Tool-assisted Threat Modeling for Smart Grid Cyber Security

Lars Halvdan Flå*†, Ravishankar Borgaonkar†, Inger Anne Tøndel† and Martin Gilje Jaatun†
*Norwegian University of Science and Technology
lars.flaa@hotmail.com
†SINTEF Digital
Trondheim, Norway
{ravi.borgaonkar,IngerAnne.Tondel,gilje}@sintef.no

***Abstract*—Threat modeling is about identifying architectural flaws and weaknesses in a system in order to mitigate them and avoid unwanted incidents caused by an attacker. Tool-assisted threat modeling has seen limited use in complex cyber-physical systems involving both Information Technology (IT) and Operational Technology (OT) systems. In this paper, we investigate the applicability of tool-assisted threat modeling to the complex cyber-physical system that is the smart grid, and present a new Smart Grid template for the Microsoft Threat Modeling Tool. We demonstrate benefits of our smart grid threat modeling template on a use-case, and discuss limitations.**

***Index Terms*—Threat modeling, Smart Grid, Data Flow Diagram, STRIDE**

I. Introduction

The smart grid is evolving into an extremely complex cyber-physical system, merging the discipline of communication and information technology with that of electrical power engineering. Cybersecurity in this domain is of great interest for several reasons. The power grid is a critical infrastructure and a necessity for modern life. Consequently, the grid has strict requirements on power availability. Due to its immense size and the increasing usage of Information and Communication Technology (ICT) in the grid, the attack surface is equally large. The ICT components in the smart grid transmit customer data, consumption data, and operator control commands, amongst others. Attacks on these components and communication have the potential to cause a blackout, device malfunction, and violation of privacy.

The 2015 and 2016 cyber-attacks on the Ukraine power grid demonstrated the vulnerability of the power grid. The 2015 attack compromised three distribution companies and caused a blackout affecting 225 000 customers [1]. According to Slowik [2], the 2016 attack was less severe with regards to impact but indicated an increase in the attacker's ambitions. Slowik argues that a more widespread blackout, along with potential physical destruction of equipment, may have been the original objective of the attack. These attacks on the Ukrainian power grid can be viewed in a larger context of increases in cyber-attacks on industrial control systems [3] and challenges to Smart Grid transformations.

To protect the future energy grid from attackers, potential threats must be identified and addressed. These threats may target the generation and distribution of power, the trade of power, or the large amounts of sensitive information generated about the consumers in the grid. These threats emerge in between the disciplines of power engineering and information technology [4], and we have seen in the past that differences in disciplines create particular problems when dealing with security in cyber-physical systems [5].

This paper describes a smart grid template [6] for the Microsoft Threat Modeling Tool (TMT), which allows asset owners to model use cases in the smart grid. The modeling process automatically enumerates potential threats to the smart grid, provides an environment to systematically treat and classify discovered threats and provides a template for creating more extensive and specialized templates in the future.

The remainder of this paper is structured as follows: Section II introduces threat modeling. Section III presents the developed smart grid template. Section IV demonstrates the application of the template on a use case. Section V discusses the template and threat modeling for the smart grid. Section VI concludes the paper.

II. Threat Modeling

Threat modeling [7] is a technique originating from software security. Although threat modeling has been applied to cyber-physical systems, tool-assisted threat modeling has not seen wide application. Microsoft has developed a freely available Threat Modeling Tool (TMT)[1] that follows the principles laid out by Swidersky and Snyder [7] and Shostack [8], facilitating drawing of Data Flow Diagrams (DFDs) and automatic threat generation.

There are other threat modeling tools available, most notably the Open Web Application Security Project (OWASP) Threat Dragon[2]. However, Threat Dragon is not yet as advanced as TMT in terms of automatic threat generation and

This work has been supported by WP2 of CINELDI – Centre for intelligent electricity distribution, an 8-year Research Centre under the FME-scheme (Centre for Environment-friendly Energy Research, 257626/E20). The authors gratefully acknowledge the financial support from the Research Council of Norway and the CINELDI WP2 partners.

[1]https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling-tool

[2]https://owasp.org/www-project-threat-dragon/

usability. Still, OWASP is continuing development on this project, and it is likely to be a contender in the near future.

There exist some attempts to apply the Microsoft TMT and STRIDE[3] in other cyber-physical domains. The cybersecurity firm *nccgroup* develops a template[4] for automotive threat modeling. Microsoft creates three templates[5] for the domains of Azure Cloud services, Medical Devices, and a default general IT template. Khan et al. [9] developed a five-stage methodology for applying STRIDE to cyber-physical systems.

Shostack [8] describes a method for threat modeling consisting of four steps: building a diagram, finding threats, addressing threats, and checking the work. By using a threat model, details are abstracted away in order to provide the full picture. Shostack identifies four reasons for threat modeling. The first is to find security bugs early. Identifying issues before the system is built saves expensive and less ideal fixes later in the development. The second is to understand your security requirements. The third is to engineer and deliver better products. Considering requirements and design in the early stages of the process results in a better product. The fourth reason is that threat modeling can help address issues missed by other techniques.

Tøndel et al. [10] investigate the threats to an AMI configuration using STRIDE. Jiang et al. [11] decompose the threat of energy theft in AMI using attack trees. Liu et al. [12] use Petri Nets to analyze threats to communication and information in a smart meter. Suleiman et al. [13] analyze threats to the Smart Grid using Security Quality Requirements Engineering and Security Requirements Engineering Process.

III. Threat Modeling Tool & Our Template

This section discusses the TMT and the Smart Grid template we developed. The TMT provides an environment for analyzing threats and for creating custom templates. When creating templates the designer is offered great flexibility in defining suitable stencils, threats and configurations. A TMT stencil can take the form of a DFD process, data store, data flow, external interactor (entity) or trust boundary. When using the template to create the DFD for threat modeling, the user will select among the available stencils when drawing processes etc. The larger threat modeling tool ecosystem can be seen in Fig. 1. The contribution of this paper is marked in green.

The template was developed using version 7.3.00929.2 of the Microsoft threat modeling tool. The tool can be downloaded from Microsoft and our template from Github[6].

A. Design Choices made for the Smart Grid Template

The creation of the template is informed by the design choices listed below.

- The threats do not include natural failure rates due to wear, tear, natural disasters, and similar. Only cyber threats are considered.

[3]STRIDE is an acronym composed of: Spoofing – Tampering – Repudiation – Denial of Service – Elevation of Privilege

[4]https://github.com/nccgroup/The_Automotive_Threat_Modeling_Template

[5]https://github.com/microsoft/threat-modeling-templates

[6]https://github.com/larshfl/MS-TMT-Smart-Grid-Template

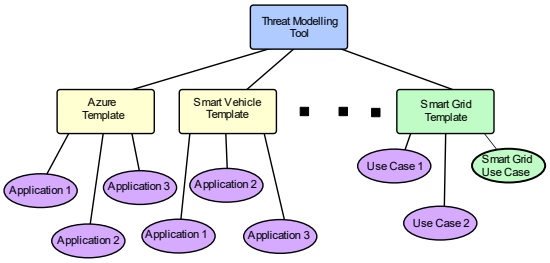

Fig. 1. Threat Modeling Tool Ecosystem

- The process stencils represent the functional behavior of equipment and are not directly transferable to the operating system or software processes.
- Only databases are included for storing data. Other forms of storage, for instance, files, memory, cache, or computer permanent storage, are not included.
- Insider and physical access threats are not included.
- Threats originating from external service or maintenance personnel, or their potentially compromised equipment, are not included.
- Threats involving sabotage are not included.
- Threats originating from forgotten data flows are not modeled. E.g., the template does not account for what threats may arise from a VPN connection not included.
- All default properties in the template are set to the values that generate the most threats. This is done to ensure that no threats are overlooked because default properties were not changed.

B. Data Flow Diagrams

The threat modeling tool is based on creating DFDs of the system or application. According to Shostack [14] and Swiderski and Snyder [7], models of data flow are well suited for threat modeling. Advantages, according to Swiderski and Snyder, include that various parts can be described in varying degrees of detail. The DFD elements used in this paper [6] have a slightly different meaning from the classical DFD elements. This is done to make them more suitable for threat modeling in the smart grid.

C. Trust boundaries

The trust boundary separates more trusted components (entities, processes, ...) from less trusted components. This is similar to how it is viewed by Khan et al. [9], who claim that the trust boundary is used to separate trustworthy and untrustworthy systems. Another way to view the smart grid trust boundary is that it should be placed on all communication interfaces where threats are expected.

All threats in the template are tied to trust boundaries. During the threat modeling, boundaries should be included on all data flows the program shall investigate and generate threats for. An example of interfaces that could be of interest is data arriving across untrusted networks. Another example

could be interfaces sending control commands to processes controlling the power in the smart grid.

D. Stencils added to the Smart Grid Template

We derived stencils from literature, the use case described in Section IV, and discussions with experts from the power system domain. We assume that all systems that we want to represent using the developed stencils have the capability of communicating and executing code. Consequently, most threats relate to the stencil categories rather than the individual stencils. This means that individual stencils, for instance, the Circuit breaker Process and the Intelligent Electronic Device (IED) Process in many cases, generate the same threats. The default value of the stencils is the option resulting in the largest attack surface, as explained in Section III-A.

General smart grid processes represent the functional behavior of smart grid components or systems. The focus is on their communication interfaces towards other processes, as threats are generated for (source, flow, target)-tuples. This is shown in Fig. 2 and described in Section III-F.

Generic External Interactors represent systems or actors that interact with the smart grid. Threats *originating from* an external interactor are included, but threats *affecting them* are not included. For instance, the threat of an attacker exploiting weak credentials on a VPN connection from a vendor organization into a smart grid process is included. The threat of a Denial of Service attack on a vendor organization, originating from the smart grid or elsewhere, is not included.

Generic data stores represent storage of data and our template only includes databases. Other forms of data storage, for instance, cache, memory, and disk are assumed to be part of the smart grid processes. Like the case with processes, the data stores do not map directly to ICT components. Data stores include the data being stored. The reading and writing of data is done by a process. Generic data flow represents the flow of all forms of "useful" or application-level data.

E. Threats added to the Smart Grid Template

This section describes the threats included in the template. The threats are derived from literature, existing templates, and cyber-attacks on OT systems [6]. Space does not permit describing all threats here, but Flå [6] provides the details in his chapters 4 and 5. We will refer to these threats using the same numbering, e.g. "Literature threat 1.g". The added threats are grouped into STRIDE-categories.

1) Spoofing: **Control input spoofing:** This is the threat of an attacker sending control input to a process, pretending it originates from a legitimate source. Such sources may be Windmill Process, Substation Process, IED Process, Automatic Voltage Regulator Process, Circuit breaker Process, Onload Tap Changer Process, phasor measurement unit (PMU) Process, Remote Terminal Unit (RTU) Process, and virtual RTU Process. In this way, the attacker can cause a process responsible for controlling the grid to behave in a malicious way. The threat is inspired by literature threat 6.d [6] relating to DERs and generalized to the processes mentioned above.

Spoofing the source: This is the threat of an attacker pretending to be a legitimate process, data store, or external interactor. The attacker would attempt to exploit this by making processes or data stores believe the communication originates from a trusted source. This could lead to unauthorized access to a process or to incorrect data being sent to a process. The threat is adapted from the default template.

Spoofing the target: This is the threat of an attacker pretending to be a legitimate process, data store, or external interactor. The attacker would attempt to exploit this by making processes, data stores, or external interactors believe it is sending data to a legitimate target. This may lead to information being sent to the attacker instead of the legitimate process. The threat is adapted from the default template and mentioned in literature threat 1.b [6].

Spoofing of data store source: This is the threat of an attacker sending malicious data to a process by pretending to be a legitimate data store. This could cause the process to behave in a malicious way by tricking it into basing decisions on false data. The threat is included from the default template.

MITM-Attack: This is the threat of an attacker performing a Man-in-the-Middle (MITM) attack on communication between any of the processes, data stores, or external interactors in the grid. This general threat is inspired by the Azure template, where the threat is generated for IoT related traffic. Different forms of MITM attacks have been identified in the literature, as indicated by literature threats 3.d, 6.i, 6.o, and 10.c [6]. The smart grid template generalizes this threat to all types of communication.

Reuse of authentication tokens: This is the threat of an attacker acquiring cryptographic key material from a legitimate IoT Device Process or an IoT Field Gateway Process and using it to communicate with an IoT Field Gateway Process or an IoT Cloud Gateway Process. Falsely authenticating as someone else may give the attacker the possibility of sending false data to the process or receiving data meant for someone else. The threat is included from the Azure template.

GPS spoofing: This is the threat of an attacker sending false GPS signals to a PMU Process. PMUs generally rely on GPS to timestamp their measurements. These measurements may later be used for state estimation, and a successful GPS spoofing attack may cause the grid operators to estimate a wrong state. The threat is inspired by literature threat 4.a [6].

Replay attack: This is the threat of an attacker capturing a message from the network and resending it at a later time. A replay attack is assumed to be possible for communication between any types of stencils if the data flow does not provide replay protection. A replay attack is essentially a way of providing the sender with bad input. The most serious consequences can be achieved if the attacker has knowledge of the target under attack. The threat is inspired by literature threat 10.d [6], which claims that plain-text SCADA systems may be vulnerable to replay attacks. The smart grid template generalizes this to all communication.

2) Tampering: **Tampering of communication:** This is the threat of an attacker tampering with a data flow. The threat is

not generated if the communication provides integrity or if the communication is human input or output. The consequence of a successful attack can be that the target stores a false value in a database or otherwise behaves in a malicious manner. An example of tampering can be found in literature threat 1.c [6].

Injection of data in optical fiber: This is the threat of an attacker injecting data into communication over optical fiber. The consequences are much the same as for the tampering of communication threat. The threat is inspired by literature threat 6.f [6], related to communication with DERs via fiber optical cables. The smart grid template generalizes this to all fiber optic communication.

SQL injection attack: This is the threat of an attacker performing an SQL attack on an SQL relational database that does not sanitize input. An SQL injection attack may corrupt the database content or reveal the content to the attacker. The threat is inspired from literature threats 3.c and 11.b [6] related to SQL attacks on SCADA historian databases and from SQL injection threats in the default template.

Corruption of data store by tampering of data flow: This is the threat of an attacker tampering with a data flow going to a data store. The consequence of such an attack is that false data is stored in the data store (adapted from the default template).

3) Repudiation: **Repudiation of sent/received data:** This is the threat of not being able to prove whether a process or data store did send or receive a message. Lack of such proof may make it hard to investigate attacks, or deny false claims motivated by financial gain. The threat is not generated if the actions on the database or data store are logged. The threat is adapted from the default template. This threat is highlighted in literature threat 1.d [6].

Repudiation of actions on smart grid process: This is the threat of not being able to prove whether an action was committed on a process or not. This can lead to repudiation claims after an attack and make it harder to attribute an attack to an actor. The threat is inspired by similar threats in the Azure template related to databases and cloud gateways.

4) Information Disclosure: **Data flow sniffing:** This is the threat of an attacker learning the contents of a data flow in the grid. If the flow does not offer confidentiality, this could lead to theft of confidential information or be used to reverse engineer commands in preparation for a later attack. The threat is based on literature threats 1.e, 4.b, 5.a, 6.a, 6.g, and 10.a [6], which relate to the disclosure of transmitted information.

Wiretapping of fiber optic cables: This is the threat of an attacker wiretapping optical fiber cables to learn the content of the communication. If the flow does not offer confidentiality, the consequences are the same as for the data flow sniffing threat. The threat is based on literature threat 6.a [6], which relates to wiretapping at the physical level.

Exploitation of weak credential transit: This is the threat of an attacker sniffing credentials as they are transmitted to processes or data stores. If transmitted credentials are not encrypted, they may be sniffed and used to obtain elevated privileges (adapted from the default template).

Exploitation of weak credential storage: This is the threat of an attacker obtaining credentials from a data store. Such credentials may be used to obtain elevated privileges. The threat is adapted from the default template. The default template argues that stored credentials may be stolen, tampered or disclosed. To prevent this, credentials should ideally be hashed or encrypted.

5) Denial of Service: **External distributed denial of service attack:** This is the threat of distributed attack on the availability of a process originating from an external network. Such an attack may cause the target to become temporarily unavailable to legitimate communication from other sources. The threat is inspired by literature threats 1.a, 3.b, 1.f, 6.b, 7.a, 7.d, 7.e, 7.f, 7.g, and 9.c [6], which all deal with generating large amounts of traffic, possibly from distributed hosts. An external network provides the possibility for an attacker to compromise many hosts without the knowledge of the asset owner and use these to launch an attack.

Smart meter-based DDoS attack on AMI server: This is the threat of an attacker compromising many smart meters and subsequently using them for a DDoS attack on an AMI Server. Such an attack could cause the AMI server to become unavailable. The threat is inspired by literature threat 9.c [6].

Denial of service through specially crafted message: This is the threat of an attack on the functional availability of a circuit breaker using a specially crafted package. According to Slowik [2], an attempt was made in the 2016 Ukraine attack to place a safety breaker in a firmware update mode, leaving it in a state unable to perform its normal function. The attack attempted this by exploiting a vulnerability in the device by sending it a specially crafted UPD packet. Literature threat 1.f [6] claims that specially crafted messages may be a way of denying service.

Signal jamming: This is the threat of an attacker jamming the wireless communication between processes, data stores or external interactors in the grid. The threat is inspired by literature threat 6.j [6] concerning communication with a DER. The smart grid template generalizes this threat to all types of wireless communication.

Protocol-specific flooding: This is the threat of an attack on the network availability of components in the grid communicating via a specific protocol (e.g., TCP or UDP). The attack is performed by generating large amounts of network traffic, blocking legitimate traffic. A successful attack causes the target to become unresponsive for a period. The threat is based on literature threat 6.b, 7.d, 7.f, and 7.g [6]. We generalized the threat to all processes using the specific protocol.

Interruption of data flow: This is the threat of an attacker disrupting the data flow, attacking the network availability of the target. This is a general threat to encourage reflection of how such an interruption may occur in the use case under consideration (adapted from the default template).

Denial of service of data store: This is the threat of an attacker making the data store inaccessible. This is a general threat to encourage reflection of how such an interruption may

occur in the use case under consideration (adapted from the default template).

6) Elevation of Privilege: **Elevation of privilege in database due to poor configurations:** This is the threat of an attacker obtaining greater privileges than intended in a database and adapted from the Azure template. More specifically, the threat is generated if access to a database is not configured based on least privilege. Least privilege implies that a user does not have more permissions than what is needed.

Execution of malware: This is the threat of an attacker executing malware in a process. Execution of malware has been observed in the Triton [3], Crashoverride [15], and Stuxnet [16] attacks on industrial control systems. Injection of malware is also regarded as a threat in the literature, as indicated by threat 1.h, 7.b, 7.d [6].

Exploitation of publicly disclosed vulnerabilities: This is the threat of an attacker exploiting a publicly disclosed vulnerability in a process or a data store in order to obtain elevated privileges. Vulnerabilities are continuously discovered and disclosed. Failure to update systems after such vulnerabilities have been made publicly known lowers the effort for conducting an attack. According to Slowik [2], an example of this can be found in the 2016 Ukrainian attack. The attackers attempted to exploit a vulnerability that was already publicly known. The threat is additionally inspired by the Azure Cloud Service template, where the threat is generated for IoT Devices or IoT Gateways. The smart grid template has extended the threat to regard all smart grid processes and data stores.

Exploitation of unused services or features: This is the threat of an attacker exploiting unnecessary functionality in order to access and obtain elevated privileges on a process or data store. The threat is adapted from the Azure Cloud Service template, where it is included for IoT devices and IoT gateways. In the smart grid template, this is extended to all smart grid processes and data storage. An example of such services may be open ports, as identified in threat 11.a [6]. ICS-CERT [1] recommends that ports are closed and unused services turned off. Staggs et al. [17] argue that system hardening can be used as a mitigation technique. This includes disabling unnecessary remote interfaces, removing unused interfaces and functionality, and adjusting default configurations to fit the operating environment.

Exploitation of lack of input validation: This is the threat of an attacker giving malicious input to a process or data store in order to obtain elevated privileges. A well-known form of such a threat is a buffer overflow attack, as highlighted in literature threat 11.c [6]. The default template states that failure to verfify input is the root cause of many exploitable issues. The smart grid template generalizes the threat to all processes and data stores.

Unauthorized access through vendor VPN: This is the threat of an attacker obtaining access to a process through a vendor VPN. This threat is inspired by literature threat 6.k [6] related to DERs and on the Crashoverride attack. ICS-CERT [1] reports that a VPN connection may have been used by attackers to open circuit breakers in the 2015 Ukraine attack. The smart grid template generalizes the threat to all processes that are configured to be accessible through VPN.

Unauthorized execution of commands: This is the threat of an attacker executing unauthorized commands on a process or data store. The threat is adapted from the Azure Cloud Service template, where is included for IoT related communication. The smart grid template generalizes this threat to all processes and data stores.

Exploitation of weak authentication: This is the threat of an attacker obtaining elevated privileges on a process or data store due to weak authentication mechanisms. This can be the case if the authentication mechanism consists of easily guessable credentials or factory default credentials. The threat is adapted from the default template. In the Azure template, it is included for databases. The smart grid template thus makes the threat relevant to both processes and data stores.

Remote control of circuit breakers: This is the threat of an attacker obtaining control of a remote circuit breaker in the grid. This threat is inspired by the 2015 Ukraine attack where ICS-CERT [1] reports that the attackers opened circuit breakers in the grid. Malware containing this functionality was also identified in the 2016 Crashoverride attack [15].

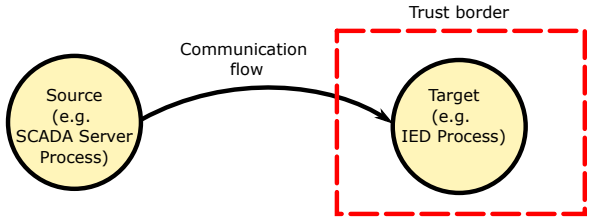

Fig. 2. Data flow across a threat boundary in the Threat Modeling Tool

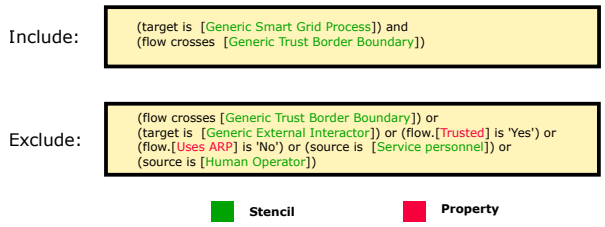

Fig. 3. Examples of include and exclude logic

F. Generation of threats

Threats in the TMT are based on a tuple of (source, communication flow, target). This method of generating threats in STRIDE is referred to as STRIDE-by-interaction, according to Shostack [8]. An illustration is given in Fig. 2.

Each threat added to the template is included or excluded in the analysis according to boolean logic related to the (source, communication flow, target)-tuple. A threat is included in the analysis if the include logic evaluates to true. Threats that otherwise would have been included may be excluded if the exclude logic evaluates to true. An example of include and exclude logic is given in Fig. 3. This logic determines if the ARP flooding threat is included in the analysis.

IV. Application to a Use Case

The Smart Grid use case is based on a setup shown in Fig. 4. The setup is a simple example of transmission and distribution line control. A Transmission System Operator (TSO) controls a generation source and the high voltage part of the line. A Distribution System Operator (DSO) controls a windmill and the medium voltage part of the line. The high and medium voltage sections are interconnected by an On-Load-Tap-Changer (OLTC).

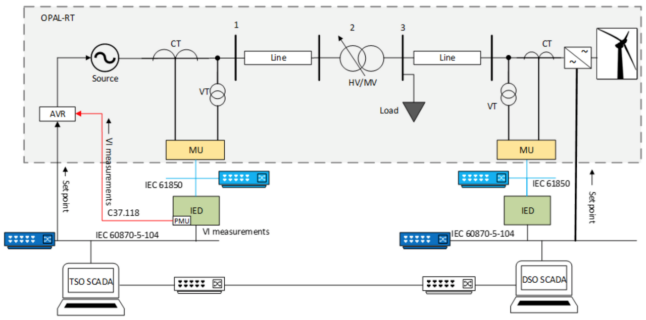

Fig. 4. A Smart Grid Use Case Setup

Modeling of the use case in the Microsoft TMT resulted in the model shown in Fig. 5. Network routers, the TSO generation source, the current transformers, and the voltage transformers were not included. This was done as they were assumed to be of less importance from a cybersecurity point of view. An extra IED was added to control the HV/MV OLTC together with a data flow to the TSO SCADA. Both the TSO and DSO SCADA processes are assumed to interact with a human operator and a database.

The use case is from the OT part of the smart grid. Many OT systems in the industry can be expected to originate from a time when cybersecurity was less relevant than it is today. Because of this, we assume that the use case consists of largely insecure units. There is some support for this in the literature. Staggs et al. [17] report that windmills often transmit command and control messages as clear text. Aloul et al. [18] claim that current smart grid devices do not have the necessary processing power or storage for advanced cryptographic operations. This resulted in template default configurations being used. The template default configurations are set to be insecure, as explained in Section III-A.

With this particular use case, a loss of availability of components is less critical than malicious values or commands. Because of this, we assume the use case to have adequate backup solutions to deal with denial of service incidents. This assumption is supported by Staggs et al. [17], who claim that most windmills are able to continue to produce and transmit power even if communication with the control center is lost.

Threat modeling was performed on the system, as shown in Fig. 5, with template version 1.0.0.784. The threat model generated 334 threats. Of the 334 threats, 73 threats were related to denial of service, 105 were related to elevation of privilege, 30 related to information disclosure, 47 related to repudiation, 63 related to spoofing, and 16 related to tampering (Due to corrections, the version number, threat numbers, and figure deviates from Flå [6]). The model in Fig. 5 can be found on Github.

V. Discussion

In the following, we will discuss briefly the applicability of STRIDE, pros and cons of the Microsoft TMT, and limitations of our Smart Grid template. For a more extensive discussion of the results, see Flå [6].

A. STRIDE-per-interaction and Threat Modeling as a Method for Smart Grid Security

Threat modeling and particularly STRIDE provide a systematic way of analyzing a system for threats. This ensures that threats are less likely to be overlooked. The smart grid is set to become increasingly dependent on ICT components and systems for it to function correctly. The smart grid stands out because of its size, amount of data generated, and the combination of IT and OT systems. Because of this, it may be beneficial to apply or draw inspiration from how the IT industry deals with security. A limitation of STRIDE is that it does not include a method for evaluating the criticality of a threat. Evaluating the criticality of a threat in the smart grid is difficult as it can be expected to depend on the specific use case being studied. It is however potentially useful as it can help structure an otherwise overwhelming number of threats. One possible approach for evaluating criticality that could be adapted to the smart grid is the use of a bug bar [8], an approach that is much used at Microsoft. One can imagine that a set of criteria for various levels of criticality can be used to categorize the threats.

It is sometimes difficult to know how to categorize a threat. For instance, one can argue that a MITM attack is a spoofing attack, as the core of the attack is to pretend to be someone else. A MITM attack can, however, also be a starting point for eavesdropping, modification, injection, and DoS. Another example is the injection of data in an optical fiber cable. One can make the argument that it is either tampering of communication, or spoofing, as one would realistically inject data that appear to originate from a legitimate source. However, these categorization challenges does not impact the ability to identify these threats as part of threat modeling.

B. The Microsoft Threat Modeling Tool

This section discusses the advantages and disadvantages of the Microsoft TMT with regards to threat modeling in the smart grid.

1) Advantages: The tool offers a large degree of freedom when defining templates. It gives the option of freely defining stencils, stencil categories, and stencil properties. This allows for threat modeling that does not need to follow the traditional DFD categories of process, data store, trust boundary, data flow, and external interactor. The same freedom is offered for defining threats, threat categories, and ways of categorizing

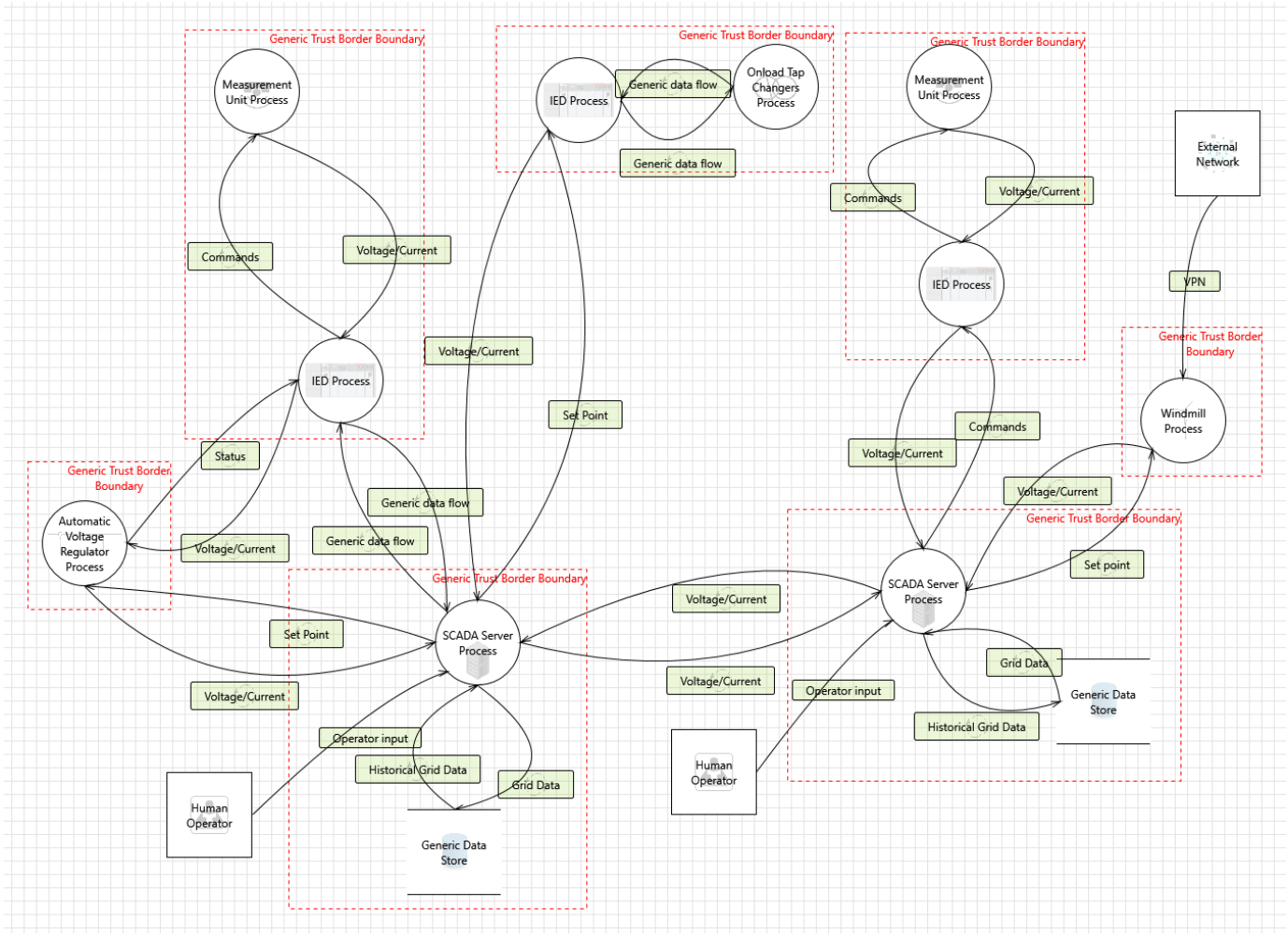

Fig. 5. Modeling the Use Case in TMT

threats. This provides the possibility of deviating from the STRIDE threats if necessary.

The Microsoft TMT provides a structured way to deal with identified threats. For each threat the user can set threat properties, decide whether the threat needs investigation, or provide a justification for the choices being made. The user can then choose to generate an HTML report containing all threats and their status. This report provides good documentation for the state of the security of the system being analyzed.

The Microsoft TMT offers the option of merging templates together. This can allow for convenient expansion of the template in the future. An interesting improvement of this functionality would be to allow merging on a stencil by stencil basis. One can imagine functionality where it is possible to merge just the stencils of interest into another template, along with all threats relating to these stencils.

2) Disadvantages: The tool allows templates to have one level of stencil inheritance. The child stencils inherit the properties of the parent stencil. Only one level of inheritance is allowed. This restriction has not caused inconvenience for the work on the Smart Grid template, but the reason for this limitation is not evident. This limitation may cause inconvenience in future expansions where the number of stencils and level of configuration detail increase.

The tool is not suited for analyzing threats that do not conform to the (source, flow, target)-tuple used in threat generation. An example from this paper is the desire to model forgotten VPN connections. VPN connections put in place by vendors can pose a threat to the smart grid asset owner. To model this scenario, configuring all smart grid process stencils to have a VPN connection by default was attempted. By not specifying a flow in the threat generation logic, the same threat is generated multiple times for each flow. Including a threat of unknown VPN connection in the use case threat modeling would have generated four duplicates for the right SCADA Server process. This problem may become even more relevant as cloud connectivity is expected to increase.

C. Smart Grid Template

The threats in the template have, in many cases, been generalized so that they affect all stencils. Such a generalization may cause the template to generate false threats. By false

threats we mean threats that would likely not be present in real systems. Choosing a narrower scope when generalizing threats would result in fewer included threats, including fewer false threats. A decision was made to accept potential false threats to minimize the risk of excluding potential real threats from the threat modeling.

The child stencils included in the template do not have many configurable properties. Most properties are added at the generic parent level. This is because most threats are generalized, as described in the paragraph above. The purpose of properties is to be used in the threat generation logic. As threats generally have been generalized to the generic level, this is where most properties are located.

The process stencils represent the functional behavior of a part or subsystem of the grid. Traditional threat modeling uses the process stencil to represent running code. A similar approach would be possible for the smart grid but is thought to would have greatly increased the complexity of the threat modeling process. A measurement unit can potentially run a communication process, measurement process, and possibly other processes related to a lightweight operating system. The number of processes can be expected to be greater for more complex template stencils such as a SCADA server or Windmill. Directly linking template processes to ICT components would increase the number of processes in a similar way. The windmill would have had to be broken down into smaller components. Instead, the functional behavior with a focus on communication interfaces was adopted. This allows for processes to represent systems of various abstraction levels. This is one possible approach. In their article on STRIDE for cyber-physical systems, Khan et al. [9] advocate for creating a DFD for each component in the system.

The smart grid template implements a database as the only data store. Other types of data stores that could have been included are cache, memory, and files. All these types of storage are assumed to be included in the processes. Consequently, only larger and more dedicated data store was included in the form of the database stencil. The smart grid template only has a single type of trust boundary. Crossing a trust boundary is a necessary condition for most threats to be included. Other templates, like the default template, includes several different trust boundaries. For our solution, we did not see a need for more than one type of trust boundary.

Four threat properties were included in the template. These are priority, loss of power, difficulties of implementing mitigation, and affected systems.

VI. Conclusion

In this paper we have demonstrated the applicability of threat modeling and the Microsoft TMT to the smart grid domain, and we have offered a template that enables tool support for threat modeling of cyber-security in the smart grid. The template serves as a starting point for others to adapt or expand. Additionally, it can serve as inspiration for developing similar templates for new domains.

References

[1] ICS-CERT, "Cyber-attack against Ukrainian critical infrastructure," IR-ALERT-H-16-056-01, U.S Department of Homeland Security, 2016. [Online]. Available: https://us-cert.cisa.gov/ics/alerts/IR-ALERT-H-16-056-01

[2] J. Slowik, "CRASHOVERRIDE: Reassessing the 2016 ukraine electric power event as a protection-focused attack," Dragos report, 2019. [Online]. Available: https://www.dragos.com/wp-content/uploads/CRASHOVERRIDE.pdf

[3] B. Johnson, D. Caban, M. Krotofil, D. Scali, N. Brubaker, and C. Glyer, "Attackers deploy new ICS attack framework "TRITON" and cause operational disruption to critical infrastructure," https://www.fireeye.com/blog/threat-research/2017/12/attackers-deploy-new-ics-attackframework-triton.html, 2017.

[4] I. A. Tøndel, J. Foros, S. S. Kilskar, P. Hokstad, and M. G. Jaatun, "Interdependencies and reliability in the combined ICT and power system: An overview of current research," *Applied Computing and Informatics*, vol. 14, no. 1, pp. 17–27, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210832716300552

[5] M. G. Jaatun, M. Bartnes, and I. A. Tøndel, *Zebras and Lions: Better Incident Handling Through Improved Cooperation*. Cham: Springer International Publishing, 2016, pp. 129–139. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-49466-1_9

[6] L. H. Flå, "Threat modeling framework for smart grids," Master's thesis, Norwegian University of Science and Technology, 2021.

[7] F. Swiderski and W. Snyder, *Threat Modeling*. Microsoft Press, 2004.

[8] A. Shostack, *Threat Modeling: Designing for Security*. Wiley, 2014.

[9] R. Khan, K. McLaughlin, D. Laverty, and S. Sezer, "Stride-based threat modeling for cyber-physical systems," in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. IEEE, 2017, pp. 1–6.

[10] I. A. Tøndel, M. G. Jaatun, and M. B. Line, "Threat Modeling of AMI," in *"Proceedings of the 7th International Conference on Critical Information Infrastructures Security (CRITIS 2012)"*, 2012.

[11] R. Jiang, R. Lu, Y. Wang, J. Luo, C. Shen, and X. Shen, "Energy-theft detection issues for advanced metering infrastructure in smart grid," *Tsinghua Science and Technology*, vol. 19, no. 2, pp. 105–120, 2014.

[12] X. Liu, P. Zhu, Y. Zhang, and K. Chen, "A collaborative intrusion detection mechanism against false data injection attack in advanced metering infrastructure," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2435–2443, 2015.

[13] H. Suleiman, I. Alqassem, A. Diabat, E. Arnautovic, and D. Svetinovic, "Integrated smart grid systems security threat model," *Information Systems*, vol. 53, pp. 147–160, 2015.

[14] A. Shostack, "Experiences threat modeling at Microsoft," in *Proceedings of the Workshop on Modeling Security (MODSEC08)*. CEUR Workshop Proceedings, 2008. [Online]. Available: http://ceur-ws.org/Vol-413/paper12.pdf

[15] J. Slowik, "Anatomy of an attack: Detecting and defeating CRASHOVERRIDE," Dragos whitepaper, 2018. [Online]. Available: https://www.dragos.com/resource/anatomy-of-an-attack-detecting-and-defeating-crashoverride/

[16] N. Falliere, L. O. Murchu, and E. Chien, "W32. stuxnet dossier," *White paper, Symantec Corp., Security Response*, vol. 5, no. 6, p. 29, 2011.

[17] J. Staggs, D. Ferlemann, and S. Shenoi, "Wind farm security: attack surface, targets, scenarios and mitigation," *International Journal of Critical Infrastructure Protection*, vol. 17, pp. 3–14, 2017.

[18] F. Aloul, A. Al-Ali, R. Al-Dalky, M. Al-Mardini, and W. El-Hajj, "Smart grid security: Threats, vulnerabilities and solutions," *International Journal of Smart Grid and Clean Energy*, vol. 1, no. 1, pp. 1–6, 2012.