Estimating ToE Risk Level using CVSS

Siv Hilde Houmb and Virginia N. L. Franqueira
Information Systems Group, CTIT, University of Twente
Drienerlolaan 5, 7522 NB Enschede, The Netherlands
Email: {s.h.houmb, franqueirav} (at) ewi.utwente.nl

***Abstract*—Security management is about calculated risk and requires continuous evaluation to ensure cost, time and resource effectiveness. Parts of which is to make future-oriented, cost-benefit investments in security. Security investments must adhere to healthy business principles where both security and financial aspects play an important role. Information on the current and potential risk level is essential to successfully trade-off security and financial aspects.**

Risk level is the combination of the frequency and impact of a potential unwanted event, often referred to as a security threat or misuse. The paper presents a risk level estimation model that derives risk level as a conditional probability over frequency and impact estimates. The frequency and impact estimates are derived from a set of attributes specified in the Common Vulnerability Scoring System (CVSS). The model works on the level of vulnerabilities (just as the CVSS) and is able to compose vulnerabilities into service levels. The service levels define the potential risk levels and are modelled as a Markov process, which are then used to predict the risk level at a particular time.

***Index Terms*—Quantifying security, Operational security, Risk estimation, Calculated risk and CVSS.**

I. Introduction

Modern society relies heavily on networked information systems. The risks associated with these systems might have serious implications, such as threatening the financial and physical well being of people and organizations. E.g., the unavailability of a telemedicine system might result in loss of life and an Internet-based organization can be put out of business as a result of a successful denial of service (DoS) attack. However, security investment must be balanced with potential losses, as an Internet-based organization may very well be put out of business if forced to raise sales prices to compensate for overspending on security measures.

Security management covers all from operational security to enterprise level security strategy, and in particular the relation between these. Often, the cause of a security problem is on the operational level while the impact is on the strategic or business/enterprise level. It then becomes important to distinguish the cause from the consequence events to effectively place security measures. In any case, the goal is to balance security investments with potential losses associated with future security breaches and with the real losses already experienced. We call this to derive at a balanced and controlled risk level and to take calculated risks. However, this cannot be done without insight into the vulnerabilities on the operational level and the impacts that these might have on the strategic level. Without this knowledge it is hard to evaluate the effectiveness of a security measure on the operational level and to build effective security strategies for the enterprise level.

This paper focuses on the operational level and how to relate local operational security flaws to the strategic level, here represented by the ISO 14508 [1] notion Target of Evaluation (ToE). The risk level estimation model is limited to the overall system perspective, called ToE in this paper, and do not discuss the financial aspects involved. Details on the financial aspects and how operational security relates to strategic or enterprise security are in Houmb (2007) [2]. A ToE can be any part of a system/network or the whole system/network and is used to denote the object in need of or being managed. The risk level estimation model focuses on vulnerabilities on the operational level (note that vulnerabilities may also be on the strategic level, such as in the security processes or the way people use a security measure) and uses the Common Vulnerability Scoring System (CVSS) to derive the risk level. The risk level is derived over impact and frequency estimates, which are directly estimated from CVSS information in e.g. the NVD (National Vulnerability Database maintained by NIST). We call the potential undesired events (consequences of vulnerability exploits) for misuse and denote the frequency and impact of these as misuse frequency (MF) and misuse impact (MI). This is to distinguish the operational level from the strategic level.

The CVSS is an effort to provide a universal and vendor-independent score of known vulnerabilities. This initiative is funded by the U.S. Department of Homeland Security and maintained by FIRST[1] (www.first.org). The CVSS score is a decimal number on the scale [0.0,10.0] and is composed of the three metrics groups: base, temporal and environmental [3]. In the ToE risk level estimation model, we extend beyond the current use of these metrics groups to estimate misuse frequency and impacts. The implementation of the model is described in Houmb, Franqueira and Engum (2008) [4].

The remainder of the paper is structured as following. Section II places the ToE risk estimation model into context of related work and outlines the contribution of this paper. Section III introduces the CVSS and describes the three CVSS metrics groups. Section IV describes how to use the CVSS to estimate misuse frequency (MF) and misuse impact (MI). Section V outlines the ToE risk level estimation model and its underlying computational procedure (procedure used to

[1] Forum of Incident Response and Security Teams

derive ToE risk level from MF and MI estimates). Section VI demonstrates how to use the ToE risk level estimation model on an example ToE. Finally, Section VII concludes the paper and points to future work.

II. Related Work

The current strategies for controlling security risks are: (i) penetration and patch, (ii) standards, (iii) security risk management/assessment and (iv) "wait and see". The latter is similar to the first, only different in that penetration and patch often includes authorised penetration and patch activities, such as tiger-team activity. "Wait and see" is a passive security strategy where problems are fixed if budget allows and only after the fact.

Standards provide tools for evaluating the security controls of systems. Examples of such are ISO 15408:2007 Common Criteria for Information Technology Security Evaluation [1] (includes schema for certification of IT Products, in addition to security best practises) and the ISO/IEC 27000 series, such as ISO/IEC 27002:2005 Information technology – Security techniques – Code of Practice for Information Security Management [5]. However, most evaluations are a qualitative and subjective activity biased by the evaluator (even though they follow a standard). The ToE risk level estimation model is based on CVSS, which is an open standard that also reveals the details behind the scores provided. Furthermore, CVSS is regularly updated and several information sources is taken into consideration when calculating the CVSS score.

Risk assessment was initially developed within the safety domain, but has later been adapted to security critical systems as security risk assessment. The two most relevant approaches are CCTA Risk Analysis and Management Methodology (CRAMM) [6] and the CORAS framework [7]. CRAMM targets health-care information systems and is asset-driven. The CORAS framework is inspired by CRAMM and has adapted the asset-driven strategy of CRAMM.

The main deficiency of most security risk assessment approaches is that the focus is not on calculated risks, meaning that there has not been a prior activity on deciding which risk to accept, and not based on some cost-benefit strategy. To do so, it is necessary to obtain knowledge on potential attack paths, cost of attacks both for the attacker and the ToE owner, probability or frequency of attacks from an operational perspective and the potential impacts these may have on the strategic level. Calculating risk requires a quantitative analysis of the risk level. These are the challenges examined in the research domain quantifying security or operational security.

An initial model towards quantitative measures for operational security was discussed in Littlewood et al. (1993) [8]. The model derives operational measures such as mean time and effort to security breach. These concepts were further explored by Madan et al. (2002) [9] and Wang et al. (2003) [10]. Madan et al. (2002) discuss how to quantify security attributes of software systems using traditional reliability theory for modelling random processes, such as stochastic modelling and Markov analysis. In Wang et al. (2003) this idea was taken one step further using the higher level formalism Stochastic Petri Nets (SPN). Most of all, [10] discussed the problem of state exploitation and the inconvenience of the memoryless property of Markov processes. SPN deals with the state exploitations and in particularly the gigantic task of manually constructing a Markov chain. This model was extended to Coloured Petri Nets (CPN) in Houmb and Sallhammar (2005) [11]. However, neither SPN nor CPN tackle the increasing challenge of lack of data on how a system may react to certain security attacks, as the trend for potential and future security attacks still is largely unknown. At this point still little was known about the motivation and behaviour of attackers. Today, some more information exists, such as in vulnerabilities bulletins and attack trend reports. CVSS is an example of such. The benefit of CVSS is that it addresses the vulnerabilities directly and in collaboration with the vendors of the affected products. That is, CVSS tries to be specific and do not attempt to categorize attacks on a general basis nor does it provide a general model for estimating risk level. CVSS purely provide information about vulnerabilities on an operational level and leaves it to the vendors to add the information specific for their products and to the customers to interpret the information in the perspective of a particular ToE.

There are also other ways to measure risk exposure, such as Annual Loss Expectancy (ALE). However, the lack of quantitative data and the rapidly changing security environment makes it hard to derive accurate measures over such a long time-period.

Security trade-off analysis, as discussed in Houmb et al. (2005a) [12] and Houmb et al. (2006) [13], looks at security from a cost-benefit perspective in respect to financial and project factors, such as budget and time-to-market. However, the challenge is still on measuring the risk level in an accurate manner. An example of such for the security attribute availability is provided in Houmb et al. (2005b)[14], where an availability estimation model based on system service levels is outlined. The ToE risk level estimation model described in this paper extend the service level idea from [14].

Regarding the CVSS and its use there are few relevant works. Boehm, Chen and Sheppard (2007) [15] and Chen (2008) [16] discuss an approach to measuring security investment benefits for off the shelf software systems using CVSS. The argument made by the authors is that the CVSS may be misleading, as it does not incorporate the value context. Rather than using the environment variables of CVSS to give context to the values, the authors propose a AHP approach that focus on stakeholders values such as productivity, reputation and privacy of the systems where the vulnerabilities are located. However, both productivity and repudiation is of a subjective nature and hard to measure. That is, different stakeholders may have different perception on the extent that a vulnerability might affect the productivity. Our opinion is that it is better to use the environmental metrics as given in the CVSS, as stakeholders most often finds it easier to evaluate confidentiality, integrity and availability than productivity, reputation and privacy.

In Dondo (2008) [17], an approach to vulnerability prioritisation using fuzzy risk analysis is presented. Here, the construct asset value (AV) is used to derive the risk level or risks to a system. The asset value (AV) is assumed given. The approach derives risk level based on the CVSS base metrics variables, a measure of time from when the vulnerability was reported and the safeguards already in the system. The author applies fuzzy rules to compute impact (I) and likelihood (L) and derive risk level as: AV x I x L. This approach is similar to our model, but our model does not use fuzzy rules. Our model uses the temporal and environmental metric groups given in the CVSS to estimate the risk level rather than asset value and safeguard. Asset value is not always easy to evaluate and might be stakeholder specific. AV is not a generalsable variable, but rather context and stakeholder specific. By basing our model on the temporal and environmental information given in the CVSS we use easily accessible and publicly open context information that is regularly updated and maintained, has a stable data model and that is used by many commercial parties. Also, the underlying equations (that is how the scores or values are computed) are public knowledge for CVSS.

III. Common Vulnerability Scoring System (CVSS)

The CVSS is an effort to provide a universal and vender-independent score of known vulnerabilities. The system is on its second version and currently maintained by FIRST. The CVSS has since it was launched in 2004 been adopted by several vendors and vulnerability tools and bulletins. Examples are hardware and software development companies like IBM, HP and Cisco as a reporting metric, in vulnerability bulletins, by scanning vendor tools like Nessus and Qualys and by the NIST, which maintains the National Vulnerability Database (NVD); the main repository of known vulnerabilities worldwide.

The CVSS score is a decimal number on the scale [0.0,10.0] and is composed of three metrics groups: base, temporal and environmental [3]. The **base metrics** group quantifies the intrinsic characteristics (i.e. attributes) of a vulnerability using two sub-scores: (i) *exploitability_sub-score* and (ii) *impact_sub-score*. The exploitability sub-score is composed of *access vector (B_AR)* (type of access required to exploit the vulnerability in terms of *local*, *adjacent network* or *network*), *access complexity (B_AC)* (complexity involved in exploiting the vulnerability after the ToE has been identified in terms of *high*, *medium* or *low*), and *authentication (B_Au)* (number of authentication instances required once the ToE has been accessed in terms of *multiple*, *one* or *none*). The impact sub-score expresses the potential impact on *confidentiality (B_C)*, *integrity (B_I)* and *availability (B_A)* that the exploitation of the vulnerability can cause in terms of *none*, *partial* or *complete*.

Experts (from NIST) analyse each known vulnerability (called CVE [2]) and assign qualitative values to each attribute. For the base metrics they assign a rating (i.e. a qualitative value) for each attribute mentioned above. Based on these qualitative values, the CVSS system calculates scores using the pre-defined rating scales shown in Tables I and II. Therefore, if a CVE is assessed by NIST experts as having access="Network", complexity="Low" and authentication="None", the CVSS calculator returns the highest possible exploitability sub-score; 10.0. Similarly, "Complete" impact for Confidentiality, Integrity and Availability returns the highest possible impact sub-score; also 10.0. Each CVE reports: (i) the base score, (ii) the exploitability and impact sub-scores and (iii) the base vector from which the base score has been derived. For example, CVE-1999-0196 has base vector: [AV:N/AC:L/Au:N/C:P/I:N/A:N] corresponding to base score 5.0, *exploitability_sub-score* 10.0 and *impact_sub-score* 2.9.

The other metric groups; temporal and environmental, are either time or context-dependent and therefore not included in the NVD. The CVSS guide [3] defines these scores as follows.

The **temporal metrics** group quantifies dynamic aspects of a vulnerability using the three attributes: *exploitability tools & techniques (T_E)*, *remediation level (T_RL)* and *report confidence (T_RC)*. The exploitability attribute refers to the availability of code or techniques for exploiting a vulnerability and is evaluated in terms of: *unproved*, *proof-of-concept*, *functional* or *high*. The remediation level attribute refers to the type of remediation available for the vulnerability in terms of *official fix*, *temporary fix*, *workaround* or *unavailable*. The report confidence attribute refers to the trustworthiness (quality) of the information available for the vulnerability in terms of: *unconfirmed*, *uncorroborated* (conflicting sources of information) or *confirmed*. For all attributes the list of options (i.e. ratings) reflects increasing levels of exploitability. Furthermore, temporal attribute values are likely to change during the vulnerability life cycle as exploit code, official fixes and more reliable information about the vulnerability become available over time.

The **environmental metrics** group quantifies three aspects of a vulnerability that are dependent on the environment and on stakeholders' values: (i) *collateral damage potential (E_CDP)*, (ii) *target distribution* and (iii) *security requirements*. The collateral damage potential is a measure of the potential damage that exploiting the vulnerability may have to loss of life, physical asset loss, loss of revenue and loss of productivity. *E_CDP* is measured according to the qualitative scale {*none*, *low*, *low-medium*, *medium-high*, *high*}. The security requirements included are *confidentiality (E_CR)*, *integrity (E_IR)* and *availability (E_AR)*, which are all measured according to the qualitative scale: {*low*, *medium*, *high*}. Although the environmental attributes can eventually change over time, they are not as dynamic as the temporal attributes. In addition, they are specific to a particular ToE, while the temporal attributes are specific to a particular vulnerability. Note that the target distribution attribute is not explored in this paper.

More details on the CVSS metrics groups and on the CVSS calculator are in the CVSS guide [3].

[2] Common Vulnerabilities and Exposures [18]

IV. Estimating MI and MF using CVSS

We use the CVSS to estimate the two variables MF and MI. In fact, we rearrange the CVSS attributes to calculate MF and MI instead of base, temporal and environmental scores. The more exploitable a vulnerability is, the more likely it is to be exploited by attackers, and thus the MF will be higher. We are able to calculate MF for each vulnerability present in the ToE by first considering the exploitability factors intrinsic to the vulnerability itself (i.e. the base metrics related to exploitability), and then the temporal factors capable to lower the exploitability. The same rationale applies to impact: the potential impact intrinsic to a vulnerability (i.e. the base metrics related to impact) can be increased or decreased depending on the security requirements to the ToE.

Tables I and II show the attributes and rating values from CVSS that we use.

CVSS metrics group	CVSS attribute	rating	rating value
base metrics	access required (B_AR)	local (L)	0.395
		adjacent network (A)	0.646
		network (N)	1.0
	attack complexity (B_AC)	high (H)	0.35
		medium (M)	0.61
		low (L)	0.71
	authentication instances (B_Au)	multiple (M)	0.45
		single (S)	0.56
		none (N)	0.704
temporal metrics	exploitability tools & techniques (T_E)	unproved (U)	0.85
		proof-of-concept (POC)	0.9
		functional (F)	0.95
		high (H)	1.0
	remediation level (T_RL)	official fix (OF)	0.87
		temporary fix (TF)	0.90
		workaround (W)	0.95
		unavailable (U)	1.0
	report confidence (T_RC)	unconfirmed (UC)	0.90
		uncorroborative (UR)	0.95
		confirmed (C)	1.0

TABLE I
CVSS ATTRIBUTES USED FOR THE ESTIMATION OF **MF**

CVSS metrics group	CVSS attribute	rating	rating value
base metrics	confidentiality impact (B_C)	none (N)	0.0
		partial (P)	0.275
		complete (C)	0.660
	integrity impact (B_I)	none (N)	0.0
		partial (P)	0.275
		complete (C)	0.660
	availability impact (B_A)	none (N)	0.0
		partial (P)	0.275
		complete (C)	0.660
environmental metrics	confidentiality requirement (E_CR)	low (L)	0.5
		medium (M)	1.0
		high (H)	1.51
	integrity requirement (E_IR)	low (L)	0.5
		medium (M)	1.0
		high (H)	1.51
	availability requirement (E_AR)	low (L)	0.5
		medium (M)	1.0
		high (H)	1.51
	collateral damage potential (E_CDP)	none (N)	0.0
		low (L)	0.1
		lowmedium (LM)	0.3
		mediumhigh (MH)	0.4
		high (H)	0.5

TABLE II
CVSS ATTRIBUTES USED FOR THE ESTIMATION OF **MI**

A. *Estimating MF from base and temporal data*

We use three attributes from the base metrics and three attributes from the temporal metrics to derive the misuse frequency (MF). These are: access required (B_AR), attack complexity (B_AC) and authentication instances (B_Au) from the base metrics, and exploitability tools & techniques (T_E), remediation level (T_RL) and report confidence (T_RC) from the temporary metrics. As the CVSS base metrics refers directly to the exploitability of a vulnerability, we use the base metrics attributes to estimate the initial misuse frequency (MF). This is done in Equation 1. From the CVSS we also get the internal dependencies between the basic metrics attributes. These are not directly shown in Equation 1 as these may change when new knowledge about attacks becomes available. The CVSS is dynamic and will be updated to reflect such knowledge. More details are in Section VI.

The initial MF is then updated with the temporal metrics attributes. The temporal metrics attributes cover the indirect factors relevant for the exploitability of a vulnerability. The updating is done in a two step manner: first an update factor (uFac) is derived in Equation 2, then this update factor is applied to the initial misuse frequency to derive the resulting misuse frequency estimate (Equation 3). The details on the categories for each attribute (both basic and temporal) is in Table I. Note that the MF equations do not evaluate the value of each of the attributes. This activity is done as part of inserting input to each attribute in the MF equation set, which is demonstrated in Section VI.

Furthermore, the resulting MF estimate (derived in Equation 3) must be normalized, as it should always be a value in the range $[0, 1]$ (negative probability does not make sense and an event cannot be higher than certain; $P = 1.0$). The value 0 means that the vulnerability will never be exploited and the value 1 means that the vulnerability will for certain be exploited. Values in the range $< 0, 0.5 >$ means low possibility for exploits and values in the range $< 0.5, 1.0 >$ means high possibility for exploits. The value 0.5 should be interpreted as that it is just as likely that the vulnerability will be exploited as it will not.

$$MF_{init} = \int_{N\_1} P(B\_AR, B\_AC, B\_Au) \qquad (1)$$

$$MF_{uFac} = \int_{N\_1} P(T\_E, T\_RL, T\_RC) \qquad (2)$$

$$MF = \int_{N\_1} (MF_{init} \times MF_{uFac}) \qquad (3)$$

B. *Estimating MI from base and environmental data*

The variable misuse impact (MI) is used to group vulnerabilities into states for the state transition model and to associate vulnerabilities to service levels. We use four attributes from the base metrics and three attributes from the environmental metrics to derive the MI estimate. These are: confidentiality impact (B_C), integrity impact (B_I) and availability impact (B_A) from the base metrics, and confidentiality requirement (E_CR), integrity requirement (E_IR), availability requirement (E_AR) and collateral damage potential (E_CDP) from the environmental metrics. The environmental metrics are context specific and puts the confidentiality, integrity and availability impacts into the perspective of the security requirements to and the collateral damage potential of a particular ToE. This means that the base metrics describe the magnitude of the effect on each security property individually, which is then made ToE specific by applying the environmental metrics to the base metrics.

Similar to MF, the base metrics are used to establish the initial impact estimate, which is a vector over confidentiality, integrity and availability, as specified in Equation 4. The environmental metrics attributes are then used to update the initial impact estimate vector and to derive the resulting impact estimate vector. The updating is done in two steps. First the initial impact estimate vector is updated to account for the collateral damage potential (E_CDP) in Equation 5. When this is done, the impact estimate vector is updated with the security requirements information from the environmental metrics in Equation 6. The resulting impact estimate vector is derived in Equation 7.

$$\begin{aligned}
MI_{init} &= [B\_C, B\_I, B\_A] && (4)\\
MI_{CDP} &= \int_{N\_1} E\_CDP[B\_C, B\_I, B\_A] && (5)\\
MI_{Env} &= [B\_CR, B\_IR, B\_AR] && (6)\\
MI &= \int_{N\_1} MI_{CDP} \times MI_{Env} && (7)
\end{aligned}$$

The resulting MI estimate vector (Equation 7) expresses the severity of a particular vulnerability. It is this information that we use to specify the service levels and to associate vulnerabilities to service levels when deriving the ToE risk level. The service levels are organized in a state transition model (Markov process). The first state is always with no impact on all of confidentiality, integrity and availability, namely $[0.0, 0.0, 0.0]$. The last state in the state transition model is always complete impact on all of confidentiality, integrity and availability taking all the environmental metrics attributes into consideration. That is: $[1.0, 1.0, 1.0]$, which is an absorbing state. This means that there is no repair ones the ToE arrives at this state. Hence, the first state is always associated with full service level $SL0$ and the last state is always associated with no service SLx. All states in-between can be full service, any level of degraded service or no service and is ToE dependent.

V. Deriving ToE risk level from MF and MI estimates

We measure ToE risk level using service levels [19] expressed as a continuous-time Markov process [20]. A service level is defined as a group of ToE states, each denoting a specified degree of normal ToE accomplishment. The service levels depend on the design and implementation of the ToE, the structure of the ToE and the application of the ToE; that is, the way the ToE is used. A degraded service may be regarded as a full service for a certain application or by a certain user. Service levels are therefore organization and stakeholder specific. As described in the previous section, the highest service level is **service level 0 (SL0)** or full service. The lowest service level is **service level x (SLx)** or no service.

The ToE risk level model is supported by a two-step computational procedure: (1) Define the state transition model from misuse impact estimates and (2) Determine state transition rates from misuse frequency estimates. Step 1 is performed by examining vulnerability bulletins and databases such as the NVD and by running a vulnerability scanner such as Nessus to derive a list of vulnerabilities resided in the ToE. However, the latter is of various reasons not always possible to carry out (not possible to open the necessary ports on the firewalls in the network, the location of the ToE, etc.). The misuse frequency and impact of each vulnerability is then estimated as described in the previous section (Section IV).

The misuse impact of a vulnerability define its severity. This does not necessary means that two vulnerabilities having the same impact value pose the same severity to the ToE and hence lead to the same decrease in ToE service level. Thus, we need to define vulnerability severity level intervals and associate these to service levels. This results in an ordered set of service levels from no service level to full service level and defines the state transition model. In Section VI we give an example of a state transition model.

> **Definition** *A service level* is a composite of a non-empty set of vulnerabilities, all with severity level within one particular severity level interval.

In Step 2 we examine the state transition model derived in Step 1 and augment it with transition rates. Transition rates specify how likely it is to move from one state to another and how likely it is to be in a particular state at a particular time t. In the ToE risk level estimation model each state refers to the aggregated severity level for a set of vulnerabilities. Hence, the state transition model describes the various risk levels that the ToE may be in or arrive at for a particular time t. To determine the transition rates, we aggregate over the relevant misuse frequency estimates.

VI. Example of the use of the ToE Risk Level Model

Our ToE is the web server (200.30.0.2) of an academic institution located at the DMZ zone in the network. Scanning tools, such as the open source Nessus [21] tool, supply experts with the following information. The server has two open ports: tcp/80 listening to HTTP traffic and tcp/22 listening to SSH traffic. In practice, the SSH connection allows System Administrators to do maintenance work remotely from within the subnet administration. The SSH service has vulnerability CVE-2004-2320 and the HTTP service has vulnerability CVE-2003-0190. Additionally, experts learn that the external firewall (200.30.0.1) allows inbound and outbound HTTP traffic from the Internet to the DMZ and that the internal firewall (10.16.0.1) allows inbound and outbound HTTP and SSH traffic between the DMZ and the LAN.

The ToE is regarded by its stakeholders (university staff and students) as an asset that has high demand on availability and integrity (considering web pages are locally

	ToE			
E_CR	low (L)			
E_IR	high (H)			
E_AR	high (H)			
E_CDP	low-medium (LM)			
	CVE-2003-0190	**CVE-2004-2320**	**CVE-2005-4762**	**CVE-2003-1562**
B_AV	network (N)	network (N)	local (L)	network (N)
B_AC	low (L)	low (L)	low (L)	high (H)
B_Au	none (N)	none (N)	none (N)	none (N)
T_E	high (H)	functional (F)	functional (F)	functional (F)
T_RL	workaround (W)	official fix (OF)	official fix (OF)	workaround (W)
T_RC	confirmed (C)	confirmed (C)	confirmed (C)	confirmed (C)
B_C	partial (P)	partial (P)	complete (C)	partial (P)
B_I	none (N)	none (N)	complete (C)	none (N)
B_A	none (N)	none (N)	complete (C)	none (N)

TABLE III
SUMMARY OF INPUT FOR CALCULATION OF MF AND MI

stored), but low demand on confidentiality. Additionally, a successful exploit might result in moderate local damage (i.e. low-medium). Thus, the environmental vector is [E(CDP):LM/E(CR):L/E(IR):H/E(AR):H].

Vulnerability CVE-2003-0190 refers to a feature of OpenSSH 3.6.1 P1 with Pluggable Authentication Modules (PAM) support enabled that "sends an error message when a user does not exist, which allows remote attackers to determine valid usernames via a timing attack" [22]. The base vector for this vulnerability is [B(AV):N/B(AC):L/B(Au):N/B(C):P/B(I):N/B(A):N]. The vulnerability is exploited via SSH command line and a *proof-of-concept* code is available for automatic exploitation of this design problem [23]. The exploitability is therefore *high*. In terms of solutions to the vulnerability, one out of three vendors involved supplies an official fix for the problem. For the others there are workarounds available. The remediation level is considered to be *workaround* and the vulnerability is *confirmed*. This gives the temporal vector: [T(E):H/T(RL):W/T(RC):C].

Vulnerability CVE-2004-2320 refers to a default configuration on BEA WebLogic Server 8.1 SP 2 that "responds to the HTTP TRACE request, which can allow remote attackers to steal information using cross-site tracing (XST) attacks in applications that are vulnerable to cross-site scripting" [22]. This vulnerability has the base vector: [B(AV):N/B(AC):L/B(Au):N/B(C):P/B(I):N/B(A):N]. Furthermore, the vulnerability is exploited [24] by a crafted HTTP containing a malicious script triggered when a victim clicks on it in a web browser. Additionally, remediation is available and requires software upgrade. The vulnerability has been confirmed although Apache Software Foundation regards it not as a security issue. This gives the temporal vector: [T(E):F/T(RL):OF/T(RC):C].

We performed similar analysis for two more vulnerabilities: CVE-2005-4762 also on BEA WebLogic 8.1 SP 2 and CVE-2003-1562 on OpenSSH 3.6.1 P1. Table III summarises the input for calculating MF and MI for each vulnerability present in the ToE.

A. Deriving MF and MI estimates for each vulnerability

To derive the initial MF value, we make use of Bayes theorem. Given the two random variables x and y, the probability P for the variable x, given the variable y, can be calculated from: $P(x|y) = \frac{P(y|x) \times P(x)}{P(y)}$. By allowing x_i to be a complete set of mutually exclusive instances, Bayes formula can be extended to calculate the conditional probability of x_i given y. Details are in Jensen (1996) [25].

Considering the earlier discussed dependencies: $P(B\_AR)$ is independent, AC is dependent on AR; $P(B\_AC) = P(B\_AC|B\_AR)$, and Au is dependent on both AR and AC; $P(B\_Au) = P(B\_Au|(B\_AR, B\_AC))$, we use the theory of probability of the intersection and union of random events. By applying Bayes theorem on the case where the random event x is conditionally dependent on the two events y_1 and y_2 we get: $P(x|(y_1, y_2)) = \frac{P(x) \times P((y_1, y_2)|x)}{P(y_1, y_2)}$. Solving this equation requires knowledge on the dependencies between y_1 and y_2. In the case of Au, which is conditionally dependent on two events (AR and AC) that are internally dependent, we use the following formulate to calculate the joint probability function: $P(y_1, y_2) = P(y_1) + P(y_2) - (P(y_1) \times P(y_2))$.

By solving Equation 1 for CVE-2003-0190 using the above theorems and the normalization factor $\frac{1}{3}$ we get the initial frequency estimate for this vulnerability:

$$MF_{init} \quad = \quad \frac{1.0 + 0.71 + 0.702}{3} = 0.805$$

The final MF estimate for CVE-2003-0190 is derived by computing the update factor from the temporal metrics attributes and then applying this to the initial MF estimate. Note that the normalization factors are $\frac{1}{3}$ and $\frac{1}{2}$ respectively.

$$\begin{aligned} MF_{uFac} = & \quad \frac{1.0 + 0.95 + 1.0}{3} = 0.983 \\ MF(CVE-2003-0190) = & \quad \frac{0.805 + 0.983}{2} = 0.894 \end{aligned}$$

MF for vulnerability CVE-2004-2320 is derived the same way as for CVE-2003-0190, which gives: $MF(CVE-2003-2320) = 0.798$. (Initial MF is 0.805 and update factor is 0.79). Rounding both to two decimals gives: $MF(CVE-2003-0190) = 0.89$ and $MF(CVE-2003-2320) = 0.80$. Similar, MF for vulnerability CVE-2005-4762 is: $MF(CVE-2005-4762) = 0.80$ and MF for vulnerability CVE-2003-1562 is: $MF(CVE-2003-1562) = 0.82$.

MI for all vulnerabilities are derived using the MI equations from Section IV-B (but without taking the collateral damage potential into consideration as E_CDP is equal for all vulnerabilities and do not affect the internal relations). This gives:

$$\begin{aligned} MI(CVE-2003-0190) &= [0.14, 0.0, 0.0] \\ MI(CVE-2003-2320) &= [0.14, 0.0, 0.0] \\ MI(CVE-2005-4762) &= [0.41, 1.0, 1.0] \\ MI(CVE-2003-1562) &= [0.14, 0.0, 0.0] \end{aligned}$$

B. Deriving ToE risk level

We create the service level state transition diagram from the MI estimate vectors. First, we need to group the vulnerabilities into service levels according to their severity or impact level. We see that there are three vulnerabilities with the same impact: CVE-2003-0190, CVE-2003-2320 and CVE-2003-1562. The impact of these three vulnerabilities is lower than the impact of the fourth vulnerability and we group these into service level 1 (SL1). The fourth vulnerability; CVE-2005-4762, has a larger impact and becomes service level 2 (SL2). In addition, we have the full service level (SL0) and the no service level (SLx). This gives four service levels: *SL0*, *SL1*, *SL2* and *SLx*.

The ToE risk level is given as a continuous-time Markov process $\{X_t\}_{t\geq 0}$ with a finite state space *E*, in which each service level *SLn* can be identified with a subset of states in *E*. Thus, *E* is the disjoint union $\{$*SL0* +...+ *SLx*$\}$, where *x* is the number of service levels. Furthermore, service levels $\{$*0,...,k*$\}$ correspond to **operational** states *O*; i.e., states specifying levels of acceptable ToE risk level. Service levels $\{$*k+1,...,x*$\}$ denote the **failed** states. Thus, *E* = *O* + *F*, where *O* = $\{$*SL0*+ ... +*SLk*$\}$ and *F* = $\{$*SL(k+1)*+ ... +*SLx*$\}$. For our example ToE we have three **operational** states (*SL0, SL1, SL2*) and one **failed** state *SLx*.

The transition rates are derived from the MF estimates. We assume that the ToE starts at the highest service level; $i_0 \in$ *SL0*. Transitions between operational states represent degradations and transitions to a failed state represent failures. We do not consider repair or online implementation of security solutions (such as dynamic software updates), meaning that no transitions take place from a failed state: $\lambda_{fj} = 0$ for all $f \in F$. Thus, failed states are absorbing. Transition rate between *SL0* (state 1) and *SL1* (state 2) is specified by $MF(SL0SL1)$. Transition rate between *SL1* (state 2) and *SL2* (state 3) is specified by $MF(SL1SL2)$. And, transition rate between *SL2* (state 3) and *SLx* (state 4) is specified by $MF(SL2SLx)$. Table IV shows the transition rates for the ToE risk level state model. Note that there are never transitions from a higher state to a lower state (no repair). Also note that in this example we assume independence between states.

The ToE will move from SL0 to SL1 if either one of the three vulnerabilities: CVE-2003-0190, CVE-2003-2320 and CVE-2003-1562, is exploited. If we assume that the three vulnerabilities are mutually exclusive the transition rate between SL0 and SL1 equals the disjoint probability of the three potential exploits (the event that a vulnerability is exploited). As this adds up to more than 1.0, we simply use the average, which is 0.84. (This is reasonable as all MF values are high and as the variance between them is relatively low). Transition rate from SL0 to SL2 equals the frequency of vulnerability CVE-2005-4762: 0.80, as we allow for the single exploitation of vulnerability CVE-2005-4762. To get to SL2 from SL1, the ToE first needs to move from SL0 to SL1. Thus, transition rate between SL1 and SL2 are dependent on transition rate between SL0 and SL1 ($MF(SL1SL2)|MF(SL0SL1)$). The same logic is used to derive the transition rate from service level 2 (SL2) to service level x (SLx). That is, we assume that the system do not go directly from full service level or SL1 to no service level (SLx).

	SL0	**SL1**	**SL2**	**SLx**
SL0	0.0	0.84	0.80	0.0
SL1	0.0	0.0	0.67	0.0
SL2	0.0	0.0	0.0	0.45
SLx	0.0	0.0	0.0	0.0

TABLE IV
ToE RISK LEVEL TRANSITION RATE MATRIX

The transition rate matrix can be used to answer questions like: "What is the risk level for a ToE at a particular time *t*?", "Is it likely that a particular vulnerability is exploited at a particular time *t*?", "What is a likely time before the ToE will enter a non-reparable state?", etc. To find answers to these questions it is necessary to put the transition rates into a proper time perspective. E.g., if we consider a time perspective of one year, it is likely that the system will reach a non-reparable state within less than 6 months (probability of 0.45). It is also likely that within less than 7 months the integrity and availability of the ToE is fully compromised. This makes sense as all four vulnerabilities have a high frequency. Another interpretation that can be used is that within any given time frame T, the ToE will be fully compromised with a probability of 0.67 and reached a non-reparable state with a probability of 0.45. (The better way to express this is that there is almost a 70% chance that the ToE will be fully compromised and a 45% chance that it will reach a non-repairable state.) However, practice and literature has shown that stakeholders, such as decision makers, have problems understanding probability expressions even when they are as simple as these [20]. One year is thus a reasonable time frame taken the high frequency and impact estimates of the four vulnerabilities considered in this example. In most cases this means that security measures, and in particular security measures targeting integrity and availability, must be employed within a reasonable time frame. A concrete expression of the ToE risk level, as that given within the time frame of one year, will communicate this message clearly to the decision maker.

VII. Conclusion

The paper describes a ToE risk level estimation model that uses CVSS to estimate misuse frequency (MF) and misuse impact (MI), and from these derive the risk level of a ToE. The model is demonstrated using an example ToE. MF is estimated from attributes in the base and temporal metrics of CVSS and MI is estimated from attributes in the base and environmental metrics of CVSS. The base metrics of CVSS is used to establish the initial estimates of both MF and MI. MF is then made attack specific by adding in factors concerning the attack tools available, the existing security measures and the report confidence. For MI, the initial MI of a potential vulnerability exploit (attack) derived from the base metrics is made ToE specific by taking the relevant security requirements into consideration. An important factor to note for MI is that there are no impacts of a potential vulnerability exploit (attack) if there are no relevant requirements. This is a general risk assessment principle that is used to limit the amount of security risks in need of treatment and to support cost-effective security budgeting.

Note that the accuracy of the resulting MF and MI estimates, and thus also the ToE risk level estimate, are dependent on the accuracy of the values assigned to the base, temporal and environmental metrics attributes that we use. That is, the attribute internal weights given in Tables I and II. These values are given by the CVSS. Lately, there have been some discussions about the accuracy of these values. However, our model is flexible in that it only uses these values as interchangeable weights. This means that we only need to update the MF and MI tables (Tables I and II) whenever CVSS provides updated values provided that no new attributes are introduced.

Future work includes a series of field studies of using the model in practise at our industrial partners. Thus far, we have tested the model on example ToEs. Future work also involve merging the ToE risk level estimation model into a security solution trade-off analysis [2] as part of a larger security budgeting support tool that we are building. An attempt to do so is currently ongoing as part of a field study.

Acknowledgment

This research is supported by the research program Sentinels (www.sentinels.nl). Sentinels is being financed by the Technology Foundation STW, the Netherlands Organization for Scientific Research (NWO) and the Dutch Ministry of Economic Affairs.

References

[1] "ISO 15408:2007 Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 2, CCMB-2007-09-001, CCMB-2007-09-002 and CCMB-2007-09-003," International Organization for Standardization (ISO), September 2007.

[2] S. Houmb, "Decision Support for Choice of Security Solution: The Aspect-Oriented Risk Driven Development (AORDD) Framework," Ph.D. dissertation, Norwegian University of Science and Technology (NTNU), November 2007.

[3] P. Mell, K. Scarfone, and S. Romanosky, "A complete guide to the common vulnerability scoring system, version 2.0," Published by FIRST - Forum of Incident Response and Security Teams, June 2007, http://www.first.org/cvss/cvss-guide.pdf.

[4] S. Houmb, V. Franqueira, and E. Engum, "Estimating Impact and Frequency of Risks to Safety and Mission Critical Systems Using CVSS," in *Supplementary Proceedings of the ISSRE 2008 Conference: 1st Workshop on Dependable Software Engineering 2008*. IEEE Computer Society, 11 November 2008, p. 6 pages.

[5] International Organization for Standardization (ISO/IEC), "ISO/IEC 27002:2005 Information technology – Security techniques – Code of Practice for Information Security Management," 2005.

[6] B. Barber and J. Davey, "The Use of the CCTA Risk Analysis and Management Methodology CRAMM in Health Information Systems," in *Proceedings of MEDINFO'92*, K. Lun, P. Degoulet, T. Piemme, and O. Rienhoff, Eds. North Holland Publishing Co, Amsterdam, 1992, pp. 1589–1593.

[7] K. Stølen, F. den Braber, T. Dimitrakos, R. Fredriksen, B. Gran, S. Houmb, Y. Stamatiou, and J. Aagedal, *Business Component-Based Software Engineering*. Kluwer, 2002, ch. Model-based Risk Assessment in a Component-Based Software Engineering Process: The CORAS Approach to Identify Security Risks, pp. 189–207.

[8] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright, J. Dobson, J. McDermid, and D. Gollmann, "Towards Operational Measures of Computer Security," *Journal of Computer Security*, vol. 2, pp. 211–229, 1993.

[9] B. Madan, K. Goseva-Popstojanova, K. Vaidyanathan, and K. Trivedi, "Modeling and Quantification of Security Attributes of Software Systems," in *Proceedings of the International Conference on Dependable Systems and Networks (DSN'02)*, vol. 2. IEEE Computer Society, 2002, pp. 505–514.

[10] D. Wang, B. Madan, and K. Trivedi, "Security Analysis of SITAR Intrusion Tolerance System," in *Proceedings of the 2003 ACM workshop on Survivable and self-regenerative systems: in association with 10th ACM Conference on Computer and Communications Security*. ACM Press, 2003, pp. 23–32.

[11] S. Houmb and K. Sallhammar, "Modelling System Integrity of a Security Critical System Using Colored Petri Nets," in *Proceeding of Safety and Security Engineering (SAFE 2005)*. Rome, Italy: WIT Press, 2005, pp. 3–12.

[12] S. Houmb, G.Georg, R.France, J. Bieman, and J. Jürjens, "Cost-Benefit Trade-Off Analysis using BBN for Aspect-Oriented Risk-Driven Development," in *Proceedings of Tenth IEEE International Conference on Engineering of Complex Computer Systems (ICECCS 2005), Shanghai, China*, June 2005, pp. 195–204.

[13] S. H. Houmb, G. Georg, J. Jürjens, and R. France, "An Integrated Security Verification and Security Solution Design Trade-Off Analysis," *Integrating Security and Software Engineering: Advances and Future Visions, Chapter 9*, 2006, 288 pages.

[14] S. Houmb, G. Georg, R. France, R. Reddy, and J. Bieman, "Predicting Availability of Systems using BBN in Aspect-Oriented Risk-Driven Development (AORDD)," in *Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics, Volume X: 2nd Symposium on Risk Management and Cyber-Informatics (RMCI'05)*. International Institute of Informatics and Systemics, July 2005, pp. 396–403, Orlando, Florida, USA.

[15] B. Boehm, Y. Chen, and L. Sheppard, "Measuring security investment benefit for off the shelf software systems – a stakeholder value driven approach," p. 18 Pages, 7–8 June 2007, http://weis2007.econinfosec.org/papers/46.pdf, accessed 30 September 2008.

[16] Y. Chen, "Stakeholder value driven threat modeling for off the shelf based systems," in *ICSE COMPANION '07: Companion to the proceedings of the 29th International Conference on Software Engineering*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 91–92.

[17] M. Dondo, "A Vulnerability Prioritization System Using A Fuzzy Risk Analysis Approach," in *Proceedings of the IFIP TC 11 23rd International Information Security Conference*, S. Jajodi, P. Samarati, and S. Cimato, Eds., vol. 278. IFIP International Federation for Information Processing (Boston:Springer), 4–6 November 2008, pp. 525–539.

[18] "Common vulnerabilities and exploitations," http://cve.mitre.org/.

[19] E. Jonsson, "On the Integration of Security and Dependability in Computer Systems," in *IASTED Int'l Conf. Reliability, Quality Control and Risk Assessment*, 4–6 November 1992, pp. 93–97.

[20] D. Vose, *Risk Analysis: A Quantitative Guide*. John Wiley & Sons Ltd., 2000.

[21] Nessus, "Tenable network security: The Nessus Security Scanner," www.nessus.org.

[22] NVD, "National vulnerability database v2," http://nvd.nist.gov/.

[23] M. Ivaldi, "OpenSSH/PAM timing attack allows remote users identification," Bugtraq Security List, 30-Apr-2003, http://marc.info/?l=bugtraq&m=105172058404810&w=2, accessed 7 May 2008.

[24] "BEA WebLogic Server and Express HTTP TRACE cross-site scripting," IBM Internet Security Systems (ISS), 27 Jan 2004, http://xforce.iss.net/xforce/xfdb/14959, accessed 7 May 2008.

[25] F. Jensen, *An introduction to Bayesian Network*. University College London: UCL Press, 1996.