CRAC: Confidentiality Risk Assessment and IT-Infrastructure Comparison*

Ayşe Morali¹, Emmanuele Zambon¹, Sandro Etalle^{1,2}, and Roel Wieringa¹

¹ University of Twente, Enschede, The Netherlands
`firstname.lastname (at) utwente.nl`

² Eindhoven Technical University, Eindhoven, The Netherlands

Abstract. In this paper we present CRAC, an IT infrastructure-based method for assessing and comparing confidentiality risks of IT based collaborations. The method determines confidentiality risks by taking into account the effects of the leakage of confidential information (e.g. industrial secrets and user credentials), and the paths that may be followed by different attackers (e.g. insider, outsider and outsourcer). We also show how the CRAC-method can be applied in practice and we evaluate its effectiveness by applying it to a real-world outsourcing case.

1 Introduction

Nowadays, most data exchange within and across organizations boundaries takes place electronically. Exchanged data often contains confidential information, the loss of which would result in economical damage. Good security (on the other hand) is also costly and even the best and most expensive countermeasures cannot mitigate all possible confidentiality incidents. This is mainly because of the impossibility of monitoring *all* confidentiality breaches. Therefore, the goal of security officers is to strike the right balance among security, budget, and system usability. To achieve this, they typically refer to well-established standards and best practices such as ISO 27002 [10] and NIST 800-30 [2]. Assessing IT (confidentiality) risks becomes particularly challenging in the presence of cross-organizational cooperations, e.g. IT outsourcing. As part of the cooperation, organizations typically connect together their IT-infrastructure and they grant access rights to each other's (confidential) information. This process establishes a so-called *IT-enabled network of organizations* which increase the complexity of an IT-confidentiality risk assessment because one has to deal with a more complex IT-infrastructure and with an extended set of potential threats.

To make informed decisions on the (security) design of its IT-infrastructure, an organization needs to fully understand the confidentiality risks that arise when there is a collaboration with other organizations. How and where confidential

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

data is stored has a big impact on the security of the IT-infrastructure. Decision makers need to be able to assess and compare different solutions about the design of IT-infrastructures based on the confidentiality risks. This can only be achieved if risks are consistently assessed for each considered solution. However, the result of typical risk analysis methods cannot be consistently compared with each other if they were carried out by different people. This happens because they are mostly based either on the subjective opinion of the different risk assessor(s) or on event histories. Confidentiality risks are even harder to assess in an inter-subjective (independent of personal judgement) way, because of their non-functional nature and the lack of logs about past incidents.

To solve these problems, in this paper we present the Confidentiality Risk Assessment and Comparison (CRAC) method. With the CRAC-method one can assess confidentiality risks by taking into account both how information assets flow in the underlying IT architecture and the different paths attackers can use to find their way in the architecture. We use *information flow* [17] to analyze where and “how much” critical information is located in the system, and a customized version of *attack paths* [18] to model how attackers with different profiles may be able to reach this information.

The main contribution of CRAC to confidentiality risk assessment is that it support decision providers by allowing them to *compare* the confidentiality risks of alternative IT-infrastructure design solutions. CRAC is meant to be employed for infrastructures used by IT-enabled network of organizations, in such a way that it reduces the subjectivity of the risk assessment results. We show the feasibility of the CRAC method by applying it to a real-world case and by evaluating its subjectivity, practicality and precision based on the success criteria that we derive from the case-study stakeholders. CRAC improves and extends our earlier work, DCRA [15], for confidentiality risk assessments. Please refer to the related works for a description of the extensions and improvements over DCRA.

2 The Industrial Case

A large multinational electronics manufacturing company (from now on: the *Company*) is outsourcing the management of its authentication and authorization system (*the System*) to a multinational IT service provider (the *Outsourcer*). The System is used by the Company’s employees to access the Company’s data and services, and by the employees of the Outsourcer for configuring, monitoring and maintaining the system (the *Managed Services*). The Outsourcer proposed to replace the IT-infrastructure that the System is currently built on with a different one. The company needs to know if the new IT-infrastructure is at least as secure as the one that is currently in use. Fig. 1 reports alternative access paths to Managed Services using two alternative IT-infrastructures that are under investigation. The first infrastructure (Alternative 1 in Fig. 1) is the one currently in use. All access attempts from the Outsourcer’s network to the Managed Services are monitored by the *session directory services*. The *terminal*

Fig. 1. Simplified version of the two architectures and access paths.

server makes applications available to the Outsourcer's employees in a terminal session. The *secure gateway (1)*, which is installed on the third party gateway (TPG) of the Company, is responsible of authenticating the Outsourcer employees to the Managed Services. The *presentation server* is used by the Outsourcer employees as an interface to manage applications. The second infrastructure (Alternative 2 in Fig. 1) contains a second access path with, which allow a special group of the Outsourcer employees to access the Managed Services in emergencies. Unlike the first path, which uses two-factor authentication (i.e., a secure ID plus a password), the second path uses IP based authentication.

The Company classifies information in three confidentiality levels: private, highly confidential and company confidential. For instance, user credentials (containing e.g. social security numbers) are *private*, business information and access control lists is considered critical for the business is *highly confidential*, IT-architectural documents on the other hand may cause a loss only if 3rd parties access them and are *company confidential*.

The stakeholders involved are two *independent* business units of the company: the *Global Infrastructure Board* (GIB) and the *Risk, Performance & Compliance Unit* (RMC). GIB owns System and wants to know which of the two IT-infrastructures is more robust to confidentiality breaches. GIB also determines the business impact of confidentiality breaches, which in this case depends on (a) the criticality of information asset, (b) the volume of information that get disclosed, and (c) to whom they get disclosed to. RMC uses a check-lists based risk assessment method to assess the risks and compliance requirements of the Companies IT systems. From here on, we will call this method the RA method. The RA method is customized and developed based on ISO 27001 [9] and NIST 800-30 [2]. It consists of two main parts: (1) Business Impact Analysis; and (2) Threat and Vulnerability Analysis. According to RMC, the RA method does not allow linking threats to the component of the IT-infrastructure under assessment. For these reasons, it's results cannot be used for comparing alternative IT-infrastructures.

3 The CRAC-method

The CRAC-method adopts and extends the IT security related aspects of ISO/IEC 13335 [11]. It tries to assess how *easy* it is to access critical information. To this end, it analyzes the components that form the IT-infrastructure. The CRAC-method is based on two ideas. (1) Information is a logical asset, so it can flow from one component to the other one. E.g., it could flow to an component because a user copies it there. (2) An attacker may penetrate into a system through different components and follow different attack paths. CRAC analysis consists of four steps, which we now illustrate (see [16] for a more detailed description of each step and related examples).

Step 0: Collecting Basic Information In this step we collect: (1) the list of information assets present on the system, their confidentiality level and homogeneity property; (2) the list of components that form the IT-architecture of the system and how components are related to each other; (3) the list of vulnerabilities; and (4) the list of possible threat agents. We adopt the following notation: L is the set of confidentiality levels; N is the set of information asset quantity classes; I is the set of single impact values; TI is the set of total impact values; P is the set of qualitative likelihood values. We call *information assets* the “semantic components of an information system that are required for an organization to conduct its mission or business” [12]. A is the set of information assets we consider. To each information asset $a \in A$ we associate a confidentiality level $l(a)$. C is the set of components (i.e. hardware, software or network segment) which may contain one or more instances of a given information asset a . The mapping $n : A \times C \rightarrow N$ is a qualitative estimate of the number of instances of a that can be retrieved from component c at once. An information asset a is *homogeneous* if the damage due to its disclosure can be considered proportional to the number of its instances that get disclosed. For example “social security numbers” are homogeneous, since the damage due to the loss of one hundred social security numbers is larger than the damage due to the loss of a single social security number. Conversely, an information asset is *non-homogeneous* if the damage due to the disclosure of one instance is as big as the damage of the disclosure of all instances. For example, if the credentials of one user get disclosed, the damage to the company is the basically same as if credentials of 100 users with equal access rights would be disclosed. We model this with the mapping $h : A \rightarrow \{\text{homogeneous}, \text{non-homogeneous}\}$.

Definition 1. (architecture graph) An architecture graph $\text{arch} = \langle C, E \rangle$ is a directed graph in which C is the set of vertices representing components and E is the set of edges $E \subseteq C \times C$. $(c_1, c_2) \in E$ iff there exists a direct connection between c_1 and c_2 such that data can flow from c_1 to c_2 .

Step 1: Analyzing information flow In this step we first analyze the logical and physical connections among components based on an infrastructure. Then, we determine the impact of each component by considering the information

assets that may flow to them. If there is a possibility for an information asset to flow to a component then we assume that that information is present on that component. We furthermore assume that information flows in a predictable way. Thus the information flow analysis can recognize all possible paths according to policies and documented properties of the components. In more detail, to model information flow we build for each information asset a a set of *flow paths*. A flow path is a path in the architecture graph which starts with a component where a is stored.

The nodes of a flow path represent the components. We represent a flow path by an ordered list $fp = [c_1, \dots, c_n]$ where $c_1 \dots c_n \in C$ with no repeated occurrences of c_i . We call $FP_a = \{fp_1, \dots, fp_m\}$ the set of flow paths of a . We use the maximum number of instances that may flow to a component c from its connected components to determine the number of instances an attacker can disclose by gaining access to c . After constructing the flow paths we determine for each information asset a , for each component c and for each flow path $fp \in FP_a$, the number of instances of a that are present in c according to fp using the function $(n : A \times C \times FP_a \rightarrow N)$. If we call $index(c, fp)$ the index of c inside fp , then $n(a, c, fp) = \min_{i \leq index(c, fp)} n(a, c_i)$, where $c_i \in fp$.

Finally, we determine for each a , c and fp , the impact of the disclosure of the instances of a which are present in c according to fp using the function $(fp\text{-}imp : A \times C \times FP_a \rightarrow I)$, which considers the number of instances of an information asset which can be extracted to a component at once, its confidentiality level and homogeneity.

$$fp\text{-}imp(a, c, fp) = \begin{cases} l(a) \odot n(a, c, fp) & , \text{ if } h(a) = \text{homogeneous}; \\ l(a) \odot all & , \text{ if } n(a, c, fp) \neq none; \\ null & , \text{ else.} \end{cases}$$

Where $\odot : L \times N \rightarrow I$ is a monotone composition operator for ordinal scale qualitative values in L and N . $\odot$ should be agreed on with the risk assessment stakeholders to guarantee that everybody understands how values are composed. We are able to compute the *impact* of the disclosure of each information asset a on each component c , $imp : A \times C \rightarrow I$, as the maximum impact with respect to all the possible flow paths in FP_a .

$$imp(c, a) = \max_{fp \in FP_a} fp\text{-}imp(a, c, fp)$$

In practice, quantitative values are difficult to obtain. Consequently, the CRAC method determines the impact with partially ordered qualitative values, as it is commonly done in many risk assessment methods. However, if quantitative values are available then the $\odot$ operator behaves as a multiplication. We can now compute the *total impact* of each component. If c contains only one information asset a , then $imp(c) = imp(c, a)$. On the other hand, if c contains two or more assets (say a_1 and a_2) then we “add” $imp(c, a_1)$ and $imp(c, a_2)$. To this end we use the monotone operator $\oplus : TI \times I \rightarrow TI$. As for $\odot$, $\oplus$ shall be agreed on with the stakeholders. More formally, the total impact of c is:

$$imp(c) = \oplus_{a \in A} imp(c, a)$$

Fig. 2. (a) Flow paths for User Credentials. (b) APPs followed by the Outsourcer.

Running example - Part 1 *Following the information classification scheme that is adopted by the Company, we use the following sets: $L = \{\text{high, medium, low}\}$ and $N = \{\text{none, single, all}\}$. We furthermore limit our assessment to the information assets User Credentials and Business Information.*

Fig. 2 (a) illustrates two information flow paths of $FP_{\text{UserCredentials}}$ in Infrastructure 2. For the sake of presentation we include in only those components that are also illustrated in Fig. 1. The remaining ones are represented with dots. User Credential Directory is the component where User Credentials reside. The left path illustrates the credentials flowing towards the Terminal Of The Outsourcer, whereas the right path illustrates synchronization of credentials between the User Credential Directory and the Identity Store. $l(\text{UserCredentials}) = \text{high}$, $h(\text{User Credentials}) = \text{non-homogeneous}$, $l(\text{BusinessInformation}) = \text{medium}$ and $h(\text{BusinessInformation}) = \text{homogeneous}$. Accordingly, we determine that the impact on all components in the architecture on which User Credentials flow is high. Furthermore, assuming that on the component Terminal Of The Outsourcer only the information assets User Credentials and Business Information are available, by applying the $\oplus$ operator we obtain it's total impact as very-high.

Step 2: Constructing APGs In the second step of the CRAC-method we build the Attack Propagation Paths (APPs) which describe how different threat agents might penetrate into the IT infrastructure. Then, we determine the likelihood of a threat agent to access the information available on each component. We call T the set of all threat agents in the System. Vulnerabilities are weaknesses of the components which allow attacks propagation. We call V the set of all the vulnerabilities in the System. We represent the fact that v is a weakness of c with a mapping $w : V \times C \rightarrow \{\text{true, false}\}$, and the likelihood that a threat agent t exploits a vulnerability v to compromise an component c with the mapping $p : T \times V \times C \rightarrow P$.

To model confidentiality breaches we build for each threat agent t a set of APPs. The nodes of an APP represent the components that an attacker compromises during an attack. We build each APP in two steps. We first add a node (c_1) to the APG for each component that can be directly reached by a threat

agent (for external threat agents we can add a special fictitious component “the internet”). Second, we iteratively add new nodes and edges as follows: if node c_1 is in the APP and c_1 is connected to the component c_2 in the architecture graph, then we add c_2 to the APP. Similarly to information flow paths, we represent an APP by an ordered list $app = [c_1, \dots, c_n]$ where $c_1 \dots c_n \in C$ with no repeated occurrences of c_i . We call $APP_t = \{app_1, \dots, app_n\}$ the set of APPs a threat agent t can follow.

After constructing APPs we determine the likelihood of each threat agent t compromising each component c by following each attack propagation path in APP_t . In doing so we need to take into account two properties. (1) Each component may have more than one vulnerability that t can exploit, in this case we assume the threat agent will exploit the vulnerability with the highest associated likelihood. (2) The threat agent needs to compromise other components in order to compromise c , in this case we assume the likelihood of compromising c is the lowest likelihood of the list (i.e. the hardest step).

Definition 2. (Attack Propagation Likelihood) Given a component c , a threat agent t , a set of vulnerabilities V , an attack path $app \in APG_t$, and $\text{index}(c, app)$ the index of c in the ordered list app , we call $p : T \times C \times APP_t \rightarrow P$ the likelihood of t compromising c by following app where

$$p(t, c, app) = \min_{i < \text{index}(c, app)} \max_{v \in \{v | v \in V, w(v, c) = \text{true}\}} p(t, v, c_i) \quad (1)$$

Finally, by merging the attack propagation likelihoods for a component with respect to the different APPs, we determine its *reachability level*.

Definition 3. (Reachability Level) Given a component c , the reachability level of c $\text{reach} : C \rightarrow P$ is given by the highest attack propagation likelihood with respect to all the possible attack propagation paths, as follows:

$$\text{reach}(c) = \max_{t \in T} (\max_{app \in APP_t} (p(t, c, app))) \quad (2)$$

Running example - Part 2 Fig. 2 (b) illustrates two APPs that the threat agent employee of the Outsourcer may follow to access User Credentials on the User Credential Directory, in the scenario of IT-infrastructure 2. The Outsourcer employee may access User Credentials Directory via the Terminal Of The Special Outsourcer or via the Terminal Of The Outsourcer.

To enumerate vulnerabilities we refer to the threat and vulnerability list the Company uses in their RA-method. Furthermore, to determine the attack propagation likelihood for each component we cross-check the attacker profiles [8, 14, 7], e.g. outsourcer has system knowledge, with necessary conditions to exploit vulnerabilities, e.g. physical access is necessary for stealing a hard drive.

. Furthermore, the attack propagation likelihood of User Credentials Directory for the Outsourcer employee is likely. That is because both attack path that leads to User Credentials Directory contains more than 1 attack propagation and the lowest likelihood in each attack paths is likely. Finally, assuming that Terminal Of The Special Outsourcer is only on the APG of the Outsourcer and a further

threat agent, Outsider, and its attack propagation likelihood, according to the APG of the Outsider is likely. Thus, we say $\text{reach}(\text{TerminalOfTheSpecialOutsourcer})$ is very-likely.

Step 3: Risk Calculation and Comparison In this step we combine the output of steps 1 and 2 to identify the weak spots in the system and compare the security of alternative IT-infrastructures. We identify the weak spots based on their confidentiality risk.

Definition 4. (Risk) *Given a component c with total impact $\text{imp}(c)$ and reachability level $\text{reach}(c)$, the risk of c is the pair $\text{risk}(c) = \langle \text{imp}(c), \text{reach}(c) \rangle$.*

After determining the risk of all components for alternative IT-infrastructures we sort them. We identify the most critical components as those components with the highest total impact and reachability level. Then we determine which infrastructure is more robust w.r.t. confidentiality risks by comparing the percentage of assets on the different architectures with the same risk level.

Running example - Part 3 *In step 1 we computed the total impact value of Terminal Of The Special Outsourcer (high), in step 2 we computed its reachability level (very-likely). Accordingly, the risk of Terminal Of The Special Outsourcer for IT-infrastructure 2 is $\langle \text{high}, \text{very} - \text{likely} \rangle$. Comparing the risk of the components of IT-infrastructure 1 and 2 we see that 1 is more robust than 2. In particular, IT-infrastructure 2 contains 2 additional components with risks $\langle \text{very} - \text{high}, \text{very} - \text{likely} \rangle$, and 3 components common with IT-infrastructure 1 that have higher reachability levels.*

For more complex systems presenting risk in a table may be unsuitable. For that cases one can calculate the percentage of components with the same total impact and reachability level, and present the results in a (smaller) matrix.

4 Evaluation

In this section we discuss how effective the CRAC method has been in our case study on bringing the stakeholders closer to their goals.

Solution Criteria According to the stakeholders a successful confidentiality risk method should satisfy the following criteria: (C1) The method should allow a detailed representation of risk; (C2) The method should be practical to implement; and (C3) The method should deliver less subjective results than the RA method. We measure how well our solution scores w.r.t. these criteria based on the following measures: (M1) The number of risk-related aspects the method is able to represent; (M2) the percentage of optional risk-related aspects; (M3) the percentage of adjustable risk-related aspects; and (M4) the percentage of inter-subjective aspects.

(C1) indicates that a good risk assessment method should allow the risk assessor to represent the complexity of the system to be risk assessed in a detailed

Table 1. Comparison of risk assessment methods.

Measures	CRAC	Check-list	CRAMM
M1: number of aspects	18	16	25
M2: percentage of optional aspects	16%	44%	4%
M3: percentage of adjustable aspects	44%	25%	36%
M4: percentage of non-subjective aspects	78%	56%	72%

manner and is justified by the goal of RMC. We measure (C1) with (M1) and (M3). (M1) expresses the number of confidentiality-related aspects a method is able to model (e.g. attacker profiles, attack propagation and the amount of instances that may get disclosed). From here on we call them *aspects*. For this comparison we assume that all the aspects are equal weighted. We implicitly assume that the more aspects the method considers the more precisely it can assess risks. (M3) indicates the possibility of using the method with risk-related aspects at different detail levels. For instance, when considering threat agents, if there is only one type of threat agent (e.g. attacker) or many types of threat agents (e.g. insider, outsider and outsourcer). The accuracy of a risk assessment method (C1) often has a negative impact on the ease of its implementation (C2). The implementation effort of a method should ideally be adjustable to the criticality of the system to be assessed (RMC needs to assess the risks of low critical systems with lower effort than for high critical systems). We measure (C2) with (M2) and (M3), which express the flexibility of the method. This is a desirable feature in the case in which acquiring complete and detailed information is not possible because of limited resources. Here, flexibility can be described as (1) how well a risk assessment method can be adjusted to work at different detail levels without compromising accuracy and (2) how easy it is to refine or abstract the method in technical level. The goal of the Company is to compare the confidentiality risks of two alternative IT-infrastructure. This requires assessing the risks of these two IT-infrastructure separately and then comparing the assessment results. Different risk assessors must be able to work on the two assessments. Therefore, the method they use must be inter-subjective (C3). Since the subjectivity of assessment depends on the subjectivity of the aspects that are used for determining the incident likelihood and impact, we measure it by the percentage of non-subjective aspects (M4). The aspects that we consider as inter-subjective are: (1) documented facts (e.g. the components determined based on IT-architectural drawings), (2) the knowledge shared among all the stakeholders (e.g. the list of vulnerabilities determined based on a publicly available vulnerability data base) and (3) any combination of the first two.

Comparison We now compare three risk assessment methods with respect to the success criteria we presented. The methods we consider are: (1) the CRAC method; (2) the RA method and (3) the well known CRAMM method [5]. For this comparison we disregard the governance related concepts of CRAMM and check-list based approach, which are outside the scope of this paper. Tab. 1 reports a summary of this comparison.

Regarding M1, using the CRAMM method one is able to take into account almost twice as many aspects than with the check-list based approach and our CRAC-method. Some of the aspects that the CRAMM method takes into account (and CRAC does not) are the number of persons using the assets, threat level and potential impact scenarios. However, there are also aspects that CRAC considers and CRAMM doesn't. They are information homogeneity and volume. We conclude that, although the CRAMM method represents in total a higher number of aspects than the other two, it does not cover all the relevant ones.

Regarding M2, the check-list method allows one to ignore 30% aspects than the CRAC-method and 40% more aspects than the CRAMM-method. However, the optional aspects in CRAC are more evenly distributed between impact and likelihood than in the check-list method and the CRAMM-method.

Regarding M3, the CRAC-method considers 19% more aspects than the check-list based approach and 8% more aspects than the CRAMM-method with adjustable granularity. For instance, if the system to be risk assessed is critical then the CRAC-method enables the assessor to determine the impact by differentiating among different volumes of instances flowing from one component to another. Consequently, with the CRAC-method the risk assessor can adjust the granularity of the impact determination depending on the criticality of the system to be risk assessed. Note that the criticality of a system depends on the confidentiality levels of the information assets that the system contains.

When we consider M4, among the three methods, CRAC uses the highest percentage of non-subjective aspects. It is followed by the CRAMM-method, which uses only 6% less non-subjective aspects than CRAC. This happens because most of the information it uses is either generally well-documented or it must previously be agreed on by all stakeholders. Although, the CRAC-method considers almost the same number of aspects as the check-list method, there are almost twice as many aspects that can be adjusted to the desired granularity at which to carry out the risk assessment. Accordingly both the CRAC-method and the CRAMM-method represent confidentiality risks better than the companies check-list method.

Repeatability of CRAC depends on satisfying two assumptions: (A1) IT-architectural drawings on the system to be risk assessed is available; and (A2) for risk assessment purposes staff with good security understanding can be interviewed. Large outsourcing providers are subject to deliver a high level IT-architectural document describing the system to be outsourced. Furthermore, large outsourcing clients employ security staff and chief security officers. Therefore, if applied to a case where the outsourcing provider and client are big organization then both assumptions are satisfying.

We believe that one of the reasons why we achieved such good results is that the CRAC method is specifically designed for assessing "confidentiality" risks, whereas the other methods aim to assess confidentiality, integrity and availability risk as a whole. Furthermore, we developed the CRAC-method with the success criteria defined by the stakeholders in our minds, whereas the CRAMM-method is not developed to serve the goals of the stakeholders in this case.

5 Related Work

Some well known risk assessment methodologies, such as CORAS [3], CRAMM [5] and OCTAVE [1], give detailed recommendations about which modeling techniques are more suitable for which step of a risk assessment. The CRAC-method extends these risk assessment methodologies by modeling confidentiality risk at IT-infrastructure level: it links vulnerabilities to components and determines reachability of these components according to the profile of a threat agent. Differentiating between threat agents and considering the effects of IT-infrastructure on the risk is essential for assessing IT risk.

In [15] we introduced the DCRA model. CRAC improves and extends DCRA for outsourced IT-systems. In such scenarios detailed information on confidentiality aspects, e.g. volume of information stored on each component, is not explicitly available. Therefore, the CRAC-method presents a more practical approach that systematically elicits information on confidentiality aspects

For confidentiality it is essential to model how information flows. In the literature we find a number of approaches for modeling security with information flow graphs, e.g. [6, 17, 13]. These approaches are distinguished according to what the nodes of the graph model and to the information flow criteria. Only Chivers [6] uses information flow trees and form attack attack paths for analyzing risk. However, the diagrams they propose can neither be used for comparing risks of two IT infrastructures.

Attack paths and attack trees are introduced by Schneier [18] and are widely used in the security literature (e.g. [4, 8]) to model different ways of compromising a system. In most of the cases the nodes of an attack graph represent threats or vulnerabilities, as threat trees do. Our approach resembles attack trees because we model how an attack propagates. However we carry out the propagation analysis at the IT-infrastructure level.

6 Conclusion and Future Work

In this paper we presented the CRAC-method and how it can be used (1) as supplement to the existing risk management approaches for practically assessing confidentiality risks of a IT-system that an industrial organization outsources to an organization that is expert in IT-systems, and (2) as a stand alone tool for comparing the security of IT-infrastructures w.r.t. confidentiality. The CRAC-method extends the concept of architecture-based confidentiality risk assessment in the absence of explicit information on confidentiality aspects by (a) eliciting impact related information by modeling the information flow and (b) eliciting the reachability information on critical information assets by modeling attack paths.

We validate CRAC by applying it to a real-world case, and evaluated it based on the success criteria defined by the stakeholders of the Company. Accordingly, the CRAC-Method represents the confidentiality risks in a more detailed manner than the currently employed check-list method. However, it is less detailed than the CRAMM-method. We also show that CRAC can be gracefully adjusted to

work at different detail levels, according to the criticality of the system to be risk assessed. Finally, we argue that our approach is a significant step towards less subjective risk assessment. In the future we intend to extend CRAC with tool support.

References

1. Alberts, C., Durofee, A.: An Introduction to the OCTAVE Method, <http://www.cert.org/octave/methodintro.html>
2. Bowen, P., Hash, J., Wilson, M.: Information Security Handbook: A Guide for Managers. NIST Special Publication 800-100 (2006)
3. den Braber, F., Dimitrakos, T., Gran, B.A., Lund, M.S., Stolen, K., Agedal, J.O.: The CORAS methodology: model-based risk assessment using UML and UP pp. 332–357 (2003)
4. Breu, R., Innerhofer-Oberperfler, F., Yautsiukhin, A.: Quantitative assessment of enterprise security system. In: Int. Workshop on Privacy and Assurance. IEEE Computer Society (2008)
5. British Government's Central Computer and Telecommunications Agency: CRAMM: Risk Analysis and Management methodology (2008), <http://www.cramm.com/>
6. Chivers, H., Clark, J., Cheng, P.C.: Risk profiles and distributed risk assessment. *Computers & Security* 28(7), 521 – 535 (2009)
7. Grunske, L., Joyce, D.: Quantitative risk-based security prediction for component-based systems with explicitly modeled attack profiles. *J. Syst. Softw.* 81(8), 1327–1345 (2008)
8. Ingoldsby, T.: Understanding risk through attack tree analysis. *Computer Security Journal* 20(2), 33–59 (2004)
9. ISO/IEC 27001 Information Security - Specifications (2006), <http://www.iso.org>
10. ISO/IEC 27002 Information technology - Security techniques - Code of practice for information security management (2005), <http://www.iso.org>
11. ISO/IEC 13335: Management of Information and Communication Technology Security - Part 1: Concepts and Models for Information and Communication Technology Security Management. (2004)
12. Krause, M., Tipton, H.: Handbook of Information Security Management. CRC Press LLC, Auerbach Publishers Inc. (1998)
13. Majorczyk, F., Totel, E., Mé, L., Saïdane, A.: Anomaly Detection with Diagnosis in Diversified Systems using Information Flow Graphs. In: SEC. pp. 301–315 (2008)
14. Moore, A., Ellison, R., Linger, R.C.: Attack Modeling for Information Security and Survivability. Tech. rep., Carnegie Mellon University (2001)
15. Morali, A., Zambon, E., Etalle, S., Overbeek, P.: IT Confidentiality Risk Assessment for an Architecture-Based Approach. In: Proc. of the 3rd IEEE Int. Workshop on Business-Driven IT Management. IEEE Computer Society Press (2008)
16. Morali, A., Zambon, E., Etalle, S., Wieringa, R.J.: CRAC: Confidentiality Risk Analysis and IT-Architecture Comparison of Business Networks. Tech. rep. TR-CTIT-09-30 (2009)
17. Osborn, S.: Information flow analysis of an rbac system. In: Proc. of the seventh ACM symposium on Access control models and technologies. pp. 163–168 (2002)
18. Schneier, B.: Attack Trees. *Dr. Dobbs's Journal* 12(24), 21–29 (1999), <http://www.schneier.com/paper-attacktrees-ddj-ft.html>