A Method for Developing Algorithms for Assessing Cyber-Risk Cost

Gencer Erdogan, Atle Refsdal and Fredrik Seehusen
SINTEF Digital
Email: {firstname.lastname@sintef.no}

Alejandra Gonzalez
Aon
Email: {alejandra.gonzalez@aon.it}

***Abstract*—We present a method for developing executable algorithms for quantitative cyber-risk assessment. Exploiting techniques from security risk modeling and actuarial approaches, the method pragmatically combines use of available empirical data and expert judgments. The input to the algorithms are indicators providing information about the target of analysis, such as suspicious events observed in the network. Automated execution of the algorithms facilitates continuous assessment.**

I. Introduction

Managers and decision makers need to know the cyber-risk they face in order to decide how to deal with such risks. Quantified estimates allow risks to be weighted against the cost of available countermeasures. However, providing quantified assessments of cyber-risk cost is difficult, due to factors such as the technical and changing nature of cyber-risks, the variations in potential cost resulting from incidents, and lack of suitable empirical data.

The contribution of this paper is a method for developing executable algorithms for quantitative assessment of expected cyber-risk cost. Our aim is to provide a method that documents risk models in a comprehensive format, is feasible without requiring prohibitive effort and facilitates exploitation of available empirical data sources. The intended users of the method are professionals interested in developing new algorithms, such as consultants or dedicated cyber-risk experts in larger organizations. The final end users of the algorithms developed by the method will typically include decision makers responsible for selecting countermeasures to implement.

In Section II, we give an overview of the method, which consists of four steps. In the next four sections, we describe each step, illustrated by a running example. Section VII relates our work to other approaches, while Section VIII concludes.

II. Overview of the method

Fig. 1 shows an overview of the method. We assume that the purpose, scope and target of analysis have already been established, and that risk levels will be defined in terms of frequency and monetary cost of incidents. Roughly speaking, this means that the context establishment of the risk analysis process has been performed [1]. The white document symbols represent inputs and outputs to the steps. Those attached to the arrow from one step to another represent output from the preceding step that serve as input to the next step.

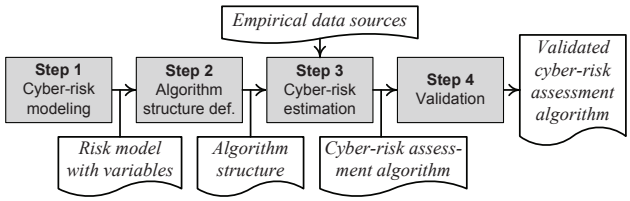

Fig. 1. Overview of method for developing cyber-risk assessment algorithms

In Step 1, we develop a graphical risk model that documents the assets, risks, threats and vulnerabilities. We also identify indicators that capture dynamic factors assumed to influence the risk level. Each indicator defines an input to the algorithm. No estimates of likelihood or consequence values are made at this point; such estimates are represented by variables. In Step 2, we define the algorithm structure based on the risk model. In Step 3, we exploit available empirical data sources to complete the algorithm by assigning estimates to the variables identified in Step 1 and defining the impact of the indicator values on the assessments. Finally, in Step 4 we validate the algorithm by executing it on selected sets of inputs and checking whether the outputs are plausible with respect to existing empirical data and expert judgments.

III. Step 1: Cyber-risk modeling

Step 1 consists of two sub-steps. In Step 1.1, the user creates a risk model and defines variables for the likelihood and consequence values to be estimated in Step 3. In Step 1.2, the user identifies indicators with respect to the risk model to support the risk estimation. The output of this step is a risk model with variables and indicators, which is used as basis for building the structure of the risk assessment algorithm in Step 2.

A. Step 1.1: Risk modeling

There are many different kinds of modeling languages for describing risks. In our method, we use CORAS [2], which is a comprehensive framework for model-driven risk analysis consisting of a language, a tool, and a method. Fig. 2 gives an overview of the CORAS notation. Threats, threat scenarios, unwanted incidents, assets, relations and vulnerabilities are collectively used to create CORAS risk models, which document risks as well as events and circumstances that can cause

The final publication is available at IEEEXplore via https://doi.org/10.1109/QRS.2017.29

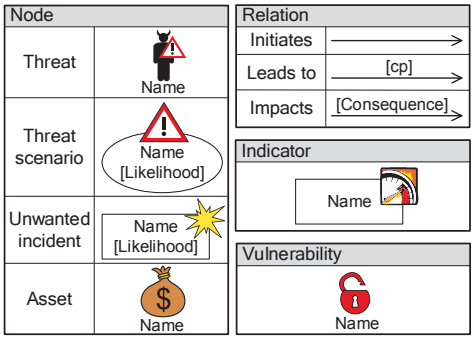

Fig. 2. Overview of the CORAS notation (cp=conditional probability).

risks. Notice that the different relations are used to connect different nodes: the *initiates* relation goes from a threat to a threat scenario or an unwanted incident. The *leads to* relation goes from a threat scenario or an unwanted incident to a threat scenario or an unwanted incident. The *impacts* relation goes from an unwanted incident to an asset.

To support risk estimation, CORAS uses likelihood values, conditional probabilities, and consequence values on certain nodes and relations as illustrated in Fig. 2. The indicator construct, which is not part of the standard CORAS notation, is introduced in our approach to capture dynamic factors that support risk estimation. Indicators are discussed in detail in Section III-B.

Fig. 3 shows a CORAS risk model of a session hijacking attack in the context of web-applications. This risk model is one of 10 risk models we developed in the WISER project [3]. These risk models were not developed for a particular target of analysis, but primarily intended for an arbitrary European SME. We will use the risk model in Fig. 3 as a running example throughout the rest of this paper. The model describes an Hacker carrying out session fixation or accesses, intercepts, or modifies HTTP cookies in order to hijack a session. The risk is that a session is hijacked, which has an impact on confidentiality.

Having created the risk model, the user needs to define identifiers for the assets, threat scenarios, and unwanted incidents, as well as variables for the likelihood values, conditional probabilities, and the consequence values. This is carried out using the naming convention in Table I. For example, we see that the two threat scenarios in Fig. 3 are identified by *S1* and *S2*, respectively, and that their corresponding likelihood variables are defined as *l_S1* and *l_S2*, respectively.

B. Step 1.2: Identifying indicators

In order to identify dynamic factors that influence the variables of the risk model, we identify so-called indicators and attach them to the relevant risk-model element. An *indicator* is a piece of information that is relevant for assessing the risk level. An indicator may be assigned to any risk-model element.

For example, consider the vulnerability *Improper management of session time and state* in Fig. 3. A potential indicator for this vulnerability could be that the target under analysis

TABLE I
NAMING CONVENTIONS FOR DEFINING LIKELIHOOD AND CONSEQUENCE VARIABLES. THE LETTERS *x* AND *y* REPRESENT INTEGERS.

Name	Meaning
Ax	Asset x
Sx	Scenario x ("S" means threat scenario)
Ux	Incident x ("U" means unwanted incident)
l_Ux	Likelihood of Ux
l_Sx	Likelihood of Sx
c_Ux_Ay	Consequence of Ux for Ay
cp_Sx_to_Sy	Conditional probability of Sx leading to Sy
cp_Sx_to_Uy	Conditional probability of Sx leading to Uy

treats invalid sessions as valid. If we gather information indicating that there are invalid sessions treated as valid sessions, we may argue that the target under analysis is most likely vulnerable to *Improper management of session time and state*. The indicators supporting likelihood estimation are defined as yes/no questions, for example, *Are any invalid sessions treated as valid?*

Indicator values may be obtained by different means. For example, in some cases it is sufficient to base the indicator value on expert knowledge provided by a representative of the target under analysis, while in other situations it may be necessary to implement sensors at the network layer in order to derive indicator values based on continuous network monitoring. We differentiate between four types of indicators.

- *Business configuration (blue):* Indicator values are obtained by asking business related questions. The indicator values are thus based on expert knowledge.
- *Test (green):* Indicator values are obtained by carrying out software tests. The indicator values are thus based on test results.
- *Network-layer monitoring (yellow):* Indicator values are obtained by monitoring the network layer.
- *Application-layer monitoring (red):* Indicator values are obtained by monitoring the application layer.

Fig. 3 illustrates all except network-layer indicators. The reader is referred to [4] for a set of guiding questions to help identify indicators.

In summary, the output of Step 1 is a CORAS risk model capturing risks, likelihood and consequence variables, as well as indicators for collecting relevant information to support the risk estimation.

IV. Step 2: Algorithm structure definition

In Step 2, we define the part of the algorithm that can be established from the CORAS diagram, without performing any estimation of likelihoods, consequences or impact from indicators. The result is an algorithm structure (skeleton), which will be extended to a complete algorithm in Step 3.

We follow an actuarial approach, where the likelihood (frequency) of unwanted incidents and the consequence (economic loss) are separately modeled through the probabilistic framework of *Bayesian Networks* (BN) [5].

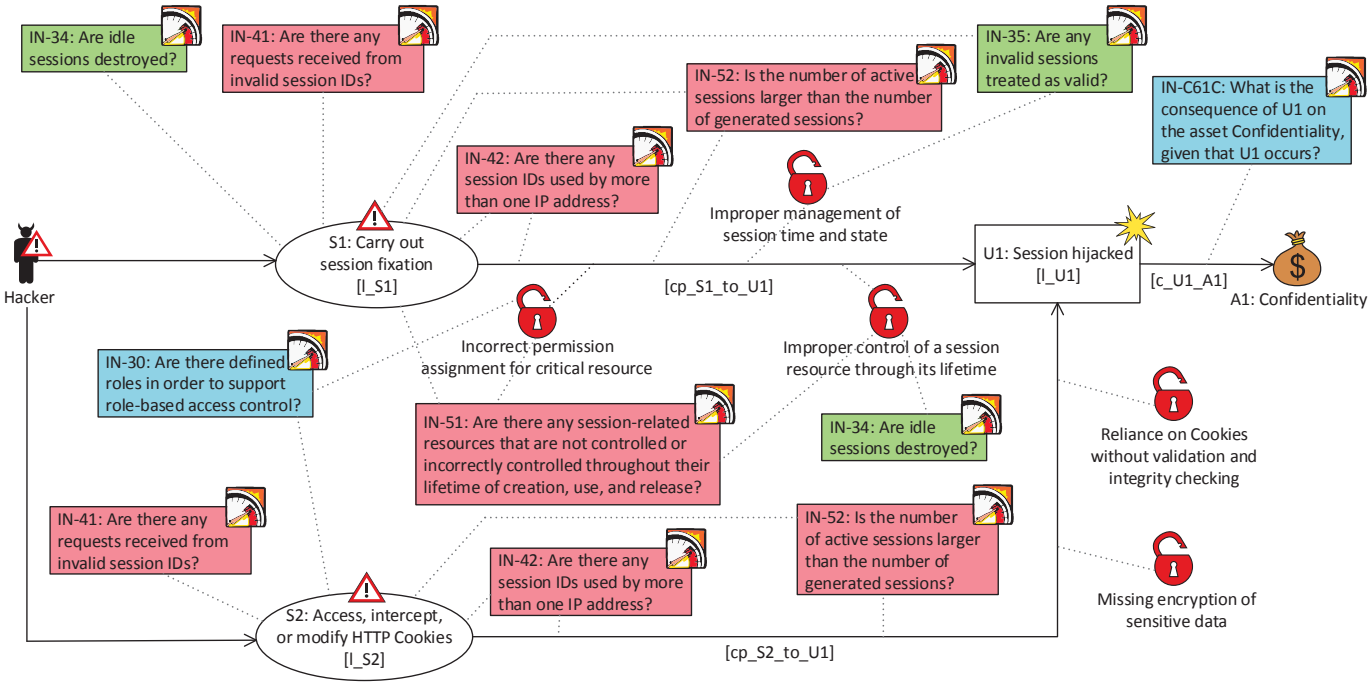

Fig. 3. Cyber-risk model *Session hijacking* expressed in CORAS.

The algorithm is implemented using the **R** programming language [6] for statistical computing and the HydeNet package [7] which provides a powerful interface to construct BNs and perform inference. The package handles hybrid BNs [8], that is networks where the random variables are not bound to be discrete or (conditionally) Gaussian. The underlying calculations are performed by MCMC (Markov-Chain-Monte-Carlo) using the '`rjags`' package.

Understanding the underlying principles of our approach should not require prior knowledge about the **R** programming language. Therefore we will not show the actual **R** code. For detailed guidelines on how to create an **R** script from a CORAS model, we refer to [4].

A. Building a BN skeleton

In our approach, the frequency of unwanted incidents is calculated following the logic of the CORAS model. The fist step is to define a BN skeleton based on the structure of the CORAS model. Fig. 4 shows the BN skeleton reflecting the CORAS model in Fig. 3. A risk captured in a CORAS diagram (by an *impacts* relation from an unwanted incident to an asset) is represented by a childless node in the BN (R1 in Fig. 4). The overall goal is to compute a risk level for risk nodes, as a function of indicators. Any risk node has two parent nodes: one representing the frequency of the unwanted incident and another representing the consequence for the asset. In our example, the risk node R1 has the parent nodes l_U1 and c_U1_A1, representing the frequency of the incident U1 and its consequence for the asset A1, respectively.

The frequency node of an unwanted incident has a parent node for each incoming *leads to* relation to the incident in the CORAS diagram, representing the likelihood contribution from each incoming relation. For example, node l_U1 in Fig. 4 has two parent nodes: l_U1S1 and l_U1S2.

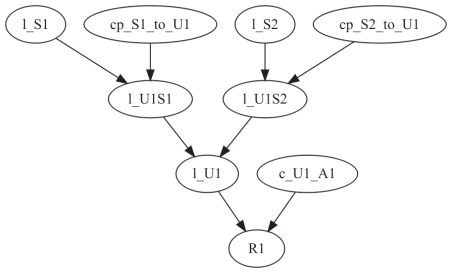

Fig. 4. BN skeleton for CORAS model in Fig. 3.

The likelihood contribution from each *leads to* relation depends on the likelihood of its threat scenario (the source node) and the conditional probability that an occurrence of this threat scenario will lead to the unwanted incident. Therefore, the node l_U1S1 depends on the root nodes l_S1 (likelihood of scenario S1) and cp_S1_to_U1 (the conditional probability that an occurrence of S1 will lead to U1). Similarly l_U1S2 depends on l_S2 and cp_S2_to_U1.

Notice that l_U1S1 and l_U1S2 represent internal nodes in the BN structure that do not occur as variables in the CORAS diagram. Moreover, the indicators, which represent the input to the final algorithm, are not represented in the BN structure. They will be used to compute the values for the parentless nodes, as further explained in Section V.

B. Nodes representing frequency of unwanted incidents

For each unwanted incident in a CORAS model, a corresponding frequency node is included in the BN structure (node l_U1 in Fig. 4). The probability distribution assigned to l_U1 is defined as follows:

$$\begin{aligned} \text{l\_U1} &= \text{l\_U1S1 + l\_U1S2} \\ &= \text{l\_S1} \cdot \text{cp\_S1\_to\_U1 + l\_S2} \cdot \text{cp\_S2\_to\_U1}. \end{aligned}$$

Notice that node l_U1 is deterministic, since its value at each step of the simulation is calculated by a formula from the values of its parent nodes.

The likelihoods of the parentless ancestor nodes of l_U1 (l_S1, l_S2, cp_S1_to_U1 and cp_S2_to_U1 in Fig. 4) are represented by uniform distributions whose extremes depend on the indicators affecting the nodes. Uniform distributions were chosen to ease the estimate elicitation. CORAS uses intervals for the same reason. The specific functions from indicator values to the extremes of the distributions are defined as a part of the estimation in Step 3.

C. Nodes representing consequence of incidents

Consequence nodes model the consequence, in terms of economic loss, generated by a unwanted incident (node c_U1_to_A1 in Fig. 4).

The main problem in identifying suitable parametric families of probability distributions to model the consequence (severity) of the losses generated by cyber-risk incidents is that there is very little historical data regarding such losses, which prevents use of standard fitting algorithms such as maximum likelihood.

The following distribution families are currently used in actuarial practice to model losses: lognormal, gamma, Weibull and other heavy-tailed (mixtures of) distributions [9]. The above families all contain at least two parameters; in general, the parameters are estimated from data using methods such as the maximum likelihood or the method of moments, see e.g. [10], [11]. This, however, requires a sufficient number of datapoints; as a rule of thumb, 50-100 datapoints seems a reasonable minimum. In scenario analysis performed for operational risk management, however, a different approach is followed. For a given risk, a two-parameter distribution is chosen for the consequence (e.g. the lognormal) and experts are requested to provide a *typical case* loss and a *worst case* loss. This provides the minimal amount of information required to describe the main features of the distribution, that is a value which is experienced frequently and a value which is extreme (experienced rarely). Usually, the typical case loss is identified with a location index, such as the median or the mode of the distribution (the mean is often considered a less stable location index, since it is influenced by the presence of outliers). The worst case loss is identified with a suitably large quantile of the distribution. This gives a nonlinear system of two equations in two variables (the parameters of the distribution), which can be solved by a Newton-like numerical approximation method [12], [13], [14].

Typically, in operational risk loss modelling one assigns a relatively small value to the typical case, while the worst case might be significantly larger. This entails that the ensuing distribution is characterised by strong asymmetry (skew) towards the upper right tail. This characteristic is in fact often observed for operational loss data.

We adopted the lognormal distribution for modeling consequence nodes. In modelling loss data, the lognormal distribution is observed to provide good fits in many cases; for this reason it is often used for modelling severity in operational risk and particularly for the scenario analysis component, see e.g. [11], [15].

As explained above, the median is typically considered in risk management, as it provides a more robust location index than the mean. To estimate the lognormal parameters, we assume one knows an estimate for the median (typical case) and an estimate for a high percentile (worst case), say corresponding to a probability level of p. This probability value can be obtained in several ways [9]. For simplicity, based on common practice, we adopted $p = 99.9\%$.

Typical and worst case loss information can be established by the user instantiating the algorithm for a specific context or organization, who might have more information regarding the context where these losses arise. Since this represents input to the algorithm, it is captured by the business configuration indicator attached to the *impacts* relation from U1 to A1 in the CORAS diagram.

D. Nodes representing risk level

Risk level nodes model the yearly aggregate loss distribution, R1 in our current example. The probability distribution assigned to R1 is defined as follows:

$$\text{R1} = \text{l\_U1} \cdot \text{c\_U1\_to\_A1}$$

In summary, the output of Step 2 is an **R** script representing the structure of the risk assessment algorithm, containing the rules for calculating the risk level of unwanted incidents, and hence their frequency and consequence. The script is incomplete at this point, as the probability distributions for the parentless nodes in the BN have not been defined yet, this is part of Step 3.

V. Step 3: Cyber-risk estimation

The purpose of Step 3 is to provide estimates for the parameters of the probability distributions of the parentless nodes of the BN skeleton defined in Step 2, as a function of the indicator values in the risk model.

In our experience, the information basis needed to estimate the parameters is not at hand or incomplete. Hence, we advocate a pragmatic approach where estimation based on expert judgment is complemented with factual statements.

In the running example, we need to estimate how often the threats scenarios leading up to session hijacking occur (that is, we need to provide numerical values for the parameters of the distributions of nodes l_S1, and l_S2), how likely it is that they succeed if initiated (provide values for the parameters of

nodes cp_S1_to_U1 and cp_S2_to_U1), and the consequence of the risk (parameter values for node c_U1_A1). Table II reports the indicators involved in this example.

As explained in Section IV, the probability distribution of node l_U1 is calculated based on the other variables in the model. However, in our experience, it is still a good idea to provide estimates of upper and lower bounds for this variable as well, as this can be useful later on in the validation step (Step 4).

TABLE II
DEPENDENCE ON INDICATORS OF PARENTLESS NODES.

Node	Related indicators
l_S1	IN-34, IN-41, IN-35, IN-51, IN-52, IN-42.
l_S2	IN-30, IN-41, IN-42, IN-52.
cp_S1_to_U1	IN-30, IN-34, IN-51, IN-35, IN-42, IN-52.
cp_S2_to_U1	IN-42, IN-52.

A. Step 3.1: Identifying data sources and facts

The purpose of Step 3.1 is to gain more information which can be used to estimate parameter values for frequency and consequence distributions, and to document this in a structured manner.

Step 3.1 has two main activities: (I) Search for documents (or other sources) that contain historical data and statistics of relevance to the risk model; (II) Search through each document and identify facts that can aid the estimation of likelihood and consequence parameters, and document these in a table containing the following information for each fact:

- *Id:* The identifier of the fact.
- *Source:* Reference to the a data source containing the fact.
- *Basis:* Description of the data basis from which the fact is derived (if applicable).
- *Fact:* Textual description of the fact.

In the development of the 10 algorithms, we identified about 40 facts from 15 different data sources found via Google. Two of these facts which are relevant to risk model of Fig. 3 are shown in Table II. Notice that the data of Table II is not specific to a particular organization. The reason for this is that the risk model of our running example is intended to apply for an arbitrary SME. However, company specific data could also be used, if available.

B. Step 3.2: Estimating baseline likelihood and consequence values

The purpose of Step 3.2 is to use the facts identified in Step 3.1 to estimate baseline values for the parameters of the likelihood and consequence distributions. The user documents the results in a table which contains the following information for each likelihood and consequence parameter in the risk model:

- *Name:* The name of the parameter to be estimated in the risk model.
- *Value:* The estimated value.
- *Description:* A description of the meaning of the parameter.
- *Rationale:* A justification of the estimated value based on previously identified facts (if possible).

For the uniform distributions representing the likelihood of threat scenarios one provides estimates for the minimum and maximum times per year a threat scenario occurs. In Table III, we give an example of an estimated baseline value for one of the variables in the risk model in the running example. The entries in the value column determine the parameters of the probability distributions at the respective nodes of the BN model. The values 0 and 25 for l_S1 are the extremes of a uniform distribution representing the number of attacks per year. Notice that in the rationale column, we referred to facts in Table II, but there are also many assumptions based on expert judgment due to lack of data. For the nodes representing consequence distributions, as explained in Section IV, parameters are obtained by providing estimates for the typical and worst case cost of an unwanted incident.

C. Step 3.3: Defining likelihood functions

The purpose of this step is to define the likelihood functions. These are documented in a table where the columns contain the indicator names and the likelihood parameter to be estimated, and the rows represent different indicator value vectors. Each row should also contain an explanation/justification of the estimate (see Table V for an example).

It is often difficult to find statistics and historical data which can be used to estimate precisely how the indicator values will affect the likelihood parameters. We therefore propose the following heuristic: Identify the indicator value vector in which all indicators are *triggered* (meaning that they are assigned the value which would increase the likelihood value the most) and estimate the likelihood for this case. Then identify the indicator which will affect the estimate the most, and estimate the value for the case where this indicator is triggered, regardless of whether or not the other indicators are triggered. Then do the same for the indicator which affects the estimate second most, and proceed in this way until we have reached the case where no indicator is triggered. Table V gives an example of the definition of an indicator function[1]. Here T stands for true, F stands for false, and * stands for ignore, i.e. either true or false.

After all the likelihood functions have been defined and documented in tables, the **R** script encoding the BN structure (developed in Step 2) is updated with the new likelihood function definitions. This updated **R** script, constituting the cyber-risk assessment algorithm, is the output of Step 3.

VI. STEP 4: VALIDATION

The purpose of Step 4 is to validate the algorithm produced in Step 3. Initially, a validation team is assembled and an analysis leader is selected among them. The team should consist of domain experts, i.e. persons that have knowledge about the risks and threats of the risk model. The validation

[1]Due to space limitation we only show two rows of the table.

TABLE III
EXAMPLE OF FACTS

ID	Source	Basis	Fact
F1	UK Cyber Security Breaches Survey 2016, p.34 and 35 [16].	Statistics collected from 1008 companies/organizations	25% of UK businesses experienced one or more cyber security breaches within a 12 month period. Among companies/organizations who had any breach or attack (428 out of 1008) 13% of the attacks were related to "Access to computers, networks or services without permission and 8% of the attacks were relate to "Personal information stolen"
F2	OWASP Top 10, Table on p. 4 [17].	N/A	Broken Authentication and Session Management is number two on the OWASP list of top ten most critical web application risks.

TABLE IV
EXAMPLE OF BASELINE ESTIMATES

Name	Value	Description	Rationale
l_S1	Occurrences per year: [0, 25]	The likelihood that session fixation attack will be initiated	Due to F2, we know that Broken Authentication and Session Management is rated as the second most critical web-application risk by OWASP. Furthermore, there are automated tools for checking whether a web-application is vulnerable to this kind of attack. We therefore believe that the attack may be fairly common against web-pages that may seem vulnerable to the attack, but less common otherwise. However, we believe that the attack is not extremely common (i.e. not much more than twice each month), since it likely that the attack has to be tailored to the web-application (not completely automated).

TABLE V
EXAMPLE OF INDICATOR FUNCTION DEFINITION FOR PARAMETER L_S1 WHOSE LIKELIHOOD IS ESTIMATED BY OCCURRENCES PER YEAR

ID	IN-34	IN-41	IN-35	IN-51	IN-52	IN-42	l_S1	Rationale
I1	F	T	T	T	T	T	[15,30]	In this case, there is a very strong possibility that a successful attack has occurred or is occurring. The baseline estimate of l_S1 ([0,25]), particularly the lower bound, has therefore been increased.
I2	*	*	*	*	*	T	[12,25]	In this case, there is a strong possibility that a successful attack has occurred or is occurring, but probably not as high as in case I1. The estimate has therefore been slightly lowered.

team should also include people that were not involved in the definition of the algorithm to reduce social biases.

Step 4 has two sub-steps. In Step 4.1, the validation leader selects a set of scenarios for validation. In Step 4.2 each of these scenarios is validated by the validation team.

A. Step 4.1: Selecting validation scenarios

A scenario is a set of indicator value vectors. The purpose of Step 4.1 is to select a subset of all possible scenarios that satisfy a given coverage criterion. The scenarios are documented in the table containing the indicators and a description of the scenarios.

There are seven indicators in the risk model of our running example. This gives 128 possible indicator value vectors. Validating all these may be infeasible. We therefore chose the following two criteria: (1) cover the borderline scenarios (yielding the minimum and maximum risk values), and (2) cover each path in the risk model, meaning that for each path p (from the threat to the unwanted incident) in the risk model, there must be a scenario where one or more indicators along the path is triggered and the indicators for all other paths are not triggered unless these indicators also affect path p.

Table VI gives an example of five scenarios that together satisfy the coverage criteria. Note that the indicators IN-42 and IN-52 affect the overall risk value so strongly that they override the contribution of the other indicators. We have therefore chosen not to let these indicators be triggered in scenario C2 and C3 to better show the difference between these scenarios w.r.t the overall risk value.

TABLE VI
EXAMPLE OF SELECTED SCENARIOS

ID	IN-30	IN-34	IN-52	IN-35	IN-41	IN-42	IN-51	Description
C1	T	T	F	F	F	F	F	No indicators triggered.
C2	F	F	F	T	T	F	T	Top path indicators triggered except IN-42 and IN-52.
C3	F	T	F	F	T	F	F	Bottom path indicators triggered IN-42 and IN-52.
C4	T	T	T	F	F	T	F	Indicators IN-42 and IN-52 triggered.
C5	F	F	T	T	T	T	T	All indicators triggered.

B. Step 4.2: Running the validation scenarios

The purpose of this sub-step is to validate the scenarios selected in Step 4.2. That is, checking whether the output of the algorithm under each of these scenarios is reasonable. This sub-step is carried out in a meeting by the validation team. There are many ways in which the algorithm can be validated. In the following we describe how we did it during the development of 10 different algorithms.

Before each validation meeting, the validation leader executed the algorithm for each scenario to be validated to produce plots/charts showing the

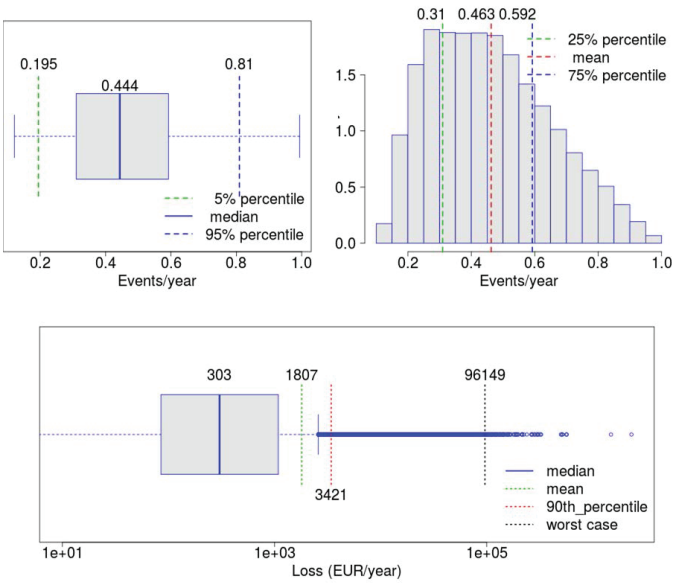

Fig. 5. Validation scenario 2

- median (typical) case and the 5% and 95% percentiles of the frequency of the risks in the risk model;
- shape of the frequency distribution for the risks;
- median (typical), mean, and 90% percentile of the risk values of the risks in the model.

During these meetings, the validation leader presented the risk model to the validation team, before showing the plots/charts to the validation team for each scenario. The validation team then checked whether the values were *roughly* correct, i.e. intuitive and/or reasonable.

As an example, the plots produced for scenario 2 are shown in Fig. 5. The top panel shows a boxplot and a histogram for the simulated distribution of the unwanted incident session hijacking (l_U1). The typical frequency (median) is 0.44 times per year (i.e. roughly once every second year) and there is a 95% probability that the frequency is less than 0.81 times per year. The histogram shows that the shape of the frequency distribution is slightly skewed to the left. The bottom panel shows a boxplot for the simulated risk value of the unwanted incident session hijacking (R1) is shown. Here we see that the typical (median) expected loss per year is 303 Euros per year, that the mean is 1807 Euros per year, and that there is 90% probability that the expected loss is below 3421 Euros per year.

VII. Related work

Unlike the method presented here, most risk assessment approaches do not support automated continuous assessment. Neither do they combine techniques from security risk analysis with actuarial techniques to provide quantitative estimates of cyber-risks. However, there are some approaches that share similarities with parts of our approach. In the following, we give an overview.

Refsdal et al. [18] present a model-based approach to make use of measurable indicators in order to obtain a risk picture that is continuously or periodically updated. However, they offer no support for exploiting empirical data sources, and little guidance on consequence assessment. The approach proposed by Ligaarden et al. [19] focuses on the security of dynamic services in the more complex setting of systems of systems. Krautsevich et al. [20] propose an approach to make use of run-time attribute monitoring to support risk-based enforcement of usage control (UCON) policies.

Saripalli et al. [21] propose a quantitative impact and risk assessment framework specialized for cloud security. Similar to our approach, they also make use of existing data sources, in particular data from the SANS institute, to support probability estimation of risks. However, they do not collect and organize existing data in a systematic approach as in our method (see Section V). Similar to our approach, Saripalli et al. [21] carry out a validation step to validate the estimated impact (consequence) and probability (likelihood) values. This is done by carrying out the Wide-band Delphi method [22], which is a forecasting technique used to collect expert opinion in an objective way, and arrive at consensus conclusion based on that. We believe a similar technique could be used during the validation step of our method, although this has not been done.

Poolsappasit et al. [23] propose an approach for dynamic security risk management using Bayesian Attack Graphs (BAGs). This approach is dynamic in the sense that it allows system administrators to tweak the probability of events captured by a BAG in order to see how this propagates in the complete risk picture. While their approach only facilitates manual update of probability of events, our method facilitates both manual as well as automatic update of the likelihood of events. The manual update in our method is based on input provided by representatives of the target under analysis, while the automatic update is based on input collected from tests, application-layer monitoring, and network-layer monitoring. Moreover, similar to our approach, Poolsappasit et al. [23] make use of estimates from statistical data sources (in particular data from the SANS institute).

As argued by Neil et al. [5] BNs provide a flexible and attractive solution to the problem of modeling (operational) risk. In particular, BNs enable an analyst to combine quantitative information (e.g. available historical data) with qualitative information (e.g. subjective judgments) regarding the loss-generating processes. In the context of cyber-risk, BNs have been used for a variety of purposes, such as to model attack graphs or loss event frequencies [23], [24].

The actuarial component of our framework is based on the Loss Distribution Approach (LDA), which is typically used to model operational risk and its insurability [9], provided that a sufficient amount of data is available. In the LDA, the temporal occurrence of the losses is frequently modeled by a Poisson process, while various families of distributions (Gamma, Generalized Pareto, Log-normal, etc.) might be used to model the severity of the losses.

Eling et al. [25] and Biener et al. [26] study whether

models which prove to be useful for operational risk can also be applied to an analysis of cyber-risk. They conclude that the LDA approach is suitable to model cyber-risk and that it provides useful insights regarding, e.g., the distinct characteristics of cyber-risk with respect to operational risk in general. Regarding the insurability of cyber-risk, they conclude that one of the main problems for pricing cyber-risk insurance is the scarcity of data, which induces high level of uncertainty regarding potential losses. In order to mitigate data scarcity, the model parameters in our approach are chosen based on a combination of available historical data and expert judgment, and hence provide a foundation for assessments which can be reassessed for specific firms or if new data becomes available.

VIII. Conclusion

The method we have presented was developed in the WISER project [3] and used to develop 10 algorithms, thereby demonstrating its feasibility. We estimate that developing the algorithm for the running example took roughly eight person days in all. The algorithms are being deployed for testing in three pilot organizations.

Guidelines for exploiting available empirical data sources are a part of the method. Following these, we were able to reach consensus in the validation team for the 10 algorithms, and to document the reasoning and empirical foundation in a structured way. Of course, we cannot claim that our work proves that the output of the algorithms correctly reflect reality.

For capturing risk models, we chose CORAS because it has been empirically shown that the language is intuitively simple for stakeholders [27]. Apart from the indicators, we employed the standard CORAS language [2]. Our experience confirmed that the involved participants had little problems understanding the models. Although we do not expect all stakeholders to understand the **R** script that constitutes an algorithm, an overall understanding can be ensured by presenting outputs from the script for selected scenarios (as in Fig. 5), as well as the corresponding CORAS model.

Cyber-risk assessment is still an immature field, often based on subjective estimates without proper documentation of the reasoning and factual foundation. We believe our work represents a step towards better cyber-risk assessment.

Acknowledgment

This work has been conducted as part of the WISER project (653321) funded by the European Commission within the Horizon 2020 research and innovation programme. The authors would like to thank the anonymous reviewers for their helpful feedback.

References

[1] *ISO 31000:2009(E), Risk management – Principles and guidelines*, International Organization for Standardization, 2009.

[2] M. S. Lund, B. Solhaug, and K. Stølen, *Model-Driven Risk Analysis: The CORAS Approach*. Springer, 2011.

[3] "Wide-Impact cyber SEcurity Risk framework (WISER)," http://www.cyberwiser.eu/, accessed March 1, 2017.

[4] A. Refsdal, G. Erdogan, G. Aprile, S. Poidomani, R. Colgiago, A. Alvarez, P. Lombardi, and R. Manella, "WISER deliverable D3.4: Cyber risk modelling language and guidelines, final version," http://www.cyberwiser.eu/, To appear.

[5] M. Neil, N. Fenton, and M. Tailor, "Using bayesian networks to model expected and unexpected operational losses," *Risk Analysis*, vol. 25, no. 4, pp. 963–972, 2005.

[6] "The R project for statistical computing," https://www.r-project.org, accessed April 9, 2017.

[7] "Hydenet: Hybrid bayesian networks using r and jags," https://cran.r-project.org/web/packages/HydeNet/index.html, accessed February 23, 2017.

[8] S. Mittnik and I. Starobinskaya, "Modeling dependencies in operational risk with hybrid bayesian networks," *Methodology and Computing in Applied Probability*, vol. 12, no. 3, pp. 379–390, 2010.

[9] A. J. McNeil, R. Frey, and P. Embrechts, *Quantitative Risk Management: Concepts, Techniques and Tools*. Princeton University Press, 2015.

[10] J. A. Rice, *Mathematical Statistics and Data Analysis*. Duxbury Press, 2007.

[11] S. A. Klugman, H. H. Panjer, and G. E. Willmot, *Loss Models: From Data to Decisions*. Wiley, 2012.

[12] J. C. P. Bus, "Convergence of newton-like methods for solving systems of nonlinear equations," *Numerische Mathematik*, vol. 27, no. 3, pp. 271–281, 1976.

[13] K. A. Atkinson, *An Introduction to Numerical Analysis*. Wiley, 1989.

[14] "Solving nonlinear systems of equation," https://cran.r-project.org/web/packages/nleqslv/nleqslv.pdf, accessed March 3rd, 2017.

[15] A. S. Chernobai, S. T. Rachev, and F. J. Fabozzi, *Operational Risk: A Guide to Basel II Capital Requirements, Models, and Analysis*. Wiley, 2007.

[16] R. Klahr, S. Amili, J. N. Shah, M. Button, and V. Wang, "Cyber security breaches survey 2016," 2016. [Online]. Available: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/521465/Cyber_Security_Breaches_Survey_2016_main_report_FINAL.pdf

[17] "Open web application security project (owasp)," https://www.owasp.org/, accessed February 8, 2017.

[18] A. Refsdal and K. Stølen, "Employing key indicators to provide a dynamic risk picture with a notion of confidence," in *Proc. 3rd IFIP International Conference on Trust Management (TM'09)*. Springer, 2009, pp. 215–233.

[19] O. S. Ligaarden, A. Refsdal, and K. Stølen, "Using indicators to monitor security risk in systems of systems: How to capture and measure the impact of service dependencies on the security of provided services," in *Transportation Systems and Engineering: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2015, pp. 1342–1377.

[20] L. Krautsevich, A. Lazouski, F. Martinelli, and A. Yautsiukhin, "Risk-aware usage decision making in highly dynamic systems," in *Proc. 5th International Conference on Internet Monitoring and Protection (ICIMP'10)*. IEEE, 2010, pp. 29–34.

[21] P. Saripalli and B. Walters, "Quirc: A quantitative impact and risk assessment framework for cloud security," in *Proc. 3rd IEEE International Conference on Cloud Computing (CLOUD'10)*. IEEE, 2010, pp. 280–288.

[22] B. W. Boehm, *Software Engineering Economics*. Prentice-hall, 1981.

[23] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using bayesian attack graphs," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 61–74, 2012.

[24] A. Le, Y. Chen, K. K. Chai, A. Vasenev, and L. Montoya, "Assessing loss event frequencies of smart grid cyber threats: Encoding flexibility into fair using bayesian network approach," in *Proc. 1st EAI International Conference on Smart Grid Inspired Future Technologies (SmartGIFT'16)*. Springer, 2017, pp. 43–51.

[25] M. Eling and J. H. Wirfs, "Modelling and management of cyber risk," k.A., Working Paper, 2015.

[26] C. Biener, M. Eling, and J. H. Wirfs, "Insurability of cyber risk an empirical analysis," *The Geneva Papers on Risk and Insurance Issues and Practice*, vol. 40, no. 1, pp. 131–158, 2015.

[27] B. Solhaug and K. Stølen, "The coras language - why it is designed the way it is," in *Proc. 11th International Conference on Structural Safety & Reliability (ICOSSAR'13)*. Taylor and Francis, 2013, pp. 3155–3162.