User-Centric Security Assessment of Software Configurations: A Case Study

Hamza Ghani, Jesus Luna, Ivaylo Petkov and Neeraj Suri

Technische Universität Darmstadt, Germany
{ghani,jluna,petkov,suri}@deeds.informatik.tu-darmstadt.de

Abstract. Software systems are invariably vulnerable to exploits, thus the need to assess their security in order to quantify the associated risk their usage entails. However, existing vulnerability assessment approaches e.g., vulnerability analyzers, have two major constraints: *(a)* they need the system to be already deployed to perform the analysis and, *(b)* they do not consider the criticality of the system within the business processes of the organization. As a result, many users, in particular small and medium-sized enterprizes are often unaware about assessing the actual *technical and economical impact* of vulnerability exploits in their own organizations, *before* the actual system's deployment. Drawing upon threat modeling techniques (i.e., attack trees), we propose a user-centric methodology to quantitatively perform a software configuration's security assessment based on *(i)* the expected economic impact associated with compromising the system's security goals and, *(ii)* a method to rank available configurations with respect to security. This paper demonstrates the feasibility and usefulness of our approach in a real-world case study based on the Amazon EC2 service. Over 2000 publicly available Amazon Machine Images are analyzed and ranked with respect to a specific business profile, before deployment in the Amazon's Cloud.

Keywords: Cloud Security, Economics of Security, Security Metrics, Security Quantification, Vulnerability Assessment

1 Introduction

The use of information systems has been proliferating along with rapid development of the underlying software elements driving them (e.g., operating systems and commercial off-the-shelf software). However, this rapid development comes at a cost, and in many cases e.g., due to limited time schedules and testing budgets for releasing new products, software is often not rigorously tested with respect to security. This results in security flaws that can be exploited to compromise the confidentiality (C), integrity (I) and availability (A) of the affected software products. These flaws are referred to as *software vulnerabilities* and are collected, quantitatively scored and categorized by a multitude of vulnerability databases (e.g., the National Vulnerability Database NVD [1] or the Open Source Vulnerability Database OSVDB [2]). It is a prevalent practice to assess the security of a software system using software analyzers (e.g., OpenVAS [3] and Nessus [4]), that query databases like NVD to ascertain the vulnerabilities affecting a specific software configuration (cf., Figure 1). Unfortunately, despite their broad usage, this approach has two main drawbacks:

1. Most (if not all) vulnerability analyzers require the *deployed* software system to perform the assessment, therefore resulting in a costly trial-and-error process.
2. Such security assessment does not take into account the *economic impact* of detected vulnerabilities. Therefore, it is common to find inconsistencies e.g., technically critical vulnerabilities that do not have the highest economic impact on the organization [5].

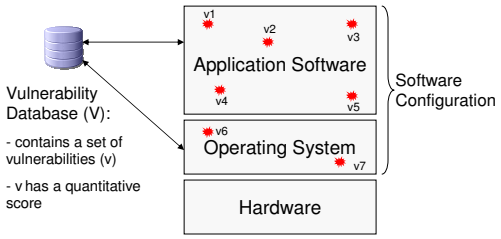

Fig. 1: System Model - Software Configurations and Vulnerabilities (v_i)

Empirical research has shown that the actual impact of vulnerability exploits varies significantly among different types of organizations (in particular the smaller/medium enterprises or SMEs) [6, 7]. Since different organizations perceive the severity of a particular vulnerability differently, they also prioritize its mitigation differently. Existing hypotheses advocate user-centric approaches [8], where the quality and customization of the performed security assessments can be improved if these correlate to the user awareness on the actual impact of a vulnerability in their particular organizational context.

In order to empower users to perform an accurate assessment and ranking of available software configurations *before deploying them*, we propose a methodology to perform the security assessment of a software configuration based on the user's organizational context (expressed in the form of both expected technical and economical impacts). Figure 2 depicts the main stages of our proposed approach, where the specific paper contributions are:

- C1: An approach to elicit the technical metrics required to quantitatively reason about the security goals (C, I, A) of a software configuration, based on the notion of threat modeling and attack trees.
- C2: A systematic approach eliciting the economic-driven factors for weighting the user's security goals, in order to improve the conclusions that can be drawn from the generated attack trees (cf., C1).
- C3: A quantitative technique to rank alternative software configurations using as input the technical (cf., C1) and economical metrics (cf., C2). Our ranking technique is based on the widely used Multiple Criteria Decision Analysis (MCDA) [9, 10].

We demonstrate the feasibility of our approach through a real-world Cloud case study, in which a data set of over 2000 software configurations (publicly

available for users of the Amazon EC2 service) are analyzed and ranked from a security perspective before the actual deployment. The contributed approach aims to enhance the usefulness of widely used security analyzers, by providing users with additional tools that take into account their own organizational contexts.

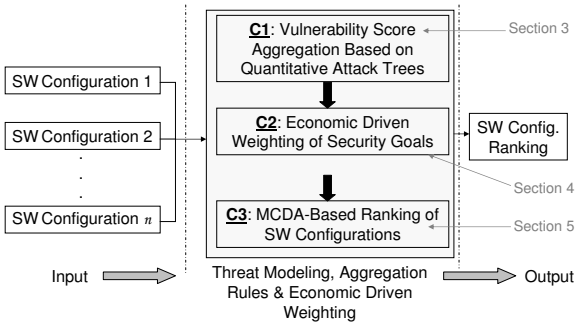

Fig. 2: Overview of the Proposed Approach

The remainder of this paper is organized as follows: Section 2 introduces a motivating case study. Sections 3 – 5 detail the paper contributions as depicted in Figure 2. The results of our evaluation using real world data from Amazon EC2 are shown in Section 6. Section 7 summarizes existing related approaches and Section 8 provides conclusions for the paper.

2 Motivating Case Study: Security-aware Selection of Amazon Machine Images

While the many economic and technological advantages of Cloud computing are apparent, the migration of key business applications onto it has been limited, in part, due to the lack of security assurance on the Cloud Service Provider (CSP). For instance the so-called Infrastructure-as-a-Service (IaaS) Cloud providers allow users to create and *share* virtual images with other users. This is the case of e.g., Amazon EC2 service where users are given the chance to create, instantiate, use and share an Amazon Machine Image (AMI) without the hassle of installing new software themselves. The typical IaaS trust model considers that users trust the CSP (e.g., Amazon), but the trust relationship between the provider of the virtual image –not necessarily Amazon – and the user is not as clear [11].

The basic usage scenario for the Amazon EC2 service requires the user to access the "AWS Management Console" in order to search, select and instantiate the AMI that fulfills her functional requirements (e.g., specific software configuration, price, etc.). Even in simple setups, the security of the chosen AMI (e.g., number and criticality of existing vulnerabilities) remains unknown to the customer *before* its instantiation. Once instantiated, it is the responsibility of the user to assess the security of the running AMI and take the required measures to protect it. However, in a recent paper Balduzzi [11] demonstrated that both the

users and CSPs of public AMIs may be exposed to software vulnerabilities that might result in unauthorized accesses, malware infections, and loss of sensitive information. These security issues raise important questions e.g., is it possible for an Amazon EC2's user to assess the security of an AMI before actually instantiating it? Or, can we provide an Amazon EC2's customer with the AMI that both fulfills the functional requirements and, also represents the smallest security risk for the organization?

3 Vulnerability Score Aggregation Based on Attack Trees

This section presents the first contribution of the proposed assessment methodology (cf., Stage 1 in Figure 2), as an approach to quantify the *aggregated* impact of a set of vulnerabilities associated with a software configuration, based on the notion of *attack trees* [12]. Quantified technical impact and proposed economic metrics (cf. Section 4), will be used as inputs to the MCDA methodology (cf. Section 5) to rank available software configurations.

3.1 Building the Base Attack Pattern

Taking into account that the basic concepts of threat modeling are both well-documented (see Section 7 for more details) and broadly adopted by the industry (e.g., Microsoft's STRIDE threat modeling methodology [13]), the initial stage of the proposed methodology is built utilizing the notion of attack trees. Attack trees, as also used in our paper, are hierarchical representations built by creating nodes that represent the *threats* to the software configuration i.e., the security properties that the attacker seeks to compromise (any of C, I or A). Then one continues adding the *attack* nodes, which are the attacker's strategies to pose a threat to the system (e.g., Denial of Service, SQL injection, etc.). Finally, the attack tree's leaf nodes are populated with the actual software *vulnerabilities* that might be exploited by the attacker to launch an attack. As mentioned in Section 1, software vulnerabilities are associated with a unique identifier and a numeric score similar to those in contemporary databases e.g., NVD [1] and OSVDB [2].

One of the main advantages related with the use of attack trees, is that they allow the creation of "attack tree patterns". The usefulness of attack tree patterns has been documented by the U.S. Department of Homeland Security [14] and, also has been researched in EU projects e.g., SHIELDS [15]. The attack tree built with the basic information presented in this section will be called *base attack pattern* in this paper. Section 6 will introduce a tool we have developed to automatically create attack trees based on the output of the Linux RPM package manager [16].

Base attack patterns can be re-used or even extended by other users to model their own organizational contexts/concerns, therefore taking advantage of the knowledge from the experts that originally created them. For example, our base attack pattern can be further extended with the different elements shown in Figure 3 (i.e., AND nodes[1], composite attacks/threats). The conclusions that can

[1] The AND relationship is only an example option and more complex logical rules can be set up by the user as needed for their applications context.

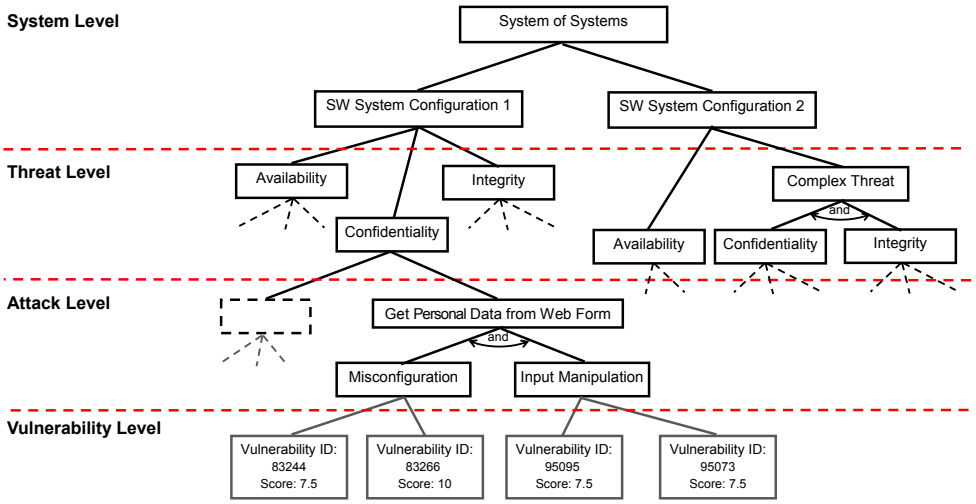

Fig. 3: Extended attack tree pattern.

be drawn from the attack tree shown in Figure 3 (i.e., the aggregated impact of a software configuration's vulnerabilities), can be greatly improved if we provide the techniques to quantitatively reason about the numeric scores associated with each node, as presented in the next section.

3.2 Quantitatively Reasoning about Attack Trees

The proposed rules for aggregating the numeric scores in an "extended attack tree pattern" (cf., Figure 3), requires that every software vulnerability in the tree has a score (similar to NVD [1]). If the vulnerability does not currently have a score, then predictive techniques like VAM [17] can be utilized to propose or predict a value. Based on widely used scoring systems like CVSS [18], we also make the conventional assumption that the provided vulnerability scores are on the interval [0, 10]. The aggregation rules proposed in this paper (cf., Table 1) are recursively applied throughout the attack tree in a bottom-up approach, starting at the vulnerability level and finishing at the threat level (cf., Figure 3). Our proposed aggregation rules are based on previous research in the Privacy-by-Design [19] and Cloud security metrics topics [20], and only need to differentiate the actual relationship among the siblings (i.e., AND/OR). Future work will analyze the effect of aggregating at a higher level of granularity on the attack tree (i.e., the system level). A detailed example on the use of extended attack tree patterns and, designed aggregation rules is presented in Section 6.

4 Economic Driven Weighting of Security Goals

In this section we investigate the trade-offs between security and those economic considerations that play a central role in the proposed methodology.

Relationship	**Aggregation rule for node *N***
AND	$Agg_N = \frac{\sum_{i=1}^{m} N_i}{m}$ where $m = N$'s number of children nodes
OR	$Agg_N = max(N_1 \dots N_m) \times \frac{m}{n}$ where $m = N$'s number of children nodes n = total number of nodes at the same level than N's children ($n \geq m$)

Table 1: Aggregation Rules for Attack Trees

4.1 Including The Economic Perspective

As information systems constitute a mean for helping organizations meet their business objectives, not considering economic aspects when assessing IT security is potentially a major issue (e.g., reputation loss caused by vulnerability exploits). As required by our model, in order to determine the user priorities w.r.t. security goals (i.e., C, I, A) and their relative importance, we propose to use a novel economic driven approach. The rationale is that the potential economic damage to the business caused by a security compromise determines significantly the weight of the security goals. As security goals do not equally influence the core business of the considered organization, they need to be quantitatively weighted following a user-centric approach taking into account the business context specificities. Next, we elaborate on the economic driven damage estimation metrics suitable for weighting an organization's security goals.

4.2 Running Example - Business Profiling

In this section we introduce a calculation model for weighting the security goals based on the notion of "business profile", which refers to the organization's *(i)* economic and *(ii)* data-centric characteristics (as suggested by the authors of [21]). Both set of characteristics, altogether denoted as CH, are the basis for evaluating the weights for the cost categories depicted in Figure 4 and Table 2. In analogy to widely used scoring systems like CVSS [18] and taking into account related works [21], we propose the following eight CH and the corresponding set of qualitative values:

- OS = {Less than 50 < Less than 250 < More than 250 employees}
- SA = {Low < Moderate < High IT dependency}
- CA = {Others < Euro zone < United States}
- SP = {No < Existing < Existing & Monitored (E & M)}
- AT = {less 10M < less 50M < more 50M USD}
- CD = {No < Personal Data < Personal & Financial (P & F)}
- ED = {No < Personal Data < Personal & Financial}
- IP (patents, blueprints, etc.) = {No < Moderate value < High value}

To perform the calculation process these qualitative values in CH will be mapped to quantitative values (e.g., $(\frac{1}{3}; \frac{2}{3}; 1)$ as used in this section). To illustrate our approach, let us consider an example with two companies i.e., *(i)* an SME X, and *(ii)* a large multinational company Y. Both have their respective company profiles depicted in Table 2. Thanks to the proposed approach, whatever

Table 2: Business Profiles: Qualitative/Quantitative Assessment

Characteristics	**SME** X		**Multinational** Y	
	Qualitative	Quantitative	Qualit.	Quant.
Organization Size (OS)	30	$\frac{1}{3}$	5500	1
Sector of Activity (SA)	Manufacturing	$\frac{1}{3}$	Direct Banking	1
Countries of Activity (CA)	Mexico	$\frac{1}{3}$	US, Euro zone	1
Security Policy (SP)	No	$\frac{1}{3}$	E & M	1
Annual Turnover (AT)	3M USD	$\frac{1}{3}$	750M USD	1
Customer Data (CD)	Personal Data	$\frac{2}{3}$	P & F	1
Employee Data (ED)	P & F	1	P & F	1
Intellectual Property (IP)	No	$\frac{1}{3}$	Risk Models	$\frac{2}{3}$

company C can be represented as a tuple $C = (OS, SA, CA, SP, AT, CD, ED, IP)$ containing the quantitative values of the characteristics of C. For instance, the SME X shown in Table 2 can be represented by the tuple $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, 1, \frac{1}{3})$. The notion of business profiles will be utilized as a basis to weight the economic driven metrics to be defined in the next section.

4.3 Economic Driven Approach for Weighting Security Goals

The methodology proposed in this paper requires a set of metrics reflecting the economic impact of potential security incidents, caused by software vulnerability exploits. To define these *Economic Driven Metrics* (EDM), one needs to investigate the expected potential costs of security incidents. The main basis for determining our set of applicable EDM is the work of Innerhofer et al. [22], in which the authors define a set of 91 cost units based on an empirical study on the costs caused by publicly known security incidents. Because in many cases (security) managers are in charge of assessing the economic impact of vulnerabilities, experience has proved that it is more convenient to evaluate higher/aggregated levels of granularity for the potential costs of a security exploit in order to be intuitive and easy to classify. Therefore, we define a small set of higher level main cost classes aggregating the 91 cost units of [22, 23]. We distinguish three levels of granularity regarding the potential costs: the highest level (most detailed) is $L1$ which contains all cost units defined in the state of the art literature [22, 23]. $L2$ aggregates the $L1$ costs into one of the five proposed cost classes (dashed rectangles in Figure 4) with context-dependent weights ((W_{LC}), (W_{RL}), etc.) reflecting the criticality of the corresponding cost (sub)class for the organization. For instance "Legal Costs (LC)" are highly dependent on the country, in which the organization is located, thus need to be weighted differently in diverse legal environments (e.g., the jurisprudence in the USA is completely different than in Germany, developing countries, etc.). The lowest level of granularity $L3$ distinguishes between two main cost classes (i) potential damage/losses, and (ii) ex-post response costs, which could result from a security incident. Figure 4 depicts the overall cost aggregation process. In earlier work, we have described our

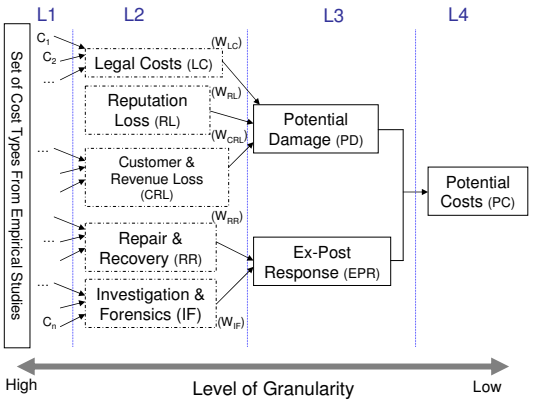

Fig. 4: Aggregation of the Proposed Economic Driven Metrics

Table 3: Mapping proposed scale - monetized scale

Qualitative Scale	**Monetized Scale (USD)**	**Quantitative Scale**
Low	$[0, C_{medium}[$	3.5
Medium	$[C_{medium}, C_{high}[$	6.1
High	$[C_{high}, \infty[$	9.3

methodology for systematically investigating all cost units and the corresponding unified cost classes [24], which constitute the underlying cost data for $L1$ (Figure 4).

Based on the same principles that CVSS [25], we propose to use an intuitive scale of three possible values (i.e., low, medium, high) to evaluate the different metrics of level $L2$ (cf. Figure 4). Furthermore, as monetized metrics have the advantage of (i) allowing easy numerical comparison between alternative scenarios within the same company, and (ii) are directly understandable by managers and executives with less technical affinity, we propose a mapping (cf., Table 3) between our proposed qualitative scale and a company-dependent monetized scale. The rationale is that absolute monetary terms do not allow an objective comparison across companies of different sizes; e.g., a cost of $100K$ EUR might be *critical* for an SME, but of *low* effect for a large multinational company.

Organizations could define their specific interval values c_x for the monetized mapping. For the calculation of our metrics, one needs also quantified factors to be mapped to the proposed scale (cf., Table 3). The quantitative scale thresholds are defined in such a way that, analogous to the CVSS thresholds, the scoring diversity is taken into consideration [26] and the intuitive and widely accepted CVSS scoring scheme is respected. To illustrate the usage of the metrics e.g., for "Reputation Loss" in the case of a *Confidentiality* compromise, the user can qualitatively estimate a value (i.e., low, medium, high) for that specific EDM, and according to the mapping depicted in Table 3, a quantitative value to be utilized for the score calculations is assigned accordingly. To calculate the metric for the overall "Potential Costs" ($L4$) we define the final outcome of calculating PC for *each* security goal (i.e., C, I, A) as depicted in Equation 1 for Confidentiality

Table 4: $L2$ Weights for SME X and Multinational Y

Weights ($L2$)	**SME** X	**Multinational** Y
W_{LC}	$\frac{1}{3}$	1
W_{RL}	$\frac{1}{3}$	1
W_{CRL}	$\frac{4}{9}$	$\frac{8}{9}$
W_{RR}	$\frac{1}{3}$	1
W_{IF}	$\frac{8}{15}$	$\frac{14}{15}$

(similarly for I, A):

$$PC_C = (LC \times W_{LC}) + (RL \times W_{RL}) + (CRL \times W_{CRL}) + (RR \times W_{RR}) + (IF \times W_{IF}) \quad (1)$$

Furthermore, the different weights for $L2$ (Figure 4) needed to compute Equation 1 are calculated as follows, where $Max(X)$ is the maximal possible value for X:

$$W_{LC} = \frac{CA}{Max(CA)} \quad (2)$$

$$W_{RL} = \frac{OS + SA + CA + AT}{Max(OS) + Max(SA) + Max(CA) + Max(AT)} \quad (3)$$

$$W_{CRL} = \frac{AT + CD + IP}{Max(AT) + Max(CD) + Max(IP)} \quad (4)$$

$$W_{RR} = \frac{OS + SP}{Max(OS) + Max(SP)} \quad (5)$$

$$W_{IF} = \frac{OS + SP + CD + ED + IP}{Max(OS) + Max(SP) + Max(CD) + Max(ED) + Max(IP)} \quad (6)$$

To calculate these weights, we utilize the business profiling values defined in Table 2. The weight calculations for SME X and Multinational Y provide the results shown in Table 4. In the next section, we introduce the last stage of our approach consisting of an MCDA-based approach to assess and rank different software configurations, taking as input the outcomes of Sections 3 and 4.

5 MCDA-Based Ranking of Software Configurations

In this section, we present a MCDA-based methodology by which the proposed raking of software configurations can be performed in a systematic way. MCDA methods are concerned with the task of ranking a finite number of alternatives (software configurations in our case), each of which is explicitly described in terms of different characteristics (i.e., the aggregated vulnerability scores from Section 3) and weights (i.e., the economic-driven metrics from Section 4) which have to be taken into account simultaneously. For our research we decided to apply the Multiplicative Analytic Hierarchy Process (MAHP) [9, 27], one of the most widely used and accurate MCDA methodologies nowadays [10]. In the following, the MAHP-background required to comprehend our approach will be briefly presented . For a detailed description of the MCDA methods (including MAHP), we refer to [28].

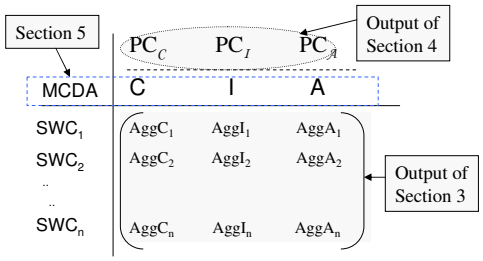

Fig. 5: MAPH-based matrix used by our approach.

At a glance, MAHP starts by building a matrix as shown in Figure 5 in order to perform the ranking. The MAHP matrix requires the aggregated impacts of a set of vulnerabilities associated with a software configuration. Furthermore, the EDMs that have been introduced in Section 4 constitute the weights (PC) of the security goals to take into account (i.e., C, I, A). Once the MAHP matrix is built, we calculate a quantitative score S_{SWC_i} for each software configuration utilizing Equation 7. The value of S_{SWC_i} is directly proportional to the overall impact (technical and economical) associated with the software configuration i.e., a low S_{SWC_i} represents also a low impact for the organization. In the next section, we will experimentally show how thanks to MAHP it is possible to quantitatively rank different Amazon EC2's AMIs configurations from a user-centric perspective.

$$S_{SWC_i} = (AggC_i)^{PC_C} \times (AggI_i)^{PC_I} \times (AggA_i)^{PC_A} \tag{7}$$

6 Evaluation: security ranking of Amazon EC2's AMIs

Further developing the Amazon EC2-based case study introduced in Section 2, performed validation experiments and obtained results are presented next.

6.1 Experimental Setup

Our validation experiments consider a SME user of the Amazon EC2 service (cf., Section 2), who is looking for an available AMI with a LAMP software configuration[2]. Our methodology aims to provide this user with quantitative security insights about alternative AMIs *before instantiating any*. In particular we will take into consideration for the assessment her organizational context (i.e., technical and economical risks).

The proposed methodology was validated using real-world vulnerability data (i.e., Nessus' reports [4]) from more than 2000 Amazon EC2's AMIs, kindly provided for research purposes (i.e., sanitized and anonymized) by Balduzzi et. al. [11]. It is also worth to highlight that this data set covers a period of five months, between November 2010 and May 2011, and as mentioned by Balduzzi [11] the

[2] LAMP stands for the software system consisting of Linux (operating system), Apache HTTP Server, MySQL (database software), and PHP, Perl or Python.

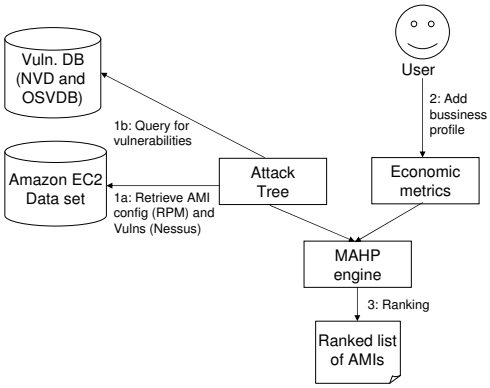

Fig. 6: Test bed used for validation.

Amazon Web Services Security Team already took the appropriate actions to mitigate the detected vulnerabilities.

The implemented test bed (cf., Figure 6) consisted of three main building blocks, namely:

- The "Attack Tree", a Java/MySQL implementation to automatically create "base attack patterns" (cf., Section 3.1) by sequentially extracting both AMI configurations (RPM-like format) and reported vulnerabilities from the data set (Step 1a). OSVDB [2] was queried (Step 1b) to classify found vulnerabilities into corresponding attacks, and then NVD [1] scores were used to compute our "coverage" metric (cf., Section 6.2). Finally, this component also aggregated the values on the resulting attack tree applying the rules presented in Section 3.2.
- The "Economic Metrics" component (web form and back-end database) where the User inputs the information related to her own organizational context (Step 2). This information is processed to create the numeric weights (i.e., PC_C, PC_I and PC_A presented in Section 4.3) required by the ranking module described next.
- The "MAHP engine" implements the MAHP technique described in Section 5, which takes as inputs both the aggregated technical impact (from the "Attack Tree" component) and the economic-driven weights (from the "Economic Metrics" module). The output is an ordered set of AMIs.

For our ranking experiments, we used the two synthetic business profiles shown in Table 2 (i.e., SME X and Multinational Y). At the time of writing this paper we still do not have the information for creating real-world profiles, however as discussed in Section 8 we have started collecting this data via targeted surveys.

6.2 Evaluating the Methodology's Coverage

The goal of this experiment was to validate if the vulnerabilities reported by our approach (cf., Step 1b in Figure 6) were *at least* the same as reported by the

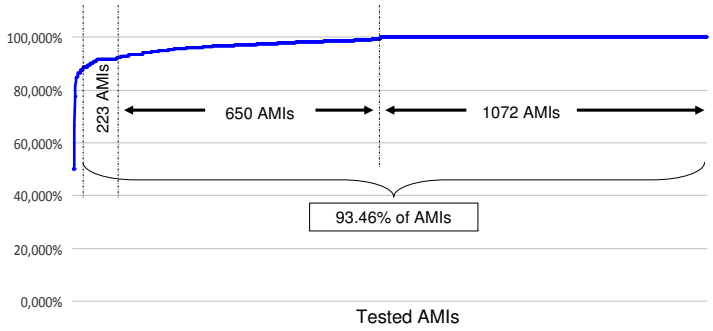

Fig. 7: Vulnerability Detection Coverage in 2081 AMIs

Nessus tool. If that is the case, then we can actually assert the validity of performing the proposed AMI's security assessment *before* instantiation. Obtained coverage results are shown in Figure 7 for all tested 2081 AMIs. A coverage rate of at least 90% was achieved in 93.46% of the AMIs, with a worst case scenario of 65% coverage in only *one* AMI.

One of the main challenging issues facing our current implementation is ensuring 100% coverage. Vulnerabilities are queried from publicly available databases (e.g., NVD [1]) based on a mapping between the actually installed software (RPM-like format [16]) and, the Nessus reported software (using the Common Platform Enumeration or CPE format [29]). Therefore, we cannot claim that our mapping is complete, as it does not contain all existing software packages. Unfortunately, at the state of the art there is no publicly available $RPM \leftrightarrow CPE$ mapping that can be applied for this purpose. So we had to manually check and complement the mapping to run our experiments meaningfully. Such a comprehensive/constantly updated mapping, could allow Amazon EC2 to actually provide its users with a realistic security assessment of existing AMIs (before instantiation).

6.3 Ranking Existing AMIs

During this experiment, we applied the business profiles presented in Table 2 to the data set of 2081 AMIs in order to rank them with the MAHP technique described in Section 5. As required by the MAHP matrix (cf., Figure 5), the base attack pattern for each available AMI was automatically created and populated in order to compute the aggregated impacts Agg_{Ci}, Agg_{Ii} and Agg_{Ai}. For the sake of automation, our experiments did not extend the base attack pattern (e.g., with the use of AND relationships).

Just as expected, ranking results show that for both business profiles (i.e., SME X and Multinational Y) the order of the best suitable AMIs is different. For instance the 2nd best AMI for SME X is ranked 20 for Multinational Y, and the 3rd one for SME is ranked 21st for the latter. Table 5 depicts the rankings of the top 10 AMIs in both scenarios (SME X and Multinational Y). Notice that in both scenarios the best ranked AMI was the same (`ami-fb6e8292`), because this configuration has both the least number of critical vulnerabilities and, relatively

Table 5: Top 10 AMI Rankings: SME X vs. Multinational Y

Rank	Multinational Y	MAHP Score	SME X	MAHP Score
1	*ami-fb6e8292*	0.231468538	*ami-fb6e8292*	0.738087398
2	ami-f857b091	0.3934929	ami-044fa56d	1.002042157
3	ami-43aa432a	0.534172853	ami-2309e44a	1.002042157
4	ami-63aa430a	0.534172853	ami-49c72920	1.002042157
5	ami-665bb00f	0.534172853	ami-6c749e05	1.002042157
6	ami-6743ae0e	0.534172853	ami-8f729fe6	1.002042157
7	ami-7d43ae14	0.534172853	ami-a236c1cb	1.002042157
8	ami-8ff38cdd	0.534172853	ami-43aa432a	1.014506003
9	ami-bb709dd2	0.534172853	ami-63aa430a	1.014506003
10	ami-c224d5ab	0.534172853	ami-665bb00f	1.014506003

low aggregated scores in the corresponding attack tree. The latter also explains the low overall score S_{SWC_i} obtained by the MAHP technique (despite the two different organizational profiles). This quantitative result proves the intuitive notion that a properly secured AMI, can provide an adequate security level to different types of users/organizations.

7 Related Work

Despite the large variety of papers devoted to vulnerability assessment in software systems, there are, to the best of our knowledge, no existing approaches that take account of both technical and economical perspectives in the assessment process. Nevertheless, we present in this section relevant existing research in the field of vulnerability assessment. For the vulnerability assessment approaches from a technical perspective, there exist varied academic and applied approaches to manage and assess vulnerabilities. Projects described in [30–32] define a list of detected vulnerabilities, typically ranked using qualitative assessment such as low, medium, high. These assessment approaches have a qualitative nature and do not consider economic aspects. Mell et. al. [33] propose quantitative metrics for calculating scores reflecting the severity of vulnerabilities based on published vulnerability reports. They introduce CVSS, which is a vulnerability scoring system providing a standardized method for rating vulnerabilities [26]. Our approach can be used to add the necessary contextual dimension to improve the usage and accuracy of CVSS scores (cf., Section 3). This aspect of our approach is important, when considering the existing works suggesting that different organizations evaluate vulnerabilities differently, based on their specific contexts [34, 8, 35, 36]. The authors of [6, 7] showed empirically that the impact of security vulnerability exploits varies with a company's context. The results of [6, 7] constitute a major driver motivating our work.

Like our approach, there exists a separate line of research applying MCDA techniques in security related fields. The authors of [37] utilize MAHP for the security assessment of power control processes. Similarly, another work in the

area of power systems [38] applies an MCDA approach to provide online quantification of a security level associated with an existing or forecasted operating condition of power systems.

In the area of economic driven metrics, the authors of [39] analyzed the incurred costs related to the resolution of individual security incidents within 18 participating US schools. Additionally, there are simple calculators of potential losses such as "Data Breach Risk Calculator" [21] from the Ponemon Institute and Symantec Corporation, and the "Data Loss Cost Calculator" [40]. These calculators provide rough numbers (mostly for illustrative purposes), and their calculation formulas and methodologies are mostly hidden. Another related field concerns cybercrime and its economic impact on the society. Anderson [41] introduces the first systematic study of the costs of cybercrime in the society, as an answer to the the UK Detica report [42]. Clearly, our approach is hence the first vulnerability assessment method that uses both technical and economic driven metrics in the calculation process and aggregates them in a holistic manner, enabling a user centric, pre-deployment assessment of security vulnerabilities of different software configurations.

8 Conclusions

In this paper we presented a methodology to quantitatively assess the security of a software configuration from a user-centric perspective. The proposed approach takes into account the overall organizational context (i.e., technical and economical risks), and does not require the actual software system to be deployed/installed. The proposed approach has been validated using real-world data from Amazon EC2. Vulnerability reports covering a total number of 2081 AMIs has been considered in the evaluation of our approach. The obtained results show that (i) our approach *not requiring a physical system deployment* is able to report *at least* the same vulnerabilities as Nessus [4] (a coverage rate of 93.46% of the tested AMIs); and (ii) given some business profiling data (e.g., turn-over, countries of activity), it is feasible to rank available Amazon EC2's AMIs with respect to security. While a limitation of our approach consists in that the results' accuracy depends on the quality of the available input data, especially business profiling data, our findings suggest that the proposed assessment could be adopted with little effort by IaaS providers, thus empowering their customers to compare different existing configurations and offers from a security level perspective.

Furthermore, we are investigating alternative aggregation rules to be applied on the attack trees, a promising direction consists in utilizing semi-ring operations in order to interpret the "AND" branches.

Acknowledgments

Research supported in part by EC FP7 SPECS, Loewe-CASED and BMBF EC-SPRIDE at TU Darmstadt. The authors would like to thank Marco Balduzzi, Jonas Zaddach, and especially Davide Balzarotti, Engin Kirda and Sergio Loureiro for sharing with us the Amazon data set for our experiments.

References

1. NVD, "National Vulnerability Database," *http://nvd.nist.gov/*, 2013.
2. OSVDB, "The Open Source Vulnerability Database," 2012. [Online]. Available: http://osvdb.org/
3. OpenVAS, "Open Vulnerability Assessment System," Online: http://www.openvas.org/, 2013.
4. Tenable Network Security, "Nessus vulnerability scanner," Online: http://www.tenable.com/products/nessus, 2013.
5. C. Fruehwirth *et al.*, "Improving CVSS-based vulnerability prioritization and response with context," *Proc. of Third International Symposium on Empirical Software Engineering and Measurement*, 2009.
6. M. Ishiguro *et al.*, "The effect of information security incidents on corporate values in the japanese stock market," in *Proc. of International Workshop on the Economics of Securing the Information Infrastructure (WESII)*, 2006.
7. R. Telang *et al.*, "An empirical analysis of the impact of software vulnerability announcements on firm stock price," in *Proc. of IEEE Transactions on Software Engineering*, 2007.
8. Y. Lai *et al.*, "Using the vulnerability information of computer systems to improve the network security," *Computer Communications*, 2007.
9. T. Saaty, *Book: The Analytic Hierarchy Process.* McGraw-Hill, New York, 1980.
10. E. Triantaphyllou *et al.*, "The impact of aggregating benefit and cost criteria in four mcda methods," *IEEE Transactions on Engineering Management*, 2004.
11. M. Balduzzi *et al.*, "A security analysis of Amazon's Elastic Compute Cloud service," in *Proc. of the Annual ACM Symposium on Applied Computing*, 2012.
12. B. Schneier, "Attack trees," *Dr Dobb's*, vol. 24, no. 12, 1999. [Online]. Available: http://www.schneier.com/paper-attacktrees-ddj-ft.html
13. F. Swiderski and W. Snyder, *Book: Threat Modeling.* Microsoft Press, 2004.
14. Department of Homeland Security, "Attack Patterns," Online: https://buildsecurityin.us-cert.gov/, 2009.
15. SHIELDS, "EU FP 7 – SHIELDS project: Detecting known security vulnerabilities from within design and development tools," Online: http://www.shields-project.eu/, 2010.
16. RPM ORG, "The RPM package manager," Online: http://rpm.org/, 2007.
17. H. Ghani *et al.*, "Predictive vulnerability scoring in the context of insufficient information availability," in *Proc. of the Intl. Conference on Risk and Security of Internet and Systems (CRiSIS)*, 2013.
18. Forum of Incident Response and Security Teams, "CVSS – Common Vulnerability Scoring System," Online: http://www.first.org/cvss/, 2012.
19. J. Luna *et al.*, "Privacy-by-design based on quantitative threat modeling," in *Proc. of the Intl. Conference on Risk and Security of Internet and Systems*, 2012.
20. J. Luna *et al.*, "Benchmarking Cloud Security Level Agreements Using Quantitative Policy Trees," in *Proc. of the ACM Cloud Computing Security Workshop*, 2012.
21. Symantec, Ponemon Institute, "Data Breach Calculator," Online: https://databreachcalculator.com, 2013.
22. F. Innerhofer *et al.*, "An empirically derived loss taxonomy based on publicly known security incidents," in *Proc. of Intl. Conf. on Availability, Reliability and Security (ARES)*, 2009.
23. M. Van Eeten *et al.*, "Damages from internet security incidents," *OPTA Research reports*, 2009. [Online]. Available: http://www.opta.nl/nl/actueel/alle-publicaties/publicatie/?id=3083

24. H. Ghani *et al.*, "Quantitative assessment of software vulnerabilities based on economic-driven security metrics," in *Proc. of the Intl. Conference on Risk and Security of Internet and Systems (CRiSIS)*, 2013.
25. Forum of Incident Response and Security Teams, "CVSS Adopters," *http://www.first.org/cvss/eadopters.html.*, 2013.
26. K. Scarfone and P. Mell, "An analysis of CVSS version 2 vulnerability scoring," in *Intl. Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2009.
27. T. Saaty, *Book: Fundamentals of decision making and priority theory with the analytic hierarchy process.* RWS publications Pittsburgh, 1994.
28. M. Zeleny, *Book: Multiple Criteria Decision Making.* McGraw-Hill, 1982.
29. NIST, "CPE – Official Common Platform Enumeration Dictionary," Online: http://nvd.nist.gov/cpe.cfm, 2013.
30. SANS-Institute, "SANS critical vulnerability analysis archive," *http://www.sans.org/newsletters/cva/*, 2007.
31. E. Johnson *et al.*, "Symantec global internet security threat report," *http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xiii_04-2008.en-us.pdf*, 2008.
32. Microsoft, "Microsoft security response center - security bulletin severity rating system," *http://www.microsoft.com/technet/security/bulletin/rating.mspx*, 2007.
33. P. Mell *et al.*, "Common vulnerability scoring system," *in IEEE Security and Privacy*, vol. 4, pp. 85–89, 2006.
34. R. Rieke, "Modelling and analysing network security policies in a given vulnerability setting," *Critical Information Infrastructures Security*, 2006.
35. G. Eschelbeck, "The laws of vulnerabilities: Which security vulnerabilities really matter," *Information Security Technical Report*, 2005.
36. Y. Chen, "Stakeholder value driven threat modeling for off the shelf based systems," 2007.
37. N. Liu *et al.*, "Security assessment for communication networks of power control systems using attack graph and mcdm," *IEEE Transactions on Power Delivery*, 2010.
38. M. Ni *et al.*, "Online risk-based security assessment," *IEEE Transactions on Power Systems*, 2003.
39. V. Rezmierski *et al.*, "Incident cost analysis and modeling project (i-camp)," *Technical Report, Higher Education Information Security Council (HEISC)*, 2000.
40. Allied World Assurance, "Tech404 Data Loss Cost Calculator," Online: http://www.tech-404.com/calculator.html, 2013.
41. R. Anderson *et al.*, "Measuring the cost of cybercrime." in *Proc. of Workshop on the Economics of Information Security (WEIS)*, 2012.
42. Detica and C. Office, "The cost of cyber crime: joint government and industry report," in *Detica report, https://www.gov.uk/government/publications/the-cost-of-cyber-crime-joint-government-and-industry-report*, 2012.