Universität Zürich[UZH]

Jan von der Assen
Muriel Figueredo Franco
Christian Killer
Eder John Scheid
Burkhard Stiller

On Collaborative Threat Modeling

IfI Technical Report — No. 2022.04

April 2022

University of Zürich UZH
Department of Informatics IfI
Binzmühlestrasse 14, CH—8050 Zürich, Switzerland

ifi

On Collaborative Threat Modeling

Jan von der Assen, Muriel F. Franco, Christian Killer, Eder J. Scheid, Burkhard Stiller
Communication Systems Group CSG, Department of Informatics IfI, University of Zürich UZH
Binzmühlestrasse 14, CH—8050 Zürich, Switzerland
E-mail: [vonderassen, franco, scheid, killer, stiller]@ifi.uzh.ch

***Abstract*—Threat Modeling is a structured process to identify critical assets in an organization and the threats posed by adversarial agents. The goal of applying such a process is to achieve a shared understanding of the inherent risks and potential countermeasures that can be put in place. Threat Modeling is, by nature, a collaborative process. However, this paper, shows that related work mainly focuses on adapting models to technical aspects of architectural decisions. Thus, non-technical stakeholders are not included in the process. Furthermore, these applications are not fit to address current trends affecting cybersecurity, such as the growing number of cyberattacks or the impacts of remote work.**

Thus, this paper proposes *CoReTM* an approach to apply existing threat modeling methodologies in a collaborative setting. The resulting approach allows organizations to extend threat modeling to non-technical stakeholders in an automated way, while supporting on-site, remote or hybrid operations synchronously or asynchronously.

***Index Terms*—Cybersecurity, Threat Modeling, Security Management, Risk Management, Information Sharing**

I. Introduction

Threat modeling is a well-known method to design systems, networks, and businesses with security in mind. After identifying potential threats, appropriate countermeasures can be considered, communicated, and implemented early in the design process or in production environments [21]. Such a method relies on abstractions to find security risks, and involves the creation of multiple models. The first model depicts abstractions of assets such as processes, users, software components or data sources that need to be protected. In contrast, the second model requires a view of these assets that comprises attack targets and vectors. Such a model can be created by taking on an adversary-centric view on the input domain [38].

Threat modeling is, by nature, a highly practical approach towards securely designing systems [3]. In this sense, collaborative approaches are important, especially with the growing number of information systems, employees, and critical assets to be protected [9]. Furthermore, the increased frequencies at which cyber-attacks are launched [1], [8] impose frequent changes in the threat landscape of today's systems. This emphasizes that threat models must be able to evolve at a fast pace. Thus, organizations must not only collaborate to create threat models, but also frequently perform post-*mortem* analysis of such models to understand if the corresponding countermeasures are still appropriate [16]. Hence, threat modeling tools should adapt to the work style to enable collaboration and be flexible to enable threat modeling among the relevant stakeholders and their skills.

Whiteboards are one of the most popular tools for threat modeling, because they provide more flexibility and visibility compared to other tools (*e.g.,* Office Suites, MTMT or diagrams.net). However, for certain settings, whiteboards limit collaboration because they require physical presence of the participants. Remediation using online setups (*e.g.,* video conferences or calls) with which distributed teams can follow the modeling process [38] is not ideal as the remote participants are not able to interactively contribute to the model. In the same way, relying exclusively on physical meetings may lead to delayed iterations of the threat modeling process. For that reason, a remote or hybrid meeting style allows the participation of geographically distant members. Thus, an asynchronous implementation would allow members to collaborate without any dependency of time and location.

Being able to collaborate outside physical meetings is critical as not only the threat landscape but also the ways of working have changed in today's workforce. 51% of knowledge workers and 76% of software engineers were projected to work in a form of remote workspace at the end of 2021 [18] [11]. Although it is unclear whether this trend will persist, remote threat modeling may become the norm rather than the exception. Based on these facts, it is critical to design collaborative threat modeling tools with remote participants as first-class citizens. In addition, depending on the asset that is being modeled, it is vital that a diverse set of stakeholders are included in the process. The ability to bring a broad set of perspectives and capabilities into the process motivates why collaborative approaches are required to securely design systems, processes or projects. For example, threat modeling solely based on existing source code would exclude business representatives, similarly as an offline information system excludes remote participants. The need to have this collaborative ability is further magnified when considering that evolved threats require federated protection services spanning multiple organizations [9]. Finally, currently available tools are not considered to sufficiently support modeling among collaborators in modern processes such as agile software engineering [3].

In this paper, *CoReTM* is proposed to enable threat modeling in a collaborative setting, including scenarios where employees work in a fully distributed setting. *CoReTM* is specifically created to enable modeling in an on-site, re-

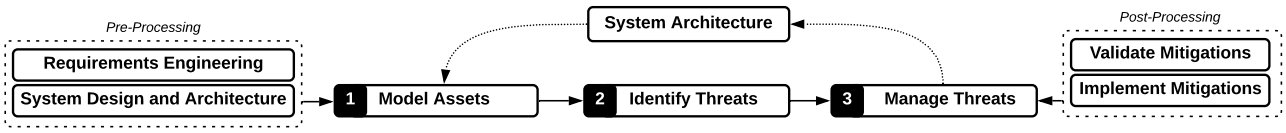

Fig. 1: Procedural Elements Related to Threat Modeling Considered in This Work Here

mote or hybrid setting while allowing both asynchronous and synchronous contribution. *CoReTM* provides *(i)* a real-time annotation-based collaborative editor, *(ii)* automated threat report generation and *(iii)* DevOps integration, supporting threat modeling in all combinations of meeting styles. Furthermore, *CoReTM* enables collaboration between a wide range of stakeholders with diverse backgrounds and skills by implementing a methodology selector, abstract visual annotation libraries and methodology knowledge bases. Thus, *CoReTM* is the first platform that allows the integration of multiple methodologies into a flexible platform so that a wide range of collaborators find an extensive set of threats.

The remainder of this paper is presented as follows. Section II introduces fundamental methodologies and provides an extensive survey of existing modeling applications. Identified limitations of related work are then contrasted with a design that is proposed in Section III. Next, the design of *CoReTM* is evaluated in Section IV. Finally, Section V concludes this work.

II. Background and Related Work

At the core of each threat modeling tool is a methodology that defines the model semantics and modeling procedures. [21] provide a survey over such methodologies. However, a plethora of software implementing these methodologies has emerged since. Further, it is unclear to which degree these applications enable and foster collaborative settings. This papers survey of related tools mainly focuses on three critical tasks of threat modeling as shown in Figure 1.

A. Threat Modeling Methodologies

Most threat modeling methodologies focus on the threat discovery phase. *STRIDE* aims to do so by providing a high-level categorization of threat families [21] [28]. Since no specific procedures or a repertoire of prevalent threats are provided, derivatives, and combinations of this methodology are common [38] [21]. Other methodologies focus on finding relevant threats based on an enumeration of concrete threats. *CAPEC* provides a publicly accessible inventory of common attack patterns described from an adversarial perspective [6]. To provide operational context from an adversarial perspective, a second methodology called the Adversarial Tactics, Techniques & Common Knowledge (ATT&CK) is often cross-referenced in these descriptions [7]. Finally, some threat enumeration libraries focus on specific technologies. While such technologies like OWASP can help organizations to increase the awareness for specific threats [32], they are often considered complementary to other threat modeling methodologies [38]. Some merely offer a representation of modeled threats to provide a way to reason about threats [38]. With *Attack Trees*, high-level threats can be decomposed into a hierarchy of related threats [21].

Methodologies that go beyond the discovery of threats exhibit higher complexity. *PASTA* is an examplary methodology that begins with the definition of business objectives and finally provides a risk analysis [43]. *NIST SP 800-154* follows a similar, yet more technology-oriented approach to such a risk assessment [29].

B. Threat Modeling Tools

There are numerous dimensions by which threat modeling tools can be categorized. This survey considers the collaborative features of 18 tools and categorizes them into one of five identified approaches, as presented in Table I. **General Purpose-Tools** are widely used applications that do not follow a specific methodology. Nevertheless, their flexibility allow threat modeling to be carried out. *Whiteboards* are popular for rapid modeling, although they do not support asynchronous and remote settingss [38]. Similarly, office suites provide a flexible way to share threat models asynchronously. Finally, *diagrams.net* is an online diagramming tool for which threat modeling addons are available [19]. Differential synchronization allows multiple remote users to work on the same diagram synchronously [23]. However, the interval at which models are synchronized make this tool prone to errors. Furthermore, none these platforms provide any guidance to untrained users.

Microsoft Threat Modeling Tool [27], *OWASP Threat Dragon* [30], *TRIKE* [4] [38], *SeaMonster* [33] and *CAIRIS* [5] [12] are all **manual modeling applications** that contain a diagram drawing component. However, they all follow a threat modeling methodology and are thus coupled to this methodology. Since many of these tools are web-based applications, it is possible to use them in asynchronous and remote settings to some extent. However, according to our knowledge and related surveys [3], it is not feasible to create models synchronously. Furthermore, being tightly coupled to a specific methodology makes them only applicable to collaborative settings where said methodology is also applicable.

In recent years, two novel threat modeling trends have emerged. **Automated Threat Modeling** approaches use existing artifacts such as source code or architecture diagrams to automate the discovery of threats and their countermeasures. Related applications such as *IriusRisk* [22], *Tutamantic* [42] [20], *securiCAD* [14], *MAL* [15] do not enable mod-

TABLE I: Comparison of Toolkit Support for Collaboration

	Async & On-site	Async & Remote	Sync & On-site	Sync & Remote
General-Purpose Tools (3)				
Whiteboards	✗	✗	✓	✗
Office Suites	✗	✓	✓	✗
diagrams.net	✗	✓	✓	✓
Manual Threat Modeling (5)				
MTMT, TRIKE, OWASP Threat Dragon, SeaMonster, CAIRIS	✗	✓	✓	✗
Automated Threat Modeling (4)				
IriusRisk, securiCAD	✗	✓	✓	✗
Tutamantic	✗	✓	✗	✗
MAL	✗	✗	✗	✗
Integrated Threat Modeling (3)				
ThreatSpec, Threagile, raindance	✗	✓	✗	✗
Hybrid Modeling Approaches (4)				
pytm, SDElements	✗	✓	✗	✗
ThreatModeler	◐	✓	◐	◐
CoReTM	✓	✓	✓	✓

✓ = provides property, ✗ = does not fully provide property, ◐ = support unclear

eling in synchronous settings. Furthermore, the tight coupling to the underlying methodology and relying on existing input or an editor which does not support multiple users at the same time may render certain collaborative use cases such as workshops impossible. However, the presence of automated threat discovery implies that some form of knowledge base exists in the system. Thus, inexperienced users can still discover threats without having to be security experts.

Similar features can be discovered when looking at the second modeling approach which is being followed by tools such as *ThreatSpec* [41], *Threagile* [35] [35] [34], and *Raindance* [10], which aim to **integrate** and link threat models to the software development life cycle by relying on sources such as markdown files and source code.

Finally, *pytm* [39], *ThreatModeler* [40] [38], *SDElements* [36] [37] combine the previously described approaches into **hybrid approaches**. Therefore these tools compare similarly in terms of support for collaborative settings.

III. The *CoReTM* Approach

CoReTM is a tool-supported approach to enable collaborative threat modeling among diverse stakeholders in real time and asynchronously. *CoReTM* features a flexible modeling editor, methodology selection procedures, action-driven process guidance, knowledge-bases, and integration to existing systems. By introducing an overarching meta-modeling process, all of these features are detached from specific methodologies. Under this process, various existing methodologies, including high-level methodologies such as STRIDE and Attack Trees, technology-oriented methodologies (*e.g.,* OWASP and CAPEC) and domain-driven risk-centric methodologies (*e.g.,* OCTAVE and PASTA), are supported. For that, a variety of input formats can be annotated in a compatible way.

Thereby, *CoReTM* promotes collaboration under circumstances where participants are not able to collaborate at the same time or location. Knowledge or skill gaps related to cybersecurity are bridged by enabling process guidance and knowledge bases. Finally, the consideration of unstructured input data as well as the integration with other systems ensures threat models are not isolated, so that collaboration is prolonged into other processes of the development cycle.

A. Architecture

The architecture of *CoReTM* is depicted in Figure 2 and described as follows. Different scenarios are applicable to model threats, including combinations of on-site, remote, synchronous, and asynchronous meeting styles. All users access *CoReTM* through a web-based interface which provides all user-facing functions. Initially, the user is carried through the modeling process defined here. There, he/she can configure the threat modeling scenario. The automated walkthrough application guides this user through the setup procedure, where a meeting style is defined, and appropriate methodologies are selected. For example, an administrating user creates two separate workshop spaces. One for project managers, focusing on modeling threats of a process model using STRIDE. At the same time, a second space will be used to model web application threats of an existing architecture component by relying on OWASP.

Once participants access the web-based interface during the actual workshop or meeting, the application guides them through the defined methodology. Thereby, the underlying methodology is explained and followed. Depending on the methodology, additional knowledge bases are available to search for specific threats. In any case, the modeling process starts with a definition of assets. Many organizations already hold models or related representations of their assets. *CoReTM* solves the issue of heterogeneous input sources by implementing an application that renders unstructured data such as PDF documents and pictures. In the same way, users can create new models using an integrated, minimalist version of the popular *diagrams.net* editor [24]. Both of these modeling applications use annotations as a means to define the semantics so that it is possible to automate report generation and integration even in the presence of multiple input data sources.

Once participants have modeled the assets, the walkthrough application guides the threat discovery phase based on the selected methodology. Discovered threats are stored using annotations so that threats elicited on structured and unstructured data sources can be processed into reports. Thus, the *CoReTM* provides *(i)* graphical elements, *(ii)* process descriptions, and if applicable to the methodology, *(iii)* knowledge bases to discover threats.

To manage these elicited threats, all threats annotated on respective documents are automatically compiled into one report, where users can manually add threat descriptions and potential countermeasures. With that, *CoReTM* allows teams to leverage heterogeneous data sources, methodologies, and users to create one threat modeling report. Finally, users can

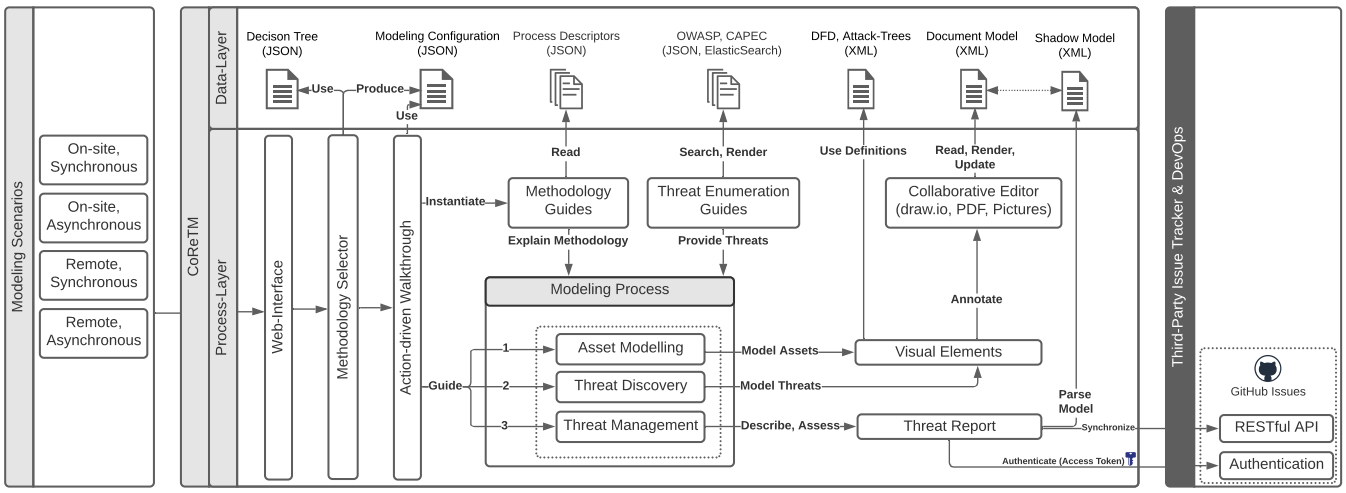

Fig. 2: Overview of *CoReTM* Components and Interfaces

link elicited threats to GitHub issues using the GitHub API. Thus, Threat model reports can be automatically updated based on DevOps pipelines on GitHub.

B. Flexible Modeling

As *CoReTM* enables modeling for stakeholders of different meeting styles and backgrounds, the core of *CoReTM* is defined to be extensible. This includes a collaborative, real-time editor that allows annotation-based threat modeling with a variety of methodologies and sources. This is possible due to the incorporation of multiple flexible key components.

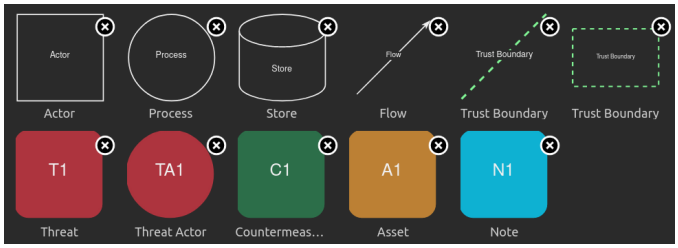

Fig. 3: Data-flow Diagram Library Used in the Diagram Editor

The real-time editing capabilities are provided by a flexible editor that supports modeling with diagrams and with existing, unstructured artifacts. The XML-based editor allows new diagrams to be created or imported. This diagram-based approach ensures that diagrams can be created for all the supported methodologies and for specific domains. *CoReTM* provides a set of abstract visual elements used across multiple methodologies. These elements contain additional meta-data, so that diagrams are machine-readable. Custom libraries to enable Data-Flow-Diagram (DFD) and attack-tree modeling are provided, since with these abstractions, threats can be annotated on newly created or imported diagrams. To annotate threats, two mechanisms are present. First, a specific color palette is prepared. Thus, in existing diagrams, users can color-code elements without having to replace parts of the diagram. Secondly, a set of visual elements can be used to create new diagrams. Figure 3 outlines visual elements for modeling threats using a DFD and Listing 1 shows the internal representation of a *coretm-threat* element which highlights the annotation meta-data.

```
<mxGraphModel>
  <root>
    <mxCell id="0" />
      <mxCell id="1" parent="0" />
      <object coretm-type="threat" id="2">
        <mxCell vertex="1" parent="1">
          <mxGeometry width="80" height="80"
                      as="geometry" />
        </mxCell>
    </object>
  </root>
</mxGraphModel>
```

Listing 1: XML Representation of “Visual Element” Annotating Threats

With the aforementioned editor and their related visual libraries, users can apply different methodologies to threat model. To support all meeting styles, the underlying representation of the model needs to be carefully synchronized. Thus, to enable asynchronous modeling, a shared work space with centralized access to the model is sufficient. In addition, participants can annotate models using notes similarly as one may leave a sticky note on a whiteboard. Modeling in synchronous manner presents greater challenges due to the possibility of synchronization conflicts. Since reliable transport is ensured through the web-based interface, a reliable storage component is used to mirror changed models to other participants. Unlike solutions such as *diagrams.net*, no regular intervals are used to exchange models. Furthermore, differential synchronization algorithms are not applicable since they depend on fuzzy text-based algorithms that are not optimal for structured content such as the stored models [17]. Instead, models are exchanged

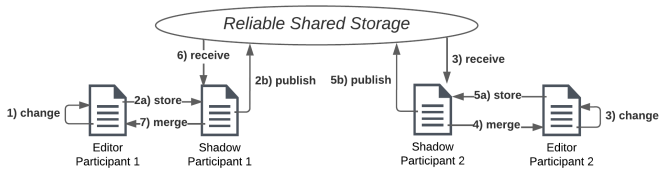

Fig. 4: Synchronization Procedures to Exchange and Merge Models between Remote Participants

whenever a substantial change is made. Each participants' editor holds the document in memory and in a copy of a shared storage component synchronized across all members. If the model changes, all participants merge the changed document from the storage component into the one they hold in memory. If there were no intermediate changes, the model in memory no longer corresponds to the one in the storage. Thus, no subsequent message is sent. However, if there were intermediate changes, the merged document is stored in the centralized storage component, which will cause remote participants to merge documents. This algorithm, which can be implemented using a reliable publish- and subscribe based storage service is shown in Figure 4.

C. Methodology Agnosticism and Model Heterogeneity

Section II introduced numerous tools that are closely related to the implemented threat discovery methodology. With that, some of these tools are able to provide powerful features, such as code-based modeling, automated threat discovery or DevOps integration. However, depending on such input data can also pose a high entry barrier. For example, creating a threat model of an architecture may not be possible with these tools if the software will be developed from scratch. Similarly, relying on code as a medium for modeling may exclude certain stakeholders such as program managers, architects, testers or requirements engineers. When it comes to threat discovery, attack enumeration-based methodologies (*e.g., OWASP* or *CAPEC*) are already focused on specific technologies and are therefore only applicable to certain scenarios.

CoReTM addresses this problem with two key functionalities. First, *CoReTM* provides an abstract modeling process to which existing methodologies can be mapped. This process is implemented in an application that actively helps users choose, combine, and set up the right methodology. Based on a survey of existing methodologies, the methodology is chosen according to applicability and requirements derived from an interactive questionnaire, as shown in Figure 5. Since methodologies can be complementary, it is desired that in certain cases multiple methodologies are selected. For example, for users who do not have a definition of assets the definition may be driven by the *PASTA* methodology, but for the discovery of threats, *STRIDE* can be applied. In any case, the respective methodology can then be applied using the previously described editors, which allow a wide variety of input formats. With that, threats gathered with any methodology are automatically compiled into the report and can be managed from there.

Secondly, a set of knowledge bases are implemented into *CoReTM*. Specifically, procedural knowledge bases help users to apply and navigate the asset and threat identification steps defined by the respective methodology. Furthermore, searchable threat enumeration databases allow for a directly accessible list of specific threats. Integrating multiple methodologies into one application component ensures that users do not have to search through various web pages in order to find threats in the appropriate abstraction.

With the combination of these two approaches, non-security experts, such as software engineers, testers, or project managers, can learn the respective methodology. This process proposed ensures the rotation and combination of complementary methodology so that users may discover additional threats.

D. Automation and Integration

Many diagram-based modeling applications do not provide machine-readable documents that preserve semantic aspects related to threat modeling. Thus, reporting is often seen as a tedious task with little value added to proceeding activities [38]. *CoReTM* implements an automation step to create simplified reports which can be linked to other processes. By linking the generated threat model to other processes, the model provides direct value. Thus, whenever the model is updated, the procedure shown in Listing 2 is automatically enacted.

```
model := URLDecode(ungzip(b64(diagram)))
threats := model.filter(type=threat OR color=#AD343E)
                .map(threat => threat.text)

assets := model.filter(type=asset OR color=#BC8034)
                .map(threat => threat.text)
```

Listing 2: Reading Threats and Assets from an Annotated Diagram

First, the encoded models are parsed to machine-readable XML. Using XML attributes and color-codes, the semantics of the elements can be extracted. Specifically, a simple visualization displays threats in tabular fashion, where additional information can be flexibly entered by users.

Once such a threat report is created, it can be further used for collaboration. First, in asynchronous settings, users can communicate over elements of the report using notes. Next, threats can be converted to a *GitHub issue* with a single click, which causes the platform to issue a request against the RESTful API provided by GitHub.

By following these approaches, a threat model can be used in other collaborative settings, such as the software development process. This also ensures that a document does not go stale. For example, once countermeasures are implemented and the respective issues are cleared in the version control system, the threat model within *CoReTM* mirrors this state. Thus, once a threat model is updated, it will automatically show the current state. This linkability is not provided when threat

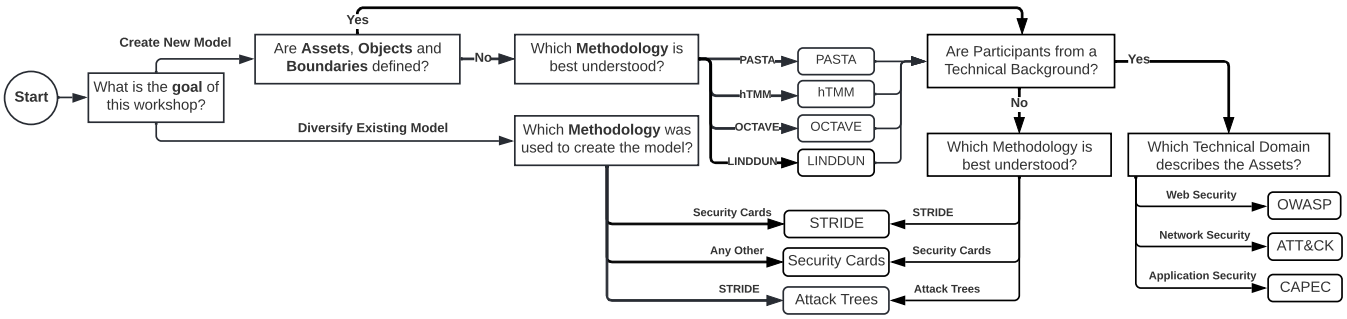

Fig. 5: Methodology Selection Graph

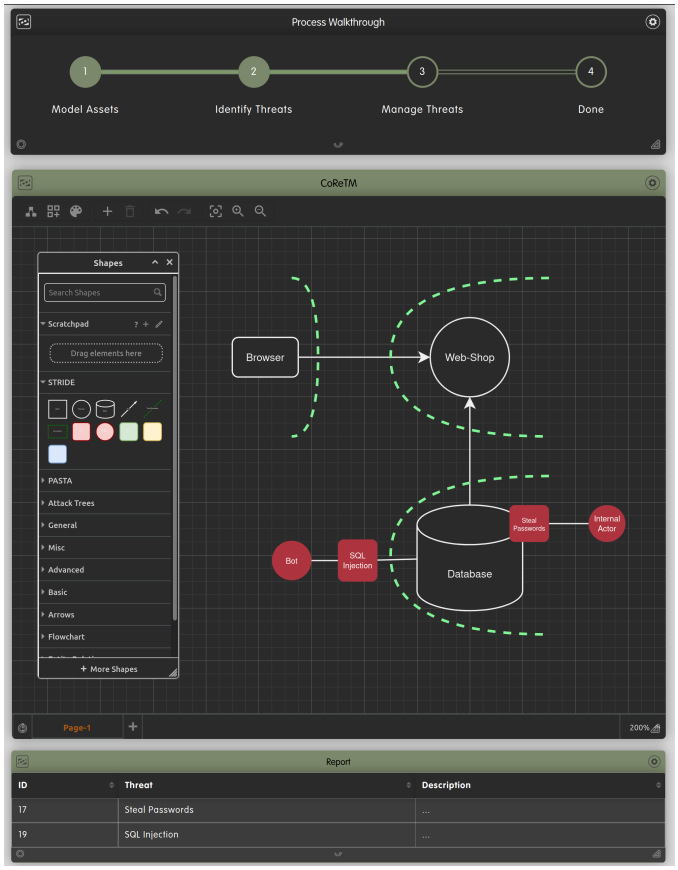

Fig. 6: Editor, Report, and Walkthrough Components of the *CoReTM* User Interface

models are created in third-party diagram editors, whiteboards or spreadsheets.

E. Prototype Implementation

To demonstrate the feasibility of the *CoReTM*, a prototype was developed. As the implementation of the editor component and its storage are clearly path-breaking, these elements are the core components. Due to the popularity of the *diagrams.net* editor among software engineers, this editor was considered. Since *diagrams.net* has a complex implementation and the already integrated storage providers (*e.g.,* Google Drive and Microsoft OneDrive) cannot be considered open platforms for further development, a custom storage integration was developed. To ensure collaborative editing, *diagrams.net* is run in an embedded mode, using *iframe* elements to render the user interface (UI). Although the editor does not encourage extension [23], it exposes a clear *JSON* protocol in this mode [25]. With that, it was possible to leverage the editing capabilities on a different platform, as shown in Figure 6.

To provide a digital whiteboard and shared storage, the orchestration platform *dizmo* was targeted, since it provides programmatic orchestration for microfrontends. Furthermore, the web-based platform can be set up on the user's premises to alleviate data privacy concerns. An XML-based custom library of visual elements to allow annotation was developed for the *diagrams.net* editor. All elements in this library contain metadata so that it is possible to create a machine-readable model of the diagram without relying on specific interfaces of the editor. Specifically, another microfrontend consumes the annotated model using a publish-and-subscribe data exchange and represents modeled threats in a table. Here, the user can freely add additional information or synchronize the threat with a GitHub issue.

All of the previously described elements cover general demands of the approach, thus are not related to any technology, standard or process aside from the high-level modeling process shown in Figure 1. In contrast, the guided brainstorming component that provides additional inspiration for threat discovery was implemented to reflect the *STRIDE* methodology. Additionally, the threat enumeration microfrontend is implemented with regard to the OWASP library. However, other libraries, including bespoke ones that an enterprise may hold, can be integrated by adding additional microfrontends.

In total, the prototype that implements *CoReTM* offers seven microfrontends, which can be characterized as follows.

The **Questionnaire** guides the user through the methodology selection process. Once the appropriate methodology is found, this static microfrontend orchestrates the configuration of the editor and brings up the methodology helpers.

An **Editor** allows the user to create new asset models and threats. Alternatively, it can be used to annotate existing documents. Custom libraries ensure that the output of applying

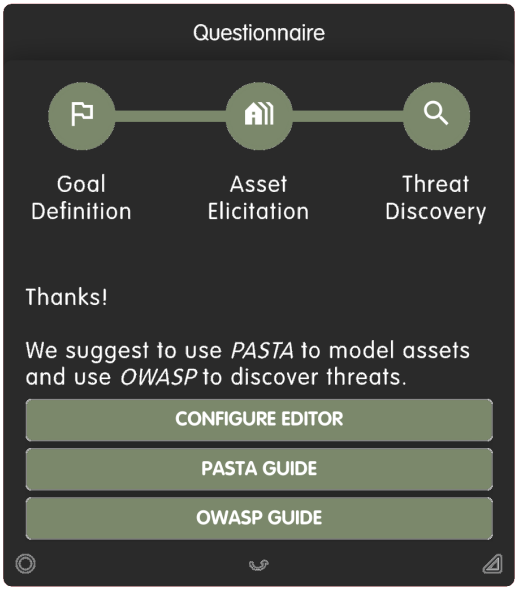

Fig. 7: Methodology Selection Result by the Questionnaire

a methodology can be processed by other microfrontends.
The **Walkthrough** provides the user with a high-level status overview of the current modeling process considering the selected methodology.
A **Report** is automatically populated using the annotated threat model and can be integrated into the DevOps pipeline by allowing synchronization with the RESTful GitHub API.
STRIDE is an interactive microfrontend outlining the threat discovery using the *STRIDE* methodology.
Security Cards is an interactive set of microfrontends representing the threat discovery game *Security Cards*.
OWASP is a static microfrontend that allows direct access to threat enumeration provided by *OWASP*.

All the previously introduced UI components are implemented using a microfrontend architecture. This choice is decisive for the prototype, since microfrontends show two clearly beneficial features to this use case. First, with such an approach, additional components can be added without increasing the complexity of the existing ones. For example, to add another threat discovery methodology, one could implement a new frontend without the need to modify or even understand the existing ones. Secondly, microfrontends can be run as dedicated applications. Thus, one can open the same component multiple times on the same workspace. For example, it is possible for two teams to use two instances of an editor on the same workspace and perform modeling with different methodologies.

IV. Case Study

The effectiveness of the prototype implementing the *CoReTM* approach is demonstrated in a two-part case study. It is assumed that a government body aims to implement a digital COVID-19 certification scheme which provide evidence of recovery, vaccination or negative diagnosis [26]. As a baseline, an initial version of the digital COVID-19 approach employed in Switzerland is assumed, where users use a certificate represented as a quick response (QR) code, which holds a digital signature issued by the government upon recovery, testing or vaccination. Such certificates can then be verified using an additional application which reads the QR-code and verifies the digital signatures contained therein. Finally, validators can use the verified information to check the verified properties, such as the personal identification or health data [13].

This use-case demonstrates the applicability of *CoReTM*. First, following the "complete protection" principle [2], which postulates that security of the overall digital certification approach cannot be considered in a piecemeal manner. Thus, it is insufficient to consider the security of information systems from a purely technical perspective.

Secondly, such a digital certification approach naturally involves experts from many domains such as software engineers, healthcare employees, and government authorities. These key properties highlight, why this case study underlines the strength of *CoReTM*, which are the inclusion of stakeholders despite differences in geographical location, time, and skills. Thus, to elicit, trace, and mitigate threats, the remainder of this case study envisions a threat modeling workshop to capture domain knowledge, including high-level threats. Then, a technical threat assessment with software engineers and external security consultants is conducted focusing on key areas identified in the first workshop.

A. Business Process Threat Modeling

Following a risk-based approach, an effective application of cybersecurity measures requires a notion of the importance of the asset to be protected [44]. By focusing the threat analysis on important assets, threat modeling can be carried out with economic effectiveness in mind. Thus, it is critical not to start a threat model from a purely technical perspective. Hence, an initial threat modeling workshop is envisioned involving an expert from a COVID-19 testing laboratory, a healthcare system representative, and government personnel. Aside from such domain experts, a software architect from the project team that implemented the first version of the prototype and a cybersecurity consultant provide the perspective on the technical domain. Due to the ongoing pandemic, it is assumed that technical experts are working in an on-site setting, while the remaining three stakeholders are dispersed over multiple physical locations. Furthermore, it is critical that at the end of the threat modeling workshop, a report of the findings can be sent to the head of the public health department.

Since a cybersecurity expert is present, he/she can set up an initial workspace in the *CoReTM* prototype by accessing the web-based interface. There, he/she opens a new instance of the *Questionnaire*, which helps him/her set up the threat modeling workshop for the audience. Fortunately, there is already an existing business process description from a requirements engineering workshop conducted at the beginning of the project. Thus, the questionnaire moves forward to the identification of skills in the audience. Since most of the stakeholders are

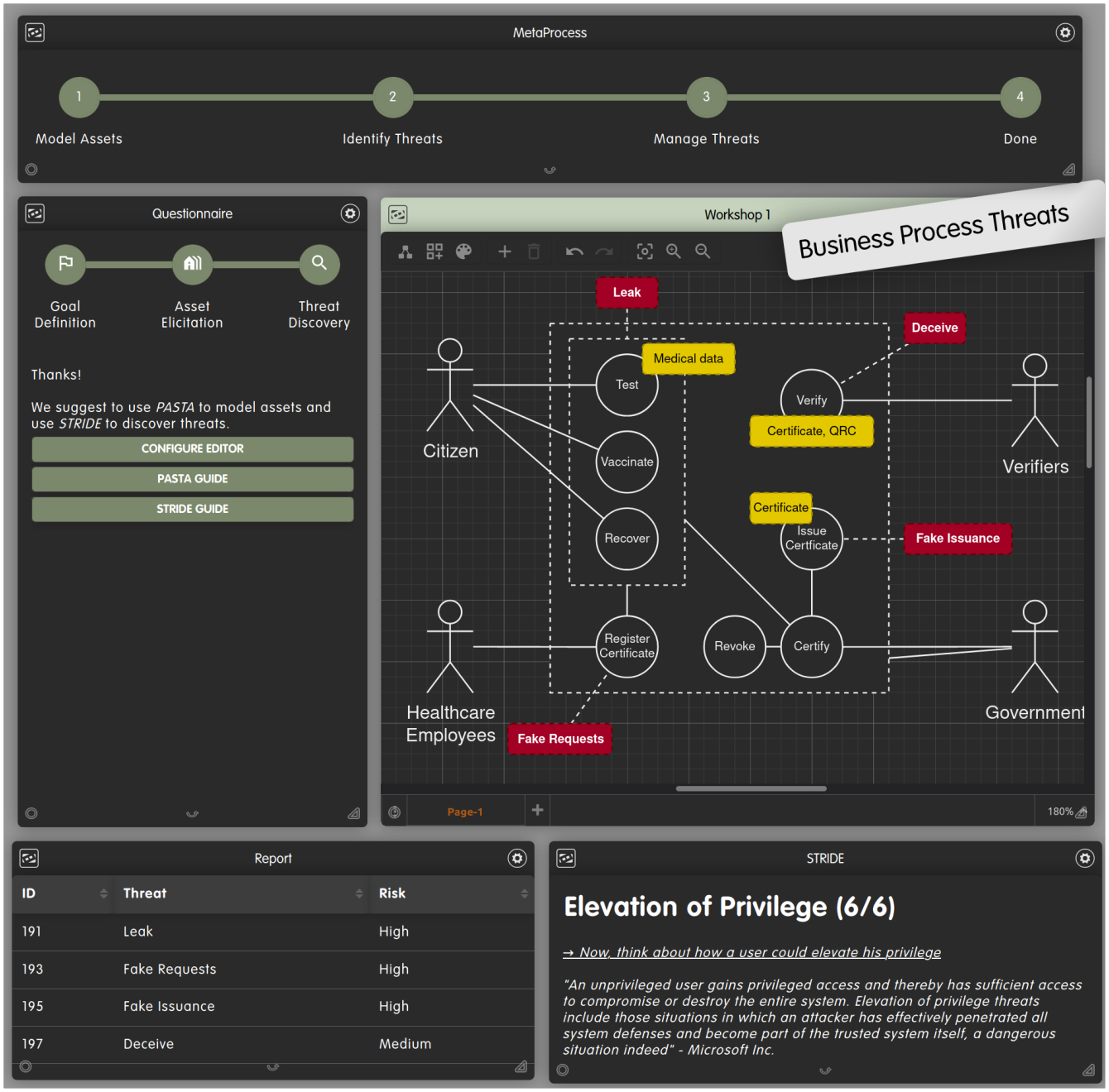

Fig. 8: Workspace After Conducting the First Workshop with *CoReTM*

not software engineers and no threat modeling methodology is known to the participants as they hold no cybersecurity knowledge, the tool determines that no asset modeling methodology is necessary, and that threat identification is best performed using the high-level *STRIDE* methodology. The tool automatically configures the modeling *editor* to provide visual elements for the *STRIDE* approach. Furthermore, the *walkthrough* component is automatically initiated to show the progress of the workshop and the *STRIDE* component provides information on how to apply the respective methodology.

At this point, the workshop leader can distribute the URL to participants using any communication channel. At the time of the scheduled meeting, all participants can open the application from their browser and work on the shared workspace in real time. First, the software architect uploads the existing process model using the editor, which is automatically rendered for all participants. Since the original model was created using the commercial tool Lucidchart, the model is converted and stored in the modeling database. By changing colors of those elements that denote assets, the solution is able to inventorize assets. Now, the *walkthrough* component shows that participants can start identifying threats. For that, the *STRIDE* component iterates over the six mnemonic threat types so that each participant can brainstorm, discuss, and annotate relevant threats in the diagram editor. Once all six threat types are considered, the *walkthrough* component moves to the threat management phase of the life cycle. Here, all annotated threats are rendered to a list, where participants can add further explanations. Furthermore, the impact of each threat is ranked by participants using a qualitative labeling from *low* to *high*. At this stage only the domain knowledge is captured. Metrics, such as the vulnerability of a threat asset or the cost of an attacker, are not yet considered.

As shown in Figure 8, with this procedure, the threat modeling workshop conducted with *CoReTM* allowed for the identification of two critical assets and three critical threats to the system, although participants had lacking cybersecurity knowledge and were not able to meet on-site. First, it appears that the largest threat to the certificates stems from personnel authorized to declare a persons' recovery, negative test or vaccination. Here, it appears vital that procedural controls are enacted to alleviate the threat of having **t**ampered information in the system. This finding stresses that not all threats are embodied in technical components. Thus, it is critical to consider domain experts which can be included with the presented solution. Furthermore, since certificates hold personal and health-related information, there is a clear risk that such **i**nformation is being disclosed. Finally, since certificates are issued by an information system, threats related to the **e**levation of privilege or data **t**ampering are considered with respect to certificate issuance. The applicability of such threats from a system vulnerability perspective has to be further modeled.

This result can be shared with the head of the public health department in an interactive form, since the full workspace can be shared by the workshop participants. In that sense, such an application is advantageous over static reports, since the full context is available, no efforts in creating a report are necessary and the possibility of extension is enabled due to the interactivity provided. These results serve as a baseline for the second threat modeling iteration, where the highlighted threat areas are investigated on a technical level to analyze the criticality of threats and vulnerability of the systems regarding these threats. Furthermore, these initial findings can already be used to justify to the management why the implementation of process control for certificate issuance needs attention.

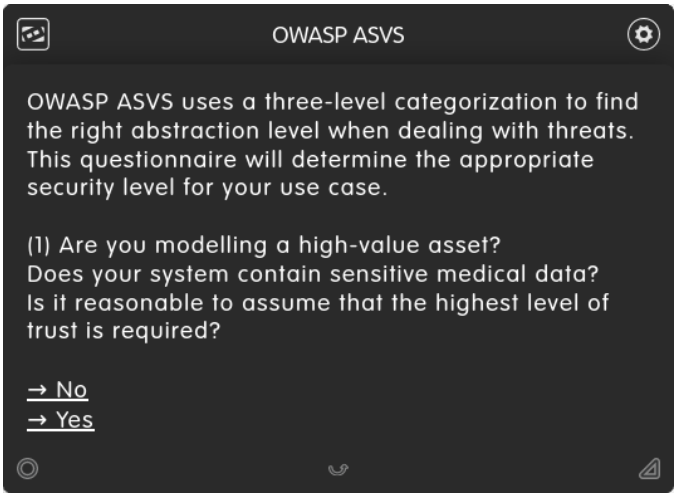

Fig. 9: Questionnaire Deriving the Security Level for the OWASP ASVS Methodology

B. Software Threat Modeling

The first threat modeling revealed that storing healthcare-related personal data and issuing certificates are critical assets. In a second iteration of the threat modeling workshop, a geographically distant security expert asynchronously guides a technically versed audience consisting of software engineers, software architects, and software testers to create a second threat model. Again, the questionnaire component guides the leader toward a "sensible" configuration. Existing diagrams depict the technical perspective of the architecture, and all participants understand the technical background of the Web application to be modeled. The questionnaire advises *(a)* using the existing diagram and *(b)* to model threats via the *OWASP Application Security Verification Standard*.

Once all the components are set up, the actual workshop is driven by the *OWASP* component without the need for the cybersecurity expert to be present at the same time or location, as it provides procedural and informational support to apply the methodology. First, the proper of three security levels is determined using a questionnaire, as presented in Figure 9. Since the system stores sensitive medical data, the highest level is proposed. Thus, the component automatically parametrizes itself as to show threats relating to this security level. It is critical to highlight that according to the *OWASP ASVS* standard, this level of security cannot be achieved in a

purely automated manner but requires *"(..) access to (..) the people involved in the development process"* [31], stressing the necessity of a collaborative system.

To provide a technical notion of the previously identified assets, the corresponding technical components are marked using the color that denotes assets. Threats can then be discovered by either searching through the catalog based on keywords or by browsing through the threat categorizations. However, as shown in Figure 10, simply searching for the keywords, that are used to describe assets already shows a promising subset of threats to be considered. This is facilitated since the reporting component summarizes assets that were annotated in the editor. For example, one term involved to denote the security of the certificate asset are the *cryptographic keys* which are used to sign certificates. A simple search for this term already yields the following hint:

> "Verify that cryptographic keys used in verification are stored securely and protected against disclosure, such as using a Trusted Platform Module (TPM) or Hardware Security Module (HSM), or an OS service that can use this secure storage." [31]

Furthermore, searching for *medical data* yields additional threats which can be annotated on the diagram and further outlined in the report component. For example, *OWASP ASVS* advises protecting such data at rest to mitigate privacy-related threats. Thus, based on interaction on the *CoReTM* platform, multiple critical threats including cryptographic key management and data privacy are discovered and assessed. In contrast with conventional reports, the resulting threat model is interactive so that it can be directly integrated into the software development process. Furthermore, it can be used in an asynchronous manner, such as to conduct an audit by security experts or to convince senior management of a key finding.

These scenarios demonstrate that technical personnel can collaboratively derive threats of a piece of software architecture without an in-depth cybersecurity knowledge. Such a collaboration is possible even for remote or asynchronous collaborators. Being able to collaboratively assess threats is particularly useful, since this latter phase shows that only builders of the application can assess the relevance of threats discovered. By using such a collaborative approach, all technical activities are driven by business value, since the threat modeling is centered around a notion of critical assets. Bridging such gaps in technical knowledge is achieved with a flexible approach toward methodology selection and application.

V. Summary and Future Work

This paper introduced *CoReTM*, a collaborative approach to discover, assess, and manage assets and the threats surrounding them. *CoReTM* stands as a management tool for risk management and architecture development even in cases where domain and subject experts are not able to collaborate due to differences in geographical location, time or skills. Therefore, *CoReTM* implements a collaborative editor where

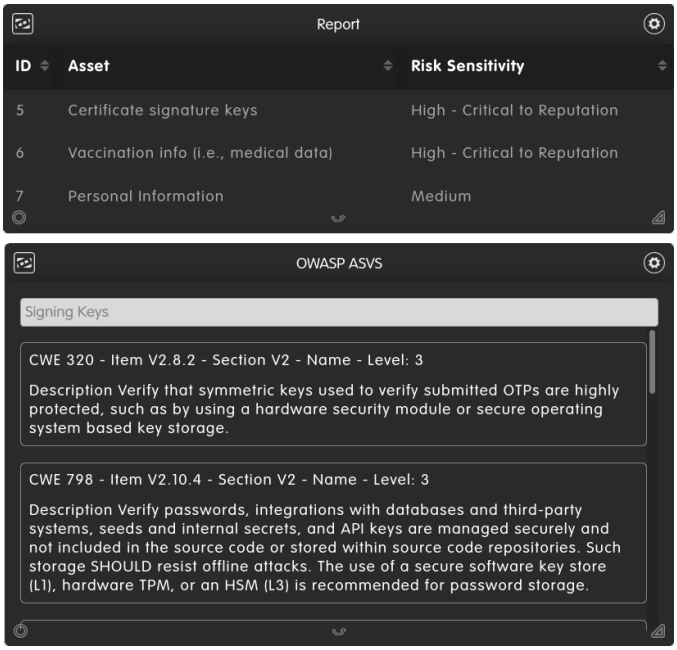

Fig. 10: Searching in an Indexed Catalog of the OWASP ASVS Standard using Asset Labels as Keywords

assets and threats are flexibly modeled. Based on the methodology that chosen by navigating the questionnaire, *CoReTM* guides collaborators through the threat modeling process and the underlying methodology.

To the best of the authors' knowledge, *CoReTM* it is the first platform specifically designed around threat modeling that supports remote collaboration. Furthermore, no other platforms consider a meta-modeling framework which considers the breadth of available methodologies with the goal of optimizing threat methodology. As demonstrated in the case study, *CoReTM* enables focusing threat discovery and assessment around a notion of critical assets, so that relevance to the business is be preserved. Thus, the contributions of this paper are as follows:

- A survey highlighting new tools that have emerged since previously conducted reviews [21].
- A meta-modeling methodology that guides users through the stages of threat modeling including methodology selection and their application for asset and threat modeling. Applying a selected methodology is guided with supportive information. Thus, *CoReTM* enables cross-functional collaboration for threat modeling, even when stakeholders are not security experts.
- *CoReTM* provides a running prototype which allows various methodologies to be applied even when collaborators cannot meet because of availability or geographical location. This addresses the limitation of current tools that do not support virtual collaboration.
- By decoupling the methodology from the tool itself, *CoReTM* stands as an extensible open source tool, so that organizations can implement their own methodologies.

Although it is considered that high-level threat discovery methodologies can be used for threat modeling on the business layer, future work includes formulating and validating a threat discovery methodology for such business processes. Furthermore, this work considers that automated threat modeling cannot completely replace human insight, especially when assessing the value of assets and relevance of discovered threats. However, as part of future work, novel techniques such as business process mining or natural language processing could be investigated with respect to their potential for business threat modeling.

ACKNOWLEDGEMENTS

This paper was supported partially by *(a)* the University of Zürich UZH, Switzerland and *(b)* the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement No. 830927, the CONCORDIA project.

REFERENCES

[1] Akamai, "2021: Volumetric DDoS Attacks Rising Fast," March 2021, https://blogs.akamai.com/2021/03/in-our-2020-ddos-retrospective, Last Visit December 2021.

[2] K. S. Anne Kohnke, Dan Shoemaker, *The Complete Guide to Cybersecurity Risks and Controls*. Boca Raton: Taylor and Francis, 2016.

[3] K. Bernsmed, D. Cruzes, M. Jaatun, and M. Iovan, "Adopting threat modelling in agile software development projects," *Journal of Systems and Software*, vol. 183, p. 111090, 09 2021.

[4] E. S. Brenda Larcom and S. Smith, "Docs — octotrike.org," http://www.octotrike.org/docs, Last Visit March 2022.

[5] CAIRIS, "Threat Modelling, Documentation and More," 2022, https://cairis.org/cairis/tmdocsmore/, Last Visit March 2022.

[6] CAPEC, "About CAPEC," April 2019, https://capec.mitre.org/about/index.html, Last Visit March 2022.

[7] ——, "CAPEC - ATT&CK Comparison," 2019 October, https://capec.mitre.org/about/attack_comparison.html, Last Visit March 2022.

[8] Cloudflare, "DDoS attack trends for 2021 Q2," July 2021, https://blog.cloudflare.com/ddos-attack-trends-for-2021-q2/, Last Visit December 2021.

[9] CONCORDIA Consortium, "Cybersecurity Roadmap for Europe," November 2021, https://www.concordia-h2020.eu/roadmap/, Last Visit March 2022.

[10] devsecops, "Raindance," June 2016, https://github.com/devsecops/raindance/blob/master/GET-STARTED.md, Last Visit March 2022.

[11] B. Doerrfeld, "Majority of Software Engineers Want Remote Work Options," August 2021, https://devops.com/majority-of-software-engineers-want-remote-work-options/, Last Visit January 2022.

[12] S. Faily and C. Iacob, "Design as Code: Facilitating Collaboration Between Usability and Security Engineers Using CAIRIS," in *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, 2017, pp. 76–82.

[13] Federal Office of Public Health of the Swiss Confederation, "Information on the COVID-19 certificate FOPH," February 2022, https://foph-coronavirus.ch/certificate/, Last Visit March 2022.

[14] forseeti, "Introduction," https://docs.foreseeti.com/docs, Last Visit March 2022.

[15] ——, "MAL - Meta Attack Language," https://mal-lang.org/, Last Visit March 2022.

[16] M. Franco, J. Von der Assen, L. Boillat, C. Killer, B. Rodrigues, E. J. Scheid, L. Granville, and B. Stiller, "SecGrid: a Visual System for the Analysis and ML-based Classification of Cyberattack Traffic," in *IEEE 46th Conference on Local Computer Networks (LCN 2021)*, Edmonton, Canada, October 2021, pp. 140–147.

[17] N. Fraser, "Differential Synchronization," in *DocEng'09, Proceedings of the 2009 ACM Symposium on Document Engineering*, 2 Penn Plaza, Suite 701, New York, New York 10121-0701, 2009, pp. 13–20. [Online]. Available: http://neil.fraser.name/writing/sync/eng047-fraser.pdf

[18] I. Gartner, "Gartner Forecasts 51% of Global Knowledge Workers Will Be Remote by the End of 2021," June 2021, https://www.gartner.com/en/newsroom/press-releases/2021-06-22-gartner-forecasts-51-percent-of-global-knowledge-workers-will-be-remote-by-2021, Last Visit December 2021.

[19] M. Henriksen, "Draw.io for threat modeling," October 2018, https://michenriksen.com/blog/drawio-for-threat-modeling/, Last Visit March 2022.

[20] G. Hill, "diagrams.net-tutamen-entry-popup," January 2021, https://github.com/geoffrey-hill-tutamantic/diagrams.net-tutamen-entry-popup, Last Visit April, 2022.

[21] S. Hussain, A. Kamal, S. Ahmad, G. Rasool, and S. Iqbal, "THREAT MODELLING METHODOLOGIES: A SURVEY," vol. 26, pp. 1607–1609, 01 2014.

[22] IriusRisk, "Threat Modeling Platform," https://www.iriusrisk.com/threat-modeling-platform, Last Visit March 2022.

[23] JGraph, "GitHub - jgraph/drawio: Source to app.diagrams.net," January 2022, https://github.com/jgraph/drawio#open-source-not-open-contribution, Last Visit March 2022.

[24] JGraph Ltd, "Diagram Software and Flowchart Maker," https://www.diagrams.net/, Last Visit March 2022.

[25] ——, "Embed mode," November 2020, https://drawio.freshdesk.com/support/solutions/articles/16000042544-embed-mode, Last Visit March 2022.

[26] G. Karopoulos, J. L. Hernandez-Ramos, V. Kouliaridis, and G. Kambourakis, "A Survey on Digital Certificates Approaches for the COVID-19 Pandemic," *IEEE Access*, vol. 9, pp. 138 003–138 025, 2021.

[27] Microsoft, "Microsoft Threat Modeling Tool," November 2020, https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling-tool, Last Visit March 2022.

[28] ——, "Microsoft Threat Modeling Tool threats," March 2022, https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats, Last Visit March 2022.

[29] National Institute of Standards and Technology (NIST), "Guide to Data-Centric System Threat Modeling," March 2016, https://csrc.nist.gov/publications/detail/sp/800-154/draft, Last Visit March 2016.

[30] OWASP Foundation, "OWASP Threat Dragon," November 2021, https://owasp.org/www-project-threat-dragon, Last Visit March 2022.

[31] OWASP® Foundation, "OWASP Application Security Verification Standard," October 2021, https://owasp.org/www-project-application-security-verification-standard/.

[32] ——, "OWASP Top Ten," October 2021, https://owasp.org/www-project-top-ten/, Last Visit March 2022.

[33] H. Per, P. H. Meland, D. Spampinato, E. Hagen, E. Baadshaug, K.-M. Krister, and K. Velle, "SeaMonster: Providing tool support for security modeling Per Hakon Meland," 09 2010.

[34] C. Schneider, "Agile Threat Modeling," https://christian-schneider.net/service/agile-threat-modeling/, Last Visit March 2022.

[35] ——, "Threagile/threagile: Agile Threat Modeling," 2020, https://github.com/Threagile/threagile, Last Visit March 2022.

[36] SecurityCompass, "SD Elements: Your essential secure development solution - Security Compass," https://www.securitycompass.com/sdelements/, Last Visit March 2022.

[37] ——, "SD Elements Datasheet v5.17," month year, https://docs.sdelements.com/release/latest/guide/docs/datasheet.html/, Last Visit March 2022.

[38] A. Shostack, *Threat modeling: Designing for security*. John Wiley & Sons, 2014.

[39] I. Tarandach, "pytm: A Pythonic framework for threat modeling," October 2021, https://github.com/izar/pytm, Last Visit March 2022.

[40] ThreatModeler Software, Inc., "Intelligent Threat Engine," https://threatmodeler.com/automated-threat-modeling-tool/, Last Visit December 2021.

[41] Threatspec, "Threatspec," June 2019, https://threatspec.org/, Last Visit March 2022.

[42] Tutamantic, "Feauture — Tutamantic," January 2021, https://www.tutamantic.com/page/features, Last Visit March 2022.

[43] T. Ucedavélez and M. M. Morana, "Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis," 2015.

[44] E. Wheeler, "Security Risk Management." Boston: Syngress, 2011, pp. i–ii. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9781597496155000244

Jan von der Assen, Muriel F. Franco, Christian Killer, Eder J. Scheid
IfI Technical Report No. 2022.04, April 2022
Communication Systems Group CSG
Department of Informatics IfI
University of Zürich UZH
Binzmühlestrasse 14, CH—8050 Zürich, Switzerland
URL: http://www.csg.uzh.ch/