between variables “Capacity” and “Security” in Figure 16), in the form of both organizational and technical measures[20]. This resulted in balancing unintended consequence loop (“B: UC Security in the shade”).

“In 2006, an auditor told him that Sony’s employees were using terrible passwords — nouns rather than random combinations of letters, numbers and symbols.” [17] Employees were violating basic security regulations, such as policy for strong passwords. There was no corporate culture in place to provide guidance about the importance of complying to Information Security regulations.

“A hack of our file server about a year ago turned out to be another employee in Europe who left himself logged into the network (and our file server) in a

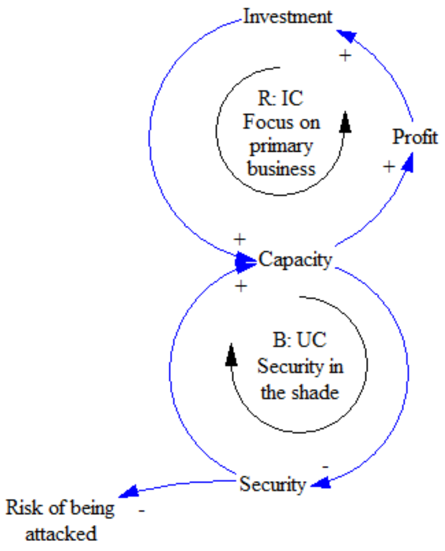


Figure 16: Security obsolescence.

cafe"[17]. Cases for violation of basic principles of using portable devices in public places were reported.

Again, because of the absence of proper culture, these cases were not prosecuted and no prevention measures were taken[17]. Information security team itself became complacent and ignorant to reported security violations. Below is given one of many examples.

"We'd report security violations to them, and our repeated reports were ignored. For example, one of our Central European website managers hired a company to run a contest, put it up on the TV network's website and was collecting personally identifying information without encrypting it"[17].

From the facts given above I assume that the risk of being attacked (represented by variable "Risk of being attacked"), was increasing simultaneously with security obsolescence (represented by the arrow with negative polarity between variables "Security" and "Risk of being attacked"). Indeed the situation remained unchanged for 9 years, until the actual attack in 2014.

In this attack not only personal data of employees was disclosed, but also a tremendous amount of commercially valuable data was compromised, for example, movie scenarios.

"A copy of the script for the upcoming James Bond film Spectre, whose release due was 2015, was obtained. Several future Sony Pictures films, including Annie, Mr. Turner, Still Alice and To Write Love on Her Arms, were also leaked"[20].

Mentioned intellectual property had an undoubtedly great value for the company. And for the first time the company understood the economical side of information security.

2.5.3 Solution

Attack resulted in comprehensive investigation. The direct impact was in the form of drastic changes in policy towards security investments. Security team was reorganized, and a proactive approach was implemented. The company started investing to security as well as core business (represented by the arrow with positive polarity between variables "Investment " and "Security resources" in Figure 17). Consequently, with security investment and team reorganization, an overall level of security in the company (represented by the arrow with positive polarity between variables "Security resources" and "Security") increased. As a result, I get a reinforcing solution loop ("R: SOL Security for growth's sake").

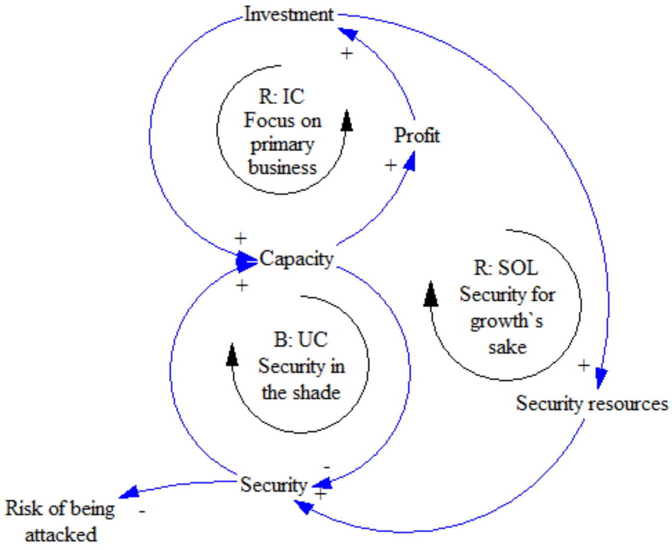


Figure 17: Investment in security.

Another implication was the discussion on the U.S. governmental level whether the framework for communication of security failures is to be implemented.

2.6 Internet Service Providers (ISPs) not willing to invest in security

In this section, I present a case describing a generalized situation of the U.S. ISPs being reluctant to invest in security. However, the situation changed in 1996 with the implementation of Communications Decency Act (CDA), but prior to this, ISPs had no clear incentive to take care of the content posted by their clients even if the content was illegal or malicious.[1]. This situation is another example of the party responsible for security (ISP) shifting liability to the third party (their customers) by stating that customers have to take care of security themselves. In the long term, this misalignment results in a demolished capacity of ISPs due to the need of vast reactive security measures.

2.6.1 Intended Consequence

Basic behavior for ISPs was by investing money (expressed by variable "Investment" in Figure 18) to maximize capacity (expressed by variable "Capacity") and profit accordingly [5]. As a result, I get a reinforcing intended consequence loop ("R: IC Focus on primary business").

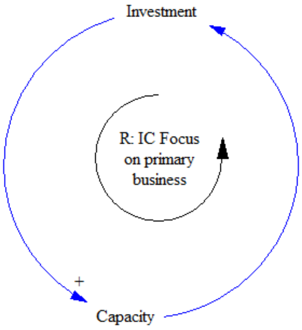


Figure 18: Investment in primary business functions.

2.6.2 Unintended Consequence

Investment in security occurred only if a particular problem influenced revenue[4].

As ISPs have to deal with customers who are themselves nodes in the network, it is important to care not only about ISPs‘s Security, but also about customer security. One of the ways of malware spreading is emails; thus, certain security investments needed to fight email malware.

ISPs were complacent as they did not see any solid connections between user security and profit itself. This assumption is supported by the following quote: "ISPs argued that emails were the personal property of recipients and that the inspection of the content of mails was the violation of privacy. Consequently, the responsibility for protecting their own machines and for dealing with spam was attributed to end users." [4]

Of course, incentives for such statements were first of minimization of own costs, rather than a proper approach to personal security regulations. The statement above gives a good example of ISPs not willing to improve security, while the capacity grows (represented by the arrow with negative polarity between variables "Capacity" and "Security" in Figure 19).

After a time delay side effects took place:

- "Flood of spam became a burden for network infrastructure that would have required additional investment" (represented by the arrow with negative polarity between variables "Security" and "Flood of spam")[4]. As a result, I get a balancing unintended consequence loop ("B: UC Spam decreases capacity").
- "Users of infected machines started to call the help desk or customer service at a fairly high cost per call to the ISPs." (represented by the arrow with negative polarity between variables "Security" and "Resources to customer support")[4]. This is an example when the wrong decision to shift the burden of proof from the company to customers fought back on ISPs. Namely, ISPs was affected by an increased load on Help Desk, thereby, hav-

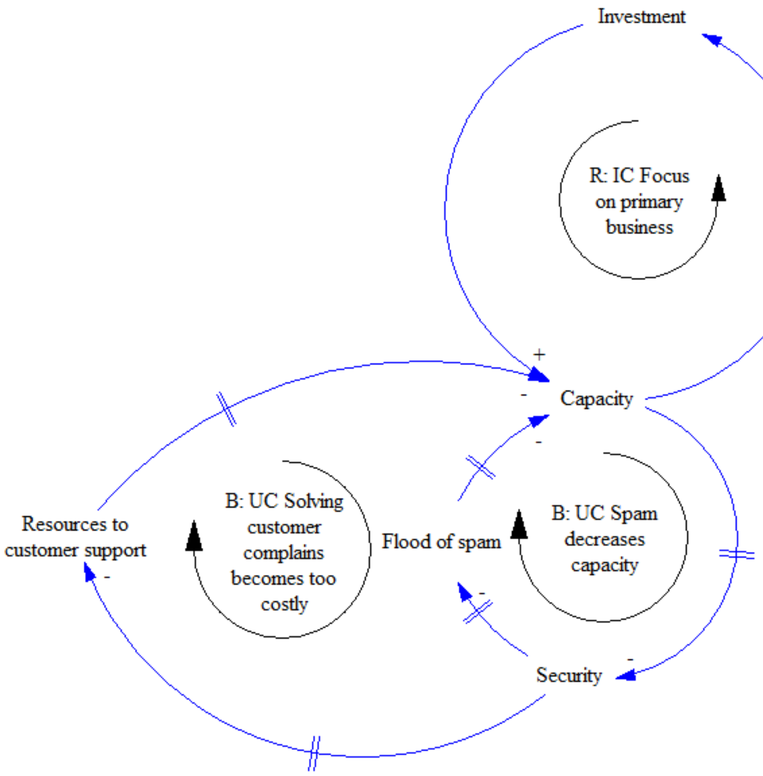

Investment
R: IC Focus on primary business
+
-
Capacity
-
B: UC Solving customer complains becomes too costly
Flood of spam
B: UC Spam decreases capacity
Resources to customer support
-
-
-
Security

Figure 19: Solving customer complains becomes costly.

ing decreased an overall capacity (represented by the arrow with negative polarity between variables “Resources to customer support” and “Capacity”). As a result, I get a balancing unintended consequence loop (“B: UC Solving customer complains becomes too costly”).

2.6.3 Solution

“ISPs reversed their stance with little fanfare and started to filter incoming mail and to manage their customers’ security more proactively (expressed by variable "Security resources" on Figure 20)[4]. As a result we get reinforcing solution loop (“R: SOL Security for growth‘s sake”).

Understanding that the solution is to take back the responsibility for security measures, ISPs changed tactics to a proactive approach. This provided gain for both customers and the company.

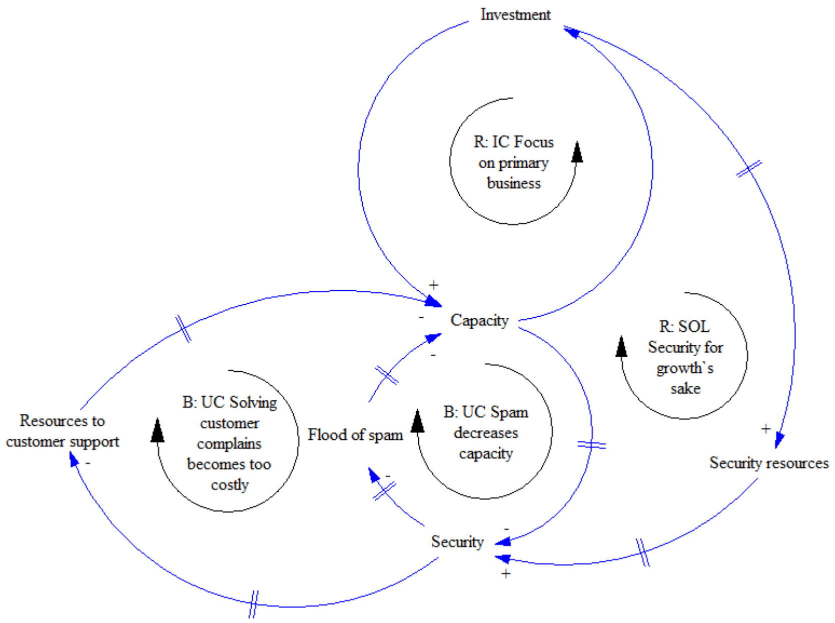

Investment
R: IC Focus on primary business
Capacity
R: SOL Security for growth`s sake
Resources to customer support
B: UC Solving customer complains becomes too costly
Flood of spam
B: UC Spam decreases capacity
Security resources
Security
+
-
-
+
-
-
-
+

Figure 20: Invest in security resources.

2.7 Policy for information disclosure

In this section I present a case describing a generalized problem of negative information (incidents, attacks, etc.) disclosure to rivals, authorities and public. Followed by an incentive not to spoil their reputation, victim companies decide to hide negative information. However, in the long term this negative information becomes known, and this creates even more damage to the company. Moreover, if the incentive initially was aligned on information disclosure, all market participants, including the company, would have benefited. The reputation loss for the company that experienced an incident would also be lower.

2.7.1 Intended Consequence

In case an attack or breach happens, companies have an option to disclose information. Routines for Information Disclosure are designed to notify other companies about past attack details. Based on the information, market players can make further assumptions about the likelihood of becoming a victim in future attacks and adopt some preventive measures.

Although sometimes an immediate effect of publishing negative information (expressed by variable "negative info available about this company" in Figure 21) about the recent attack is a significant fall in company‘s stock prices[1].

This can damage financial health of the company and subsequently its reputation (represented by the arrow with negative polarity between variables “negative info available about this company” and “Reputation”) which is highly undesired by the company‘s management. Following this incentive, most companies opt for a non-disclosure policy and even after publicity becomes aware of the incident, they try to give as few details as possible (represented by the arrow with negative polarity between variables “Reputation” and “Resistance to disclose information”) . This action derives an intended outcome by way of less reputational damage, both in the form of goodwill and stock prices (as a result, I get a balancing intended consequence loop “B: IC Save reputation by hiding details”)[1].

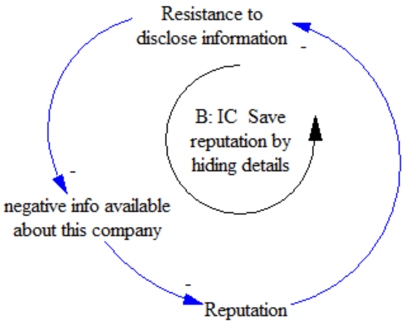

Resistance to disclose information
-
B: IC Save reputation by hiding details
-
negative info available about this company
-
Reputation

Figure 21: Saving reputation by hiding details.

2.7.2 Unintended Consequence

Even though a non-disclosure policy has a negative effect on everyone, except for the company, in the long run a situation changes. Hidden information (expressed by variable "Hidden information " in Figure 22) becomes known (expressed by variable "effect of hidden info becomes known ").The goodwill of the company

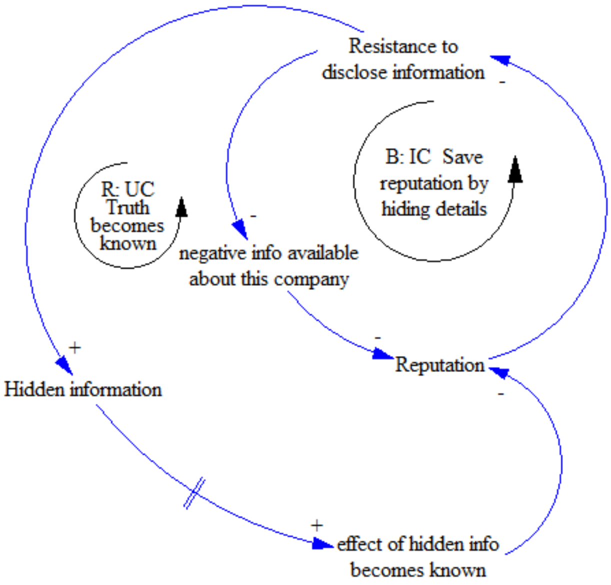

Resistance to disclose information
-
B: IC Save reputation by hiding details
R: UC Truth becomes known
-
negative info available about this company
-
Reputation
+
Hidden information
-
+
effect of hidden info becomes known

Figure 22: Truth becomes known.

drops and consequentially decreases trust between the company and contracting parties (represented by the arrow with negative polarity between variables “effect of hidden info becomes known” and “Reputation”)[1].

As a result, I get a reinforcing unintended consequence loop (“R: UC Truth becomes known”).

2.7.3 Solution

One of prospective solutions to this problem is mandatory information disclosure (expressed by variable "Mandatory disclosure " in Figure 23)[1]. This would be a powerful tool for eliminating information asymmetry and correcting misaligned incentives. More specifically, the company will because of regulation publish all hidden information and, in such a way, less information will be hidden (represented by the arrow with negative polarity between variables “negative info available about this company” and “Hidden information”).

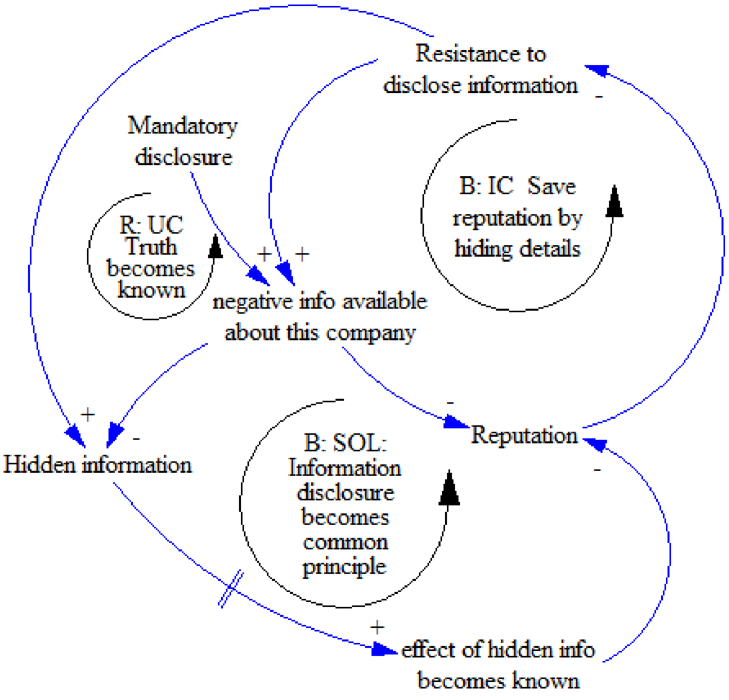

Resistance to disclose information
-
Mandatory disclosure
B: IC Save reputation by hiding details
R: UC Truth becomes known
+
+
negative info available about this company
+
-
-
Hidden information
Reputation
-
B: SOL: Information disclosure becomes common principle
+
effect of hidden info becomes known

Figure 23: Mandatory disclosure as solution.

In case it becomes a common practice, some unintended consequences will be eliminated over time (I get a reinforcing solution loop “B: SOL: Information disclosure becomes a common principle”)[1].

2.8 The TARGET corporation data breach

The TARGET is one of the largest retail chains in the U.S. It operates around 1,801 stores all over the U.S. There are three types of stores: a discount store, hypermarket and small-format stores. The company has headquarters in Minneapolis and around 347 000 employees[21]. TARGET company followed incentive to over-rely on a technological security solution they had and, hence, to neglect organizational aspects. This misalignment resulted in a demolished security level and made the attack of 2013. The author assumes that there is a high chance that an attack would not happen in case incentives were properly aligned on both technological and organizational security measures.

2.8.1 Intended Consequence

A typical retailer company is able to discover security breaches in only 5% of cases. This was revealed in a study carried out by Verizon Enterprise Solutions[22]. The primary reason for such a low likelihood is incentive to concentrate most attention on a primary mission, rather than on security[23].

TARGET company was not fitting this pattern. The information security department had around 300 employees and front new intrusion detection software FireEye was adopted (expressed by variable "Investment in Technology solution" in Figure 24). Superiority features of this security solution are described in the following quote: “Unlike antivirus systems, which flag malware from past breaches, FireEye’s is not as easily tricked when hackers use novel tools or customize their attack”[22].

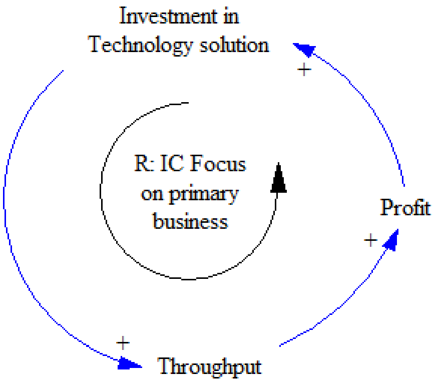


Figure 24: Investment in Technology solution.

This allowed remaining protected against various security treats and, thus, maintaining a high throughput (expressed by variable «Throughput»). Throughput ultimately generated more profit (expressed by variable "Profit") and this allowed investing further in technology (represented by the arrow with negative polarity between variables “Profit” and “Investment in Technology solution”).

As a result, I get a reinforcing intended consequence loop (“R: IC Focus on primary business”).

2.8.2 Unintended Consequence

Having such good capabilities for defense created incentive for TARGET to neglect human and organizational aspects (expressed by variable "Neglecting Organizational factors" in Figure 25). As an example of bad organizational aspect, TARGET‘s IS team opted to turn off the option “automatically delete detected malware”. Incentive was to give more control to IS team, but it might actually create a lot of pressure on the team when a quick action is needed[22]. In general, it decreases the security level in the company (represented by the arrow with negative polarity between variables “Neglecting Organizational factors” and “Security”).

A demolished organizational aspect in Information Security approach made 2013 TARGET attack possible. The attack started in November, 30, and the first alarm came soon. Although, TARGET was neglecting them for the next 15 days. TARGET ignored notifications from its own security team in India, notification from FireEye system, and from the third parties. Only after notification from the US Federal Law enforcement in December, 12, TARGET took emergency actions to stop breach.

The direct financial impact was about $61 million as a response to the breach. Since TARGET faced a certain decrease of goodwill, many customers were not willing to shop there any more[23]. Anyway throughput of the company to perform core business has dropped (represented by the arrow with positive polarity between variables “Security” and “Throughput”).

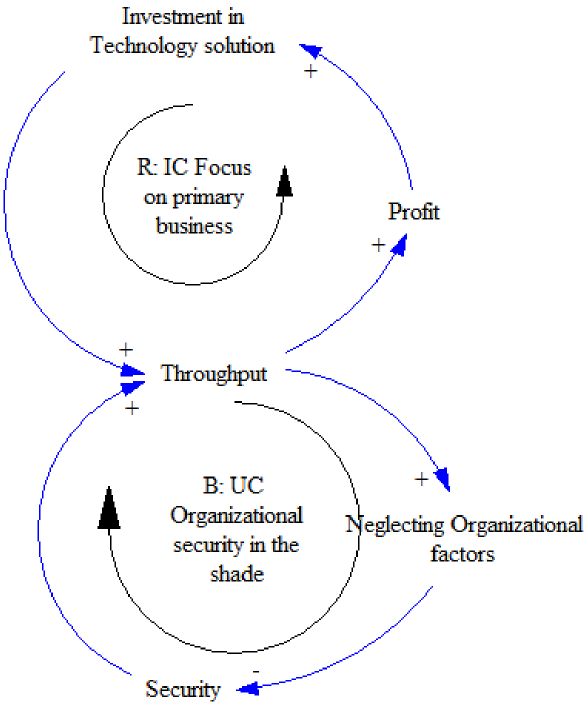

Investment in Technology solution
+
R: IC Focus on primary business
Profit
+
+
Throughput
+
B: UC Organizational security in the shade
+
Neglecting Organizational factors
Security
-

Figure 25: Neglecting Organizational factors.

As a result, I get a balancing unintended consequence loop (“B: UC Organizational security in the shade”).

2.8.3 Solution

The analysis of a given case gives insights into the importance of focusing not only on technical, but on organizational and human aspects of information security (represented by the arrow with negative polarity between variables “Investment in Technology solution” and “Organizational aspects training” in Figure 26). As a result, I get a reinforcing solution loop (“R: SOL Organizational aspects of security for growth‘s sake”).

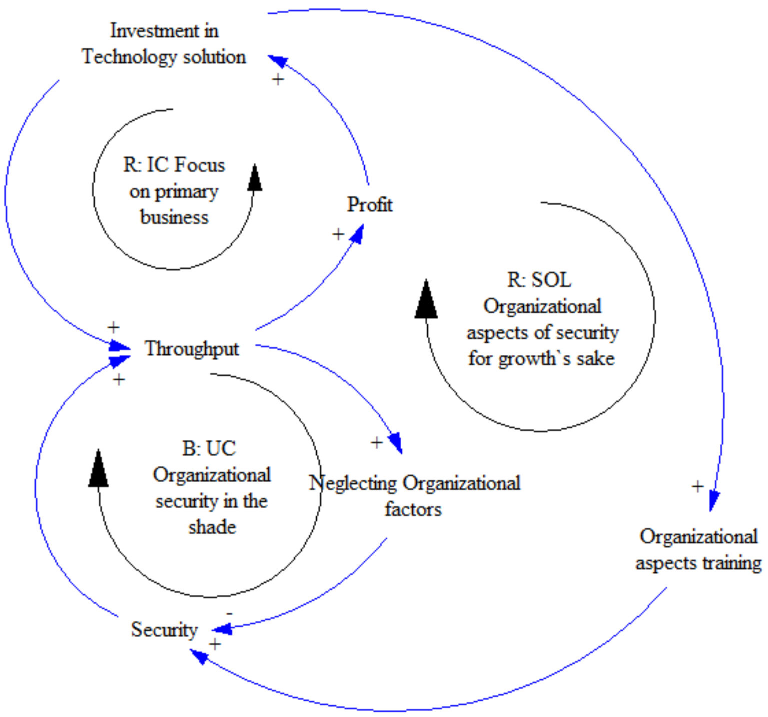

Investment in Technology solution
R: IC Focus on primary business
Profit
R: SOL Organizational aspects of security for growth`s sake
Throughput
B: UC Organizational security in the shade
Neglecting Organizational factors
Organizational aspects training
Security
+
+
+
+
+
+
+
+
-

Figure 26: Organizational aspects of security for growth‘s sake.

After TARGET breach concepts of rising awareness and cybersecurity legislation were widely discussed on the governmental level[21]. TARGET worked on improving organizational aspects and adopted a new approach, when automated incident response is always preferable than trying to manually investigate and respond every occurring incident.

2.9 Office of Personnel Management (OPM) data breach

“The United States Office of Personnel Management is an independent agency of the United States government that manages the civil service of the federal government.” [24] In this section, I present a case describing data breach happened in June, 2015, when records of almost 22.5 million people were leaked. This event is among the largest breaches of the governmental data in the history of the United States[24]. In the organization of a similar scale, IS department is expected to be responsible for security measures; however, the responsibility was shifted to the third parties: all departments were told to take care of security themselves. The author assumes that this misalignment was one of the reasons for 2015 data breach.

2.9.1 Intended Consequence

Many years of operation without incidents resulted in the established confidence in good security and security complacency. In order to decrease workload and, hence, increase throughput (expressed by variable "Throughput" in Figure 27), it was decided that the central IT staff should not govern projects in OPM, the duty and responsibility should be placed on the division level (expressed by variable “Shift to distributed IT systems”)[24].

Although for cost reduction many systems were operated by agency contractors, which were of course not under the direct control of the central IT department (expressed by variable “Shift to distributed IT systems”).

As a result, I get a reinforcing intended consequence loop (“R: IC Increase throughput by making distributed IT”).

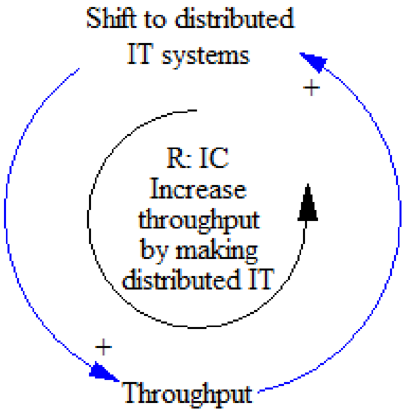

Shift to distributed IT systems
+
R: IC Increase throughput by making distributed IT
+
Throughput

Figure 27: Shift to distributed IT systems.

2.9.2 Unintended Consequence

The given structure made impossible to manage the IT system based on comprehensive risk assessments and incident response plans[24].

The whole structure security level was decreasing, while throughput increased (represented by the arrow with negative polarity between variables “Throughput” and “Security” in Figure 28). Some security flaws are listed below: “Eleven out of forty seven major systems belonging to OPM IT, not contractors, were not certified as secure.In other words 65% of all OPM data was stored on uncertified systems.”[24].

As a result we get balancing unintended consequence loop (“B: UC Security breaches due to poor

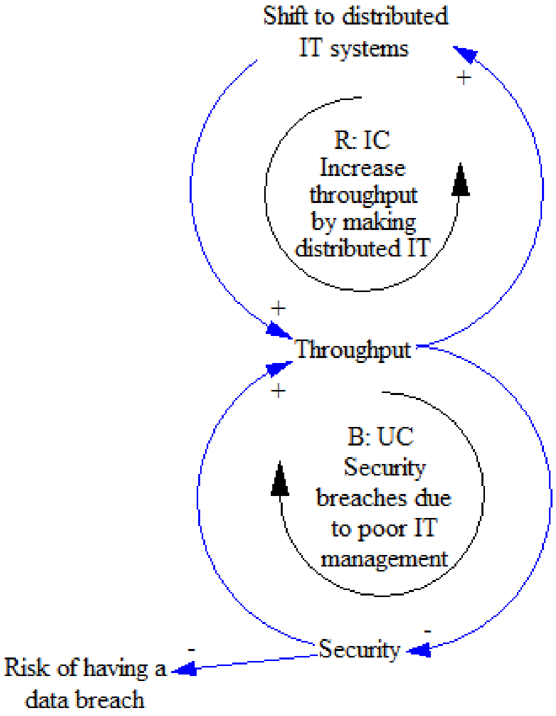

Shift to distributed IT systems
+
R: IC Increase throughput by making distributed IT
+
Throughput
+
B: UC Security breaches due to poor IT management
-
Security
-
Risk of having a data breach

Figure 28: Security obsolescence.

IT management”). The company suffered from data breach that begun in Nov. 2013 and lasted for almost one year (expressed by variable "Risk of having a data breach"). During this time personal information of nearly 22.5 million Americans was exfiltrated and the motives of attacker are still unknown. Although some trace it to Chineese military[24]. Data was stored in form of so-called Standard Forms that are extremely detailed and has mature content. Some of the affected people had highly classified clearances.

2.9.3 Solution

One of the prospective solutions was the implementation of an integrated security management approach and transferring all security responsibilities to the centralized information security department (expressed by variable "Centralized management of IT systems" in Figure 29).

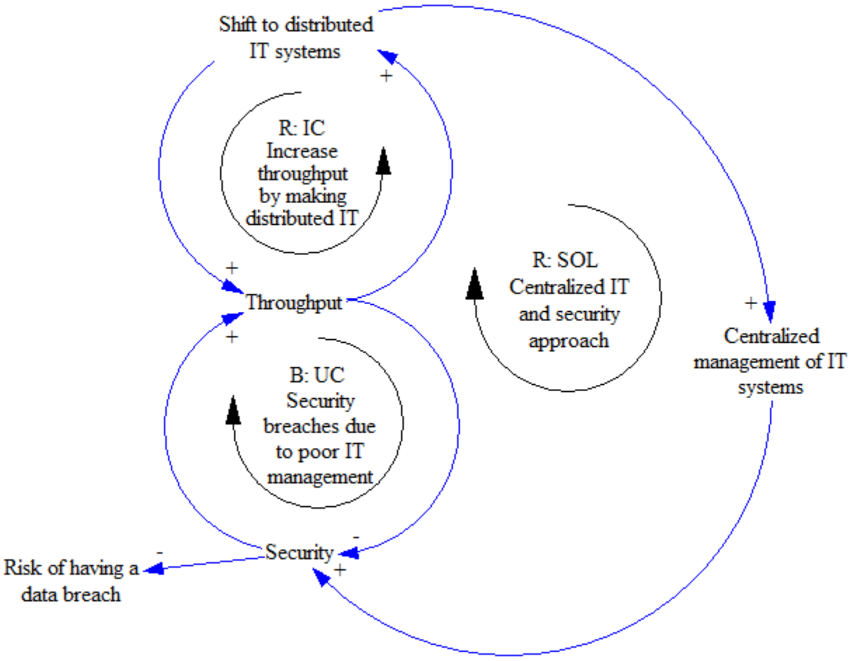

Shift to distributed IT systems
R: IC Increase throughput by making distributed IT
R: SOL Centralized IT and security approach
Throughput
B: UC Security breaches due to poor IT management
Centralized management of IT systems
Risk of having a data breach
Security
+
-

Figure 29: Centralized management of IT systems as solution.

As a result I get a reinforcing solution loop (“R: SOL Centralized IT and security approach”).

2.10 Selection of cases for modeling

Although the cases described above are interesting and have a distinctive learning outcome, my Master’s Thesis aims to perform system dynamics modeling only for one selected case - ATM Fraud in the US and Europe. The selection of one case only is mostly due to the inability to gather reliable numerical data for other cases necessary for modeling. From my perspective, the selected case will result in quite a robust stock and flow model, because it is particular to the extent that core system components and their interdependencies can be identified from the literature review. From a different angle, the case is not limited to a particular bank, so it is general enough to describe a trend in ATM introduction between 1982 and 1993. Not least, contrary to other examples with evidence of misaligned incentives, in this case I have two independent actors, the US banks and European banks, who are dealing with the same problem in a different manner. Therefore, the benefit of stock and flow model developed in Chapter 3 would be in providing a comparison of outcomes for these two actors.

3 The ATMs in the USA and Europe and Quantitative models (stock-and-flow diagrams)

3.1 Introduction

The following description is derived from the study [16] concerning events occurred between 1982 and 1993 for various vulnerabilities and backdoors. A lot of customers suffered from frauds, bank employees' mistakes and technology failures, also known as "phantom withdrawals". Many customers who experienced "phantom withdrawals" were not satisfied by banks. On the other hand, banks suspected customers of trying to cheat and commit a fraud.

But for some cases being taken to court and the solution found, the problem remained complicated. It seemed that the case would ultimately be lost by whichever party has the burden of proving that the other party is wrong. Consider a customer approaching court and claiming that he has not made a transaction. The bank refuses the customer's claim, and the court assigns a customer to prove their innocence, or in other words, give explicit evidence where exactly the vulnerability in a bank system that was used to perform withdrawal is. This is a hard task, especially without proper background and comprehensive knowledge of all bank systems. So, the customer will lose anyway. In case a bank is asked to prove that their systems are secure, it is very challenging to be performed, so that it is convincing enough for hostile experts.

In the US the first known case was Dorothy Judd v Citibank [25]. In this case Dorothy Judd, a customer of Citibank, experienced a phantom withdrawal of 800 USD form her account. The Citibank denied to accept technology failure and insisted that Dorothy Judd had actually made transaction. After the case was brought to the Civil Court of the City of New York it was decided that the customer's claim that she had not made transaction outweighed the bank expert's claim that she must have done so.

Another case happened to Norma Hendy from Plymouth. Her colleague experienced a phantom withdrawal. Among employees it was considered acceptable to ask one another to go and launder the cash; therefore, victim‘s PIN was known. Among many prospective suspects, Norma was the one to be arrested, and the reason was that the victim‘s purse was left intact in her vehicle for the whole day. The police neglected the fact that the car was unlocked, so potentially anyone might have utilized the card. Of course. Norma denied the accusation, but the bank supported it saying that the withdrawal could not possibly have been made in any other way except with the card and PIN issued to the victim. This was untrue as both theft by bank staff using extra cards, and card forgery by outsiders were known to affect this bank's customers. The Defense, therefore, asked the bank to provide security policy, including results of recent audits. The bank refused to do so, and the court decided that Norma was innocent. A similar situation has happened in Great Yarmouth. One of the cab drivers was charged for stealing 50 USD from his fellow employee. In reality, there was a phantom withdrawal in place. Similarly to the previous case, the advocacy asked the bank for security policy and audit results. Once the bank refused to provide them, the case collapsed.[26].

Based on these cases the US Federal Reserve passed regulations that "require banks to refund all disputed transactions unless they can prove fraud by the customer" [7].

British bankers, of course, were aware of this regulation, but claimed that applying this regulation in Britain would have bad consequences, for instance, the avalanche of customer frauds (i.e. a customer has actually withdrawn money, but wants to deliberately steal money from his bank by claiming that he has not made a transaction) [27]. In most situations British banks claimed that their systems cannot be misused; subsequently, it is impossible to have a successful withdrawal transaction if no card was actually inserted into ATM and correct PIN was applied. Customers who objected were frequently told that they were either mistaken or trying to deceive the bank. Another popular response was that they were likely to have been affected by the fraud performed by their friend or family member (i.e. by steeling their card and overlooking their PIN), and therefore, the bank bore no liability for losses. After a while banks made a small security improvement: British banks introduced hardware security modules (the US banks used encryption at a software level instead). However, it had no significant influence on a fraud trend; therefore, the author of [28] concluded that this was a measure to sound more convincing when fighting a customer, rather than a real security improvement.

3.2 Stock and flow models

In Section 2.1 I have developed a qualitative model, also known as a causal loop diagram in order to picture system components and show causal relations between them. Causal loop diagrams are good for conceptualization, but in order to quantify the behavior of the system I need more complex stock and flow diagrams. I will now develop a model that will comprise both European and the US ATM cases. Once model is built, one will be able to test both strategies and, depending on initial conditions, see quantitative differences in the behavior of the system. It is also possible to simulate the model on different time intervals, which provide a good learning effect.

In Figure 30 the domain of ATM vulnerabilities is shown. It is a part of the complete diagram which I will introduce in this chapter. Vensim models are built using different types of variables. Variables shown in boxes (like 'ATM vulnerabil-

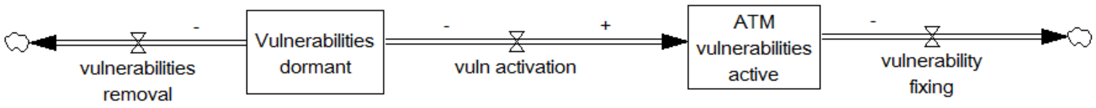


Figure 30: ATM Security domain.

ities active') are stock variables that accumulate or de-accumulate through flow variables (like 'vuln activation' or 'vulnerability fixing'. The equations defining these variables are reformatted by Vensim and represent integration (stocks) or differentials (flows). There are also auxiliary-type variables appearing on further diagrams that are defined by equations involving the inputs represented by the single-lined arrows. "Variables" having only outgoing influence arrows are actually constant parameters (which may be changed by decision makers playing serious games to conduct what-if-experiments and learn from the experiences).

Contrary to the approach that I used while developing causal-loop diagrams (namely, step by step going along the intended consequences, then unintended ones, and finally, approaching a solution) I will build different domains of the model first and then bring them together. However, once the model is completed, it will make possible to verify that the stock and flow model actually corresponds to the causal-loop diagram. The domains that I use will sometimes overlap because they have common variables and are tightly interconnected. I will begin model introductions with a domain describing ATM Security and then proceed with domains describing fraud rate and banks' attitude to security. In our explanation I will sometimes jump from one domain to another in order to describe relations between variables in the best possible manner.

3.3 ATM Security domain

Consider ATMs that banks were using. For the scope of our problem the most interesting part is vulnerabilities residing inside ATMs. According to [17] technological implementation, standing behind ATMs has not changed a lot; hence, for modeling purposes I can assume that when ATMs were introduced, they had a certain number of vulnerabilities, but no new vulnerabilities were added afterwards. However, it is important to distinguish between the vulnerabilities that have not been discovered yet (in Figure 30 expressed by variable "Vulnerabilities dormant ") and vulnerabilities that potentially can be exploited currently ("ATM vulnerabilities active").

According to the problem formulation, "ATM vulnerabilities active" result from "Vulnerabilities dormant " becoming discovered (expressed by variable "vulnactivation"). I will implement it as a classic structure of two stocks and flow transferring content of one stock to another.

After some time, the bank will understand that a particular class of fraud is committed by exploiting a particular active vulnerability. Essentially, the bank will fix this vulnerability (expressed by outflow “vulnerability fixing”) [29]. In our model I am not interested in the analysis of patched vulnerabilities (e.g. fixed vulnerabilities that are outside the system boundary); thus, I denote by cloud the end of flow “vulnerability fixing”.

In general, it is possible for a bank to use a proactive approach: find and patch “Vulnerabilities dormant” before they become known to public. This proactive activity is expressed as outflow "vulnerabilities removal" from stock "Vulnerabilities dormant". Similarly to the reactive approach, removed vulnerabilities are outside the system boundary which is denoted by cloud.

3.4 Fraud domain

As shown in[16] there were certain frauds committed in ATM infrastructure. In my analysis it is important to distinguish between frauds committed by bank customers and by non-bank customers, further referred as crooks. Crooks in general were people with some insider knowledge, for example bank employees, or without it. An accumulated number of frauds was steadily increasing over time, while the fraud rate (e.g. the number of frauds committed per month) was fluctuating based on fraud opportunities available. Now let me consider the nature of these possibilities - they were in most cases created by vulnerabilities in ATMs [30]. Moreover, only “ATM vulnerabilities active” can be used to commit a fraud. In other words, the fraud rate is dependent on “ATM vulnerabilities active”. This reasoning can be applied to both "customer fraud rate" and "crook fraud rate"; that is why, we introduce two similar structures as shown in Figure 31.

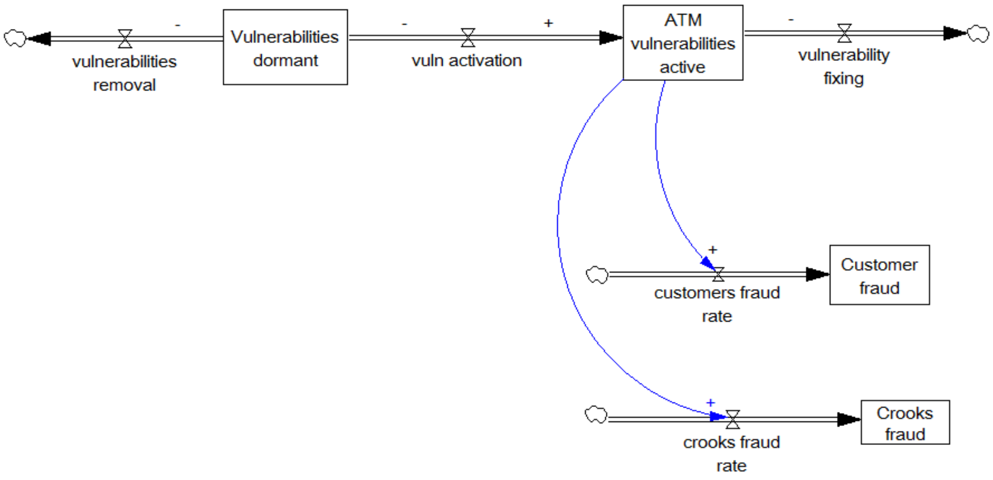


Figure 31: Fraud domain.

Indeed, many frauds were caused by vulnerabilities. Let me consider the following vulnerabilities that were present both in European and the US ATMs:

- *A complete account number is printed on the receipt*. Supposedly, a crook is capable of observing customers' PINs, while standing in a line to ATM. After a customer has finished his session with ATM, a crook would collect dropped receipts. Afterwards a crook buys some blank cards with a magnetic strip and copy the customer‘s account number to it. Having a PIN and such a card, the crook will be able to loot the customer's account at arbitrary ATM [31].
- *ATM authenticate and authorize responses were not encrypted.* In this case an attacker can install foreign hardware on ATM, so that they will be able to record ATM‘s pay response. Once recorded, this command can be repeated until ATM‘s cash storage is empty [32].
- *Backdoors created for testing purposes* ATMs in one bank had a backdoor: if a defined 14 digit sequence is entered, ATM will give 10 banknotes. To explore this vulnerability randomly is quite challenging because of a number of all possible combinations. However, the task was greatly simplified as this option was described in one of old manuals [33].
- *Utility at cost of security*. One bank treated an inserted telephone card as a valid card of the previous customer. The only challenge for a crook would be to observe the PIN of a previous customer and successfully launder money [34].

There is a positive intervention of a "total fraud rate" on "vulnerability fixing", yet, additional factors should be taken into account. Firstly, it is not a "total fraud rate" that serves as a primary indicator for a bank, but rather the quotient of a "total fraud rate" divided by the industry‘s average "worst case fraud rate". Secondly, a "vulnerability fixing" process largely depends on whether a bank uses the US or European strategy, in other words, whether there is "a burden of proof on a customer or a bank". The resulting structure of ATM and Fraud domain is displayed in Figure 32.

3.5 Formulating equations for dependent variables.

Let me consider the model that I have so far. I can see three types of variables: stocks, flows and auxiliary variables. There is a certain equation standing behind each variable. The list of all models' equations is given in Appendix A, but here I discuss some examples in order to describe the approach I used.

3.5.1 Stock equations

According to the System Dynamics concept [35] the stock variables can be changed only through inflows and outflows, but not by any auxiliary variables. If I draw a structure of stocks as in Figure 32, Vensim will automatically assign equations for stocks, using the following principle: the value of stock at a current time step

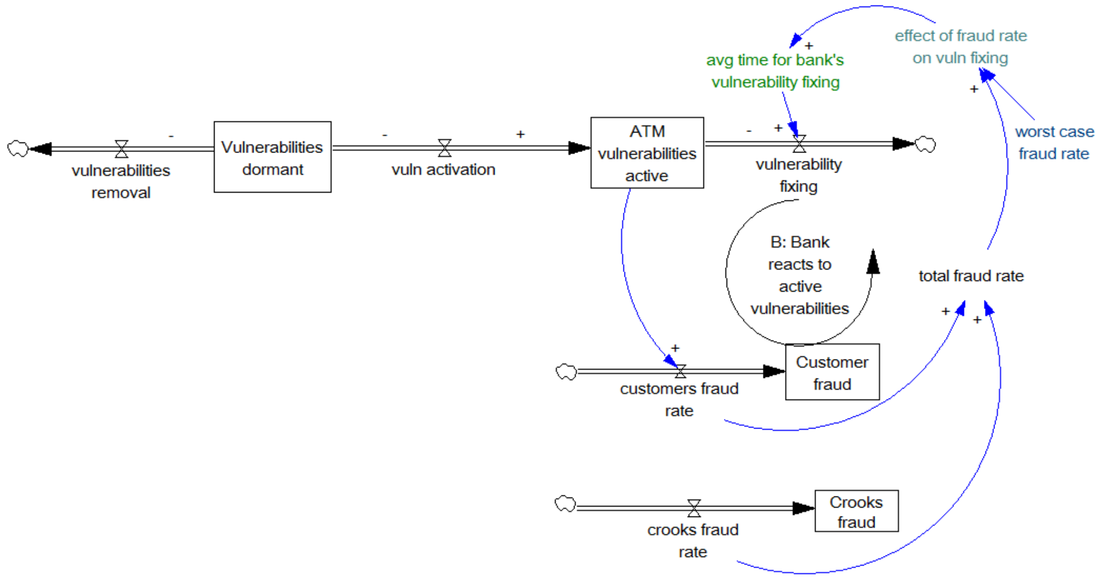


Figure 32: Fraud and ATM domains.

equals to the value of stock at the previous time step added to all inflows and subtracted from all outflows. For example,

$$\text{Vulnerabilities dormant} = \int(-\text{vulnerabilities removal}+\text{vuln activation})\,dt$$

The only challenge left is to formulate the initial value for the stock. Let me assume that there is a certain number of vulnerabilities in ATM system. At the initial point of time some of them are dormant, some are active. There might be several scenarios, depending on the target bank, so I picked up one of them, where 90% of all vulnerabilities were dormant and 10% were active. Thus, the equations are the following:

$$\text{INITIAL(Vulnerabilities dormant)} = 0.9 * \text{max number of vulnerabilities in system}$$

$$\text{INITIAL(ATM vulnerabilities active)} = 0.1 * \text{max number of vulnerabilities in system}$$

3.5.2 Flow equations

Contrary to stocks, flows are changed by auxiliary variables. The first challenge comes with keeping a consistent dimension of the flow. Dimension is defined by all the variables used in the equation; however, it should always be equal to the dimension of stock divided by time unit. If I am dealing with outflow, the challenge is also to formulate outflow in a way that it will never make the stock having a negative value. Consider the way I formulate "vulnerability fixing":

$$vulnerability\ fixing = \frac{ATM\ vulnerabilities\ active}{avg\ time\ for\ bank's\ vulnerability\ fixing}$$

The nature of vulnerability fixing depends on two factors: how many vulnerabilities active are actually there, and what is the time that bank uses to fix these vulnerabilities. Still, it is important to note the asymptotic nature in this process: in fact, the stock of vulnerabilities will never reach 0, but will asymptotically decrease to its value[36]. There is certainly some ratio at which a bank will fix vulnerabilities (denoted as "normal time for vulnerability fixing"), but if its management starts worrying, they will impose pressure on a security team. To address this pressure I use "the effect of fraud rate on vuln fixing". This variable is defined by so-called table function and will be discussed in detail below, but at the moment it is more important that the range of this variable is from "1" to "min value". This effect is multiplicative; as a result, "1" corresponds to the situation when there is no pressure at all and "min value" corresponds to the situation with maximum pressure. Then, the resulting formulation of outflow is:

$$vulnerability\ fixing =$$

$$= \frac{ATM\ vulnerabilities\ active}{effect\ of\ fraud\ rate\ on\ vuln\ fixing * normal\ time\ for\ vulnerability\ fixing}$$

3.5.3 Table functions

Sometimes the dependence between variables cannot be described as an equation, so it is necessary to introduce a table function (or look-up function, as it is called in Vensim). Initially, a model developer specifies some known pairs of (input,output) values. After this the output for any possible input is computed based on the approximation procedure of existing points. There are certain requirements to defining a table function:

- **Dimensionless**. Input and output of the function should have no dimension.
- **Normalized**. Input and Output domains are in range from 0 to 1 [37].

In Figure 32 "the effect of fraud rate on vuln fixing" is a look-up function. I will now walk through the procedure of its definition.

As can be derived from the variable name, the main goal is to express the influence of fraud rate (dimension "Fraud/Month") and the time of vulnerability fixing (dimension "Month"). In order to have dimensionless and normalized input, I take the quotient of "total fraud rate" divided by "worst case fraud rate". Because both variables have the same dimension "Fraud/Month", the input is indeed normalized. Now consider the nature of "worst case fraud rate"- it is a fraud rate that is assumed to be the biggest possible; hence, "the total fraud rate" will normally be lower than "the worst case fraud rate". Both variables are non-negative; therefore, our input is, indeed, normalized.

I start the definition of my table function with the margin values of input (0 and 1). If the input equals to *0*, it means that the total fraud rate also equals to *0*. Hence, there should be no pressure on the security team to remove vulnerabilities. Keeping this in mind, it is a multiplicative effect, I assign the output value 1, so that all the vulnerability fixing will be done through "the normal time for vulnerability fixing". Now consider having *1* as the input. This means that "the total fraud rate" equals to "the worst case fraud rate", so there is maximum pressure on the security team. However, the capacity of the security team is limited and, hence, there is some minimum value of "avg time for a bank's vulnerability fixing". I assume security team to be under pressure to fix vulnerabilities 5 times faster compared to vulnerability fixing without pressure. Therefore, for input *1* I assign the output *0,2*.

The next step is to define the shape and slope of function. According to the way the function is formulated, it will have a negative slope, because the greater the "total fraud rate", the faster the security team fixes vulnerabilities and, hence, the smaller is "avg time for a bank's vulnerability fixing". The "avg time for a bank's vulnerability fixing" is directly proportional to "the effect of fraud rate on vuln fixing"; as a corollary, the latter will also decrease if the fraud increases. Consider now the perception of "the total fraud rate" by a bank's management. The management is likely to underact if faced with small figures of "the total fraud rate". The mechanism standing behind is their willingness not to account random fluctuations and delayed perception. However, if "the total fraud rate" continues to increase, the management will not be able to neglect it and will start putting proportional pressure on their a security team. Closer to the limits of security team capacity, the increase in pressure will not result in the increase in performance. From the facts presented I can conclude that the figure will be S-shaped. Now, I can draw this function considering all of the facts presented, keeping in mind that it might be altered during a calibration phase in Chapter 4.

Resulting graph for table function is shown in Figure 33.

3.6 Decision making domain

My stock and flow model aims to present both the US and European cases. Although the structure of model is valid for both cases, some variables would have different behavior depending on selected strategy. The classical approach to such a problem is the introduction of a so-called decision variable and setting such variables as conditional using if-then-else structure. This modeling approach reflects a real situation of decision making process. Consider a variable "the burden of proof on customers or bank " in Figure 34. This is a decision variable, and it

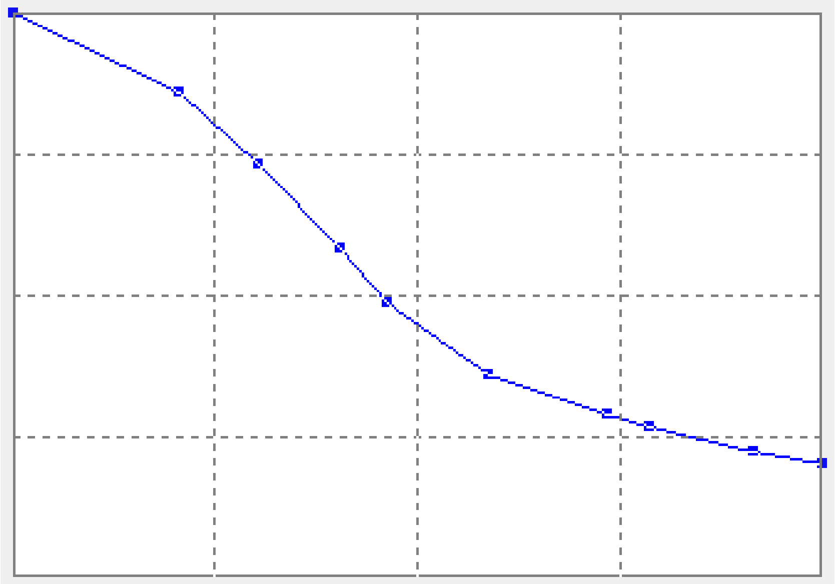


Figure 33: Effect of fraud rate on vulnerability fixing.

takes value 1 (in this case the burden of proof is on customers) or 0 (whereby the burden of proof is on the bank).

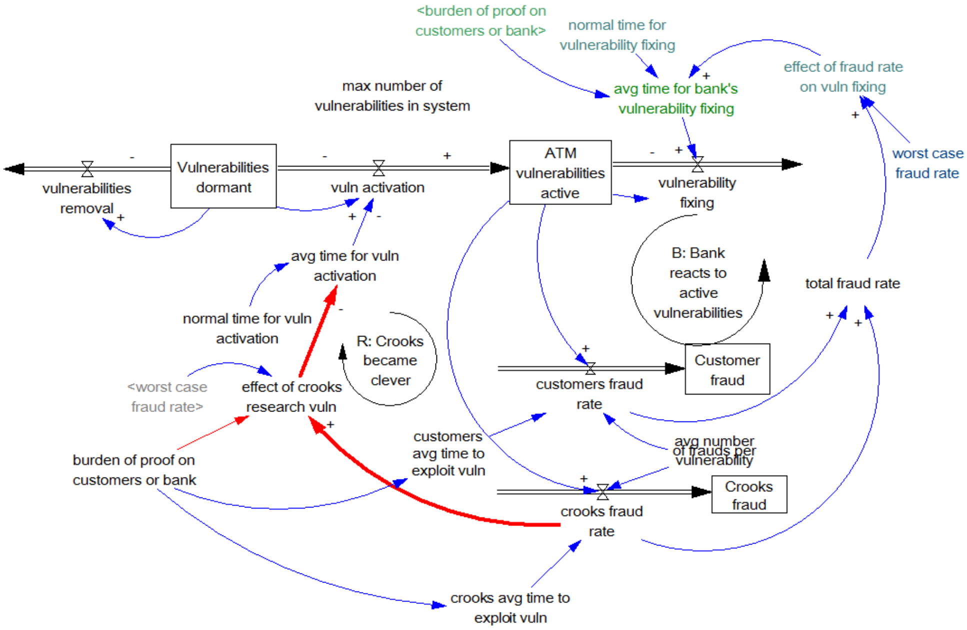

<burden of proof on customers or bank>
normal time for vulnerability fixing
effect of fraud rate on vuln fixing
max number of vulnerabilities in system
avg time for bank's vulnerability fixing
worst case fraud rate
Vulnerabilities dormant
ATM vulnerabilities active
vulnerabilities removal
vuln activation
vulnerability fixing
avg time for vuln activation
B: Bank reacts to active vulnerabilities
total fraud rate
normal time for vuln activation
R: Crooks became clever
Customer fraud
<worst case fraud rate>
effect of crooks research vuln
customers fraud rate
customers avg time to exploit vuln
avg number of frauds per vulnerability
burden of proof on customers or bank
Crooks fraud
crooks fraud rate
crooks avg time to exploit vuln
+
-

Figure 34: Effect of selected strategy on fraud.

3.7 Effect of strategy on fraud

According to the facts presented in [6], [10], the bank‘s strategy was affecting both ATM security and fraud domains. As presented in [38] the fraud occurring had different patterns in European as well as the US cases. Hence, there should be a clear interdependence between the value of "the burden of proof on customers or bank" and the fraud domain in my model.

Consider the sources of "fraud rate". As discussed above, they are "ATM vulnerabilities active". However, by "ATM vulnerabilities active" I mean that these vulnerabilities are known to at least one person. It takes time until they become known (i.e. by means of the word off a mouth) to the remaining pool of customers or crooks. To reflect this in our model shown in Figure 34 I introduce "avg time to exploit vulnerability", which means time that will pass until most people will discover a particular vulnerability. Once discovered, they will try to commit a fraud and according to survey in [39] will repeat it several times. To reflect the possibility of committing multiple frauds based on one vulnerability, I introduce "avg number of frauds per vulnerability".

According to [40] crooks, not only committing a fraud based on vulnerabilities, but also research dormant vulnerabilities (expressed by outflow "vuln activation") have been discovered. Vulnerabilities are becoming known as a matter of chance, but frequently there is a need to define initial conditions and/or implement some hardware devices, so that this vulnerability can be exploited. This activity requires time and, hence, I can introduce time variable "avg time for vuln activation" representing the time horizon over which most of the dormant vulnerabilities become active. Now let us consider the nature of this time variable. There should be some "normal time for vuln activation" which is not dependent on crooks’ intentions and reflects random nature of vulnerability discovery process. If I take crooks activity into account (expressed by "the effect of crooks research vuln"), it will decrease "avg time for vuln activation" from the level of its "normal time for vuln activation" to some value below. The only challenge remaining now is to formulate "the effect of crooks’ research vuln".

Firstly, if I look at patterns of frauds in the US and Europe, I can conclude that crooks’ activity of researching new vulnerabilities is interdependent on the strategy a bank implements: in the US it was higher than in Europe. Therefore, "the effect of crooks research vuln" is influenced by "the burden of proof on customers or a bank". Secondly, according to [41] crooks,the activity had a reinforcing pattern of behavior, meaning that the more crooks have succeeded in performing frauds today, the higher the incentive for other crooks to perform frauds tomorrow will be. As an individual crook makes decisions based not on the absolute fraud rate, but rather on the ratio of current fraud rate to the industry‘s known worst fraud rate, I assume that "the effect of crooks research vuln" is dependent on the quotient of "crooks fraud rate" divided by "the worst case fraud rate". The resulting reinforcing loop "R:Crooks became clever" is shown in Figure 34.

3.8 Bank‘s attitude to security

In the ATM security domain, shown in Figure 30, there is an outflow "vulnerabilities removal". By this variable I reflect the process of a bank‘s security team discovering and patching dormant vulnerabilities, so that crooks will not be able to activate them. Similarly to the definition of "vuln activation", I will formulate it as a quotient of "Vulnerabilities dormant" and "avg time to remove vuln". Essentially this time the parameter will reflect a bank‘s attitude to proactive security measures, or in other words, the implementation of the selected strategy.

In order to remove dormant vulnerabilities, they should be firstly discovered by security team and then patched. Some of vulnerabilities will become known due to random events and some - as a result of the aimed search. In order to account the fact of vulnerabilities being discovered unintentionally by a security team, I introduce "the normal time to remove vuln". "Avg time to remove vuln" can either equal to this normal time or be decreased by the effect of a bank‘s intentions for proactive approach.

In order to make a model structure explicit for an external reader I opted to have two feedback loops, one for the European case, in Figure 35 denoted as "R: Bank complacent " and one for the US case, denoted as "R: Bank acts proactively". As discussed above, the resulting effect of a bank‘s strategy is incorporated in variables "the effect of burden of proof on customers on ATM security" for European and "the effect of burden of proof on a bank on ATM security" for the US case.

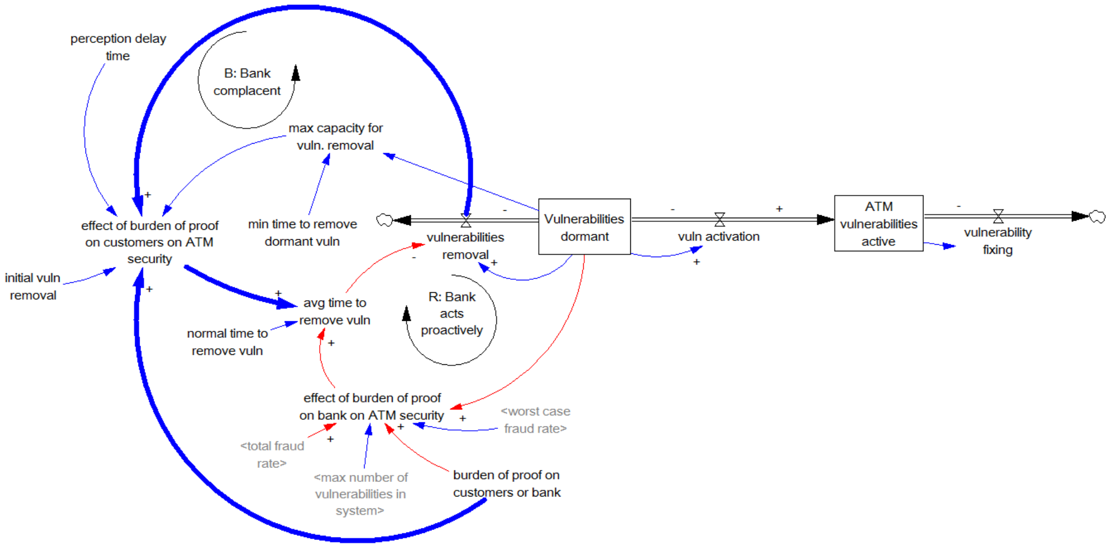

perception delay time
B: Bank complacent
max capacity for vuln. removal
effect of burden of proof on customers on ATM security
min time to remove dormant vuln
vulnerabilities removal
Vulnerabilities dormant
vuln activation
ATM vulnerabilities active
vulnerability fixing
initial vuln removal
avg time to remove vuln
R: Bank acts proactively
normal time to remove vuln
effect of burden of proof on bank on ATM security
<worst case fraud rate>
<total fraud rate>
<max number of vulnerabilities in system>
burden of proof on customers or bank
+
-

Figure 35: Effect of selected strategy on ATM security.

To begin with, I will explore the US case. According to [42] the US banks decided to implement a proactive approach to security after the regulation to refund customer claims if the guilt of a customer cannot be proven[43] was imposed. Now, consider the factors influencing "the effect of burden of proof on a bank on ATM security". Firstly, it is perceived as a fraud rate in a bank. Similarly to incentive towards vulnerability fixing, it is not viewed as an absolute value, but rather in comparison to the industry‘s known "the worst case fraud rate". Secondly, the more dormant vulnerabilities there are in ATM system, the easier it is to find new samples, and vice versa. Therefore, I also include the ratio "vulnerabilities dormant" / "max number of vuln in a system" as a factor of influence. As a result, I get a reinforcing feedback loop "Bank acts proactively".

I present complete model on Figure 36.

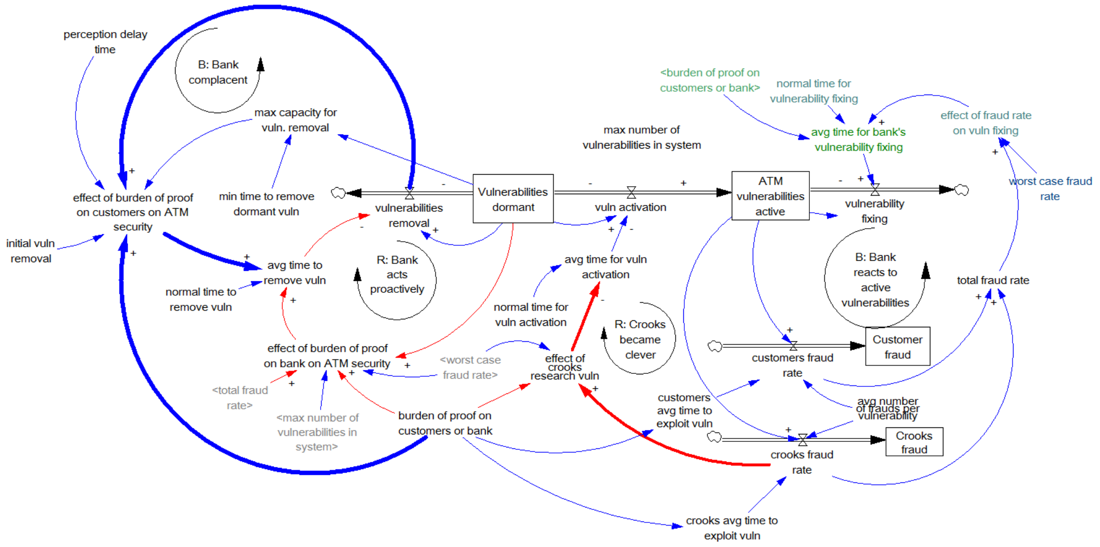

perception delay time
B: Bank complacent
max capacity for vuln. removal
effect of burden of proof on customers on ATM security
min time to remove dormant vuln
vulnerabilities removal
Vulnerabilities dormant
vuln activation
max number of vulnerabilities in system
<burden of proof on customers or bank>
normal time for vulnerability fixing
avg time for bank's vulnerability fixing
effect of fraud rate on vuln fixing
worst case fraud rate
ATM vulnerabilities active
vulnerability fixing
initial vuln removal
avg time to remove vuln
normal time to remove vuln
R: Bank acts proactively
avg time for vuln activation
normal time for vuln activation
R: Crooks became clever
B: Bank reacts to active vulnerabilities
total fraud rate
Customer fraud
customers fraud rate
effect of burden of proof on bank on ATM security
<worst case fraud rate>
effect of crooks research vuln
<total fraud rate>
<max number of vulnerabilities in system>
burden of proof on customers or bank
customers avg time to exploit vuln
avg number of frauds per vulnerability
Crooks fraud
crooks fraud rate
crooks avg time to exploit vuln
+
-

Figure 36: Complete model for ATMs in US and Europe.

4 Model Simulation and Analysis

In this chapter I will compute (section 4.1) and analyze (section 4.2) the output of my model. The developed stock and flow model corresponds to the real-life situation happened to the European and US banks, as described in Chapter 3. When I am talking about the model output, I mean the behavior of model variables over time. Owing to the fact that feedback-based models are quite robust, it is better to limit our analysis to several most interesting variables. This is a consistent approach because in stock and flow models variables are interdependent, so that the behavior of one variable is influenced by that of the others.

The analysis of a simulated model represents state-of-the-art procedures, carried out to ensure that the model does correspond to the target problem and the results are consistent. At the analysis stage I will use techniques suggested in [44], such as margin value boundary adequacy tests, extreme condition tests and sensitivity analysis tests.

4.1 Model simulation

As mentioned above, the challenge is to select proper variables that will represent the behavior of the whole model. Based on the classification in Chapter 3 I have a few different domains in my model. As stated above, the aim of my work is to provide evidence that putting liability on the third parties will not be beneficial in the long run. According to [12] the negative result over time for European banks was the epidemics of fraud. Therefore, it is essential to take the variable from a fraud domain and view its behavior through the simulation time. The most representative is "the total fraud rate".

As described in [29] and [32], the fraud was higher in the US than in Europe. According to the way I have built my model, before simulation I needed to determine the strategy of the bank first (e.g. American or European), and then based on this strategy the model provides output. In order to show that in my model the fraud in the European case is higher than that in the American one, I need to have a graph simultaneously displaying fraud rates in both cases. It is important to note that according to the way I have defined the fraud rate, it measures the number of frauds occurring per time unit, or from a different angle, it is the derivative of aggregated frauds. As it is impossible to do during one simulation, I conduct two simulations and save them as different "run"s. The result is displayed in Figure 37

Although the lines corresponding to the European and American cases have the same shape, the US fraud rate is lower than the European one throughout

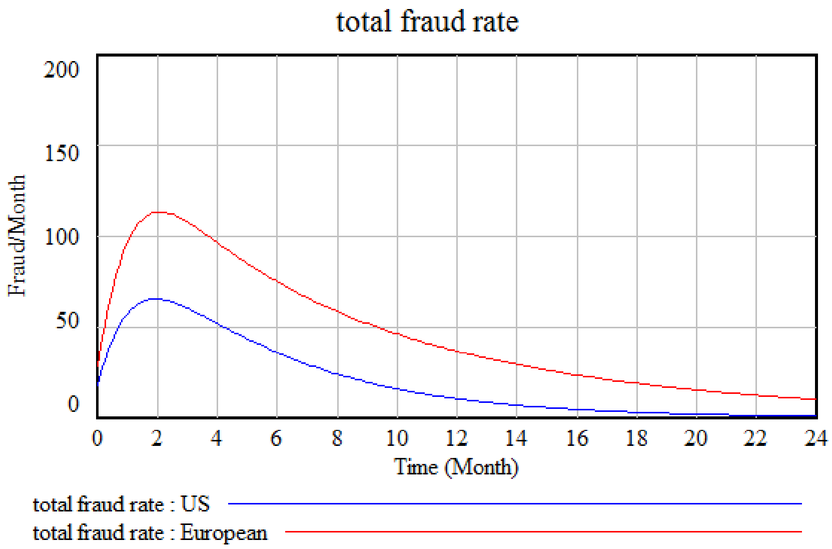


Figure 37: Total fraud rate.

the whole simulation time. This is a clear indication that our model corresponds to historical facts [40]. To explain this behavior we need one more time to review key assumptions of the model. Firstly, I assume that once ATMs were introduced, they had a certain number of dormant vulnerabilities, that could be activated over time, but no new dormant vulnerability was added as ATM security design remained unchanged. Secondly, frauds may occur only by utilizing active vulnerabilities. From these two assumptions I can draw the first conclusion: eventually, the total fraud rate will asymptotically approach *0* because vulnerabilities are removed and their total number is reduced respectively. However, it does not exclude possible fluctuations in the total fraud rate during simulation time. For simplifying the reasons I will proceed with the description of the European case because lines for both European and American banks have the same shape.

The prime cause of frauds "ATM vulnerabilities active" is provided in Figure 38. It is clearly seen that its shape is similar to that of "the total fraud rate".

Another fact mentioned in [40] is that in the US customer fraud was greater than that committed by crooks, while in Europe crooks fraud was greater than the customer one. In Figure 39 I show the European case by thick lines and the US one - by thin lines. It is important to bear in mind the difference between the fraud rate and fraud itself. The fraud rate is a flow variable representing the number of frauds committed in a current month. Contrary, the fraud is a stock representing the number of frauds committed during simulation time. This stock has no outflows (e.g. once a fraud has been committed, it cannot be reversed) therefore its value will steadily increase with time passing by. Therefore, all the four lines show the increase through all the simulation time and are asymptot-

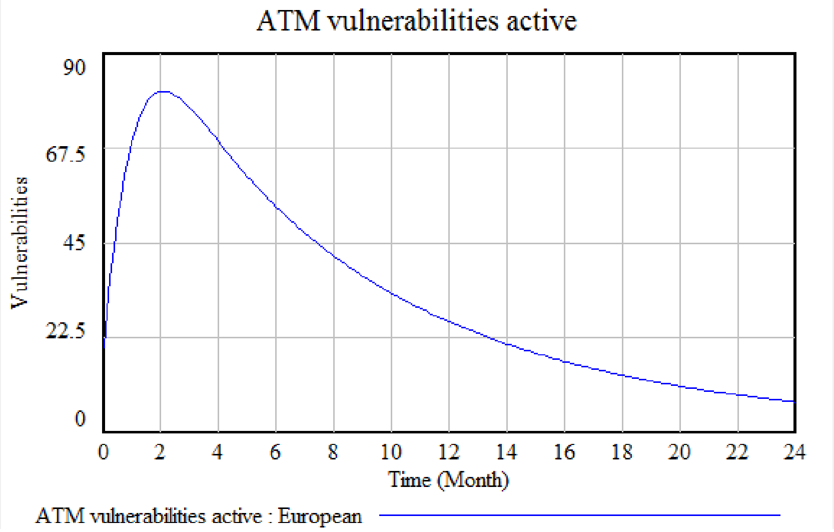


Figure 38: ATM vulnerabilities active.

ically approaching some level. The reason for asymptotic behavior is shown in Figure 38. When the number of active vulnerabilities is decreasing - fewer and fewer frauds per month will be committed. Hence, the latter would increase in a value of "Fraud"stock.

From a different angle, at the qualitative level it is clear that the sum of crook and customer frauds in the European case is greater than in the US case, which supports behavior displayed in Figure 37.

4.2 Model Analysis

Although the simulation results presented in the previous section seem to correspond to historical description, I would like to perform some tests that will show the validity of the model and give me confidence that even if different initial values are used, they will still remain valid and trustworthy. Owing to the problem formulation, I decided to perform selected tests: Structure assessment, Extreme conditions and Integration error.

4.2.1 Structure assessment

The structure assessment tests cannot be treated as conventional tests because it is hard to conduct them. For example, they may involve doing significant changes in the model. The acceptable way to do them is to define a set of questions and critically challenge answers to them. One of the most important questions that are applicable to my case is whether the model has an appropriate level of aggregation [45].

Now I will consider all the aggregation assumptions, used in my model. Firstly, I consider that there is no difference between frauds. Owing to the fact that the

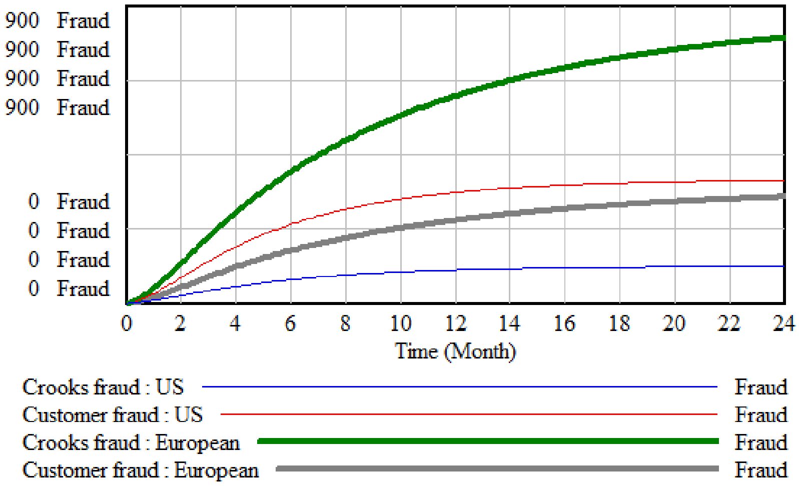


Figure 39: Customer fraud and crooks fraud.

aim of my project is education, I have built a model that is applicable to any average bank located in the US or Europe. Of course, if the task was formulated to model a situation in one particular bank, then it would be worth distributing frauds into subcategories, but then the model gets a narrower application. It is important to state that our audience is supposed to understand the structure of the model apart from playing with its output, and then it is important to avoid too much detail.

However, from [34] I have understood that it is important to disaggregate vulnerabilities into two classes. The incentive comes from the fact that the design of ATMs has remained almost unchanged between 1982 and 1993. Most vulnerabilities were there from the very beginning; however, crooks and customers explored them with a flow of time. Therefore, I decided to disaggregate them into undiscovered ("dormant") and discovered ("ATM vulnerabilities active") vulnerabilities.

4.2.2 Extreme conditions

Extreme conditions incorporate tests developed to check whether the model has correct behavior in case of an enormously big or small value of a certain parameter even if such extreme values of parameters are not expected to be achieved in a normal mode of operation. This can be a powerful tool to eliminate all the ad hoc relations between parameters. Sterman [5] suggests to check the formulation of outflows first, e.g. if there are any circumstances when the stock can become negative. The stock can be reduced only through outflow. Therefore, I need to test stocks "Vulnerabilities dormant" and "ATM vulnerabilities active" for non-negativity. Since such stocks as "Customer fraud" and "Crooks fraud" have

Test	Input condition	expected result	actual result
1	"avg time for vuln activation"=TIME STEP and "avg time to remove vuln" =TIME STEP	"Vulnerabilities dormant">=0	"Vulnerabilities dormant">=0
2	"avg time for bank's vulnerability fixing"=TIME STEP	"ATM vulnerabilities active">=0	"ATM vulnerabilities active">=0
3	"max number of vulnerabilities in system"=0	"total fraud rate"=0	"total fraud rate"=0
4	"avg number of frauds per vulnerability"=0	"total fraud rate"=0	"total fraud rate"=0

Table 1: Extreme condition testing

no outflows, they cannot become negative. In the Table 1 I first two tests stand to verify the right formulation of outflows.

Secondly, it is suggested to assess the relation between variables representing resource instances and variable standing for the product made by means of resources. The rule is that if there are no resources available, no product can be produced. In our model, vulnerabilities are the reasons for fraud. Therefore, no fraud can be done without vulnerabilities. Test 3 in Table 1 corresponds to this requirement by setting the "max number of vulnerabilities in system" equal to 0. Another parameter influencing the fraud rate is "avg number of frauds per vulnerability"). In Test 4 from Table 1 I simulate the situation when there is no vulnerability that can lead to fraud. As expected, in this case the total fraud rate equals to 0.

4.2.3 Integration error

System dynamics models are developed to describe a process that goes continuously over a period of time. For models involve stocks and flows, naturally there are integration and differentiation processes standing behind. Contrary to real life, the integration into system dynamics models cannot be done directly, but rather by means of numerical methods. Because these methods require setting a value of time step, it is important to ensure that the model behaves properly. For example, processes that are meant to happen as discrete events are not represented as a continuous process.

A good example of an improper choice of time step is given in [5], when discussing the model for life cycle of Tsembaga tribe in New Guinea. According to historical data, there were sharp transitions from peace to war, while the model generated a continuously high mortality rate, not depending on war or piece time. To discover this error, modelers are recommended to check the model

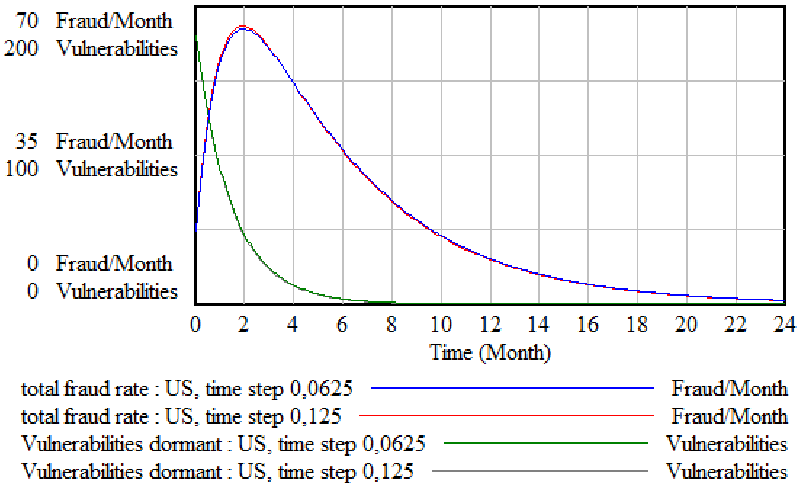


Figure 40: US simulation under different time steps.

behavior, when a time step is set to half of its initial value. If the behavior has not changed significantly, than a time step is correct. Otherwise, it should be reduced.

In my model, a time step is set to 0,125 per month. To perform the test I will take a time step of 0,0625 and observe the behavior for stock variables from the domains of vulnerabilities and frauds. In the vulnerability domain I need to observe the behavior of both "Vulnerabilities dormant" and "ATM vulnerabilities active". In the fraud domain it is enough to observe the behavior of the total fraud rate provided that it represents both customer and crook fraud. I have conducted a test for both European and US cases and tests have been passed successfully. One of the examples is illustrated in Figure 40. There I do simulation for the US case and observe differences in the behavior of "Vulnerabilities dormant" and "total fraud rate". As it can be observed, these variables are not sensitive to the chosen time step, so the test has been passed successfully. Therefore, I conclude that the time step of 0,125 per month is appropriate for my model.

5 Online version of model

This project was devoted to the simulation of historical cases of ATM frauds in the US and Europe. The knowledge derived form this case is about learning why badly aligned information security incentives possess a great treat to the party responsible for making decisions. In the preceding chapters I have dealt with the desktop model developed in Vensim environment. However, to achieve the availability and convenience in its usage, I have decided to develop a separate internet-based training application in Forio environment, based on my desktop model. The free version of Forio is a software environment, where a developer uploads a Vensim model, creates a user interface to help end users specify input parameters, run the model as well as view the output in the most convenient manner. The primary goal of Forio is to run desktop models (i.e. models written in languages such as Vensim, Python, Julia, R, SimLang)online, so that anyone interested can access it. Because of the primary goal this environment has no model development capabilities: it simply accepts the finalized desktop models. For those who are interested in running simulations Forio is a good facilitation tool since it does not require any knowledge of Vensim environment and the graphical interface of the model can be built in the most user friendly way. The other benefit of Forio is that all its functions to transfer data between Vensim models and a user interface have already been made available. A simple user interface can be created within minutes by using a built-in graphical editor. For more elaborate tasks, one can insert custom HTML and JS codes.

5.1 How to create simulations in Forio

Below, I will give an example of running a simple model in Forio environment. I will take a part of our ATM model, representing the process of active vulnerabilities removal and step by step show the model export to Forio, user interface development and model simulation.

5.1.1 Preparing Vensim model for export

The respective stock and flow model is shown In Figure 41. This model has the same equations as the original model of ATM fraud, but because it is only a part of complete model, some changes were implemented to delete dependencies on out of system boundary variables. I had to make "*avg time for a bank's vulnerability fixing*" as a constant variable. Now I will create a simple simulation that will allow to view the dynamics of *ATM vulnerabilities active* in an output graph. I will also let the user set initial values for "*avg time for a bank's vulnerability fixing*" and *the max number of vulnerabilities in a system*.

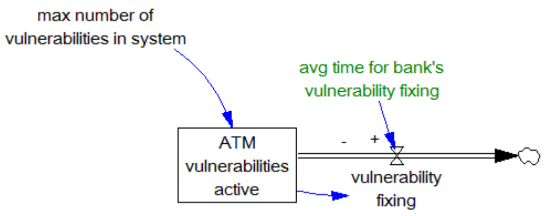

max number of vulnerabilities in system
avg time for bank's vulnerability fixing
ATM vulnerabilities active
-
+
vulnerability fixing

Figure 41: Simplified structure for ATM vulnerabilities active.

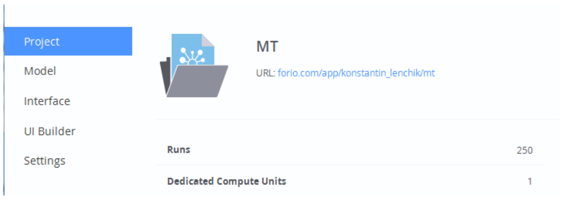
The Vensim files normally can be directly exported to Forio, but the equations for variables that are meant to be specified by the user should be altered, so that they contain *GAME()* a function. This function allows the user to interact with model during simulation time. Compared to a non-game mode of operation, such model does not run to the final simulation time, but completes a specified number of time steps and then waits for the user input. The user provides input for a game variable and the model completes the next set of time steps. Such an interactive approach is extremely beneficial when the user is supposed to learn from observing an outcome of different strategies, not just running a model with one hard-coded strategy. [46]

The last,yet important step is to save the model as a Binary Format Model file (*.vmf), as Vensim does not support the import of Text Format Models (*.mdl)

5.1.2 Creating a new project and importing Vensim model

First of all, Forio-user needs to create a new project and specify the language of the model (i.e. Vensim, Python, Julia, R, SimLang). Forio gives options to create an empty project or modify one of the example projects. After choosing to create an empty project, the dialogue window also asks to give a name to the project and desired ID that will be a part of the project‘s URL.

After this the user is forwarded to project the control panel, shown in Figure42 Tabs on the left stand for:



MT

URL: forio.com/app/konstantin_lenchik/mt

Runs	250
Dedicated Compute Units	1

Figure 42: Forio control panel.

- **Project** General information about your project and some statistics (number of runs)
- **Model** Here the user should upload his/her desktop model

- **Interafce** By selecting this tab, the user is directed to the project folder on Forio‘s File Server. Here one can add auxiliary files, for example, additional graphics, and edit existing files. In case of complicated projects, capabilities of UI Builder are insufficient, therefore, developers need to add custom CSS, HTML or Flow.js.
- **UI Builder** A graphical tool to build a user interface and set up communication between the client and server side. According to Forio architecture, the model is run on the server side. I will discuss the possibilities of UI Builder in the next subsection.
- **Settings** This tab contains access and computation setting together with API keys for accessing application-level data

5.1.3 Developing simple interface in UI Builder

Once the model is imported into Forio, I go to UI Builder to create a simple interface. The UI Builder is shown in Figure43. The left part of the window contains default components, that can be added to the canvas of a future user interface (on the right). All the components are grouped into 3 categories:

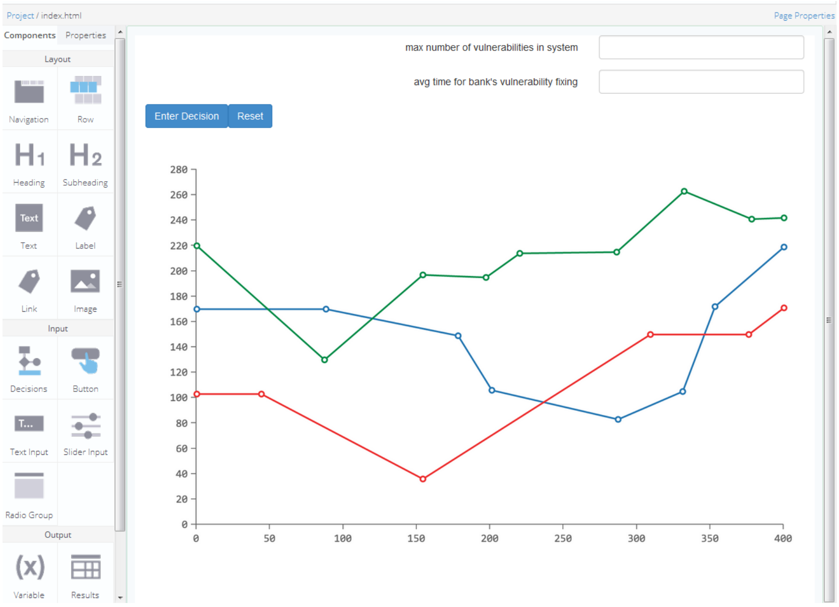


Figure 43: Forio UI Builder.

- **Layout** Components such as Labels, Texts or Images add more graphics or information.
- **Input** Components to obtain user input (i.e. text input) or control flow components. The only control flow component is Button component with *onclick()* property, that can be programmed to execute certain command

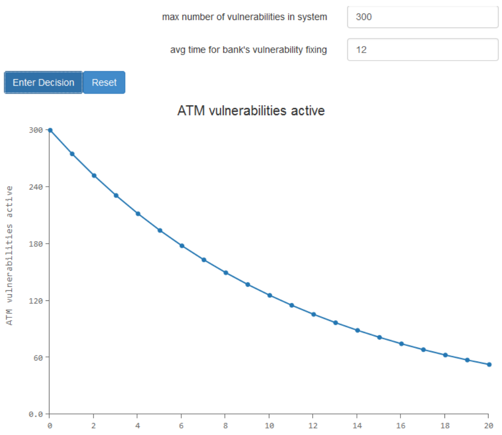
(i.e. run model or reset model).

- **Output** Different components to display the output of simulation. [47]

All the components can be placed on canvas by a usual drag and drop operation. For this simple model selection of input components is straight forward, whereas the choice of output components is not that evident. In my model to observe a result, I need to see the behavior of variable *ATM vulnerabilities active*. As an output variable, Forio allows the following options: a single value, table of values, line graph and bar chart. Owing to the fact that I would like to see the dynamics of variable behavior over time, I have opted for a line graph.

5.1.4 Running a model

Once a user interface is created, the model can be accessed through direct URL. The result of online simulation is shown in Figure 44. The user interface corresponds to the one built in UI Builder. I built it in a way that the interaction with the model does not require any knowledge of system dynamics or programming skills.



max number of vulnerabilities in system 300

avg time for bank's vulnerability fixing 12

Figure 44: Running simple model in Forio.

5.2 Creating online version of ATM fraud model

Despite the simplicity of the procedure, described in the previous section, a free version of Forio has limited developmental capabilities. Hence, all auxiliary elements in a user interface should be coded in HTML, CSS and Flow.js. For example, in my case a big challenge was to allow a user to run the model two times during one session, while storing the results of the previous run. This means the possibility to compare the behaviors of the US and European cases in one graph.

When the end user wants to run the training application, he is not required to have any knowledge of system dynamics or install additional software. The application has a user-friendly design and can be accessed directly via URL [48].

The web page is divided into three sections. In the first section I give a short historical background and explain how to interact with the simulation. The second section provides simulation itself, and in the third section one can observe our stock and flow model.

The web page has a simple design, but according to the best practices, it was built on Bootstrap technology. Therefore, the web page has a responsive design and can be viewed flawlessly on tablets and mobile devices. The graph building technique is also mobile-friendly.

I show the results of model simulation in Figure 45. The blue line represents behavior in the European case and a pink line stands for the US case. As we can see, the lines corresponding to the European and US cases have the same shape. However, the US fraud rate is lower than the European one throughout the whole simulation time. This is a clear indication that my model corresponds to historical facts [1]. In our model "the total fraud rate" is a cause of "ATM vulnerabilities active". That is why the shape is a result of change in the number of active vulnerabilities: firstly, there were only a few; later their quantity increased, but because no new vulnerabilities were added, finally most of vulnerabilities were removed, so the fraud rate asymptotically approaches 0. A more detailed analysis of behavior was provided in Section 4.1. After simulation, users are also encouraged to

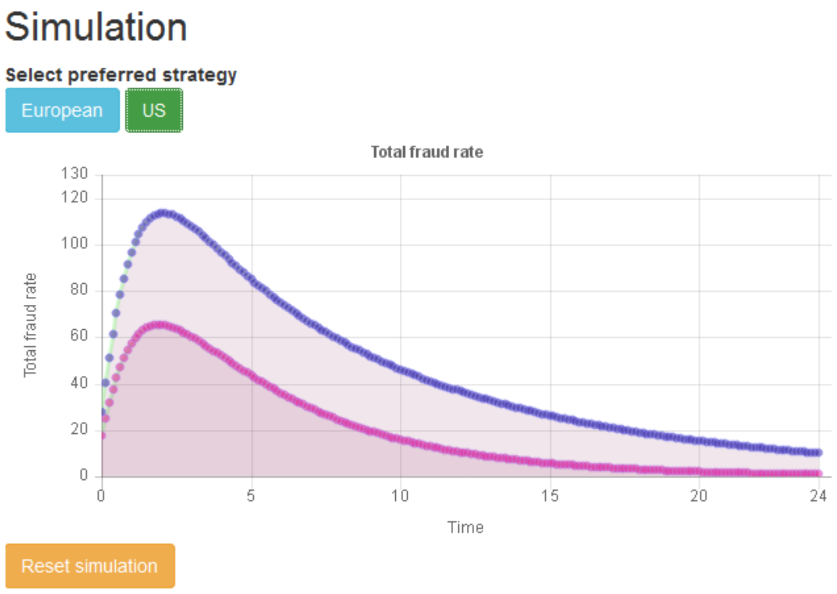


Figure 45: Running ATM model in Forio.

explore the stock and flow model in order to understand its underlying structure and relations between components.

6 Discussion

This chapter discusses the validity of results achieved in my Master's Thesis. By "validity" I mean correlation between the research carried out and actual results obtained. In the following section, I will discuss the validity and limitations of my research. Then, I proceed with giving outline for prospective future work.

6.1 Validity of results

As stated above, it is important to assess how valid the results are before they are presented. I will discuss both internal and external validity of my results according to [49]. Internal validity is a characteristic feature of the study, showing how credible its results actually are; in other words, whether there are any systematic errors in the study that have actually altered the study‘s output. Applied to my problem, the assessment of internal validity should give the answer whether there are any systematic errors in the way my model is designed. On the contrary, external validity refers to the extent to which results of the study can be successfully applied for solving similar problems, e.g. how well results of the study can be extrapolated.

Internal validity

Internal validity was partially verified in Chapter 4. In Chapter 4, devoted to the analysis of the model, I ran special system dynamics tests, suggested in [35]. The purpose of tests was to discover all ad-hoc relations in our model. Owing to the time constrains and model domain, I selected 3 tests that fit best. However, the model passed all the tests successfully, which instilled more confidence in it. My model possesses a high internal validity, because field-specific factors have been taken into account.

External validity

My model is based on a particular problem and, therefore, at first glance has a rather narrow scope of application. All the online simulations are developed by practitioners who have education as the primary aim; however, some of them are based on a specific problem that requires a lot of background knowledge of the domain for example General Motors simulation of relations between new and used car markets in the US [36] , while others represent a general problem, for example, relationships among the Manufacturer, Distributor, Wholesaler and Retailer in a so-called Beer Game [50]. The simulation of the first type targets people who are involved in solving this particular problem, while second type is more about the education of people who might in future encounter problems of a similar origin.

Even though my model clearly belongs to Type One models that presumably have a low external validity, I applied some efforts to make it also Type Two model. Firstly, taking note that the case of interest is 20 years old, I have chosen an aggregation level that does not include many particular details, for example, precise techniques of fraud implementation. Secondly, I opted for an understandable paradigm of frauds and vulnerabilities. Thirdly, I accounted the nature of people's decision making, also known as people's perception factor. All this was carried out with respect to maintain internal validity. Hence, my audience will be able to investigate hidden relations in our model which, despite being field-specific, can be extrapolated to other domains and possess a descent teaching effect.

6.2 Future work

As discussed above, this project is a completed model, supported by the necessary tutorial and documentation. Therefore, there is little room for future development. However, the topic of online simulations developed for information security economics is considered quite prospective by the author. One of possible development vectors in this field is to create new simulations accounting counter intuitive behavior in complex systems. In simulation collections like [35] and [36] there are a lot of examples from the fields of health, economics and justice, but almost no cases from information security domain.

Another prospective direction is to take a case representing middle sized business with a relatively simple inner structure and investigate the possibilities of an agile approach to investments in information security. The agile approach in this case is about an intentionally neglecting proactive approach to a certain extent in order to maintain high Return On Security Investments index. When it comes to small and middle sized organizations that are relatively young and are still growing, it turns out that ensuring mature information security is too costly for them. Thus, an interesting question comes to the fore: whether a descent trade of between the lack of proactive security measures and acceptable probability of success can be eventually found.

7 Conclusion

The aim of this thesis was to provide evidence for the following assumption: when the subject (person or company) possesses the infrastructure with an insufficient information security level and instead of investing into better security solutions, shifts its liability to the third parties (i.e. their customers or contractors), he will in the end be affected by the unintended outcome of such a decision. I started with a state-of-the-art research in Chapter 2 and identified 9 different cases that can give evidence supporting my assumption. All the cases were illustrated by qualitative diagrams, so that the reader can better understand the reasons for such a behavior. After this, I selected a case of ATMs introduction in the US and Europe over a period of 1982 and 1993 and completed all the stages of building a model for it step by step:

- the review of literature to capture the essence of a problem
- the development of a respective system archetype.
- search for quantitative data
- the development of a quantitative model (stock and flow model)
- model simulation
- model testing and analysis
- the online version of the model

The resulting product of my work is a model, consisting of a desktop version with technical documentation and online version including user tutorial. The main objective of this work is education of audience. I expect our audience to consist of people responsible for information security investment and students of Information Security tracks. However, on a larger scale, it might also be interesting for anyone involved in policy design or studying hidden relationships in complex systems.

Online simulations have been used for educational purposes for the last decade. Normally, simulation models are presented on the Internet-based resources, containing a tutorial and direct link to run the simulation. Among the most popular resources are Forio (a leading solution to creating online simulations) [51] and MIT Sloan (MIT-based school of management) [52] databases containing online simulations. The author gained a good insight on best practices for simulation development, while going through these resources. However, the problem of boomerang effects in information security policy making had never been addressed there before.

Now assume that the European banks have chosen to invest a few thousand euros in system dynamics modeling: order development of such archetypes and a conceptual model so as to understand the impact of different strategic choices. Instead of blindly going for costly strategy, they would have firstly analyzed its unintended consequences. Author assumes that there is little doubt that the European banks would have declined their initial strategy and opted for a better security solution. In several application areas it has been shown that investing in simulation models for strategy analysis cost in the order of 0.5% of the cost of failures done by bad decisions.

To sum up, I would like to add that all the goals of the research have been accomplished. Below I outline main contributions of thesis:

- The model can be explored on different levels of detalization. Naturally, readers are expected to start with online simulation, trying to discover underlying casual relations practically and, afterwards, examine model structure, to proof or discard their previous assumptions.
- For university students the developed model is a mature case-study material for learning how to take counter intuitive decisions in information security.
- For people involved in security decision making my thesis gives good evidence that badly aligned incentives can be as a drastic treat to IS as bad security design solutions.
- Simulation approach described in this thesis can be viewed as a scientific tool to approach problems originated from behavior of complex systems. Not only the final result is beneficial, but the whole process can provide a significant advance in knowledge. Both modelers and field experts learn a lot about hidden relations between system components, while creating a qualitative and subsequent quantitative models.

Bibliography

[1] Moore T., Clayton R. and Anderson R. 2009, The economics of online crime. *The Journal of Economic Perspectives*, volume 23. American Economic Association.

[2] Anderson R. and Moore T. 2006, The economics of information security. *Science*, volume 314. American Association for the Advancement of Science.

[3] Moore, T. 2010, The economics of cybersecurity: Principles and policy options. International Journal of Critical Infrastructure Protection.

[4] Anderson R. and Moore T. 2006, Why information security is hard. Science.

[5] Sterman J. 2000, Business dynamics: systems thinking and modeling for a complex world. volume 19. Irwin/McGraw-Hill Boston.

[6] Wright, Marie A. 1991, Security controls in atm systems. *Computer Fraud & Security Bulletin*. Elsevier.

[7] Essinger J. 1987, Atm networks-their organisation, security and future. Elsevier.

[8] Winn, Jane K. and Wright B. 2000, The law of electronic commerce. Aspen.

[9] Bauer J., Eeten M. 2009, Cybersecurity: Stakeholder incentives, externalities, and policy options. Telecommunications Policy 33.

[10] McConville M. 1994, Standing accused: The organisation and practices of criminal defence lawyers in britain. Oxford University Press.

[11] Lewis, B. 1992, How to rob a bank the cashcard way. *Sunday Telegraph*.

[12] Krebs, Brian. 2008, 'money mules' help haul cyber criminals' loot. *Washington Post*.

[13] Brantley R. 1987, Atm crimes pose a growing threat. *Saving Institutions*, volume 108.

[14] Anderson R. and Bezuidenhout J. 1995, Cryptographic credit control in pre-payment metering systems. IEEE.

[15] Anderson R. and Bezuidenhout J. 1996, On the reliability of electronic payment systems. *Software Engineering, IEEE Transactions on*, volume 22. IEEE.

[16] Anderson, R. 1993, Why cryptosystems fail. Proceedings of the First ACM Conference on Computer and Communications Security.

[17] Anderson R. 1994, Liability and computer security: Nine principles, In *Computer Security—ESORICS 94*. Springer.

[18] Akerlof, George. 1995, The market for "lemons": Quality uncertainty and the market mechanism. Springer.

[19] Fusion. Sony pictures hack was a long time coming (online). URL: "http://fusion.net/story/31469/sony-pictures-hack-was-a-long-time-coming-say-former-employees/" (Visited May. 2016).

[20] Wikipedia. Sony pictures entertainment (online). URL: "https://en.wikipedia.org/wiki/SonyPicturesEntertainmenthack" (Visited May. 2016).

[21] A "kill chain" analysis of the 2013 target data breach. Committee on commerce, science, and transportation.

[22] Markowsky G. and Markowsky L. 2014, From air conditioner to data breach, In *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

[23] Hirsch D. The glass house effect: Big data, the new oil, and the power of analogy'(2014). *Maine Law Review*, volume 66.

[24] OPM. Handing over the keys to the castle (online). URL: http://icitech.org/wp-content/uploads/2015/07/ICIT-Brief-OPM-Breach2.pdf (Visited May. 2016).

[25] Kolstad C., Ulen T. and Johnson G. 1990, Ex post liability for harm vs. ex ante safety regulation: substitutes or complements. *The American Economic Review*. JSTOR.

[26] Davies D. 1986, Special report on atm fraud down under. *Computer Law & Security Review*, volume 1. Elsevier.

[27] McConnell, S. 1992, Barclays defends its cash machines. *The Times*, volume 7.

[28] Birch D. 1996, Over the water–the view from the uk. *Journal of Internet Banking and Commerce*, volume 1.

[29] Anderson, Ross J. 1996, Crypto in europe—markets, law and policy, In *Cryptography: Policy and Algorithms*. Springer.

[30] Levi M. 1988, The prevention of fraud. volume 17. Citeseer.

[31] Sharma, Sunil. 1993, Behind the diffusion curve: an analysis of atm adoption. University of California Working Paper.

[32] Zagaris B. and MacDonald S. 1992, Money laundering, financial fraud, and technology: The perils of an instantaneous economy. *Geo. Wash. J. Int'l L. & Econ.*, volume 26. HeinOnline.

[33] Levi, Michael. 1991, The prevention of cheque and credit card fraud. Citeseer.

[34] 1995, Nations world ministerial conference on organized transnational crime, naples, november 21-23, 1994. *Trends in Organized Crime*. Springer.

[35] Doebelin, Ernest. 1998, System dynamics: Modeling, analysis, simulation, design. CRC Press.

[36] Barabba V. and Huber C. 2002, A multimethod approach for creating new business models: The general motors onstar project. *Interfaces*, volume 32, INFORMS.

[37] Forrester, Jay W. 1995, The beginning of system dynamics. *McKinsey Quarterly*, MCKINSEY & CO INC.

[38] Anderson, Ross. Papers on smartcard engineering, In *PROC. OF FIRST SMART CARD RESEARCH AND ADVANCED APPLICATION CONFERENCE*. Citeseer.

[39] Berger A. and Kashyap A. 1995, The transformation of the us banking industry: What a long, strange trip it's been. *Brookings papers on economic activity*, volume 1995. JSTOR.

[40] Wasik, Martin. 1988, Documents on fraud issues. *Computer Law & Security Review*, volume 4. Elsevier.

[41] Schreiber, Francis B. 1994, Atm crime & security newsletter. volume 4.

[42] Anderson, Ross J. 1999, The formal verification of a payment system, In *Industrial-Strength Formal Methods in Practice*. Springer.

[43] Lane, D. 1992, Where cash is king. *Banking Technology*.

[44] Senge P. and Forrester J. 1980, Tests for building confidence in system dynamics models. *System dynamics*, volume 14.

[45] Forrester, Jay Wright. 1997, Industrial dynamics. *Journal of the Operational Research Society*, volume 48, Palgrave Macmillan.

[46] Vensim. Games in vensim (online). URL: https://www.vensim.com/documentation/index.html?20780.htm (Visited May. 2016).

[47] Forio. Creating your user interface (online). URL: https://forio.com/epicenter/docs/public/creating_your_interface/ (Visited May. 2016).

[48] Lenchik K. Internet based model of atm fraud (online). URL: http://forio.com/app/konstantinlenchik/mt (Visited May. 2016).

[49] Wohlin C. and Henningsson K. 2003, Empirical research methods in software engineering, In *Empirical methods and studies in software engineering*. Springer.

[50] Beer game supply chain simulation (online), Harvard Business School Publishing. URL: http://forio.com/simulate/harvard/the-root-beer-game-supply-chain-simulation/overview/ (Visited May. 2016).

[51] Learning edge system dynamics simulations (online), MIT Sloan. URL: https://mitsloan.mit.edu/LearningEdge/simulations/Pages/System-Dynamics.aspx (Visited May. 2016).

[52] Forio. Forio simulate (online). URL: http://forio.com/simulate/showcase (Visited May. 2016).

A Model equations

In this appendix I present a complete list of equations for my system dynamics model. The equation list was created using built in function in Vensim environment "Document All". Each of the model variable is presented here with the respective equation. Some of them are also followed by comments (highlighted in green color)

1. *ATM vulnerabilities active*= INTEG (*vuln activation-vulnerability fixing*, *max number of vulnerabilities in system**0.1) Units: Vulnerabilities NUmber of vulnerabilities that can be exploited to commit fraud. Range for initial value [5..40]

2. *avg number of frauds per vulnerability*= 2 Units: Fraud/Vulnerabilities

3. *avg time for bank's vulnerability fixing*= IF THEN ELSE(*burden of proof on customers or bank*=1, 10, 5)*(*normal time for vulnerability fixing* **effect of fraud rate on vuln fixing*) Units: Month Other things equal if the burden of proof is on the customers the average time for vulnerability fixing is 5 times long than if the burden of proof was on the bank

4. *avg time for vuln activation*= *normal time for vuln activation***effect of crooks research vuln* Units: Month AVG time to activate vulnerability

5. *avg time to remove vuln*= *effect of burden of proof on bank on ATM security***effect of burden of proof on customers on ATM security* **normal time to remove vuln* Units: Month AVG time that takes to discover and patch vulnerability

6. *burden of proof on customers or bank*= 1 Units: Dmnl This variable can assume two possible values: 1 (the bank puts the burden of proof on fraud claims on the customer) or 0 (the bank has the burden of proof that the claim by the customer is fraudulent)

7. *crooks avg time to exploit vuln*= IF THEN ELSE(*burden of proof on customers or bank*=1, 2 , 10) Units: Month Average time for crooks to exploit the active ATM vulnerabilities depending on whether the burden of proof is on the customers (1) or on the bank

8. *Crooks fraud*= INTEG (*crooks fraud rate*, 0) Units: Fraud Cumulated number of frauds, due to NON bank‘s customers (crooks) cheating Initial value

is 0, as we are not interested in history

9. *crooks fraud rate*= *ATM vulnerabilities active*/*crooks avg time to exploit vuln** *avg number of frauds per vulnerability* Units: Fraud/Month

10. *Customer fraud*= INTEG (*customers fraud rate*, 0) Units: Fraud Cumulated number of frauds, due to bank's customers cheating Initial value is 0, as we are not interested in history

11. *customers avg time to exploit vuln*= IF THEN ELSE(*burden of proof on customers or bank*=1, 5, 3) Units: Month Average time for customers to exploit the active ATM vulnerabilities depending on whether the burden of proof is on the customers (1) or on the bank

12. *customers fraud rate*= *ATM vulnerabilities active***avg number of frauds per vulnerability*/*customers avg time to exploit vuln* Units: Fraud/Month

13. *effect of burden of proof on bank on ATM security* = WITH LOOKUP ((1-*burden of proof on customers or bank*)* (*total fraud rate*/*worst case fraud rate*)* (*Vulnerabilities dormant*/*max number of vulnerabilities in system*),([(0,0)-(1,1)],(0,1),(0.168196,0.71),(0.360856,0.58),(0.443425, 0.28),(0.498471,0.18),(0.541284,0.14),(0.602446,0.12),(0.721713,0.11),(1,0.1))) Units: Dmnl S shape behaviour influenced by next factors: if burden is on bank, bank is eager to research vuln., else (burden on customer) bank removes dormant vulnerabilities that were discovered accidentally the higher total fraud rate, the more bank is investing in research of vulnerabilities the more dormant vulnerabilities we have now, the easier it is to find next one

14. *effect of burden of proof on customers on ATM security* = WITH LOOKUP (DELAY1I ((*burden of proof on customers or bank*)*(*vulnerabilities removal*/ "*max capacity for vuln. removal*"),*perception delay time* ,*initial vuln removal*), ([(0,0)-(1,1)],(0,0.1), (0.1957,0.11),(0.345566,0.12), (0.470948,0.14) , (0.498471 ,0.2),(0.593272, 0.5), (0.764526,0.625),(1,1))) Units: Dmnl "mirrored" S shape behaviour influenced by factors: burden on bank makes this effect =1, meaning, that it will have no "avg time to remove vuln" vulnerabilities removal/"max capacity for vuln. removal" decreases effect meaning that the more bank removes vulnerabilities the more it is aware of problem, and the less complacent it becomes.

15. *effect of crooks research vuln* = WITH LOOKUP ((1-*burden of proof on customers or bank*)**crooks fraud rate*/*worst case fraud rate* , ([(0,0)-(1,1)],(0,1),(0.2,0.83),(0.3,0.7),(0.4,0.6),(0.5,0.4),(0.6,0.3), (0.7,0.22), (0.7,0.22), (0.8,0.21),(1,0.2))) Units: Dmnl Range of this function is [0,2..1]. The minimum value is 0.2, not 0, because even if burden is on customer or

crooks fraud rate is low, there will be some crooks who research vulnerabilities as a hobby. The behaviour is close to classical S shape, meaning that outer conditions doesn‘t stimulate a lot of crooks activity. But as "(1-burden of proof on customers or bank)*crooks fraud rate/worst case fraud rate" increases, crooks start to research vulnerabilities faster and faster, but at some point comes close to capacity limit for vulnerability research, so even worse fraud rate doesn’t result in proportional growth in speed for vulnerabilities research.

16. *effect of fraud rate on vuln fixing* =WITH LOOKUP (*total fraud rate/worst case fraud rate*, ([(0,0)-(1,1)],(0,1),(0.207951,0.859649),(0.30581,0.7324 56),(0.40367,0.583333),(0.461774,0.486842),(0.590214,0.359649),(0.73 9345,0.285088),(0.785933,0.263158),(0.917431,0.223684), (1,0.2))) Units: Dmnl Range of this function is [0,2..1]. The minimum value is 0.2, not 0, because bank has limited capacity for vulnerability fixing, in other words bank is unable to fix all vulnerabilities instantly. The behaviour is close to classical S shape, meaning that when "total fraud rate"/"worst case fraud rate" is relatively low, bank nearly ignores it. But as "total fraud rate"/"worst case fraud rate" increases, bank understands severity of situation and starts to fix vulnerabilities faster and faster, but at some point comes close to capacity limit for vulnerability fixing, so even worse fraud rate doesn’t result in proportional growth in speed for vulnerabilities fixing.

17. FINAL TIME = 24 Units: Month The final time for the simulation.

18. INITIAL TIME = 0 Units: Month The initial time for the simulation.

19. *initial vuln removal*= INITIAL(0.2) Units: Vulnerabilities/Month

20. "*max capacity for vuln. removal*"= *Vulnerabilities dormant/min time to remove dormant vuln* Units: Vulnerabilities/Month Max possible capacity to remove dormant vulnerabilities

21. *max number of vulnerabilities in system*= 200 Units: Vulnerabilities

22. *min time to remove dormant vuln*= 0.5 Units: Month Min time that bank s security team can spend to discover and patch vulnerability

23. *normal time for vuln activation*= 2.5 Units: Month Min possible time for avg vulnerability to become activated

24. *normal time for vulnerability fixing*= 1 Units: Month A bank concerned with good practice and reputation under low time pressure (few frauds) would fix known active vulnerabilities in such normal time

25. *normal time to remove vuln*= 50 Units: Month

26. *perception delay time*= 2 Units: Month Delay in Banks perception for effect of burden of proof on customers on ATM security

27. SAVEPER = TIME STEP Units: Month The frequency with which output is stored.

28. TIME STEP = 0.125 Units: Month The time step for the simulation.

29. *total fraud rate*= *crooks fraud rate*+*customers fraud rate* Units: Fraud/Month Cumulated number of frauds, due to bank‘s customers and NON customers cheating in a month

30. *vuln activation*= *Vulnerabilities dormant*/*avg time for vuln activation* Units: Vulnerabilities/Month Rate of discovery of dormant vulnerabilities. Once they are discovered they are "active" in the sense that they can be exploited to commit fraud.

31. *Vulnerabilities dormant*= INTEG (-*vuln activation*-*vulnerabilities removal*, *max number of vulnerabilities in system**0.9) Units: Vulnerabilities Undiscovered vulnerabilities in ATM system. Range for initial value[100..400]

32. *vulnerabilities removal*= *Vulnerabilities dormant*/*avg time to remove vuln* Units: Vulnerabilities/Month Process of discovering and removing dormant vulnerabilities

33. *vulnerability fixing*= *ATM vulnerabilities active*/*avg time for bank's vulnerability fixing* Units: Vulnerabilities/Month

34. *worst case fraud rate*= 200 Units: Fraud/Month