Risk Centric Threat Modeling - A Misuse Case based approach

Malik Nadeem Anwar
Department of Computer Science,
Jamia Millia Islamia, New Delhi, India.

Mohammed Nazir
Department of Computer Science,
Jamia Millia Islamia, New Delhi, India.

Abstract
In fact, security is an inevitable and common concern nowadays, ensuring it early in SDLC may significantly reduce risk, time and effort. Moreover, it enhances reliability and quality of the applications. Consequently, it establishes the confidence of user with the product. Misuse Case, a classical and effective modeling technique, which has been widely used for eliciting and modeling security threats at the requirements stage. However, this technique lacks the ability to model ‘assets, vulnerabilities and risk’, which are important risk-related concepts. In order to incorporate such risk related concepts, we propose an idea of extending misuse case model in line with the ISSSRM model. Thereby, we propose an extended version of misuse case model, which incorporates ‘assets, vulnerabilities and risk-spots’. Further, a modeling process has also been suggested for helping the designers to model security related risks in an effective manner during the requirements phase, itself. Furthermore, the model is validated using a case study on an e-voting system.

Keywords: Misuse case model, Software security, ISSRM model, Security requirements, Requirements Engineering, Risk modeling, Threat modeling

I. INTRODUCTION
In today’s scenario, software security is a challenging and an inevitable aspect. It has been widely acknowledged that security issues should be addressed at the initial stages of software development [1] [2]. In fact, software defects cost reduces significantly when they are addressed early at the requirements phase [3]. Moreover, early considerations of security facilitate developers to model anticipated threats, their vulnerabilities and countermeasures well in advance, rather than reacting to possible attacks after the system is developed [4]. Quick fixes to attacks in the already developed software, conflicts with other functionalities and requirements thereby, giving rise to a new set of issues in security domain.

Several modeling languages are available in the literature that caters security aspects at the early stages of SDLC [5] [6] [7] [8] [9]. Misuse case model has proved to be an effective and widely used technique among them for elicitation of security requirements at the initial stages of development process. It can model harmful usage scenarios, not desirable in the system. Because of UML like syntax, even a non-security practitioner can easily use them for eliciting threats and security requirements [10]. On the other hand it has some limitations; it is a general technique with no specifications or precise guidelines available and relies completely on human intuition, imagination and experience [11] [12]. The Misuse case diagrams lack certain constructs that are essentially needed to model security risk.

It is an attempt to devise a mechanism to manage the security risk early at the requirements stage. Consequently, an extension of misuse case model is proposed that includes ‘assets, vulnerabilities and risk-spots’ as new modeling elements to the existing model. Efforts have been made by researchers for improving the current misuse case model, with a better understanding and support for security risk. The syntax and process for risk analysis is aligned with the ISSRM domain model, a well-established standard for security risk management that covers all pertinent fundamental issues and concepts related to software security.

The paper is organized as follows. Section 2 discusses the methodology adopted. Section 3 provides a brief description of the related work. Section 4 discusses the gaps identified after analysing the related work. Section 5 describes the proposed model (abbreviated as MUCX), while Section 6 illustrates the process of applying the MUCX model. The model is validated in Section 7. A brief discussion on paper contributions and related findings has been described in Section 8. Section 9 discusses the threats to validity while Section 10 provides conclusion and future directions of the work.

II. METHODOLOGY
The paper explores the possibility of extending misuse case model in line with ISSRM model, with an aim to model security risk at the requirements phase. In order to achieve this objective, we adopted a methodology that is demonstrated in Figure 3. First, we examined the current state of literature related to ISSRM and Misuse case model along with its minor extensions. Next, we studied the proposals related to alignment of misuse case with ISSRM. All these steps are depicted as ‘Revisiting related work’ in Fig. 1.

After studying these proposals, we compared the misuse case model and its extensions proposed in the literature with the core concepts of the ISSRM model to understand the limitations of the current version of the model.

After understanding the limitations of the misuse case model in its inability to model risk, the next step was to suggest possible improvements to the misuse case model in the form of meta-model, graphical syntax, modeling process and this situation is shown in Figure 1 as ‘Proposed idea’.
III. RELATED WORK

A. ISSRM (Information System Security Risk Management)

ISSRM is a focused and systematic approach, which tackles the security-risk related issues of an information system (IS) [13] [14], compliant with several security and risk standards [15][16][17][18]. The model offers risk management at three different conceptual levels of assets, risk and risk treatment. It supports the alignment of other security modeling languages with the core risk management concepts. ISSRM model is depicted in Fig. 2. The three categories which describe the core concepts of ISSRM model are:

1) Asset-Related Concepts: Anything that has value to the organization, necessary for achieving its objectives is defined as an asset. It can be further classified as business asset and IS asset. Constraints on business asset are expressed by security criterion in the form of integrity, confidentiality and availability.

2) Risk-related Concepts: Risk is defined as potential harm to business. It is composed of a threat with one or more vulnerabilities that if executed successfully, harms the asset known as impact. An event is a combination of threats along with vulnerability, where vulnerability is the weakness in a system that can be exploited by threat agent. A threat is also a way to inflict an attack. Threat Agent is an attacker that initiates a threat in order to harm the IS asset. Attack Method is the manner and means through which a threat agent executes a threat.

3) Risk treatment-related Concepts: The decision to treat the identified risk is known as Risk treatment. Refinement of this decision in the form of requirements is defined as security requirements. Control is nothing but the countermeasure that implements the security requirement.

The model is implemented by a six-step iterative process [13] described below:

1. Identify the assets and define the context of organization.
2. Determine security objectives for the identified assets.
3. Analyze the risks to the identified assets.
4. Once the risks are identified, various risk treatment decisions like risk (avoidance/reduction/transfer/retention) are taken.
5. In order to mitigate the risk, security requirements of the IS are determined.

Finally security controls (counter-measure) are determined as possible implementation of the security requirements. Since the process is iterative, it may identify new risk and security controls after each iteration.

B. Misuse case model

It is an extension of the UML Use case model, used as a tool for modeling threats and security requirements at the requirements stage, often used in conjunction with the corresponding use case model [19] [11]. A misuse case diagram depicts both misuse case (threats) and misuser (attacker) along with an association relationship directed from the attacker to the misuse case.

Sindre and Opdahl define misuse case as a sequence of successful steps by an attacker that can harm the system. They consider misuser as a type of actor that has ill-intention towards the system. Fig. 3. shows a visual description of
misuse case model along with various components like use case, misuse case and security use case. Security use case represents the countermeasure needed to mitigate the threats. The textual template of misuse case model provides a detailed description of the steps needed for initiating the threat by an attacker [19].

C. Extensions to Misuse case model

Several researchers have worked on the misuse case model and their minor modifications/extensions, most notable among them are: John McDermott and Chris Fox [20] have extended the UML based use cases to incorporate elements like threats and countermeasures. They referred to this new model as Abuse case. Ian Alexander used misuse case as a tool for elicitation of non-functional requirements [21]. He further applied the model for requirements/design trade-offs in an industrial case-study and managing conflicts in goal driven requirements engineering [22] [23]. The modeling elements used by him are almost similar to the misuse case model, but includes extra relationships aggravates and conflicts. Donald G. Firesmith [24] proposed a new modeling element security use case as an extension to the classical misuse case model. It captures the mitigation or counter-measure to the threats. Rostad, L. [25] have proposed an extension in the form of vulnerabilities and insider threats, as two new modeling elements in the misuse case model and validated the model on a case study on healthcare system. Okubo [26] have enhanced the misuse case model by introducing two new elements assets and security goals. They have also introduced a lightweight process for applying the model for elicitation of security requirements.

---

**Table I. Analyzing misuse case and its extension**

<table>
<thead>
<tr>
<th>The ISSRM model</th>
<th>Misuse Case Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset</td>
<td>Actors and use case</td>
</tr>
<tr>
<td>IS asset</td>
<td>x</td>
</tr>
<tr>
<td>Business asset</td>
<td>Use case</td>
</tr>
<tr>
<td>Security criteria</td>
<td>x x x x x x x</td>
</tr>
<tr>
<td>Risk</td>
<td>x x x x x</td>
</tr>
<tr>
<td>Impact</td>
<td>x x x x x x</td>
</tr>
<tr>
<td>Event</td>
<td>x x x x x</td>
</tr>
<tr>
<td>Threat</td>
<td>Misuse case Misuse case Misuse case Misuse case Misuse case</td>
</tr>
<tr>
<td>Vulnerability</td>
<td>x x x</td>
</tr>
<tr>
<td>Threat agent</td>
<td>Misuser Misuser Misuser Attacker/Insider Attacker Attacker Misuser</td>
</tr>
<tr>
<td>Attack method</td>
<td>x x x x x</td>
</tr>
<tr>
<td>Risk-treatment</td>
<td>x x x x x</td>
</tr>
<tr>
<td>Security requirement</td>
<td>Use-case Counter-measure Security use case Security use case Counter-measure Security requirements</td>
</tr>
<tr>
<td>Control</td>
<td>x x x x x</td>
</tr>
</tbody>
</table>

**Extensions of the Misuse Case Model**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS asset</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business asset</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security criteria</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risk</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impact</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Threat</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vulnerability</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Threat agent</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attack method</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risk-treatment</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security requirement</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
IV. GAP ANALYSIS

As already discussed several extensions of misuse case model are available in the literature. Many researchers have also aligned the misuse case model with ISSRM [4] [13] [27]. A comparative analysis of various extensions to the misuse case model in line with ISSRM model is presented in Table 1. The misuse case model and its extensions lack risk related constructs that are required for modeling threats and risk. For e.g. the extension proposed by Rostad, L [25] includes vulnerabilities and insider threats as modeling elements, while Okubo. T et al. [26] consider assets and security goal as new elements.

None of the extensions consider risk-spots as its modeling constructs. As obvious from Table 1, no work is reportedly done till now that address all the constructs ‘assets, vulnerabilities and risk’ at the same time in a model. Modeling the threats along with vulnerabilities and the assets can be significant to perceive and visualize an attack. For addressing risk related issues, these concepts need to be defined using graphical constructs and represented in the form of a model.

In this paper, we have proposed a new idea of extending misuse case model (MUCX), incorporating all the three elements namely assets, vulnerabilities and risk in our model, that may significantly help the user in modeling threats. Consequently, risk-spots can be outlined, and mitigation mechanism may be applied accordingly. Following section describes our proposed model in detail.

V. THE PROPOSED MODEL

This section describes ‘MUCX’, which is an extended version of misuse case model. Rules and constructs of the model are formulated and defined in the form of a meta-model, followed by brief description of the model, linking rules and syntax.

A. Meta-model of MUCX

Meta-model is a model that expresses the logical or syntactical structures of another model with the help of handful of classes and association to lay down rules for defining the model [28]. A meta-model is a “model of a model” [29]. It gives a view of the model at a higher level of abstraction. It has gained wide acceptance for defining syntax of any modeling language, with the extensive use of UML in software development [30]. A meta-model of MUCX was created, (see Fig. 4.) to lay down the rules and syntax of the language. Different classes used in the meta-model are use case, misuse case, security use case, actor, misuser, assets and vulnerabilities.

![Fig. 4. Meta model of MUCX](image-url)
Table II. Relationship tags for modeling elements in MUCX

<table>
<thead>
<tr>
<th>Relationship Tag</th>
<th>Precise description on Linking rules</th>
</tr>
</thead>
</table>
| includes         | - Connects multiple instances of different entities like actors/misusers/use case/misuse case vulnerabilities among themselves, and captures parent-child relationship.  
                   - Use case and the security use case may also be connected by includes relationship.  
                   - Directed arrow indicates the parent entity from the child. |
| extends          | - Models the optional behavior extending from one entity to another.  
                   - Instances of use case or misuse case can be connected among themselves using the extends relationship.  
                   - Directed arrow indicates the base entity from the extended one. |
| Association      | - Misuser/User should be linked with misuse case/use case using this relationship.  
                   - Multiple misusers /users can associate with one misuse case/use case.  
                   - Multiple use/misuse cases can associate with a single or multiple misuse/user. |
| threatens        | - It defines a link directed from a threatening misuse case to the threatened use case using directed arrow.  
                   - Multiple use case entities may be threatened by a single misuse case.  
                   - Multiple misuse case may target a single use case. |
| mitigate         | - It defines a link directed from a security use case to the misuse case.  
                   - Directed arrow from a security use case to the misuse case.  
                   - A single use case may mitigate multiple misuse cases. |
| exploit          | - It defines a link directed from a security use case to the misuse case.  
                   - Denoted by a directed arrow from misuse case to the vulnerability.  
                   - A single misuse case may exploit one or more vulnerabilities. |
| harms            | - This relationship connects an instance of misuse case with assets.  
                   - It is denoted by a directed arrow from misuse case to the asset.  
                   - Multiple misuse case may harm one or more assets. |
| imperils         | - It defines a link directed from a security use case to the misuse case.  
                   - This relationship is denoted by a directed arrow from vulnerability to the asset.  
                   - Multiple vulnerabilities may be connected to zero or more assets |

**B. Brief description of MUCX model**

The classical misuse case model only includes misuse case, use case, misusers and actors. For getting a complete and comprehensive view of an attack, it is essential to add some extra constructs to the model in the form of assets, vulnerabilities and risk spots. In the context of our model, the assets, vulnerabilities, risk and their extensions are defined in accordance to the ISSRM model [13] [14]

- **Assets**: Anything valuable to the organization needed for achieving its objectives.
- **Vulnerabilities**: Flaws in the system that can be compromised for realization of threat.
- **Risk**: It is the combination of threat with one or more vulnerabilities, thereby causing negative impact on the assets

After defining the essential constructs needed for modeling risk, we hereby describe the extensions carried in our model with respect to these constructs.

A misuse case (threat) is linked to one or more assets via harms relationship, indicating the negative impact of threat on an asset. In this way, we can capture a clear scenario of ‘what’ should be protected (assets) against whom. Next extension in the model is to incorporate vulnerabilities. They are highlighted by a grey ellipse with exploits relationship between misuse case and the vulnerabilities. Vulnerabilities are driving force behind the realization of threats. Thus, it is essential to include vulnerabilities in our model to get a clear picture of the weaknesses in the system. The third element risk is not represented in our model as a separate entity; rather assets, vulnerabilities and threats are together depicted by red colored danger symbol indicating the risk-spots. Such spots represent loopholes or attack surfaces in a system.

The graphical constructs used in MUCX are listed in Appendix A. An abstract model of MUCX is shown in Fig. 5.

VI. MODELING PROCESS FOR MUCX

This section describes the modeling process (Fig. 6), clearly illustrating the steps for capturing security risk at the requirements phase using the proposed model. The process is iterative, incremental in nature and is designed in line with the ISSRM model. The Modeling process mainly consists of three phases. Each phase has many sub-steps that need to be completed in order to achieve their individual purpose. The first phase is sequential while second and third phases are iterative. Each of these phases has been discussed in detail in the subsequent section.

1. **Identifying users, use case and assets**: The intention at this stage is to investigate the purpose of the system, its users, functional requirements (use-case) and assets. Actors initiate a use-case. The actors can be human (users), components (hardware, software) or sub-systems. Each of the actors initiates a key role or activity referred to as use case. Data flow diagrams can be used to identify and outline system boundary. In order to achieve the task
of identifying actors, use cases and assets, various methods like interviews with domain experts and stakeholders, analyzing SRS document, brainstorming, storyboards, scenarios etc. can be adopted.

2. **Identifying risk-spots:** It involves careful investigation of a use case and explores the possibilities that can be exploited for malicious purpose. Here, the focus is to identify the ways through which a malicious user can harm a system along with vulnerabilities associated with the threats. We list out the steps for identifying malicious users, their abnormal actions and the vulnerabilities associated with them to model risk-spots in a system. The stepwise actions (S\textsubscript{1} – S\textsubscript{5}) are as follows:

**S\textsubscript{1}** - Identifying misusers,

**S\textsubscript{2}** - Identifying threats using STRIDE methodology,

**S\textsubscript{3}** - Identifying vulnerabilities responsible for each threat by exploring NVD [31]

**S\textsubscript{4}** - Examining each vulnerability, as it may give rise to new threats,

**S\textsubscript{5}** - Identifying possible relationships among threats, vulnerabilities and assets using rules listed in Table 2 for outlining risk-spots.

3. **Identifying security requirements:** The aim in this phase is to safeguard use-cases by providing countermeasure in the form of security use case. It can be achieved by carefully examining the risk spots in a system and suggesting measures to counter them. Each countermeasure may also have vulnerabilities, and they are required to be analyzed further for possible weaknesses in phase 2. The stepwise actions in this phase are steps (S\textsubscript{1} – S\textsubscript{3}) as follows:

**S\textsubscript{1}** - Identifying possible countermeasure in the form of security use case,

**S\textsubscript{2}** - Examine security use case for weaknesses,

**S\textsubscript{3}** - Link the security use-case to misuse case.

### VII. E-VOTING SYSTEM – A CASE STUDY

The proposed model ‘MUCX’ is validated using a case study on an E-Voting system. This system was first used for conducting municipal and regional elections of 2011 in Norway, followed by the parliamentary elections in 2013. The main feature of this project is its openness as all the documents related to the project like SRS documents and architectural designs are publicly available. The case study is based on the System specification document [32] describing in detail the requirements of the system provided by the Ministry of Security and Service Organization (DSS), Norway. The overall functional model of e-voting system is divided into three phases:

- **Phase I:** Preparation.
- **Phase II:** Voting.
- **Phase III:** Counting and auditing

The preparation phase supports all the preparations that are needed before the voting starts. It supports various activities like configuration of the system, managing users and electoral roll, approval of candidates and voters and other preparations related to counting and auditing the system. The voting phase consists of two phases that are e-voting (remote voting) and p-voting (in person voting). In this case study our main focus is on e-voting, as it is the most vulnerable activity in the system. The e-voting phase supports different activities like register voters, mark voters in the electoral roll, secure login cast e-votes and store e-votes. The counting and auditing phase deals with various activities like counting of e-votes, declaring results and creating logs for auditing. We have successfully applied our model on all three phases of e-voting system. Risk spots are depicted by a triplet \( R(T, V, A) \) where \( T, V, \text{ and } A \) denotes threat, vulnerability and asset respectively while \( C \) represents countermeasure.

![Fig. 5. Abstract model of MUCX](image-url)
Appendix B provides the detail of all the threats, vulnerabilities, assets, risk-spots and the countermeasure uncovered during the case study.

The expression ‘C mitigates T’ exhibit the fact that a countermeasure is available to resolve a threat. In the Preparation phase, the counter measures \{C₁, C₂, C₃, C₄\} can mitigate threats \{T₁, T₂, T₃, T₄\}. In the voting phase, \{C₁, C₂, C₃, C₄\} can mitigate \{T₁, T₂, T₃, T₄\} while in counting phase, counter measures \{C₁, C₂, C₃, C₄, C₅\} can mitigate threats \{T₁, T₂, T₃, T₄, T₅\}. The diagrams depicted in Fig. 7, Fig. 8 and Fig. 9 are evolved when MUCX modelling process is applied on the e-voting system. The process significantly facilitates for outlining risk-spots in the system. Section 8 discusses the effectiveness of MUCX modeling process and consequent results.

VIII. DISCUSSION

In this paper, we have critically analyzed the limitations of misuse case model; primarily its inability to model security risk. The misuse case model has been extended by aligning it with various risk-related concepts defined in ISSRM model in terms of syntax and semantics both. The syntax of proposed model was laid down by defining various modeling elements, along with linking rules in the form of a meta-model. Researcher have incorporated ‘assets, vulnerabilities and risk-spots’ as new modeling elements along with relationship tags like ‘harms, imperils and exploits’.

With the help of a modeling process (Fig. 6.), MUCX identifies threats, vulnerabilities, assets and outlines the possible risk-spots. These concepts are visually represented, and the model provides a clear picture of threats and risk associated with them. MUCX can be applied at the initial phase of system development (i.e. requirements phase), thereby, modeling threats and risks in a proactive manner. Further, it provides a comprehensive framework for modeling threats and performing risk analysis. Moreover, it also gives an insight for organizing threats and associated risks. The viability of model is validated using a case-study on an E-voting system. Consequently, 15 risk-spots, 12 vulnerable assets, 15 threats and 12 potential vulnerabilities are identified in the case study.

Risk analysis is a data intensive business decision that is based on knowledge about threats, vulnerabilities, assets, their impacts and probabilities [33]. The classical formula for risk is stated as:

\[
Risk = \text{probability of threat} \times \text{consequence}
\]

The current misuse case diagram only provides an overview of threats and the attacker. However, this information is insufficient to model risk as it lacks two components namely ‘possibility of threat’ and the ‘consequences of attack’. The paper addresses both these issues after extending the misuse case model by adding new elements ‘vulnerabilities, asset and risk-spot’ to effectively model the misuse case-risk scenario. Threats and vulnerabilities capture the likelihood of threats occurrence, while the harm caused to assets and their severity determines the possible consequences. In the context of our model, risk is represented as ‘risk-spots’, that can be defined mathematically as a function of Threat (T), Vulnerability (V) and Asset (A) in Eq. 2

\[
\text{Risk-spot} = f(T, V, A)
\]

Risk-spots may significantly facilitate system designers in analyzing threats and the risk associated with them. The process adapted for modeling the risk is iterative in nature. It may identify new threats and vulnerabilities at each of the iterations thereby, assuring better threat coverage at the early stage and providing secure application. The risk-spots may conspicuously assist designer for suggesting better countermeasures to the threats.

Risk analysis is an established technique in almost all engineering disciplines. With the help of risk analysis, system analyst and requirements engineer can easily anticipate and assess the gravity of threat at the requirements phase, itself. Many available approaches have proposed extensions to misuse case model, but these are only minor extensions in terms of assets, vulnerabilities and security goals. No work is yet reported in the literature that extends misuse case model, incorporating all the core concepts of risk. MUCX extends misuse case model including risk and represents it as risk-spots. Table 1 reveals that 4 out of the 7 risk-related concepts (defined in the ISSRM model) are supported by MUCX, while other misuse case-based extensions support only 2 to 3 concepts.
Fig. 6. Modeling process for MUCX
Fig. 7. MUCX for preparation phase
Fig. 8. MUCX for voting phase
Fig. 9. MUCX for Counting and Auditing Phase
**IX THREATS TO VALIDITY**

Following the guidelines in [34] we adopted the following scheme for ‘threats to validity’ in the context of our case study

*Internal validity:* The viability of MUCX is validated using a case study on an e-voting system. During the course of the work we uncovered 15 threats, 12 vulnerabilities and 12 vulnerable assets. Thereby we outline 15 risk-spots in the e-voting system. Threat analysis depends on STRIDE methodology, which in turn is based on intuition, judgment and brainstorming of researchers. Hence, it may affect results of the study. Further, vulnerabilities responsible for the corresponding threats are extracted from the NVD database that is dynamically updated on regular intervals. Therefore, the coverage and quality of extracted information may also have an influence on the results. Furthermore, examining the vulnerabilities, identification of assets and risk spots obviously depends on the skills and imagination of researcher/designer that may be a source of biasness.

*External validity:* It deals with the extent up to which our results can be generalized. Since the model has been applied on a single case study, thus it imposes an external validity threat. However, the case study is based on a system, implemented in a real time scenario and effectively implemented for conducting regional and municipal elections and in this sense the system can be considered as robust enough. In order to further assure the applicability of our model, it must be validated on a broader range of applications including safety critical applications and business intensive systems.

*Reliability:* In fact, the study has been carried out only by the authors of this paper and this may introduce some level of threat to reliability. However, since researchers are both the performers of the case study and the designers of model, having strong understanding about threat modeling and risk analysis and have expertise in the area. Therefore, the study conducted, and results obtained may be considered as reliable. Nevertheless, we acknowledge the fact that more subject experts or stakeholders should be involved in the process of assessment. In future, the model will be validated empirically and its assessment will be performed on range of applications using multiple participants.

**X. CONCLUSION AND FUTURE WORK**

Security is an inevitable and challenging aspect. Modeling threats and risk at the requirements phase significantly reduces time and efforts of the developer. In order to effectively model threats and risk at the requirements stage we have proposed an extended version of UML misuse case model in line with the ISSRM model. The new model MUCX visualizes all the possible threats and their effects, along with possible vulnerabilities, and assets affected. The risk-spots in MUCX diagram capture the combined effect of threats and vulnerabilities on the assets. The viability of the model is assessed by applying it on an e-voting system. The case study successfully identifies several potential threats, vulnerabilities and risk-spots, thereby producing a complete MUCX diagram for each phase in e-voting system. MUCX comparison with other similar proposals shows that proposed model quite effectively incorporates concepts like vulnerabilities and risk which are not considered by most of the researchers in their model.

In future, we plan to develop a tool in order to assist the developer at the requirements phase itself, while modeling risk. In order to assure the accuracy of the process and its effectiveness in business domain and safety critical applications, we intent to formalize MUCX model using formal languages. Further, we intent to validate the model empirically on a broader range of users and subject experts.

**REFERENCES**


[9] Fabian, B., Gürses, S., Heisel, M., Santos, T., &


## APPENDIX A

**Graphical constructs used in MUCX**

<table>
<thead>
<tr>
<th>The ISSRM model</th>
<th>Proposed model (MUCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Constructs</strong></td>
<td><strong>Concrete syntax</strong></td>
</tr>
<tr>
<td><strong>Asset</strong></td>
<td><strong>Use case</strong></td>
</tr>
</tbody>
</table>
| **Business asset** | 1) Stakeholders  
|                  | 2) Business use case  |
| **IS asset**    | **Assets**            |
| **Security criteria** | -                   |
| **Risk**        | **Misuse Case**       |
| **Impact**      | **-**                 |
| **Event**       | **-**                 |
| **Threat**      | Misuse Case/threat    |
| **Vulnerability** | Weakness in the system |
| **Threat agent** | Misuser/attacker      |
| **Attack method** | Sequence of actions by misusers |
| **Risk-treatment related** | Security use case |
| **Control**     | **-**                 |
## APPENDIX B

### Applying MUCX modeling process on an e-voting system – Summarized list of MUCX modeling elements and step-wise application on an e-voting system

<table>
<thead>
<tr>
<th>MUCX modeling process (activities) applied on E-Voting System</th>
<th>E-voting system (Phase-wise Narratives)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Preparation</strong></td>
<td><strong>Voting</strong></td>
</tr>
<tr>
<td>Understanding system boundary</td>
<td>Functional model along with the DFD was analyzed</td>
</tr>
<tr>
<td>Identification of actors</td>
<td>System administrator, voter, candidates, electoral committee, security expert and political party</td>
</tr>
<tr>
<td>Identification of use cases</td>
<td>Configure the e-voting system, management of users, assign roles to users, register for electoral roll, fill form for candidate, authorize new voters, maintain electoral rolls, approve candidates for electoral roll, security control.</td>
</tr>
<tr>
<td>Identification of misusers</td>
<td>External attacker as well as insider (staff managing the system)</td>
</tr>
<tr>
<td>Identification of misuse cases (threats)</td>
<td>$T_1$: Inject malicious codes, $T_2$: DOS attack, $T_3$: Spoofing, $T_4$: Tampering</td>
</tr>
<tr>
<td>Identification of risk spots</td>
<td>Risk spots $\rightarrow R_1, R_2, R_3, R_4$</td>
</tr>
<tr>
<td>Linking countermeasure to threats:</td>
<td>$C_1$ mitigates $T_1$, $C_2$ &amp; $C_3$ mitigates $T_2$, $C_3$ mitigates $T_3$, $C_4$ mitigates $T_4$, $C_5$ mitigates $T_5, C_6$ mitigates $T_6$</td>
</tr>
</tbody>
</table>