Generating Threat Models and Attack Graphs based on the IEC 61850 System Configuration description Language

Engla Rencelj Ling
KTH Royal Institute of Technology
Stockholm, Sweden
englal@kth.se

Mathias Ekstedt
KTH Royal Institute of Technology
Stockholm, Sweden
mekstedt@kth.se

ABSTRACT

Due to our dependency on electricity, it is vital to keep our power systems secure from cyber attacks. However, because power systems are being digitalized and the infrastructure is growing increasingly complicated, it is difficult to gain an overview and secure the entire system. An overview of the potential security vulnerabilities can be achieved with threat modeling. The Meta Attack Language (MAL) is a formalism that enables the development of threat modeling languages that can be used to automatically generate attack graphs and conduct simulations over them. In this article we present the MAL-based language SCL-Lang which has been created based on the System description Configuration Language (SCL) as defined in the IEC 61850 standard. With SCL-Lang one can create threat models of substations based on their SCL files and automatically find information regarding potential cyber attack paths in the substation automation system configuration. This enables structured cyber security analysis for evaluating various design scenarios before implementation.

CCS CONCEPTS

• **Security and privacy**; • **Hardware → Energy distribution**; • **Computing methodologies → Simulation languages**; • **Computer systems organization → Embedded and cyber-physical systems**;

KEYWORDS

Threat Modeling; Attack Graphs; Cyber Security; IEC 61850; System Configuration description Language

ACM Reference Format:
Engla Rencelj Ling and Mathias Ekstedt. 2021. Generating Threat Models and Attack Graphs based on the IEC 61850 System Configuration description Language. In *Proceedings of the 2021 ACM Workshop on Secure and Trustworthy Cyber-physical Systems (SAT-CPS'21), April 28, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3445969.3450421

1 INTRODUCTION

Power system substations are one of the cyber-physical systems that are becoming increasingly automated and intelligent. This digitalization brings new capabilities and increased efficiency for power system operations. However, it also brings a higher exposure to cyber security risks [2]. Another consequence of the digitalization of substations is that many different protocols are developed by different vendors. To solve the problem of interoperability the IEC 61850 standard was developed [4].

The IEC 61850 is an international standard that describes the communication of digitalized power systems. One part of the IEC 61850 standard is the System Configuration description Language (SCL). The SCL consists of four files that are used to describe the communication of a substation in terms of its Intelligent Electronic Devices (IEDs). IEDs are the devices in substations that can perform automation tasks. The four files of SCL can, for example, be used by vendors to share information to their customers, or by customers to share their design requirements to the vendors.

In this paper, we propose the novel approach to use the information in SCL to generate threat models that can be used for cyber security assessments. A threat modeling assessment typically starts with the creation of a model of the system. This can however be a difficult task because the stakeholder may not agree with what and how to model the system [13]. By using the information in the SCL files, we can help this process since the SCL files describes the system as-is. Additionally, with this approach we empower end users with the capability to make cyber security analyses at no additional work since the threat models are based on existing SCL files. To produce the threat models, we will make use of the previous work with the Meta Attack Language in [5], in which domain specific languages can be constructed that are used to automatically generate attack/defense graphs (attack graph for short) from system instance specifications. An attack graph is a formalism used to describe how an attacker may reach a goal by taking different paths. Attack graphs are an extension of attack trees, as popularized by Schneier [10].

2 RELATED WORK

To the best of our knowledge there is no existing work that use information in SCL to create a threat model language to build threat models and automatically generate attack graphs of substations. There are, however, related work of modeling cyber attacks in substations. In [1] supply-chain attacks in substations are modelled. Supply-chain attacks are possible because of vulnerabilities that have been added, with intent, to the products during the supply-chain. One scenario of an attack that illustrates a combination of cyber and physical attacks is presented in [3]. Common to these two papers is that the threat model has been created from attack scenarios and not based on the design of a substation.

The related work of this paper also includes similar efforts to map SCL to languages or other models. In [7] the SCL is mapped

SAT-CPS'21, April 28, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8319-6/21/04.
https://doi.org/10.1145/3445969.3450421

to the enterprise modeling standard Archimate to help stakeholders understand the architecture of the substation. The purpose of that paper is similarly to this one, to utilize the already existing information in SCL. The authors of [9] used the SCL files to create a Human-Machine Interface (HMI). The HMI was created by extracting how the communication and IED specification looked like from the SCL files. There has also been previous work on using SCL in the security domain. In [12] the information in SCL is used to create an intrusion detection system (IDS).

The Meta Attack Language has been used for several domain specific threat modeling languages, e.g. in the vehicles domain [6]. Closest to our language design approach of directly mapping a system design language into MAL without any human intervention is previous work with Amazon's cloud services [11].

3 SYSTEM CONFIGURATION DESCRIPTION LANGUAGE

The System Configuration description Language (SCL) was developed as part of the IEC 61850 standard to describe the automation of substations in terms of Intelligent Electronic Device (IED) communications and configurations. SCL was mainly developed to create a standard for all manufacturers to help with compatibility for engineers as described in the document IEC 61850-6 "Configuration description language for communication in electrical substations related to IEDs" [4]. There are four files that can be generated with SCL. These files are the System Specification Description (SSD) that describes an entire substation system, which may consist of several substations, the Substation Configuration Description (SCD) that describes one substation, the IED Capability Description (ICD) that includes all IEDs that are in the substation, and the Configured IED Description (CID) that describes a specific IED.

According to SCL, there are two objects that are central for the automation system communication in substations. These are the logical node and the access point. A logical node is the key object with which the automation is constructed. Logical nodes are interfaces for encoding instructions of what to execute, for example, opening a circuit breaker. A logical node can also be used to send measurement data to execute a task automatically by combining multiple logical nodes into one function. The access point is an interface for communication within the substation. The interface can either be a physical port or an IP address.

4 META ATTACK LANGUAGE

Threat models are produced to identify potential threats and weaknesses of a system. There are different ways of how to represent threat models. For example, threat models can be mathematical or illustrated as flow diagrams. Common to all these methods is that the threat model has to be reproduced for every new system or if there are any changes made to the system. Meta Attack Language (MAL) aims to aid in this step by describing the rules of how to produce a threat model for a given domain [5]. The framework MAL is used to create threat modeling languages. The threat modeling language is used to create threat model instances including attack/defense graphs describing different attack paths over the modeled system. The paths are the different possible attacks, and each step can be associated with a time and likelihood of success. MAL defines five main constructs, these are the system assets, asset associations, attack steps, defense steps, and attack/defense step dependencies. The former two describe the system configuration and the latter three form the attack graph.

Figure 2 shows a small example of a MAL specification. The language consists of three assets, `Computer`, `Secret` and `Password`. The last part of the specification describes their associations. The `Secret` is stored on the `Computer` and the `Computer` is protected by the `Password`. Each asset of the associations has a role that defines how the asset is used in the association. Thus far, MAL follows common conceptual system modeling languages, such as, the Unified modeling Language (UML).

The assets have different attack steps related to them, i.e. *login*, *find* and *obtain*. The asset `Password` has the attack step *obtain*, which leads to the attack step *login* on asset `Computer`, if the two assets are associated. To specify that an attack step is depending on an attack step in another asset, the syntax `role.attackstep` is used.

The attack steps in the example language are prefixed with the symbol |. This means that the attack step is of type "OR" and can be reached without dependencies on any other attack step. Other types of attack steps in MAL are "AND" depicted by & and "EXISTS" denoted by E. The "AND" type requires that all previous, parent, attack steps have been reached successfully before the current one. The "EXISTS" type is a logic check if a certain asset exists or not. For instance, one can have an attack step with type "EXISTS" for an asset that is `Firewall`. If the asset `Firewall` exists, the attack step cannot be reached. Because of this logic, the EXISTS attack steps can be used to model defenses for a system.

In this example, in order to get the `Secret`, the attacker must first *obtain* the `Password`, *login* to the `Computer` and finally *find* the `Secret`. Based on this threat model language, one can generate models of the system and automatically generate attack graphs.

The threat model language is compiled with the MAL compiler[1] and the simulated attacks are generated with securiCAD Professional version 1.6.1[2]. All information regarding the tools and development of MAL are accessible from its official website[3].

5 THREAT MODEL LANGUAGE GENERATION

The first step of creating a MAL specification is to identify the assets and associations and the next step is to define the attack steps. These two steps are described in the following two subsections. The domain specific threat model language presented in this article is referred to as SCL-Lang. SCL-Lang was developed by studying the SCL in detail and translating the SCL object model as seen in Figure 1 to MAL. Whenever the translation was not clear from the SCL specification, experts in the industry were consulted. Then, attack scenarios were created to illustrate if SCL-Lang could capture them as we had intended and intuitively expected. The accuracy of the simulation results was evaluated together with the industry experts to see if the results mimic a real-life scenario.

The full MAL specification of SCL-Lang can be found in the MAL languages Git repository[4].

[1] https://github.com/mal-lang/malcompiler [Accessed 18 Feb 2021]
[2] https://www.foreseeti.com/securicad [Accessed 18 Feb 2021]
[3] http://mal-lang.org/ [Accessed 18 Feb 2021]
[4] https://github.com/mal-lang/SCL-Lang [Accessed 18 Feb 2021]

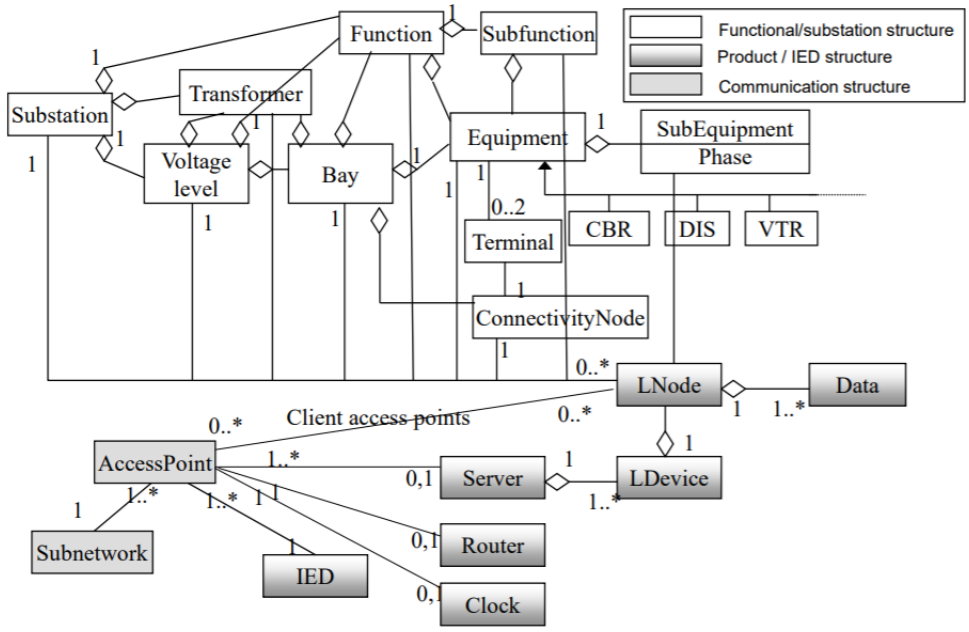

Figure 1: IEC 61850 SCL Object Model [4].

```
category System {
    asset Computer {
        | login
            -> secret.find
    }
    asset Secret {
        | find
    }
    asset Password {
        | obtain
            -> computer.login
    }
}

associations {
    Secret [secret] * <-- Stored -->
        1 [computer] Computer
    Password [password] 1 <-- Protect -->
        1 [computer] Computer
}
```

Figure 2: MAL Specification of SimpleLang.

5.1 Assets and Associations

The SCL object model has been used to create SCL-Lang by using the classes and relations described in IEC 61850-6 [4]. Note that the threat models generated in this paper are solely based on the assets and relations as defined by the SCL. Therefore, components such as firewall and HMI (Human Machine Interface) are not included. Further development of SCL-Lang to model the entire scope of a substation is part of future work.

There are some exceptions to a direct translation from the SCL object model in Figure 1, to SCL-Lang. First, CBR (circuit breaker), DIS (disconnector), and VTR (voltage transformer) are examples of `Equipment`. These examples have not been added to the MAL specification. Next, at the time of writing this paper, the current MAL compiler does not support a multiplicity of "2", in this case the multiplicity was replaced with "many". Lastly, in some relations, the multiplicity is not specified and in this case the multiplicity was specified as "many" to allow modeling of any scenario.

5.2 Attack Steps

There are many different possible attacks on substations and to limit the scope of SCL-Lang in this article, tactics of attacks has been modeled instead of each individual attack that belong to the tactics. These common attack tactics are inspired by the ATT&CK for Industrial Control Systems developed by MITRE [8]. The ATT&CK Matrix is an open database of known common cyber attacks categorized by tactics. The tactics instead of individual techniques are used because SCL-Lang serves as a foundation and a more extensive language can be built based on it as part of future work.

The attack steps used in this article are *access*, *communicate*, *execution*, *impact* and *hasRouter*. These attack steps were chosen because they are required to illustrate use cases of attack scenarios that can occur within the substation as described by the SCL files. Not all of the ATT&CK matrix' 11 tactics were included to keep the complexity down in the first version of SCL-Lang. *access* is the MITRE tactics Initial Access, which is described as an attack trying to get into the network. *execution* and *impact* are also taken from MITRE and represent when the attacker runs malicious commands or when they maliciously alter data, respectively. For example, the attack Denial of Service (DoS) is included in this tactic. The attack steps *communicate* and *hasRouter* have been added outside of MITRE tactics scope. *communicate* was added to allow the modeling of network communication and *hasRouter* was added to model if the routing function is enabled on the IED or not.

5.2.1 access. The *access* attack step indicates that assets within the system have been reached and that the attacker can, for example, communicate or execute certain commands from that point. We assume that *access* is the entry point of an attacker.

5.2.2 communicate. In an IEC 61850 substation, the assets *communicate* through `AccessPoints`, which can be used by an attacker to move between the assets. The communication can be either over TCP/IP or within LANs depending on which protocol is used. For instance, `Clock` is not considered to have an entry point and does not have an *access* attack step. However, it is possible to use the `Clock` to *communicate* within the substation and therefore `Clock` has the *communicate* attack step.

5.2.3 execution. The attack step *execution* can be defined as to put a plan of actions into effect. In substations, instructions of execution are sent via `LogicalNodes`. A `LogicalNode` can be used to, for example, execute the opening of a circuit breaker. A circuit breaker is an automatic switch that can be opened to stop the flow of electricity. If an attacker can *impact* a `LogicalNode`, they can maliciously execute the opening of a circuit breaker and cause disruption in the power system.

5.2.4 impact. The *impact* attack step indicates that an *impact*, a read or write, has occurred. This could be either maliciously altering `Data` or a `LogicalNode`. Because measurements are sent and actions are taken by sending specific `LogicalNodes`, we assume that reaching this attack step normally is the final goal of an attacker.

When the *impact* attack step is reached on a logical node it is possible to, as explained in the *execution* attack step, maliciously execute the opening of a circuit breaker and cause disruption in the power system. The attacker can also use *impact* to alter measurement data and cause disruptions to the power system.

5.2.5 hasRouter. According to the UML description in [4] the class `AccessPoint` has a boolean attribute called *router*. This indicates if the `IED` connected to the `AccessPoint` has a routing function or not. If the `IED` has a routing function, it enables *communication* between different `SubNetworks`.

In terms of MAL, this means that the asset `AccessPoint` can either have the asset `Router` existing or not. The attack step to check if the `Router` exists or not is of type EXISTS. If the `Router` exists then the attacker can reach the attack step and *communicate* across `SubNetworks`. However, if no `Router` exists the attacker will not succeed to *communicate*. This is because the attack step *communicate* is of type "AND", represented with an & sign. The "AND" means that *communicate* can only be reached if *hasRouter* is reached first.

6 ATTACK SCENARIOS

To illustrate the capabilities of SCL-Lang, five attack scenarios are modelled. These threat models have been put together with the software SecuriCAD as introduced in section 4. In SecuriCAD the attacks have been simulated automatically to generate attack graphs for each use case. The attack graph shows the shortest path as the most likely path of an attacker. The scenarios chosen for this paper showcase potential cyber security attacks in substations and the attack graphs suggests where the substation is the most vulnerable.

Attack scenarios 1, 4 and 5 are examples of successful attacks and attack scenarios 2 and 3 are unsuccessful, as expected. The attack scenarios of successful attacks include figures of both the model and the attack path from the SecuriCAD simulation. The starred asset indicates the end goal of an attack, as chosen in the specific example, and the attack begins where the `Attacker` is connected.

6.1 Attack Scenario 1: Attacking LogicalNode from SubNetwork

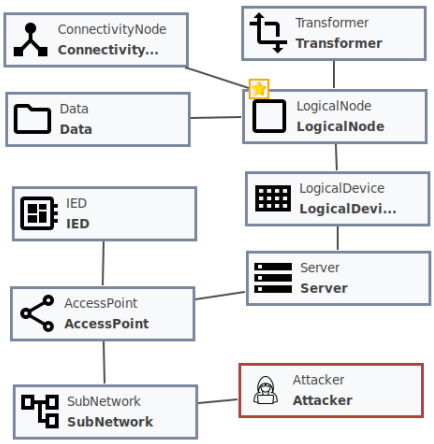

Figure 3: Model of Attack Scenario 1.

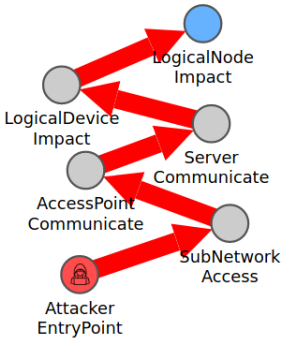

Figure 4: Resulting Attack Graph of Attack Scenario 1.

When an attacker has access on a `SubNetwork` they can *communicate* via the `AccessPoint` to finally make an *impact* on a `LogicalNode` as seen in Figure 3. The automatically generated attack path taken can be seen in Figure 4.

6.2 Attack Scenario 2: Attacking LogicalNode1 from LogicalNode2

In attack scenario 2, which is an extension of scenario 1, there are two `LogicalNodes` in two different `LogicalDevices`, `Servers` and `SubNetworks`. It is not possible for the attacker to *impact* the `LogicalNode2` and then in some way *impact* `LogicalNode1`. This is because there is no attack step from a `LogicalNode` to a

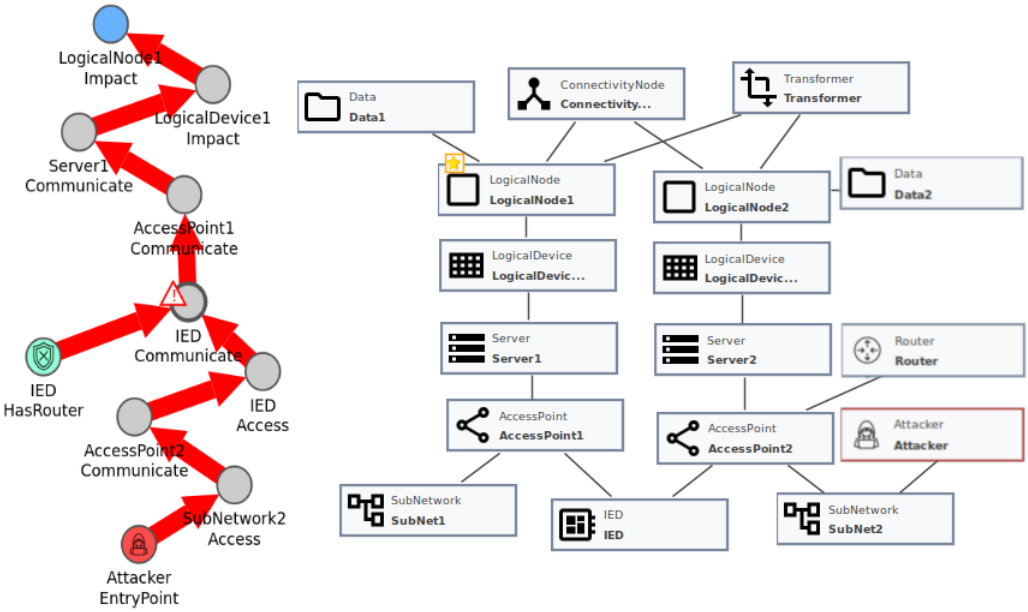

Figure 5: Resulting Attack Graph and Model of Attack Scenario 4.

`LogicalDevice`. The attack is not successful and stops instantly at the entry point.

6.3 Attack Scenario 3: Attacking LogicalNode1 from SubNet2

Similar to attack scenario 2, there are two `LogicalNodes` in two different `LogicalDevices`, `Servers` and `SubNetworks`. However, the attacker's starting point is `SubNet2`. This means that the attack potentially could make *impact* on the `LogicalNode`, but there is no connection between `SubNet2` and `LogicalNode1`. This is because the network is segmented into two different `SubNetworks` and no `Router` exists to *communicate* traffic in between. Therefore this attack will not be successful.

6.4 Attack Scenario 4: Attacking LogicalNode1 from SubNet2 when SubNet1 and SubNet2 are on the same IED and a router exists

An `IED` of a substation can have a routing function which enables communication between different `SubNetworks`. This routing function is the same as the standard definition of routing in TCP/IP networks which means that communication can occur between `SubNetworks`. When this routing function exists an attacker can make *impact* on a `LogicalNode` in a different `SubNetwork` than the one that they accessed the substation with. We assume that when the routing function is enabled, a valid route between the subnetworks exists as well.

As seen in Figure 5, there is only one `IED` in this scenario but two `Servers`. This means that the `IED` must support multiple `Servers`, which may not always be the case. A `Server` is a container where `LogicalDevices` are stored so that they can be accessed by other external `SubNetworks`. The `Router` exists in this scenario, as seen connected to `AccessPoint2`, and therefore there is no defense in terms of segmentation of the two `SubNetworks`. The attacker can move between the `SubNetworks` and *impact* `LogicalNode1` in `SubNet1` from `SubNet2`. Outside of the scope of SCL is the possibility of having an external route function by connecting an external networking device to an access point. This could be a standard router, also known as a layer 3 switch. In this case the outcome of this scenario would be the same, because of the route in between `SubNetworks` the attacker can successfully make *impact* on a `LogicalNode` in `SubNet1` from `SubNet2`.

6.5 Attack Scenario 5: Attacking LogicalNode1 from SubNet2

A `Server` can be connected by more than one `AccessPoint`. This means that two different `SubNetworks` can access the same `Server` via two different `AccessPoints`. An attacker can therefore communicate between two different `SubNetworks` via the `Server` as seen in Figure 6. This makes it possible for an attack to make an *impact* on `LogicalNode1` from `Subnet2`.

It is not possible to derive from Figure 6, which `IED` the `Server` is located on and for this scenario it does not influence the result. This is because regardless on which IED the server is contained on; it can still be accessed by both `SubNets`.

7 DISCUSSION AND FUTURE WORK

The work presented in this paper describes how SCL can be translated to threat models to run attack simulations. This makes it possible to make a security evaluation of a substation by using already existing configuration files without adding any additional work. The paper includes five attack scenarios to showcase SCLLang. Future work could also be to build a solution for generating a threat model based on an SCD file automatically. This process can

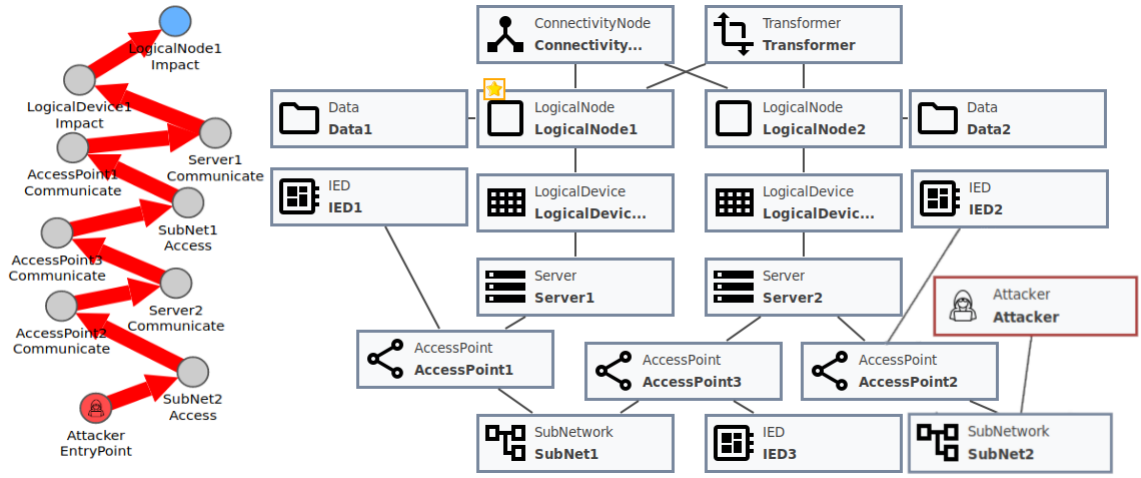

Figure 6: Resulting Attack Graph and Model of Attack Scenario 5.

be automated by programming a parser that would input the SCD file and output a threat model.

A typical substation does not have segmentation or security policies in place within the operational parts. However, it is possible to segment the network with subnetworks and in that way prevent an attacker from moving across the entire system. As far as SCL is concerned, this is the only available protection against cyber attacks within a substation except the addition of authentication strength as a parameter. This makes it possible for a user to get an overview of the potential attack steps taken by an attacker that has gained access to one of the substations subnetworks as portrayed by the attack scenarios.

From the attack scenarios we have been able to see that attacks between subnets are successful if a router exists with a valid route in between the subnets as seen in Figure 5. The attack scenarios also show that attacks can occur via a server if the server is connected to two different subnetworks as seen in Figure 6. SCL-Lang and the results from the attack scenarios have been evaluated in discussions with experts in the energy systems domain. It is the expectation that future work will include more evaluation and validation once the language is further developed and includes more information than that from IEC 61850 and Mitre ICS Matrix.

Planned future work is to extend SCL-Lang to include more assets, such as, Remote Terminal Units (RTUs) and firewalls to enable modelling of an entire substation and not only the assets as described in SCL.

ACKNOWLEDGMENTS

This research was funded by Swedish Centre for Smart Grids and Energy Storage (SweGRIDS).

REFERENCES

[1] O. Duman, M. Ghafouri, M. Kassouf, R. Atallah, L. Wang, and M. Debbabi. 2019. Modeling Supply Chain Attacks in IEC 61850 Substations. In *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. 1–6.

[2] Muhammed Zekeriya Gunduz and Resul Das. 2020. Cyber-security on smart grid: Threats and potential solutions. *Computer Networks* 169 (2020), 107094. https://doi.org/10.1016/j.comnet.2019.107094

[3] P. J. Hawrylak, M. Haney, M. Papa, and J. Hale. 2012. Using hybrid attack graphs to model cyber-physical attacks in the Smart Grid. In *2012 5th International Symposium on Resilient Control Systems*. 161–164.

[4] IEC. 2018. Communication networks and systems for power utility automation - Part 6: Configuration description language for communication in electrical substations related to IEDs IEC 61850-6.

[5] Pontus Johnson, Robert Lagerström, and Mathias Ekstedt. 2018. A Meta Language for Threat Modeling and Attack Simulations. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*. ACM, 38.

[6] Sotirios Katsikeas, Pontus Johnson, Simon Hacks, and Robert Lagerström. 2019. Probabilistic Modeling and Simulation of Vehicular Cyber Attacks : An Application of the Meta Attack Language. In *Proceedings of the 5th International Conference on Information Systems Security and Privacy*.

[7] J. Konig, K. Zhu, L. Nordstrom, M. Ekstedt, and R. Lagerstrom. 2010. Mapping the Substation Configuration Language of IEC 61850 to ArchiMate. In *2010 14th IEEE International Enterprise Distributed Object Computing Conference Workshops*. 60–68. https://doi.org/10.1109/EDOCW.2010.35

[8] MITRE. 2020. ATT&CK® for Industrial Control Systems. [online] Available at: https://collaborate.mitre.org/attackics/index.php [Accessed 19 Oct 2020].

[9] Seong-Jeong Rim, Sheng-Wu Zeng, and Seung Lee. 2009. Development of an Intelligent Station HMI in IEC 61850 Based Substation. *Journal of Electrical Engineering and Technology* 4 (2009), 13–18. https://doi.org/10.5370/JEET.2009.4.1.013

[10] Bruce Schneier. 1999. Attack Trees. *Dr. Dobb's Journal of Software Tools* 24 (1999), 21 – 29.

[11] Amandeep Singh Virdi. 2018. *AWSLang: Probabilistic Threat Modelling of the Amazon Web Services environment.* Master's thesis. KTH Royal Institute of Technology.

[12] Y. Yang, H. Xu, L. Gao, Y. Yuan, K. McLaughlin, and S. Sezer. 2017. Multidimensional Intrusion Detection System for IEC 61850-Based SCADA Networks. *IEEE Transactions on Power Delivery* 32, 2 (2017), 1068–1078.

[13] Koen Yskout, Thomas Heyman, Dimitri Van Landuyt, Laurens Sion, Kim Wuyts, and Wouter Joosen. 2020. Threat Modeling: From Infancy to Maturity *(ICSE-NIER '20)*. Association for Computing Machinery, New York, NY, USA, 9–12. https://doi.org/10.1145/3377816.3381741