Should Cyber-Insurance Providers Invest in Software Security?

Aron Laszka[1(✉)] and Jens Grossklags[2]

[1] Vanderbilt University, Nashville, TN, USA
laszka.aron@gmail.com
[2] Pennsylvania State University, University Park, PA, USA

Abstract. Insurance is based on the diversifiability of individual risks: if an insurance provider maintains a large portfolio of customers, the probability of an event involving a large portion of the customers is negligible. However, in the case of cyber-insurance, not all risks are diversifiable due to software monocultures. If a vulnerability is discovered in a widely used software product, it can be used to compromise a multitude of targets until it is eventually patched, leading to a catastrophic event for the insurance provider. To lower their exposure to non-diversifiable risks, insurance providers may try to influence the security of widely used software products in their customer population, for example, through vulnerability reward programs.

We explore the proposal that insurance providers should take a proactive role in improving software security, and provide evidence that this approach is viable for a monopolistic provider. We develop a model which captures the supply and demand sides of insurance, provide computational complexity results on the provider's investment decisions, and propose different heuristic investment strategies. We demonstrate that investments can reduce non-diversifiable risks and can lead to a more profitable cyber-insurance market. Finally, we detail the relative merits of the different heuristic strategies with numerical results.

Keywords: Economics of security · Cyber-insurance · Software security · Vulnerability discovery

1 Introduction

Most software suffers from vulnerabilities. Partly, the reason is technical and related to the inherent complexity of software development projects. In addition, economic factors play a significant role. For example, software companies may find it undesirable to invest heavily in the security of their products because customers may not immediately reward such actions (in particular, when they impact the time-to-market, or create backwards-compatibility issues). However, the quality of software critically impacts the security of most parts of an organization's information system. Moreover, popular software products influence the security of *many* organizations. Even though systems may be independently

G. Pernul et al. (Eds.): ESORICS 2015, Part I, LNCS 9326, pp. 483–502, 2015.
DOI: 10.1007/978-3-319-24174-6_25

owned and administrated, they may often exhibit similar software configurations leading to so-called monoculture risks [3,12].

It is a matter of considerable debate on how to address these monoculture risks. For example, organizations may desire some security warranties for the software they deploy, however these are not offered as part of software licenses for commercial software which may even contain substantial warranty disclaimers. In response, a number of public policy changes have been proposed. For example, assigning loss liability for security breaches related to insecure software products to software vendors has been argued to be beneficial [27] and can be welfare-enhancing [2]. But such proposals have not found sufficient policy support.

Some organizations have partially taken matters into their own hands by improving the security of software which is critical for their own operations. For example, Samsung and Google have invested a significant amount of resources into making key software products, such as the Linux kernel, more secure by finding and patching vulnerabilities [8]. In addition, several large companies are now running software and web vulnerability rewards programs to limit the risks related to their own businesses. However, these isolated efforts cannot fully address the security risks related to the diverse landscape of widely used software products, such as popular web-based content-management systems etc.

As an alternative, companies of various sizes may wish to purchase *cyber-insurance* to transfer risks related to the consequences of potentially insecure software. This raises the question whether cyber-insurers would find the prospect of offering such contracts attractive.

From an insurance provider's perspective, the total risk related to each insured company can be decomposed into two parts: *diversifiable risk* and *non-diversifiable risk* (also known as systematic risk or market risk). Diversifiable risk arises from vulnerabilities that pertain to a particular company. For example, the possibility of insider attacks, hardware failures, weak passwords, configuration errors, and human mistakes all contribute to the diversifiable risk of a company (e.g., [22]). In contrast, monoculture risks associated with widely used software products in its client base are a key contributor to non-diversifiable risk of a cyber-insurer.

The existence of diversifiable risk is typically desirable from the perspective of an insurance provider: it provides incentives for companies to purchase insurance, and insurers can account for those risks by maintaining a large and diverse portfolio. In contrast, non-diversifiable risk can cause significant fluctuations in the arrival of cyber-insurance claims, which requires an insurer to set aside a substantial safety capital and may provide a price-barrier impeding the growth of the cyber-insurance market [6].

Insurance providers often incentivize companies to reduce risk with security investments by offering premium reductions. However, typical security investments such as the purchase of security products (including firewalls, IDS, and IPS) and the hiring of auditors who can point out and fix company-specific vulnerabilities do not address non-diversifiable risks. Further, most companies lack both the resources and expertise to make valuable contributions to improving

the security of widely used software products. Consequently, these incentives lower the level of diversifiable risk without having a significant impact on the level of non-diversifiable risk. An insurer would prefer the reverse outcome to increase revenue and to limit its exposure to significant risks.

In this paper, we tackle two interrelated issues. First, we propose a model about the insurability of monoculture risks. Second, we propose to lower these risks by investigating a scenario which provides direct incentives to increase the security of widely used software products.

More specifically, we explore the proposal that cyber-insurers should take a *proactive approach* to improve the security of widely used software products in its customer population and to reduce its aggregate non-diversifiable risk. Specifically, we study whether an insurance provider would find it beneficial to adhere to the following two propositions: (1) An insurer should *not* ask companies to individually invest in security *in exchange* for lower premiums, which is the currently dominant practice. (2) An insurer should rather invest the surplus from the resulting higher premiums into making widely used software products more secure. Measures facilitated by the insurer could include: (1) targeted direct investments in software companies (similar to economically targeted investments of public funds which aim to provide positive collateral benefits [15]), (2) vulnerability reward programs which benefit the software used by its customers, and (3) the hiring of external developer teams for popular open-source software.

For the case of a monopolistic cyber-insurer, we provide evidence that this approach is viable. We develop a model which captures the supply and demand sides for insurance when security outcomes are related to the software products chosen by the insured companies, and insurers can invest in the security of the utilized software. We provide theoretical results highlighting the computational complexity of the insurer's decision-making problem, and propose different heuristic strategies to allocate an investment budget to software security. We demonstrate how investments in software security reduce the occurrence of non-diversifiable risk and lower the insurer's required safety capital. We further detail the relative merits of the different heuristic strategies with numerical analysis.

The proposed approach would constitute a paradigm change for insurance. We argue that novel ways to overcome the currently existing impediments are needed to make cyber-insurance viable for non-diversifiable risks. The approach is feasible because insurance companies are strongly incentivized to lower the magnitude of non-diversifiable risks to reduce their probability of ruin, and they have access to privileged information which could guide their investment decisions. Finally, the approach would have significant positive spillover effects on home users and other typically uninsured entities.

The remainder of this paper is organized as follows. In Sect. 2, we summarize relevant previous work from the areas of cyber-insurance and software-security investments. In Sect. 3, we introduce our modeling framework for cyber-insurance. Then, we present our theoretical and numerical results in Sects. 4 and 5, respectively. Finally, in Sect. 6, we provide concluding remarks and outline future work.

2 Related Work

2.1 Cyber-Insurance

A key objective of our work is to improve the insurability of risks from an insurer's perspective. A functioning market for cyber-insurance and a good understanding of the insurability of diversifiable and non-diversifiable risks both matter, because they signal that stakeholders are able to manage modern threats that cause widespread damage across many systems [1,5]. However, the market for cyber-insurance is developing at a frustratingly slow pace due to several complicating factors, which are discussed in the detailed review of the security economics and cyber-insurance literature by Böhme and Schwartz [7].

In particular, from an attacker's perspective, a group of defenders might appear as a very appealing target because of a high correlation in the risk profiles of the defended resources. For example, even though systems may be independently owned and administrated, they may exhibit similar software configurations leading to monoculture risks [3,12]. Böhme and Kataria study the impact of correlation which is readily observable for an insurer and found that the resulting insurance premiums to make the risks insurable would likely endanger a market for cyber-insurance [6]. Similarly, Chen et al. study correlated risks by endogenizing node failure distribution and node correlation distribution [9]. Lelarge and Bolot model interdependent security with insurance, but assume that there is an insurance provider with an exogenously priced premium [23]. Johnson et al. study the viability of insurance in the presence of weakest-link interdependencies [17].

Non-diversifiable risks may also be caused by interdependent security issues, which have been thoroughly studied outside the context of cyber-insurance (e.g., [13,28]). These works have been reviewed by Laszka et al. [20]. Recently, Johnson et al. investigated interdependent security from an insurance provider's perspective [18,19,21]. They found that real-world networked systems can exhibit substantial non-diversifiable risk, and that estimating the magnitude of this risk is a complex problem due to both theoretical and practical challenges.

2.2 Software Security Investments

Potential improvements to software security frequently focus on finding vulnerabilities in deployed code which is also most relevant to our context (since we focus on widely used software). Public vulnerability disclosure programs (VDP), such as the BugTraq mailing list that emerged more than 20 years ago, have been an important source for companies and the public to receive vulnerability reports from white hats. See also recent work on the Wooyun VDP [29]. However, there has always been a debate on whether VDPs are beneficial to society [10]. On the one hand, Rescorla showed that the pool of vulnerabilities in a software product is very deep with respect to the effort and potential impact of vulnerability discovery efforts [26]. On the other hand, Ozment showed that the pool of vulnerabilities in OpenBSD 2.2 is being depleted and vulnerability rediscovery

is common. He concludes that vulnerability hunting by white hats is socially beneficial [25].

Conceptual work has discussed different approaches to organize and design vulnerability markets [4]. For example, Ozment proposed a vulnerability auction mechanism that allows a software company to measure its software quality as well as encourage vulnerability discovery at an acceptable cost [24]. In addition, some companies such as Facebook, Google and Mozilla have established vulnerability reward programs (VRP) that pay white hats to hack. A study based on the Google VRP and Mozilla VRP has shown that harvesting vulnerabilities from the white hat community is cost effective, and compares favorably to hiring full-time vulnerability researchers [11].

3 Model

Now, we present our modeling framework for studying security investments for cyber-insurance. First, in Sect. 3.1, we describe our model of software-security investments and how software security determines the probability of a company suffering an incident. Then, in Sect. 3.2, we discuss cumulative risks, that is, the expected value and variability of the aggregate loss over all companies. Next, in Sect. 3.3, we first describe the demand-side of the insurance model, which is based on utility-maximizing risk-averse companies. Finally, in Sect. 3.4, we discuss the supply-side and how the insurance provider's profit is affected by individual and cumulative risks. For a list of symbols used in this paper, see Table 1.

3.1 Software Security and Individual Risks

We assume that there are N software products that the insurance provider might invest into, and we let d_i denote the amount of resources that the provider invests into the ith product. For every software product, there is a non-zero probability that a new vulnerability is discovered and exploited before it is patched. We call this probability the vulnerability level of software i and let $V_i(d_i)$ denote its value. We assume that the vulnerability level V_i decreases exponentially with the value of the provider's investment, that is,

$$V_i(d_i) = BV_i \cdot e^{-\gamma_i d_i} \ , \tag{1}$$

where BV_i is the level of vulnerability when there is no security investment from the provider, and γ_i is the efficiency of security investments into software product i.

We assume that there are M companies that want to purchase insurance from the provider. Each company j may use any subset $\mathcal{S}_j$ of all the N software products in our model. We assume that each software product $i \in \mathcal{S}_j$ has a vulnerability with V_i probability independently of the other software products, and a company suffers an incident if any of its software products has a vulnerability. Furthermore, a company may also suffer an incident due to an individual

Table 1. List of Symbols

Symbol	Description
Constants	
BV_i	base vulnerability level of software i
γ_i	efficiency of investing into software i
IR_j	individual risk of company j
W_j	base wealth of company j
L_j	loss of company j in case of an incident
I	interest rate for the insurer
ε	insurer's probability of ruin
Variables and Functions	
V_i	vulnerability level of software i
R_j	risk level of company j
d_i	insurer's investment into securing software i
D	insurer's sum investment into securing software products (i.e., $D = \sum_i d_i$)
S	insurer's safety capital

vulnerability, such as a configuration error, which occurs with IR_j probability. Formally, the probability of company j suffering an incident, denoted by R_j, is

$$R_j = 1 - (1 - IR_j) \prod_{i \in \mathcal{S}_j} (1 - V_i) \ . \tag{2}$$

3.2 Cumulative Risk

In the previous subsection, we described a stochastic risk model that captures security vulnerabilities and individual incidents. Now, consider an aggregate outcome of this model:

1. each software product i had a vulnerability with probability $V_i(d_i)$ (independently of the other software products);
2. every company j that uses a vulnerable software had an incident;
3. each remaining company j had an incident with probability IR_j (independently of the other companies).

We are interested in the total amount of losses over all companies due to incidents. Let L_j denote the loss suffered by company j when an incident happens, and let TL denote the sum of the loss values L_j over all the companies j that suffered incidents (either due to vulnerable software or due to individual vulnerabilities).

First, notice that we can compute the expected total amount of losses $\mathrm{E}[TL]$ easily as

$$\mathrm{E}[TL] = \sum_j L_j R_j \ , \tag{3}$$

where each R_j can be computed efficiently (i.e., in polynomial time) using Eq. (2).

On the other hand, measures of variability (e.g., variance) and quantiles cannot be computed simply from the companies' risk levels R_j, due to the correlations between the incident events caused by the software products. For example, consider two companies with $R_1 = R_2 = 0.5$ and $L_1 = L_2 = 1$. Then, from these values only, we cannot determine the probability of both companies suffering an incident (i.e., the probability $\Pr[TL = 2]$): It is possible that the two companies use completely different sets of software, which means that there are no correlations between the incidents and $\Pr[TL = 2] = 0.25$. However, it is also possible that both companies use exactly the same set of software and $IR_1 = IR_2 = 0$, which means that there is perfect correlation and $\Pr[TL = 2] = 0.5$. In Sect. 4.1, we will show that computing certain properties of TL, which are crucial to providing insurance, is in fact computationally hard.

3.3 Demand-Side Model

For a functioning cyber-insurance market, we need both demand and supply: companies that are willing to purchase insurance and insurers that are willing to provide it.

Now, we introduce our demand-side model, which is based on utility-maximizing risk-averse companies. As it is usual in the literature (see, e.g., [6]), we assume that companies have Constant Relative Risk Aversion (CRRA) utility functions. Furthermore, we also assume that the constant of the relative risk aversion is equal to 1, which means that for a given amount of wealth w, a company's utility is $\ln(w)$. Finally, we let the initial wealth of company j (i.e., the amount of wealth when no incident occurs) be denoted by W_j. Then, the expected utility of company j is

$$R_j \ln(W_j - L_j) + (1 - R_j) \ln(W_j) \ . \tag{4}$$

In the above equation, the first term corresponds to the case when the company suffers an incident and loses L_j, which happens with probability R_j, and the second term corresponds to the case when the company does not suffer an incident, which happens with probability $1 - R_j$.

Since companies are risk averse, they are interested in trading off expected wealth for decreased risks. In the case of purchasing insurance, this means that the company pays a fixed premium p_j to the provider, but in case of an incident, the provider will pay the amount of loss L_i suffered by the company. Hence, when company j purchases insurance for premium p_j, its expected utility is simply

$$\ln(W_j - p_j) \ . \tag{5}$$

As companies are assumed to be utility maximizing, it is optimal for company j to purchase insurance if and only if its utility with insurance is greater than or equal to its expected utility without insurance. Building on Eqs. 5 and 4, we can express the condition for purchasing insurance as

$$\ln(W_j - p_j) \geq R_j \ln(W_j - L_j) + (1 - R_j) \ln(W_j) \tag{6}$$

$$W_j - p_j \geq e^{R_j \ln(W_j - L_j) + (1 - R_j) \ln(W_j)} \tag{7}$$

$$p_j \leq W_j - e^{R_j \ln(W_j - L_j) + (1 - R_j) \ln(W_j)} \ . \tag{8}$$

In our model, we assume that all companies purchase insurance from the provider, who chooses the maximum premiums such that purchasing insurance is the optimal choice for the companies.

3.4 Supply-Side Model

Next, we discuss the final piece in our model, the supply-side of insurance. We assume a monopolist insurance provider who maximizes its expected profit, where profit is defined as the difference between income and expenditure. Besides maximizing its profit, the insurance provider is also risk-averse in the sense that it keeps the probability of ruin below a certain threshold by setting aside a safety capital, which we will discuss shortly.

First, the insurance provider's income is the sum of all the premiums paid by the companies, that is,

$$\text{Income} = \sum_j p_j \ . \tag{9}$$

Since the provider is assumed to be a monopolist, it can ask for the maximal premium (see Eq. 8); hence, we can compute the income as

$$\text{Income} = \sum_j W_j - e^{R_j \ln(W_j - L_j) + (1 - R_j) \ln(W_j)} \ . \tag{10}$$

We assume that insurance premiums are flexible in the sense that the premium values p_j are affected by the provider's investments d_i: higher investment values d_i lead to lower vulnerability values V_i, which in turn lead to lower risk levels R_j and lower premiums p_j. The flexibility of premiums poses challenges to the provider, which we will discuss in Sect. 5.3.

Second, the insurance provider's expected expenditure is

$$\text{Expenditure} = \mathrm{E}[TL] + \sum_i d_i + A + I \cdot S \ , \tag{11}$$

where

- $\mathrm{E}[TL]$ is the expected total amount of claims (i.e., the sum of the losses suffered by the companies),
- $\sum_i d_i$ is the total amount of investments into software security,
- A is the sum of all administrative costs,
- I is the interest rate,
- and S is the safety capital required to keep the probability of ruin below a given probability ε.

The safety capital is set aside by the provider to ensure that it remains solvent. To see why this capital is required, consider the total amount of losses TL: On average, the insurance provider has to pay the expected value $\mathrm{E}[TL]$ of these losses (hence the first term in the right-hand side of Eq. (11)). However, in many outcomes, the realization of the total amount of losses TL exceeds $\mathrm{E}[TL]$; hence, the provider has to set aside S to be able to pay all the claims. More formally, the safety capital is the amount necessary to ensure that

$$\Pr[TL > \mathrm{E}[TL] + S] \leq \varepsilon \ . \tag{12}$$

Since this capital has to be set aside, the provider bears the opportunity cost $I \cdot S$.

4 Theoretical Results and Heuristic Investment Strategies

In this section, we study the computational problems faced by the insurance provider. First, in Sect. 4.1, we show that determining whether a given safety capital is sufficient is computationally hard. Then, in Sect. 4.2, we prove that simulations can approximate the amount of necessary safety capital and, hence, the provider's profit. Finally, in Sect. 4.3, we propose efficient heuristic investment strategies.

4.1 Complexity of Computing the Optimal Safety Capital

Assume for the following analysis that the security-investment values d_i are given and fixed for every software product i, and the insurance provider's decision space is limited to choosing the amount of safety capital S. Recall from Eq. (11) that higher amounts of safety capital lead to higher expenditures for the provider. Consequently, a rational and profit-maximizing provider will try choose the minimum amount of safety capital that will keep its probability of ruin below a threshold ε. We show that this problem is computationally challenging by proving that its decision version, that is, determining whether a given amount of safety capital keeps the probability of ruin below ε, is an NP-hard problem.

Theorem 1. *Given a safety capital S and a threshold probability of ruin ε, determining whether the probability of the total amount of losses TL exceeding $S + \mathrm{E}[TL]$ is greater than or equal to ε is NP-hard.*

The proof of the theorem can be found in Appendix A.1.

4.2 Approximating the Loss Distribution

From Theorem 1, we have that it is computationally hard to find the minimal amount of safety capital that keeps the provider's probability of ruin below a given threshold ε. Consequently, computing the provider's profit for given

security-investment values $(d_1, \ldots, d_N)$ is also computationally hard, since the provider's expenditure is determined by the amount of safety capital.

However, we can approximate the minimal amount of safety capital using simulations as follows. First, generate K outcomes of the risk model as described in Sect. 3.2, and let $tl_1, tl_2, \ldots, tl_K$ be the realizations of TL. Second, let the approximate safety capital $\hat{S}$ be the $\lceil (1-\varepsilon)K \rceil$-th smallest realization (note that if multiple realizations have the same value, they have to be counted separately). The following theorem shows that the probability of ruin for the approximate safety capital $\hat{S}$ converges quickly to the actual probability of ruin.

Theorem 2. *Let $TL_1, TL_2, \ldots, TL_K$ be K independent random variables having the same distribution as TL, and let $\hat{S}$ be the $\lceil (1-\varepsilon)K \rceil$-th smallest of these random variables. Then,*

$$\Pr[TL > \hat{S}] \leq \varepsilon + \frac{1}{K} \ . \tag{13}$$

The proof of the theorem can be found in Appendix A.2.

4.3 Investment Strategies

Since computing the provider's profit is challenging, so is finding the investments $(d_1, \ldots, d_N)$ that maximize the profit. In this subsection, we propose heuristic investment strategies, which we will evaluate numerically in Sect. 5.

First, suppose that we are given an aggregate investment amount D, and our goal is to find the optimal investments $(d_1, \ldots, d_N)$ satisfying $\sum_i d_i = D$, that is, we have to divide the aggregate amount D among the N software products. Here, we propose four heuristic strategies for dividing D: uniform, most-used, proportional, and greedy. Then, we can find good investments $(d_1, \ldots, d_N)$ using these heuristics by searching for the best value of D, which is a simple scalar optimization problem.

Uniform. The *uniform* strategy invests the same amount into all software products. Formally, for every software product i,

$$d_i = \frac{D}{N} \ . \tag{14}$$

The rationale behind this heuristic is that the provider needs to mitigate all common vulnerabilities in order to decrease non-diversifiable risks.

Most-Used. The *most-used* strategy invests only into the most popular software product. Let P_i denote the number of companies that use software product i, that is, $P_i = |\{j : i \in \mathcal{S}_j\}|$. Then, for every software product i,

$$d_i = \begin{cases} D & \text{if } P_i = \max_l P_l \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

The rationale behind this heuristic is that the provider needs to invest into the most-used software only, since vulnerabilities in less popular software cannot cause a large number of incidents.

Proportional. The *proportional* strategy invests into each software product an amount that is proportional to the number of companies using that software. Formally, for every software product i,

$$d_i = \frac{P_i}{\sum_l P_l} \ . \tag{16}$$

This heuristic is a middle ground between the first two heuristics, combining their advantages.

Greedy. The *greedy* strategy divides the aggregate investment amount D according to the following greedy algorithm. First, let the investment into each software be zero. Then, the investments are increased iteratively: in every iteration, compute for each software product i how much would the profit increase if we invested an additional δ into software i, and invest into the software product for which the profit increase is maximal. Formally, the greedy strategy divides the aggregate investment amount D as follows:

$\forall i: \ d_i \leftarrow 0$
while $\sum_i d_i < D$ **do**
  for $i = 1, \ldots, N$ **do**
    $\text{Profit}_i \leftarrow \text{Profit}(d_1, \ldots, d_{i-1}, d_i + \delta, d_{i+1}, \ldots, d_N)$
  end for
  $i^* \leftarrow \text{argmax}_i \text{Profit}_i$
  $d_{i^*} \leftarrow d_{i^*} + \delta$
end while

5 Numerical Results

In this section, we present numerical results on our insurance modeling framework. With these results, we strive to answer two important questions:

- Can the insurance provider increase its expected profit by investing into software security?
- Which heuristic investment strategy leads to the highest expected profit?

First, in Sect. 5.1, we describe how we instantiate our model. Then, in Sect. 5.2, we present the resulting loss distributions both in the case of no software-security investments and in the case of substantial investments. Finally, in Sect. 5.3, we compare the various investment strategies in terms of expected profit to answer the above questions.

5.1 Setup

We instantiated the model with exemplary values to illustrate the relative effect of the investment strategies. First, we generated a set of 15 software products such that, for each software i,

- base vulnerability BV_i was randomly drawn from $[0.09, 0.11]$,
- investment efficiency γ_i was randomly drawn from $[0.9, 1.1]$.

Second, we generated a set of 1500 companies such that, for each company j,

- individual risk IR_j was randomly drawn from $[0.4, 0.6]$,
- base wealth W_j was randomly drawn from $[10, 20]$,
- loss in case of an incident L_j was randomly drawn from $[0.25 \cdot W_j, 0.75 \cdot W_j]$.

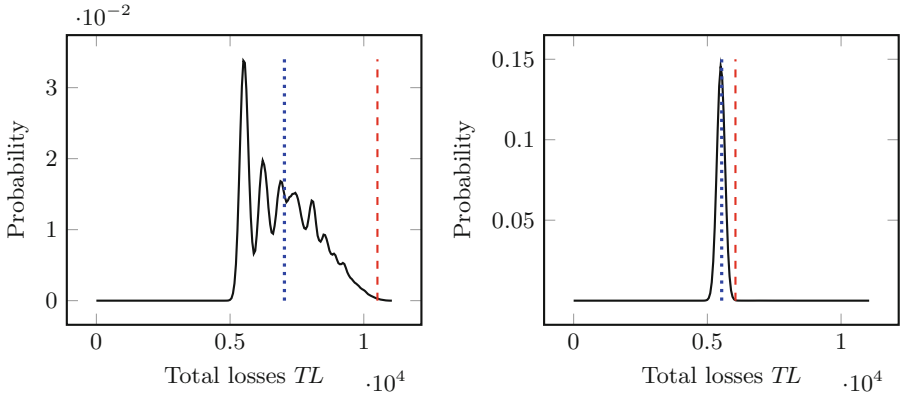
For each company, we choose 3 software products to be used by the company using a popularity-based preferential-attachment model as follows. For the first few companies, the set of software products used by the company was chosen uniformly at random. For the remaining companies, the probability of choosing each software was proportional to the number of companies already using the software. This process models the widely-observed phenomena in which businesses and people tend to choose more popular software products with higher probability, leading to a long-tailed usage distribution [14,16].

Finally, we let the insurance provider's probability of ruin ε be 0.1 %, the interest rate I be 5 %, and the administrative costs A be 0 (i.e., negligible). Note that the value of administrative costs does not affect our analysis, since it is a constant term in the provider's profit, which does not depend on the investment strategy.

5.2 Distribution of the Total Amount of Losses

Figure 1a shows the distribution of the total amount of losses (or, equivalently, the total amount of claims) without any security investments from the provider. We can see that the distribution has a very heavy tail with multiple local maxima, each of which corresponds to vulnerabilities being discovered in one or more widely used software products. Due to this heavy tail, the provider has to set aside a substantial safety capital to avoid ruin: even though the expected amount of claims to be paid is only $\mathrm{E}[TL] = 7\,032$ (marked by dotted blue line on the plot), the amount exceeds 10 510 with probability 0.1 %, that is, $\Pr[TL > 10\,510] = 0.1\,\%$ (marked by dashed red line on the plot). Consequently, in order to keep the probability of ruin below 0.1 %, the provider has to set aside a safety capital of $10\,510 - 7\,032 = 3\,478$.

Figure 1b shows the distribution of the total amount of losses with uniform security investments $d_i = 7.5$ into every software product i. As expected, we can see that the investments decrease both the expected value of the total amount of losses (i.e., total amount of claims) and the necessary safety capital. The expected amount of claims to be paid is $\mathrm{E}[TL] = 5\,536$ (marked by dotted blue line on the plot), while the 0.999 % quantile is 6 051, that is, $\Pr[TL > 6\,051] = 0.1\,\%$ (marked by dashed red line on the plot). Hence, the amount of safety capital that the provider needs to set aside is only $6\,051 - 5\,536 = 515$.



(a) without any investments ($\mathrm{E}[TL] = 7\,032$ and $Q_{TL}(0.999) = 10\,510$)

(b) with uniform investments $d_i = 7.5$ ($\mathrm{E}[TL] = 5\,536$ and $Q_{TL}(0.999) = 6\,051$)

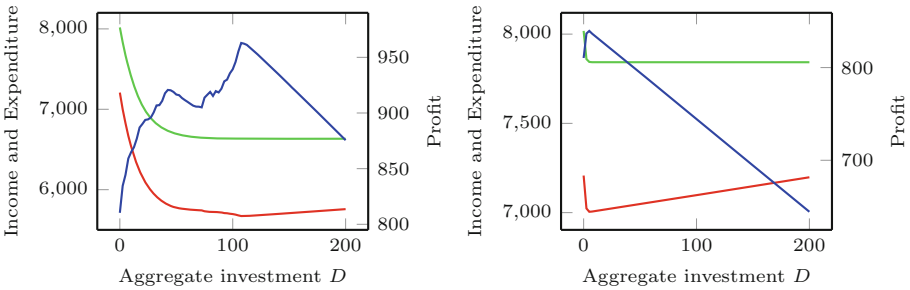
Fig. 1. Probability distribution of the total amount of losses with and without investments. The dotted blue lines mark the expected values, while the dashed red lines mark the 99.9 % quantiles $Q_{TL}(0.999)$ of the distributions (Color figure online).

5.3 Security Investment Strategies

Now, we compare the various investment strategies that we have introduced in Sect. 4.3. For each investment strategy, we compute the insurance provider's income (see Eq. (10)), expenditure (see Eq. (11)), and profit for aggregate investment amounts $D = \sum_i d_i$ ranging from 0 to 200. In each case, we divide the aggregate investment amount D among the software products according to the investment strategy (e.g., with uniform strategy, we let $d_i = \frac{D}{N}$), and approximate the resulting expenditure value using 500 000 simulations of the risk-model outcome.

Recall from Sect. 3.4 that insurance premiums are flexible, that is, the premium values take into account the reductions in risk levels due to the provider's security investments. Consequently, as we increase the value of security investments, we will see a decrease not only in the provider's expenditure, but also in its income due to the decreasing premium values. If we assumed fixed premiums, that is, if the premium values were determined by the base vulnerability levels, then the provider's profit would be strictly higher. Hence, by assuming flexible premiums, we study the conservative scenario, where investments are less beneficial for the insurer (or where the benefits of the security investments are shared between the insurer and the insured companies).

First, Fig. 2a shows the provider's income, expenditure, and profit for the uniform investment strategy. We observe that, as expected, the provider's expenditure drops sharply at first as we increase the investments, due to the rapid decrease in the non-diversifiable risks caused by software vulnerabilities and, hence, in the necessary safety capital. However, once the aggregate investment amount reaches around 110, further investments cannot significantly decrease



(a) uniform investment strategy (b) most-used investment strategy

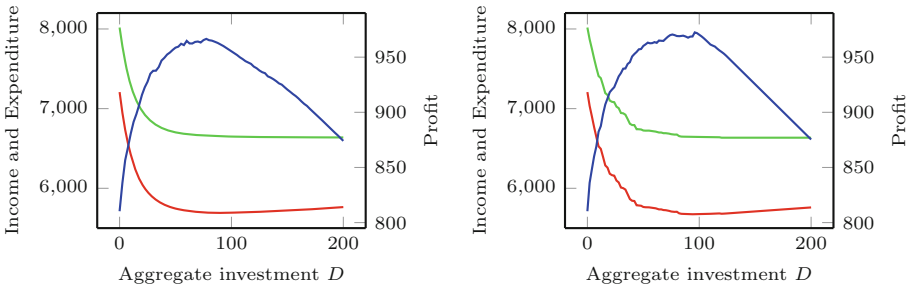
Fig. 2. Income (green), expenditure (red), and profit (blue) of the uniform and the most-used investment strategies for various aggregate security investments. Please note that the scale of the vertical axis for the most-used strategy differs from that for the other strategies (Color figure online).

the necessary safety capital; hence, the expenditure starts increasing due to the increasing cost of investments. The provider's income also drops sharply at first as we increase the investments, due to the rapid decrease in risk levels and, hence, in premium values. Even though the income decreases monotonically for all investment values, once the aggregate investment reaches around 70, the decrease becomes negligible.

On the other hand, the insurance provider's profit is a highly irregular function of the aggregate investment amount, with many local maxima. These irregularities are caused by the combined effects of decreases in expenditure and income, which make finding the optimal investment amount non-trivial. In this example, the maximum profit for the uniform investment strategy is 962, and the maximizing aggregate investment is 107.5. Note that this is substantially better than the case of zero investments, where the profit is only 810.

Second, Fig. 2b shows the provider's income, expenditure, and profit for the most-used investment strategy. Similarly to what we observed for the uniform strategy, we see that the provider's expenditure and income drop sharply at first as we increase the investment, while the profit increases rapidly. However, the profit quickly reaches its maximum value 840 at the investment value 5; and after this point, it decreases monotonically. The explanation for this is the following: securing the most used software eliminates the non-diversifiable risk caused by it, which has a substantial impact due to the large number of companies that are affected; however, once this software product is secure, any further investments will only increase the insurance provider's investment costs without eliminating the non-diversifiable risks caused by the other software. Compared to the other investment strategies, the most used strategy is clearly inferior.

Third, Fig. 3a shows the provider's income, expenditure, and profit for the proportional investment strategy. Again, we see that the income and expenditure take a sharp drop at first, after which the income decreases slowly but monotonically, while the expenditure starts increasing after reaching its minimum at the



(a) proportional investment strategy (b) greedy investment strategy

Fig. 3. Income (green), expenditure (red), and profit (blue) of the proportional and the greedy investment strategies for various aggregate security investments (Color figure online).

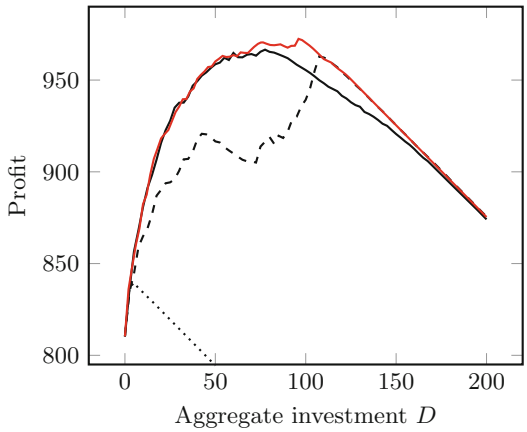


Fig. 4. Profits of the proportional (solid line), uniform (dashed line), most-used (dotted line), and greedy (red line) investment strategies for various investment values. Please note that the profit of the most-used strategy is outside of the plotted vertical range for investment values 50 and above (Color figure online).

aggregate investment 90. However, the profit is a surprisingly smooth function of the investment: it is approximately concave with only a few local maxima, none of which deviate from the general trend substantially.[1] For this strategy, the maximum profit is 967 and the maximizing investment value is 77.5, which means that this strategy is slightly better than the uniform strategy, but the difference is not significant.

Fourth, Fig. 3b shows the provider's income, expenditure, and profit for the greedy investment strategy with increment size $\delta = 2$. We see that the income, expenditure, and profit functions are all very similar to the ones plotted for

[1] Note that these deviations do not diminish as we increase the number of iterations.

the proportional strategy. However, both the maximal profit value 972 and the maximizing investment value 96 are greater than those of the proportional strategy, which shows that this strategy is superior. Furthermore, compared to not investing in security, the maximum profit of the greedy strategy is 20 % higher.

Finally, Fig. 4 compares the proportional (solid line), uniform (dashed line), most-used (dotted line) and greedy (red line) investment strategies for various aggregate investment amounts. This comparison shows how the greedy strategy outperforms the other strategies: For lower investment amounts, where the proportional strategy is optimal (among the considered strategies), the profit of the greedy strategy is almost indistinguishable from that of the proportional strategy. After the proportional strategy reaches its maximum at 77.5, the greedy strategy keeps increasing, until it reaches its maximum at 96. Then, the profit of the greedy strategy decreases until it reaches the maximum of the uniform strategy at 96, after which the profits of the uniform and greedy strategies are almost indistinguishable.

6 Conclusion

In this paper, we have introduced a model for cyber-insurance which incorporates software-security investments. Based on this model, we have shown that the insurance provider's decision-making involves computationally hard problems, and we have proposed different heuristics for security investments. Using numerical results, we have demonstrated that security investments can substantially decrease non-diversifiable risks and increase the profitability of cyber-insurance. Our results show that the viability of the cyber-insurance market, which has been growing very slowly, could be increased through software-security investments. Even though this approach requires a paradigm shift for insurance providers, we believe that they are strongly incentivized to take such a more proactive role.

Our proposal would have significant positive spillover effects on home users and other typically uninsured entities. In future work, we aim to quantify this effect and to also explore the viability of the approach in competitive insurance markets when multiple insurers have to make decisions about which software products to improve.

Acknowledgments. We thank the reviewers for their comments. We gratefully acknowledge the support by the National Science Foundation under Award CNS-1238959, and by the Penn State Institute for CyberScience.

A Proofs

A.1 Proof of Theorem 1

Proof. We prove NP-hardness by showing that a well-known NP-hard problem, the Set Cover Problem, can be reduced to the above decision problem in polynomial time. Given an instance of the Set Cover Problem, that is, a base set U, a

set of subsets $\mathcal{F}$, and limit k on the number of subsets, we construct an instance of our problem as follows:

- For every element of the base set U, there exists a corresponding company.
- For every set in $\mathcal{F}$, there exists a corresponding software product.
- Let the vulnerability level V_i of every software be $\frac{1}{|\mathcal{F}|!}$.
- Let the individual risk IR_j and loss L_j of every company be 0 and 1, respectively.
- Let company j use software i if and only if the corresponding element j is a member of the corresponding set i.
- Let the safety capital S be $|U| - 1 - \mathrm{E}[TL]$.
- Finally, let the probability ε be $\frac{1}{|\mathcal{F}|!^k}$.

Firstly, observe that the above reduction can be performed out in polynomial time.

Next, observe that, in the above instance of our problem, the safety capital S is insufficient to cover all claims if and only if $TL = \sum_j L_j = |U|$, that is, if and only if all companies suffer an incident. Since the individual risk IR_j of every company is 0, this can happen iff, for every company i, there is a vulnerable software product j that is used by i. In other words, it can happen iff the sets in $\mathcal{F}$ corresponding to the compromised software form a cover of the base set U. Hence, it remains to show that the probability of the compromised software forming a set cover is greater than or equal to ε if and only if there exists a set cover of size at most k.

First, suppose that there exists a set cover $\mathcal{C}$ such that $|\mathcal{C}| \leq k$. Then, the probability of all the software products corresponding to the sets in $\mathcal{C}$ being vulnerable is $\frac{1}{|\mathcal{F}|!^k}$. Since $\mathcal{C}$ is a set cover, for every company j, there exists a software product i such that j uses i. Thus, with probability at least $\frac{1}{|\mathcal{F}|!^k}$, every company will suffer an incident and the total amount claims TL will exceed $S + \mathrm{E}[TL]$.

Second, suppose that, for every set cover $\mathcal{C}$, $|\mathcal{C}| > k$. Then, the probability of every company suffering an incident is

$$\Pr[TL > S + \mathrm{E}[TL]] = \Pr\left[\begin{smallmatrix}\text{some collection } C \text{ of software}\\ \text{forming a cover of } U \text{ is vulnerable}\end{smallmatrix}\right] \tag{17}$$

$$= \Pr\left[\begin{smallmatrix}\text{some collection } C \text{ of software}\\ \text{such that } |\mathcal{C}| > k \text{ is vulnerable}\end{smallmatrix}\right] \tag{18}$$

$$= \sum_{l=k+1}^{|\mathcal{F}|} \binom{|\mathcal{F}|}{l} \left(\frac{1}{|\mathcal{F}|!}\right)^l \tag{19}$$

$$< |\mathcal{F}|! \left(\frac{1}{|\mathcal{F}|!}\right)^{k+1} \tag{20}$$

$$= \left(\frac{1}{|\mathcal{F}|!}\right)^k = \varepsilon\ . \tag{21}$$

Since the inequality is strict, we have that the probability of ruin is less than ε if there is no set cover size at most k, which concludes our proof. □

A.2 Proof of Theorem 2

Proof. Let $A_1, A_2, \ldots, A_K, A_{K+1}$ be $K+1$ independent random variables having the same distributions as TL. Then, since all the random variables in $A_1, \ldots, A_{K+1}$ are independent, it follows readily from the definition of $\hat{S}$ that $\Pr[TL > \hat{S}]$ is equal to the probability of a randomly chosen variable in $A_1, \ldots, A_{K+1}$ being greater than $\lceil (1-\varepsilon)K \rceil$ of the other variables in $A_1, \ldots, A_{K+1}$.

Now, we introduce an upper bound for the latter probability as follows. Suppose that we order the realizations $a_1, \ldots, a_{K+1}$ of the random variables $A_1, \ldots, A_{K+1}$ according to their values, with equal realizations being ordered in an arbitrary way. Then, the probability of a randomly chosen variable A_i being greater than $\lceil (1-\varepsilon)K \rceil$ other variables is less than or equal to the probability of choosing a random variable whose realization is not among of the first $\lceil (1-\varepsilon)K \rceil$ realizations, that is, choosing a random variable whose realization is among the last $K+1-\lceil (1-\varepsilon)K \rceil$ realizations. Note that the two probabilities are not necessarily equal because multiple realizations may have the same value. Since we choose a variable from $A_1, \ldots, A_{K+1}$ at random, the probability of picking one whose realization is among the last $K+1-\lceil (1-\varepsilon)K \rceil$ realizations is

$$\frac{K+1-\lceil (1-\varepsilon)K \rceil}{K+1} \tag{22}$$

$$= \frac{K+1-(K-\lfloor \varepsilon K \rfloor)}{K+1} \tag{23}$$

$$= \frac{1+\lfloor \varepsilon K \rfloor}{K+1} \tag{24}$$

$$\leq \frac{1+\varepsilon K}{K} \tag{25}$$

$$= \varepsilon + \frac{1}{K} \, . \tag{26}$$

Consequently, $\Pr[TL > \hat{S}]$ has to be less than or equal to $\varepsilon + \frac{1}{K}$. □

References

1. Anderson, R.J.: Liability and computer security: nine principles. In: Gollmann, D. (ed.) ESORICS 1994. LNCS, vol. 875, pp. 231–245. Springer, Heidelberg (1994)
2. August, T., Tunca, T.: Who should be responsible for software security? A comparative analysis of liability policies in network environments. Manag. Sci. **57**(5), 934–959 (2011)
3. Birman, K., Schneider, F.: The monoculture risk put into context. IEEE Secur. Priv. **7**(1), 14–17 (2009)
4. Böhme, R.: A comparison of market approaches to software vulnerability disclosure. In: Müller, G. (ed.) ETRICS 2006. LNCS, vol. 3995, pp. 298–311. Springer, Heidelberg (2006)
5. Böhme, R.: Towards insurable network architectures. IT - Inf. Technol. **52**(5), 290–293 (2010)

6. Böhme, R., Kataria, G.: Models and measures for correlation in cyber-insurance. In: Proceedings of the 5th Workshop on the Economics of Information Security (WEIS) (2006)
7. Böhme, R., Schwartz, G.: Modeling cyber-insurance: Towards a unifying framework. In: Proceedings of the 9th Workshop on the Economics of Information Security (WEIS) (2010)
8. Brodkin, J.: Google and Samsung soar into list of top 10 Linux contributors (2013). http://arstechnica.com/information-technology/2013/09/google-and-samsung-soar-into-list-of-top-10-linux-contributors/
9. Chen, P., Kataria, G., Krishnan, R.: Correlated failures, diversification, and information security risk management. MIS Q. **35**(2), 397–422 (2011)
10. Egelman, S., Herley, C., van Oorschot, P.: Markets for zero-day exploits: ethics and implications. In: Proceedings of the 2013 New Security Paradigms Workshop (NSPW), Banff, Canada, pp. 41–46 (2013)
11. Finifter, M., Akhawe, D., Wagner, D.: An empirical study of vulnerability rewards programs. In: Proceedings of the 22nd USENIX Security Symposium, Washington, DC, August 2013
12. Geer, D., Pfleeger, C., Schneier, B., Quarterman, J., Metzger, P., Bace, R., Gutmann, P.: Cyberinsecurity: The cost of monopoly. How the dominance of Microsoft's products poses a risk to society (2003)
13. Grossklags, J., Christin, N., Chuang, J.: Secure or insure?: a game-theoretic analysis of information security games. In: Proceedings of the 17th International World Wide Web Conference, pp. 209–218 (2008)
14. Hanson, W., Putler, D.: Hits and misses: Herd behavior and online product popularity. Mark. Lett. **7**(4), 297–305 (1996)
15. Hoffer, D.: A survey of economically targeted investments: opportunities for public pension funds (2004). http://www.vermonttreasurer.gov/sites/treasurer/files/pdf/misc/econTargetInvestReport20040216.pdf
16. Huang, J., Chen, Y.: Herding in online product choice. Psychol. Mark. **23**(5), 413–428 (2006)
17. Johnson, B., Böhme, R., Grossklags, J.: Security games with market insurance. In: Baras, J.S., Katz, J., Altman, E. (eds.) GameSec 2011. LNCS, vol. 7037, pp. 117–130. Springer, Heidelberg (2011)
18. Johnson, B., Laszka, A., Grossklags, J.: The complexity of estimating systematic risk in networks. In: Proceedings of the 27th IEEE Computer Security Foundations Symposium (CSF), pp. 325–336 (2014)
19. Johnson, B., Laszka, A., Grossklags, J.: How many down? toward understanding systematic risk in networks. In: Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security (ASIACCS), pp. 495–500 (2014)
20. Laszka, A., Felegyhazi, M., Buttyan, L.: A survey of interdependent information security games. ACM Comput. Surv. **47**(2), 23:1–23:38 (2014)
21. Laszka, A., Johnson, B., Grossklags, J., Felegyhazi, M.: Estimating systematic risk in real-world networks. In: Christin, N., Safavi-Naini, R. (eds.) FC 2014. LNCS, vol. 8437, pp. 412–430. Springer, Heidelberg (2014)
22. Laszka, A., Johnson, B., Schöttle, P., Grossklags, J., Böhme, R.: Managing the weakest link. In: Crampton, J., Jajodia, S., Mayes, K. (eds.) ESORICS 2013. LNCS, vol. 8134, pp. 273–290. Springer, Heidelberg (2013)
23. Lelarge, M., Bolot, J.: Economic incentives to increase security in the internet: the case for insurance. In: Proceedings of the 33rd IEEE International Conference on Computer Communications (INFOCOM), pp. 1494–1502 (2009)

24. Ozment, A.: Bug auctions: vulnerability markets reconsidered. In: Proceedings of the 3rd Workshop on the Economics of Information Security (WEIS), Minneapolis, MN, May 2004
25. Ozment, A.: The likelihood of vulnerability rediscovery and the social utility of vulnerability hunting. In: Proceedings of the 4th Workshop on the Economics of Information Security (WEIS), Cambridge, MA, June 2005
26. Rescorla, E.: Is finding security holes a good idea? IEEE Secur. Priv. **3**(1), 14–19 (2005)
27. Schneier, B.: Schneier on security: liability changes everything (2003). https://www.schneier.com/essays/archives/2003/11/liability_changes_ev.html
28. Varian, H.: System reliability and free riding. In: Camp, J., Lewis, S. (eds.) Economics of Information Security, pp. 1–15. Kluwer Academic Publishers, Dordrecht (2004)
29. Zhao, M., Grossklags, J., Chen, K.: An exploratory study of white hat behaviors in a web vulnerability disclosure program. In: Proceedings of the 2014 ACM Workshop on Security Information Workers (SIW), pp. 51–58 (2014)