Empirical and statistical analysis of risk analysis-driven techniques for threat management

Koen Buyens, Bart De Win, and Wouter Joosen
DistriNet research group, Department of Computer Science, K.U.Leuven,
Celestijnenlaan 200A, B-3001 Leuven, Belgium
{koen.buyens,bart.dewin,wouter.joosen}@cs.kuleuven.be

Abstract

One of the challenges of secure software construction (and maintenance) is to get control over the multitude of threats in order to focus mitigation efforts on the most relevant ones. Risk analysis is one class of techniques for achieving threat reduction, but few studies are available that evaluate the quality of these techniques. In this paper, a selected set of risk analysis techniques have been evaluated and compared based on a realistic case study. The foundations for this analysis were threefold: we defined a set of high-level criteria, we compared the results of the different methods and we used statistical analysis techniques for studying additional characteristics. This analysis was performed on an independently developed case study of a significant size. For this experiment, the benefits of applying these methods were limited for the categorization and the reduction of threats. Therefore, we also suggest ways to improve or complement these methods.

1 Introduction

Building secure software involves analyzing and controlling the various threats to a system that can be the basis for attacks and, subsequently, lead to successful system compromises. Threats to a system include regular information processing threats, threats specific to the technologies and platform being used, threats related to the business processes and even social engineering threats. For a system of non-trivial size, the number of actual threats that have to be taken into account is often considerable. One of the challenges of the construction (and maintenance) of secure software is to reduce this multitude of threats by discarding low priority ones. As a result, the complexity for addressing and validating the remaining threats will decrease, lowering the cost to build secure software.

The reduction of the list of threats during analysis can be achieved in different ways. A well known approach is to prioritize and discard threats based on the risk they pose for the system and its environment. This approach is taken by most common risk management techniques. Risk, in this context, is basically the cost (or impact) of a threat multiplied by the probability of this threat happening. Other useful techniques could be based on exploiting knowledge on relationships between threats. For instance, the dependency between two threats (as represented in threat trees) can support the removal of the child threat in case the parent threat is considered not relevant. Or, if a group of threats is considered complementary (or optional) the selection of a single relevant element in the set might lead to the pruning of the other elements in the set.

For one of the applications that we are building internally, we were interested in the strength (performance) of risk-driven threat reduction techniques, since the number of identified threats grew rapidly (to over 600). Since the available literature on qualitative studies is fairly limited in this area (e.g., [8]), we decided to take our application as a research vehicle to study, compare and possibly improve these techniques. The primary research question that we wanted to address at this point was "which threat reduction technique performs best?", which essentially boils down to (i) reducing the number of low important threats, while (ii) preserving high priority threats.

The main contribution of this paper is the in-depth evaluation and comparison of a selected set of risk analysis techniques based on a realistic case study. The foundations for this analysis were threefold: the measurement of a set of high-level criteria, the comparison of the raw output of the different methods and the use of statistical techniques for the analysis of a method’s general behavior. This has lead to a number of remarkable observations concerning the quality and the categorization capabilities of these techniques. Furthermore, the paper also discusses some desirable characteristics for risk analysis techniques in general.

This paper is structured as follows. Section 2 elaborates on the methodology and the criteria used to compare differ-
ent risk analysis techniques. Section 3 gives a description of the application that was used as a starting point for the evaluation. Section 4 presents and discusses the results of the application of each of the different methodologies based on the identified criteria. Section 5 further analyzes the techniques used in the experiment, among others by applying statistical analysis techniques. Finally, section 6 concludes our findings and describes future work.

2 Method and criteria for evaluation

This section describes the approach taken for the evaluation of the different risk analysis methods, as well as the set of evaluation criteria and corresponding metrics.

The evaluation of the methods has been based on an experiment, which purpose was to simulate the use of the different methods. Three different techniques have been used as part of the analysis: (i) a number of high-level criteria (and corresponding metrics) were defined to assess the general characteristics of a particular method, (ii) the output of the different methods were compared in order to evaluate the raw performance of the methods and (iii) additional statistical techniques were used to further understand, evaluate and compare the effects of a method. Note that, since this evaluation was based on a single experiment executed by one person, the correctness of the results cannot be claimed with confidence: the population is limited and the values provided to the methods can be subjective. However, because of the size of the experiment, the authors are confident in the overall outcome of this evaluation.

In order to have a fair comparison of methods, the experiment was set up as follows. A uniform list of threats was fixed upfront in order to act as the input of the different methods[4]. As a result, activities covering threat elicitation that are specific to a particular risk analysis method have not been used. This approach was chosen to focus the evaluation on the core reduction method. Using this fixed input set, similar input has been provided to the different reduction methods. In many cases, this meant transforming values (in case different scales were being used) or introducing new values to be used as input (in case new types of parameters were being used). For the generation of these values, we made sure to respect the subjective “risk characteristic” of a specific threat. Although one cannot exclude the introduction of errors, care was taken to generate the input with sufficient precision. For the evaluation of the output, we opted to compare the outputs of the different methods, rather than comparing these with a model of the output, since the latter is hard to obtain, and subjective without a good set of criteria (which then constitutes a risk analysis method in itself).

Based on SQUARE[8] we identified the following high-level criteria (and corresponding metrics) for comparing the characteristics of risk assessment reduction methods:

1. **Time frame**: This criterion evaluates the time required to complete a reduction activity (measured in hours). This measurement includes both the method’s complexity and its overhead. In our opinion, previous experience with the methodology may influence the time needed to complete the reduction step, but does not significantly alter the result. The person conducting the experiment applied each methodology for the first time. Note that the time measured does not include the time to collect additional data.

2. **Additional data collection**: Starting from a list of threats, what extra data has to be collected in order to execute the method? The metrics being used are: (i) the number (and type) of additional data required and (ii) the time needed to collect this additional data.

3. **Complexity**: This criterion evaluates the difficulty of a reduction technique. The metric being used is the number of parameters of the risk reduction formula. Note that this metric only gives a rough idea of the difficulty in understanding the elicitation technique; the measuring of errors made during the execution of a reduction technique is more difficult.

4. **Learning curve**: The speed with which applicants can comprehend the reduction technique. The metric used is the time needed to study the elicitation technique and to apply it to a toy example. The time is categorized as follows: less than a day (fast), less than three days (normal), and longer than three days (slow).

In the following section, the case study used in the experiment is briefly explained. Afterwards, the results of the experiment are presented and discussed.

3 Case study: publishing system

This section describes a simplified version of a publishing system, its assets and associated threats. A more elaborate version of the publishing system can be found in[7] and [4].

The main features of the publishing system (see Figure 1) are input management, content management, content distribution, and user management. In the architecture, the components responsible for these features are the following. An Input Management System (IMS) is used to annotate and prepare produced content to be stored in the Content Management System. The Content Management System (CMS) is responsible for storing and retrieving content items, and a Service News Desk is responsible for making content ready to be published. A Newspaper Service (NS) is utilized for distributing editions towards the media consumer and an User Management System (UMS) takes care
of authenticating, authorizing, and accounting users that are using services of the publishing system.

The main actors involved in the publishing system are the input source, the media consumer and the service news desk worker. The input source is the entity that produces content. This can be, for instance, an author, or a musician. At the other end of the content consumption chain, the media consumer is the individual who wants to obtain and consume content. For example, a home user who wants to download and read the news of the day. The service news desk worker uses the publishing system to distribute finished content, and forms the bridge between consumers and producers.

In order to illustrate the functionality of the system, consider the following example scenario. Suppose that a Media Consumer wants to obtain the latest newspaper, which has a free book attached with it. In order to do this, he contacts the Service Controller, that has three successive tasks. First, the service controller obtains the media consumers credentials and sends them to the User Management System, which verifies these credentials. Second, it contacts the Newspaper Service, that obtains the newspaper from the Content Management System. Finally it forwards the reading material to the consumer.

The remainder of this section lists the main assets and threats of the system. Two relevant types of assets were identified: (i) information, which is the business product; and (ii) infrastructure, which aids the primary business model. For the example scenario, the information assets are the media consumer credentials, which are used to identify media consumers; the newspaper, which can be seen as an edition limited in time; and the additional book, which is not limited in time and a free asset. The infrastructural assets are the software and hardware of the newspaper service, the CMS, the UMS and the client of the media consumer.

The publishing architecture is susceptible to numerous threats, including information disclosure of an edition limited in time, tampering of media consumer credentials, denial of service of the Newspaper Service, and spoofing of the Newspaper Service. The complete list of identified threats can be found in [4].

4 Risk reduction experiments

This section describes a selection of risk reduction methodologies, adaptations to the methodologies where relevant, and results for each comparison criteria.

There are dozens of risk reduction methodologies, but for reasons of input and practicality we decided to focus on four of them: (i) DREAD, a risk reduction technique devel-
oped by Microsoft and the variant described by OWASP; (ii) NIST SP800-30, a risk management methodology developed by the National Institute for Standards and Technology; (iii) OCTAVE-S, a risk management methodology developed by the Software Engineering Institute (SEI); and (iv) CORAS, the synthesized result of an EU/IST project.

Other methodologies including AS/NZS 4360 were not selected due to the lack of publically available information.

4.1 DREAD MS

The main focus of STRIDE/DREAD[10], as defined by Microsoft1 is the identification and prioritization of threats during the development phase of an application. These two elements, identification and prioritization of risks, are realized in the STRIDE and DREAD methods, respectively. This paper focuses on DREAD, the risk reduction part of the methodology. DREAD defines the following five key attributes to measure the criticality of a vulnerability:

- **Damage potential (dp)**: If the threat exploit is successful, how much damage will be caused?
- **Reproducibility (r)**: How easy is it to reproduce the threat exploit? What is the cost to the attacker once he has a working exploit for the problem?
- **Exploitability (e)**: What is needed to exploit the threat? What is the cost to develop an exploit for the problem?
- **Affected users (a)**: What users are actually affected if an exploit were to be widely available?
- **Discoverability (di)**: How easy is it to discover a threat?

Risk is defined as the sum of these attributes:

\[
risk = dp + r + e + a + di
\]

Microsoft recommends to assign to each attribute a rating between 1 (low) and 3 (high). The sum of ratings falls in the range of 5 to 15. Threats with overall ratings of 15 to 12 are classified as high risk, 11 to 8 as medium risk, and 7 to 5 as low risk.

The DREAD methodology was applied to the initial set of threats without modification. This resulted in the following measurements:

**Time frame.** Applying DREAD took approximately 17 hours for 187 threats. The measured time included the application of three activities, namely rating each parameter using DREAD’s rating table[10], motivating the given ratings, and reviewing the given ratings.

**Additional data collection.** Additional information collection has been performed during approximately six hours for three types of information: attacker profiles, timing windows, and a list of features used by the systems users. An attacker profile describes the skills of the attacker executing the threat and is used for determining exploitability. The attacker profiles were reused for comparable threats. A timing window is defined as the time period when the threat can be executed and is used for determining reproducibility. However, attacker profiles and timing windows were not really useful, because the execution of identified threats was not limited in time. The list of features describes which features are used by whom and is used for determining affected users and discoverability. The list of features was already available in the form of use cases and, hence, no time was spent on this.

**Complexity.** The DREAD risk formula has five parameters (damage potential, reproducibility, exploitability, affected users, and discoverability) which are combined in a linear way.

**Learning curve.** DREAD was learned fast.

**Number of threats.** DREAD classifies almost half of the threats as high (81), approximately half of the threats (99) as medium and few threats (7) as low. The large number of high and medium categorized threats should be mitigated in order of importance. For each low threat, someone should check whether it needs mitigation or not.

4.2 DREAD OWASP

The DREAD risk reduction method as explained at the OWASP website[9] is slightly different from the one described by Microsoft, because OWASP gives a more formal definition of the attributes, introduces weights, and gives a different value to each attribute. Risk is defined as the sum of the weighted attributes:

\[
risk = w_{dp} \cdot dp + w_r \cdot r + w_e \cdot e + w_a \cdot a + w_{di} \cdot di
\]

OWASP advises to give each attribute a rating between 1 and 10, to rate discoverability (di) for applications under security revision 10, and to weight affected users (w_a) and damage potential (w_{dp}) attributes more than the other attributes. The sum of the weights equals 1, and consequently the sum of all given weighted ratings falls in the range of 1 to 10.

The OWASP DREAD methodology was slightly extended for the experiment. OWASP does not mention risk category ranges that assign threats to high, medium, and low categories, hence we introduced ranges ourselves. Threats with overall ratings between 10 and 7 are classified as high risk, between 7 (exclusive) and 4 are classified as medium risk and between 4 (exclusive) and 1 as low risk. This seems like a reasonable distribution over the output.

---

1DREAD is no longer advertised by Microsoft. The threat modeling stage of its new methodology (SDL[6]) assigns risk categories, based on the threat type. For instance, permanent modification of data at the client has a major risk.
The remainder of this section describes the results related to the comparison criteria and elaborates the number of threats in each risk category.

**Time frame.** Applying DREAD OWASP took 17 hours for 187 threats. The time needed was the same as the time spent on DREAD MS. However, the definition of OWASP parameters was slightly different and the weights were already defined.

**Additional data collection.** For the collection of additional information, approximately five hours were spent on two types of data: attacker profiles and weights. Attacker profiles were needed for determining a threat’s exploitability and were reused for most threats, while the initial weights were needed to calculate the risk reduction formula. The initial weights for each of the parameters were 2/7, 1/7, 1/7, 2/7, and 1/7 respectively.

**Complexity.** DREAD OWASP has five parameters: one for each letter of the acronym.

**Learning curve.** DREAD OWASP was learned fast.

**Number of threats.** Most of the threats were categorized as high (115), one third as medium (72) and none as low. OWASP does not suggest the handling of particular types of risk. Similar to the Microsoft methodology, it would be natural to handle high and medium threats, and to determine for each low threat whether it is acceptable.

### 4.3 NIST SP800-30

The National Institute for Standards and Technology’s special publication 800-30 provides guidance on risk assessment and risk management practices. Its nine step methodology describes a systematic approach to risk assessment with guidance on inputs and outputs from the processes elaborated in each step. We are interested in the following steps of the NIST risk assessment approach:

1. **Step 5: Likelihood Determination.** The probability that an attacker may exploit a potential vulnerability in an information asset is determined during this step.
2. **Step 6: Impact Analysis.** Impact resulting from a successful exploitation of a vulnerability by a threat source is determined during this step.
3. **Step 7: Risk Determination.** During this step levels of risk for each threat are determined.

The proposed risk reduction method is defined in terms of impact and likelihood. NIST’s impact is defined as a combination of loss of assets, harm of system mission and injury of humans. Its likelihood is defined in terms of attacker (threat-source) motivation and capability, nature of the vulnerability and existence and effectiveness of current controls. These parameters can be rated qualitatively (categories) or quantitatively (numbers). The exact formula for quantitative NIST is:

\[
\text{risk} = \text{impact} \times \text{likelihood}
\]

\[
\text{impact} = \max(\text{loss}, \text{harm}, \text{injury})
\]

\[
\text{likelihood} = \begin{array}{ccc}
\text{Likelihood} & \text{Motivations} & \text{Controls} \\
\text{high} & \text{high} & \text{AND} & \text{high} \\
\text{medium} & \text{medium} & \text{AND} & \text{medium} \\
\text{low} & \text{low} & \text{OR} & \text{low}
\end{array}
\]

NIST combines impact and likelihood in a straightforward way. Threats with overall ratings between 100 and 50 (exclusive) are classified as high risk, between 50 and 10 (exclusive) are classified as medium risk and between 10 and 1 as low risk.

For our experiment we opted for the qualitative method, because it was easier to compare to qualitative DREAD and we did not have any historical information.

The remainder of this section elaborates on the results related to the comparison criteria.

**Time frame.** Applying NIST took approximately as long as DREAD, namely 16 hours for 187 threats.

**Additional data collection.** For the collection of additional data, approximately six hours were spent on five types of information: attacker profile, available controls, organization mission, system and data criticality, and finally system and data sensitivity. The attacker profile and the available controls are used to determine the likelihood of the threat. The organization mission is used to determine the impact of the threat. The system and data criticality and sensitivity are used to determine the value of the assets. In this experiment, the organization mission and the available controls were not useful. (The former because there was no organization, and in consequence no explicit organizational mission; the latter because the system was not build yet, so controls against attacks were not available.) Obtaining data sensitivity and criticality was done using our domain knowledge.

**Complexity.** NIST has 2 parameters consisting of 5 subparameters.

**Learning curve.** NIST was learned at normal rate.

**Number of threats.** NIST categorizes 52 threats as high, 54 as medium and 81 as low. The high and medium categorized threats should be mitigated in order of importance. For the low threats, someone must determine whether actions are required.

### 4.4 OCTAVE-S

The Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) approach[1, 3, 2] defines a risk-based strategic assessment and planning technique for security that addresses organizational as well as technical information security risks.
The OCTAVE methodology comes in two versions depending on the size of the organization. We focus on the risk reduction method of the small version, OCTAVE-S. Risk is defined in terms of impact, and optional probability. Impact consists of a selection of at least five weighted parameters: reputation/customer confidence, financial loss, productivity, safety/health, fines/legal penalties, and self-defined criteria. Each parameter possibly consists of several subparameters.

\[
risk = impact = w_r \ast reputation + w_f \ast financialloss + w_p \ast productivity + w_s \ast safety + w_l \ast legalpenalties + w_o \ast others
\]

In our experiment, we applied OCTAVE-S for all assets, not only for the critical ones, because we wanted to compare the results of this methodology with the other ones. We did not use probability, because we were applying OCTAVE-S for the first time and by consequence lack objective data related to threats. We did not select the health parameter, because it was not relevant.

The remainder of this section describes the results related to the comparison criteria and elaborates the number of threats in each risk category.

**Time frame.** Applying OCTAVE-S took much longer than the previous methodologies, namely approximately 28 hours.

**Additional data collection.** For the collection of additional data, five hours were spent on five types of data: actor history, actor motivation (included its profile), security requirements for the critical assets, financial loss, fines, and productivity. Actor history is a list of the previous actions of the attacker, and actor motivation is the motivation for executing an attack. In this experiment, security requirements were not useful, because we use the outcome of risk assessment to identify security requirements. Attacker history was not useful either because we did not have attacks on a non-existing system. Information concerning fines, productivity, etc. were gathered by using our domain knowledge, but are subjective, and differ probably from reality.

**Complexity.** In this experiment, OCTAVE-S contains 4 parameters with several subparameters.

**Learning curve.** OCTAVE-S was learned slowly due to the complicated methodology.

**Number of threats.** OCTAVE-S categorizes 1 threat as high, 78 as medium and 108 as low. OCTAVE-S suggests to look at each individual threat and decide whether to accept or mitigate it. We opted for mitigating the high and medium threats, while accepting the low ones.

### 4.5 CORAS

The CORAS[5] project developed a tool-supported methodology for model-based risk analysis of security-critical systems, which integrates aspects from different risk assessment methods and modelling methodologies.

CORAS’ risk reduction method is a variant of the general risk reduction formula, which is defined in terms of impact and likelihood. CORAS defines impact (consequence) in terms of loss of value for the targeted asset and likelihood (frequency) as the probability that an unwanted incident occurs:

\[
risk = consequence \ast frequency
\]

Both ratings can be estimated based on historical information. If the rating is not sufficiently reliable, one should apply other techniques, such as FME(C)A, Markov-models and Fault Tree Analysis. CORAS can be applied in a quantitative or qualitative way (see 4.3).

In our experiment we opted for qualitative CORAS and reduced the number of risk output categories from four to three in order to compare CORAS to the other methodologies. The reduction consisted of merging CORAS moderate and major category into a medium category. This results in the following measurements:

**Time frame.** Applying CORAS took longer than most other methodologies, namely approximately 27 hours, because the additional techniques were harder to apply.

**Additional data collection.** Two types of data were necessary: historical data and threat trees. For the former, we chose not to take these into account. The latter were already available because of the application of STRIDE for generating threats. Hence, no additional time was spent on collecting data.

**Complexity.** For this experiment, CORAS contains 2 parameters.

**Learning curve.** CORAS was learned rather slowly due to the additional algorithms.

**Number of threats.** CORAS categorizes 69 threat as high, 118 as medium and 0 as low. The medium and high categorized threats should be mitigated.

Table 2 summarizes the application of the different methodologies. In this experiment OCTAVE-S and NIST categorize a lot of threats as low, approximately one third of the threats as medium, and few threats as high. DREAD and CORAS categorize a few threats as low, approximately one third of the threats as medium, and a lot of threats as high.

### 5 Discussion

In order to better understand the characteristics, benefits and drawbacks of the different reduction formulas, we applied further analysis techniques. This section first briefly describes the techniques that were used for this purpose and afterwards elaborates on the results of this analysis.

The analysis techniques included:
<table>
<thead>
<tr>
<th>Methodology</th>
<th>High</th>
<th>Medium</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>DREAD</td>
<td>20.99%</td>
<td>70.37%</td>
<td>8.64%</td>
</tr>
<tr>
<td>DREAD OWASP</td>
<td>20.94%</td>
<td>73.12%</td>
<td>5.94%</td>
</tr>
<tr>
<td>CORAS</td>
<td>24.00%</td>
<td>52.00%</td>
<td>24.00%</td>
</tr>
<tr>
<td>NIST SP800-30</td>
<td>29.63%</td>
<td>50.93%</td>
<td>19.44%</td>
</tr>
<tr>
<td>OCTAVE-S</td>
<td>20.99%</td>
<td>70.37%</td>
<td>8.64%</td>
</tr>
</tbody>
</table>

Table 1. Overview of results of the distribution analysis.

<table>
<thead>
<tr>
<th>Methodology / # classified threats</th>
<th># High</th>
<th># Medium</th>
<th># Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>DREAD MS</td>
<td>81</td>
<td>99</td>
<td>7</td>
</tr>
<tr>
<td>DREAD OWASP</td>
<td>115</td>
<td>72</td>
<td>0</td>
</tr>
<tr>
<td>NIST SP800-30</td>
<td>52</td>
<td>54</td>
<td>81</td>
</tr>
<tr>
<td>OCTAVE-S</td>
<td>1</td>
<td>78</td>
<td>108</td>
</tr>
<tr>
<td>CORAS</td>
<td>69</td>
<td>118</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2. Number of high, medium and low categorized threats for each methodology.

- **K-means**: The k-means method is a clustering algorithm that provides a categorization of the input without taking into account the reduction formula, nor the output. As such, it evaluates the grouping quality of a formula.

- **Classification trees**: Classification trees were built by using information entropy as criteria. The input (e.g., d=3,r=2,e=1,a=2,d=3) and output (e.g., high, medium and low) of a methodology under review (e.g., DREAD) can be used to determine the importance of each variable (e.g., d,r,e,a,d), which aids in understanding a risk reduction formula.

- **Sensitivity analysis**: Sensitivity analysis is the study of how the variation in the output (e.g., risk categories) of a model changes when the input (e.g., input value tuples) is slightly modified.

- **Distribution analysis**: Distribution analysis is used to determine the distribution of the output values (e.g., risk categories), given all possible input value tuples (e.g., d⊗r⊗e⊗a⊗d). Note that every possible tuple is given once to the formula.

We discuss remarkable results of this extended analysis by elaborating on reduction of threats, category assignment of threats, risk category distribution, relationship between threats and finally optimization of risk reduction methodologies.

**Reduction.** Table 2 summarizes the number of threats that were classified as high, medium and low risk per methodology. As can be seen in the table, DREAD, DREAD OWASP, and CORAS are rather conservative in their removal of threats, because these methodologies categorized approximately half of the threats as high risk, half of the threats as medium risk and almost none of the threats as low risk. On the other hand, NIST SP800-30, and OCTAVE-S are rather aggressive in its removal of threats, because almost half of the threats were categorized as low risk. The conservativeness of each methodology can be explained by two factors: distribution analysis and importance of parameters including assigned ratings, which will be discussed next.

**Distribution analysis** (see Table 1) reveals that the low portion of DREAD, DREAD OWASP and OCTAVE-S is much smaller than the medium and high portions, which means that it is more likely for a threat to be ranked high risk or medium risk than low risk. NIST’s and CORAS portion of low threats is much bigger than the low portion of the other methodologies, hence it is more likely for a threat to be categorized as low risk by NIST and CORAS than by other methodologies.

According to the classification tree experiment, the values assigned to important parameters influenced the results of DREAD, NIST SP800-30 and OCTAVE-S. DREAD’s affected users parameter has the most influence on the risk categorization. This parameter was typically rated as high in this experiment and, hence, DREAD was more likely to rate threats as high. For NIST, the motivation parameter has the most influence on the risk categorization according to the classification tree experiment. This parameter was typically rated as low, possibly due to the experiment, which results in low likelihood. Hence, NIST was more likely to rate threats as medium and low risk. OCTAVE-S’ fines parameter, which was an important one, was typically rated as low, hence OCTAVE-S was more likely to rate threats as medium and low risk.

**Category assignment.** The risk category assigned to a threat can be different for each methodology. Small differences, for instance rating a threat high risk instead of medium risk, or rating it low risk instead of medium risk or vice versa can be explained by the methodology’s characteristics. For instance, a methodology can give some parameters a higher weight than another methodology, or use different parameters. However, we also noticed a complete mismatch between two methodologies. Take for instance NIST, which classifies several threats as low, while other methodologies classify these threats as high. A more detailed look at NIST’s reduction formula, reveals that a low value for motivation results in most of the time low or medium risk. This parameter has a much smaller influence on low risk in for instance DREAD or NIST, as demon-
strated by a k-means analysis.

**Distribution.** A number of observations concerning the distribution of the risk assignment of each methodology are interesting. It is to be expected that a risk reduction methodology distributes threats uniformly across the different risk categories, or classify a significant portion of the identified threats as low, because one will typically focus on high risk threats due to financial and/or time constraints. A uniform distribution means that a threat has a one third chance for being classified as high, medium or low. However, distribution analysis reveals that the low category of most methodologies is very small compared to the high and medium portions.

Note that the distribution analysis is a theoretical means of analyzing a formula and practical applications can differ from this (see Table 1 and Table 2). For instance, in our experiment OCTAVE-S has a lot of threats categorized as low, while distribution analysis reveals that only approximately 8% of all threats should be categorized as low.

**Relationship.** Every method is different and it would be nice to combine the benefits of all methods in a single formula. A way to achieve this, is by taking into account relationships between parameters of different formulas. For example OCTAVE-S' productivity can be partially mapped onto DREAD’s affected users and NIST’s humans, because the productivity of a worker decreases if his tools are broken, the people he is working with are injured or the worker himself has been hurt. As another example, DREAD’s exploitability is similar to NIST’s capability of the attacker, because in DREAD an attackers capability is used to calculate exploitability. We are planning further work in this direction, but a first effort of combining all parameters into a global formula produced no useful results.

**Optimization.** Since risk management is often an expensive operation, a possible optimization could consist in removing threats related to unimportant assets and/or misactors before applying the actual risk reduction formula, by using business value and/or trust respectively. Most methodologies have parameters related to the value of the assets, hence an initial reduction should work. However, the complete methodology takes a lot of parameters into account, while this initial reduction only reduces on asset value. So, there is a trade-off between time and accuracy and further experiments will be useful to compare the benefits and drawbacks.

6 Conclusion and Future Work

In this paper five techniques for risk analysis (DREAD MS, DREAD OWASP, NIST SP800-30, OCTAVE-S and CORAS) have been evaluated and compared by applying them in the context of a digital publishing case study. The analysis included the measurement of high-level criteria, the comparison of raw output and the statistical analysis of behavioral characteristics. In summary, most techniques were quite conservative in reducing the number of threats for our experiment but there was a clear mismatch in category assignment between NIST and the other techniques. Also, the theoretical distribution of output values puts a strong focus on medium level threats, which significantly reduces the benefit of these methods because of the method’s indecisiveness.

In the future, we would like to repeat the same evaluations for other experiments in order to get confirmation and increase our confidence in the current results of this work. Furthermore, it would be interesting to master the relationships between different risk analysis formulas and use this as a basis to construct a single formula that combines all relevant parameters of the different methods. This could then be further optimized by techniques such as asset and threat reduction. Finally, we will definitely try to combine risk analysis techniques with other reduction techniques (such as the ones exploiting relationships between threats) in order to improve the overall technique of threat reduction.

Acknowledgments The authors would like to thank Johan Grégoire for his valuable comments on the paper and for proof reading the text.

References