Risk-based Design Security Analysis

Laurens Sion, Koen Yskout, Dimitri Van Landuyt, Wouter Joosen
imec-DistriNet, KU Leuven
Heverlee, Belgium
{laurens.sion,koen.yskout,dimitri.vanlanduyt,wouter.joosen}@cs.kuleuven.be

ABSTRACT

Implementing security by design in practice often involves the application of threat modeling to elicit security threats and to aid designers in focusing efforts on the most stringent problems first.

Existing threat modeling methodologies are capable of generating lots of threats, yet they lack even basic support to triage these threats, except for relying on the expertise and manual assessment by the threat modeler.

Since the essence of creating a secure design is to minimize associated risk (and countermeasure costs), risk analysis approaches offer a very compelling solution to this problem. By combining risk analysis and threat modeling, elicited threats in a design can be enriched with risk analysis information in order to provide support in triaging and prioritizing threats and focusing security efforts on the high-risk threats. It requires the following inputs: the asset values, the strengths of countermeasures, and an attacker model.

In his paper, we provide an integrated threat elicitation and risk analysis approach, implemented in a threat modeling tool prototype, and evaluate it using a real-world application, namely the SecureDrop whistleblower submission system. We show that the security measures implemented in SecureDrop indeed correspond to the high-risk threats identified by our approach. Therefore, the risk-based security analysis provides useful guidance on focusing security efforts on the most important problems first.

CCS CONCEPTS

• **Security and privacy → Software security engineering**; • **Software and its engineering → Risk management**;

KEYWORDS

Security, design, threat modeling, risk analysis

ACM Reference Format:
Laurens Sion, Koen Yskout, Dimitri Van Landuyt, Wouter Joosen. 2018. Risk-based Design Security Analysis. In *SEAD'18: IEEE/ACM 1st International Workshop on Security Awareness from Design to Deployment, May 27, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3194707.3194710

SEAD'18, May 27, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5727-2/18/05…$15.00
https://doi.org/10.1145/3194707.3194710

1 INTRODUCTION

Security by Design (SbD) and Privacy by Design (PbD) are principles that are increasingly being recognized as essential for the prevention of security- and privacy-compromising flaws in the design [1]. The importance of these principles has recently been reinforced with the introduction of the EU-wide General Data Protection Regulation (GDPR) [7], which effectively mandates privacy by design for software systems that process people's personal data. One way to realize such a *by design* approach for security and privacy is by using threat modeling approaches, which offer a systematic and methodical approach towards analyzing the design for security and privacy threats. Examples of such threat modeling methodologies are STRIDE [10, 11] for security and LINDDUN [5, 24] for privacy. Both use a Data Flow Diagram (DFD) representation of the system and systematically iterate over the elements to identify potential security or privacy threats.

Subsequently, these identified threats (which can be a long list!) have to be assessed to determine their relevance (for example based on likelihood and impact). In existing threat modeling methodologies this involves a very coarse-grained, manual classification of the threats (e.g., applicable/not applicable, or high/medium/low). Hence, the threat elicitation (i) does not consider which threats involve critical system assets, (ii) does not consider types of attackers and how capable they are, and (iii) does not consider how effective existing security and privacy countermeasures are against those types of attackers. Even with the application of risk analysis approaches [9, 12] in conjunction with threat modeling, a substantial and manual analysis effort is required on a per-threat basis.

In this paper, we introduce a risk-based security analysis approach that embeds risk analysis in a threat modeling approach to conduct threat assessment during the threat elicitation and enable subsequent triaging based on the estimated risk. The risk analysis is based on FAIR [9] and explicitly uses of estimates to support uncertainty in the inputs (e.g., in asset values) to the risk analysis. To estimate a threat's risk, Monte Carlo simulations are performed in which samples are taken from the various uncertainty distributions (parameterized by the inputs) and combined according to FAIR [9]. The approach has been implemented in a prototype tool (TMaRA) and is evaluated on a real-world application (i.e., SecureDrop). The evaluation verifies to which extent the risk-based prioritization is a useful metric for focusing security efforts by verifying whether there is a high mitigation rate for high-risk threats and a lower mitigation rate for low-risk threats.

The combination of threat modeling and risk analysis provides a more nuanced picture on the relevance of the elicited threats with a relative prioritization. The integration of both approaches in a single design security analysis activity achieves the following benefits: (i) it provides guidance in triaging threats and focusing on the most important, high-risk threats first, (ii) it supports including and

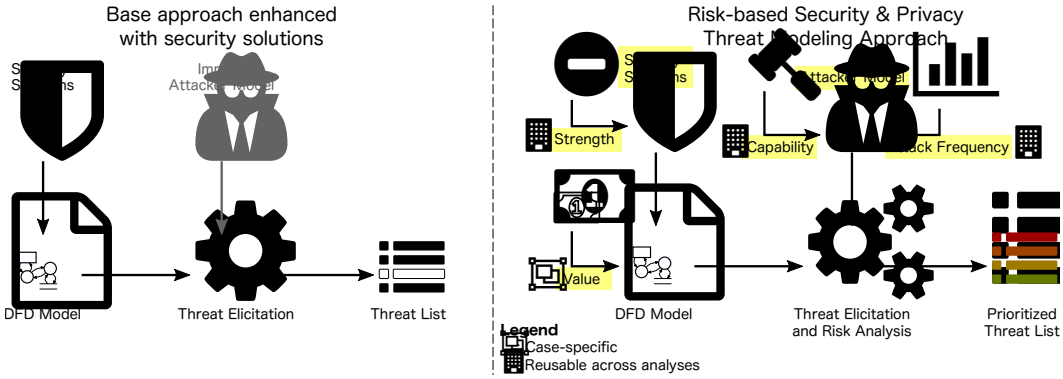

Figure 1: Graphical overview of our approach
The left-hand side shows the base threat modeling approach, enhanced with security solutions to enable the elimination of threats. Note that eliminating threats is a binary *operation; threats are either applicable or not. The right-hand side shows our risk-based approach, with the inputs highlighted in color. The DFD model elements are enriched with values, the solutions' countermeasures are enriched with strength estimates, and the attacker model is made explicit with a concrete capability estimate and attack frequency. In this improved approach, the applicability of threats is determined on a* continuous *scale, based on the calculated risk.*

considering existing security countermeasures in the threat and risk analysis, (iii) it replaces the binary or categorical (high/medium/low) classification of threats with a more nuanced view based on the calculated risk, and (iv) overall risk mitigation progress measured.

This paper is structured as follows. Section 2 provides some background on threat modeling. Section 3 introduces the risk-based threat analysis approach, with the integration of risk analysis in the context of treat modeling. After that, Section 4 looks in detail to the risk analysis itself and the factors it is composed of. Section 5 covers the evaluation, consisting of both a functional validation and an evaluation on a real-world application. Finally, Section 7 discusses related work and Section 8 concludes the paper.

2 BACKGROUND ON THREAT MODELING

This section provides some background information on threat modeling in general, and the previously developed extension [18] we rely on to enable the risk analysis techniques discussed further on. Note that other security or privacy extensions [2, 4, 21] could be used as well. In this paper, we refer to the interaction-based approach to threat elicitation, as discussed by Shostack [17].

Base approach Our approach offers an extension to DFD-based threat modeling techniques [17]. These techniques start from the creation of a DFD model of the system under consideration. This DFD model offers an abstraction of the system using four types of elements: *process*, *data store*, *external entity*, and *data flow*. Besides the four standard DFD element types, DFDs used in threat modeling come with an additional element: a *trust boundary*, to enable the specification of threats that only apply when crossing trust boundaries (e.g., across systems or networks, depending on the context). To indicate when threats should apply, expressions specifying DFD element combinations are used. For example, *spoofing* of an external entity can be found by matching elements to the following expression: '`ExternalEntity–flow–*`'. Each threat type has such an expression to indicate when it can occur in a DFD model. By systematically attempting to match elements in the DFD, all applicable threats can be elicited.

Security and privacy extensions The base approach discussed above does not take into account any information on existing (or foreseen) security or privacy countermeasures. Since security and privacy countermeasures have a considerable impact on the relevance of threats, the analysis also takes these into account.

Existing tools, such as the Microsoft Threat Modeling Tool [13], already have some limited support for the inclusion of this information by allowing properties with security information to be attached to the DFD model elements. These properties specify the effects of the security measures that are already applied. For example, data flows have the property *Provides confidentiality* which can be set to *Yes/No* to specify that some countermeasure, such as encryption, has been applied to provide this security property.

Since security solutions are more complex and often span multiple elements, a more extensive representation is used in our approach. We rely on a more expressive representation of security countermeasures as patterns [18]. These patterns specify roles for the DFD elements involved in the pattern and specify the countermeasures that apply security protections to those roles. This is a much more flexible mechanism for expressing security mechanisms as also enables the specification of asymmetric security effects. For example, consider a typical application of *SSL/TLS* with server authentication, confidentiality and integrity of the data flow, but no client authentication. By representing this solution using a 'TLS is used' Boolean property on a data flow, it's not possible to distinguish between a flow from client to server and a flow in the opposite direction. Therefore, the asymmetric authentication guarantees about client and server cannot be correctly taken into account. By implementing this solution as a pattern with separate roles for the client, server, sending data flow, and receiving data flow, this security solution can be specified in a single pattern which incorporates the differences between the two flow directions.

3 THREAT MODELING RISK ENRICHMENTS

A graphical overview of our approach is presented in Figure 1. As the left-hand side of the overview shows, eliminating security and privacy threats based on the presence of countermeasures provides a very coarse-grained approach towards the applicability of security and privacy threats. Instead, the applicability of security and privacy threats is much more nuanced and is based on (i) the countermeasures (if any) that are applied and how effective these countermeasures are in protecting the assets against a certain type of attacker, (ii) the types of attackers that are considered, and (iii) the value of the assets involved. Security and privacy threat elicitation should take this more nuanced reality into account. The right-hand side of the Figure 1 shows how the incorporation of risk analysis into the elicitation process does exactly that, by including the above information inthe threat assessment.

This section sketches how our approach enriches existing threat modeling artifacts (e.g., the basic DFD model elements and the above-mentioned extensions) with (numerical) risk analysis information. We only provide some intuitions about the required information in this section; a more precise definition, as well as an explanation of how they are combined, are given in Section 4.

Since it is often impossible to determine the required information presented below with absolute certainty (e.g., the strength of a security countermeasure, or the capability of a specific attacker type). Our approach has therefore been designed to take uncertainty into account. In particular, all of the properties mentioned below are expressed as estimates. More details about these estimates will again follow in the next section.

Security Solutions Security solutions are enriched with information regarding their strength. A security solution can contain multiple countermeasures with varying strengths. Therefore, the strength of a security solution is documented in a more fine-grained way as a combination of different *countermeasure strengths*. Consider, for example, TLS, which offers (i) authentication of the server, (ii) encryption of the data, and (iii) integrity checking of the data. Each of these individual countermeasures has a different strength in protecting against a certain type of threat. When multiple countermeasures against a single type of threat are combined, the strength depends on how they are combined. In case of a defense-in-depth strategy, a security solution is only defeated when all countermeasures are broken. When security solutions are applied in parallel (e.g., logging in with a password vs. answering security questions to access an account), they are defeated when any one of the countermeasures is defeated.

Attacker Model The attacker model has three properties relevant for the risk analysis: (i) the capability of the attacker, (ii) the probability of action, and (iii) the frequency of contact.

The *capability* of an attacker is specified on the same scale as the countermeasure strength. It captures the strengths of countermeasures that the attacker is likely able to break. This property of the attacker model is reusable across multiple analyses.

The *probability of action* and *contact frequency* can be more case- or organization-specific, and together they capture the likelihood of an attacker maliciously interacting with the system (attack frequency). To illustrate, Internet-facing applications have a much higher contact frequency and probability of action for many attacker types, as it is easy to mount an attack from somewhere else in the world. For an organization's internal system these numbers will be different as outsiders do not come into contact with it, while employees (that do) will have a much lower probability of trying to attack the system when they come into contact with it. Therefore, these numbers may need to be customized to better fit the system being threat modeled. To reduce the required effort, a set of common attacker profiles can be provided, which could then be further tweaked to fit the analyzed case. The profiles currently provided are: (i) script kiddie; (ii) motivated with limited capability; (iii) motivated and capable; (iv) motivated, capable, and organized; and (v) nation state.

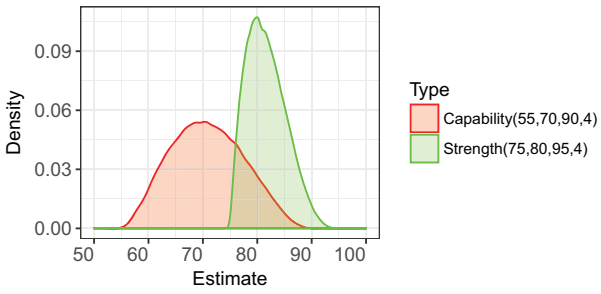

Figure 2: Two example modified-PERT distributions
The first distribution shows the capability of an attacker. The second shows the strength of a countermeasure. The vulnerability is estimated by sampling from both these distributions and comparing the samples.

DFD Model Asset Values The final elements requiring enrichments for risk analysis are the DFD model elements. In order to determine the risk, the potential damage to the assets involved needs to be known. To estimate this, DFD model elements are enriched with *value* estimates that quantify the potential damage.

Since the damages may vary depending on the threat being realized, the enrichments can be specified at this level of granularity. Besides a default value estimate, they can also contain values for specific threat types that override this default. This enables the specification of varying amounts of damage depending on the type of threat. For example, a denial of service attack that temporarily prevents access to a certain database will probably inflict less damage to the organization than an information disclosure attack in which all the customer records are exposed.

4 RISK ANALYSIS

This section explains how the risk analysis itself is conducted in our approach. First, the estimates that are used to support uncertainty in the risk analysis are explained. Next, the risk components themselves, their units, and how they are combined are discussed in more detail.

4.1 Estimates

To support uncertainty in the various parameters used in the risk analysis, estimates are used to enable expressing these uncertain

values with a customizable degree of uncertainty. An estimate consists of four values:

$$\text{Estimate} = (\text{minimum}, \text{probable}, \text{maximum}, \text{confidence})$$

These values are actually the parameters for a modified-PERT distribution [23]. This distribution is a continuous probability distribution which is commonly used in risk analysis (as well as in project management, for example) to take into account uncertainty in expert estimates. Depending on the parameters, the shape of the distribution can range from flat (i.e., the uniform distribution) for very uncertain estimates to very peaked for almost-certain estimates. Figure 2 shows two examples, which express the capability of an attacker and the strength of a countermeasure.

4.2 Risk Analysis Components

Our approach for risk analysis is based on FAIR [9] and inspired by Bedra's [3] application of Monte Carlo simulations for estimating the *vulnerability* (based on the attacker's *capability* and the countermeasure's *strength*). Instead of only performing Monte Carlo simulations for estimating the vulnerability, however, our approach is widened to apply the Monte Carlo simulations for estimating all the relevant FAIR [9] risk components.[1] Furthermore, we directly apply the risk analysis in a threat modeling context, leveraging the availability of the system's design (the DFD model), knowledge about specific threat types, the attacker model, and the applied security solutions. An overview of the combination of these risk components is shown in Figure 3; in essence, for every elicited threat, a specific risk estimate is calculated based on the available information. Each of the subcomponents involved in the risk calculation are discussed next, together with the threat modeling elements they are influenced by.

The **strength** S is used to specify how good a security or privacy countermeasure is in resisting an attacker that tries to break that mechanism. There is no specific unit to express this in, but it should be on the same scale as the *threat capability (TC)*, so the two values can be compared. The strength value is directly tied to a countermeasure against a specific threat. For example, the TLS security solution has an *encryption* countermeasure with a strength of x against information disclosure, and an *integrity checking* countermeasure with a strength of y against tampering.

The **threat capability** TC represents the strength of a type of attacker, analogously to the strength of a security or privacy countermeasure. It specifies how capable an attacker is in breaking countermeasures. An attacker can break a concrete security countermeasure when its capability is greater than the strength of the countermeasure ($TC > S$).

The **contact frequency** CF is used to specify how frequently a type of attacker comes into contact with the system (either with or without malicious intent). This enables the distinction between potential attackers that have very frequent (or likely) contact with the system (e.g., an insider, a customer, or a script kiddie) and attackers with less frequent contact with the system. The contact frequency is specified as a number of contacts per year.

[1] Loss because of secondary risk (e.g., how the customers of an organization using the system would react on a loss event) is considered out of scope in our analysis. However, it is possible to add these losses to the primary loss magnitude if they need to be taken into consideration as well.

The **probability of action** PoA specifies how likely an attacker is to attempt an attack (successful or not) once the attacker comes into contact with the system. It is expressed as a probability ($[0, 1]$).

The **vulnerability** V is derived from the countermeasures and the attacker type. It specifies whether attackers succeed in attacking the system by breaking the countermeasure(s). The vulnerability is calculated as the percentage of successful attacks, estimated by sampling from both the *strength* and *threat capability* distributions.

The **threat event frequency** TEF is used to specify the frequency of *attempted* attacks on the system. It is derived from the *contact frequency* of the attacker and the *probability of actions* of that attacker. Note that the threat event frequency does not specify the frequency of successful attacks; this is derived later by incorporating the vulnerability.

The **loss event frequency** LEF is the frequency of successful attacks that actually result in a loss. It is derived from the threat event frequency (i.e., the number of attempted attacks) and the vulnerability (i.e., the percentage of successful attacks).

The **loss magnitude** LM specifies the loss or damage of a successful attack. It is based on the *value* of the threatened DFD element for the considered type of threat. It is most naturally expressed as a currency (e.g., in EUR), but it is also possible to use relative numbers (e.g., the relative importance of the elements in the model) to compare the relative risks for different elements.

Finally, the **risk** R for each elicited threat is calculated by multiplying the loss magnitude (the value attached to the threatened DFD element) with the loss event frequency (the frequency of successful attack, given the countermeasures applied in the context of the threat and the attacker profile). This provides an annualized risk (expected loss per year). Complementary to the annualized risk, the loss magnitude of a single attack (loss event) should also be taken into consideration for threats with a very high loss but a very low frequency. Additional measures such as insurance may be required to mitigate these risks.

4.3 Practical considerations

From the components in Figure 3, the leaves of the shown tree (strength S, threat capability TC, contact frequency CF, probability of action PoA and loss magnitude LM) are estimates that need to be provided by the stakeholders. The loss magnitude is provided once per DFD element, the strength once per countermeasure, and the others once per attacker model. Therefore, many of these values are reusable across models. Furthermore, by pre-defining a few common value estimates (e.g., 'no value', 'low value' 'average value', 'high value', 'critical value'), the burden of entering all value estimates can be further reduced.

The calculation of the risk for a single threat happens via Monte Carlo simulations. In particular, every input distribution is sampled 2000 times, and these samples are combined according to Figure 3 to yield a risk distribution for that particular threat. This process is subsequently repeated for each elicited threat.

5 EVALUATION

For the evaluation the risk-based threat modeling approach, we apply the risk-based threat elicitation of the TMaRA prototype on the SecureDrop whistleblower submission system [8]. After

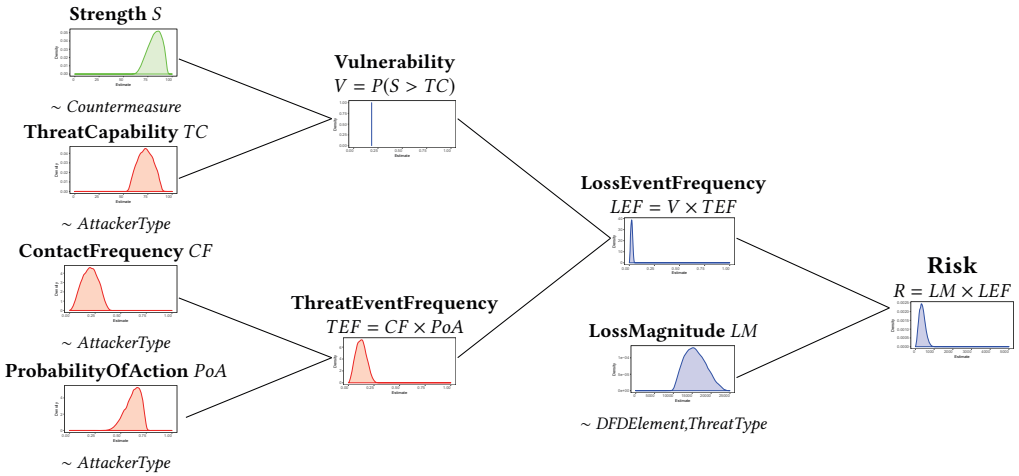

Figure 3: Overview of the FAIR risk components and their combinations
Risk components from FAIR [9] to be used in the context of a single threat, attacker type, and model element with a value. These are used to estimate the risk using Monte Carlo simulations. The nodes show example estimate distributions (graphs) and the threat modeling artifacts that influence these distributions (~). The lines illustrate which components are combined.

that, the performance of the risk analysis in the proof of concept implementation is discussed.

5.1 SecureDrop

We evaluate to what degree the risk-based prioritization is effective for identifying the most important threats by verifying whether high-risk threats correspond with the security decisions and assumptions in the SecureDrop whistleblower submission system [8], a real-world application which allows whistleblowers to anonymously contact journalists and submit documents to them.

The SecureDrop application was chosen because: (i) it is an open-source system, (ii) it has a publicly available DFD model, (iii) it has detailed documentation on the security assumptions and the attacker capabilities that are considered, and (iv) it has stringent security and privacy requirements.

The following inputs are used in the evaluation: (i) the threat modeling document with the SecureDrop DFD; (ii) the SecureDrop documentation (source, journalist, admin, and developer guides); (iii) the SecureDrop source code; and (iv) the documentation of other components explicitly referred to by the SecureDrop documentation or source code (e.g., the recommended hardware firewall for its functionality, the libraries used for authentication).

SecureDrop DFD Model The DFD model of SecureDrop, displayed in Figure 4, consists of 81 elements (8 trust boundaries, 6 external entities, 17 processes, 7 data stores, and 43 data flows). As part of the process explained below, the model is furthermore enriched with 36 security solutions and assumptions.[2]

Methodology

A) Assignment of value estimates To minimize bias in the value assignments, a static scheme is used to consistently determine the value of an element, based on the importance of protecting the identities and data of the subjects involved. The following valuation scheme is used: **+3** if data of a source (whistleblower) is involved (*Estimate(min:2, probable:3, max:4, confidence:4)*); **+2** if data of an admin is involved (*Estimate(min:1, probable:2, max:3, confidence:4)*); **+1** if data of a journalist is involved (*Estimate(min:0, probable:1, max:2, confidence:4)*). Each estimate has the same deviation of $+/-1$ and a confidence of 4. Where data of multiple entities is processed, the estimates are combined (e.g., source + admin: *Estimate(min:3, probable:5, max:7, confidence:4)*), leading to 7 different valuations.

B) Adding Security Solutions The SecureDrop DFD model is enriched with security solutions for all the countermeasures implemented in SecureDrop, as well as any explicitly documented security assumptions in the threat modeling documentation provided by the project [8]. All solutions encountered in the threat model, documentation, and source code are encoded as patterns which protect certain DFD elements against certain types of threats. A generic countermeasure strength is used across the solutions, as only the presence of a countermeasure is needed for the evaluation, not the actual strength of the countermeasure.

C) Risk Analysis & Collected Data To obtain the risk analysis results, we ran our automated risk analysis on the modeled SecureDrop DFD. The result is a list of threats following the STRIDE mnemonic. To elicit threats, STRIDE by interaction is applied as described by Shostack [17]. For each threat in the list, the potential risk (i.e., assuming no countermeasures are present) is calculated, as well as whether the threat is mitigated by some countermeasure(s). Note that, for this evaluation, we thus take a binary interpretation of threat's mitigation status, i.e., a threat is either mitigated or not, regardless of how effective a countermeasure actually is.[3]

[2]Note that these are not necessarily distinct solutions. A firewall, for example, is instantiated for each pair of communicating entities.

[3]While our approach does support calculating to what degree implementing a specific countermeasure actually reduces the risk, these outcomes cannot be evaluated based solely on the available SecureDrop information.

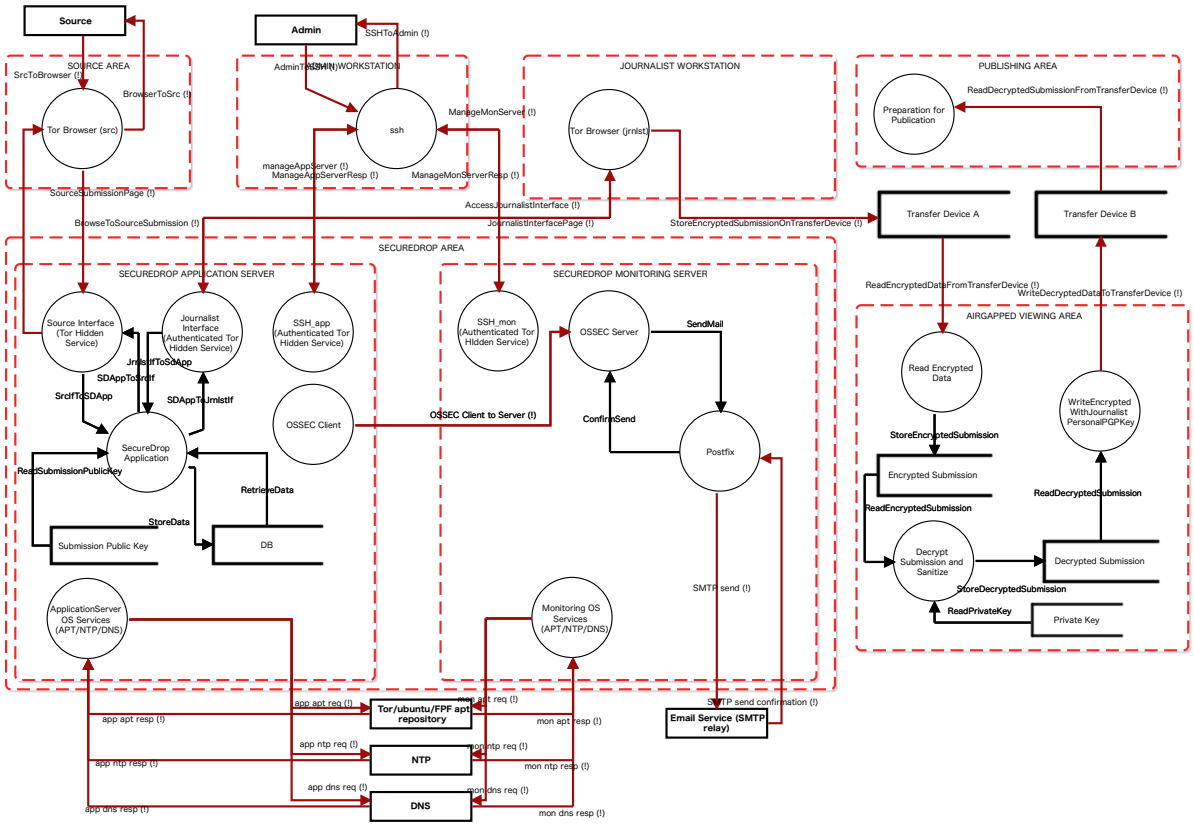

Figure 4: SecureDrop DFD model

DFD model of the SecureDrop application. The model is largely based on the SecureDrop threat modeling document. Since the threat modeling document refers to an earlier version of SecureDrop, any inconsistencies between the threat model and the other SecureDrop documentation (i.e., source/journalist/admin guide, installation guide, development guide) are resolved by referring to the latest version of the other SecureDrop documentation and the source code (if necessary).

D) Relating Mitigation Status and Potential Risk After conducting the threat elicitation, the effectiveness of the risk analysis results for threat prioritization is evaluated as follows. The presence of a countermeasure indicates that the threats it protects against are important enough to spend the effort in implementing the countermeasure, or writing out the detailed security assumptions; so, we consider the presence of a countermeasure as an indication of the importance of a threat. Therefore, to evaluate the effectiveness of the risk estimates for prioritizing threats, we evaluate to what degree threats with a high-risk estimate from our approach are actually prevented in SecureDrop itself. Focused threat mitigation efforts would then result in a high coverage of high-risk threats and (likely) a lower coverage of low-risk threats. The opposite would indicate either that using risk analysis information is a sub-optimal way for prioritizing threats, or that the SecureDrop project misdirected their security efforts.

Results The results of running the threat elicitation and risk analysis are shown in Figure 5. The figure shows the density of mitigated (top curve) and unmitigated (bottom curve) threats according to their potential risk. The line is a smoothed version of the density difference between both, to highlight trends. The plot shows that high-risk threats (towards the right of the figure) have often been explicitly mitigated in SecureDrop by means of security countermeasures or assumptions, more often than the lower-risk threats (towards the left of the figure).

If we assume that SecureDrop is serious about its security (which can be corroborated by the explicit inclusion of a threat model in its documentation, as well as the multiple deployments at large, well-known newspapers, for example), our results—even with the simplistic estimates that we have used for performing this evaluation—give strong indications that *the risk-based priorities generated by our approach have been able to pinpoint the most important threats.*

5.2 Functional validation

The evaluate the feasibility of the approach, we implemented a proof of concept of the risk analysis approach in a threat modeling tool prototype TMaRA. We have conducted an initial performance evaluation by running a threat and risk analysis on the SecureDrop DFD Model, shown in Figure 4. The model contains 81 DFD elements and 36 solution instances and results in 262 threats. The

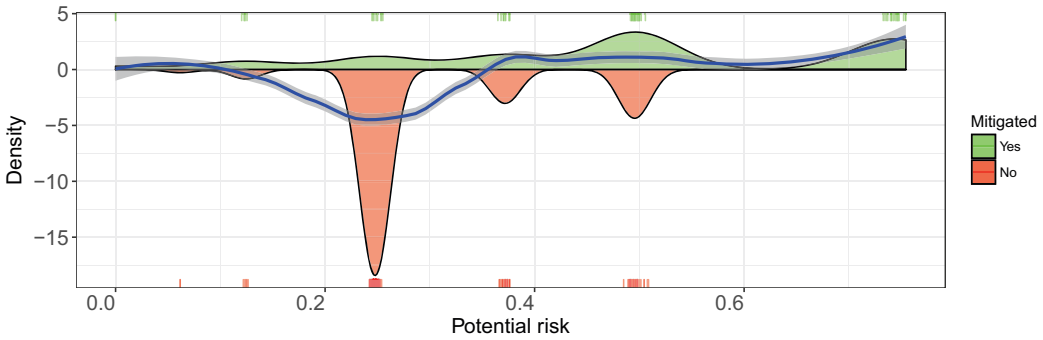

Figure 5: Distribution of the elicited threats according to their potential risk
Density plot of the distribution of the elicited threat according to the potential risk they pose. The plots are grouped by the presence or absence of security countermeasures in SecureDrop that mitigate them. The distributions show a high mitigation-rate for the high-risk threats, indicating the estimated high-risk threats are considered important enough to counter them, and a lower mitigation-rate for low-risk threats, indicating these are considered to be less important.

total analysis time (averaged over 10 runs) is 5.214 s, which includes loading the model, the query engine, pattern matching (for eliciting threats), risk analysis, and presenting the results to the user. Performing 100 runs for the just the threat elicitation and risk analysis (with 2000 samples per distribution) results in a mean value of 1521 ms (95% CI: 1446 ms–1595 ms). The evaluations were run on a 2017 2.6GHz Intel Core i5 with 8GB of RAM without any performance optimizations in the implementation.

Additionally, in earlier work, we have also performed a qualitative evaluation of the DFD security enrichments, by comparing it with security property-based solutions such as the Microsoft Threat Modeling Tool 2016 [13], showing positive improvements in terms of semantic quality, traceability, separation of concerns, and dynamism [18].

5.3 Threats to Validity

While the evaluation is applied in the context of a single application, the risk analysis approach (FAIR) is a pre-existing approach which is already used in other contexts outside of threat modeling. The current application shows that, given reasonable value assignments, it can be applied in the context of threat modeling as well and it, in such case, leads to realistic prioritization of security threats.

The evaluation with SecureDrop relies on the assumption of the SecureDrop developers focusing their security efforts. However, this is not an unreasonable assumption given it is the only open source project the authors have encountered that has such detailed threat modeling documentation available. Additionally, since the threat mitigation checks rely on the presence of actual security countermeasures, they provide strong evidence that such threats have actually been considered.

6 DISCUSSION AND FUTURE WORK

In this section we discuss open issues and planned future extensions to our risk-based threat elicitation approach.

Risk component units The assignment of security countermeasure strength and the attacker capability values happen on a relative scale. Such a relative scale may make it difficult when a lot values are already assigned to make sure that newly assigned values still make sense relatively compared to previously assigned values. A reusable absolute scale with clearly defined units may improve this situation, but there currently is no such scale to measure the 'security'-strength of solutions, nor the capabilities of attackers.

The impact of this problem can be reduced by providing an extensive set of security solutions and attacker models where these numbers are already properly set and are verified against each other that the differences in relative values are sound.

Effort trade-off between adding information and prioritizing threats

Processing large lists of automatically elicited threats to find the most relevant ones requires considerable effort. To reduce this effort in our approach, we introduced a risk analysis activity and subsequently use the estimated risk for the threat prioritization. However, this risk analysis step requires additional information to be present in the model (e.g., valuations of the model elements). The threat modeler does have to provide this additional information, thus introducing a trade-off exercise between the effort in enriching the model for the risk analysis and the effort in manually conducting the threat prioritization afterwards. Although the enrichment of the model can also be considered as the explicit documentation and consistent use of information that would otherwise still be used implicitly in the manual prioritization.

In future work, we intend to look at additional extensions to assist the threat modeler in enriching the model with this information with reduction in effort from the threat modeler.

Difficulty in determining the risk component estimates

An issue closely related to the effort in adding the estimates above, is the difficulty in assessing the values themselves to assign to the model elements. Model value estimates can be assigned in a business-driven fashion, in which case the data is already available. However, in cases where such data is not available, other approaches could be used to determine these values. Examples of this are heuristics which could attempt to make estimates based on the model types of the elements involved or a using a pre-defined set of estimates to assign to all elements, as in the evaluation.

Catalog of security solutions To improve the usability, an extensive set of existing security solutions should be available to enable the threat modeler to enrich the model with all existing security solutions. An extensive catalog prevents the effort of manually defining security solutions and the strength of the countermeasures they contain.

Taking insiders into account The current approach for determining an elements vulnerability with the mechanism strength and attacker capability is not well suited for analyses on insider attacks. While it is possible to make the insider more capable to express its privileged position in a system, this capability would no longer be in accordance with the strengths of the other mechanisms applied internally. Additionally, such internal mechanisms could by hard for an insider to break, while they would be (relatively) easy for outside attacker that has already been able to break the much stronger mechanisms protecting the system form these outsiders. To better model these types of attacker support should be added for varying frequencies of contact and probabilities of action.

7 RELATED WORK

Türpe [22] discussed how security needs arise from the interactions of three dimensions: design, goals, and threats, and observed how many efforts focus only on a single dimension. We structure this section according to the threat–design interactions and the threat–goal interactions, as threat modeling and risk analysis focus on the interactions between these dimensions.

Threat modeling activities belong to the threat–design interactions. Threat modeling was introduced by Microsoft a part of its security development lifecycle [10, 11, 16, 19, 20] and has proven popular since, with multiple real-world applications in industry [6, 16, 20] and readily-available tool support from Microsoft [13]. In these existing approaches and applications, data flow diagrams remain largely security and privacy agnostic models, with only minor, and often ad-hoc, additions for security or privacy. However, recently there have been several proposals for extensions to these data flow diagrams in order to have a more systematic representation of security- and privacy-relevant information in order to elicit the most relevant security and/or privacy threats [2, 4, 18, 21].

Risk analysis, on the other hand, focuses the threat–goal interactions. Risk analysis approaches elicit security requirements starting from security goals, anti-goals, such as in CORAS [12], or attack trees [15], and can be used in a complementary fashion to threat modeling [14]. Instead of conducting such analyses in isolation, we presented an integrated approach that includes the risk information in the DFD model used for threat modeling, thereby integrating both activities and enabling them to reinforce each others results.

8 CONCLUSION

Existing threat modeling approaches are based on the design of the system, but lack support for the prioritization of the elicited threats grounded in concrete data on the system, its security and privacy solutions, and relevant attacker models. Conversely, existing risk analysis approaches are applied in a disconnect from the concrete system design, and the threats such a design elicits.

Our risk-enhanced threat modeling approach resolves this disconnect by enriching the involved threat modeling elements with relevant risk analysis information. This enhanced model enables the execution of per-threat simulations to calculate a risk estimate based on the concrete model of the system under design.

By enriching the resulting threat list with estimated risk, threats can be triaged according to their risk, and security and privacy efforts can be focused on the most important threats first. Additionally, progress in reducing the risk to the software system can be measured across time to enable managing and monitoring the evolution of risk in the system under design.

In the future, we intend to explore advanced extensions such as dynamically updating security catalogs over time, to consider changes in the effectiveness of existing security solutions, culminating in a continuous threat analysis and risk assessment approach.

ACKNOWLEDGMENTS

This research is partially funded by the Research Fund KU Leuven.

REFERENCES

[1] Majed Alshammari and Andrew Simpson. 2016. Towards a Principled Approach for Engineering Privacy by Design. (2016).
[2] Thibaud Antignac, Riccardo Scandariato, and Gerardo Schneider. 2016. *A Privacy-Aware Conceptual Model for Handling Personal Data.* Springer, 942–957.
[3] Aaron Bedra. 2017. Adaptive Threat Modeling, GOTO Conference Chicago. (2017). https://www.youtube.com/watch?v=YTtO_TGV2fU
[4] Bernhard J. Berger, Karsten Sohr, and Rainer Koschke. 2016. Automatically extracting threats from extended data flow diagrams. *Lecture Notes in Computer Science* 9639 (2016), 56–71.
[5] Mina Deng, Kim Wuyts, Riccardo Scandariato, Bart Preneel, and Wouter Joosen. 2011. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering* 16, 1 (2011), 3–32.
[6] D Dhillon. 2011. Developer-Driven Threat Modeling: Lessons Learned in the Trenches. *IEEE Security Privacy* 9, 4 (jul 2011), 41–47.
[7] European Union. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016. OJ L 119, 04.05.2016, p. 1–88. *Official Journal of the European Union* 59, L 119 (may 2016), 1–88.
[8] Freedom of the Press Foundation. 2018. SecureDrop | The open-source whistleblower submission system. (2018). https://securedrop.org/
[9] Jack Freund and Jack Jones. 2014. *Measuring and managing information risk: a FAIR approach.* Butterworth-Heinemann.
[10] Shawn Hernan, Scott Lambert, Tomasz Ostwald, and Adam Shostack. 2006. Threat Modeling: Uncover Security Design Flaws Using The STRIDE Approach. *MSDN Magazine* 6 (nov 2006).
[11] Michael Howard and Steve Lipner. 2006. *The Security Development Lifecycle.* Microsoft Press.
[12] Mass Soldal Lund, Bjørnar Solhaug, and Ketil Stølen. 2010. *Model-driven risk analysis: the CORAS approach.* Springer Science & Business Media.
[13] Microsoft Corporation. 2016. Microsoft Threat Modeling Tool 2016. http://aka.ms/tmt2016. (2016).
[14] Tobias Rauter, Nermin Kajtazovic, and Christian Kreiner. 2016. Asset-Centric Security Risk Assessment of Software Components. *2nd International Workshop on MILS: Architecture and Assurance for Secure Systems* (2016).
[15] Bruce Schneier. 1999. Attack trees. (1999).
[16] Adam Shostack. 2008. Experiences threat modeling at microsoft. In *Modeling Security Workshop. Dept. of Computing, Lancaster University, UK.*
[17] Adam Shostack. 2014. *Threat Modeling: Designing for Security.* John Wiley & Sons, Indianapolis, Indiana. 590 pages.
[18] Laurens Sion, Koen Yskout, Dimitri Van Landuyt, and Wouter Joosen. 2018. Solution-aware Data Flow Diagrams for Security Threat Modelling. In *Proceedings of SAC 2018: The 6th track on Software Architecture: Theory, Technology, and Applications (SA-TTA).*
[19] Frank Swiderski and Window Snyder. 2004. *Threat modeling.* Microsoft Press.
[20] Peter Torr. 2005. Demystifying the threat modeling process. *IEEE Security & Privacy Magazine* 3 (2005), 66–70. https://doi.org/10.1109/MSP.2005.119
[21] Katja Tuma, Riccardo Scandariato, Mathias Widman, and Christian Sandberg. 2017. Towards security threats that matter. In *3rd Workshop On The Security Of Industrial Control Systems & Of Cyber-Physical Systems (CyberICPS 2017).*
[22] Sven Türpe. 2017. The Trouble With Security Requirements. *25th IEEE International Requirements Engineering Conference* (2017).
[23] David Vose. 2008. *Risk analysis: a quantitative guide.* John Wiley & Sons.
[24] Kim Wuyts. 2015. *Privacy Threats in Software Architectures.* Ph.D. Dissertation. KU Leuven.