IncidentResponseSim: An Agent-Based Simulation Tool for Risk Management of Online Fraud

Dan Gorton(✉)

Center for Safety Research, Department of Transport Science,
KTH Royal Institute of Technology, Stockholm, Sweden
dan.gorton@abe.kth.se
http://www.kth.se

Abstract. IncidentResponseSim is a multi-agent-based simulation tool supporting risk management of online financial services, by performing a risk assessment of the quality of current countermeasures, in the light of the current and emerging threat environment. In this article, we present a set of simulations using incident response trees in combination with a quantitative model for estimating the direct economic consequences. The simulations generate expected fraud, and conditional fraud value at risk, given a specific fraud scenario. Additionally, we present how different trojan strategies result in different conditional fraud value at risk, given the underlying distribution of wealth in the online channel, and different levels of daily transaction limits. Furthermore, we show how these measures can be used together with return on security investment calculations to support decisions about future security investments.

Keywords: Risk management · Online fraud · Incident Response Tree (IRT) · Value at Risk (VaR) · Simulation · Return on Security Investment (ROSI)

1 Introduction

Banking is one part of our critical infrastructure [1], and as such, a threat against online banking may become a threat against the society.

Over the years, cyber criminals have become better organized and attacks against online banking services have grown more sophisticated [2]. A recent threat report from the European Network and Information Security Agency (ENISA) shows increasing trends for most of the attack vectors needed for online fraud [3].

To counteract this situation, authorities like the Federal Financial Institutions Examination Council (FFIEC) in the US and the European Central Bank (ECB) in Europe are stepping up their expected minimum security requirements for financial institutions, including requirements for risk management of online banking [2][4].

S. Buchegger and M. Dam (Eds.): NordSec 2015, LNCS 9417, pp. 172–187, 2015.
DOI: 10.1007/978-3-319-26502-5_12

For the financial institution, these requirements translate into a need to understand the incident response process protecting online services. However, existing tools like attack and protection trees [5][6][7][8][9], fail to capture chronological ordering of events [10].

In previous articles, we have presented the fundamentals of a derivation of event tree analysis for online fraud, which we call incident response tree (IRT) [11], and added a quantitative model inspired by current models for estimating credit risk [12].

In this article, we introduce IncidentResponseSim, an online bank simulation tool modeling the consequences of online fraud directed against online financial services.

The rest of this article is organized as follows. Section 2 presents the proposed model. In section 3, we simulate different scenarios using IncidentResponseSim. Section 4 presents an analysis of the consequences of different trojan strategies. In section 5, we present how the results from IncidentResponseSim can be used together with return on security investment (ROSI). Section 6 presents a discussion, and section 7 wraps up the article with conclusions.

2 Background

2.1 Online Banking

Wikipedia defines online banking as "... an electronic payment system that enables customers of a financial institution to conduct financial transactions on a website operated by the institution, such as a retail bank, virtual bank, credit union or building society" [13].

Financial institutions often provide multiple solutions for online banking, called "channels", with differing security and usability. Most often, a more secure channel provides more services, more information, and allows higher transaction amounts.

2.2 The Incident Response Process

The incident response process of online financial services is very important in that it protects customer and company assets. As noted, ENISA finds that the threat landscape is getting worse, making effective risk management of incident response a priority for financial institutions and governing bodies alike [3].

However, to investigate, develop, test, and improve different parts of the incident response process, detailed information about the current threat landscape and the effectiveness of current countermeasures is needed. Thus, information sharing is needed not only within the financial institution, but also in the form of information sharing among financial organizations. Additionally, it is very important to be able to model how emerging threat landscapes will affect current countermeasures. Information from, for example, simulations of potential future scenarios will make it possible to better plan for what might happen next.

For example, we can imagine a financial institution preparing for emerging threats which it has learned about from previous victims. The possible direct economic consequences are then estimated using simulation and further precautions are initiated, if deemed necessary. Other relevant situations to model include:

- Soon to be entered markets (with a different threat landscape)
- The introduction of a high usability (but less secure) online service
- Single point of failure (concerning for example prevention, detection, or response).

Furthermore, there is a lack of research in the domain of incident response. One reason for this is that information about fraud, and the effectiveness of current countermeasures is sensitive, something that is shared only sparsely within the financial institution, and preferably not at all with external parties.

Threats and Countermeasures. According to Julisch [14], there are three types of threats against online banking:

- Impersonation
- Deception
- Server-side attack.

Impersonation, which is the main focus of this article, corresponds to fraud where the fraudster impersonates the real user, using, for example, phishing, man-in-the-middle, man-in-the-browser, or social engineering, resulting in the user giving up his or her credentials. Deception corresponds to fraud where the fraudster tricks the user into register transactions on behalf of the fraudster, for example, using various fraud schemes like Nigeria Letters. Server-side attack corresponds to fraud where the fraudster hacks into the online banking environment and issues transactions from these servers directly. A recent example is the attacks by Carbanak (also known as Anunak) [15].

From an online financial service perspective, there are different ways to mitigate the effects of fraud; however, finding the right balance between different parts of countermeasures is not an easy task [16]. The likelihood of attacks can be mitigated by more effective preventive measures like multi-factor authentication, and more effective fraud detection and response. Additionally, the direct economic consequences of fraud can be lowered by temporarily closing down services like foreign payments, or lowering daily transaction limits.

To mitigate this situation, we developed a tool, based on event tree analysis which we called an incident response tree (IRT) [11].

Incident Response Trees. By using IRTs we are able to measure the effectiveness of different parts of countermeasures, for example, prevention, detection, and response.

Figure 1 shows an IRT for the initial event being a customer with a banking trojan which is actively targeting an online banking channel.

To measure the effectiveness of prevention, detection, and response, the incident response team needs to collect four different types of consequences, C_1 to C_4, to populate a basic IRT (most probably, some or all of these statistics are already collected separately by the fraud response team):

- C_1. The number of active attacks which were not detected by the bank. (This includes fraud detected by the customer after the fact, or by the financial institution during back testing against known money mules.)
- C_2. The number of active attacks which were detected by fraud detection, but not stopped by fraud response. (This includes fraud detected by batch fraud detection, where responsive countermeasures are not quick enough.)
- C_3. The number of active attacks which were detected by fraud detection, and stopped by fraud response.
- C_4. The number of active attacks which were identified and stopped by preventive measures, e.g., authentication and intrusion detection.

Using C_1 to C_4 frequencies, it is possible to create an IRT specific for the incident response process of the financial institution. Of course, with more detailed information documented during the incident response process, it should be possible to create more complex IRTs, for example, distinguishing between online and batch fraud detection, and automatic and manual response measures.

Furthermore, by using the frequencies, it is possible to calculate the conditional probabilities of prevention, detection, and response, given an active trojan attack. The generated statistics are then used as an indication of how effective the current countermeasures are against the specific threat, i.e., the higher the conditional probabilities for prevention, detection, and response, the less successful is the trojan.

However, to be able to estimate the direct economic consequences of an active trojan attack, the IRT tool needs to be complemented with an additional model.

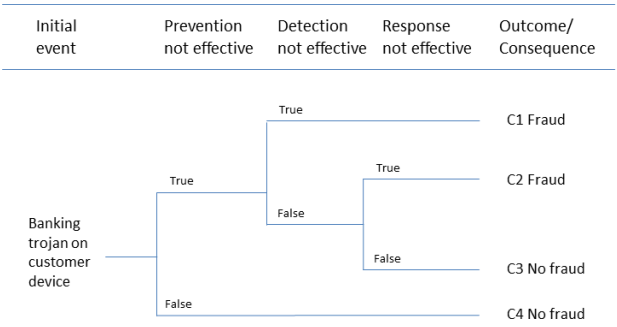

Fig. 1. Basic incident response tree [11].

Estimating the Direct Economic Consequences. When analyzing the possible direct economic consequences of fraud it is clear that the account balance of the victims and the transaction limit of the chosen online banking channel are important parameters; they both limit the size of the fraud. In [12], we notice that this situation is similar to current modeling of credit risk [17]. The model used is thus inspired by current models for credit risk, reusing some concepts from the advanced internal ratings-based approach for calculating credit risk [17].

By transferring these concepts to the domain of online financial fraud we end up with the following three concepts [12]:

- Exposure at Fraud (EAF_i) of customer i. Calculated as the minimum of the transaction limit (TL) and customer i:s account balance
- Probability of Fraud for a time period of one year (PF). Calculated from historical statistics or expert knowledge, as the number of fraud cases divided by the number of channel customers
- Loss Given Fraud (LGF_i) of customer i. Calculated for each customer, or based on historical statistics or expert knowledge, as the fraction of EAF stolen from defrauded customers.

In analogy with EL calculated for estimating the current credit risk [17], Expected Fraud (EF) is then calculated as:

$$EF = PF \cdot \sum_{i=1}^{N} (EAF_i \cdot LGF_i) \tag{1}$$

where N represents the number of channel customers, EAF_i represents the current exposure at fraud for each possible victim customer i, and PF and LGF_i represents values derived from historic events.

Thus, EF is conditioned on previous attacks targeting the financial institution, given:

- Number of defrauded customers
- Number of channel customers
- Account balance of each victim customer
- Transaction limit used for the specific channel
- Strategy used by the trojan.

However, with so many conditions, it will be hard for a fraud prevention manager to estimate how representative the calculated EF value is. One way to measure the accuracy of EF is to recalculate EF using a different set of victim customers, ceteris paribus. In [12], we present a conditional fraud value at risk (VaR) measure, which is defined as the level of loss that, for a specific scenario, will not be exceeded with a given level of confidence (e.g., 95%).

Conditional fraud VaR is calculated using simple random sampling over K different sets of victim customer accounts, where the number of customers in these sets is taken to be the expected annual number of defrauded customers I (which can be based on historical data known to the financial institution) [12].

Each sample will result in a sample-specific fraud loss (FL_k), calculated as:

$$FL_k = \sum_{i=1}^{I} (EAF_i \cdot LGF_i) \tag{2}$$

Additionally, K iterations of simple random sampling will generate a distribution of FL:s. EF is calculated as the mean of this distribution, and conditional fraud VaR is calculated by choosing the 95th percentile.

Furthermore, to be able to plan ahead for potential risks of an emerging threat environment, the fraud prevention manager will need to simulate future adverse scenarios, resulting in scenario-specific conditional fraud VaR.

To remedy this situation, we continue our research by creating a simulation tool for risk management of online fraud, which we call IncidentResponseSim. Additionally, we show how the results from these simulations can be used together with return on security investment calculations to support decisions about future investments.

3 Model

Our initial aim with IncidentResponseSim is to be able to simulate the effects of current and emerging threat landscapes directed against online financial services. The simulation environment is built on the concept of multi-agent-based simulation (MABS), which is a "class of computational models for simulating the actions and interactions of autonomous agents with a view to assessing their effects on the system as a whole" [18].

IncidentResponseSim is built using the Mason simulation environment [19]. This platform has previously been used in fraud detection research [20][21]. In contrast to earlier fraud detection research using the Mason simulation environment [20][21], IncidentResponseSim takes a broader perspective as it does not focus on any specific countermeasure technology, for example detection, but rather on the process of incident response as a whole, including the EF, and consequences of fraud using conditional fraud VaR.

The basic principle of IncidentResponseSim is the concept of fraudulent transactions. The banking trojan's objective is to attack the customers and transfer money out of the victim accounts. The concept of the channel plays a special role in the simulation. It serves as the scheduler for the next step of the simulation. Given the specific step of the simulation, the channel generates a supply of customers and malware.

In our simulation environment, the interaction between agents is always between malware and customer. The trojan randomly chooses a customer to attack. The agents do not perform any specific learning activities. Their behavior is given by probabilistic Markov models where the probabilities can be extracted from real incident response statistics, i.e., statistics calculated from an IRT.

Currently, there are three agents in IncidentResponseSim: (Online) Channel, Customer, and Trojan (Figure 2):

- Channel. This agent is responsible for scheduling customers and threats for the next step of the simulation.
- Customer. This agent is a fictional customer at the bank. The balance of the customer is drawn from a Beta distribution ($\alpha = 2, \beta = 7$) multiplied by 100,000. The assumption is that most customer balances are skewed towards lower amounts. Ideally, the bank estimates the actual distribution using the same underlying data that is required for reporting credit risk. Each customer can be in three different states: unaffected (non-victim), victim, and defrauded.
- Trojan. This agent is the only threat modeled in this first version of IncidentResponseSim. The Trojan can be in different states: active/non-active, and greedy/non-greedy (a greedy trojan is set to only attack wealthier customers with a current account balance above a set "greedy-limit").

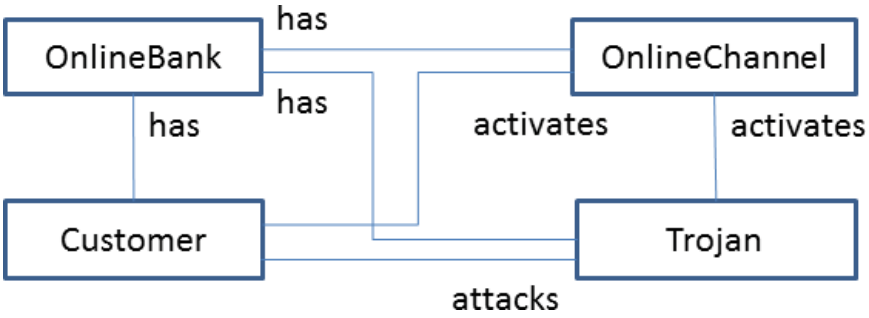

Fig. 2. Simplified model of IncidentResponseSim.

A normal step of the simulation starts out by identifying customers in the victim state. The customer is then "cured" with a probability P. In the initial version of IncidentResponseSim, this probability is set to 1.0, i.e., all previous victim customers are "cured" in the first part of every step.

In the second part of the step, the trojan agent is activated. The trojan randomly attacks a number of customers in the current online channel. Each victim customer is assigned a consequence, C_1 to C_4 (see Section 2.2), and all customers with a consequence of C_1 and C_2 are then defrauded according to the chosen trojan strategy; Max, Random, and Mean Transaction (Section 5).

3.1 Support for Different Types of Simulations

To be a valuable tool for a fraud prevention manager, IncidentResponseSim needs to support different types of simulations. We can imagine a fraud prevention manager that wants to estimate:

- The distribution of the number of victims, given scenario-specific IRT statistics

– The distribution of the direct economic consequences, given a scenario-specific number of victims and trojan strategy.

To accomplish this, IncidentReponseSim uses two main types of simulation; Simple Random Sampling of Defrauded Customers, and Simple Random Sampling of Direct Economic Consequences. Additionally, IncidentResponseSim supports a third simulation type, called Multi-step Simulation of Direct Economic Consequences, which first calculates the number of defrauded customers from IRT statistics, directly followed by a calculation of the direct economic consequences. They are all detailed below.

Simple Random Sampling of Defrauded Customers. During this simulation, the same step is performed N times by repeatedly calculating the number of defrauded customers, $C_1 + C_2$, given the conditional probabilities of prevention, detection, and response. Each customer is assigned as a victim with a probability of P_{IR}, according to the infection rate. Each victim customer is then further classified into the different consequences C_1 to C_4 according to the consequence-specific probabilities. The victim customer is first assigned to consequence C_4 with a probability P_P, according to the conditional probability of prevention. All victim customers that were not categorized into C_4, are then tested for C_1 with a probability $(1 - P_D)$, where P_D is the conditional probability of detection. Finally, the remaining customers are then categorized into C_3 with a probability P_R, the probability of effective response, and into C_2 with a probability $(1 - P_R)$. The number of defrauded customers per step is $y_n = C_1 + C_2$, resulting in a mean of:

$$\overline{y} = \frac{1}{N} \cdot \sum_{n=1}^{N} y_n \tag{3}$$

In this article, we are conservative, aiming for an upper bound, and choose to set the number of defrauded customers as the 95 percentile of the distribution of y_n.

Simple Random Sampling of Direct Economic Consequences. In this first version of IncidentResponseSim, the simulation is started with a fixed number, M, of defrauded customers. During this simulation, the same step is performed K times by simple random sampling of M victim customers. Every victim is then defrauded according to the chosen trojan strategy (e.g., 90%), for example, $z_i = min(30000, x_i \cdot 0.9)$, where 30,000 represents the TL, and x_i represents the account balance of the customer. EF and conditional fraud VaR are then calculated according to the algorithm presented in Section 2.2.

Multi-step Simulation with Direct Economic Consequences. During this simulation, the behavior of the simulation is primed by relevant scenario statistics, for example, infection rate, and the conditional probabilities of prevention, detection, and response (given an actively targeting trojan). Additionally, the

parameters can be changed from step to step to simulate a given fictive scenario. The actual logic is comparable to repeatedly performing one step of Simple Random Sampling of Defrauded Customers followed by one step of Simple Random Sampling of Direct Economic Consequences.

3.2 Data

Ideally, the data needed to support IncidentResponseSim is collected both internally from the fraud incident response process, and from external sources.

To be able to collect internal data, the financial institution probably needs to adjust its current incident response process in such a way that useful data is documented by the fraud response team, for example:

- The number of customers in the channel
- The frequencies of IRT consequences, C_1 to C_4
- Fraud-specific information:
  - The transaction limit of the channel
  - The account balance
  - The amount stolen
- The distribution of wealth in the channel (in line with existing credit risk calculations).

The financial institution also needs external data concerning the current and emerging threat landscape, as well as information about the quality of different types of countermeasures. The former is typically available from different types of security organizations, and the latter is data either acquired by information sharing, for example, between financial institutions, or by direct experience.

4 Simulating Relevant Scenarios Using IncidentResponseSim

In this section, we first present a baseline scenario describing our estimated current conditions, followed by scenarios where we evaluate how the current countermeasures handle different kinds of potential future stress, including:

- Newly entered markets (with different threat landscapes)
- Single point of failure (of different parts of the countermeasures)
- Emerging threat landscapes.

When simulation results are unacceptable, countermeasures, transaction limits, etc. can be adjusted to yield acceptable results (this should be tested using further simulation).

By performing "what-if" analysis using simulations like these, the fraud prevention manager can estimate, for example, scenario-specific conditional fraud VaR, which can be used together with models like return on security investment (ROSI) to support decisions about future security investments.

4.1 Baseline Scenario

Our fictional online channel has 100,000 customers. The maximum account balance of the online channel is set to 100,000 SEK, with a daily transaction limit of 30,000 SEK. According to our estimated (fictional) statistics, the trojan infection rate is 0.01, and the conditional probabilities of prevention, detection, and response is set to 0.8, 0.9, and 0.9 respectively. As this is our baseline scenario, the infection factor is set to 1.0. All simple random sampling simulations are set to 999 iterations. If further accuracy is needed, the number of iterations needs to be set to a higher value. All simulations use the random number seed 822075070.

The data used in the simulations are summarized below:

- Number of Customers = 100,000
- Max Balance = 100,000 SEK
- Transaction Limit = 30,000 SEK
- Number of Iterations = 999
- Infection Rate, $P_{IR} = 0.01$
- Infection Factor = 1.0
- Conditional Probability of Prevention, $P_P = 0.8$
- Conditional Probability of Detection, $P_D = 0.9$
- Conditional Probability of Response, $P_R = 0.9$
- Trojan Strategy = Max (i.e., the minimum of TL and account balance).

Throughout all simulations, we will be on the conservative side aiming for the 95 percentile results. The "Simple Random Sampling of Defrauded Customers" simulation generates a mean value of 38.10, a standard deviation of 6.07, and a 95% percentile of 48 (for $C_1 + C_2$).

We then insert the calculated number of defrauded customers at the 95 percentile, i.e., 48, into the "Simple Random Sampling of Direct Economic Consequences" simulation.

During the 999 iterations, EF was 941,426 SEK, with an FL standard deviation of 62,548 SEK. In 95% of all iterations, the stolen amount did not exceed 1,042,431 SEK, which is our conditional fraud VaR.

4.2 Newly Entered Markets

In this simulation, we can imagine a fraud prevention manager trying to model what might happen when the financial institution enters a new market using an online service which is very similar to one where they have access to baseline statistics. The main difference is the new threat landscape. We assume that the financial institution has a 2.75 times higher probability of native customers being infected with malware on the new market, using public malware infection statistics as a proxy [23]. In IncidentResponseSim we change the following information (compared to the baseline scenario):

- Infection Factor = 2.75

The "Simple Random Sampling of Defrauded Customers" simulation generates a mean value of 104.58, a standard deviation of 10.19, and a 95% percentile of 121 (for $C_1 + C_2$).

During the 999 iterations, EF was 2,379,053 SEK, with an FL standard deviation of 97,137 SEK. In 95% of all iterations, the stolen amount did not exceed 2,545,100 SEK, which is our conditional fraud VaR.

4.3 Single Point of Failure

In this simulation, we can imagine a fraud prevention manager trying to analyze what might happen when one of the main components of the countermeasures fails open, i.e., when the prevention system fails open, the detection system fails open, or the response system fails open. Information about the probable consequences of single points of failure will be helpful when estimating the fault tolerance of the incident response process.

Firstly, we set Probability of Prevention to 0. Simple Random Sampling of Number of Defrauded Customers results in 213 defrauded customers at 95 percentile.

Secondly, we set Probability of Detection to 0. Simple Random Sampling of Number of Defrauded Customers now results in 225 defrauded customers at 95 percentile.

Thirdly, we set Probability of Response to 0, which results in 225 defrauded customers at 95 percentile.

We then calculate the direct economic consequences given that 225 customers are defrauded (i.e., we chose to use the highest number). This results in an EF of 4,422,002 SEK, and an FL standard deviation of 135,992 SEK. In 95% of all iterations, the stolen amount did not exceed 4,636,140 SEK, which is our conditional fraud VaR.

4.4 Emerging Threat Landscapes

In this simulation, we model what might happen when the fraud prevention team proactively identifies an emerging threat that is highly contagious, and also very effective at overcoming current preventive measures. In IncidentResponseSim we need to change the following information (compared to the baseline scenario):

- Infection Rate, $P_{IR} = 0.02$
- Conditional Probability of Prevention, $P_P = 0.6$

The Simple Random Sampling of Number of Defrauded Customers results in 171 defrauded customers at the 95 percentile, an EF of 3,352,588 SEK, with an FL standard deviation of 114,013 SEK. In 95% of all iterations the defrauded money did not exceed 3,545,783 SEK, which is our conditional fraud VaR.

5 Trojan Strategies Versus Transaction Limits

In this simulation, we model the direct economic effects of various banking trojan strategies using "Simple Random Sampling of Direct Economic Consequences". The number of defrauded customers is set to 24 in all simulations, changing only the trojan strategy and the daily transaction limit (Figure 3).

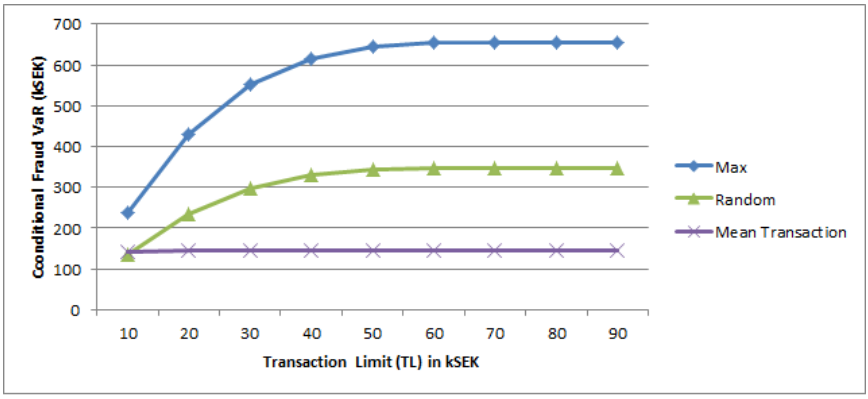

Fig. 3. Conditional fraud VaR (at the 95th percentile) for different trojan strategies (specified in the text), given a specific online channel.

5.1 Max

In this simulation, we simulate a trojan agent going for the account balance, up to the daily transaction limit. In Figure 3, we can see that the curve rises sharply and then levels off. The reason for this is the distribution of wealth among the channel customers (Section 3), with only a few customers with account balances "in the right tail". This implies that a transaction limit above 50,000 SEK makes little sense for this scenario. Thus, the effect of the transaction limit is dependent on both the distribution of wealth in the channel, and the expected fraud scenario.

From a defender perspective, this strategy would be easy to detect using fraud detection.

5.2 Random

In this simulation, we change the odds a bit and simulate a trojan agent going for a random uniformly distributed amount between 0 and the minimum of the account balance and the daily transaction limit. As expected, conditional fraud VaR is much lower than for the previous strategy, although the general form of the curve is the same.

From a defender perspective, the random strategy would be a bit harder to detect using behavior-based fraud detection. However, some fraudulent transactions will be high, and may attract attention from fraud detection and also be easily spotted by customers.

5.3 Mean Transaction

Lastly, we simulate a trojan agent going for the average size of previous customer transactions. Every customer has his/her "mean transaction" simulated by setting it to 500 SEK, plus a random figure between 0 and 10,000 SEK. Using this strategy, the maximum fraudulent transaction generated would be 10,500 SEK.

From a defender perspective, the mean transaction strategy would be hard to detect using behavior-based fraud detection by only analyzing the defrauded amounts; however, the beneficiary account would be new.

6 Estimating Return on Security Investment Using Simulation Results

6.1 Return on Security Investments (ROSI)

In this section, we will evaluate different security investments using a framework that is used within the information security domain called Return on Security Investment (ROSI) [22]. In this framework, $ROSI$ is defined as:

$$ROSI = \frac{MLR - COS}{COS} \tag{4}$$

where MLR is defined as monetary loss reduction, and COS is defined as cost of solution. These values are discounted, when needed, and $ROSI$ should be positive for the investment to be profitable.

To calculate the monetary loss reduction, an Annual Loss Expectancy (ALE) is first calculated as the product of the Annual Rate of Occurrence (ARO) and the Single Loss Expectancy (SLE).

$$ALE = ARO \cdot SLE \tag{5}$$

Then, either a modified ALE ($mALE$) is calculated, or a mitigation ratio is estimated (including the potential benefits of the implemented solution, i.e., the countermeasures). Thus, we have:

$$MLR = ALE - mALE \tag{6}$$

or

$$MLR = ALE \cdot MR \tag{7}$$

where MR is defined as the mitigation ratio.

6.2 Simplified Example

We can imagine a chief information security officer (CISO) deciding if, and how, to mitigate a probable threat (Section 4.1). The CISO has four different mitigating actions in mind: adding 10% prevention, 5% detection, or 5% response, or doing nothing.

In this simplified example, we can see that by adding preventive measures we get the highest $ROSI$, i.e., 0.15 (Table 1).

Table 1. ROSI analysis using IRTs and conditional fraud VaR calculated by IncidentResponseSim (at the 95th percentile).

Action	COS	#Frauds	Cost	MLR	$ROSI$
Do nothing	0	48	1,042,431	0	N/A
Add prevention (+0.1)	400,000	26	581,281	461,150	0.15
Add detection (+0.05)	300,000	38	826,431	215,999	-0.28
Add response (+0.05)	200,000	38	826,431	215,999	0.08

7 Discussion

Admittedly, the basic IRT implemented in this simulation is simple. However, it is possible to elaborate on the IRT so that it better represents the actual risk situation and the countermeasures applied, assuming that necessary data is available. Ways to do this include, for example, separating response into automatic and manual response, or allowing for more than binary outcomes. In [11], we argued that the simplicity of the basic IRT suits the work effort needed during an active attack using banking trojans. In the same article, we argued that event tree analysis, which IRTs are making use of, is made possible by low under-reporting, i.e., IRTs need "good enough" frequency statistics to accurately document the quality of the different parts of countermeasures. For different reasons, there will be under-reporting, but ways to minimize the problem exist, like reimbursing customers who report fraud, or if governing bodies legislate that retail and corporate customers must report when being defrauded. Furthermore, it is possible to back-test for non-reported transfers to known mule accounts. This can be used to both adjust frequency counts, and to estimate the quality of the current reporting practices by calculating the ratio of reported versus non-reported fraud.

The method for calculating conditional fraud VaR is partly inspired by credit risk methodology. We use simple random sampling to create the distribution of FL.

In this article, we have presented several different use cases, or scenarios, which are relevant to a fraud prevention manager either during the design of new online channels, or during risk management of existing ones. The presented

scenarios are still quite simple, but the underlying simulation platform makes it possible to include more complex behavior with regard to the agents when needed.

Additionally, we have presented a simple example using simulation results together with $ROSI$ calculations. We have opted for using conditional fraud VaR in our calculations, to be on the conservative side. One reason for this is that it is hard to measure the indirect costs of fraud.

8 Conclusions

IncidentResponseSim makes it possible to simulate the effects of active attacks using impersonation, like banking trojans. The simulation platform can, for example, be used during the design phase of online banking services, during active attacks, and during stress testing. Initially, IncidentResponseSim will be most valuable for financial institutions which are able collect the necessary statistics from their own incident response process.

In this article, we presented how to generate a set of plausible scenarios using IncidentResponseSim, including estimating risks pertaining to newly entered markets, single points of failure, and emerging threat landscapes. Additionally, we presented how to evaluate security investments using our proposed conditional fraud VaR model together with $ROSI$, to be able to support decisions about future security investments.

We argue that IncidentResponseSim can be a valuable tool for risk management of online financial services. However, further investigation and experimentation using real data are needed, and the results need to be validated by subject matter experts.

Future work includes adding functionality to IncidentResponseSim, like social network analysis for the analysis of the effects of different ways to warn customers about ongoing attacks, and potentially adding expert knowledge using Bayes, or more dynamic models like game theory. It would also be interesting to directly integrate more detailed models than ROSI.

Acknowledgments. I would like to thank Lars-Göran Mattsson, Per Näsman, and Torbjörn Thedéen at KTH Royal Institute of Technology, Stockholm, Sweden, and Stefan Axelsson at Blekinge Institute of Technology, BTH, Sweden, for feedback while writing this article.

References

1. ENISA: Methodologies for the identification of critical infrastructure assets (2015)
2. ECB: Recommendations for the Security of Internet Payments (2013)
3. ENISA: ENISA Threat Landscape (2014)
4. FFIEC: Supplement to Authentication in an Internet Banking Environment (2011)
5. Schneier, B.: Secrets & Lies: Digital Security in a Networked World, pp. 318–333. John Wiley & Sons, New York (2000)

6. Mauw, S., Oostdijk, M.: Foundations of attack trees. In: Won, D.H., Kim, S. (eds.) ICISC 2005. LNCS, vol. 3935, pp. 186–198. Springer, Heidelberg (2006)
7. Edge, K.S., Dalton II, G.C., Raines, R.A., Mills, R.F.: Using attack and protection trees to analyze threats and defenses to homeland security. In: MILCOM. IEEE (2006)
8. Edge, K.S., Raines, R.A., Grimaila, M., Baldwin, R., Bennington, R., Reuter, C.: The use of protection trees to analyze security for an online banking system. In: The Proceedings of the 40th Hawaii International Conference on System Sciences (2006)
9. Kordy, B., Mauw, S., Radomirović, S., Schweitzer, P.: Foundations of attack–defense trees. In: Degano, P., Etalle, S., Guttman, J. (eds.) FAST 2010. LNCS, vol. 6561, pp. 80–95. Springer, Heidelberg (2011)
10. Pat-Cornell, M.E.: Fault trees vs. event trees in reliability analysis. Journal of Risk Analysis **4**(3), 177–186 (1984)
11. Gorton, D.: Using Incident Response Trees as a Tool for Risk Management of Online Financial Services. Journal of Risk Analysis **34**(9), 1763–1774 (2014)
12. Gorton, D.: Modeling fraud prevention of online services using incident response trees and value at risk. In: The Proceedings of the International Conference on Availability, Reliability and Security (ARES) (2015)
13. Wikipedia: Online banking. http://en.wikipedia.org/wiki/Online_banking (Accessed: August 30, 2015)
14. Julisch, K.: Risk-Based Payment Fraud Detection. Research Report, IBM Research, Zurich (2010)
15. Kaspersky: The Great Bank Robbery: Carbanak cybergang steals $1bn from 100 financial institutions worldwide. http://www.kaspersky.com/about/news/virus/2015/Carbanak-cybergang-steals-1-bn-USD-from-100-financial-institutions-worldwide (Accessed: August 30, 2015)
16. Florncio, D., Cormac, H.: Phishing and money mules. In: IEEE International Workshop on Information Forensics and Security, pp. 1–5 (2010)
17. Bank For International Settlements: An Explanatory Note on the Basel II IRB Risk Weight Function (2005)
18. Wikipedia: Agent-based Models. http://en.wikipedia.org/wiki/Agent-based_model (Accessed: August 30, 2015)
19. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G.: MASON: A Multi-agent Simulation Environment. Simulation, 517–527 (2005)
20. Lopez-Rojas, E.A., Gorton, D., Axelsson, S.: Using the RetSim Simulator for Fraud Detection Research. Int. Journal of Simulation and Process Modeling, 144–155 (2015)
21. Lopez-Rojas, E.A., Axelsson, S.: BankSim: a bank payment simulation for fraud detection research. In: The 26th European Modeling and Simulation Symposium (EMSS), pp. 144–152 (2014)
22. ENISA: Introduction to Return on Security Investment (2012)
23. PandaLabs: PandaLabs Annual Report 2012 Summary (2013)