2019 IEEE 23rd International Enterprise Distributed Object Computing Conference (EDOC)

DOI 10.1109/EDOC.2019.0002

Creating Meta Attack Language Instances using ArchiMate: Applied to Electric Power and Energy System Cases

Simon Hacks*‡, Alexander Hacks†, Sotirios Katsikeas‡, Benedikt Klaer§ and Robert Lagerström‡
*Research Group Software Construction, RWTH Aachen University, Aachen, Germany
hacks@swc.rwth-aachen.de
† Universität Duisburg-Essen, Duisburg, Germany
alexander.hacks@uni-due.de
‡Network and Systems Engineering, KTH Royal Institute of Technology, Stockholm, Sweden
{shacks|sotkat|robertl}@kth.se
§Institute for High Voltage Technology, RWTH Aachen University, Aachen, Germany
klaer@ifht.rwth-aachen.de

***Abstract*—Cyber-attacks on power assets can have disastrous consequences for individuals, regions, and whole nations. In order to respond to these threats, the assessment of power grids' and plants' cyber security can foster a higher degree of safety for the whole infrastructure dependent on power. Hitherto, we propose the use of attack simulations based on system architecture models. To reduce the effort of creating new attack graphs for each system of a given type, domain-specific attack languages may be employed. They codify common attack logics of the considered domain.**

Previously, MAL (the Meta Attack Language) was proposed, which serves as a framework to develop domain specific attack languages. We extend the tool set of MAL by developing an approach to model security domains in ArchiMate notation. Next, those models are used to create a MAL instance, which reflects the concepts modeled in ArchiMate. These instances serve as input to simulate attacks on certain systems. To show the applicability of our approach, we conduct two case studies in the power domain. On the one hand, we model a thermal power plant and possible attacks on it. On the other hand, we use the attack on the Ukrainian power grid for our case study.

***Index Terms*—Meta Attack Language, threat modeling, attack simulation, Domain Specific Language, ArchiMate, electric power and energy systems**

Acknowledgement

This project has received funding from the European Unions H2020 research and innovation programme under the Grant Agreement No. 832907.

1. Introduction

Cyber-attacks on power assets can have disastrous consequences for individuals, regions, and whole nations as proven by the recent deliberate disruptions of electrical power and energy systems [1], [2]. Attackers can exploit malicious code to manipulate the controls of power grids, energy providers, and other critical infrastructure [3], [4]. Those manipulations can result in real-world catastrophic physical damage, like major power outage or city-wide disruptions of any service that requires electric power [1], [2], [5]. In order to respond to these threats, the assessment of power grids' and plants' cyber security can foster a higher degree of safety for the whole infrastructure dependent on electric power.

However, assessing the cyber security of power grids and power plants is difficult. In order to identify vulnerabilities, the security-relevant parts of the system must be first understood, and all potential attacks have to be identified [6]. There are three challenges related to these needs: First, it is challenging to identify all relevant security properties of a system. Second, it might be difficult to collect this information. Last, the collected information needs to be processed to uncover all weaknesses that can be exploited by an attacker.

Hitherto, we have proposed the use of attack simulations based on system architecture models (e.g., [7], [8]) to support these challenging tasks. Our approaches facilitate a model of the system and simulate cyber-attacks in order to identify the greatest weaknesses. This can be imagined as the execution of a great number of parallel virtual penetration tests. Such an attack simulation tool enables the security assessor to focus on the collection of the information about the system required for the simulations, since the simulation tackles the first and third challenges.

As the previous approaches rely on a static implementation, we propose the use of MAL (the Meta Attack Language) [9]. This framework for domain-specific languages (DSLs) defines which information about a system is required and specifies the generic attack logic. Since MAL is a meta language (i.e. the set of rules that should be used to create a new DSL), no particular domain of interest is represented. Therefore, this work aims to create and evaluate a MAL-based DSL for simulation of known cyber-attacks on power

grids and power plants.

So far, MAL-based DSLs are very similar to program code. This may hinder security experts, who are not familiar with such a way of modeling, to adapt to our approach. Additionally, this impedes the reuse of existing models like EA (enterprise architecture) models, which can serve as input for the assets of MAL.

To overcome these shortcomings, we propose to use ArchiMate [10] for modeling instances of MAL. This offers three advantages: First, there exists already tool support for visual modeling, e.g., the open source tool Archi[1]. Second, researchers have already elaborated on methods to model security in ArchiMate [11]–[13], which can be reused in our case. Third, EA models containing IT assets and modeled in ArchiMate can serve as input avoiding the need to model them twice. To realize the first two advantages, we have first to align the way security can be modeled in ArchiMate with the way it is expected in MAL. Therefore, we formulate the following research question:

RQ 1. How can established security modeling approaches in ArchiMate be aligned to the security modeling of MAL?

When this question is answered, we have to think about, how to transfer the modeled information from the ArchiMate model to a proper instance of MAL. Accordingly, we formulate the following technical question:

RQ 2. How can ArchiMate models be transformed to a MAL instance?

Next, we will present related work, before we detail the facilitated research method in section 3. In section 4, we show how we aligned existing ArchiMate security modeling research with the MAL and how we transformed the ArchiMate model to a MAL instance. To give a deeper understanding of this, we conducted two case studies, where domain experts modeled attacks on thermal power plants (cf. section 5.1) and on power grids (cf. section 5.2). Those models were transformed to an instance of MAL, and subsequently, we created concrete instances of both models, and, then, we performed attack simulations. The results of these simulations are discussed with the domain experts in section 6, which is followed by our conclusion.

2. Related Work

Our work relates to three domains of previous work: model-driven security engineering, attack/defense graphs, and security modeling in ArchiMate. First, there are domain-specific languages for security analysis of software and system models defined in the domain of model-driven security engineering. Second, attack/defense graphs are applied as formalism for its analysis. Last, security modeling in ArchiMate acts as input for our security modeling.

Model-driven security engineering induced a large number of domain-specific languages [14]–[17]. These languages facilitate the capability to model a system's design according to components and their interaction. Furthermore, they also enable to model security properties such as constraints, requirements, or threats. They are built upon different formalisms and logics like the Unified Modeling Language and the Object Constraint Language. Model checking and searches for constraint violations are applied to conduct security analysis in these languages.

Apart from the languages mentioned before, some security languages also exist, which however do not support automated analysis purposes [18], [19]. They offer only the capability to model security relevant properties. An analysis needs to be conducted manually without any further support.

The concept of attack trees is commonly based on the work of Bruce Schneier [20], [21]. They were formalized by Mauw & Oostdijk [22] and extended to include defenses by Kordy et al. [23]. As summarized in [24], there are several approaches elaborating on attack graphs, e.g. [25], [26]. Elaborating on the theoretical achievements of the previously presented papers, different tools using attack graphs were developed. These tools mostly build up on collecting information about existing system or infrastructure and automatically create attack graphs based on this information. For example, the TVA tool [27] models security conditions in networks and uses a database of exploits as transitions between these security conditions.

A sub domain of attack graph modeling are probabilistic attack graphs, e.g., facilitating Bayesian networks. In [28], the authors apply the TVA-tool to generate attack graphs, transform them to dynamic Bayesian networks, and enrich them with probabilities using CVSS (Common Vulnerability Scoring System) scores. CVSS is also utilized by [29] to model uncertainties in the attack structure, attacker's actions and alerts triggering.

The approaches of attack graphs and system modeling are united in our previous works: e.g., P2CySeMoL [8], and securiCAD [7]. The central idea of these works is to automatically generate probabilistic attack graphs from a given system specification. The attack graph serves as an inference engine that produces predictive security analysis results from the system model. This is also done in ArchiMate itself. For example, Manzur et al. [13] enhanced ArchiMate to xArchiMate, which is capable to support the simulation, experimentation and analysis of EAs. Therefore, they enrich the ArchiMate meta-model by adding behavioral information, adding new element types, and removing element types that have a meaningful behavior.

Several domain specific languages have been built in MAL serving as good examples of the capabilities a MAL-based DSL has and how it can be developed. One example is vehicleLang [30], which is a DSL for modeling cyber-attacks on modern vehicles. Another example is coreLang, which as its name suggests is a core modeling language that contains the most common IT entities and attack steps. coreLang is included in the presentation of MAL [9].

Lastly, we refer to some related work from the domain of security modeling in ArchiMate. Grandry et al. [11] present a mapping of the concepts of an information system security risk management to the ArchiMate enterprise architecture modeling language. Further, they illustrate the application

1. https://www.archimatetool.com/

of the proposed approach through the handling of a lab case. This work is extended by Band et al. [12], who demonstrates the linkage between ArchiMate and broadly accepted risk and security concepts. Therefore, they discuss security modeling in ArchiMate along the context of different frameworks like the TOGAF framework, the COSO ERM framework, the SABSA framework, and The Open Group Risk Taxonomy standard. They identify that the majority of common risk and security concepts can be realized in ArchiMate by either reusing the ArchiMate standard or defining risk and security-specific specializations of ArchiMate concepts.

Not directly related to our research itself are works, which elaborate on creating reference architectures in the power domain, like smart grids. Those works are related to the domain that we use as case study. For example, Jiang et al. [31] proposes a DSL and a repository to represent power grids and related IT components that control the power grid. Further, the SGAM (Smart Grid Architecture Model) [32] provides a technical reference architecture, which represents the functional information data flows between the main domains of smart grids and integrates several systems and subsystems architectures. Additionally, SGAM includes a mapping from its concepts to the concepts of ArchiMate.

To summarize, existing research is based on modelling security within ArchiMate. However, research on using ArchiMate models appear very rarely. Therefore, we provide a means to use ArchiMate models to be transformed to MAL that will allow to integrate those reference models modeled in ArchiMate into the existing MAL environment so that those can be used for analysis. We do not aim to propose a further reference architecture for (smart) power grids.

3. Research Method

DSR (Design Science Research) is a widely applied and accepted means for developing artifacts in IS (information systems) research. It offers a systematic structure for developing artifacts, such as constructs, models, methods, or instances [33]. As our research objective indicates the development of an artifact, the application of a DSR is appropriate. We stick to the approach of Peffers et al. [34], which splits the problem up into six single steps and two possible feedback loops:

1 **Identify Problem & Motivate:** Power grids and power plants are under attack, leading to partially disastrous consequences. Therefore, it is necessary to harden the infrastructure to be more resistant towards cyber-attacks. This can inter alia be achieved by assessing abstract models of the infrastructure under attack. So far, we have proposed MAL as a tool to provide an environment for security assessors including already known attacks on assets. However, to provide this environment it is necessary to be aligned with MAL's DSL. Furthermore, the DSL impedes the reuse of existing models like EA models.

2 **Define Objectives:** To tackle the previously stated problems, we want to develop a solution, which allows security experts to model threats on assets and their connections visually, so that there is no need to learn MAL's DSL directly. Additionally, we want to reuse existing EA models to avoid unnecessary effort for creating assets and their structure twice.

3 **Design & Development:** As foundation for our modeling, we rely on ArchiMate, since it is widespread and accepted [35], open source [10], and provides an open source tool support called Archi. Additionally, ArchiMate offers a well-documented, XML-based exchange format [36]. As we opt for ArchiMate as a modeling tool, we can reuse existing research on modeling security issues in ArchiMate. More concretely, we facilitate the mapping of Grandry et al. [11] to model threats and events on our assets. When the modeling is done, we transform the ArchiMate model to a MAL-based instance of DSL. The overall process is sketched in Figure 1.

4&5 **Demonstration & Evaluation:** To demonstrate our approach, we conduct two case studies. First, we model the components of a thermal power plant and their related threats and events. This is transformed to a MAL instance, which then serves as a base to model a concrete thermal power plant. Second, we model a power grid based on the Ukrainian scenario [1]. The evaluation of both case studies is two-fold. On the one hand, we define test cases, which ensure that the modeled attacks are present in the created MAL instance. On the other hand, we conduct simulations on the concrete instances of the power plant and the power grid. The results of the simulations are then discussed with domain experts in the related field.

6 **Communication:** The research is communicated by the publication of the paper itself and presentation at the conference.

4. Threat Modeling in ArchiMate

Next, we tackle our first research question and present how we align established security modeling in ArchiMate with the security modeling in MAL. Therefore, we first sketch, the way security is modeled in MAL.

First, a DSL created with MAL contains all the main elements that are found on the domain under study. Those are called `assets` in MAL. The assets contain `attack steps`, which represent the actual attacks/threats that can happen on them. An attack step can be connected with one or more following attack steps to create an attack path. Those are used to create attack graphs which are facilitated when the simulation is run. Assets also have `associations` between each other which are used for the creation of the model. Inheritance between assets is also possible and each child asset inherits all the attack steps of the parent asset. Finally, the assets can be organized into categories and probability distributions can be assigned to

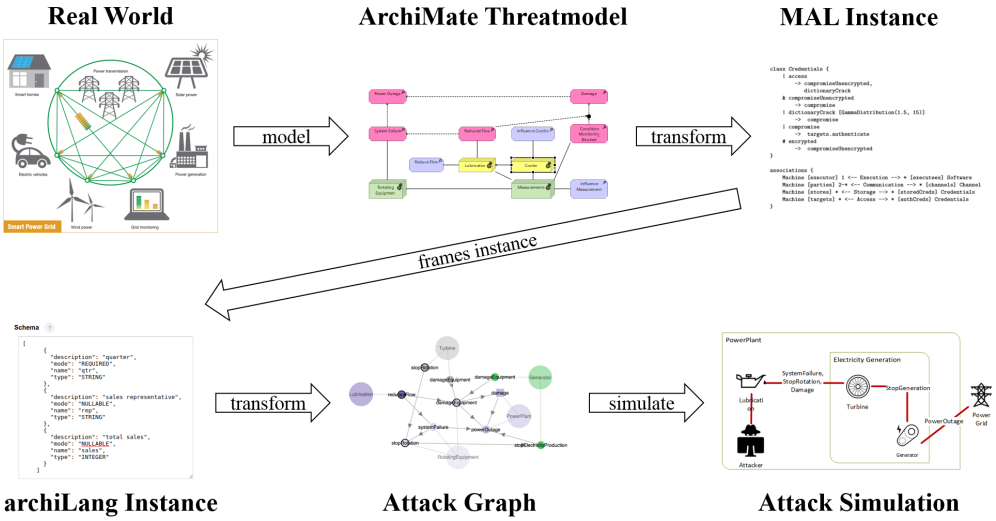

Figure 1. Overall transformation process – From the real world to the simulation

the attack steps in order to represent the effort needed to complete the related attack step.

Next, a short example on how a MAL-based DSL will look like follows. On this example, which could be a snippet of the complete DSL, we can see that attack steps on three assets are modeled. We can then see how the attack steps are connected with each other, for example if an attacker achieves `blockingOperation`, she is then able to reach `overspeed` on `Turbine` and as a result finally lead to `plantDamage` and `powerOutage` on the power plant. In the last lines of the example the associations between the assets are defined.

```
category PowerPlantAssets {
  ...
  asset PowerPlant extends Facility
  {
    | plantDamage
      -> powerOutage
    | powerOutage
      -> city.blackout
  }
  asset Turbine extends RotatingEquipment
  {
    | systemFailure
      -> plant.powerOutage
    | overspeed
      -> systemFailure,
         plant.plantDamage
  }
  asset Controller extends Equipment
  {
    | closeValves
      -> controlledSystem.shutDown
    | reduceFlow
      -> controlledSystem.materialFailure
    | blockingOperation
      -> rotatingEquipment.overspeed
    | influenceMeasurement
      -> controlledSystem.manipulate
    }
  ...
}

associations {
  PowerPlant  [plant]
      1 <-- ComprisedOf --> *
      [controllers]   Controller
  PowerPlant  [plant]
      1 <-- ComprisedOf --> *
      [equipment] RotatingEquipment
  Controller  [controller]
      1 <-- Controls --> 1
      [equipment] RotatingEquipment
  Controller  [controller]
      1 <-- Controls --> *
      [system]  ControlledSystem
}
```

To summarize, MAL follows a simple approach to model security on assets. In contrast, other approaches are often more complex (e.g., the ISSRM (Information System Security Risk Management) [37]). Consequently, we can reduce the complexity when aligning MAL and existing ways to model security in ArchiMate. We can achieve this either directly in the modeling or in the translation from the ArchiMate model to the MAL instance.

We decided to keep the modeling as simple as possible, supporting the modeler focusing on security aspects and not on modeling aspects. Therefore, we use a `Threat`-element, which represents a single attack step in MAL. According to Grandry et al. [11], we model this as an `Assessment` in ArchiMate and add a property `Type` with the value `Threat` to identify it as such.

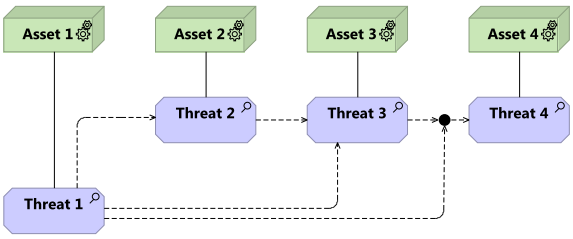

Figure 2. Threat Modeling in ArchiMate

To relate `Threats` to each other, we facilitate the `Influence` relation of ArchiMate, which expresses that reaching one `Threat` allows to conduct the influenced `Threat`. Figure 2 sketches three different options of relating `Threats` to each other. On the one hand, there is a simple one-to-one relation between *Threat 1* and *Threat 2* that describes that an attacker reaching *Threat 1* is able to elaborate on *Threat 2* next. On the other hand, there are `Threats`, which have several preceding `Threats`. For example, *Threat 1* and *Threat 2* precede *Threat 3* symbolized by two `Influence` relations approaching *Threat 3*. This construct is translated to an `OR` in MAL, which means that it is sufficient for the attacker to reach one of the preceding `Threats` to elaborate on the next `Threat`. An `AND` relation, which describes that all preceding `Threats` need to be owned by the attacker, is modeled with a `Junction` in ArchiMate. This is visualized in Figure 2 by the relations between *Threat 1*, *Threat 3*, and *Threat 4*. Therefore, the outgoing relations of *Threat 1* and *Threat 3* are united by a `Junction` before approaching *Threat 4*.

Further, ISSRM differentiates `Business Asset` and `IS Asset`, which are related to a `Threat`. Grandry et al. represent assets by active structure elements in ArchiMate [11]. However, MAL does not differentiate between different assets and, therefore, we map all active structure elements to assets in our MAL instance.

The assets can also be related to each other. We can differentiate basically three types of relations between assets: First, a `Specialization` in ArchiMate describes an inheritance relation between two assets, where one asset adds further functionality to another (abstract) asset according to the inheritance definition in UML (Unified Modeling Language) [38]. We simply translate this to a `Specialization` in MAL as well. Second, there are relations present in ArchiMate, which describe inclusion relationships (`Composition` and `Aggregation`). Those relations are also in accordance with UML [38] and are transformed to `Association` in MAL where the parent element has the cardinality of `1` and the child of `*`. Last, there are relations like `Flows`, `Triggers`, or `Association`, which describe that a certain asset causes an event or something comparable in another asset. Those relations are transformed to `Association` in MAL. Additionally, the first two relations become directed and the other relations undirected.

So far, we have presented how to model `Threats` and `Assets` in ArchiMate and the modeling of relations along their own types. Next, we show how `Threats` and `Assets` are related to each other. As Figure 2 already sketches, we facilitate `Associations` to link both concepts. From the point of MAL every `Threat` needs to be related to exactly one `Asset`. This leads to the fact that the ArchiMate model may contain several `Threats` having the same name, but describing different behavior as they are related to different `Assets`.

After the ArchiMate model is translated to a MAL-based DSL it is time to create the concrete instance of our model and run simulations on it. For the creation of an instantiated model, the typical procedure of creating test cases in MAL was followed. In summary, this procedure is comprised of the following steps: i) The creation of instances of all the assets that are found in the model ii) The establishment of connections between the assets, based on the associations defined on MAL, iii) Specifying of an entry point for the attacker and finally, iv) Running the simulation and performing compromise assertions on the attack steps that are of interest.

The results of the probabilistic simulations when using MAL are first a complete attack graph displaying all the connections between different attack steps over all the assets found on the model and second the TTC (calculated time to compromise) for each one of those attack steps. Therefore, the output of each of the aforementioned assertions is a value that represents the likelihood of this attack step to happen. Of course, the sum of all the attack step's TTC values that are in the same path on the attack graph (i.e. constituting one single attack) represents the total TTC of each corresponding complete attack. The higher the total TTC for an attack the harder the attack is to mount.

So the most important benefit of using MAL is that we not only get a security assessment of the model but we also get an estimate on how secure/insecure the architecture is.

It is worth mentioning that several test case models, including three concrete instantiated models, were constructed to ensure the automatically generated DSL's functionality and sanity. As those test cases are heavily based on the business domain, we will present them after the introduction of the domains in the next section.

5. Application Scenarios of Attack Simulations

Some techniques that security experts can use to model security in ArchiMate creating an instance of MAL have already been presented. The next step is to model a concrete system to simulate based on the created MAL instance. Accordingly, we following present two case studies of attacks on thermal power plants and on power grids.

5.1. Attack Simulation on Thermal Power Plants

The system model for thermal power plants is based on the concept study for a hard coal reference power plant [39]. The model considers the thermodynamic cycle with numerous auxiliaries. The IT is strongly simplified and the

electric power supply within the power plant is neglected even though consideration of these systems may add further attack vectors.

The overall goal in this paper is to cause a loss of production of electrical power. As the thermodynamic cycle can only operate if all parts work properly, shutting down one main component, if not redundant, will cause a reduction of electrical power output to the grid or a complete power outage.

Starting from `Feed Water Tank` three redundant combinations of `Pump/Compressor` and `Valve` bring water to the boiler (`Heater`) which uses `Fuel` to evaporate the water and superheat the steam before it flows into the high-pressure turbine through two valves. The steam from this turbine then enters the reheat (second `Heater`) before entering the intermediate-pressure turbine through two different valves followed by the low-pressure turbine. The three assets `Turbine` connected to `Generator` convert the energy of the steam first to mechanical energy and then to electrical energy, which is transferred to `Transmission Grid` via `Grid Transmission Switch`. After the turbines, the steam enters the condenser (`Cooler`), which is operated with `Coolant`. The condensate is then pumped to `Feed Water Tank` by two redundant combinations of `Pump/Compressor` and `Valve`.

The auxiliaries are required to operate the main components of the cycle in the thermodynamic cycle. The auxiliaries are connected to the assets from the thermodynamic cycle and are divided in three groups. The first relates to the rotating equipment such as `Pump/Compressor`, `Turbine` and `Generator`, the second to non-rotating equipment such as the `Heater` and `Cooler` and the third to those related to general control, which are `Measurements`, `Valve` with `Hydraulic control system`, `Load Controller` and `Grid Transmission Switch`. A note that must be done is that `Pump/Compressor` and `Cooler` are parts of the main components of the thermodynamic cycle but also part of the auxiliaries e.g. to pump and cool lubricants for `Lubrication`.

Cyber-attacks are carried out via the decentralized control. The model is splitted up into two levels. The control levels are the normal operator network `DCS` (distributed control system) and the `SIS` (safety instrumented system) network. Each networks collects measurement data from `Sensor` via `Measurement`, carries out an action according to `Firmware` of `PLC` (programmable logic controller) and gives the control signal to `Controller` which then acts upon e.g. `Valve` to carry out a physical action. The SIS level prevents the plant from entering potentially hazardous situations and may shut down units within the power plant if it detects an unsafe state. The included `PLC` with its sensors and controllers have a suitable redundancy, for critical equipment usually with triple redundancy [40]. They are separated from the DCS and may overwrite DCS control signals.

Nowadays however, this separation is reduced in favor of more efficient communication and data acquisition [40]. An attack on a plant in Saudi Arabia in 2017 showed that attacks on SIS level are possible by infiltrating to the `Engineering Workstation` [41]. Since SIS is the last line of automated defense (followed by mechanical safety systems), deactivating SIS may cause severe damage to the equipment if it enters an unsafe state e.g. caused by an attack on the DCS. Further, an SIS system does not continuously operate and is only checked in regular intervals. Thus, a modification of the system may not be noticed for a considerable period.

We present one of the studied instantiated models first in ArchiMate notation (Figure 3) and in a simple test case diagram (Figure 4). It shows an example attack on the lubrication of the rotating equipment, e.g. the turbine. Lubrication is widely used in power plants for bearings of rotating equipment such as the turbine but also pumps and compressors or the generator. Three threats are identified. `Influence Cooling` relates to change of the oil conditions, while `Reduce Flow` directly relates to the flow rate of the oil towards the turbine bearings. If the amount of lubricant is insufficient or its condition is inappropriate (`Reduced Flow`), the load capacity of the bearings reduces, they heat up and the turbine reaches an unsafe state. This causes the `SIS` to stop the rotation of the turbine (`System Failure`) leading to `Power Outage`. As the main goal of an attack is to cause a power outage, the goal is already achieved. However, if the previously described attack is combined with the third threat `Influence Measurement` then the unsafe conditions may not be detected in time leading to `Damage`, which again leads to `Power Outage`. Since now the turbine is physically damaged, bringing the plant back to operation will take significantly longer. It should however be noted that numerous measurements need to be influenced to achieve this secondary goal, requiring detailed knowledge about the plant's instrumentation and safety system.

When looking at this test case from the MAL point of view, the attacker has as its entry point access on the `Lubrication` asset of a power plant and more specifically is able to perform *reducedFlow*. The simulation ran by MAL provides the following results. By using this entry point the attacker is then able to aim towards the `Turbine` asset and achieve *systemFailure*, *damageEquipment* and *stopRotation* on the `Turbines`, which in turn will lead to *stopElectricityProduction* on the `Generator` and *powerOutage*. In Figure 5 the attack graph of the aforementioned attack is presented.

5.2. Attack Simulation on Power Grids – The Ukrainian Scenario

Following, we present the example of the IT attack on the Ukrainian electric power grid in December 2015 [1]. This attack was characterized by its coordinated and targeted approach to the critical infrastructure power supply and represents the first documented successful cyber attack that led to a local blackout for about 225,000 people in different parts of Ukraine. The attack involved a total of

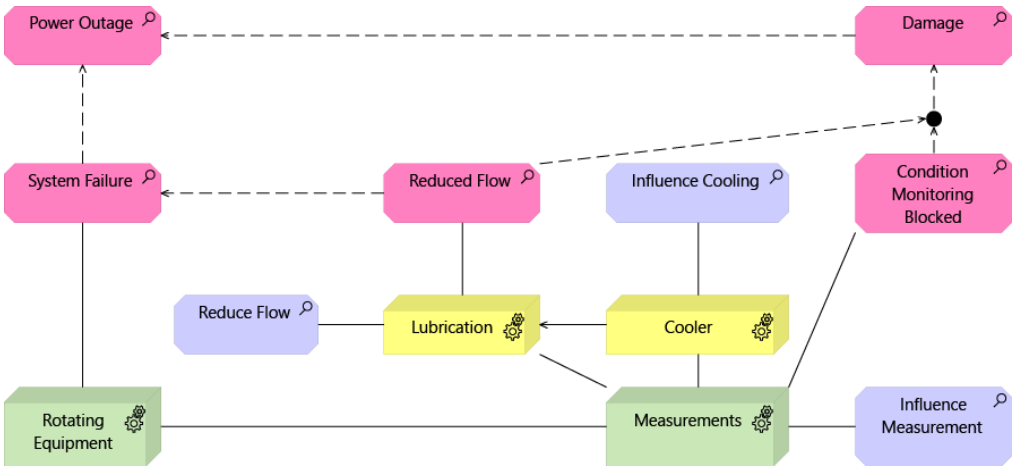

Figure 3. Attack on lubrication system

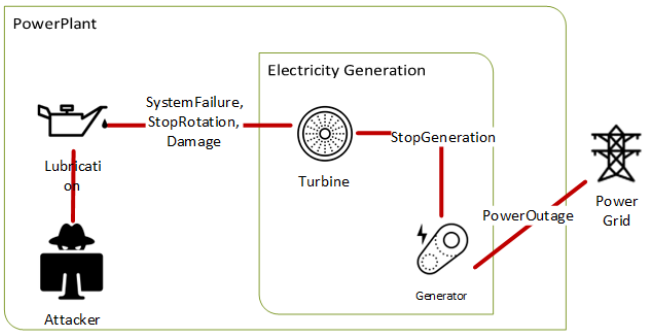

Figure 4. Example Test Case Diagram

seven substations with 110 kV and 23 substations with 35 kV over a period of three hours. Manual interventions resulted in the return to normal operations.

To model a generic `substation`, we follow the standard IEC 61850 [42] and give an overview of it structure in Figure 6. `Substations` are the interface between the `transmission grid` and `distribution grid` and convert the network voltage profiling by means of `transformer` from `high voltage` (e.g., 110 kV) into `medium voltage` (e.g., 35 kV). Both –high and medium voltage side `busbars`– serve as nodes of the respective voltage level and connect several `feeders` by means of `circuit breaker` with the network or subordinate urban areas.

Within the `substation`, components of the IT (blue icons in Figure 6) control the primary technical components (green icons in Figure 6). For example, `protection devices` are used in each of the `feeders`, which detect an electrical fault. If there is a electrical fault, the `protection devices` automatically switch off the associated `feeder`. `Substations` typically have many more components to control, monitor, and protect the assets. However, we only included those assets in Figure 6, which were part of the attack and hid all others.

The attackers on the Ukrainian scenario facilitated spearphishing attacks on the `office PCs` of the network operators as initial attack vectors [1]. The malware BlackEnergy 3 [43] allowed them to gain remote access to different PCs in the office zone. This enabled the attackers to capture VPN (virtual private network) credentials and move sideways within the substation. They aimed to control the central `SCADA` (supervisory control and data acquisition) systems of the network operators. The control of the `HMI` (human machine interface) systems allowed access to several `switches`, which led primarily to the blackout. At the same time, firmware manipulation attacks were carried out against serial-to-Ethernet `gateways` in the process network, to the uninterruptible power supply and KillDisk commands on `operator workstations`. This led to a refusal of service of these devices and increased the downtime and aggravation of the network rebuilding by the personnel [1]. As the `HMIs` needed to be operated manually, the coordinated attack on multiple distributed power grids was limited and, therefore, the consequences were still manageable.

6. Discussion

Hitherto, we have presented the process of modeling the domain, its related threats, its transformation to MAL, and, finally, its simulation. Next, we will discuss the outcome of the simulations. But before that, we will discuss the met objectives of our work and its shortcomings.

First, we managed to reuse existing EA models and the domain experts could concentrate on modeling the threats related to the existing model. To produce the models, we spent for each model two to three workshops with two hours in average. Finally, our transformation to the MAL instance was successful and our experts could affirm that our simulation results are equal to their expectations.

However, our experts remarked some negative points, which are mainly related to the modeling: First, the domain experts struggled using the ArchiMate notation. This is grounded in the fact that they came from the non-computer

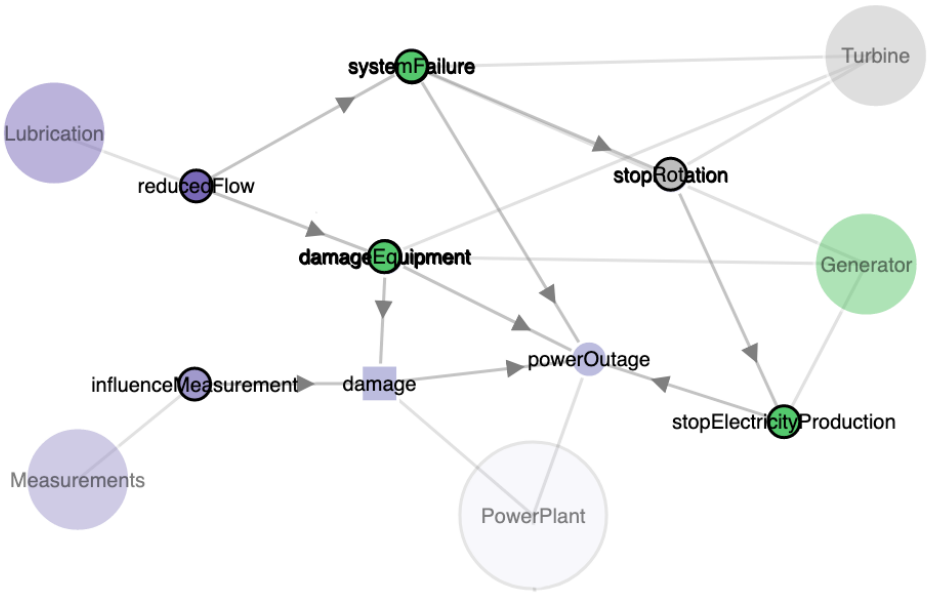

Figure 5. Attack graph of the example Test Case

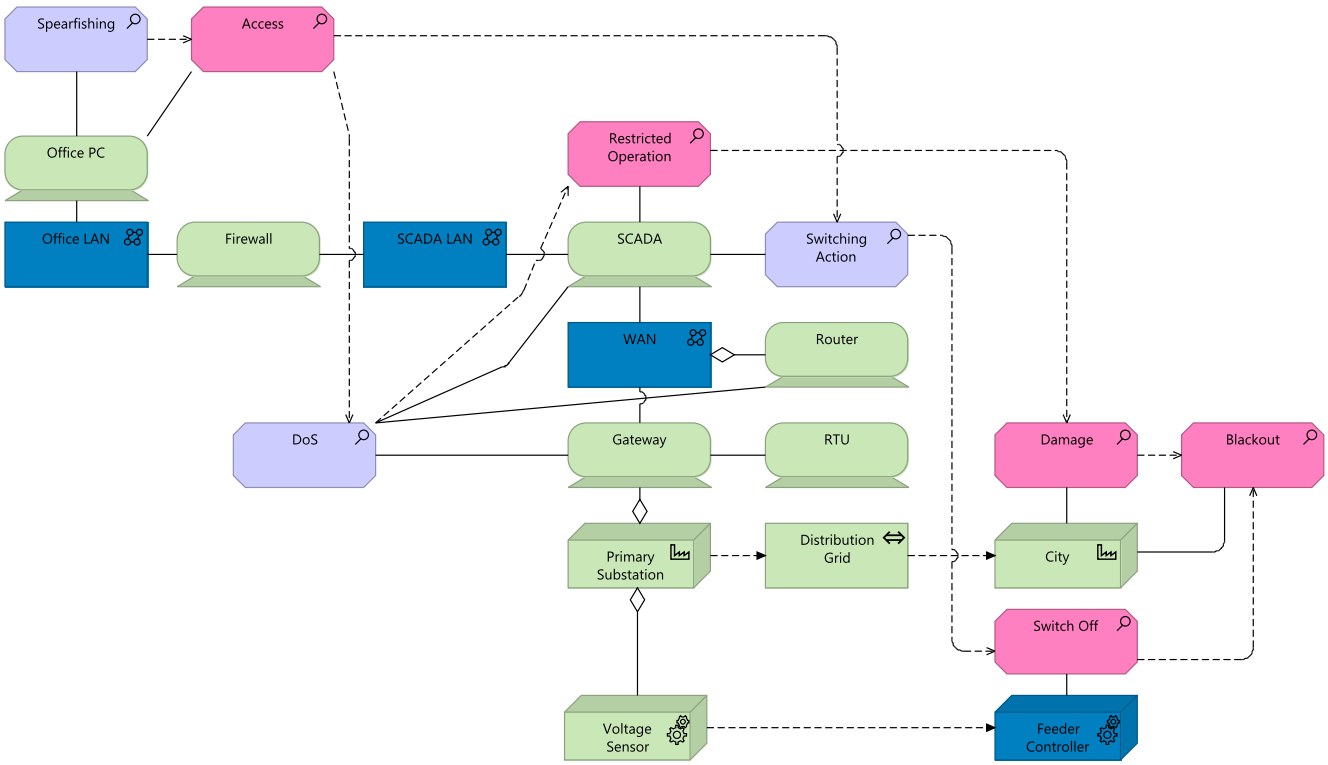

Figure 6. Power grid model - Excerpt of the Components exploited in the Ukrainian scenario

science related fields of thermodynamics and high voltage technology and, therefore, are not familiar with UML-like languages. Since assets and relations in ArchiMate have different properties and functions than those used in modeling tools of the related field, they had trouble to create the equivalent structure in ArchiMate. It proved to be especially difficult to set the right relations. However, after some explanations and practice they could model the threats completely independently.

Second, MAL expects that all threats are linked to a certain asset. In the first iteration of modeling, our experts did not relate all threats to an asset. This is caused by leaving certain steps of an attack not modeled, since the experts did not perceive every threat as important or introduce further threats for logical structuring, which cannot be related to an asset. To solve this issue, we added relations between those threats and assets, which are most likely related to it.

Leaving some threats not modeled leads also to the third shortcoming, because there is not always a link between assets that should be related as described by the related threats. To overcome this, we added "virtual" relations to our generated MAL instance that link assets to each other, which own each other related threats.

Last, we expected that every threat is related to *one* asset. However, our experts reused threats with the same name, which led to a not proper instance of MAL. Therefore, we conducted some refactoring and created several threats with the same name but different context. Next, we will discuss the simulation results in more details.

The modeled attacks on thermal power plants show the possibility of cyber-attacks to cause a power outage by taking large electricity providers off the grid. However, the cycle employed in a thermal power plant contains many main and auxiliary systems creating a highly complex environment. As such detailed knowledge about the employed systems, their architecture and the safety features is required. However, even an attack on an auxiliary system may cause a power outage as shown by the employed simple models, which demonstrate the general attack path and may be further subdivided to take care of the increased complexity. Furthermore, the attack on the plant in Saudia Arabia in 2017 [41] showed, that there exist groups capable of carrying out such attacks. This emphasizes the need for detailed modelling of these attacks to improve safety.

Certainly, providing a complete model of all thermodynamic, mechanical, electrical and IT main and sub systems goes far beyond the simplified model provided in this work. This complete model will be massive and experts from different fields will be required, especially to define possible entry points for cyber-attacks and assets reachable by these. Identifying these paths and a strict separation of different control lines such as DCS and SIS may reduce the size of the model and thus support this work.

7. Conclusion

Cyber-attacks on power assets can have disastrous consequences for individuals, regions, and whole nations as proven by the recent deliberate disruptions of electrical power and energy systems. In order to respond to these threats, the assessment of power grids' and plants' cyber security can foster a higher degree of safety for the whole infrastructure dependent on power. However, assessing the cyber security of power grids and power plants is difficult. Hitherto, we have proposed the use of attack simulations based on system architecture models to support security experts. As the previous approaches rely on a static implementation, we proposed MAL that defines which information about a system is required and specifies the generic attack logic.

So far, MAL and its instances are modeled using a DSL that is similar to program code. This may hinder security experts, who are not familiar to such a way of modeling, to adapt our approach. Additionally, this impedes the reuse of existing models like EA models, which can serve as input for the assets of MAL.

To overcome this issue, we proposed an approach, which allows the reuse of existing EA models notated in ArchiMate. Those models solely need to be enriched by information regarding the assets' threats. Afterwards, we created a transformation mechanism from the ArchiMate model to a proper instance of MAL (i.e. a MAL-based DSL) containing in total 56 attack steps over 28 different assets. This instance allows the security experts to model a concrete model instance of their domain (e.g., a concrete power plant), which serves as input for the MAL related simulation engine.

To show the applicability of our approach, we conducted two case studies in the power domain. First, we modeled a thermal power plant and possible attacks on it. Second, we facilitated the attack on the Ukrainian power grid. Based on these cases, we defined test case models, which ensure that the modeled attacks are present in the created MAL instance. Additionally, we conducted simulations on the concrete instances of the power plant and the power grid.

The results of the simulations were discussed with domain experts, who confirmed that our approach meets their expectations. However, our results show that a modeling of threats in ArchiMate notations might not the best choice, as domain experts usually are not common with those modeling concepts. This raises a point for future work: One might think about transforming existing models into MAL and, afterwards, providing a simple environment to model threats.

Further, we did not compare our approach with other existing approaches. Such comparison can be settled on the usability of the approaches as well as on their effectiveness and efficiency.

References

[1] Defense Use Case, "Analysis of the cyber attack on the ukrainian power grid," *Electricity Information Sharing and Analysis Center (E-ISAC)*, 2016.

[2] T. Petermann, H. Bradke, A. Lüllmann, M. Poetzsch, and U. Riehm, *Was bei einem Blackout geschieht: Folgen eines langandauernden und großflächigen Stromausfalls*. Büro für Technikfolgen-Abschätzung, 2011, vol. 662.

[3] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, p. 13, 2011.

[4] J.-W. Wang and L.-L. Rong, "Cascade-based attack vulnerability on the us power grid," *Safety science*, vol. 47, no. 10, pp. 1332–1336, 2009.

[5] M. Rosas-Casals, S. Valverde, and R. V. Solé, "Topological vulnerability of the european power grid under errors and attacks," *International Journal of Bifurcation and Chaos*, vol. 17, no. 07, pp. 2465–2475, 2007.

[6] I. Morikawa and Y. Yamaoka, "Threat tree templates to ease difficulties in threat modeling," in *2011 14th International Conference on Network-Based Information Systems*, Sep. 2011, pp. 673–678.

[7] M. Ekstedt, P. Johnson, R. Lagerström, D. Gorton, J. Nydrén, and K. Shahzad, "securiCAD by foreseeti: A CAD tool for enterprise cyber security management," in *Enterprise Distributed Object Computing Workshop (EDOCW), 2015 IEEE 19th International*. IEEE, 2015, pp. 152–155.

[8] H. Holm, K. Shahzad, M. Buschle, and M. Ekstedt, "P^{2}CySeMoL: Predictive, probabilistic cyber security modeling language," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 6, pp. 626–639, 2015.

[9] P. Johnson, R. Lagerström, and M. Ekstedt, "A meta language for threat modeling and attack simulations," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*. ACM, 2018, p. 38.

[10] The Open Group, *ArchiMate 3.0.1 Specification*, 2017.

[11] E. Grandry, C. Feltus, and E. Dubois, "Conceptual integration of enterprise architecture management and security risk management," in *2013 17th IEEE International Enterprise Distributed Object Computing Conference Workshops*, Sep. 2013, pp. 114–123.

[12] I. Band, W. Engelsman, C. Feltus, S. G. Paredes, and D. Diligens, "Modeling enterprise risk management and security with the archimate®," *Language, The Open Group*, 2015.

[13] L. Manzur, J. M. Ulloa, M. Sánchez, and J. Villalobos, "xarchimate: Enterprise architecture simulation, experimentation and analysis," *Simulation*, vol. 91, no. 3, pp. 276–301, Mar. 2015.

[14] J. Jürjens, *Secure systems development with UML*. Springer Science & Business Media, 2005.

[15] D. Basin, M. Clavel, and M. Egea, "A decade of model-driven security," in *Proceedings of the 16th ACM symposium on Access control models and technologies*. ACM, 2011, pp. 1–10.

[16] M. Alam, R. Breu, and M. Hafner, "Model-driven security engineering for trust management in sectet." *JSW*, vol. 2, no. 1, pp. 47–59, 2007.

[17] E. Paja, F. Dalpiaz, and P. Giorgini, "Modelling and reasoning about security requirements in socio-technical systems," *Data & Knowledge Engineering*, vol. 98, pp. 123–143, 2015.

[18] M. S. Lund, B. Solhaug, and K. Stølen, *Model-driven risk analysis: the CORAS approach*. Springer Science & Business Media, 2010.

[19] M. Almorsy and J. Grundy, "Secdsvl: A domain-specific visual language to support enterprise security modelling," in *Software Engineering Conference (ASWEC), 2014 23rd Australian*. IEEE, 2014, pp. 152–161.

[20] B. Schneier, "Attack trees," *Dr. Dobbs journal*, vol. 24, no. 12, pp. 21–29, 1999.

[21] S. Schneier, "Lies: digital security in a networked world," *New York, John Wiley & Sons*, vol. 21, pp. 318–333, 2000.

[22] S. Mauw and M. Oostdijk, "Foundations of attack trees," in *International Conference on Information Security and Cryptology*. Springer, 2005, pp. 186–198.

[23] B. Kordy, S. Mauw, S. Radomirović, and P. Schweitzer, "Foundations of attack–defense trees," in *International Workshop on Formal Aspects in Security and Trust*. Springer, 2010, pp. 80–95.

[24] B. Kordy, L. Piètre-Cambacédès, and P. Schweitzer, "Dag-based attack and defense modeling: Don't miss the forest for the attack trees," *Computer science review*, vol. 13, pp. 1–38, 2014.

[25] K. Ingols, M. Chu, R. Lippmann, S. Webster, and S. Boyer, "Modeling modern network attacks and countermeasures using attack graphs," in *Computer Security Applications Conference, 2009. ACSAC'09. Annual*. IEEE, 2009, pp. 117–126.

[26] L. Williams, R. Lippmann, and K. Ingols, *GARNET: A graphical attack graph and reachability network evaluation tool*. Springer, 2008.

[27] S. Noel, M. Elder, S. Jajodia, P. Kalapa, S. O'Hare, and K. Prole, "Advances in topological vulnerability analysis," in *Conference For Homeland Security, 2009. CATCH '09. Cybersecurity Applications Technology*, Mar. 2009, pp. 124–129.

[28] M. Frigault, L. Wang, A. Singhal, and S. Jajodia, "Measuring network security using dynamic bayesian network," in *Proc. of the 4th ACM workshop on Quality of protection*. ACM, 2008, pp. 23–30.

[29] P. Xie, J. H. Li, X. Ou, P. Liu, and R. Levy, "Using Bayesian networks for cyber security analysis," in *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP Int. Conf. on*. IEEE, 2010, pp. 211–220.

[30] S. Katsikeas, P. Johnson, S. Hacks, and R. Lagerström, "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*, 2019.

[31] Y. Jiang, M. Jeusfeld, Y. Atif, J. Ding, C. Brax, and E. Nero, "A language and repository for cyber security of smart grids," in *2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC)*, Oct 2018, pp. 164–170.

[32] CEN-CENELEC-ETSI, Smart Grid Coordination Group, "Smart grid reference architecture," 2012.

[33] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[34] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007.

[35] A. Barbosa, A. Santana, S. Hacks, and N. v. Stein, "A taxonomy for enterprise architecture analysis research," in *Proceedings of the 21st International Conference on Enterprise Information Systems*, vol. 2, INSTICC. SciTePress, 2019, pp. 493–504.

[36] The Open Group, *ArchiMate Model Exchange File Format: Version 2*, 2015.

[37] N. Mayer, "Model-based Management of Information System Security Risk," Theses, University of Namur, Apr. 2009.

[38] M. Fowler and C. Kobryn, *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley Professional, 2004.

[39] PowerTech eV, VGB and others, "Konzeptstudie referenzkraftwerk nordrhein-westfalen (rkw nrw)," *Report prepared by VGB Power Tech eV (co-ordinator), Babcock Borsig Power Systems GmbH, E. ON Kraftwerke GmbH, Universität Duisburg-Essen, Mark-E AG, RWI, RWE Power AG, Siemens AG Power Generation, STEAG AG, and Wuppertal Institute. Essen*, 2003.

[40] S. Basu, *Plant hazard analysis and safety instrumentation systems*. Academic Press, 2016.

[41] A. Di Pinto, Y. Dragoni, and A. Carcano, *TRITON: The First ICS Cyber Attack on Safety Instrument Systems*. Black Hat USA, 2018.

[42] IEC Standard, "IEC 61850: Communication networks and systems in substations," *Int. Electrotech. Commission, Geneva, Switzerland*, 2003.

[43] ThreatSTOP. (2016) Black energy. [Online]. Available: https://www.threatstop.com/sites/default/files/threatstop_blackenergy.pdf