Using Trust-Based Information Aggregation for Predicting Security Level of Systems

Siv Hilde Houmb, Sudip Chakraborty, Indrakshi Ray, Indrajit Ray

Using Trust-Based Information Aggregation for Predicting Security Level of Systems*

Siv Hilde Houmb[1], Sudip Chakraborty[2], Indrakshi Ray[3], and Indrajit Ray[3]

[1] Telenor GBD&R siv-hilde.houmb@telenor.com
[2] Valdosta State University schakraborty@valdosta.edu
[3] Colorado State University {iray, indrajit }@cs.colostate.edu

Abstract. Sometimes developers must design innovative security solutions that have a rapid development cycle, short life-time, short time-to-market, and small budget. Security evaluation standards, such as Common Criteria and ISO/IEC 17799, cannot be used due to resource limitations, time-to-market, and other constraints. We propose an alternative time and cost effective approach for predicting the security level of a security solution using information sources who are trusted to varying degrees. We show how to assess the trustworthiness of each information source and demonstrate how to aggregate the information obtained from them. We illustrate our approach by showing the security level prediction for two Denial of Service (DoS) solutions.

1 Introduction

Often times there is a need to build a security solution, that has a rapid development cycle, short life-time, and short time-to-market. It is important to predict the security level of such a solution before it can be deployed. Predicting the security level of a solution using standards, such as the Common Criteria [1] has drawbacks. First, the result of a Common Criteria evaluation is not given as a statement of the security level of a system, but rather as the level of assurance that the evaluator has based on whether the set of security features present provide adequate security. Second, Common Criteria evaluations are time and resource intensive. Third, the documentation and tests required by Common Criteria may not be suitable for the required system [2].

Such shortcomings motivated us to propose an alternative approach for predicting the security level of a system using information collected from different sources, not all of whom are equally trustworthy. We propose a model of trust to capture the trustworthiness of information sources, specifically that of domain experts. Trust is a relationship between a truster and a trustee with respect to some given context. Here, the entity trying to obtain information from the sources is the truster, the information source is the trustee, and the problem for which the information is requested is the trust context. The trustworthiness of an information source depends on two factors, namely, its *knowledge level* and *expertise level*. Knowledge level captures the level of knowledge possessed by the information source with respect to the problem being addressed. Expertise level captures the experience and qualifications of the information source. We show how to

* This work was supported in part by AFOSR under contract number FA9550-07-1-0042.

evaluate these factors and quantify the trustworthiness of sources which are later used for security level prediction.

The rest of the article is organized as follows. Section 2 presents the approach for predicting the security level of a solution. Section 3 illustrates our approach by predicting the security level of two DoS solutions. Section 4 summarizes the related work in this area. Section 5 concludes the paper with pointers to future directions.

2 Predicting the Security Level of Security Solutions

The first step in security level prediction is assessing the trustworthiness of an information source. The trustworthiness of a source depends on the *knowledge level* and *expertise level* of an information source. *Knowledge level* of an information source is defined as a measure of awareness of the information source about the knowledge domains related to the security level prediction of the security solution. It is represented in terms of a number called *knowledge score*. *Expertise level* of an information source is defined as a measure of degree of ability of the information source to assess the security level of a security solution. It is represented in terms of a number called *expertise score*. *Trustworthiness* of an information source is defined as a measure of the competence of the information source to act desirably and to provide information to the best of its abilities. It is represented in terms of a number called *trustworthiness score*. Trustworthiness score is derived from knowledge score and expertise score.

2.1 Evaluating Knowledge Score of an Information Source

The knowledge score of an information source gives a measure of how closely his/her knowledge is related to the desired knowledge in the problem context. It is calculated from two scores – *reference knowledge domain score* and *information source knowledge domain score* which are derived from the *reference knowledge domain model* and *information source knowledge domain model* respectively. The reference knowledge domain model provides the relative importance of different knowledge domains regarding the problem context. The information source knowledge domain model gives an assessment, by a third party, of the relative importance of knowledge level of an information source corresponding to the knowledge domains identified in reference knowledge domain model.

Reference Knowledge Domain Model

Prediction of security level of a security solution typically involves knowledge in several domains, not all of which are equally important. Knowledge level of an information source measures his/her awareness about these knowledge domains. We develop a reference knowledge domain model that captures the domains that are of interest and their relative importance with respect to the problem context. The relative importance of a domain is measured in terms of *importance weight* which is defined to be the percentage of the whole reference knowledge domain covered by that particular knowledge domain. Figure 1 shows a reference knowledge domain model for a security solution consisting of four domains: domain A (network security) domain B (Internet Protocol), domain C (authentication) and domain D (access control). All these domains cover the

whole knowledge domain equally (25%), and hence have equal importance. Thus the importance weight of each domain is 0.25.

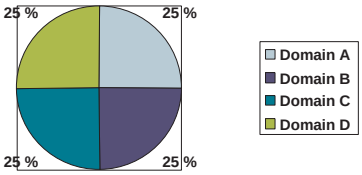


Fig. 1. Reference knowledge domain model

In the computation of reference knowledge domain score, we find out the knowledge domains that are of interest for the particular security level case prediction, arrange the knowledge domains in some order, and find their respective importance weight. A vector, called *reference knowledge domain scores*, represents the relative importance of all knowledge domains pertinent to the security level prediction of the target security solution. Each element of the vector indicates the importance weight of the corresponding domain.

Calculating Reference Knowledge Domain Score

Each knowledge domain in the reference model has a particular importance weight associated to it. Since multiple stakeholders are often involved in formalizing the problem context, the different stakeholders may assign different weights to it. Suppose the stakeholders are denoted by the set X and the cardinality of the set is q. We use x to denote an individual stakeholder. Suppose m is the number of knowledge domains in the problem context. The importance of knowledge domains, from the point of view of a stakeholder x, are represented as an m-element vector. This vector is denoted by $W_{Kimp}(x)$ where $W_{Kimp}(x) = [w_{Kimp}(x(j))]_{j=1}^{m}$ (Equation 1). Here, $w_{Kimp}(x(j)) \in [0,\ 1]\ \forall j = 1,\ldots,m$ and $\sum_{j=1}^{m} w_{Kimp}(x(j)) = 1$. Note, we obtain such vector for each of the q stakeholders in the set X. The importance of the m different domains given by q stakeholders is presented in a $q \times m$ matrix denoted by $W_{allKimp}(X)$ (Equation 2). The next step is to aggregate the information obtained from q stakeholders using an aggregation function, denoted by $f_{aggregation1}$, on the $q \times m$ matrix $W_{allKimp}(X)$ to merge the rows, resulting in a vector of size m. Equation 3 indicates the result of this aggregation. Here we do the aggregation by taking the arithmetic average for each m elements from all q number of vectors and put them into a single vector (for X), $W_{aggregatedKimp}(X)$, which is given by $[w_{aggregatedKimp}(X(j))]_{j=1}^{m}$. To normalize this vector, the normalization factor is obtained using Equation 4. Finally, the weight of each domain in the problem context is obtained by normalizing each element in the vector $W_{aggregatedKimp}(X)$ by the above normalization factor to obtain the vector $W_{refKnowledgeDomainScore}(X)$ (Equation

5). This vector derives the relative importance for each knowledge domain in the reference knowledge domain model.

$$W_{Kimp}(x) = [w_{Kimp}(x(j))]_{j=1}^{m} \tag{1}$$

$$W_{allKimp}(X) = [W_{Kimp}(x)]_{x=1}^{q} \tag{2}$$

$$\begin{aligned} W_{aggregatedKimp}(X) &= f_{aggregation1}(W_{allKimp}(X)) \\ &= [w_{aggregatedKimp}(X(j))]_{j=1}^{m} \end{aligned} \tag{3}$$

$$f_{refKnorm} = \frac{1}{\sum_{j=1}^{m} w_{aggregatedKimp}(X(j))} \tag{4}$$

$$\begin{aligned} W_{refKnowledgeDomainScore}(X) &= f_{refKnorm} \times W_{aggregatedKimp}(X) \\ &= [w_{refKnowledgeDomainScore}(X(j))]_{j=1}^{m} \end{aligned} \tag{5}$$

If simple average is used as an aggregation technique then we do not need to normalize the vector $W_{aggregatedKimp}(X)$ as each element of the vector will be in $[0,\ 1]$ and sum of all elements will be 1. In that case, we can ignore Equation 4 and we will have $W_{aggregatedKimp}(X) = W_{refKnowledgeDomainScore}(X)$.

Information Source Knowledge Domain Model

An information source may not have knowledge in all the knowledge domains represented in the reference domain model. The information source knowledge domain model provides the relative importance of the knowledge level of the source corresponding to the knowledge domains in reference knowledge domain model. This relative importance is assessed by a third party to reduce the bias involved in self-assessment.

Consider the reference knowledge domain example shown in Figure 1. Now, for an information source, say b, a third party assessor assesses the relative importance of knowledge level of b on the identified knowledge domains as 30% on domain A, 30% on domain B, and 40% on domain D. Thus, the relative importance of b's knowledge level on the domains, as assessed by a third party, is $[0.3, 0.3, 0.0, 0.4]$.

Suppose we have n information sources, denoted by $b_1, b_2, \ldots, b_n$, in a security level prediction. Suppose Y is the set of third parties assessing the knowledge of these n information sources. Suppose, cardinality of Y is z and an individual third party in the set Y is denoted by y. Then, information source knowledge domain score is represented as an m-element vector where each element corresponds to some knowledge domain of the information source. Each element indicates the relative weight of that domain and has a weight between 0 and 1. Equations 6–10 show how to compute the information source knowledge domain score for a source b_i.

$$W_{Kis}(y(b_i)) = [w_{Kis}(y(b_i(j)))]_{j=1}^{m} \tag{6}$$

$$W_{allKis}(Y(b_i)) = [W_{Kis}(y(b_i))]_{y=1}^{z} \tag{7}$$

$$\begin{aligned} W_{aggregatedKis}(Y(b_i)) &= f_{aggregation2}(W_{allKis}(Y(b_i))) \\ &= [w_{aggregatedKis}(Y(b_i(j)))]_{j=1}^{m} \end{aligned} \tag{8}$$

$$f_{isKnorm} = \frac{1}{\sum_{j=1}^{m} w_{aggregatedKis}(Y(b_i(j)))} \tag{9}$$

$$\begin{aligned} W_{isKnowledgeDomainScore}(Y(b_i)) &= f_{isKnorm} \times W_{aggregatedKis}(Y(b_i)) \\ &= [w_{isKnowledgeDomainScore}(Y(b_i(j)))]_{j=1}^{m} \end{aligned} \tag{10}$$

Each third party $y \in Y$ provides a vector, denoted by $W_{Kis}(y(b_i))$, of m-elements. Each element represents the assessed weight of knowledge level of the information source b_i corresponding to the domain represented by that element as shown in Equation 6. This step is repeated for each y in the set Y and results in z such vectors. To aggregate information from all y for the information source b_i, these z vectors are first combined in a $z \times m$ matrix in Equation 7 and then aggregated using some aggregation function in Equation 8. The aggregation function is denoted as $f_{aggregation2}$ in the equation. The aggregation technique used here is arithmetic average. We normalize this vector using the normalization factor obtained in Equation 9. Finally, the weight of each domain in the problem context is obtained by normalizing each element in the vector $W_{aggregatedKis}$ by the above normalization factor to obtain the vector $W_{isKnowledgeDomainScore}$ (Equation 10). The result gives one vector for the set Y holding the relative knowledge domain scores for the information source b_i. All these steps are then repeated n times (as we have n number of information sources in the security level prediction).

Calculating Knowledge Score of Information Sources

The knowledge score of an information source b_i, denoted by $K_{score}(b_i)$, gives a measure of the source's knowledge level and is calculated using the reference knowledge domain score and the information source knowledge domain score of b_i. For an information source b_i, this score is calculated as follows.

$$K_{score}(b_i) = \sum_{j=1}^{m} \{w_{refKnowledgedomainScore}(X(j)) \times w_{isKnowledgeDomainScore}(Y(b_i(j)))\} \tag{11}$$

The result of the above equation is a real number derived by component-wise multiplication of the two vectors $W_{refKnowledgeDomainScore}(X)$ and $W_{isKnowledgeDomainScore}(Y(b_i))$ and then adding all the product values.

2.2 Evaluating Expertise Score of an Information Source

Expertise level of an information source with respect to assessing the security level of a security solution is represented by the *expertise score*. We propose to evaluate the expertise score using questionnaires to reduce the bias of subjective assessment. Each questionnaire consists of a set of *calibration variables* which are further divided into *categories*. Table 1 provides an example questionnaire.

Each information source is assessed on each calibration variable according to the information source's category for that variable. The importance value for each calibration variable and the value associated with each category is determined by some external source, such as an expert[4]. To derive expertise score of an information source, we develop *calibration variable importance weight model* and *calibration variable category importance weight model*.

Calibration Variable Importance Weight Model

The relative importance of a calibration variable is assessed by external sources. Suppose the set of such external sources is denoted by X' and the cardinality of the

[4] Interested readers are referred to Cooke [3] and Goossens et al. [4] for an overview of the general challenges and benefits related to expert judgments.

Variables	Categories
level of expertise	low, medium and high
age	under 20, [20-25), [25-30), [30-40), [40-50), over 50
years of relevant education	1 year, 2 years, Bsc, Msc, PhD, other
years of education others	1 year, 2 years, Bsc, Msc, PhD, other
years of experience from industry	[1-3) years, [3-5) years, [5-10) years, [10-15) years, over 15 years
years of experience from academia	[1-3) years, [3-5) years, [5-10) years, [10-15) years, over 15 years
role experience	database, network management, developer, designer, security management and decision maker

Table 1. Example calibration variables for determining expertise level of information sources

set is u. Each calibration variable that is pertinent to the problem context is associated with an importance value. A member x' of the set X' assigns an importance value from the range $(0,1]$ to a calibration variable such that the sum of the importance value of all the calibration variables used is 1. Let there be p calibration variables denoted by $l_1, l_2, \ldots, l_p$ and $W_{l_1}, W_{l_2}, \ldots, W_{l_p}$ be their relative importance value assigned by the external source x'. This is represented by a vector $W_l(x') = [w_{l_j}(x')]_{j=1}^{p}$ and shown in Equation 12. All u members of X' will assign such values. For each calibration variable, the final importance value is derived by applying an aggregation function, $f_{aggregation3}$, on $W_l(X')$ (Equation 14). Since, $w_{l_j}(x') \in (0,\ 1]$ for all $j = 1, \ldots, p$ and for each $x' \in X'$, the aggregation function is so chosen that each element of $W_I(X')$ is in $(0,\ 1]$ and $\sum_{j=1}^{p} W_{l_j}(X') = 1$.

$$W_l(x') = [w_{l_j}(x')]_{j=1}^{p} \tag{12}$$

$$W_l(X') = [W_l(x')]_{x'=1}^{u} \tag{13}$$

$$W_{aggregatedCalwt}(X') = f_{aggregation3}(W_l(X')) \tag{14}$$

Calibration Variable Category Importance Weight Model

Each category in a calibration variable is also associated with a value that denotes the importance weight of the category of that calibration variable. These values are assigned by the external sources in X'. Let the calibration variable l_j have s categories denoted by $l_{j_1}, l_{j_2}, \ldots, l_{j_s}$ where $l_{j_k} \in [0,\ 1]$ for all $k = 1, \ldots, s$ (Equation 15). All u members of X' assign weights and then an aggregation function is used to derive the category weights for calibration variable l_j (Equations 16 and 17 respectively).

$$W_c(x'(l_j)) = [w_c(x'(l_j(i)))]_{i=1}^{s} \tag{15}$$

$$W_c(X'(l_j)) = [W_c(x'(l_j))]_{x'=1}^{u} \tag{16}$$

$$W_{aggregatedC}(X'(l_j)) = f_{aggregation4}(W_c(X'(l_j)) \tag{17}$$

Therefore, $W_{aggregatedC}(X'(l_j))$ holds the importance weight (as derived by all external sources in X') of each category of the calibration variable l_j. The above is done for all

the calibration variables ($j = 1, \ldots, p$). Here, note that all the p calibration variables may not have s categories.

Information Source Calibration Variable Category Score Model

An information source (b_i) receives scores for applicable categories within each calibration variable by a set Y' of external sources where cardinality of y' is v. This score is computed as follows. Each information source b_i is required to fill the questionnaire. Each member of Y' assesses the filled questionnaire and assigns a score in the range $[0,\ 1]$ for applicable categories within each calibration variable. Equation 18 shows such scores, assigned by an $y' \in Y'$, for the calibration variable l_j. All v members of Y' assigns such scores and then an aggregation is used to reduce it to single set of values (Equations 19 and 20). Hence, information source calibration variable category score model is designed as

$$W_{isCat}(y'(b_i(l_j))) = [w_{isCat}(y'(b_i(l_j(m))))]_{m=1}^{s} \tag{18}$$

$$W_{isCatAll}(Y'(b_i(l_j))) = [W_{isCat}(y'(b_i(l_j)))]_{y'=1}^{v} \tag{19}$$

$$W_{isCatAggregated}(Y'(b_i(l_j))) = f_{aggregation5}(W_{isCatAll}(Y'(b_i(l_j)))) \tag{20}$$

The above is done for all calibration variables considered for the problem. Note, for some calibration variable, the members of Y' may not need to assign any score. For example, for the calibration variable *level of expertise*, the importance weight of the applicable category (according to filled questionnaire) can work as the score. Hence, members of Y' can assign simply 1.0 to the category.

Calculating Expertise Score of Information Sources

The set X' of external experts assigns importance weights of each category within each calibration variable. Also the information source b_i receives scores for applicable categories within each calibration variable by another set of experts Y'. These two are combined to derive the information source's score for each calibration variable. Equation 21 gives the value obtained by b_i for calibration variable l_j. The weighted sum of all these calibration variable scores, where the weight is the importance weight of the corresponding calibration variable, gives the expertise score of b_i, denoted by $E_{score}(b_i)$ as demonstrated by Equation 22.

$$W_{calScore}(b_i(l_j)) = \sum_{m=1}^{s} W_{aggregatedC}(X'(l_j(m))) \times W_{isCatAggregated}(Y'(b_i(l_j(m)))) \tag{21}$$

$$E_{score}(b_i) = \sum_{j=1}^{p} W_{aggregatedCalwt}(X'(j)) \times W_{calScore}(b_i(l_j)) \tag{22}$$

2.3 Computing Information Source Trustworthiness

The information sources involved in the security level prediction have varying degrees of trustworthiness, which depends on their knowledge levels and expertise levels. Therefore, the knowledge score and the expertise score must be combined to compute the trustworthiness of an information source. Here again, the problem context will determine the relative importance of each score. Let k and e be the relative importance of the knowledge and expertise score. The following relations hold: $0 \leq k, e \leq 1$ and

$k+e=1$. The values of k and e can be set by the evaluator (or, truster). The trustworthiness score for information source b_i, denoted by $T_{score}(b_i)$, is computed as follows.

$$T_{score}(b_i) = k \times K_{score}(b_i) + e \times E_{score}(b_i) \tag{23}$$

2.4 Computing Security Level of a Security Solution

The trustworthiness score of an information source is used to compare the security level of different security solutions. The information obtained from each source b_i (in the form of a number $\in [0,\ 1]$), denoted by $b_i(I)$, is multiplied by the trustworthiness score of that source. This is done for all the sources. The results are then added and divided by n. This gives the initial security level for the security solution s_j as shown by Equation 24. This is done for all s_j in the set of security solutions S. Since the r security solutions are compared against each other, we must obtain a relative security level for each solution. The relative security level of s_j is computed using Equation 25.

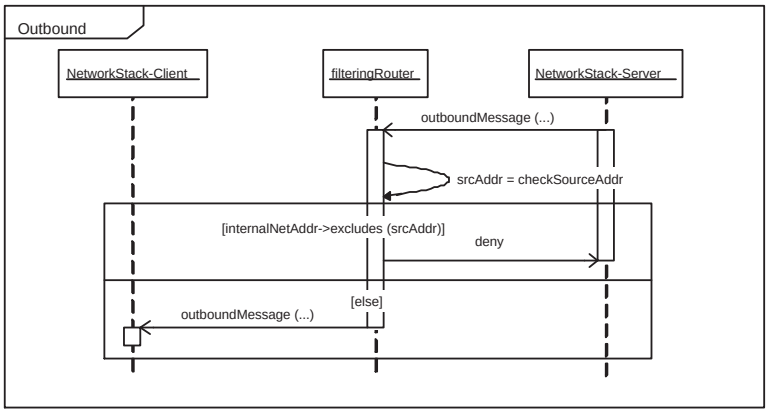
$$F_{initialSL}(s_j) = \frac{\sum_{i=1}^{n}\{b_i(I) \times T_{score}(b_i)\}}{n} \tag{24}$$

$$F_{SL}(s_j) = \frac{F_{initialSL}(s_j)}{\sum_{j=1}^{r} F_{initialSL}(s_j)} \tag{25}$$

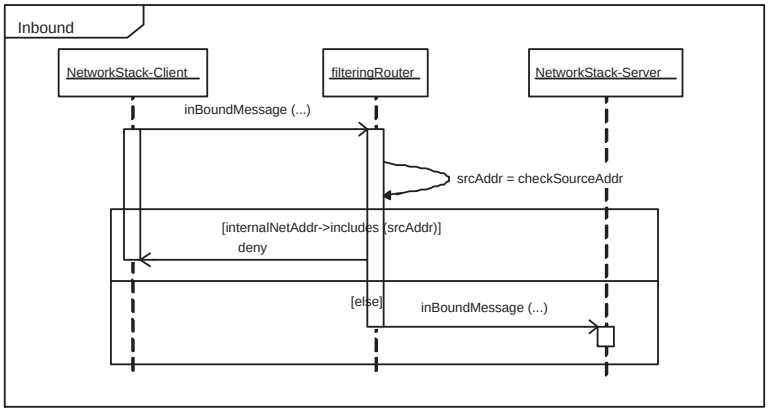
3 Example Application: Evaluating DoS Solutions

We now describe how to use our approach to predict the security level of two solutions for protecting against Denial of Service (DoS) attacks that can be launched at the user authentication mechanism of ACTIVE, an e-Commerce platform that was developed by the EU EP-27046-ACTIVE project [5]. Here we evaluate two such solutions – a cookie solution and a filtering mechanism. The cookie solution adds a patch to the network stack software that keeps track of sessions and their states. It begins by sending a cookie to the client. If the client does not respond within a short period of time, the cookie expires and the client must re-start the request for a connection. If the client responds in time, the SYN-ACK message is sent and the connection is set up. Adding the cookie message makes it unlikely that an attacker can respond in time to continue setting up the connection. If the client address has been spoofed, the client will not respond in any event. The filtering mechanism works a bit differently. The filtering mechanism has an outbound and an inbound part (Figures 2(a) and 2(b)) that checks the source address (srcAddr) against a set of accepted source IP addresses stored in internalNetAddr. The filtering mechanism is implemented on the server side (usually on a firewall or an Internet router) and configured to block unauthorized connection attempts.

A decision maker (truster) A needs help to choose between the two security solutions. For this purpose A seeks help of information sources regarding anticipated number of DoS attacks for the two solutions. In our example, we have five information sources; one honeypot [6] and four domain experts from a pool of 18 domain experts.



(a) Outbound



(b) Inbound

Fig. 2. Filter mechanism

The four chosen domain experts are denoted as b_4, b_6, b_{15}, b_{18} and the honeypot is denoted by $b_{honeypot}$. These five information sources provide information on the anticipated number of DoS attacks for the two involved solutions to A. The truster A has complete trust in the abilities of honeypot to provide accurate and correct information on the potential number of successful DoS attacks and therefore $T_{score}(b_{honeypot}) = 1$. Elicitation of expert judgments are done using a combined knowledge level and expertise level questionnaire as shown in Table 2.

The reference knowledge domain model was created by a third party who has experience with secure systems; thus, the set of external sources X has only one member x_1. Here the relevant knowledge domains are *security management (50%), design (10%), network management (20%), database (15%)*, and *developer (5%)*. The importance vector, obtained using Equation 1, is $W_{Kimp}(x_1) = [0.5, 0.2, 0.15, 0.1, 0.05]$. Since we have

Expert no.	Calibration variable	Information provided
4	level of expertise	medium
	years of relevant of education	Bsc
	years of experience from industry	0
	role experience	database, security management
6	level of expertise	low
	years of relevant of education	Bsc
	years of experience from industry	0
	role experience	database
15	level of expertise	high
	years of relevant of education	Bsc
	years of experience from industry	0
	role experience	designer, developer, security management
18	level of expertise	low
	years of relevant of education	Bsc
	years of experience from industry	0.5
	role experience	developer

Table 2. The combined knowledge and expertise level questionnaire and the information provided

only one external source x_1, we obtain, $W_{aggrgatedKimp}(X) = W_{allKimp}(X) = W_{Kimp}(x_1)$. The knowledge domains are already normalized and we do not need to normalize the elements in the vector $W_{aggrgatedKimp}(X)$. Hence, $(W_{refKnowledgeDomainScore}(X) = W_{aggrgatedKimp}(X) = [0.5, 0.2, 0.15, 0.1, 0.05]$.

An external source y_1 assesses relative weights for each knowledge domain for each information source. Here y_1 is same as x_1 who assessed the importance weights in reference knowledge domain model. The weights that each of the experts has for the knowledge domains are: for b_4, 85% on security management and 15% on database; for b_6, 100% on database; for b_{15}, 60% on design, 30% on developer, and 10% on security management; for b_{18}, 100% on developer. Equation 6 gives the information source knowledge domain vectors for the sources as follows.

- $W_{Kis}(y_1(b_4)) = [0.85, 0.0, 0.15, 0.0, 0.0]$
- $W_{Kis}(y_1(b_6)) = [0.0, 0.0, 1.0, 0.0, 0.0]$
- $W_{Kis}(y_1(b_{15})) = [0.1, 0.0, 0.0, 0.6, 0.3]$
- $W_{Kis}(y_1(b_{18})) = [0.0, 0.0, 0.0, 0.0, 1.0]$

Since there is only one external source y_1 $(= x_1)$ in the set Y of external sources providing assessment on the information sources, we have $W_{isKnowledgeDomainScore}(Y(b_i)) = W_{aggregatedKis}(Y(b_i)) = W_{Kis}(y_1(b_i))$, for $i = 4, 6, 15, 18$.

The knowledge score for each information source is derived using Equation 11:

- $K_{score}(b_4) = 0.85 * 0.5 + 0 * 0.2 + 0.15 * 0.15 + 0 * 0.1 + 0 * 0.05 \approx 0.45$
- $K_{score}(b_6) = 0 * 0.5 + 0 * 0.2 + 1 * 0.15 + 0 * 0.1 + 0 * 0.05 = 0.15$
- $K_{score}(b_{15}) = 0.1 * 0.5 + 0 * 0.2 + 0 * 0.15 + 0.6 * 0.1 + 0.3 * 0.05 \approx 0.13$
- $K_{score}(b_{18}) = 0 * 0.5 + 0 * 0.2 + 0 * 0.15 + 0 * 0.1 + 1.0 * 0.05 = 0.05$

The level of expertise of an information source is derived using the calibration variables described in Table 2. The external expert x_1 gives the relative importance values for calibration variables and the weights of categories for each calibration variable. Hence, $X' = Y = X = \{x_1\}$. We use three calibration variables to determine level of expertise – *level of experience* denoted by l_1, *years of relevant education* denoted by l_2, and *years of experience from industry* denoted by l_3. This gives the following vectors of categories for the three calibration variables: (i) $l_1 = [low, medium, high]$, (ii) $l_2 = [Bsc]$, (iii) $l_3 = [no\_of\_year]$ Suppose the expert x_1 assigns the following weights for the categories of calibration variables: $w_c(l_1(low)) = 0.2$, $w_c(l_1(medium)) = 0.5$, $w_c(l_1(high)) = 1.0$, $w_c(l_2(Bsc)) = 0.2$ and $w_c(l_3(no\_of\_year)) = 0.2$ for each year of industrial experience. Therefore,

- $W_{aggregatedC}(x_1(l_1)) = [0.2, 0.5, 1.0]$
- $W_{aggregatedC}(x_1(l_2)) = [0.2]$
- $W_{aggregatedC}(x_1(l_3)) = [0.2]$

Suppose the importance value given to the calibration variables by the external expert are 0.3 for *level of experience*, 0.2 for *years of relevant education*, and 0.5 for *years of experience from industry*. Therefore, $W_{l_1} = 0.3$, $W_{l_2} = 0.2$ and $W_{l_3} = 0.5$.

We then look at the information about categories of calibration variables provided by the information sources b_4, b_6, b_{15}, b_{18} in the questionnaire. We do not need to aggregate these scores as we are considering assessment from only one external expert. The scores for the four sources are as follows:

- $W_{isCat}(b_4(l_1)) = [0,1,0]$, $W_{isCat}(b_4(l_2)) = [0,0,1,0,0,0]$, $W_{isCat}(b_4(l_3)) = [0]$.
- $W_{isCat}(b_6(l_1)) = [1,0,0]$, $W_{isCat}(b_6(l_2)) = [0,0,1,0,0,0]$, $W_{isCat}(b_6(l_3)) = [0]$.
- $W_{isCat}(b_{15}(l_1)) = [0,0,1]$, $W_{isCat}(b_{15}(l_2)) = [0,0,1,0,0,0]$, $W_{isCat}(b_{15}(l_3)) = [0]$.
- $W_{isCat}(b_{18}(l_1)) = [0,0,1]$, $W_{isCat}(b_{18}(l_2)) = [0,0,1,0,0,0]$, $W_{isCat}(b_{18}(l_3)) = [0.5]$.

Using the above information, the evaluator calculates the expertise score of the information sources using Equations 21 and 22.

- $E_{score}(b_4) = 0.3*0.5 + 0.2*0.2 + 0.5*0 = 0.15 + 0.04 + 0 = 0.19$
- $E_{score}(b_6) = 0.3*0.2 + 0.2*0.2 + 0.5*0 = 0.06 + 0.04 + 0 = 0.10$
- $E_{score}(b_{15}) = 0.3*1.0 + 0.2*0.2 + 0.5*0 = 0.3 + 0.04 + 0 = 0.34$
- $E_{score}(b_{18}) = 0.3*1.0 + 0.2*0.2 + 0.5*(0.5*0.2) = 0.3 + 0.04 + 0.5 = 0.84$

The knowledge and expertise scores are combined into an information source trustworthiness weight using Equation 23. The truster A has assigned relative importance of the knowledge and expertise score as 0.6 and 0.4 respectively. Recall that $T_{score}(b_{honeypot}) = 1.0$. Thus, the trustworthiness score for the experts $b_{honeypot}$, b_4, b_6, b_{15}, b_{18} are derived as,

- $T_{score}(b_{honeypot}) = 1.0$.
- $T_{score}(b_4) = 0.6*0.45 + 0.4*0.19 = 0.27 + 0.076 = 0.346$.
- $T_{score}(b_6) = 0.6*0.15 + 0.4*0.1 = 0.09 + 0.04 = 0.130$.
- $T_{score}(b_{15}) = 0.6*0.13 + 0.4*0.34 = 0.078 + 0.136 = 0.214$.
- $T_{score}(b_{18}) = 0.6*0.05 + 0.4*0.84 = 0.03 + 0.336 = 0.366$.

Now we predict the security level of the two solutions of the DoS problem. Let us denote the *cookie solution* by s_1 and *filter mechanism* by s_2. To derive the security level for s_1 and s_2, the information provided by the different information sources are interpreted and combined with their trustworthiness score using the Equations 24 and 25 mentioned in Section 2.4. The honeypot reports 1.5 average monthly successful attack for cookie solution (s_1) and 4.0 average monthly successful attack for filter mechanism (s_2). The information provided by the experts are as follows:

- $b_4(s_1) = medium,\ b_4(s_2) = low$
- $b_6(s_1) = medium,\ b_6(s_2) = medium$
- $b_{15}(s_1) = medium,\ b_{15}(s_2) = low$
- $b_{18}(s_1) = high,\ b_{18}(s_2) = low$

In order to calculate the security level from these pieces of information, the information must be at the same level of abstraction and comparable. The honeypot reports less number of average monthly successful attack for cookie solution than filter mechanism. This shows that according to the information source $b_{honeypot}$, the cookie solution s_1 has higher security level. To measure this level, we transform the average monthly successful attack inversely and the reciprocal of this average value is used to calculate the security level. This gives: $b_{honeypot}(s_1) = 1/1.5 = 0.667$ and $b_{honeypot}(s_2) = 1/4.0 = 0.25$. For the other information sources, we assign 0.2 for the level *low*, 0.5 for the level *medium* and 1.0 for the level *high*. Hence, the initial security level of the security solutions are evaluated as

- $F_{initialSL}(s_1) = (0.667*1.0+0.5*0.346+0.5*0.130+0.5*0.214+1.0*0.366)/5 = 0.2756$
- $F_{initialSL}(s_2) = (0.25*1.0+0.2*0.346+0.5*0.130+0.2*0.214+0.2*0.366)/5 = 0.10004$

Using Equation 25 the initial security level is updated to relative security level for each solution, which gives

- $F_{SL}(s_1) = 0.2756/(0.2756+0.10004) \approx 0.734$
- $F_{SL}(s_2) = 0.10004/(0.2756+0.10004) \approx 0.266.$

This relative security level is a prediction and should not be considered as the actual security level, but rather an expression of the difference in security level between the two DoS solutions. The actual security level depends on many uncertain factors, such as future attacks, changes in the security environment, relevant operational procedures, maintenance strategy, the resources available etc. What we can infer from the result is that the cookie solution is a much better choice when it comes to security solutions of DoS attacks than the filter mechanism. The relative difference between the two solutions is 2.76 ($\frac{F_{SL}(s_1)}{F_{SL}(s_2)} = \frac{0.734}{0.266} \approx 2.759$), which means that the cookie solution is almost three times a better choice than the filter mechanism.

3.1 Validation of Example Application Results

DoS attacks are often performed using legitimate protocols and services; the malicious activities differ from legitimate ones only by intent and not by content. Since it is hard

to measure intent, many of the existing DoS solutions do not offer a proper defense. In [7] Karig and Lee gives an overview of common DoS attacks and potential countermeasures for DoS attacks. In this context, the filtering mechanism is categorized as a network device level countermeasure while the cookie solution is categorized as an OS level countermeasure. A network device level DoS solution provides measures to protect against potential misuse of a communication protocol. Thus, the protection is often on the IP or transport layer and hence there are possible ways around the mechanism, such as those discussed in [7]. The main shortage of filtering mechanisms are their inability to filter out spoofed packets [7]. There are, however, more efficient filtering mechanisms available, such as the one discussed in [8]. The other DoS solution discussed in this paper, the cookie solution, operates on the OS level. An OS level DoS solution integrates protection into the way a protocol is implemented in a particular operating system. Thus, the measure is deployed on the source (target) and refers to a host-based protection solution. Hence, the cookie solution represents a more defense-in-depth DoS solution than the filtering mechanism. Furthermore, the cookie solution discussed in this paper is a SYN cookie, which has been well tested and is well understood. SYN cookies have also been incorporated as a standard part of Linux and Free BSD and are recognized as one of the most effective DoS mechanisms [9].

In general, a DoS solution should be effective, transparent to existing Internet infrastructure, have low performance overhead, be invulnerable to attacks aimed at the defense system, be incrementally deployable and have no impact on the legitimate traffic [10]. The filtering mechanism is somewhat effective in stopping attacks on the spot. It is not transparent to existing Internet infrastructure and results in some performance overhead. The filter mechanism can also be vulnerable to attacks due to its scanning of each packet and hence may have impact on legitimate traffic. However, the mechanism can be incrementally deployed. The cookie solution is documented to be effective against DoS attacks, but has been demonstrated to be somewhat unable to detect and prevent against zombie attacks. The mechanism is transparent to the network infrastructure, but leads to some performance overhead, but in practice no impact on legitimate traffic. The cookie solution is already included in some operating systems and is easy to deploy. Thus, we can conclude that the cookie solution is a better choice than filtering mechanism for DoS attacks. Our trust-based information aggregation approach also shows that the cookie solution is approximately 2.76 times better than the filtering mechanism.

4 Related Work

Jøsang [11, 12] proposed a model for trust based on a general model for expressing relatively uncertain beliefs about the truth of statements. Cohen et al. [13] proposed an alternative, more differentiated concept of trust called Argument-based Probabilistic Trust model (APT). Yahalom et al. [14, 15] proposed a formal model for deriving new trust relationships from existing ones. Beth et al. [16] extended the ideas presented by Yahalom et al. to include relative trust. Xiong and Liu [17] presented a coherent adaptive trust model for quantifying and comparing the trustworthiness of peers based on a transaction-based feedback system. Bacharach and Gambetta [18] defined trust

as a particular belief, which arises in games with a certain payoff structure. Purser [19] presented a simple, graphical approach to model trust and discussed the relationship between trust and risk. Ray and Chakraborty [20] and Ray et al. [21] described the factors on which trust depends, showed how to quantify these factors and obtain a quantitative value for trust. Other works include logic-based formalisms of trust [22–25].

Littlewood et al. [26] was one of the earliest works on measuring operational security. Subsequently, Ortalo et al. in [27] proposed a quantitative model for known Unix security vulnerabilities using a privilege graph. Madan et al. [28] discussed how to quantify security attributes of software systems using traditional reliability theory for modeling random processes, such as stochastic modeling and Markov analysis. Jonsson and Olovsson [29] looked at the problem in a more practical way by analyzing attacker behavior through controlled experiments.

Several efforts have been devoted to developing structured and systematic security risk assessment approaches. The three main approaches are the OCTAVE [30], CRAMM [31] and the CORAS frameworks [32]. Security management standards aid in the overall and detailed management of security in an organization. The most important standards in this area are the ISO/IEC 27002:2005 Information technology – Code of Practice for information security management [33], ISO/IEC TR 13335:2004 Information technology – Guidelines for management of IT Security [34] and the Australian/New Zealand standard for risk management AS/NZS 4360:2007 [35].

TCSEC is the oldest known standard developed in the U.S. for evaluation and certification of information security in IT products. Subsequently, the European countries collaborated and produced their own standard ITSEC. The International Organization for Standardization (ISO) developed the Common Criteria, as a response to the various types of evaluation criteria that were developed by different nations, which has replaced TCSEC and ITSEC.

Our work refines that proposed by Houmb et al. [36] by (i) extending the sophistication with which knowledge score, experience score, and relative trustworthiness is calculated and (ii) allowing for the direct evaluation and comparison of security solutions using whatever security-related information that is available.

5 Conclusion

In this article we present a trust-based information aggregation approach to predict security level of security solutions. We have proposed a quantitative approach for evaluating the trustworthiness of sources and using this information to predict the security level of a solution. We have demonstrated our approach for predicting the security level of two solutions used for preventing DoS attacks on an example .NET e-commerce system. Our results help validate that one solution is superior than the other for preventing DoS attacks. Future work includes controlled experiments, and eventually a case study, to gain realistic experience with the current version of the trust-based information aggregation approach. Investigating how to reduce the subjectivity of the approach also needs to be investigated. Future work also involves transitioning from the deterministic trust model to a probabilistic one which will allow reasoning with uncertainty and implementing such a model using existing Bayesian Belief Network tools, such as HUGIN.

Incorporating such a trust model in other applications, such as social networks, is also planned for the future.

References

1. : ISO 15408:1999 Common Criteria for Information Technology Security Evaluation. Version 2.1, CCIMB–99–031, CCIMB-99-032, CCIMB-99-033 (August 1999)
2. : Common Criteria for Information Technology Security Evaluation (2010) http://en.wikipedia.org/wiki/Common_Criteria.
3. Cooke, R.: Experts in Uncertainty: Opinion and Subjective Probability in Science. Oxford University Press (1991)
4. Goossens, L., Harper, F., Kraan, B., Meacutetivier, H.: Expert Judgement for a Probabilistic Accident Consequence Uncertainty Analysis. Radiation Protection and Dosimetry **90**(3) (2000) 295–303
5. EU Project EP-27046-ACTIVE: EP-27046-ACTIVE, Final Prototype and User Manual, D4.2.2, Ver. 2.0, 2001-02-22. (2001)
6. Østvang, M.E.: The Honeynet Project, Phase 1: Installing and Tuning Honeyd using LIDS (2003) Project assignment, Norwegian University of Science and Technology.
7. Karig, D., Lee, R.: Remote Denial of Service Attacks and Countermeasures. Technical report CE-L2001-002, Department of Electrical Engineering, Princeton University (October 2001)
8. Barkley, A., Liu, S., Gia, Q., Dingfield, M., Gokhale, Y.: A Testbed for Study of Distributed Denial of Service Attacks (WA 2.4). In: Proceedings of the IEEE Workshop on Information Assurance and Security. (June 2000) 218–223
9. Bernstein, D.: SYN Cookies (Accessed November 2006) http://crypto/syncookies.html.
10. Lin, S., Chiueh, T.: A Survey on Solutions to Distributed Denial of Service Attacks. Technical report RPE TR-201, Department of Computer Science, Stony Brook University (September 2006)
11. Jøsang, A.: A Subjective Metric of Authentication. In: Proceedings of the 5th European Symposium on Research in Computer Security. (September 1998) 329–344
12. Jøsang, A.: An Algebra for Assessing Trust in Certification Chains. In: Proceedings of the 1999 Network and Distributed Systems Security Symposium. (February 1999)
13. Cohen, M., Parasuraman, R., Freeman, J.: Trust in Decision Aids: A Model and a Training Strategy. Technical Report USAATCOM TR 97-D-4, Cognitive Technologies Inc. (1997)
14. Yahalom, R., Klein, B., Beth, T.: Trust Relationship in Secure Systems: A Distributed Authentication Perspective. In: Proceedings of the IEEE Symposium on Security and Privacy. (May 1993) 150–164
15. Yahalom, R., Klein, B., Beth, T.: Trust-based Navigation in Distributed Systems. Computing Systems **7**(1) (Winter 1994) 45–73
16. Beth, T., Borcherding, M., Klein, B.: Valuation of Trust in Open Networks. In: Proceedings of the 3rd European Symposium on Research in Computer Security. (November 1994) 3–18
17. Xiong, L., Liu, L.: A Reputation-Based Trust Model For Peer-To-Peer Ecommerce Communities. In: Proceedings of the IEEE Conference on E-Commerce. (June 2003) 275–284
18. Bacharach, M., Gambetta, D.: Trust as Type Identification. In: Trust and Deception in Virtual Societies. Kluwer Academic Publishers (2000) 1–26
19. Purser, S.: A Simple Graphical Tool For Modelling Trust. Computers & Security **20**(6) (September 2001) 479–484
20. Ray, I., Chakraborty, S.: A Vector Model of Trust for Developing Trustworthy Systems. In: Proceedings of the 9th European Symposium on Research in Computer Security. (September 2004) 260–275

21. Ray, I., Ray, I., Chakraborty, S.: An Interoperable Context Sensitive Model of Trust. Journal of Intelligent Information Systems **32**(1) (2009) 75–104
22. Abdul-Rahman, A., Hailes, S.: Supporting Trust in Virtual Communities. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences. (January 2000) 4–7
23. Burrows, M., Abadi, M., Needham, R.: A Logic of Authentication. ACM Transactions on Computer Systems **8**(1) (February 1990) 18–36
24. Jones, A., Firozabadi, B.: On the Characterization of a Trusting Agent – Aspects of a Formal Approach. In: Trust and Deception in Virtual Societies. Kluwer Academic Publishers (2000) 157–168
25. Jajodia, S., Samarati, P., Subrahmanian, V.: A Logical Language for Expressing Authorizations. In: Proceedings of the IEEE Symposium on Security and Privacy. (May 1997) 31–42
26. Littlewood, B., Brocklehurst, S., Fenton, N., Mellor, P., Page, S., Wright, D., Dobson, J., McDermid, J., Gollmann, D.: Towards Operational Measures of Computer Security. Journal of Computer Security **2** (1993) 211–229
27. Ortalo, R., Deswarte, Y.: Experiments with Quantitative Evaluation Tools for Monitoring Operational Security. IEEE Transaction on Software Engineering **5**(25) (September/October 1999) 633–650
28. Madan, B., Popstojanova, K.G., Vaidyanathan, K., Trivedi, K.: Modeling and Quantification of Security Attributes of Software Systems. In: Proceedings of the International Conference on Dependable Systems and Networks. (June 2002) 505–514
29. Jonsson, E., Olovsson, T.: A Quantitative Model of the Security Intrusion Process based on Attacker Behavior. IEEE Transaction on Software Engineering **4**(25) (April 1997) 235–246
30. Alberts, C., Behrens, S., Pethia, R., Wilson, W.: Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) Framework, Version 1.0. Technical report, Software Engineering Institute, Carnegie Mellon University (June 1999)
31. Barber, B., Davey, J.: The Use of the CCTA Risk Analysis and Management Methodology CRAMM in Health Information Systems. In: Proceedings of the International Medical Informatics Conference. (September 1992) 1589–1593
32. CORAS (2000–2003): IST-2000-25031 CORAS: A Platform for Risk Analysis of Security Critical Systems (Accessed February 2006)
33. International Organization for Standardization (ISO/IEC): ISO/IEC 27002:2005 Information Technology – Security Techniques – Code of Practice for Information Security Management (2000)
34. International Organization for Standardization (ISO/IEC): ISO/IEC TR 13335:2004 Information Technology – Guidelines for Management of IT Security (2001)
35. Australian/New Zealand Standards: AS/NZS 4360:2007 Risk Management (2004)
36. Houmb, S., Ray, I., Ray, I.: Estimating the Relative Trustworthiness of Information Sources in Security Solution Evaluation. In: Proceedings of the 4th International Conference on Trust Management. (May 2006) 135–149