7th European STAMP Workshop & Conference
18 - 20 September 2019, Helsinki

Aalto University

Exploring the Modeling of Attack Strategies for STPA

Abdullah Altawairqi and Manuel Maarek
Heriot-Watt University (Edinburgh, Scotland, UK)

ABSTRACT

System analysis for security and for safety are both focused on identifying potential accidents and attacks, to implement prevention strategies. Security system analysis aims to counter intentional acts that could make the system vulnerable. Systems-Theoretic Process Analysis (STPA) is a holistic approach to system safety analysis. In this paper, we explore the possibility to combine STPA analysis with Attack-Defence Trees (ADTrees) modeling to strengthen a system security analysis. We also discuss how the identification of the intentions and capabilities of the attackers could focus the priorities of the analysis and reduce its scope. We suggest an approach on how to combine ADTrees' attack modelling and STPA to elicit unsecure control actions. To illustrate this approach, we apply it on a case study.
.

Keywords: STPA; attack defence tree; attack modeling; security

1. INTRODUCTION

System security analysis and safety analysis have in common the identification of the circumstances that could threaten the functions of the system or its integrity. While safety is concerned with avoiding accidents, the analysis of the security system aims to prevent the system from suffering from intentional acts. In security analysis, this modeling of the intention of an attacker is carried out with the identification of security targets and, in combination with the mapping of the attack surface, helps to specify the defence of the system to be built.

Systems-Theoretic Process Analysis (STPA) is a hazard analysis technique based on Systems-Theoretic Accident Model and Processes (STAMP) which structures the system analysis around the notion of control loops. Understanding how a control loop could malfunction or could fail to achieve its goal lead to identifying the system's safety constraints. STAMP integrates the notion of causal factor to elicit these safety properties. Such modeling is intrinsically focused on the point of view of the system while a security analysis should include the attacker's intention and capabilities. Because of its safety focus, STPA does not capture the attacker's intention in the analysis. Gleaning such intentions could lead to the identification of system vulnerabilities that are more likely to be used in an attack scenario. As the system's control loops are the core of STPA analysis, we propose to integrate their modeling of the system in the attack strategy modeling. Attack trees are an example of attack strategy modeling. Attack trees are a graphical representation for modeling and analysing potential attack strategies. They could be extended to consider defensive patterns in Attack-Defence Trees (ADTrees).

In this paper, we explore the possibility of combining STPA analysis and ADTrees modeling to strengthen a system security analysis. STPA is a top-down approach to identify unsafe control actions from the control structure. Each element of the STPA control loops of a system could be the direct or indirect target of an attack. Deriving ADTrees is in itself a top-down analysis so we suggest guiding its refinement process with steps to make explicit the way an attack impacts a control loop. A bottom-up approach to security analysis starts by considering the system's attack surface to evaluate how potential vulnerabilities could be exploited. We propose to integrate this

attack surface perspective in our approach to combine ADTrees and STPA. To complement the modeling of attack intentions, we suggest to include attack profiles in our ADTree modeling to describe the potential attacker in terms of its skills and motivation. Associating attack profiles with attack scenarios help to narrow the scope of an analysis.

The structure of the paper is as follows: Section 2 gives background and related work on safety and security analysis approaches, Section 3 proposes an approach to integrate ADTrees with STPA, Section 4 applies the proposed approach to a case study of the steel plant, and Section 5 concludes.

2. BACKGROUND AND RELATED WORK

In this section we introduce the main background of our work, ADTrees and STPA. We then present some related works.

Attack trees (Schneier, 1999) are a graphical representation of the potential scenarios of an attack as a tree of potential attack strategies. Attack trees aim to provide a way of thinking about the system exposure to attacks. The root node of an attack tree is the goal of the attacker. The relationship between a parent node and its children nodes is following a logical structure called variance. Children nodes are either considered as conjunctions (AND) of actions that lead to the parent node state, or a disjunction (OR) of actions all resulting in the same parent node state. Attack trees have many flavours. Jhawar, Kordy, Mauw, Radomirović, & Trujillo-Rasua (2015) introduced the sequential conjunctive operator (SAND) that enforces an order in which the actions are to be conducted in the attack. Kordy, Mauw, Radomirović, & Schweitzer (2014) extended attack trees by considering defensive patterns in the so-called Attack Defence Trees (ADTrees). According to Kordy et al. (2014), original attack trees do not address the interactions between attacks scenario and the defences of the system. ADTrees contain defensive nodes called countermeasures. Those nodes could appear at any level on the tree and follow the logical structure AND and OR. Defensive nodes are system actions that are to prevent attack steps (Kordy et al., 2014). In ADTrees, a defensive node drawn as child or an attack node indicates that the attack is prevented by the defence.

STPA is safety analysis approach based on STAMP. STPA's approach focuses on accidental causes and safety constraints. STPA identifies the root of accidents which are hazardous scenarios to define safety constraints that need to be fulfilled by the system to prevent these accidents. It is top-down process to identify failure states of the system by analysing the controls of the system and how they can fail. This analysis leads to stating safety constraints the system must fulfil (Leveson & Thomas, 2013). STPA has four basic analysis steps. First, to define the purpose of the safety assessment, system losses and system hazards. Second, to identify the control actions of the system's control model. Third, to establish the safety constraints and requirements from the identified unsafe control actions. Fourth, to identify causal scenarios. While STPA guides the analysis in identifying causal scenarios leading to failed control loop, it does not provide guidance for the identification of intentional causal scenario based. A security causal scenario is characterised for instance by the attacker's intention, the attacker's capabilities, the system's surface of attacks.

A number of ongoing researches are proposing to extend safety system analysis for security. We discuss some of these works as they relate to the approach we are discussing in this paper.

STPA-Sec (Young & Leveson, 2013) aims at providing a solution to this security modeling need with a semi integrated approach between safety and security. It follows the STPA top-down approach but focuses on identifying losses and vulnerable states in order to strengthen the security of a system. STPA-Sec has the same basic process of STPA where vulnerabilities replace hazards. Even though STPA-Sec is an analysis approach for safety and security, it does not distinguish between intentional causal scenarios that are central to security analysis.

The Failure Mode Vulnerabilities and Effect analysis (FMVEA) is a step by step approach for investigating vulnerabilities-based failure mode and the potential effects these weaknesses could have in terms of decreased readability and availability of the system (Schmittner et al., 2014). It is an extension from The Failure Mode and Effect analysis (FMEA) used in safety to document the analysis of the impact of a component failure on the overall system. FMEVA proposes to include vulnerabilities and attack models to identify potential attack vectors of concern for the system. FMVEA uses cause effect chains into vulnerabilities, threat agents, threat modes, threat effects and attack probabilities in its modeling of attacks.

STPA-SafeSec (Friedberg, McLaughlin, Smith, Laverty, & Sezer, 2016) is a fully integrated approach between combining safety analysis and security analysis. The authors explain that their approach goes beyond STPA-Sec as it provides guidance to evaluate the safety impact the constraints derived from the security analysis could have. STPA-SafeSec extends the core of STPA's approach by considering security causal factors on integrity and availability. It claims to overcome limitations in STPA-Sec's approach by adding physical components layer into the control loop analysis to model the surface of attack and its link with the core safety features of the system. It also advocates for mapping security and safety constraints to the control layer in order to mitigate potential safety and security conflicts.

S-cube were introduced as a joint safety and security analysis model for industrial control system (Kriaa, Bouissou, & Laarouchi, 2015). S-cube is enabling formal modeling for system architecture and automates the generation of attack and failure scenarios. The automation results are depending on assigned hypothesis.

3. INTEGRATING ATTACK MODELING AND SYSTEM ANALYSIS

In this section, we will explain the proposed approach of the attack model for system security analysis. In section 3.1 we will present a brief on the proposed approach. In section 3.2, we explore to link the attackers' intentions and their strategies with STPA control loops of the system. Section 3.3 relates the ADTrees analysis and the attack surface of the system. In Section 3.4, we suggest extending this approach to using attack profile to focus the analysis of attack strategies.

3.1. Integrating Attack Modeling to the STPA Process

Attack modeling for system security analysis is an approach on top of STPA. Figure 1 illustrates the attack modeling process. The approach extends STPA process with three main steps which are the identification of attack profiles, the identification of unsecure control actions and the refinement into attack strategies. The attack profiles are defined to focus the analysis on specific attacker's capabilities.

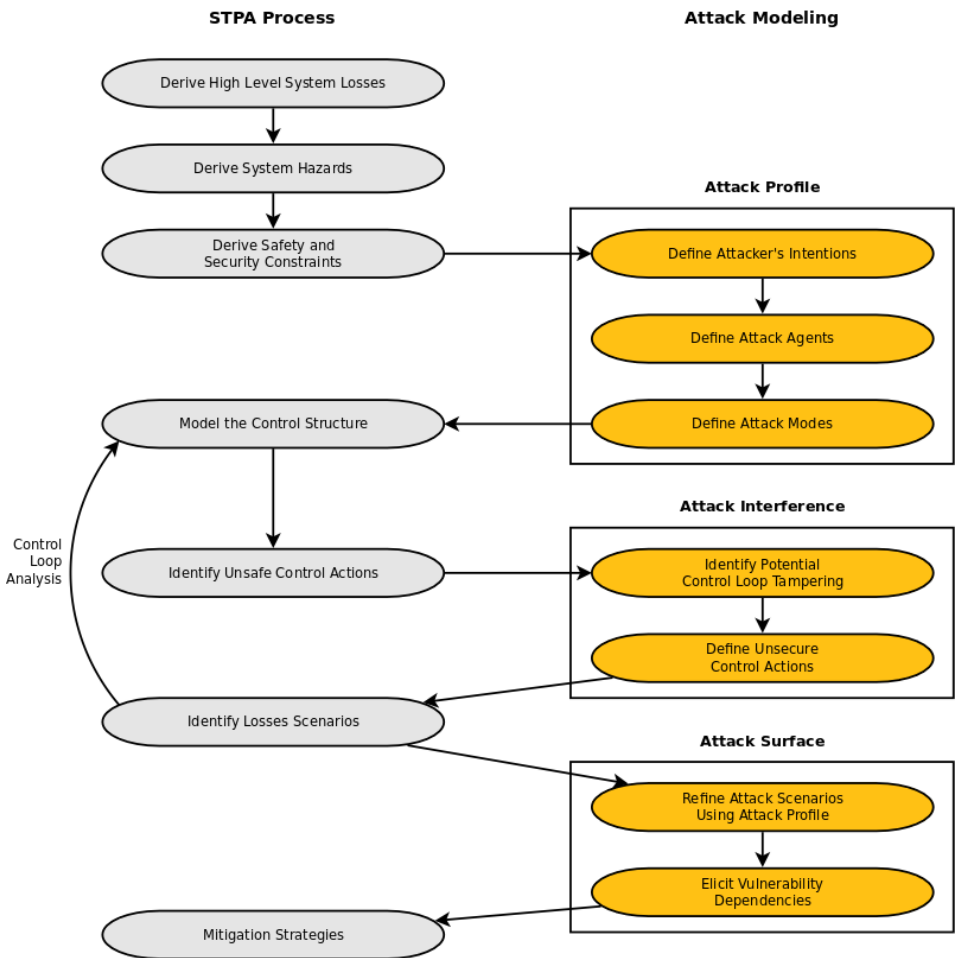

Figure 1: Extension of STPA Process (Leveson & Thomas, 2013) with Attack Modeling

3.2. Bridging STPA Control Loops and Attack Defence Trees

In this section we explore how intention becomes action and how intention affects the STPA control loop. The difference between safety analysis and security analysis lies in the fact that the first does not consider the intention of an attacker. The attacker's intention and potential attack scenarios to accomplish the attacker's goal could be modelled using ADTrees with the root and upper attack nodes of the tree representing the attacker's intention and these nodes being decomposed further down the tree into specific attack steps.

To combine STPA and ADTrees modeling for security analysis, we suggest structuring the attack modelling following this pattern. Taking one attacker's aim as root of an ADTrees, we decompose it into more strategic intentions the attacker could envision to pursue the goal. We name this top part of the tree the *Intention Tier*. This part is composed only of attack nodes and is free of elements from the system's modeling.

From each resulting individual intention, we continue the ADTrees modelling by building subtrees which are now in the *Control Tier*. This step is done by considering the attacker's intention faced with the STPA control loop of the system. The attack node is systematically decomposed into sub attack nodes targeting or tempering with each element of the STPA control loop. We name this refinement between single intention attack node and its control-loop specific attack sub-nodes a *Tampering Chords* as its aim is to identify how an intention could tamper with one element of the control loop and eventually resonate with the entirety of the control loop. This represents how the attacker could tamper with the system's control and therefore trigger unsecured actions. Tampering Chords are the sets of connections between the Intention Tier and the Control Tier.

Different strategies or phases of an attack are analysed with regards to the STPA control loops of the system in the Control Tier of the ADTree. In the Control Tier the description of attack scenarios remains high level. The attack and defence nodes at the Control Tier are associated with an STPA control loop. Tampering Chords are the connections between the nodes in the Intention Tier and the Control Tier, they are the bridges to translating the attacker's intentions into specific disruptions to the system's STPA control loops. Tampering Chords are the key to establishing *unsecure control actions* triggered by the attacker's actions that need to be prevented.

The generic STPA control loop consists of four main elements which are Controller, Actuator, Controlled process and Sensor. The control model presented in STPA is meant to define unsafe control action for the controller and the control process. An attacker could tamper with any element of the control loop in a way that would trigger an unsecure control action by the system. Missing or inadequate actions in a control loop could be hazardous for the system.

Suspected system behaviour could be expressed as an intentional system failure (triggered by an attacker's action) and non-intentional system failure. The STPA causal factors can provide the rationale for how non-intentional system failures can occur. These could be complemented by ADTrees to give the rational for how intentional system failures can occur. We can use security constrains as countermeasures for the attack scenarios. Defence nodes are a shortcut for establishing security requirements.

3.3 Attack Surface and ADTrees

Individual attack nodes within the Control Tier are related to an element of the STPA Control Loop. These individual attack nodes might have sub defence nodes which will correspond to security constraints. They could also be refined further into more concrete actions. This refinement should reach a point where the attacker is exploiting a vulnerability of a component of the system to start or continue its attack. Such concrete attack actions are leaves in the ADTree. These leaves represent the attack surface of the system. The way they are combined gives the dependency between components' vulnerabilities. We name this part of the ADTree the *Component Tier* as it is closely related to the physical implementation of the system. Steps of concrete attacks in the Component Tier are combined to reach nodes of the Control Tier. By using these separate tiers, we distinguish the system surface of attack's exploit and the deception of the intelligence of the system by attacking its control loops.

Figure 2 illustrates how Tempering Chords seat at the boundary between Intention Tier and Control Tier, and how Components Tier refines the attack strategies by highlighting the attack surface vulnerabilities they exploit.

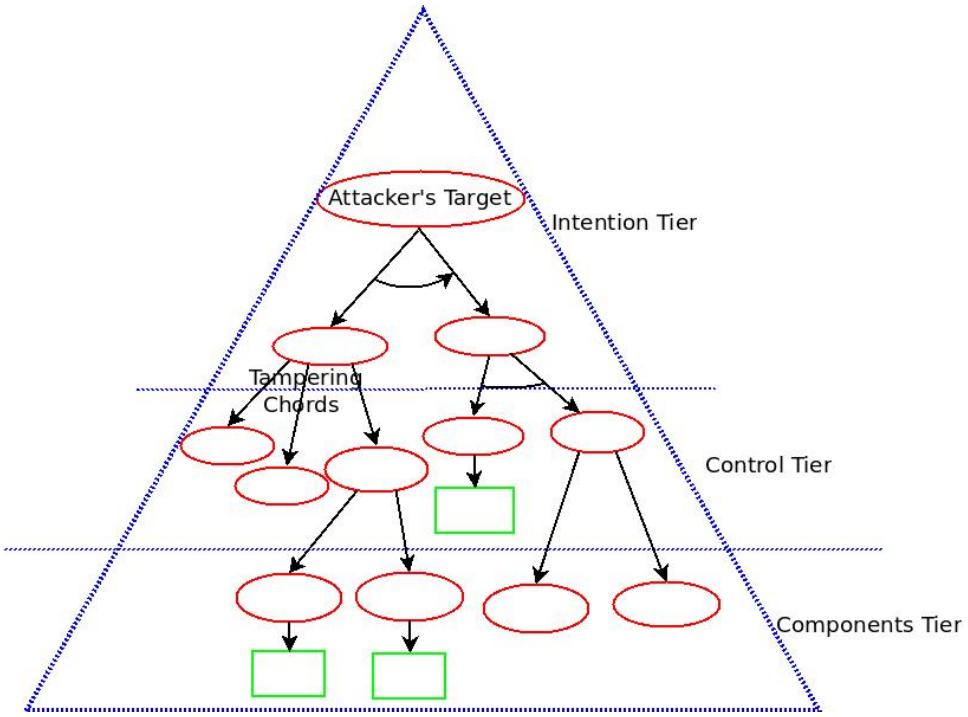

Figure 2: Tampering Chords and Ties of the ADTree Analysis

3.4. Enhancing attack scenarios by attaching Attack Profile

An *Attack Profile* is a way of expressing the attacker's abilities. Inspired by attack characterisations in (Schmittner, Ma, & Smith, 2014), we define attack profile as being is composed of two mains categories: the attack agent and the attack mode.

Attack agent is the abilities, knowledge and capabilities to attempt attack regardless of the intention. They could be categorised into script kiddie (attacker with a medium expertise, and he can apply self-learning material), blue hat (experience attacker who makes his attack with the purpose of showing his skills), black hat (experienced attacker who makes his attack for the purpose of terrorising, for money, for political ideals or religion motives) and elite hacker (expert designing and deploying their own tool to sell vulnerabilities that they discover in the black market). The capabilities of cyber-attacks are very high because the attackers have access to wide recourses and because such skills are related to intelligence.

Attack mode could be malicious, denial of service, spoofing identities and publish tools. Malicious code can be defined as a piece of code that usually connects to another program and can cause the system to behave unpredictably. Each code is designed for different reasons. The activation time depends on the design, for example trojans, worms and viruses. Their propagation is variable. User interaction may not be required like with viruses. Denial of Service: it is intensive connection from a group which aims to block the service provider and cause network congestion which lead to service delays. Spoofing Identities: is defined as a process in which a single computer, email, or other account associated with the service or a computer receive is hijacked or stolen by hackers. It necessitates some technique like fishing or social engineering.

Defining appropriate attack profiles and attaching attack profiles to ADTrees could help to focus the analysis by employing only capabilities related to the profile to narrow down the assessment. However, this approach should not prevent exploring wider attack profiles but helps to organise the analyses.

4. CASE STUDY

4.1 Cyber Attack Steel mill in Germany

The second known cyber-attack that resulted in a damage to physical systems concerned a German steel factory in December 2014. The Federal Office for Information Security announced the steel mill accident in their annual report without mentioning the name of the factory. Reportedly, the attackers used phishing email to gain access to the plant's network and then gain access to the production mill's network. The malware, which redirected to a malicious website, was downloaded to the targeted computer from a trusted email. The attacker was able to cause system components failure. This had a specific impact on the shutdown of critical components, which led to the impossibility of stopping the blast furnace (Lee, Assante, & Conway, 2014).

The steel mill was targeted with the intention to cause physical damage. The general network of the facility was hacked at the beginning of the attack. Then, the plant's production network which contains the management software of the steel mill was penetrated. The attacker took control of the plant's controlling system and succeeded in disabling the furnace's safety settings which caused serious damage to the infrastructure. According to the report, the attacker had a good knowledge and experience of the system. Figure 3 shows the design of the blast furnace's controlling system (Lee et al., 2014). The controlling system has a dashboard with several indicators such as the temperature of the furnace, its pressure, the water level in the tank. An operator has access to the dashboard and can require in an emergency the pumping of more water from the backup tank into the main tank or to stop hot blast and water bump. The computer of the controlling system controls the temperature of the furnace automatically by opening the blast furnace hot air valve and closing it. The cooling system is also controlled by the computer automatically using water pumps. The temperature in the furnace must be between 1500°C and 2000°C in order to produce steel.

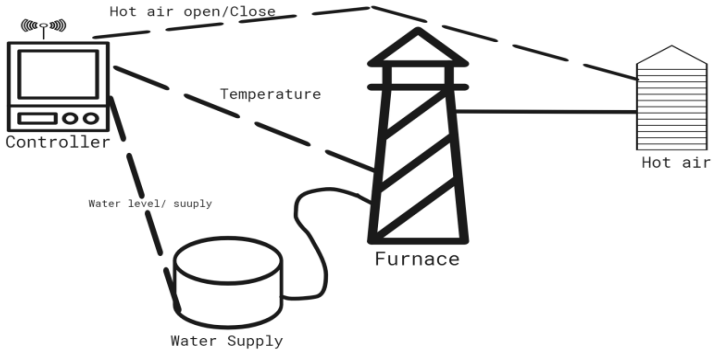

Figure 3: Steel Mill Simple Design System

4.2 STPA Analysis for Steel Mill Case Study

The first step in STPA is to define the purpose of the analysis, the system boundary, and losses and hazards for the system (see below).

Objective: to produce and sell steel
Losses:
L1- People die or injured in the steel mill.
L2- Steel mill production is stopped.
Hazards:
H1- Furnace is overheated [L1, L2]
H2- Furnace is unable to produce steel [L2]
H3- Furnace is physically injuring people [L1]
Safety constraints:
SC-1 Furnace temperature must be operated within limits [H1,H2,H3]
SC-1.1 Furnace temperature must not exceed 2000C [H1,H2,H3]
SC-1.2 Furnace temperature must not get lower than 1500C [H2]

The second step is to model the control structure. The analysis must identify the physical process and controllers, then define an unsafe control structure. Figure 4 shows the model of the control structure for the cooling mechanism, the heating of the furnace and their interactions.

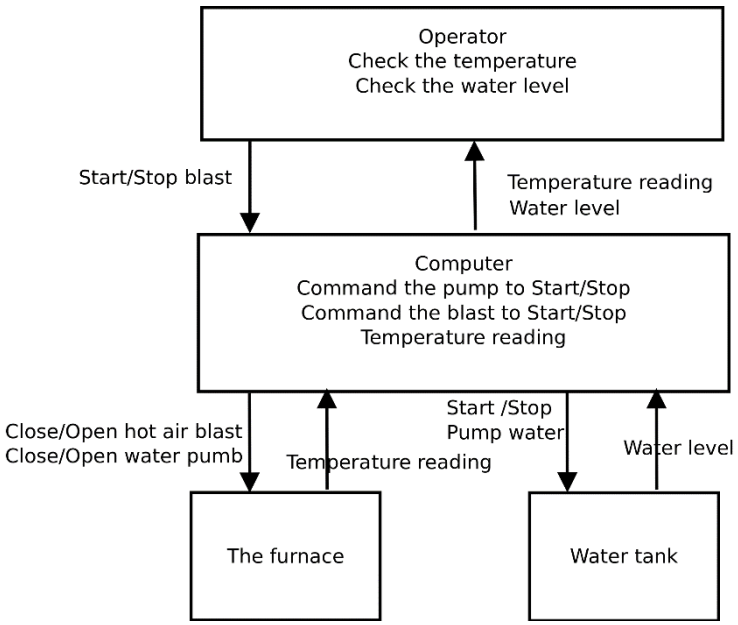

Figure 4: Steel Mill STPA Simple Control Loop

The third step in STPA is to identify unsafe control actions from the control structure which is mainly to find the behaviour to be prevented. Table 1 gives the system's unsafe control actions.

Table 1: STPA Unsafe Control Actions

	Not providing causes hazard	**Providing causes hazard**	**Too early, too late, order**	**Stopped too soon/Applied too long**
Open water pump	UCA-1: Computer does not provide open water valve when hot air valve close		UCA-2: Computer provides open water pump more than X seconds after hot air open	UCA-3: Computer stops providing open water pump too soon before the hot air valve fully open
Close water pump		UCA-4: Computer provides close	UCA-5: Computer provides close	

		water pump while hot air open [H1,2]	water pump more than X seconds before hot air close	
Open hot air		UCA-6: Computer provides open hot air while water pump is closed [H1,2]	UCA-7: Computer provides open hot air more than X seconds before water pumps open	
Close hot air	UCA-8: Computer does not provide close hot air when water pump is closed		UCA-9: Computer provides close hot air more than X seconds after water pumps close	UCA-10: Computer stops providing close hot air too soon before water pump is closed

Therefore, we can establish safety constraints (see below) from these unsafe control actions.

SC-1: The computer must not supply the open water valve when the hot air valve closes [UCA-1]
SC-2: The computer must not supply the open water pump for more than X seconds after opening the hot air [UCA-2]
SC-3: The computer must not supply the open water pump too early before fully opening [UCA-3]
SC-4: The computer must not supply a closed water pump when hot air is open [UCA-4]
SC-5: The computer must not supply the water pump closed more than X seconds before the hot air closes [UCA-3]
SC-6: The computer must not supply open hot air while the water pump is closed [UCA-6]
SC-7: The computer must not supply open hot air for more than X seconds before the water pumps open [UCA-6]
SC-8: The computer must supply hot air nearby when the water pump is closed [UCA-8]
SC-9: The computer must not supply hot air closed more than X seconds after the water pumps are closed [UCA-9]
SC-10: The computer must not interrupt the supply of hot air nearby too soon before closing [UCA-10]

The last step in STPA is to identify loss scenarios. This step is to explain how unsafe system behaviours could occur. For these scenarios, we consider multiple potential unsafe control actions. The updated model of Figure 5 includes the unsafe control action with the generic control diagram in blue. In Figure 6, the process model in red indicates what the controller believes. The process model for the water level indicates that the controller is to pump water from the reserve tank or use the backup pump when the water level is low. The temperature is normal. Thus, we need to redefine the process model in such a way that the computer should generate an alarm whenever the water level is getting low, helping the operator to send the command to stop the hot air or choose to do it manually.

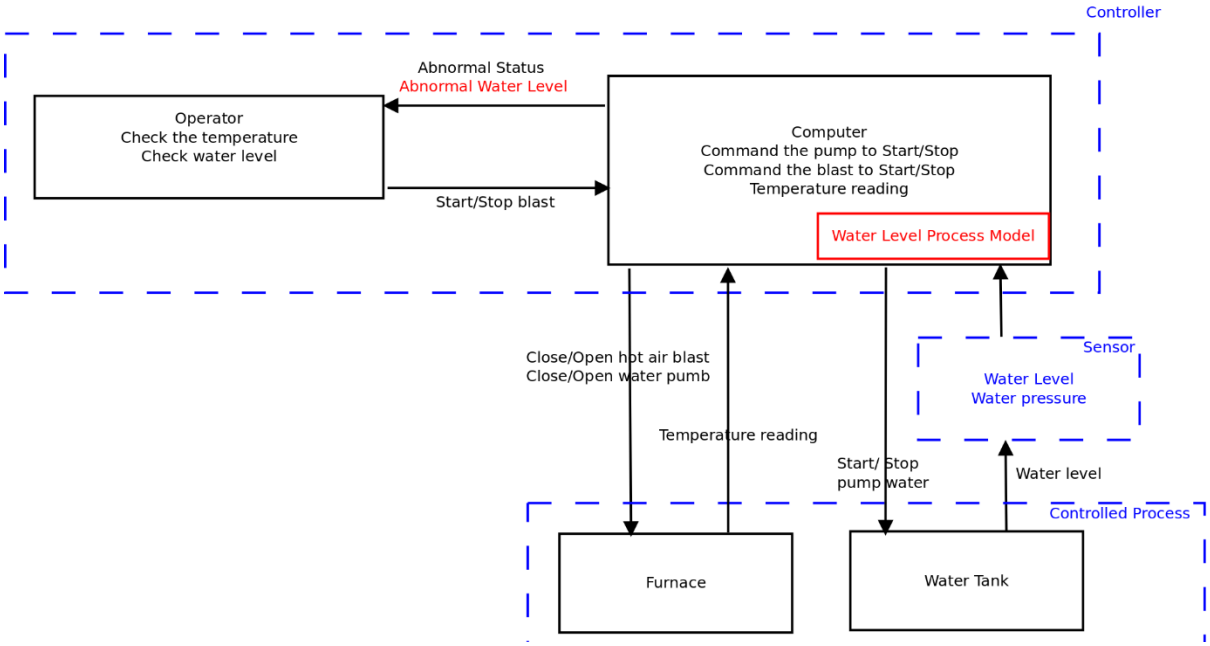

Figure 5: STPA Control Loop with Process Model

S-1: The operator did not recognize the rapid increase in the temperature indicator because the indicator showed normal status.

S-2: The operator responses to the water level decreases by pumping more water into the cooling system and switching the backup pump.

S-3: The rapid increase of the temperature leads to water leak; which results in more water being pumped to the cooling system; which results in the mixing of water and iron; which leads to the explosion.

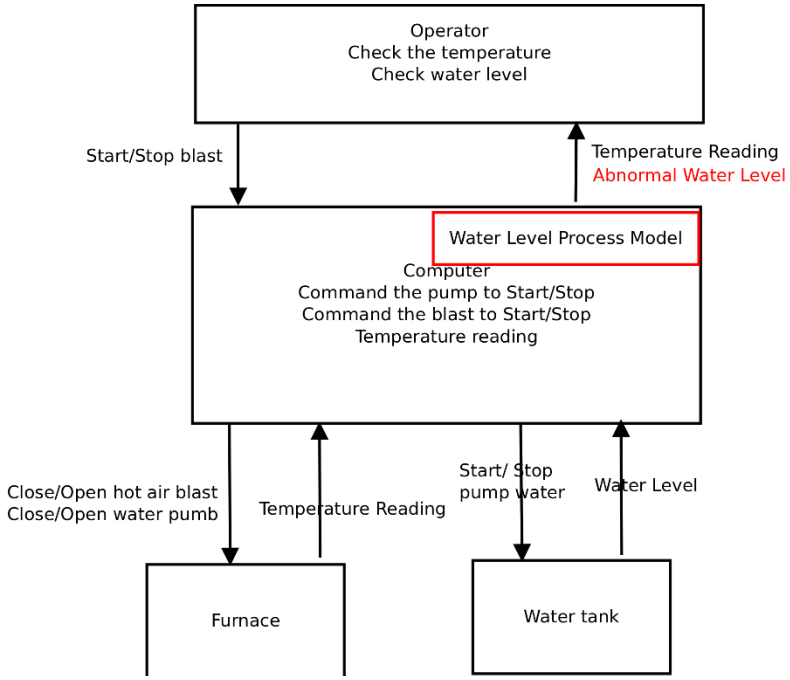

Figure 6: STPA Water Level Process Model

4.3 Steel Mill Attack Modeling

In this section we use ADTrees to model potential attacks in relation with the STPA analysis. In this case study, the intention of the attack is to cause life losses or enormous physical damage.

We build our ADTree, see Figure 7, following the steps of Section 3. The goal of the attacker is decomposed into intentions. Here, to simplify the tree, we showed a single intention. We then consider how this intention can tamper with the control loop of the system. We created three attack nodes as children of the intention attack node. These three nodes, which are unsecure control actions (USECA), could be later refined into specific attack sequences. In the example of Figure 7, the leftmost sub-tree corresponds to the successful attack described in the report. They exploited vulnerabilities in the networks and operating system (the sub-tree reaches elements of the attack surface). The rightmost sub-tree shows an example of a defence node.

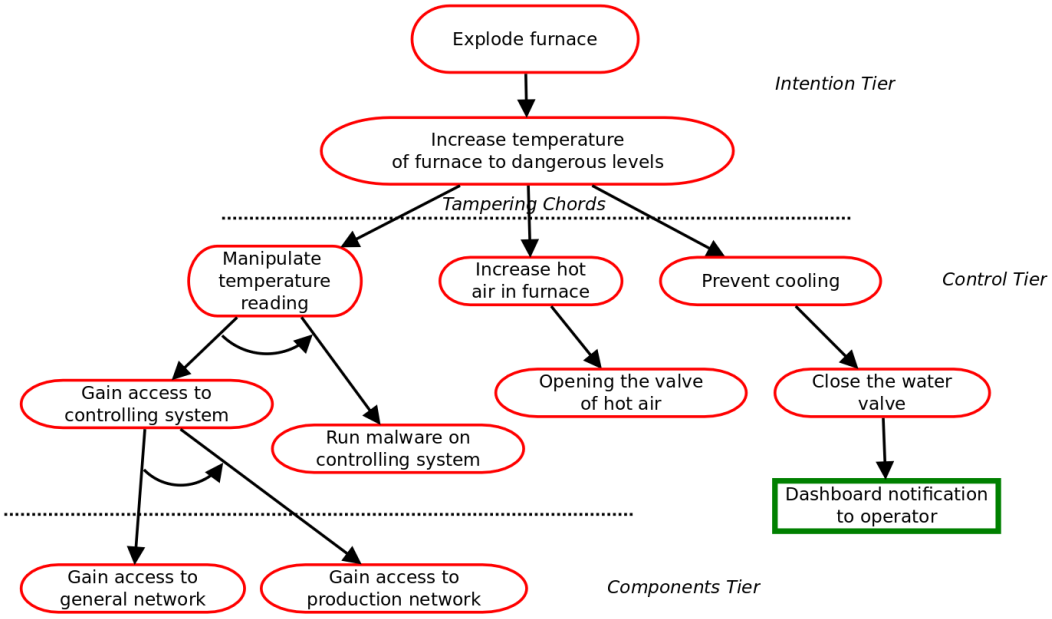

Figure 7: ADTrees Scenario with vulnerabilities dependencies

The following USECAs correspond to the attack scenarios of Figure 7.
USECA-1 Attacker manipulates temperature reading
USECA-2 Attacker increases hot air in furnace
USECA-3 Attacker prevents cooling

For each USECA we could derive scenarios of attack from the ADTree, for instance the following scenario.
USECA-1
Components: Sensor
Control action 1: Gain access to controlling system
Control action 2: Gain access to general network
Control action 3: Gain access to production network
Control action 4: Run malware on controlling system

The attack leaves of the tree correspond to the vulnerabilities of the system. Note that an attack scenario would exploit individual vulnerabilities. The scenarios of attack indicate how these exploits could be combined. Such scenarios are therefore building a set of vulnerability dependencies which map the attack surface of the system.

The scenarios modelled with ADTrees can be refined using attack profiles. For instance, exploiting the networks and operating system vulnerabilities could well be done by attackers with

different levels of expertise (e.g. script kiddie or elite hacker) which will make steps in the attacks to be more or less likely to take place. Their intentions might also differ. Attaching these attack profiles to the ADTrees could focus the analysis by bringing additional realistic aspects to the scenarios of attack.

5. CONCLUSION

In conclusion, we explored the modeling of attack strategies together with control structure of STPA using ADTrees. This should facilitate the elicitation of vulnerabilities most likely to cause harm to the system, and to define attack countermeasures. We propose to guide the building of ADTrees by scrutinising the way attacker's intention meet the control loops of the system. We also suggest using attack profiles to produce capability-focused attack scenarios. We applied this approach on a case study. We believe that this example shows the potential to help narrow down the attack scenarios modelled with the help of attack profiles. The connection between the STPA control loop and ADTrees elements offers a perspective in the design of modeling tools to establish unsafe actions in STPA, including the attacker's intention. This work is still in progress. We are developing prototype modeling tools to evaluate the implementation and effectiveness to help assess in the security of complex systems.

REFERENCES

Friedberg, I., McLaughlin, K., Smith, P., Laverty, D., & Sezer, S. (2016). STPA-SafeSec: Safety and security analysis for cyber-physical systems. *Journal of Information Security and Applications*. https://doi.org/10.1016/j.jisa.2016.05.008

Jhawar, R., Kordy, B., Mauw, S., Radomirović, S., & Trujillo-Rasua, R. (2015). Attack Trees with Sequential Conjunction. *ICT Systems Security and Privacy Protection*, 339–353. https://doi.org/10.1007/978-3-319-18467-8_23

Kordy, B., Mauw, S., Radomirović, S., & Schweitzer, P. (2014). Attack–defense trees. *Journal of Logic and Computation*, *24*(1), 55–87. https://doi.org/10.1093/logcom/exs029

Kriaa, S., Bouissou, M., & Laarouchi, Y. (2015, October 20). *A Model Based Approach For SCADA Safety and Security Joint Modelling: S-cube*. https://doi.org/10.1049/cp.2015.0293

Lee, R. M., Assante, M. J., & Conway, T. (2014). German steel mill cyber attack. *Industrial Control Systems*, *30*, 62.

Leveson, N., & Thomas, J. (2013, August). *An STPA Primer*. Retrieved from http://sunnyday.mit.edu/STPA-Primer-v0.pdf

Schmittner, C., Ma, Z., & Smith, P. (2014). FMVEA for Safety and Security Analysis of Intelligent and Cooperative Vehicles. In A. Bondavalli, A. Ceccarelli, & F. Ortmeier (Eds.), *Computer Safety, Reliability, and Security* (pp. 282–288). Springer International Publishing.

Schneier, B. (1999). Attack Trees. *Dr. Dobb's Journal*. Retrieved from http://www.schneier.com/paper-attacktrees-ddj-ft.html

Young, W., & Leveson, N. (2013). Systems Thinking for Safety and Security. *Proceedings of the 29th Annual Computer Security Applications Conference*, 1–8. https://doi.org/10.1145/2523649.2530277