Urgency in cybersecurity risk management: toward a solid theory*

Zoltán Ádám Mann
University of Amsterdam
Amsterdam, The Netherlands

***Abstract*—IT systems are exposed to a rapidly changing landscape of serious security risks. Given the limited resources available to an organization, it is becoming more and more important to properly prioritize security risks, so that the organization can focus its efforts on the most critical risks. Traditionally, risks are assessed in terms of two aspects: occurrence probability and caused damage. However, for real-time risk prioritization, a third aspect is also of critical importance: urgency. Urgency stems from time-related considerations, such as the time needed by adversaries to exploit a vulnerability or the time needed for system administrators to put a countermeasure in place. These time-related considerations are orthogonal to the traditional aspects of occurrence probability and caused damage, and are largely ignored by existing risk management approaches.**

This paper proposes a way for introducing the notion of urgency into risk assessment. Our aim is to devise an intuitive approach for assessing risks, taking urgency into account, based on a solid theoretical underpinning. We establish a mathematical model using probability theory, and derive formulas for time-aware risk assessment in different settings.

***Index Terms*—cybersecurity, IT security, security risks, risk assessment, risk prioritization, risk mitigation, probabilistic models**

I. Introduction

Organizations in all sectors rely increasingly on Information and Communication Technology (ICT). At the same time, ICT services become increasingly complex and interconnected. As a result, cybersecurity threats lead to numerous incidents, incurring high costs and threatening critical services [1].

Because of the size and complexity of their ICT landscape, organizations typically face many cybersecurity threats at the same time. However, given their limited resources, organizations typically cannot address all cybersecurity threats at the same time. Therefore, *prioritization* of cybersecurity threats is crucial to ensure that organizations use their resources for addressing the most critical threats.

For prioritization, risk-based approaches have proven useful. Risk-based assessment and prioritization approaches are widely used in various management disciplines, such as project management [2] or test management [3]. Such approaches allow a systematic and intuitive way of assessing and quantifying risks, which can be used as a basis for sound decisions on which risks to mitigate. In IT security, well-known risk-based approaches include the ISO 27001 standard [4] and the NIST Cybersecurity Framework [5].

* This paper was published in the *Proceedings of the 37th IEEE Computer Security Foundations Symposium (CSF)*, pp. 589-602, 2024

Most current methodologies for risk assessment advocate estimating the likelihood and the impact of the risks [6]. This way, risks can be arranged in two dimensions, often referred to as a risk matrix [7]. For the purpose of prioritization, a risk score can be derived from the estimates of likelihood and impact. The risk score should be monotonously increasing in both likelihood and impact. Often, the risk score is calculated as the product of likelihood and impact.

Security risks can be assessed in different situations. For example, risk assessment can be performed during system design and development [8], as part of a security audit [9], or after a security breach happened [10]. A common characteristic of these situations is that risks are not exploited by adversaries in real time, i.e., during the assessment of the risks. In these situations, the traditional approach of quantifying risks based on likelihood and impact seems to work fine [11].

An increasingly important use case for risk assessment entails assessing risks in real time during the operation of the ICT landscape of an organization [12], [13], [14]. Likelihood and impact are important characteristics of risks in this situation as well. However, real-time risk assessment requires the consideration of an additional dimension: *urgency*. In time management, importance and urgency are recognized as the two independent dimensions relevant for prioritization decisions [15]. In real-time security risk assessment, importance is given by a combination of likelihood and impact (just as in non-real-time risk assessment), and urgency is an additional dimension.

Urgency stems from time-related factors, such as the time needed by adversaries to exploit a vulnerability or the time needed by system administrators to put a countermeasure in place. These factors can have a significant impact on real-time risk prioritization decisions. For example, a vulnerability that adversaries can exploit quickly may require faster mitigation than a vulnerability that is more time-consuming to exploit, and countermeasures with a long lead time may need to be started earlier than countermeasures with a shorter lead time.

Traditionally, risk mitigation by appropriate countermeasures has been seen as a separate step in the risk management process after risk assessment [16], [17]. However, in real-time risk assessment, the duration of countermeasures may impact prioritization decisions. This intertwining of risk assessment and selection of mitigation actions is an additional difference between real-time and non-real-time risk assessment.

These considerations highlight the need for a novel security

risk assessment method that can be used in real-time settings. Such a method must take into account, beside likelihood and impact of a risk, also its urgency, stemming from time-related features of attacks and countermeasures. Such a method should yield a sound prioritization of security risks, and should ideally be intuitive and simple to understand. In addition, the method should strike a balance between considering the duration of countermeasures and the time needed for the analysis of countermeasures.

This paper makes significant steps toward achieving these goals. We define a process for real-time security risk management, featuring two phases of risk analysis. In the initial risk analysis, all identified risks are analyzed, without considering countermeasures, yielding an initial prioritization. For the highest-ranked risks, countermeasures are devised and analyzed, which allows a second, more detailed analysis of these risks, taking into account the duration of countermeasures as well.

Using probability theory as a sound theoretical basis, this paper suggests a mathematical framework for real-time risk assessment. This framework allows us to quantify the risk stemming from different types of vulnerabilities. We identify four different cases, depending on whether a successful exploitation of a vulnerability generates a one-time damage or continual damage, and whether multiple exploitations of the same vulnerability lead to increased damage. In each case, we derive formulas for quantifying risks.

The contributions of this paper can be summarized as follows:

- A process for real-time security risk management, featuring an initial and a detailed risk analysis phase.
- A general mathematical model for real-time security risk assessment, based on probability theory.
- Specific formulae for quantifying the risk of four different types of vulnerabilities.
- Examples to showcase the practical applicability of the process, the general mathematical model, and the specific formulae.

The rest of the paper is organized as follows. Section II clarifies basic notions and describes the specifics of real-time risk quantification. Section III introduces a general mathematical model for quantifying risks in run-time risk assessment, while Section IV concretizes the general mathematical model for different classes of vulnerabilities. Afterwards, Section V discusses the results and Section VI reviews related work. Finally, Section VII concludes the paper.

II. Real-time risk management

A. System model and terminology

Inspired by previous work [12], [18], we use the system model shown in Fig. 1, which can be described as follows:

- We want to protect a socio-technical **system**, comprising of a set of **assets**. We focus here on digital assets, such as hardware, software, and data.

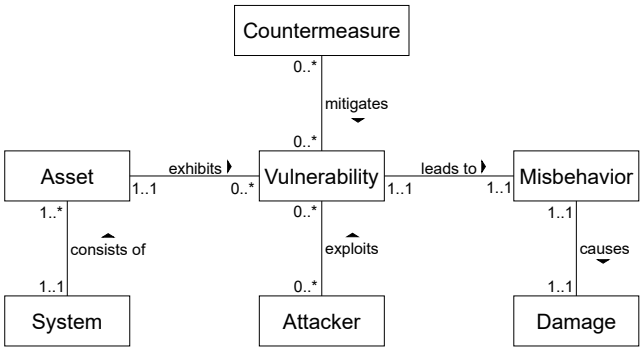

Fig. 1. System model, using UML notation. Rectangles represent types, edges represent relationships between types. Each relationship has a label to describe its meaning and a triangle showing in which direction the label should be interpreted. In addition, the ends of an edge show the allowed cardinalities in the form $\min .. \max$. Here, $\min$ is a number specifying the minimum cardinality, while $\max$ is either a number specifying the maximum cardinality or $*$ if the cardinality is not bounded from above. For example, the edge between System and Asset specifies that a system consists of at least one asset, while an asset belongs to exactly one system.

- An asset may exhibit certain **vulnerabilities**. A vulnerability is an unintended property of the asset, which could potentially lead to a **misbehavior** of the asset.
- A misbehavior causes **damage**.
- An **attacker** may *exploit* a vulnerability to cause a misbehavior.
- A **countermeasure** (also called mitigation action) may *mitigate* a vulnerability, eliminating the misbehavior that would result from the vulnerability.

In addition, we use the following terminology concerning risks throughout the paper:

- A cybersecurity **risk** is the probabilistic event that an attacker manages to exploit a vulnerability, leading to a misbehavior and thus to damage.
- A **risk value** (also called risk score) is a non-negative real number associated with a risk. Risk values are used for prioritizing risks. Risks with a higher risk value should be prioritized over risks with a lower risk value.

B. Risk management process

Fig. 2(a) shows a simplified overview of a traditional risk management process, as mandated for example by the ISO 2700x family of standards. In such a process, first the noteworthy risks are identified. The identified risks are analyzed in terms of their likelihood and impact, and prioritized based on these metrics. Finally, for the risks with sufficiently high priority, countermeasures are devised and implemented.

The real-time risk management process is a bit different, as shown in Fig. 2(b). The main difference stems from the role of mitigation actions. On the one hand, the prioritization of risks should take the duration of mitigation actions into account. This is because threats for which mitigation takes longer may require starting the mitigation sooner than for other threats that can be mitigated quickly. On the other hand, devising and analyzing mitigation actions for a threat may take significant

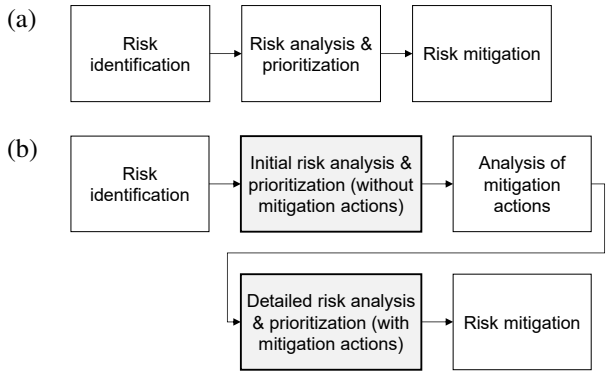

Fig. 2. Risk management process: (a) traditional, (b) real-time

effort and time. Thus, the organization may not be able to afford devising and analyzing mitigation actions for all risks. This is why Fig. 2(b) contains two risk analysis steps. First, the identified risks are analyzed and prioritized without knowledge of mitigation actions (*initial risk analysis*). For the risks with the highest priority, mitigation actions are then devised and analyzed. On the basis of the duration of mitigation actions, risks are re-assessed and re-prioritized (*detailed risk analysis*). Finally, mitigation is implemented for those risks that have the highest priority after the second prioritization.

III. Mathematical model

To enable risk prioritization, risks should be quantified. For each vulnerability v, a risk value $R(v) \in \mathbb{R}_{\geq 0}$ should quantify its criticality.

A. Traditional risk quantification

The widely-adopted method for risk quantification is that the risk value associated with a misbehavior is a monotonously increasing function of the probability and of the consequence (i.e., caused damage) of the misbehavior. As function, typically multiplication is used; that is, the risk value $R(v)$ associated with a misbehavior caused by vulnerability v is[1]

$$R(v) = P(v) \cdot D(v), \tag{1}$$

where $P(v)$ is the probability that the misbehavior occurs and $D(v)$ is the damage caused if the misbehavior occurs [19], [20], [21].

It is useful to establish that Equation (1) calculates the risk value as the *expected damage* in a simple probabilistic model. In this model, there are two possible outcomes: the vulnerability is either exploited or not. The first outcome has probability $P(v)$ and leads to damage $D(v)$; the second outcome has probability $1 - P(v)$ and leads to damage 0. In this model, the expected value of the damage is exactly $P(v) \cdot D(v)$. Formally, let ξ_v be a random variable corresponding to the damage caused by the potential exploitation of vulnerability v; then, $\Pr(\xi_v = D(v)) = P(v)$ and $\Pr(\xi_v = 0) = 1 - P(v)$, and

$$R(v) = \mathbb{E}[\xi_v]. \tag{2}$$

[1]Table I gives a summary of the notation used in the paper.

TABLE I
Notation overview

Notation	Description
v	A vulnerability
$R(v)$	Risk value of vulnerability v
$P(v)$	Probability of an attacker exploiting vulnerability v
$D(v)$	Damage caused if vulnerability v is exploited
ξ_v	Random variable representing the damage from vulnerability v
T	Look-ahead horizon
$\xi_{v,t}$	Random variable representing the damage from v in $[0, t)$
$\lambda(v)$	Rate of attackers succeeding in exploiting vulnerability v
$R_T(v)$	Risk value of vulnerability v within look-ahead horizon T
$\delta(v)$	Marginal damage per unit time
m	A mitigation action
$\tau(m)$	Time to implement mitigation action m
$R(v, m)$	Risk value of vulnerability v, considering mitigation action m

B. Real-time risk quantification

For real-time risk assessment, we can reuse the idea of Equation (2). The details are different depending on whether risk quantification is performed as part of the initial risk analysis (without mitigation actions) or the detailed risk analysis (with mitigation actions), as introduced in Section II-B.

1) Initial risk analysis: The point in time in which risk assessment is performed is denoted as $t = 0$. Any damage incurred in the past ($t < 0$) is sunk cost, and as such, should not be taken into account in decision making [22]. The risk value should thus be the expected damage of the future, i.e., for $t \in [0, \infty)$.

However, there are several problems with aggregating damage in $[0, \infty)$. First, every vulnerability will become exploited if attackers have sufficient time, yielding an occurrence probability of 1 for every risk, and thus blurring the difference between vulnerabilities that are easily exploited and the ones that are hard to exploit. Second, for all vulnerabilities leading to damage proportional to the time after successful exploitation, the aggregation would lead to infinite damage, again blurring any differences among these vulnerabilities. Third, it is practically not feasible to correctly reason about the far future, since organizations cannot know exactly how their ICT will evolve over time and how that evolution will impact vulnerabilities and their impact.

For these reasons, it is useful to fix a *look-ahead horizon* $T > 0$, and to assess the expected damage for $t \in [0, T)$. More formally, for any $t > 0$, let $\xi_{v,t}$ be a random variable corresponding to the damage caused by the exploitation of vulnerability v in the time interval $[0, t)$. (Note that if the vulnerability is exploited multiple times in this time interval, then $\xi_{v,t}$ is the total damage stemming from these exploits.) Then, we can define the risk value of v, with respect to the look-ahead horizon T, as

$$R_T(v) = \mathbb{E}[\xi_{v,T}]. \tag{3}$$

By using the same look-ahead horizon T in the calculation of the risk value $R_T(v)$ using Equation (3) for all identified vulnerabilities, we can compare the expected damage stemming from the vulnerabilities. This provides a sound basis

for selecting the vulnerabilities with the highest expected damage. These vulnerabilities are then analyzed further to devise appropriate countermeasures for mitigating them.

2) Detailed risk analysis: After mitigation actions have been devised for the selected risks, we can also take the time-related effect of the mitigation actions into account. For a mitigation action m, let $\tau(m)$ denote the time it takes to implement m. For different mitigation actions, this duration can be of very different magnitude [23]. In some cases, implementing the mitigation action can be almost instantaneous, such as in the case of activating a setting in a firewall. Implementing some other mitigation actions may take several minutes or even hours (e.g., for spinning up a new server in an appropriate network segment and migrating applications to that server). In some cases, for example if implementing the mitigation action involves the procurement of new hardware or contributions from experts with limited availability, then this may take days, weeks, or even longer.

We perform risk assessment at time $t = 0$. We assume that a mitigation action has no effect on the damage before it is fully put in place, and that the vulnerability causes no further damage after the mitigation action has been put in place [23]. That is, if the mitigation action m for vulnerability v is started immediately, then v will cause no further damage after time $t = \tau(m)$. Therefore, in this case, the total expected damage of v is $\mathbb{E}[\xi_{v,\tau(m)}]$. In other words, expected damage can be calculated similarly as in the initial risk analysis, but the considered time interval is $[0, \tau(m))$ instead of $[0, T)$.

The above holds if the mitigation action is started immediately. The expected damage $\mathbb{E}[\xi_{v,\tau(m)}]$ will be incurred in any case, and is thus irrelevant for prioritization decisions. For prioritization, the interesting case is when not all mitigation actions can be started immediately, for example because there are not enough resources for implementing all mitigation actions in parallel. The additional damage stemming from delaying the implementation of some mitigation actions is what should be considered during prioritization, since this additional damage depends on the prioritization decisions. If mitigation action m for vulnerability v is delayed by Δt, then the mitigation action will only have an effect after $\tau(m) + \Delta t$ time instead of after $\tau(m)$ time. Therefore, the expected damage stemming from vulnerability v will be $\mathbb{E}[\xi_{v,\tau(m)+\Delta t}]$ instead of $\mathbb{E}[\xi_{v,\tau(m)}]$. Using a differential approximation, the *additional expected damage* resulting from the delay by Δt can be estimated as (see also Fig. 3):

$$\mathbb{E}[\xi_{v,\tau(m)+\Delta t}] - \mathbb{E}[\xi_{v,\tau(m)}] \approx \frac{d}{dt}\mathbb{E}[\xi_{v,t}]\bigg|_{t=\tau(m)} \cdot \Delta t. \quad (4)$$

Prioritization should aim at minimizing this additional damage resulting from delayed implementation of mitigation actions. Therefore, vulnerabilities with a high additional expected damage should be prioritized. Based on Equation (4), risk values should be assigned as follows:

$$R(v, m) = \frac{d}{dt}\mathbb{E}[\xi_{v,t}]\bigg|_{t=\tau(m)}. \quad (5)$$

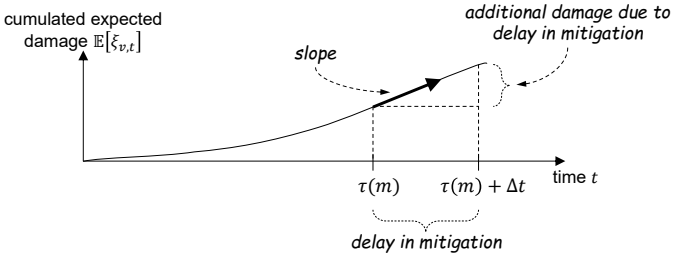

Fig. 3. The derivative of $\mathbb{E}[\xi_{v,t}]$ in $t = \tau(m)$ helps estimate the additional expected damage resulting from a delay of Δt in implementing mitigation action m

By prioritizing ⟨vulnerability, mitigation action⟩ pairs for which $R(v, m)$ is the highest, the total additional expected damage can be minimized.

IV. Calculating expected damage over time

From the previous section it can be seen that calculating $\mathbb{E}[\xi_{v,t}]$ is the main step in both the initial risk analysis and the detailed risk analysis. In the initial risk analysis, risk values are obtained by evaluating $\mathbb{E}[\xi_{v,t}]$ at $t = T$, where T is the chosen look-ahead horizon, which is the same for all vulnerabilities. In the detailed risk analysis, risk values are obtained by evaluating the derivative of $\mathbb{E}[\xi_{v,t}]$ at $t = \tau(m)$, where m is the mitigation action found for vulnerability v. Therefore, in this section, we concentrate on how to calculate $\mathbb{E}[\xi_{v,t}]$ for different types of vulnerabilities.

A. Types of vulnerabilities

For different types of vulnerabilities, the expected damage over time can be calculated using different formulas. In particular, the following factors play an important role in the calculation of the expected damage over time:

- *One-off versus time-proportional damage*. For some vulnerabilities, a successful exploit leads to a one-off damage immediately or a short time after the successful exploit. For example, a successful ransomware attack may lead to a one-time ransom payment. For other vulnerabilities, a successful exploit leads to damage proportional to the time after the exploit during which the caused misbehavior persists. For example, the damage created by a successful denial-of-service attack is proportional to the duration of the unavailability of the service.
- *Single versus multiple exploitation*. For some vulnerabilities, only the first successful exploit creates damage. For example, if exploiting a vulnerability crashes a system, then the first successful attack will lead to damage, subsequent attacks not, because the system is already down. For other vulnerabilities, each successful exploit leads to additional damage. For example, if a vulnerability can be exploited for mining cryptocurrencies, then each successful attacker will incur additional damage.

Although not relevant in the traditional (non-real-time) setting, these dimensions of differentiation are important in real-time risk assessment because they influence how much

TABLE II
CLASSIFICATION OF CASES FOR REAL-TIME RISK QUANTIFICATION

Exploitation	Damage	
	One-off	*Time-proportional*
Single	Case 1	Case 2
Multiple	Case 3	Case 4

damage is incurred by a vulnerability in a given amount of time. These two dimensions of differentiation lead to 4 different cases (see Table II); each case leads to a different formula for the expected damage over time, as detailed in the following subsections.

The description of each case follows the same structure: we first describe the intuitive idea behind our calculations, then provide a formal treatment, discuss the used parameters, and give an example of the practical application of the derived formula. In the formal treatment, we first devise a formula for $R_T(v) = \mathbb{E}[\xi_{v,T}]$ for initial risk analysis, followed by the formula for $R(v,m) = \frac{d}{dt}\mathbb{E}[\xi_{v,t}]\big|_{t=\tau(m)}$ for detailed risk analysis. The application examples focus on the initial risk analysis for now; a more comprehensive case study including also the detailed risk analysis is presented in Section IV-F.

B. Case 1: single exploitation, one-off damage

Intuitive description. This case is applicable for vulnerabilities that, upon the first successful exploitation, lead to a one-off damage, and no further damage is created by further exploitations. The only time-related aspect of this case is the – unknown – time that it takes an attacker to exploit the vulnerability. Vulnerabilities that can be more quickly exploited have a higher probability of being exploited within the look-ahead horizon than vulnerabilities that take more time to exploit. The idea is sketched in Fig. 4.

Formalization. The process of attackers succeeding in exploiting a vulnerability and causing a misbehavior can be modeled as a stochastic process. In line with previous research [24], [25], [26], we assume that a Poisson process can be used to approximate the process of successful attacks. This implies that the time until the first successful attack is a random variable χ that follows an exponential distribution. Thus, the probability of a successful attack within time T is

$$P_T(v) = \Pr(\chi < T) = 1 - e^{-\lambda(v)\cdot T}, \tag{6}$$

where the Poisson rate $\lambda(v) > 0$ specifies how quickly attackers can exploit vulnerability v. The notation $P_T(v)$ is used to emphasize that the probability of a successful exploit *within the look-ahead horizon* T is considered, as opposed to $P(v)$, which does not make reference to a specific time frame.

Each vulnerability can be associated with a different value of $\lambda(v)$, signifying how quickly the given vulnerability may be exploited. After fixing a look-ahead horizon T, Equation (6) yields a probability value for each vulnerability that takes into account how quickly the given vulnerability could be exploited. Plugging these probability values into Equation (3) yields the following formula for the risk value:

$$R_T(v) = P_T(v) \cdot D(v) = \left(1 - e^{-\lambda(v)\cdot T}\right) \cdot D(v). \tag{7}$$

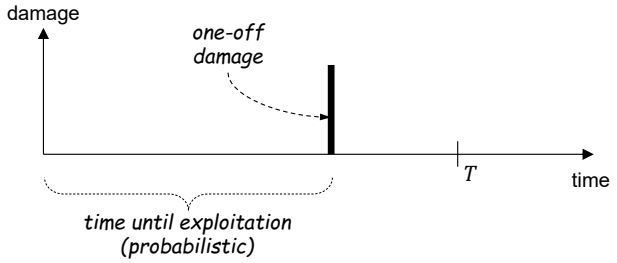

Fig. 4. Case 1: exploitation of the vulnerability happens according to a stochastic process and causes a one-off damage

The risk value for the *detailed risk analysis*, where also a risk mitigation action m is available, can be obtained as follows. Analogously to Equation (7), we have

$$\mathbb{E}[\xi_{v,t}] = \left(1 - e^{-\lambda(v)\cdot t}\right) \cdot D(v). \tag{8}$$

From this, we get

$$\frac{d}{dt}\mathbb{E}[\xi_{v,t}] = D(v) \cdot \lambda(v) \cdot e^{-\lambda(v)\cdot t}, \tag{9}$$

and thus

$$R(v,m) = \frac{d}{dt}\mathbb{E}[\xi_{v,t}]\bigg|_{t=\tau(m)} = D(v) \cdot \lambda(v) \cdot e^{-\lambda(v)\cdot \tau(m)}. \tag{10}$$

Parameters. The look-ahead horizon $T > 0$ is a global parameter. T is time-dimensional, and can be freely chosen by the organization. Reasonable values of T are in a time range where attackers can already exploit some vulnerabilities, but probably not all vulnerabilities, so that the model can differentiate well between easy and time-consuming exploits. See also Section V-D for further discussion.

In addition, the parameter $\lambda(v) > 0$ is introduced for each vulnerability v, as the rate parameter of the exponential distribution. The dimension of this parameter is the inverse of time, with possible units such as sec^{-1} or hour^{-1}. The intuitive meaning of this parameter is given by the fact that the expected value of the exponential distribution with rate λ is λ^{-1}. Thus, $\lambda(v)$ is the inverse of the expected time needed by an attacker to exploit vulnerability v. For example, if the expected time to exploit a vulnerability is 10 hours, then the corresponding $\lambda(v) = 0.1\ \text{hour}^{-1}$.

Application. Let us assume that a vulnerability scanner identified two vulnerabilities in two different software assets. Software asset 1 is vulnerable to a buffer overflow attack that could potentially crash the system. Software asset 2 is vulnerable to an SQL injection attack that could potentially let an attacker gain access to sensitive data. It is estimated that an attacker needs on average about 10 hours to exploit the first vulnerability, and about 40 hours to exploit the second vulnerability. If exploited, the first vulnerability leads to an expected

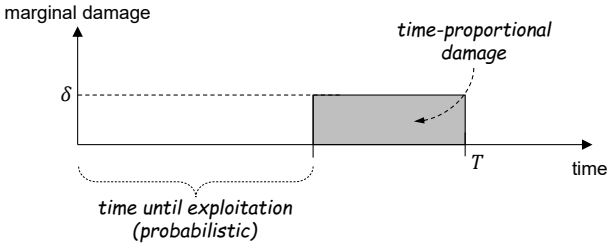

Fig. 5. Case 2: a misbehavior that persists for Δt time leads to a damage of $\delta \cdot \Delta t$

damage of 10,000 USD, while the second vulnerability leads to an expected damage of 50,000 USD. Information about possible mitigation actions is not yet available. Assuming that the organization cannot address both vulnerabilities at the same time (e.g., because of limited resources), which vulnerability should be addressed first?

The answer to this question is not at all clear. The second vulnerability leads to higher damage, so it may make sense to focus on that first. However, the first vulnerability can be exploited faster, so it may be a better strategy to address that risk first. Decision-making is complicated by the fact that the attacker's capabilities are not known; thus, the time the attacker really needs to exploit the vulnerabilities may deviate significantly from the estimated average. Without the model introduced here, the organization has no sound way to make a decision. In particular, Equation (1) cannot be applied because no probabilities are known.

With the model introduced here, risks can be quantified. Since attackers need on average 10 hours to exploit vulnerability v_1, we have $\lambda(v_1) = 1/10 \text{ hour}^{-1} = 0.1 \text{ hour}^{-1}$. Similarly, since attackers need on average 40 hours to exploit vulnerability v_2, we have $\lambda(v_2) = 1/40 \text{ hour}^{-1} = 0.025 \text{ hour}^{-1}$. The damage stemming from the two vulnerabilities, if successfully exploited, is $D(v_1) = 10000$ \$ and $D(v_2) = 50000$ \$. Using $T = 24$ hour as look-ahead horizon, Equation (7) yields

$$R_T(v_1) = \left(1 - e^{-0.1 \cdot 24}\right) \cdot 10000 \text{ \$} \approx 9093 \text{ \$}$$

and

$$R_T(v_2) = \left(1 - e^{-0.025 \cdot 24}\right) \cdot 50000 \text{ \$} \approx 22559 \text{ \$}.$$

Thus, the second vulnerability should be prioritized.

C. Case 2: single exploitation, time-proportional damage

Intuitive description. For some vulnerabilities, the damage created by a successful exploit is time-dependent: the longer the misbehavior persists, the more damage it creates. For example, if a denial-of-service attack interrupts a service, this may lead to damage. The longer the service is unavailable, the higher the damage. In this case, we assume that the damage is proportional to the duration of the misbehavior. The idea behind this case is sketched in Fig. 5.

Formalization. To formalize time-dependent damage, let us assume that misbehavior v leads to a *marginal* damage of $\delta(v) \geq 0$. That is, if the misbehavior exists for a period of length Δt, this leads to a damage of $\delta(v) \cdot \Delta t$. The misbehavior – and thus the accumulation of damage – starts when the vulnerability has been successfully exploited by an attack.

Like in Section IV-B, we assume that a Poisson process can be used to approximate the process of successful attacks, leading to exponential distribution for the time until a successful exploit. According to Equation (6), the probability that vulnerability v has been successfully exploited by time t is $1 - e^{-\lambda(v) \cdot t}$. If the vulnerability has already been exploited by time t, then the damage in an infinitesimal time interval $[t, t + \Delta t)$ is $\delta(v) \cdot \Delta t$. Otherwise, the damage in this interval is 0. Thus, the expected damage in this time interval is $\left(1 - e^{-\lambda(v) \cdot t}\right) \cdot \delta(v) \cdot \Delta t$. The risk value can be computed as the total expected damage in $[0, T)$:

$$R_T(v) = \int_{t=0}^{T} \left(1 - e^{-\lambda(v) \cdot t}\right) \cdot \delta(v) \, dt. \tag{11}$$

This can be simplified as follows:

$$\begin{aligned} R_T(v) &= \delta(v) \cdot \left(T - \int_{t=0}^{T} e^{-\lambda(v) \cdot t} \, dt\right) = \\ &= \delta(v) \cdot \left(T + \frac{e^{-\lambda(v) \cdot T} - 1}{\lambda(v)}\right). \end{aligned} \tag{12}$$

The risk value for the *detailed risk analysis*, where also a risk mitigation action m is available, can be obtained as follows. Analogously to Equation (12), we have

$$\mathbb{E}[\xi_{v,t}] = \delta(v) \cdot \left(t + \frac{e^{-\lambda(v) \cdot t} - 1}{\lambda(v)}\right). \tag{13}$$

From this, we get

$$\frac{d}{dt}\mathbb{E}[\xi_{v,t}] = \delta(v) \cdot \left(1 - e^{-\lambda(v) \cdot t}\right), \tag{14}$$

and thus

$$R(v, m) = \left.\frac{d}{dt}\mathbb{E}[\xi_{v,t}]\right|_{t=\tau(m)} = \delta(v) \cdot \left(1 - e^{-\lambda(v) \cdot \tau(m)}\right). \tag{15}$$

Parameters. Beside the parameters already used in the previous case, this case introduces the marginal damage $\delta(v)$ for every vulnerability v. Marginal damage measures damage per time unit, thus its unit could be, for example, USD/hour.

Application. Let us assume that a vulnerability scanner identified vulnerabilities in two software assets. Software asset 1 is vulnerable to a privilege escalation attack that could make asset 1 unavailable. Software asset 2 is vulnerable to an SQL injection attack that could make asset 2 unavailable. It is estimated that an attacker needs on average about 10 hours to exploit the first vulnerability, and about 40 hours to exploit the second vulnerability. The unavailability of asset 1 leads to a damage of 1000 USD per hour, while the unavailability of asset 2 leads to a damage of 1500 USD per hour.

Quantifying the two risks is challenging because neither the exploit probability nor the damage of a successful exploit is known; thus, Equation (1) is not applicable. The model proposed here resolves this problem because it allows

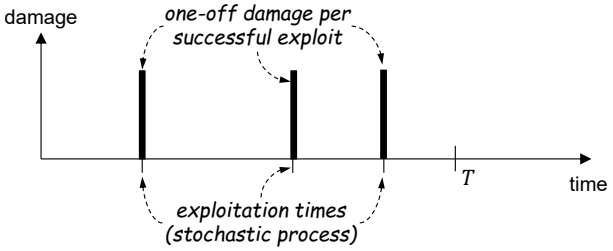

Fig. 6. Case 3: one-off damage for each successful exploitation of the vulnerability. Successful exploitations occur according to a stochastic process.

calculating the risk value from the rate parameter and the marginal damage associated with the vulnerabilities. In the specific case, $\lambda(v_1) = 0.1$ hour^{-1}, $\lambda(v_2) = 0.025$ hour^{-1}, $\delta(v_1) = 1000\ \frac{\$}{\text{hour}}$, and $\delta(v_2) = 1500\ \frac{\$}{\text{hour}}$. We continue using 24 hours for the look-ahead horizon T. Using Equation (12), we get

$$R_T(v_1) = 1000\ \frac{\$}{\text{hour}} \cdot \left(24\ \text{hour} + \frac{e^{-0.1 \cdot 24} - 1}{0.1\ \text{hour}^{-1}}\right) \approx 14907\ \$$$

and

$$R_T(v_2) = 1500\ \frac{\$}{\text{hour}} \cdot \left(24\ \text{hour} + \frac{e^{-0.025 \cdot 24} - 1}{0.025\ \text{hour}^{-1}}\right) \approx 8929\ \$.$$

Thus, the first risk should be prioritized.

D. Case 3: multiple exploitation, one-off damage

Intuitive description. In the cases so far, we assumed that the damage arises if one attacker manages to exploit the vulnerability. For some vulnerabilities, this assumption is appropriate. For example, if exploiting the vulnerability results in the failure of a service, then the damage is incurred as soon as one attacker manages to exploit the vulnerability, and further attacks do not lead to additional damage, since the service is already unavailable. For other types of vulnerabilities, this assumption is inappropriate. For example, for a vulnerability enabling data breaches or ransomware attacks, if multiple attackers manage to exploit the vulnerability, this leads to higher damage, such as multiple fines or multiple ransom payments.

As in Case 1, we assume that attackers stochastically succeed in exploiting the vulnerability. When an attacker succeeds in exploiting the vulnerability, this incurs a one-off damage. As time goes by, more and more attackers succeed in exploiting the vulnerability, and – in contrast to Case 1 – this leads to additional damage. The idea of this case is sketched in Fig. 6.

Formalization. Assuming that successful attacks form a Poisson process, the probability that exactly n attackers managed to exploit vulnerability v within time T is given as $\frac{(\lambda(v) \cdot T)^n}{n!} \cdot e^{-\lambda(v) \cdot T}$. We assume that the damage from multiple successful attacks is additive, as in the above examples of multiple fines or multiple ransom payments. If n attackers managed to exploit vulnerability v, the resulting damage is $n \cdot D(v)$. Thus, the expected damage in the time interval $[0, T)$ is

$$\begin{aligned} R_T(v) &= \sum_{n=0}^{\infty} \frac{(\lambda(v) \cdot T)^n}{n!} \cdot e^{-\lambda(v) \cdot T} \cdot n \cdot D(v) = \\ &= e^{-\lambda(v) \cdot T} \cdot D(v) \cdot \sum_{n=1}^{\infty} \frac{(\lambda(v) \cdot T)^n}{(n-1)!} = \\ &= e^{-\lambda(v) \cdot T} \cdot D(v) \cdot \lambda(v) \cdot T \cdot \sum_{n=1}^{\infty} \frac{(\lambda(v) \cdot T)^{n-1}}{(n-1)!} = \\ &= e^{-\lambda(v) \cdot T} \cdot D(v) \cdot \lambda(v) \cdot T \cdot e^{\lambda(v) \cdot T} = \\ &= D(v) \cdot \lambda(v) \cdot T. \end{aligned} \tag{16}$$

The risk value for the *detailed risk analysis*, where also a risk mitigation action m is available, can be obtained as follows. Analogously to Equation (16), we have

$$\mathbb{E}[\xi_{v,t}] = D(v) \cdot \lambda(v) \cdot t. \tag{17}$$

From this, we get

$$\frac{d}{dt}\mathbb{E}[\xi_{v,t}] = D(v) \cdot \lambda(v), \tag{18}$$

and thus

$$R(v, m) = \left.\frac{d}{dt}\mathbb{E}[\xi_{v,t}]\right|_{t=\tau(m)} = D(v) \cdot \lambda(v). \tag{19}$$

Parameters. In this case, exactly the same parameters are used as in Case 1: $D(v)$ and $\lambda(v)$ for each vulnerability, and the look-ahead horizon T as a global parameter.

Application. A misconfiguration of a database management system may lead to data breaches. On average, an attacker would need 40 hours to exploit this misconfiguration and gain access to sensitive data. Such a data breach would lead to a fine of 40,000 USD, and multiple breaches would lead to multiple fines of the same amount. Also a second vulnerability is discovered, which could also lead to a data breach. On average, an attacker would need 20 hours to exploit this second vulnerability and gain access to sensitive data. Such a data breach would lead to a fine of 30,000 USD, and multiple breaches would lead to multiple fines of the same amount. Again, the question is which of the two risks to prioritize over the other.

We have $\lambda(v_1) = 0.025$ hour^{-1}, $\lambda(v_2) = 0.05$ hour^{-1}, $D(v_1) = 40000\ \$$, and $D(v_2) = 30000\ \$$. Using Equation (16) and 24 hours as look-ahead horizon, we get

$$R_T(v_1) = 40000\ \$ \cdot 0.025\ \text{hour}^{-1} \cdot 24\ \text{hour} = 24000\ \$$$

and

$$R_T(v_2) = 30000\ \$ \cdot 0.05\ \text{hour}^{-1} \cdot 24\ \text{hour} = 36000\ \$.$$

Thus, the second vulnerability should be prioritized.

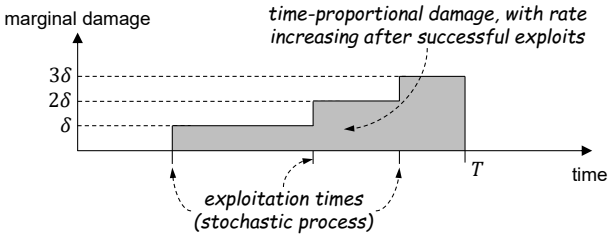

Fig. 7. Case 4: each successful exploit of the same vulnerability increases marginal damage by the same amount δ

E. Case 4: multiple exploitation, time-proportional damage

Intuitive description. This case combines features of Case 2 and Case 3. Like in Case 2, we use time-proportional damage: when an attacker succeeds in exploiting the vulnerability, this incurs a damage from that moment, with the given marginal damage associated with the vulnerability. Like in Case 3, we assume multiple exploitation: as time goes by, more and more attackers succeed in exploiting the vulnerability, and this leads to increased marginal damage. The idea behind this case is sketched in Fig. 7.

Formalization. The mathematical treatment of this case is slightly more complicated. As in Case 2, we need to integrate over time, and as in Case 3, we need to sum over the possible numbers of successful exploits.

Using again the assumption that successful attacks form a Poisson process, the probability that exactly n attackers managed to exploit vulnerability v within time t is given as $\frac{(\lambda(v)\cdot t)^n}{n!}\cdot e^{-\lambda(v)\cdot t}$. If exactly n attackers managed to exploit vulnerability v, the resulting marginal damage is $n\cdot\delta(v)$. Thus, the expected damage in an infinitesimal interval $[t, t+\Delta t)$ is

$$\begin{aligned}
&\sum_{n=0}^{\infty}\frac{(\lambda(v)\cdot t)^n}{n!}\cdot e^{-\lambda(v)\cdot t}\cdot n\cdot\delta(v)\cdot\Delta t = \\
&= e^{-\lambda(v)\cdot t}\cdot\delta(v)\cdot\Delta t\cdot\sum_{n=1}^{\infty}\frac{(\lambda(v)\cdot t)^n}{(n-1)!} = \\
&= e^{-\lambda(v)\cdot t}\cdot\delta(v)\cdot\Delta t\cdot\lambda(v)\cdot t\cdot\sum_{n=1}^{\infty}\frac{(\lambda(v)\cdot t)^{n-1}}{(n-1)!} = \\
&= e^{-\lambda(v)\cdot t}\cdot\delta(v)\cdot\Delta t\cdot\lambda(v)\cdot t\cdot e^{\lambda(v)\cdot t} = \\
&= \delta(v)\cdot\Delta t\cdot\lambda(v)\cdot t.
\end{aligned} \tag{20}$$

The risk value can be computed as the total expected damage in the $[0, T)$ time interval:

$$\begin{aligned}
R_T(v) &= \int_{t=0}^{T}\delta(v)\cdot\lambda(v)\cdot t\;dt = \delta(v)\cdot\lambda(v)\cdot\int_{t=0}^{T}t\;dt = \\
&= \delta(v)\cdot\lambda(v)\cdot\frac{T^2}{2}.
\end{aligned} \tag{21}$$

The risk value for the *detailed risk analysis*, where also a risk mitigation action m is available, can be obtained as follows. Analogously to Equation (21), we have

$$\mathbb{E}[\xi_{v,t}] = \frac{1}{2}\cdot\delta(v)\cdot\lambda(v)\cdot t^2. \tag{22}$$

From this, we get

$$\frac{d}{dt}\mathbb{E}[\xi_{v,t}] = \delta(v)\cdot\lambda(v)\cdot t, \tag{23}$$

and thus

$$R(v,m) = \left.\frac{d}{dt}\mathbb{E}[\xi_{v,t}]\right|_{t=\tau(m)} = \delta(v)\cdot\lambda(v)\cdot\tau(m). \tag{24}$$

Parameters. The same parameters are used as in previous cases: the marginal damage $\delta(v)$ and the Poisson rate parameter $\lambda(v)$ for every vulnerability, and the look-ahead horizon T as global parameter.

Application. A vulnerability in the e-mail server may allow attackers to abuse the mail server for relaying masses of spam messages. Another vulnerability in a dispatcher of computational workflows may allow attackers to abuse the compute resources of the organization for mining cryptocurrencies. In both cases, a successful attacker would want to use the compromised resources for as long as possible. Thus, a successful attacker will keep their success secret, and will also not overuse the resources in order not to get caught. Therefore, other attackers may exploit the same vulnerability, creating additional malicious load on the organization's resources.

If both vulnerabilities are uncovered, the question is: which one to address first? Both vulnerabilities have the properties that (i) it takes some – unknown – time for attackers to exploit the vulnerability, (ii) the created damage is proportional to the time the attacker has access to the exploited asset, and (iii) the expected damage is proportional to the number of attackers that successfully exploited the vulnerability. Neither the original Equation (1), nor any of the previously introduced formulas can cope with all of these properties.

Equation (21) helps solve this problem. Assume that the expected time for an attacker to exploit the first vulnerability is 10 hours (thus, $\lambda(v_1) = 0.1$ hour^{-1}), while it is 20 hours for the second vulnerability (thus, $\lambda(v_2) = 0.05$ hour^{-1}). Moreover, assume that the first vulnerability, if exploited, leads to a marginal damage of $\delta(v_1) = 1\ \frac{\$}{\text{hour}}$, while the second leads to a marginal damage of $\delta(v_2) = 10\ \frac{\$}{\text{hour}}$. Using again $T = 24$ hour, Equation (21) yields risk values

$$R_T(v_1) = 1\ \frac{\$}{\text{hour}}\cdot 0.1\ \text{hour}^{-1}\cdot(24\ \text{hour})^2/2 = 28.8\ \$$$

and

$$R_T(v_2) = 10\ \frac{\$}{\text{hour}}\cdot 0.05\ \text{hour}^{-1}\cdot(24\ \text{hour})^2/2 = 144\ \$.$$

Therefore, the second vulnerability should be addressed first.

F. Case study

In each of the Subsections IV-B–IV-E, we introduced two example vulnerabilities as an application example, and calculated $R_T(v)$ for both. These are summarized in the first 7 columns of Table III. We give each of these vulnerabilities an identifier: e.g., $v_{1.2}$ is the second example in Case 1.

Now, we present a more comprehensive case study using these vulnerabilities. It is Monday morning, and the IT department of company XYZ is starting their daily operation

TABLE III
OVERVIEW OF THE VULNERABILITIES IN THE CASE STUDY

Case	Section	Vulnerability	$\lambda(v)$ $\left[\frac{1}{\text{hour}}\right]$	$D(v)$ [\$]	$\delta(v)$ $\left[\frac{\$}{\text{hour}}\right]$	$R_T(v)$ [\$]	$\tau(m)$ [hour]	$R(v,m)$ $\left[\frac{\$}{\text{hour}}\right]$
1	IV-B	$v_{1.1}$	0.1	10000		9093		
1	IV-B	$v_{1.2}$	0.025	50000		22559	0.5	1235
2	IV-C	$v_{2.1}$	0.1		1000	14907	5	393
2	IV-C	$v_{2.2}$	0.025		1500	8929		
3	IV-D	$v_{3.1}$	0.025	40000		24000	2	1000
3	IV-D	$v_{3.2}$	0.05	30000		36000	1	1500
4	IV-E	$v_{4.1}$	0.1		1	28.8		
4	IV-E	$v_{4.2}$	0.05		10	144		

by going through the real-time risk management process of Fig. 2(b). In the first step (risk identification), they use the results of automated vulnerability scanning tools, penetration testing, and threat intelligence to identify the set of relevant cybersecurity risks. These are the vulnerabilities listed in Table III and described in the relevant subsections referenced in the second column of the table.

In the second step of the real-time risk management process, initial risk analysis is carried out. This entails, for each of the 8 identified vulnerabilities, the following sub-steps:

1) Deciding which of the four cases the vulnerability belongs to. For example, exploiting vulnerability $v_{1.1}$ (which is, as described in Section IV-B, a buffer overflow attack that could potentially crash Asset 1 of the company) would lead to a one-off damage, with no additional damage stemming from subsequent exploitations; thus, this vulnerability belongs to Case 1.
2) Estimating the relevant parameters, depending on the case. $\lambda(v)$ is relevant in all cases; $D(v)$ is relevant in Cases 1 and 3, $\delta(v)$ is relevant in Cases 2 and 4. In the example of $v_{1.1}$, the security experts of XYZ estimate that an attacker needs on average about 10 hours to exploit the vulnerability, and thus, $\lambda(v_{1.1}) = 0.1$ hour^{-1}, while the expected damage is $D(v_{1.1}) = 10000$ \$.
3) Calculating the risk value $R_T(v)$ using the formula corresponding to the given case. In the example of $v_{1.1}$, this means Equation (7), and the calculation results in $R_T(v_{1.1}) \approx 9093$ \$.

At this point, the first 7 columns of Table III are filled. The calculated risk values ($R_T(v)$) are the basis for a first prioritization. For this, it is important that all risk values were calculated with the same look-ahead horizon T (24 hours in our scenario), as this makes them directly comparable with each other. The IT department of XYZ has capacity for the detailed analysis of up to four risks in parallel, so the four vulnerabilities with the highest $R_T(v)$ value are selected. These are $v_{3.2}$, $v_{3.1}$, $v_{1.2}$, and $v_{2.1}$.

In the third step of the real-time risk management process, the experts of XYZ analyze the four prioritized risks in more detail to find suitable countermeasures against them and to estimate the amount of time needed to put these countermeasures in place. In our case study, they find that vulnerability $v_{3.2}$ can be mitigated by applying a patch to the database management system. The patch can be downloaded, tested, and installed in an hour ($\tau(m_{3.2}) = 1$ hour). For mitigating vulnerability $v_{3.1}$, a reconfiguration of the database management system is necessary. This requires a backup and additional testing, taking approximately two hours ($\tau(m_{3.1}) = 2$ hour). Vulnerability $v_{1.2}$ can be mitigated by an appropriate change of firewall rules, which can be put in place in half an hour ($\tau(m_{1.2}) = 0.5$ hour). For mitigating vulnerability $v_{2.1}$, an upgrade of the operating system of the affected server is needed. Because of dependencies, this implicates an upgrade of several other applications, requiring altogether approximately 5 hours ($\tau(m_{2.1}) = 5$ hour). The time needed to implement countermeasures for the selected four vulnerabilities is shown in the penultimate column of Table III.

In the fourth step of the real-time risk management process, the risk values of the four selected vulnerabilities are calculated again, this time using the formula for detailed risk analysis, taking into account the duration of mitigation actions. For vulnerabilities $v_{3.2}$ and $v_{3.1}$, which belong to Case 3, this entails using Equation (19). For vulnerability $v_{3.2}$, we get

$$R(v,m) = D \cdot \lambda = 30000\ \$ \cdot 0.05\ \text{hour}^{-1} = 1500\ \frac{\$}{\text{hour}}.$$

For vulnerability $v_{3.1}$, we get

$$R(v,m) = D \cdot \lambda = 40000\ \$ \cdot 0.025\ \text{hour}^{-1} = 1000\ \frac{\$}{\text{hour}}.$$

For vulnerability $v_{1.2}$, which belongs to Case 1, Equation (10) is to be used, yielding

$$R(v,m) = D \cdot \lambda \cdot e^{-\lambda\tau} = \\ = 50000\ \$ \cdot 0.025\ \text{hour}^{-1} \cdot e^{-0.025 \cdot 0.5} \approx 1235\ \frac{\$}{\text{hour}}.$$

For vulnerability $v_{2.1}$, which belongs to Case 2, Equation (15) is to be used, yielding

$$R(v,m) = \delta \cdot \left(1 - e^{-\lambda\tau}\right) = \\ = 1000\ \frac{\$}{\text{hour}} \cdot \left(1 - e^{-0.1 \cdot 5}\right) \approx 393\ \frac{\$}{\text{hour}}.$$

These risk values are summarized in the last column of Table III. Based on these risk values, the order of the vulnerabilities according to their priority is: $v_{3.2}$, $v_{1.2}$, $v_{3.1}$, $v_{2.1}$. This is the order in which XYZ implements the identified countermeasures. This order is similar to the one after initial risk analysis, but not exactly the same, since $v_{1.2}$ and $v_{3.1}$ are now swapped.

V. DISCUSSION

This section discusses further details of the proposed model and its variants. We start with an overview of the results, followed by an analysis of how the risk value depends on various parameters. Finally, we discuss limitations and potential generalizations or modifications to the proposed approach.

TABLE IV
OVERVIEW OF THE FORMULAS FOR THE RISK VALUE IN THE DIFFERENT CASES. FOR THE PARAMETERS, SEE TABLE I.

Case	Section	Multiple exploitation	Time-propor. damage	$R_T(v)$	$R(v,m)$
1	IV-B	N	N	$\left(1-e^{-\lambda(v)\cdot T}\right)\cdot D(v)$	$D(v)\cdot\lambda(v)\cdot e^{-\lambda(v)\cdot\tau(m)}$
2	IV-C	N	Y	$\delta(v)\cdot\left(T+\frac{e^{-\lambda(v)\cdot T}-1}{\lambda(v)}\right)$	$\delta(v)\cdot\left(1-e^{-\lambda(v)\cdot\tau(m)}\right)$
3	IV-D	Y	N	$D(v)\cdot\lambda(v)\cdot T$	$D(v)\cdot\lambda(v)$
4	IV-E	Y	Y	$\frac{1}{2}\cdot\delta(v)\cdot\lambda(v)\cdot T^2$	$\delta(v)\cdot\lambda(v)\cdot\tau(m)$

A. Overview of the results

In Section IV, we presented four different cases for quantifying risks. Each case leads to two formulas: one for initial risk analysis (without considering mitigation actions) and another for detailed risk analysis (with mitigation actions). Table IV gives an overview about these formulas.

Each of the four cases has its own applicability area:

- Case 1 is applicable if the first successful exploitation of the vulnerability leads to a one-off damage, and further exploitations do not lead to additional damage. Example: a vulnerability leading to a crash of a system.
- Case 2 is applicable if the damage depends on the time that the misbehavior persists, but not on the number of successful exploits. Example: service unavailability due to a denial-of-service attack.
- Case 3 is applicable if each successful exploitation of the vulnerability leads to a one-off damage. Example: a vulnerability leading to a data breach, resulting in a separate fine for each data breach.
- Case 4 is applicable if the damage depends not only on the duration of the misbehavior but also on the number of successful exploits. Example: abuse of resources for cryptocurrency mining.

B. Linear influence of parameters on the risk value

It is interesting to observe how the risk values depend on the parameters. In the traditional model of Equation (1), the risk value depends linearly on both the probability and the impact of a successful exploit. This linear dependence is retained in several of the new formulas as well, although the role of $P(v)$ is taken by $\lambda(v)$ and the role of $D(v)$ is sometimes taken by $\delta(v)$. Specifically:

- In Case 1, both $R_T(v)$ and $R(v,m)$ depend linearly on $D(v)$.
- In Case 2, both $R_T(v)$ and $R(v,m)$ depend linearly on $\delta(v)$.

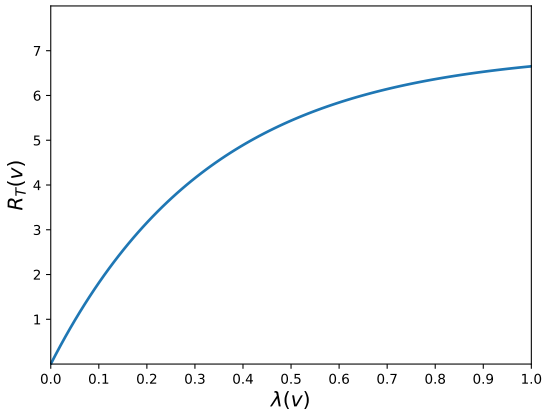

Fig. 8. Case 1, initial risk analysis: dependence of the risk value $R_T(v)$ on $\lambda(v)$, for $T=3$ and $D(v)=7$

- In Case 3, both $R_T(v)$ and $R(v,m)$ depend linearly on both $\lambda(v)$ and $D(v)$.
- In Case 4, both $R_T(v)$ and $R(v,m)$ depend linearly on both $\lambda(v)$ and $\delta(v)$.

Thus, we have a linear dependence on $D(v)$ or $\delta(v)$ in each formula. The dependence on $\lambda(v)$ is also linear in the formulas of Cases 3 and 4, but not in Cases 1 and 2.

It should be noted though that the way $\lambda(v)$ replaces $P(v)$ is not the same as how $\delta(v)$ replaces $D(v)$. This is because $D(v)$ is proportional to $\delta(v)$, whereas $P(v)$ is not proportional to $\lambda(v)$.

C. Non-linear dependence on $\lambda(v)$

In those formulas where the dependence is more complicated than a linear relationship, it is interesting to investigate whether the risk value is at least monotonously increasing in its parameters. We can answer this question by looking at the partial derivative of the risk value function. For the sake of simplicity, we omit the dependence on v and m.

In Case 1, in the formula for initial risk analysis, we have

$$\frac{\partial R}{\partial \lambda} = D\cdot T\cdot e^{-T\cdot\lambda} > 0, \tag{25}$$

thus, the risk value is monotonously increasing in λ. Fig. 8 shows an example for the risk value's dependence on λ.

Still in Case 1, for the formula for detailed risk analysis, the result is quite different:

$$\begin{aligned}\frac{\partial R}{\partial \lambda} &= D\cdot e^{-\lambda\cdot\tau} + D\cdot\lambda\cdot(-\tau)\cdot e^{-\lambda\cdot\tau} = \\ &= D\cdot e^{-\lambda\cdot\tau}\cdot(1-\lambda\cdot\tau).\end{aligned} \tag{26}$$

This expression is positive if and only if $\lambda\cdot\tau<1$. Thus, the risk value increases in λ as long as $\lambda<1/\tau$, reaches its maximum at $\lambda=1/\tau$, and decreases afterwards (see Fig. 9 for an example). It may seem counter-intuitive at first sight that the risk value may decrease with increasing λ. However, this does make sense in light of how risk values are defined in the detailed risk analysis phase: they correlate with the rate of increase in expected damage resulting from delaying the

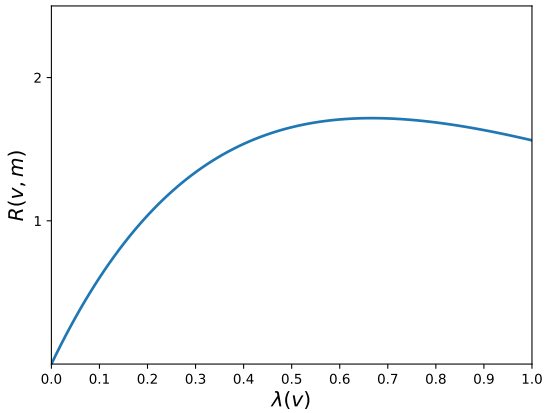

Fig. 9. Case 1, detailed risk analysis: dependence of the risk value $R(v, m)$ on $\lambda(v)$, for $\tau(m) = 1.5$ and $D(v) = 7$

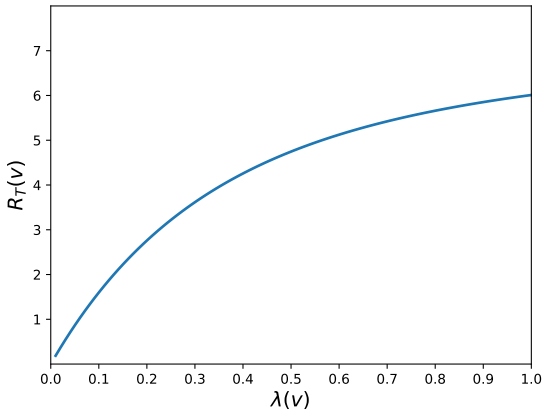

Fig. 10. Case 2, initial risk analysis: dependence of the risk value $R_T(v)$ on $\lambda(v)$, for $\delta(v) = 1.5$ and $T = 5.0$

implementation of mitigation actions. Specifically in Case 1, if $\tau \geq 1/\lambda$, this means that the mitigation takes more time than what attackers need on average to exploit the vulnerability. If λ is further increased, this means that it becomes even more probable that the vulnerability has already been exploited before the countermeasure would be in place, even if implementing the countermeasure is started immediately. Since Case 1 is associated with a one-off damage, such a countermeasure will be largely ineffective. Thus, delaying the implementation of the countermeasure does not increase the expected damage significantly anymore, and if λ increases further, the increase in expected damage resulting from delaying the countermeasure becomes even more insignificant. Note that this only applies to Case 1, where the damage is created at a single point in time, making later countermeasures pointless. In the other cases, further damage is created either by upholding the misbehavior or by subsequent repeated exploitation of the same vulnerability, thus making also late mitigations useful.

In Case 2, in the formula for initial risk analysis, we have

$$\begin{aligned} \frac{\partial R}{\partial \lambda} &= \delta \cdot \frac{-T \cdot e^{-\lambda \cdot T} \cdot \lambda - (e^{-\lambda \cdot T} - 1)}{\lambda^2} \\ &= \frac{\delta}{\lambda^2} \cdot \left(1 - (\lambda \cdot T + 1) \cdot e^{-\lambda \cdot T}\right). \end{aligned} \tag{27}$$

It is well known that for any $x < 1$, $e^x < \frac{1}{1-x}$. Using this inequality for $x = -\lambda \cdot T$, we get

$$\frac{\partial R}{\partial \lambda} > \frac{\delta}{\lambda^2} \cdot \left(1 - (\lambda \cdot T + 1) \cdot \frac{1}{1 + \lambda \cdot T}\right) = 0. \tag{28}$$

Thus, the risk value is monotonously increasing in λ. Fig. 10 shows an example for the dependence of the risk value on λ.

Still in Case 2, for the formula for detailed risk analysis, the derivative is:

$$\frac{\partial R}{\partial \lambda} = \delta \cdot \tau \cdot e^{-\lambda \cdot \tau} > 0, \tag{29}$$

thus, the risk value is monotonously increasing in λ.

Incidentally, the formula for detailed risk analysis in Case 2 has the same form as the formula for initial risk analysis in Case 1, only with δ instead of D and τ instead of T (cf. Table IV). Therefore, the dependence of $R(v, m)$ on $\lambda(v)$ in detailed risk analysis in Case 2 looks the same as the dependence of $R_T(v)$ on $\lambda(v)$ in initial risk assessment in Case 1, which was shown in Fig. 8.

Visually comparing Figures 8 and 10 suggests that the curves are similar. This is only partially true. The formulas in Table IV reveal that the functions are mathematically different: the risk value is an exponential function of λ in Case 1 but not in Case 2 (because of the division by λ). However, the functions share several common properties:

- They are both monotonously increasing, as we have seen.
- They both start at 0. This is obvious for the formula of Case 1. For the formula of Case 2, using $\lim_{x \to 0} \frac{e^x - 1}{x} = 1$, it can be seen easily that $\lim_{\lambda \to 0} R_T(v) = 0$.
- They both converge to some finite limit as $\lambda \to \infty$. As can be easily seen from the formulae, this limit is $D(v)$ in the first case and $\delta(v) \cdot T$ in the second case.

D. Dependence on T

In the initial risk analysis, every vulnerability's risk is assessed using the same look-ahead horizon T to make them comparable. However, the choice of the value of T is arbitrary. Looking at the penultimate column of Table IV, we can establish that $R_T(v)$ is in each case strictly monotonously increasing in T, but otherwise the dependence of $R_T(v)$ on T is quite different in each case. Thus, it could happen that for two vulnerabilities v_1 and v_2 that belong to different cases, v_1 has higher priority than v_2 for one value of T, while v_2 has higher priority than v_1 for another value of T. In Cases 1 and 2, this could even happen if the two vulnerabilities belong to the same case but have different values of λ. (However, this is not possible in Cases 3 and 4, because in these cases T influences the risk value of each vulnerability homogeneously.)

This is an intrinsic difficulty of real-time risk management. A "short-sighted" risk management approach (i.e., a small value of T) prioritizes the handling of immediate threats, which can be very useful, but might potentially pay insufficient attention to long-term threats. On the other hand, an overly future-focused risk management (large value of T) may not

react quickly enough to immediate threats. One possibility to resolve this conundrum is to calculate risk values for multiple values of T, and selecting those vulnerabilities for further processing that exhibit high risk for at least one value of T. Finding a sound way for establishing the best values for T could be an area for future research.

E. Dependence on $\tau(m)$

It is interesting to observe in the last column of Table IV how the formulae for detailed risk analysis in the different cases differ in terms of the dependence of $R(v, m)$ on $\tau(m)$.

In Case 1, $R(v, m)$ is strictly monotonously decreasing in $\tau(m)$. This is related to the already mentioned specialty of Case 1: since damage occurs only once, mitigation actions with a large lead time make little sense, since the damage was likely already incurred before the mitigation action would show effect. Hence, in this case, mitigation actions with a large lead time receive lower priority than mitigation actions requiring less lead time.

In Cases 2 and 4, where damage is proportional to the time the misbehavior persists, $R(v, m)$ is strictly monotonously increasing in $\tau(m)$. This is logical because, in these cases, further delaying mitigation actions that have a long lead time anyway would lead to a large increase in expected damage.

In Case 3, $R(v, m)$ does not depend on $\tau(m)$. In this case, in contrast to Case 1, damage is incurred over and over again; thus, even mitigation actions with a long lead time are important. However, in contrast to Cases 2 and 4, less damage is created over time in this case; thus, further delaying mitigation actions with a long lead time is less dangerous.

F. Further possibilities

This paper is a first step toward a theory of urgency in cybersecurity risk management. Naturally, the proposed approach has some limitations, and addressing these limitations could be the subject of future research. Here, we would like to highlight some possibilities for the further generalization, formalization, and evaluation of the presented model, as well as discuss potential alternative approaches.

Stochastic processes for modeling successful attacks. In our calculations, we made the assumption that successful exploitations of a vulnerability follow a Poisson process. This assumption is in line with previous research [24], [25], [26]. Nevertheless, our methodology is not limited to Poisson processes. Other, possibly more general, families of stochastic processes could also be considered, leading to different formulae for the individual cases.

Formal proof of appropriateness. For evaluating the proposed model, it would be useful if we could formally prove its appropriateness. However, this does not seem to be fully possible. The appropriateness of a model for cybersecurity risk management depends on at least the following factors:

- The model is relatable to real cybersecurity threats.
- The inputs to the model can be easily and accurately determined.
- The model can be quickly evaluated.
- The model's output leads to decisions that reduce cybersecurity risks as much as possible.

The last point seems amenable to formal reasoning. For example, game theory can be used as a formal framework to define cybersecurity as a game between a defender and one or more attackers, and to define an objective function for the defender [27]. In such a game-theoretic setting, the defender could use the formulae presented in this paper to prioritize risks. Mathematical analysis, automated methods like model checking, or simulation could then be used to derive results about the defender's performance. From these results, conclusions could be drawn regarding the usefulness of the presented formulae in reducing cybersecurity risks.

The first two points of the above list, however, do not seem amenable to formal reasoning. For determining the adequacy of our model in these respects, field studies in different organizations would be needed.

More sophisticated model of risk mitigation. This paper uses a rather simple model of countermeasures: putting a countermeasure in place takes a given time $\tau(m)$, during which the countermeasure has no effect, and after $\tau(m)$ time the vulnerability ceases to exist. Also, the time to put the countermeasure in place is the only attribute of the countermeasure that is considered. In reality, risk mitigation could follow more sophisticated strategies. For example, risk mitigation may start with a quick fix that makes it harder but not impossible for attackers to exploit the given vulnerability, followed by a more time-consuming countermeasure to completely eliminate the vulnerability. Also, the effect of countermeasures on, for example, costs could be taken into account [13], [28]. Extending the proposed approach to more sophisticated models of risk mitigation is a topic for future research.

Risk prioritization as global optimization. A fundamental underlying assumption of this work was that each risk should be assigned a risk value specifying its priority. This makes the actual prioritization trivial, as prioritization only entails choosing the risks with highest risk value. However, accurately capturing all important aspects of a security risk with a single number is intrinsically difficult, and might not be fully possible. While perhaps suitable for non-real-time risk management, this approach might not be able to fully capture the complexity of real-time risk management. As we have seen, the method and model proposed in this paper entail some arbitrary choices (the choice of the look-ahead horizon) and approximations (using the derivative to approximate future development) and such inaccuracies may be intrinsically necessary if priority of a risk must be captured as a single number.

A completely different approach could look at the problem of risk prioritization as a global optimization problem. Instead of assigning a risk value to each risk and then using these numbers to schedule mitigation actions, it would be conceivable to directly optimize the schedule of mitigation actions. Using the model and formulae described in this paper, the overall expected damage of a schedule of risk mitigation actions could be predicted. On this basis, a global optimization method [29]

could be applied to find the schedule that leads to the lowest overall expected damage.

Learning to manage risks. Another, completely different approach could be to use machine learning instead of theoretically-derived formulas. For example, assigning a risk value to a risk with given features could be regarded as a prediction problem that could be solved using machine learning. Alternatively, the whole problem of prioritizing a set of risks, based on the features of those risks, could be regarded as a machine learning problem. Then, different types of machine learning algorithms could be applied to solve the problem. For example, reinforcement learning could be used to learn the best risk management policy using trial and error in a simulated environment.

VI. Related work

Cybersecurity risk management has been intensively studied from different angles. However, most cybersecurity risk management approaches suggested so far are oblivious to time aspects [30]. In this section, we focus on work on cybersecurity risk management that does consider the effect of time in some way. We start with the most relevant related papers and then move on to less relevant ones.

Joh and Malaiya propose an approach for risk assessment based on the concept of the vulnerability lifecycle [21]. A vulnerability can be in different states (e.g., discovered, exploited, patched), which is modeled with a Markov process. This allows for a probabilistic analysis, which can be used to reason about time-dependent exploitation probabilities. This could be comparable to our handling of Case 1 for initial risk analysis, although it is not used for risk prioritization. However, Joh and Malaiya stop at this point, without considering the further time-dependent aspects that we consider in this paper. Also, the approach of Joh and Malaiya assumes many parameters, and in practice, it could be difficult to find the proper values for those parameters. In contrast, we try to keep the number of parameters at the necessary minimum.

Zmiewski et al. propose a method for quantifying data protection risks [31], as part of the run-time data protection risk management framework RADAR [28]. Similarly to our work, the approach of Zmiewski et al. is also based on probabilistic reasoning about the damage created by a misbehavior over time. However, their work is limited to a specific kind of risk: attackers getting unauthorized read access to large amounts of sensitive data. Accordingly, their model is specific to this type of risk. In contrast, our model is much more general, and can be applied to a wide range of cybersecurity risks. In addition, we also consider the duration of countermeasures which is not considered by Zmiewski et al.

Chen et al. address specific types of software vulnerabilities that attackers can exploit to let network nodes fail [32]. Since the damage is based on the downtime of the nodes, the model of Chen et al. shows some similarity with our handling of initial risk analysis in Case 2. Chen et al. use queuing theory to estimate the downtime of nodes. However, Chen et al. do not investigate the time dependence of risk values. Rather, their focus is on correlated failures in multiple nodes.

Khosravi-Farmad et al. suggested using the Temporal metrics of the Common Vulnerability Scoring System (CVSS) in order to make risk assessment more accurate [33]. The CVSS temporal metrics were used and refined in some other papers as well [34], [35]. However, using the CVSS temporal metrics boils down to multiplying the probability of a successful exploit by a given factor. This factor may change over time, but its value has to be supplied by an expert. In contrast, our approach accounts for time-dependent effects automatically.

Wu et al. propose a risk assessment method for cyber-physical systems [36]. Beyond attack success probability and attack consequence, their model includes a third factor called "attack severity". Attack severity depends on the frequency and intensity of attacks. Thus, attack severity also has the potential to capture time-dependence. The authors also measure in a specific scenario how attack severity changes over time. However, their formulas do not explicitly take this time-dependence into account.

Awan et al. investigate temporal variance in security risks in computer networks [37]. However, the mathematical model that they use for risk quantification is only based on a snapshot, without taking into account temporal aspects. Time-dependence is only introduced by measuring parameters at different points in time and evaluating the formulas for risk quantification at different points in time.

Li uses the metaphor of the human immune system to assess security risks [38]. While doing so, an evolution over time is considered as security characteristics change. For example, if the system administrator opens a port, this leads to a change. Also, in the model of Li, the detection mechanisms evolve over time. However, unlike in our approach, the duration of attacks and countermeasures is not considered.

VII. Conclusions

In this paper, we have considered the problem of cybersecurity risk assessment in the context of real-time risk prioritization. We have argued that in this setting, the traditional approach of determining risk values based on occurrence probability and impact of an exploit is insufficient, as it does not take into account the urgency of mitigating a vulnerability. To account for urgency in risk assessment, the time-dependence of risks needs to be considered. For this purpose, the paper introduced a process and a general mathematical model for calculating risk values in initial risk analysis (without accounting for countermeasures) and detailed risk analysis (accounting also for countermeasures). Moreover, the mathematical model is concretized in four different cases, in which we derived specific formulas for risk quantification. The four cases differ in their assumptions on the possible exploitation of vulnerabilities and the resulting damage, and thus can be applied to different types of vulnerabilities.

The investigated cases lead to altogether 8 closed formulae for computing risk values. We have provided a detailed anal-

ysis of the mathematical properties of the resulting formulae (e.g., monotonicity and convergence properties).

Our work lays the foundation for a sound handling of urgency in cybersecurity risk assessment. Next steps should include the application of the proposed method and model in different realistic settings to collect practical experience with using them. Such practical experience may lead to refinements of the proposed model to make it more useful in specific situations. In addition, our approach could be combined with many other directions that have been proposed in risk management and that are orthogonal to the time-related considerations of our work. Also, alternative approaches for real-time security risk management could be investigated, for example using global optimization or machine learning techniques.

References

[1] M. Ahsan, K. E. Nygard, R. Gomes, M. M. Chowdhury, N. Rifat, and J. F. Connolly, "Cybersecurity threats and their mitigation approaches using machine learning—a review," *Journal of Cybersecurity and Privacy*, vol. 2, no. 3, pp. 527–555, 2022.

[2] S. Ward and C. Chapman, "Transforming project risk management into project uncertainty management," *International Journal of Project Management*, vol. 21, no. 2, pp. 97–105, 2003.

[3] S. Amland, "Risk-based testing: Risk analysis fundamentals and metrics for software testing including a financial application case study," *Journal of Systems and Software*, vol. 53, no. 3, pp. 287–295, 2000.

[4] International Organization for Standardization, "ISO/IEC 27001:2022 International Standard," https://www.iso.org/standard/27001, 2022.

[5] National Institute of Standards and Technology, "Framework for improving critical infrastructure cybersecurity, version 1.1," https://doi.org/10.6028/NIST.CSWP.04162018, 2018.

[6] T. R. Peltier, *Information security risk analysis*, 2nd ed. CRC press, 2005.

[7] D. W. Hubbard and R. Seiersen, *How to measure anything in cybersecurity risk*, 2nd ed. John Wiley & Sons, 2023.

[8] G. McGraw, "Managing software security risks," *Computer*, vol. 35, no. 4, pp. 99–101, 2002.

[9] D. Landoll, *The security risk assessment handbook: A complete guide for performing security risk assessments*, 3rd ed. CRC Press, 2021.

[10] F. A. Shaikh and M. Siponen, "Information security risk assessments following cybersecurity breaches: The mediating role of top management attention to cybersecurity," *Computers & Security*, vol. 124, p. 102974, 2023.

[11] R. A. Caralli, J. F. Stevens, L. R. Young, and W. R. Wilson, "Introducing OCTAVE Allegro: Improving the information security risk assessment process," Software Engineering Institute, Carnegie Mellon University, Tech. Rep., 2007.

[12] N. Gol Mohammadi, T. Bandyszak, M. Moffie, X. Chen, T. Weyer, C. Kalogiros, B. Nasser, and M. Surridge, "Maintaining trustworthiness of socio-technical systems at run-time," in *Trust, Privacy, and Security in Digital Business: 11th International Conference (TrustBus 2014)*. Springer, 2014, pp. 1–12.

[13] D. Ayed, E. Jaho, C. Lachner, Z. Á. Mann, R. Seidl, and M. Surridge, "FogProtect: Protecting sensitive data in the computing continuum," in *Advances in Service-Oriented and Cloud Computing: International Workshops of ESOCC 2020*. Springer, 2021, pp. 179–184.

[14] D. Ayed, P.-A. Dragan, E. Félix, Z. Á. Mann, E. Salant, R. Seidl, A. Sidiropoulos, S. Taylor, and R. Vitorino, "Protecting sensitive data in the cloud-to-edge continuum: The FogProtect approach," in *2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. IEEE, 2022, pp. 279–288.

[15] A. Mackenzie and P. Nickerson, *The time trap: The classic book on time management*, 4th ed. Amacom, 2009.

[16] X. Zhang, N. Wuwong, H. Li, and X. Zhang, "Information security risk management framework for the cloud computing environments," in *10th IEEE International Conference on Computer and Information Technology (CIT 2010)*. IEEE, 2010, pp. 1328–1334.

[17] I. D. Sánchez-García, T. S. F. Gilabert, and J. A. Calvo-Manzano, "Countermeasures and their taxonomies for risk treatment in cybersecurity: A systematic mapping review," *Computers & Security*, vol. 128, p. 103170, 2023.

[18] M. Surridge, B. Nasser, X. Chen, A. Chakravarthy, and P. Melas, "Run-time risk management in adaptive ICT systems," in *2013 International Conference on Availability, Reliability and Security*. IEEE, 2013, pp. 102–110.

[19] B. Karabacak and I. Sogukpinar, "ISRAM: information security risk analysis method," *Computers & Security*, vol. 24, no. 2, pp. 147–159, 2005.

[20] R. Bojanc and B. Jerman-Blažič, "An economic modelling approach to information security risk management," *International Journal of Information Management*, vol. 28, no. 5, pp. 413–422, 2008.

[21] H. Joh and Y. K. Malaiya, "A framework for software security risk evaluation using the vulnerability lifecycle and CVSS metrics," in *Proc. International Workshop on Risk and Trust in Extended Enterprises*, 2010, pp. 430–434.

[22] R. H. Frank and B. Bernanke, *Principles of Microeconomics*, 3rd ed. McGraw-Hill, 2006.

[23] C. Keller and Z. Á. Mann, "Towards understanding adaptation latency in self-adaptive systems," in *Service-Oriented Computing–ICSOC 2019 Workshops*. Springer, 2020, pp. 42–53.

[24] M. D. Smith and M. E. Paté-Cornell, "Cyber risk analysis for a smart grid: How smart is smart enough? A multiarmed bandit approach to cyber security investment," *IEEE Transactions on Engineering Management*, vol. 65, no. 3, pp. 434–447, 2018.

[25] Q. A. Al-Haija, "On the security of cyber-physical systems against stochastic cyber-attacks models," in *2021 IEEE International IoT, Electronics and Mechatronics Conference (IEMTRONICS)*. IEEE, 2021.

[26] Y. Li, X. Hu, and P. Zhao, "On the reliability of a voting system under cyber attacks," *Reliability Engineering & System Safety*, vol. 216, p. 107996, 2021.

[27] C. T. Do, N. H. Tran, C. Hong, C. A. Kamhoua, K. A. Kwiat, E. Blasch, S. Ren, N. Pissinou, and S. S. Iyengar, "Game theory for cyber security and privacy," *ACM Computing Surveys*, vol. 50, no. 2, 2017.

[28] Z. Á. Mann, F. Kunz, J. Laufer, J. Bellendorf, A. Metzger, and K. Pohl, "RADAR: Data protection in cloud-based computer systems at run time," *IEEE Access*, vol. 9, pp. 70 816–70 842, 2021.

[29] Z. Á. Mann, *Optimization in computer engineering–Theory and applications*. Scientific Research Publishing, 2011.

[30] P. A. Ralston, J. H. Graham, and J. L. Hieb, "Cyber security risk assessment for SCADA and DCS networks," *ISA Transactions*, vol. 46, no. 4, pp. 583–594, 2007.

[31] S. S. Zmiewski, J. Laufer, and Z. Á. Mann, "Automatic online quantification and prioritization of data protection risks," in *Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES)*, 2022.

[32] P.-y. Chen, G. Kataria, and R. Krishnan, "Correlated failures, diversification, and information security risk management," *MIS Quarterly*, pp. 397–422, 2011.

[33] M. Khosravi-Farmad, R. Rezaee, and A. G. Bafghi, "Considering temporal and environmental characteristics of vulnerabilities in network security risk assessment," in *11th International ISC Conference on Information Security and Cryptology*. IEEE, 2014, pp. 186–191.

[34] J. A. Wang, F. Zhang, and M. Xia, "Temporal metrics for software vulnerabilities," in *Proceedings of the 4th Annual Workshop on Cyber Security and Information Intelligence Research: Developing Strategies to Meet the Cyber Security and Information Intelligence Challenges Ahead*, 2008.

[35] J. Wang, M. Xia, and F. Zhang, "Metrics for information security vulnerabilities," *Journal of Applied Global Research*, vol. 1, no. 1, pp. 48–58, 2008.

[36] W. Wu, R. Kang, and Z. Li, "Risk assessment method for cyber security of cyber physical systems," in *2015 International Conference on Reliability Systems Engineering (ICRSE)*. IEEE, 2015.

[37] M. S. K. Awan, P. Burnap, and O. Rana, "Identifying cyber risk hotspots: A framework for measuring temporal variance in computer network risk," *Computers & Security*, vol. 57, pp. 31–46, 2016.

[38] T. Li, "An immunity based network security risk estimation," *Science in China Series F: Information Sciences*, vol. 48, pp. 557–578, 2005.