Technical Report

Nr. TUD-CS-2014-0809
May 9th, 2014

CASED

Expert Knowledge for Contextualized Warnings

Authors
Steffen Bartsch and Melanie Volkamer

Expert Knowledge for Contextualized Warnings

Steffen Bartsch[1] and Melanie Volkamer[1]

Technische Universität Darmstadt
Hochschulstraße 10, 64289 Darmstadt, Germany
{steffen.bartsch,melanie.volkamer}@cased.de

Abstract. Users are bothered by too many security warnings in a variety of applications. To reduce the number of unnecessary warnings, developers cannot continue to report technical security problems. Instead, they need to consider the actual risks of the context for the decision of whether and how to warn – contextualized warnings. For this risk assessment, developers need to encode expert knowledge. Given the number and complexity of the risks – for example, in Web browsing –, eliciting and encoding the expert knowledge is challenging. In this paper, we propose a holistic methodology for an abstract risk assessment that builds upon prior concepts from risk management, such as decision trees. The result of the methodology is an abstract risk model – a model to assess the risk for the concrete context. In a case study, we show how this methodology can be applied to warnings in Web browsers.

Keywords: Risk assessment, Security interventions, Web-browser warnings

1 Introduction

When implementing security warnings, software developers often conveniently off-load the risk assessment of the situation to the user. They just report every technical security problem. However, the resulting prevalence of warnings leads to habituation, and this, in turn, to users ignoring them. This is the case for a variety of applications. One example is that of mobile applications, where technical lists of required permissions are ignored [4, 7]. Even more often, it has been observed how Web-browser warnings are ignored [13, 10]. To solve this problem, the number of false positives need to be reduced: those warnings that *are* shown need to count.

The required proper risk assessment prior to showing a warning comes with a high cost: A significantly more complex warning logic than the current hard-wired warnings that are directly tied to technical indicators. One such example for Web browsing is to show a certificate warning in case the trust chain of the TLS server certificate cannot be established. In previous work [1], we have proposed to support the developers by lowering the threshold for risk-based decisions of whether and how to intervene. These decisions are then "contextualized" since they take more context into account. In particular, the risk assessment needs to

be based on more complex risk models than "technical problem → warning." This risk model needs to encode whether and how experts estimate the risk level based on the context – encoding expert knowledge.

The expert knowledge links the description of a situation with the decision of whether and how to intervene. To give an example from Web browsing, typical indicators describing the situation include the security of the current connection, the type of the website, and trustworthiness ratings of the website. Experts might link missing connection security to the risk of eavesdropping of credentials, with the severity depending on the website type. They might also employ the website trustworthiness rating and website type to deduce risks of forged products or privacy issues. Either case may require an active intervention that blocks the user without interacting with the warning or a passive intervention, only highlighting a potential problem – for example, through a symbol in the browser chrome.

A major part of the additional effort for software developers from contextualized interventions lies in the formalization of the implicit knowledge of experts. The goal of this work thus is to guide this process with a holistic methodology. Implicit and unstructured expert knowledge is formalized to an abstract risk model – a model that allows to derive risk levels for concrete contexts. The abstract risk model thus enables developers to implement contextualized decisions of whether to intervene. More specifically, the knowledge is transformed so that it allows the assessment of risk for concrete situations based on indicators, such as the connection security or the type of the website. The main contribution is the methodology for this process. Its viability is shown in a case study on Web-browser warnings.

2 Background on risk assessment

There is a wealth of publications available on risk management [2, 3]. First, there are national and international standards on risk management. ISO 27005 [6] provides a general framework for risk management in IT systems. The U.S. FIPS-65 standard, withdrawn in 1995, applied the well-known quantitative approach "Annualized Loss Expectancy" (ALE, [8]). The most widely-used management approaches are of qualitative nature, though. Among other problems, quantitative estimations suggest a precision that is not realistic with the available input [12, 9]. Many aspects, such as attacker motivation in case of insider threats, are difficult to quantify. A well-known qualitative approach is the NIST Special Publication 800-30 [12] that supersedes the quantitative FIPS-65 standard. Similar to other standard methods, NIST 800-30 starts with a qualitative valuation of assets and impacts of incidents. Then, threats and vulnerabilities are identified and categorized with qualitative likelihood estimations. In a risk assessment step, the inputs are combined into risk estimations and, in the risk management part, mitigations are chosen.

These risk management processes are powerful methods for the assessment of risks. However, these are tailored for risk assessments of a concrete situation – for example, how likely is an attack on the company network. Conversely,

our method aims to create abstract risk models that describe risks for changing situations since we need to assess for each situation whether an intervention is appropriate.

More focused on the application in warnings and interventions are approaches that model the expert knowledge [5]. These emphasize how the risk needs to be assessed and employ decision trees. We build upon these approaches in the method laid out below.

3 Methodology to construct an abstract risk model

The goal in this paper is to propose a method to create an abstract risk model. The model is abstract in the sense that it can be applied for the risk assessment of concrete situations. A concrete situation is described through indicators, such as whether a connection is secured. The main challenges are the collection and formalization of expert knowledge. For these tasks, security practitioners and domain experts gather as the risk assessment team and provide their risk estimations. Specifically, the team meets for workshops to construct the abstract risk model, or to adapt an existing one to the changing threat landscape. The abstract risk assessment is based upon the concepts of qualitative risk analysis as laid out in ISO 27005 [6] and NIST 800-30 [12].

As outlined in Figure 1, the risk assessment team completes five steps from an early brainstorming to actually assigning abstract risk estimates. These are structured by individual preparatory steps to limit the cognitive load and improve the repeatability of the process. The result is an abstract model of the risks in an application area based on expert knowledge.

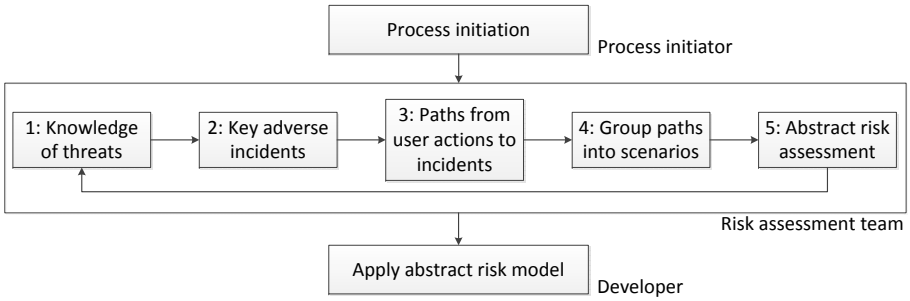

Fig. 1. Overview of the methodology to construct the abstract risk model

Also shown in Figure 1, the process initiator as step zero firstly defines the application area and the scope of risks to cover – for example, limiting the analysis to risks in Web browsing. Secondly, this role recruits security practitioners and domain experts as members of the risk assessment team. The result of the

abstract risk assessment, the abstract risk model, is then used by developers to implement the algorithm for when and how to intervene.

In the following, we describe each of the steps in turn. For illustration, we use a limited running example from Web browsing.

3.1 Structure knowledge of threats

As the first step, the risk assessment team needs to structure their explicit and implicit knowledge on a high level to support the later steps of the method. At this point, the emphasis lies on the relevant threats, that is, what potentially could happen, and how the threats relate. Explicit knowledge primarily refers to threats mentioned in documents, such as white papers, on security in the application area. Members of the team collect these documents and extract threats prior to the actual workshop on an individual basis. Implicit knowledge is the additional knowledge from training and experience that experts can employ to complete the picture (add missing threats) and connect the dots – relate the threats.

In this process, the structuring of the knowledge of threats occurs as a mind-mapping exercise. The goal is to lay out how actions of users potentially lead to adversary actions and to consequences for the user. The resulting graph should show how user actions and threats interrelate. Edges symbolize potential causations, that is, a next step in an attack or a consequence. We call the result "threat dependency graph" (TDG). Figure 2 shows a limited example for Web browsing to illustrate the concepts.

The risk assessment team can employ typical mind-mapping tools, beginning with offline media, such as whiteboards, and later digitalizing the resulting graph.

3.2 Identify key adverse incidents

The later risk assessment should be based on the actual impact for the end user. Accordingly, the granularity for the results of the risk assessment needs to corresponds with relevance of threats for end users. Specifically, in the example in Figure 2, it might be difficult to assess the impact of a technical threat such as "Eavesdropping" for the end user, while it is realistic to assess "Spam."

Since both types of threats, directly relevant and rather technical, are present in the TDG, the risk assessment team explicitly has to identify the relevant threats in the TDG in this step. The main question to ask for each threat in the TDG is, "Does the threat have a direct consequence to the end users?" That is, for example, do they incur a financial loss or have a negative perception. We call these "adverse incidents." In the example in Figure 2, they are marked red.

3.3 Elicit paths from user actions to incidents

To assess risks, the assessment team often need to consider how an attack would occur. To provide this information, the team in this step needs to elicit the

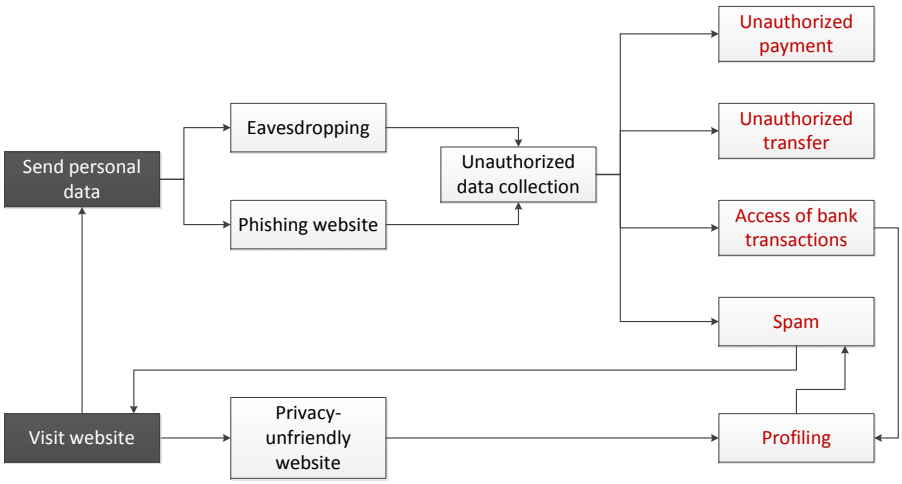

Fig. 2. A limited example of a threat dependency graph for Web browsing – result of the threat brainstorming; boxes with dark background represent user actions

relevant paths that connect user actions with incidents. These paths are already present in the TDG through the edges between threats, but directly considering all paths through the TDG will result in an unmanageable number for the risk assessment. Thus, in this step, the team explicitly chooses the relevant paths from the entirety in the TDG.

For the limited example, a detail from the result of this step is shown in Figure 3. The paths are represented here as a tree with the root representing a user action (here: "Send personal data") and the leaves being the incidents. Other incidents from the TDG, such as "Spam," have been left out since these appeared to be irrelevant in comparison to others, such as "Unauthorized payment." Through this selection, a first risk assessment step is conducted, since paths are excluded if they are redundant or obviously irrelevant. The process of selecting paths is technically supported from the TDG data.

As an alternative visualization, the roots can also represent the incidents and leaves user actions, then resembling attack trees [11].

3.4 Group paths into scenarios

Typically the previous step results in multiple relevant paths from user actions to incidents. An example can be found in Figure 3 in the case of "Unauthorized transfer", which can be a result of a phishing website or of eavesdropping on an unprotected connection. These paths will have very different risk assessments – in the above example, the trustworthiness of a website will be differently gauged than the connection security. To differentiate these risks in the assessment, the risk assessment team groups these paths as scenarios in this step. In the above

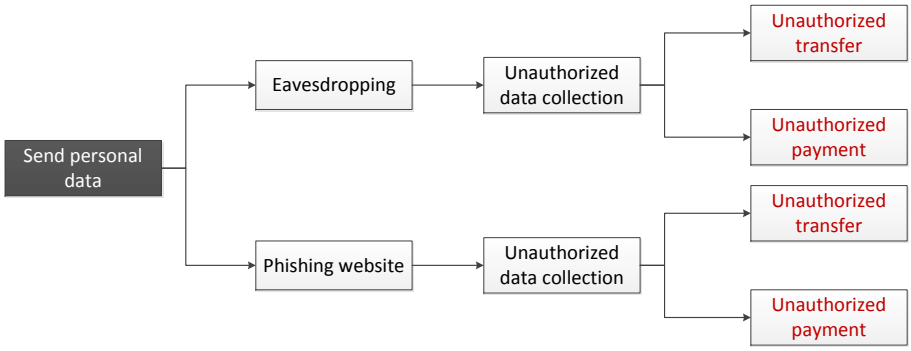

Fig. 3. Detail of a tree with paths from user actions to adverse incidents

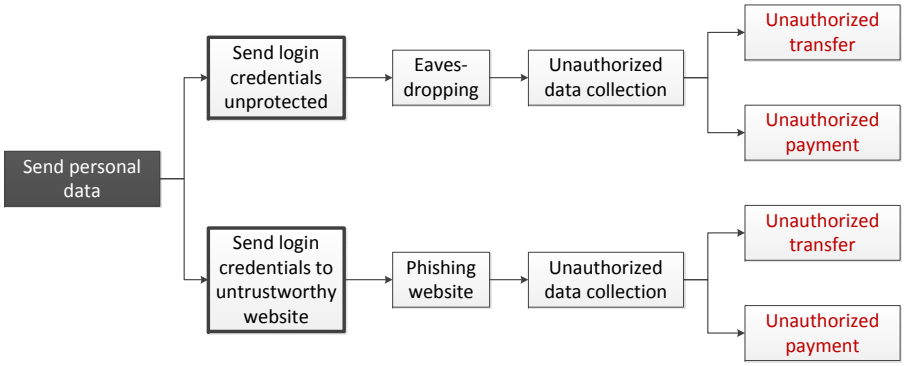

Fig. 4. Scenarios (thick border) added to paths from user action to incidents

example, this could result, as shown in Figure 4, in "Send login credentials unprotected" for one path, and in "Send login credentials to untrustworthy website" for another. Note, that each scenario typically encompasses multiple incidents, but only one relevant path to each incident.

3.5 Conduct abstract risk assessment

The previous steps have resulted in adverse incidents to be assessed, paths leading from user actions to the incidents, and groupings of user actions and paths to incidents as scenarios. Given these artifacts, the risk assessment team can abstractly assess the incidents' risks. The risk assessment should result in an abstract risk model that may then be applied for a concrete situation – for example, an actual security status of the connection.

The team conducts the risk assessment per scenario. Within a scenario, all relevant incidents are considered – that is, all incidents that are endpoints in the paths from user actions in the scenario. For each incident, two decision trees are

constructed, one for the probability of an incident and one for the severity of the incident. These are split to follow the independent assessment of probability and severity in qualitative risk assessment (cf. e.g. NIST 800-30 [12]). A detail of an example decision tree for the probability of "Unauthorized transfer" in the scenario "Send login credentials unprotected" is given in Figure 5.

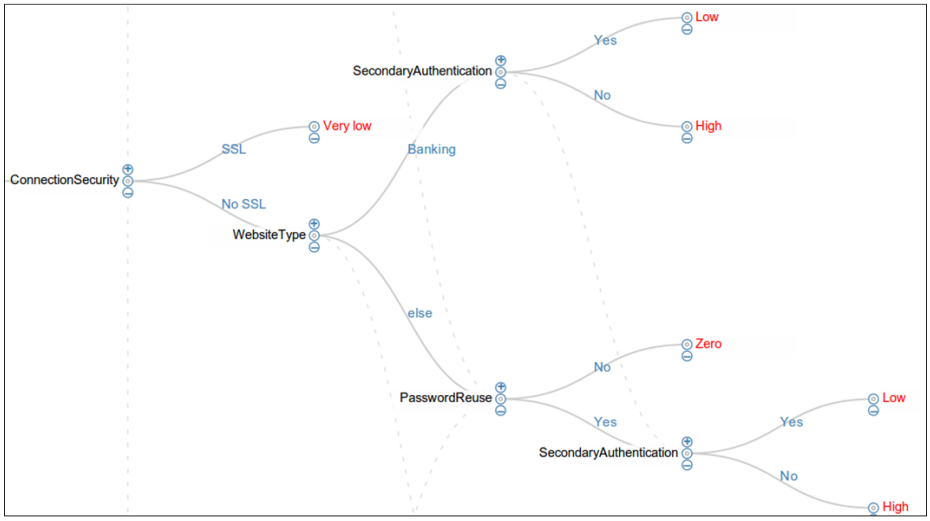

Fig. 5. Detail of a decision tree for the probability of the adverse incident "Unauthorized transfer," scenario "Send login credentials unprotected," nodes are indicators

The nodes of the decision trees represent indicators that describe the concrete situation – for example, the connection security. Edges from an indicator represent the different options for the indicator, e.g. EV-SSL, SSL, no protection for connection security. The risk assessment team identifies appropriate indicators from their security and domain knowledge. Leaves of the decision tree are the outcomes, either probabilities or severities. Since users may perceive different types of risks differently [14], severities are qualified by their type – for example, "financial" or "social" risk. The risk assessment team defines these categories as appropriate for the domain, but may start from findings on risk perception (e.g. from [14]).

Interventions are only appropriate, if users are likely to eventually take the action that leads to an incident. For example, if no password field is on a webpage, we can exclude the immediate danger of login credentials being transmitted. Therefore, each scenario also includes a decision tree for the probability of the respective action.

4 Applying the abstract risk model in practice

Once the risk assessment team has constructed the abstract risk model for an application, developers can implement it for the decision of whether and how to intervene. Specifically, the abstract risk model is used to assess the risk of a concrete situation.

In practice, the developer implements the risk assessment to run each time the situation changes – for example, when a new website is loaded or the connection security switches to HTTPS. In the proposed method, the change in situation can be seen as a change in one of the relevant indicators of the risk model. Thus, the appropriateness of an intervention is rechecked for new information becoming available and further actions that a user takes. For example, as part of the mobile application installation procedure, the assessment is run several times:

1. When an application is selected (indicator "application selected": "yes"),
2. On display of the installation page ("installation button visible": "yes"),
3. On clicking the installation button ("installation button clicked": "yes").

This allows the system to decide each time, whether an intervention is necessary and which is appropriate.

The basic process of risk assessment is shown in Figure 6: The change of an indicator (the situation) leads to an evaluation of all scenarios for the risk levels of their respective incidents. The lists of incidents from all scenarios is then ranked by risk level. Lastly, the maximum of the incidents' risk levels is applied to decide on the appropriateness of available interventions. The list of incidents, ordered by risk level, can also be used for the communication of selected concrete risks to the end user.

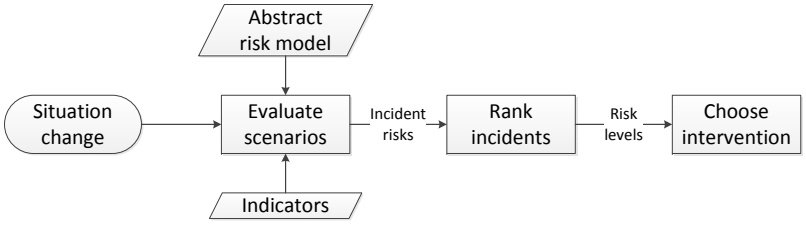

Fig. 6. Process of applying the abstract risk model

To deduce the risk levels for each of the incidents within a scenario, four steps are taken:

1. The decision trees for the incident *probabilities* are evaluated per incident.
2. The decision trees for the incident *severities* are evaluated per incident.

3. The decision tree for the *action probability* of the end user is evaluated for the scenario.
4. The outcomes from the decision trees are combined by transforming the qualitative values into numerical values and multiplication of the numerical values:

$$\begin{aligned}\text{Risk}_{Incident}(Situation) = \text{Probability}_{Incident}(Situation) * \\ \text{Severity}_{Incident}(Situation) * \\ \text{Probability}_{Action}(Situation)\end{aligned}$$

For the decision on the appropriate intervention, thresholds for risk levels of individual incidents are defined. Thus, the type of intervention – for example, whether passive information or an active warning – is deduced from the maximum risk level of the incidents. The risk levels of the incidents are not further aggregated because they are derived from qualitative data.

5 Case study: Web browser warnings

To evaluate the viability of the proposed approach, we applied the method in a case study for Web-browser warnings. Web browsing is an interesting case with its broad range of threats – from eavesdropping to phishing and malicious software downloads – and its broad range of use cases – including online shopping and banking. Users can also interact with Web sites in several ways, beginning with the visiting of a website, but also encompassing the sending of login credentials and personal data, and downloading documents and software. Making sense of this breadth of actions and threats is a good case for evaluating the method for practical viability.

5.1 Constructing the abstract risk model

The first part of the methodology is constructing the abstract risk model. The scope of risks was defined as those relevant to the end user while browsing the Web. The risk assessment team at this point consisted of researchers, the authors of this paper, another security and a legal researcher. We applied the steps as described in Section 3. Key numbers from the process are shown in Table 1. The brainstorming phase (step 1) included combing through scientific and white

Table 1. Count of items in the Web-browsing case study

Item	No.	Examples
Actions	4	Visit website; Submit personal data
Scenarios	9	Send login credentials unprotected; Visit untrustworthy website
Incidents	17	Unauthorized payment; Spam
Threats in TDG	74	Eavesdropping in transport; Collect behavioral data
Threat relations	111	Payment credentials theft → Unauthorized payment

papers on Web risks. The risk assessment team grouped the identified threats and related them with each other for the TDG. Also, key adverse incidents could be identified (step 2). A detail from the resulting TDG is shown in Figure 7.

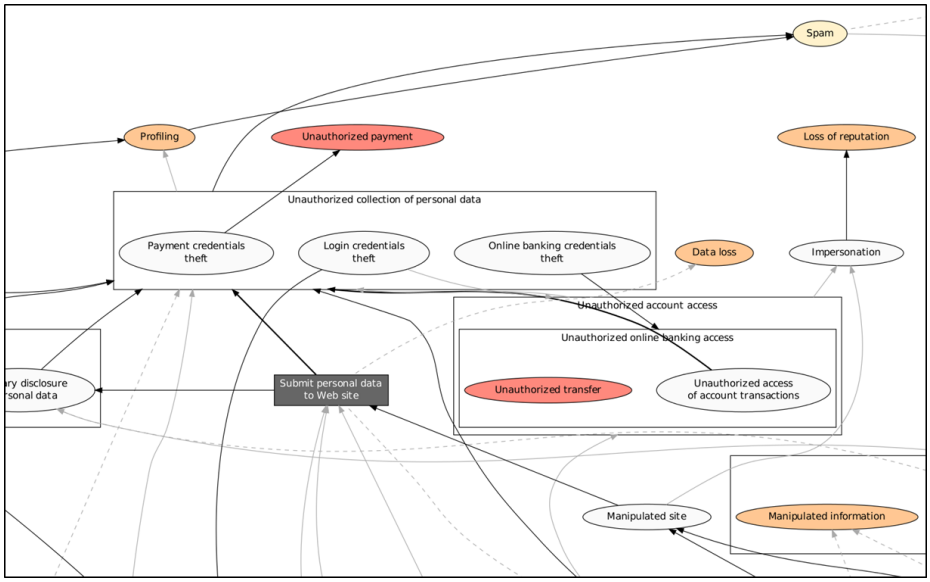

Fig. 7. Detail of the threat dependency graph in the case study; dark gray background represents a user actions

The team elicited paths from user actions to incidents (step 3) in a prototype tool developed for the purpose. As part of this exercise, they added additional threats to the TDG, including data loss for user data on websites. Similarly, the team found that they needed to add relations between threats – for example, the missing link between "Manipulation of website content" and "Submitting personal data" to an untrustworthy website. One important observation was that some experts would prefer to build these trees from the incidents, not from the user actions as originally suggested. For step 4, the team could identify relevant scenarios as groupings of the paths (step 4).

The team also conducted the actual abstract risk assessment in the purpose-built tool by constructing the decision trees per incident for the scenarios. They found that they needed the list of paths between user action and the incident to elicit the relevant indicators. We accordingly implemented an extra output of these paths in the tool. Also, indicators were introduced which might prove difficult to implement in software – for example, whether the current user is reusing login credentials between websites. The team also identified redundancies between the decision trees for different incidents – for example, with regard to whether the user is expected to log in on the website later on. We introduced

the new concept of *Abstract indicators* to capture the notion that some parts of decision trees can be "factored out". For example, the abstract indicator "Will login" takes indicators, such as the website type, into account. It could then be applied as any other indicator in the decision trees. These additions will be evaluated as future work.

5.2 Applying the abstract risk model

The output from applying the abstract risk model to a concrete situation are shown in the Case Explorer, where the input of indicators can be simulated. The result, of which a detail is shown in Figure 8, is a list of incidents for the scenarios with risk levels.

Scenario	Incident	Risk	Risk type
Visiting site unprotected	**Impersonation**	47.5	financial
Visiting site unprotected	**Spam**	0.48	annoyance
Visiting site unprotected	**Unauthorized transfer**	85.5	financial
Visiting site unprotected	**Unauthorized payment**	4.75	financial
Visit untrustworthy site	**Unethical content**	8.55	annoyance
Visit untrustworthy site	**Client-side attack**	4.75	financial
Visit untrustworthy site	**Manipulated information**	8.55	personal
Visit untrustworthy site	**Site annoyances**	8.55	annoyance
Visit untrustworthy site	**Profiling**	0.95	personal
Visit untrustworthy site	**Manipulate behavior**	0.95	personal

Fig. 8. Detail of the Case Explorer showing incidents with their risk level for an unprotected online banking website

Figure 8 shows the case that a website has no SSL protection and is categorized as "online banking." Accordingly, the abstract risk model results in a high risk (85.5) for the incident "Unauthorized transfer." This would justify an active warning, blocking the access to the online-banking site. However, for a different case, not in the screenshot, of an online-shopping site without SSL protection, the risk of "unauthorized payment" is assessed to be moderate (47.5), thus rather suggesting a passive warning in the browser chrome. Once a password field is focused, "unauthorized payment" is evaluated as high risk, justifying an active intervention. In the current implementations of Web browsers, missing SSL protection would not have resulted in an active warning at all, only a passive signal in the browser chrome – for example, a missing lock icon.

6 Discussion

This paper proposes a methodology for constructing and applying abstract risk models. We demonstrate how these models capture expert knowledge and inform the decision whether and how to intervene. While applying the method in a case study on Web browsing, we saw that it is realistic to formalize the expert knowledge and that it can be advantageous to apply the abstract risk model in

practice. Warnings can be more precise and thus less of a hindrance and more of a help. While our evaluation only considered a fraction of relevant cases where a more nuanced decision is helpful, we will explore its practical advantages further in an implementation of warnings in Web browsers as a browser extension.

We also noted a number of potentials for future improvement of the method from the experience that we will take into account in the next iteration of the method. It is also necessary to consider the completeness of the expert knowledge in the abstract risk model and its adaption to changing risks over time. Both aspects can be addressed through the iterative nature of the method: Process steps are expected to be repeated over time if threats are missing or the risks change. Another important observation was that some of the indicators employed by the risk assessment team might not be readily available at a sufficiently high precision, such as the type of a website. We will explore this aspect in future work. A further open issue is the question of accountability for risk-based interventions in comparison to today's rather technical interventions: Will the developer or the risk assessment team be held accountable when a warning is not shown?

References

1. Bartsch, S., Volkamer, M.: Towards the Systematic Development of Contextualised Security Interventions. In: Proceedings of Designing Interactive Secure Systems, BCS HCI 2012. BLIC (2012)
2. Baskerville, R.: Information systems security design methods: implications for information systems development. ACM Comput. Surv. 25(4), 375–414 (1993)
3. Campbell, P.L., Stamp, J.E.: A Classification Scheme for Risk Assessment Methods. Tech. Rep. SAND2004-4233, Sandia National Laboratories (2004)
4. Felt, A.P., Ha, E., Egelman, S., Haney, A., Chin, E., Wagner, D.: Android Permissions: User Attention, Comprehension, and Behavior. In: Proceedings of the Eighth Symposium on Usable Privacy and Security. pp. 3:1–3:14. SOUPS '12, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2335356.2335360
5. Fischhoff, B., Eggers, S.: Mental Models of warning decisions: Identifying and addressing information needs. Routledge (2006)
6. ISO/IEC 27005:2008: Information technology – Security techniques – Information security risk management. ISO, Geneva, Switzerland (2008)
7. Kelley, P.G., Consolvo, S., Cranor, L.F., Jung, J., Sadeh, N., Wetherall, D.: A Conundrum of Permissions: Installing Applications on an Android Smartphone. In: Blyth, J., Dietrich, S., Camp, L.J. (eds.) Financial Cryptography and Data Security, pp. 68–79. No. 7398 in Lecture Notes in Computer Science, Springer Berlin Heidelberg (Jan 2012)
8. NIST: FIPS 65: Guidelines for Automatic Data Processing Risk Analysis. Tech. rep., NIST (1975)
9. Peltier, T.R.: Information security risk analysis. CRC press, Boca Raton, FL (2005)
10. Schechter, S., Dhamija, R., Ozment, A., Fischer, I.: The Emperor's New Security Indicators. In: S&P '07: Proceedings of the 2007 IEEE Symposium on Security and Privacy. pp. 51 –65 (May 2007)
11. Schneier, B.: Attack trees. Dr. Dobb's journal 24(12), 21–29 (1999)
12. Stoneburner, G., Goguen, A., Feringa, A.: Risk Management Guide for Information Technology Systems – NIST Special Publication 800-30. Tech. rep., National Institute of Standards and Technology (2002)

13. Sunshine, J., Egelman, S., Almuhimedi, H., Atri, N., Cranor, L.F.: Crying Wolf: An Empirical Study of SSL Warning Effectiveness. In: USENIX Security 2009 (2009)
14. Weber, E.U., Blais, A.R., Betz, N.E.: A domain-specific risk-attitude scale: measuring risk perceptions and risk behaviors. Journal of Behavioral Decision Making 15(4), 263–290 (2002), http://onlinelibrary.wiley.com/doi/10.1002/bdm.414/abstract