HA-Grid: Security Aware Hazard Analysis for Smart Grids

Luca Maria Castiglione*, Zhongyuan Hau*, Pudong Ge†, Kenneth T. Co*,
Luis Muñoz-González*, Fei Teng†, and Emil Lupu*
* Department of Computing, Imperial College London
† Department of Electrical and Electronic Engineering, Imperial College London

***Abstract*—Attacks targeting smart grid infrastructures can result in the disruptions of power supply as well as damages to costly equipment, with significant impact on safety as well as on end-consumers. It is therefore of essence to identify attack paths in the infrastructure that lead to safety violations and to determine critical components that must be protected. In this paper, we introduce a methodology (HA-Grid) that incorporates both safety and security modelling of smart grid infrastructure to analyse the impact of cyber threats on the safety of smart grid infrastructures. HA-Grid is applied on a smart grid test-bed to identify attack paths that lead to safety hazards, and to determine the common nodes in these attack paths as critical components that must be protected.**

I. Introduction

As we continue to observe more Advanced Persistent Threat (APT) Groups targeting the energy sector with Operational Technology (OT) specific malware such as BlackEnergy [22], Crashoveride [13] and more recently, Pipedream [7], there is an increasing need to evaluate the safety consequences of cyber attacks on smart grid systems. The attack on the Ukrainian power grid in 2015 resulted in an hour of power outage up for 225,000 end-consumers; a subsequent attack in 2016 resulted in a loss of 20% of the capital city's power consumption [4]. With the increasing threats on OT systems, coupled with the adoption of network-enabled smart devices in the OT environment, it is inevitable that the attack surface (and the attack paths) to compromise the OT system increases significantly. Tools are therefore needed to identify and evaluate the attacks that lead to safety violations.

To this end, we present a hazard-driven security analysis methodology for secure **H**azard **A**nalysis of Smart **Grid** infrastructures (HA-Grid) that is able to enumerate attack paths that lead to safety violations. Our proposed methodology consists of 4 steps: 1) We perform safety analysis to obtain the safety model of the system [19]. 2) We then map the elements of the safety model to the system's architectural components and 3) model the effect of an attacker that *Tampers* and *Spoofs* information flows and process model variables in the safety model. 4) Finally, we use MulVAL [23] to enumerate the attack paths that lead to safety violations. Generating attack paths that lead to safety violations in smart grid infrastructures allows defenders to identify safety-critical assets such as components that are common modes of failure (i.e., common nodes in attack paths), and focus hardening efforts towards these components.

Although incorporating security elements into safety analysis of smart grids has been previously explored [8], we are the first to leverage the integration with the architecture of the smart grid and generate attack paths as sequences of privileges that an attacker must obtain to cause an accident. The ability to generate feasible attack paths that lead to safety violations provides valuable information for system designer and defenders to secure the system. In summary, we propose a safety-driven security analysis methodology for smart grid infrastructure (HA-Grid) and apply it to a real life smart grid infrastructure testbed. The following are the key contributions:

- We perform System Theoretic Process Analysis (STPA) of Electric Power and Intelligent Control (EPIC) [1].
- We model threats against the safety model of the EPIC testbed to discover *safety-critical attacks*.
- We discover attack paths leading to the execution of *safety critical attacks*.

The novelty of this work lies in the integrated application of threat modelling on STAMP and attack graphs to evaluate the impact of individual vulnerabilities and weaknesses on the overall safety of the CPS. The rest of the paper is organised as follows. Sections II and III introduce the related work and an overview of safety and security methodologies that we use for our analysis. In Section IV we introduce HA-Grid and apply it to a real world use case. Finally, in Section V we summarise our findings and discuss possible improvements.

II. Related Works

The resilience of AC-Microgrids to attacks has been analysed in the literature from several perspectives, in terms of its threat vector or in terms of its consequence. The general consequences of cyber-attacks on grid operations have been summarised in [9]. To trace the attack vector, attacks can be classified as data availability, integrity or confidentiality attacks. Attacks on data availability [21] (e.g., Denial-of-Service (DoS)) transfer malformed packets to the target or flood the network/communication layer by exhausting the routers' processing capacity, network bandwidth, or servers' memory. Integrity attacks [11, 30] can be conducted by modifying the information flowing in the system. Confidentiality attacks

978-1-6654-3254-2/22/$31.00

aim to eavesdrop on the communication network to retrieve information about customers.

The use of System Theoretic Accidents Models and Processes (STAMP) and STPA in conjunction with security methodologies has been previously explored in the literature. Khan et al. [15] use STRIDE to model threats to the STAMP model of a Cyber-Physical System (CPS). Friedberg et al. [8], go further and propose *STPA-SafeSec* as a hybrid framework to model the impact of different types of threats on system components and the overall safety. Although STPA-SafeSec considers the deployment architecture of the CPS, it only considers generic threats and does not take into account component vulnerabilities. In contrast, HA-Grid leverages a mapping to the system architecture to discover the role that individual vulnerabilities (e.g., a buffer overflow on a deployed component) play with respect to the overall safety. *STPA-SafeSec* has been successfully applied to discover high-level safety critical scenarios threatening a micro-grid. [17] uses STPA to analyse the impact of cyber-attacks against an industrial control system. More recently, Khan and Madnick [16] proposed a framework grounded in STPA to identify mitigations against attacks to the CPS. Authors of [6] proposed a methodology based on STPA and simulations of system behaviour to evaluate the impact of different types of attacks against a Communication Based Train Control System, and suggested the use of logical attack graphs to quantify the risk of safety critical attacks. The work presented in this paper, on the other hand, has the objective of discovering and evaluating attack paths leading to hazardous scenarios. Finally, on the front of attack graph generation using MulVal, Stan et al. [26] proposed a rich set of rules to describe complex attacks using MulVal logic attack graph representation.

III. Models for Safety & Security

Several methodologies for safety and security are widely in use today. We briefly describe below those on which our methodology relies.

Safety Analysis. Systems Theoretic Accident Model and Process (STAMP) was introduced to describe the safety model of control systems [18, 19]. STAMP models three aspects of a CPS that are relevant to its safe operation: hazards and safety constraints, the control hierarchy and process models. The control hierarchy and the process model are described in the *Safe Control Structure (SCS)* [6], which is defined as a tuple (C, D, K) where C is the set of components (controllers, actuators, sensors and physical processes), $D = (CA \cup F)$ is the set of control actions and feedback signals and K is a set of subsystems. The latter, group together components according to a logic (e.g., physical closeness) specified by the designer. A *Process Model (PM)* is defined for each controller in the SCS and describes the controller's view of the underlying physical processes. *System Theoretic Process Analysis* was introduced to study the dynamics leading to accidents and their respective losses on a STAMP model [19]. In this paper, we use the same definition of *Accidents* and *Hazards* as proposed in [19]. STPA is a two stage process that sees accidents as the result of the application of *Hazardous Control Actions (HCAs)*. An HCA can happen because the controller has an incomplete or inaccurate view of the underlying physical process (e.g., flaw in process model or feedback) or because the commanded control action is not actuated properly. More specifically, a control action is hazardous when it is either *applied* or *not applied*, applied with the *wrong timing* or for the *wrong duration* [19]. The first step in STPA consists in the identification of the HCAs, whilst the second step comprises the identification of the safe control structures and of the factors leading to the application of HCAs.

Threat Model. We model an attacker that can *tamper* and *spoof* [25] messages comprising *control actions* and *feedback*, and *process models* in the SCS. For simplicity, at this stage we are not considering attacks against the availability. The attacker's objective is to drive the system towards a hazardous state through the application of one or more hazardous control actions to cause accidents and, eventually, cause losses. Formally, we define an *attack* $\mathcal{A}$ as an ordered sequence of N_A *attack steps* such that:

$$\mathcal{A} = \{a_1, \ldots, a_{N_A}\}$$

An *attack step* is a tuple $a_i = (t, e, v)_i$ where t is the type of threat, e is the targeted element in the STAMP model and v is the value injected through *spoofing* or *tampering*. Each attack step is enabled by one or more sets of privileges. We use MulVal [23, 24] to generate attack paths that enable an attacker to acquire the privileges needed to carry out the attack. *MulVal* is a *logic-based network security analyser*, which takes as input the architecture of the system, a list of vulnerabilities that affect its components, and a set of attack objectives to build the attack paths towards the objective. MulVal uses *facts* and *rules* to derive the attacker's progression throughout the computer network. Facts are logic predicates representing security attributes (e.g., privileges, vulnerabilities, network connectivity, etc.), rules, on the other hand, explain the relations among the facts and are expressed through Horn clauses [12]. We have extended MulVal with two categories of rules to model the impact of attacker on the CPS information flow and bind an attack step to the privileges required to actuate it. The first category of rules describes *tampering* with the information flow or with a process model variable x. For example:

$$canTamper(x) \leftarrow execCode(C, root), controlsPMV(C, x) \tag{1}$$

which states that an adversary can tamper with the STAMP element x, if they have obtained high-privilege code execution rights on controller C, and the controller C controls x. Similarly, the adversary can tamper with the element of the STAMP model x if they have obtained high-privilege code execution rights on host C, and C transports (e.g., router, relay, etc.) x:

$$canTamper(x) \leftarrow execCode(C, root), transportsFlow(C, x) \tag{2}$$

The second category of derivation rules we have added describes the privileges an attacker needs to *spoof* x. For

example, an adversary can spoof the element of the STAMP model x if they have obtained access to the OT network where controller C is located, and the controller C controls the element of the STAMP model x:

$$canSpoof(x) \leftarrow netAccess(C, Prot, P), transportsFlow(C, x) \quad (3)$$

The output of MulVal is a graph $AG = (V, E, \mathcal{L}, \mathcal{G})$, where $V = (V_p \cup V_r \cup V_d)$ is the vertex set, with V_d, V_p and V_r being Primitive Facts (PF), Derivation Rules (DR) and Derived Facts (DF) [23]. E, $\mathcal{L}$ and $\mathcal{G} \subseteq V$ are the set of edges, labels and attacker goals respectively.

IV. HA-Grid Analysis on EPIC Testbed

In this section, we elaborate the proposed methodology (HA-Grid) and show its use to analyse the Energy Management System (EMS) of a smart grid test-bed.

The *Electric Power and Intelligent Control (EPIC)* test-bed [1] is a smart grid infrastructure comprising *four* sections: power generation, transmission, micro-grid and smart-home. The latter also includes a secure water treatment (SWaT) test-bed. Applying our analysis to EPIC allows us to model realistic safety and security aspects of a smart grid. The attack paths generated can also be validated on the actual test-bed. The architecture of the test-bed is described in [1, 2] and [14]. In EPIC, the physical process is controlled through a SCADA workstation, which is connected to a master PLC that coordinates other PLCs, each responsible for controlling a different test-bed section (see also Section IV-B). The SCADA workstation is a *Win7* host vulnerable to *EthernalBlue* (CVE-2017-0144), while the PLC controllers run a version of *Dropbear SSH* that is exposed to remote code execution (RCE) vulnerabilities (CVE-2016-7406, CVE-2016-7407, CVE-2016-7408 and CVE-2016-7409). PLCs also expose a network service that enables an attacker with access the same network to upload and run code without authorisation (CVE-2012-6068). Microsoft Windows computers in the system also run a SMB server vulnerable to CVE-2017-0267, CVE-2017-0268 and CVE-2017-0269 [2]. The latter allow the adversary to breach the confidentiality and the availability of messages

A. Overview of HA-Grid

HA-Grid consists of *four* steps. We use STAMP to describe the safety model and STPA to derive the control actions that can drive the CPS towards a hazardous state if applied in a unsafe context (❶). We establish a map between the elements of the STAMP model and the components in the deployed architecture of the CPS (❷). Then, we model *Tampering* and *Spoofing* threats against elements of the STAMP model and derive attacks leading to hazardous scenarios (❸). Finally, we use MulVal to uncover attack paths that enable attackers to acquire the privileges needed to perform these attacks and, thus, drive the CPS towards an unsafe state (❹).

B. Safety Model

The diagram in Figure 1 shows the electrical layout of EPIC with respect to the $four$ loads: *critical* and *non-critical* loads, the $motor-G3$, and the output towards an external system (water treatment plant). The critical load (shown in red- in Figure 1) is located in the Smart Home (SH) section.

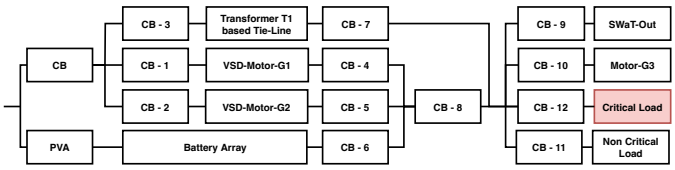

Fig. 1. Power dependencies of the main loads in the EPIC testbed.

Continuous power supply to the critical load (e.g., an ICU ward) is essential to maintain system safety. To this extent, there are are $four$ parallel paths (Figure 1) supplying power to the critical load [1]. $Three$ of these paths, are located in the micro-grid (MG) section:

- $P_1 = CB, CB-1, CB-4, CB-8, CB-12$
- $P_2 = CB, CB-2, CB-5, CB-8, CB-12$
- $P_3 = CB-6, CB-8, CB-12$.

For MG to function in islanded mode, the power to the critical load must be provided by one of the two generators or by the battery array. Paths P_1 or P_2 are considered *active* if the power is supplied by the first or the second generator respectively. At the same time, P_3 is active if the load is powered through the battery. A $fourth$ path

- $P_4 = CB, CB-3, CB-7, CB-12$

powers the load through the Transmission Network (TN) [1]. The circuit breaker CB in the *generation* section of EPIC is controlled by a PLC ($gPLC$). Similarly circuit breakers in the TN ($CB-3, CB-7$ and $CB-8$) are regulated by $tPLC$. The PLC located in the MG ($mPLC$) controls $CB-4$, $CB-5$ and the smart-home ($sPLC$) controls $CB-12$.

We employ STAMP to model the safety aspects of the testbed. The Control Structure is shown in Figure 2 and consists of a total of 34 controllers, sensors and actuators. The control structure focuses on the control flow of the EMS and does not show details intrinsic to the physical process such as the flow of energy between power subsystems. Electrical dependencies in the physical process are shown in Figure 1. The EMS is supervised by a human operator who controls the functioning of the micro-grid through the SCADA workstation, which in turn, controls the main PLC and the photovoltaic array (PVA). The main PLC acts as coordinator for $four$ lower level controllers, one for each stage of the EMS system. Each of these PLCs controls part of the EMS through dedicated Intelligent Electric Devices (IEDs). The IEDs are low level controllers that measure frequency and voltage through an Advanced Measuring Interface (AMI) and control the physical network by opening and closing the 12 Circuit Breakers (CB). The IED can open a circuit breaker following an instruction received from the commanding PLC or to adapt to frequency deviations. Circuit breakers $CB-1$, $CB-2$ and $CB-6$ are electrical safety interlocks, are not controlled by operators and are opened when an unbalance is detected. At the same time CB, $CB-4$, $CB-5$, $CB-7$, $CB-8$ and $CB-12$ are

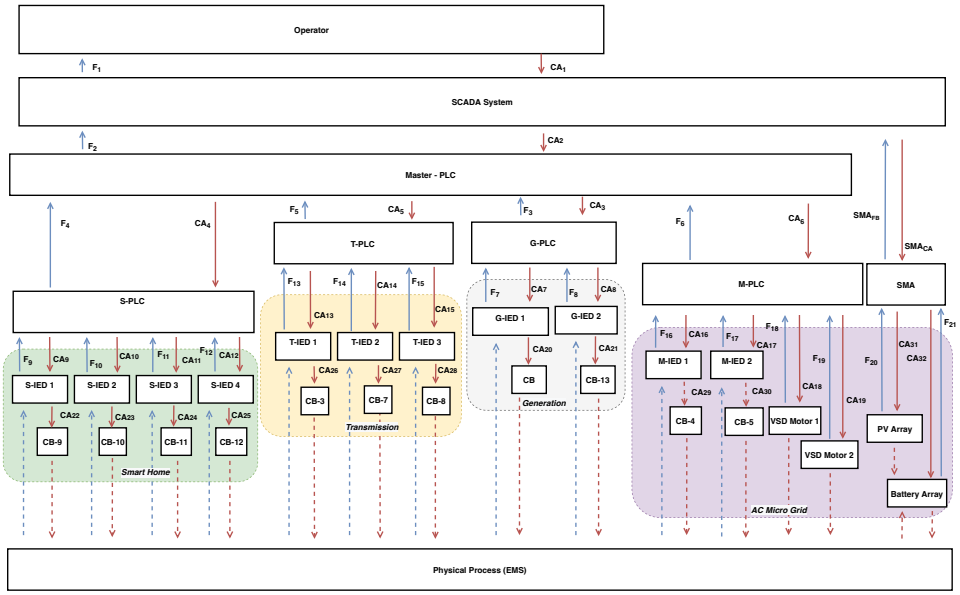

Fig. 2. STAMP model of EPIC

controlled by operators but they can also open automatically, as result of a nuisance tripping, to preserve operational safety. A SMA inverter is controlled by the SCADA workstation and regulates the photovoltaic array in the micro-grid. For simplicity, we do not show the AMI in the SCS shown in Figure 2. The process models of SCADA, PLCs and IEDs contain the variables and the control algorithms needed to regulate their respective stages of the EMS.

We leverage expert knowledge to identify system level hazards and relevant losses and the accidents that they can cause. We consider the following losses: (L_1) Loss of lives, (L_2) Loss of safety of the operating environment, (L_3) Major damages to equipment, and (L_4) Economic losses.

We summarise considered accidents in Table I. A_1, A_2 lead, respectively to L_1 (e.g., deriving from the power being cut to the critical load), and to L_1 and L_2. (L_1). A_3 leads to losses L_1, L_2, L_3 and L_4. A_4 leads to losses L_2 and L_3, while A_5 leads to L_3 and A_6 leads to L_4. $A_1 - A_4$ are the accidents with the highest losses as they threaten human lives and the safety of the operating environment. The safety analysis proceeds with the investigation of system hazards that can lead to accidents $A_1 - A_6$. Found hazards are then refined to identify the responsible controllers. The objective of this step is the definition of safety requirements to enforce on responsible controllers to avoid the hazards. For the sake of brevity, we focus on the hazards that can lead to accidents A_1 and A_2. Following an analysis of the electrical dependencies of the critical load (Figure 1), we find a total of 13 hazards that lead to A_1 and A_2 (Table II).

We now apply STPA to identify the HCAs leading to $H_1 - H_{13}$ which, in turn, can cause A_1 and A_2.

For brevity, we only report the following 7 HCAs:

- HCA_1: CA_{20} is **applied** with value *open* when the *critical load* is powered through paths P_1, P_2 or $P_4 \longrightarrow H_1$.
- HCA_2: CA_{20} is **not applied** with value *close* when the *critical load* has to be powered P_1, P_2 or $P_4 \longrightarrow H_1$.
- HCA_3: CA_{20} is **applied** with value *close* making the breakers configuration unsafe $\longrightarrow H_{12}$.
- HCA_4: CA_{25} is **applied** with value *open* $\longrightarrow H_2$.
- HCA_5: CA_{25} is **not applied** with value *close* $\longrightarrow H_2$.
- HCA_6: CA_{25} is **applied too late** (*close*) $\longrightarrow H_2$.
- HCA_7: F_1 is **applied** with wrong parameters $\longrightarrow H_{13}$.

The second step of STPA entails the discovery of causal factors leading to the HCAs throughout the whole control hierarchy (Figure 2). These can be caused by a design flaw in the controller's process model or by miscommunications

TABLE I
ACCIDENTS

Acc	Description	Loss(es)
A_1	Interruption of power supply to *critical* load.	L_1
A_2	Nuisance tripping.	L_1, L_2
A_3	Physical damages to motors.	L_2, L_3
A_4	Physical damages to batteries.	L_2, L_3
A_5	Physical damages to IED.	L_3
A_6	Interruption of power supply to *non-critical* load.	L_4

TABLE II
HAZARDS LEADING TO ACCIDENTS A_1, A_2.

Hazard	Description
H_1	The main circuit breaker CB is open while BA is discharged.
H_2	The circuit breaker $CB-12$ is open.
H_3	$CB-8$ is open while power is supplied through P_1, P_2 or P_3..
H_4	$CB-7$ is open while the power is supplied through P_3.
H_5	$CB-4$ is open while the power is supplied through P_1.
H_6	$CB-5$ is open while the power is supplied through P_2.
H_7	$CB-6$ is open while the power is supplied through P_3..
H_8	$CB-3$ is open while the power is supplied through P_4.
H_9	$CB-1$ is open while the power is supplied through P_1 .
H_{10}	$CB-2$ is open while the power is supplied through P_2.
H_{11}	Delay in closing breaker causes *nuisance tripping*.
H_{12}	Unsafe configuration of breakers
H_{13}	SCADA not synchronised with Physical Process.

with feedback/actuation components [19]. HCA_1, HCA_2 and HCA_3 concern the hazardous application of the control action CA_{20} with parameter *open*. The control action is issued by $G-IED1$ following the interpretation of a command from $gPLC$, or as protection, in case of unsafe readings (e.g., low frequency). Likewise, commands issued by $gPLC$ can be the result of autonomous decisions (e.g., given current state and inputs), or of a direct command from a higher controller ($Master-PLC$). The latter is controlled by the $SCADA$ workstation. Formally, the application of HCA_1, HCA_2 and HCA_3 can arise as a result of flaws within one of the following elements of the control hierarchy:

- **Process Model** of controllers: $G-IED1$, $gPLC$, $Master-PLC$, and, $SCADA-Workstation$.
- **Control Actions**: CA_1, CA_2, CA_3, CA_7, CA_{20}.
- **Feedback**: F_1, F_2, F_3, F_7.

Using the same logic, we derive that the application of HCA_4, HCA_5 and HCA_6 can take place as result of flaws within one of the following elements of the control hierarchy:

- **Process Model** of controllers: $S-IED4$, $sPLC$, $Master-PLC$, and, $SCADA-Workstation$.
- **Control Actions**: CA_1, CA_2, CA_4, CA_{12}, CA_{25}.
- **Feedback**: F_1, F_2, F_4, F_{12}.

C. Safety-Critical Attacks

Having identified the HCAs, we now discover threats against elements in the STAMP model that would enable attackers to arbitrarily cause their actuation. For each HCA and for each controller in its control hierarchy, the adversary can *spoof* issued control actions and received feedback. They can also *tamper* issued control actions, received feedback and process model variables. In the first case (spoofing) the attacker impersonates a component to issue a fake control action or feedback. In the latter (tampering), they change the values of a control action, feedback signal or process model variable to deceive the controller in thinking that the process is in a different state. An example of a successful attack leading to H_3 is spoofing (or tampering with) the control action CA_2 aimed to open $CB-12$. Table III, enumerates the possible attack steps that an attacker can execute against the SCS to change CA_{25}, which controls $CB-12$. By controlling CA_{25}, the attacker can apply hazardous control actions $4, 5$ or 6. Some of the attack steps are higher in the hierarchy, meaning that they can also be applied to carry out broader attacks. For example, the attack $\mathcal{A}_1 = \{a_2\}$ requires only one attack step (a_2) and enables the attacker to apply HCA_4, HCA_5 or HCA_6 depending on the value of cmd (*open breaker* or *close breaker*, in this case). More complex attacks can involve more attack steps. The attack $\mathcal{A}_2 = \{a_2, a_{14}\}$ enables a skilled attacker to $open/close$ breaker $CB-12$ while, at the same time, blind the SCADA workstation.

D. Generation of Attack Paths

After identifying the safety critical attacks, we now leverage MulVal to infer the attack paths leading to their execution. The process, shown in Figure 3, takes *four* inputs: the system architecture, a set of security properties, a set of safety properties that the attacker aims to violate, and a function that maps the elements of the safety model (control logic) to components in the system architecture. We have manually described the *architecture* and the *security properties*, following the specifications in [1], [2] and [14]. However, MulVal provides the tool to combine [3] the output from network mapping (e.g., nmap [20]) and assessment (e.g., OpenVAS[10], Nessus [27]) tools to automatically describe the structure of the system. The third input in Figure 3 is the list of attack steps, identified in Section IV-C. As the attack steps are defined on the control logic, a map between the control logic and the system architecture (*fourth* input) is also needed. We define the *fact* in (Eq. 4) to bind the STAMP process model variable $e \in PMV_i$ to the responsible host $h \in H$. With PMV_i the set of process model variables of the controller $c_i \in C$ and H the set of deployed components.

$$controlsPMV(h, e) \tag{4}$$

Similarly, (Eq. 5) binds the control action (or feedback) $d \in D$ to $h \in H$:

$$transportsFlow(h, d) \tag{5}$$

Using MulVal, we derive the attack graph for the EPIC testbed to identify the attack paths leading to hazard H_2 (HCA_4, HCA_5, and HCA_6). We assume that IED devices are secure and that the attacker does not have physical access to sensors and actuators (i.e., cannot carry out physical attacks). Under these assumptions, the attacker can cause H_2 with any attack

TABLE III
ATTACK STEPS

Attack Step	Description	HCA
a_1	$\{(spoof, CA_{12}, cmd)\}$	$4, 5, 6$
a_2	$\{(tamper, CA_{12}, cmd)\}$	$4, 5, 6$
a_3	$\{(spoof, CA_4, cmd)\}$	$4, 5, 6$
a_4	$\{(tamper, CA_4, cmd)\}$	$4, 5, 6$
a_5	$\{(spoof, CA_2, cmd)\}$	$1-10$
a_6	$\{(tamper, CA_2, cmd)\}$	$1-10$
a_7	$\{(spoof, CA_2, cmd)\}$	$1-10$
a_8	$\{(tamper, CA_2, cmd)\}$	$1-10$
a_9	$\{(tamper, sPLC, memory)\}$	$4, 5, 6$
a_{10}	$\{(tamper, Master-PLC, memory)\}$	$1-10$
a_{11}	$\{(tamper, SCADA, memory)\}$	$1-10$
a_{12}	$\{(spoof, F_{12}, \bar{v})\}$	$4, 5, 6, 11$
a_{13}	$\{(tamper, F_{12}, \bar{v})\}$	$4, 5, 6, 11$
a_{14}	$\{(spoof, F_4, \bar{v})\}$	$4, 5, 6, 11$
a_{15}	$\{(tamper, F_4, \bar{v})\}$	$4, 5, 6, 11$
a_{14}	$\{(spoof, F_2, \bar{v})\}$	$1-11$
a_{15}	$\{(tamper, F_2, \bar{v})\}$	$1-11$
a_{16}	$\{(spoof, F_1, \bar{v})\}$	$1-11$
a_{17}	$\{(tamper, F_1, \bar{v})\}$	$1-11$

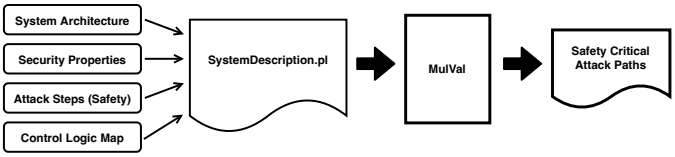

Fig. 3. Derivation of Attack Graph

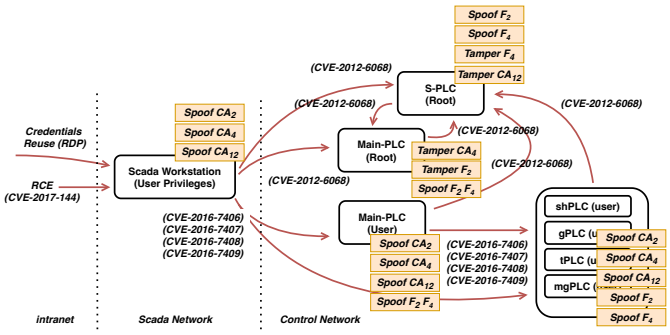

Fig. 4. Resulting Attack Graph

$\mathcal{A}_i = \{a_i\}$ with $i \in [1, \ldots, 17]$. The attack graph resulting from the analysis using $[a_1, \ldots, a_{17}]$ as the attack objectives is summarised in Figure 4.

The first interesting result is that the feasibility of each attack step ($[a_1, \ldots, a_{17}]$) depends on the system configuration and posture. Only a subset of these steps can be effectively implemented. We assume that the adversary has already gained a foothold in the network and has visibility of the SCADA workstation which runs a RDP daemon. They can access the SCADA workstation by exploiting *Eternal Blue* ($CVE - 2017 - 144$) or through credential re-use [4]. The attacker the network visibility the SCADA workstation has of the PLCs to craft packets and spoof control actions CA_2, CA_4 and CA_{12}. At the same time, with current privileges and in the absence of *Privilege Escalation* (PE) vulnerabilities, they cannot tamper with the workstation memory to change the feedback given to the operator or downstream control actions. From the SCADA workstation, the attacker can also exploit $CVE{-}2012{-}6068$ on $S{-}PLC$ and $Main{-}PLC$ to execute code, with high privileges, on these devices. By doing so they can tamper with controller memory and falsify control actions CA_4 and CA_{12} and feedback F_2 and F_4. They can also use leverage CVEs $2016 - 7406$, $2016 - 7407$, $2016 - 7408$ and $2016 - 7409$ to pivot to other devices. Given the topology of the control network, the adversary can pivot in any direction (Figure 4).

E. Discussion

Applying our proposed methodology HA-Grid on EPIC testbed, we found that given the current system architecture and security posture (e.g., vulnerabilities, network configurations, etc.), only a subset of threats identified during the threat modelling activity on the STAMP model are concretely exploitable by an attacker. With the enumerated attack paths, we determined that the SCADA Workstation is the "crown jewel" and that gaining access to the SCADA Workstation is sufficient to perform single-step safety critical attacks. A skilled adversary, however, might instead aim to execute covert attacks that require multiple steps to be less easily detected and delay incident response operations. Under the current configuration, the sole unprivileged access to the SCADA workstation is not sufficient as the attacker also needs the privileges to *spoof* or *tamper* the operator feedback. An example of such a covert multi-step attack is $A_2 = \{a_2, a_{15}\}$ (Table III), which requires privileged execution on at least one of the PLCs . Finally, the analysis of the attack graph suggests that mitigating $CVE - 2012 - 6068$ can prevent the actuation of the totality of *tampering* attacks. Similarly, we observe that mitigating $CVE - 2017 - 144$ on the SCADA workstation, together with stricter policies on operator accesses can protect against actions $[a_1, \ldots, a_{17}]$. Furthermore, installing devices such as network diodes can prevent the attacker from spoofing control actions from the SCADA server to lower level controllers.

The methodology explored in this work presents two limitations. First, STPA is mainly driven by expert knowledge and requires an important amount of manual input. The issue is acknowledged by the community and several efforts have been done to automate some if its aspects [28] to scale towards wider and more complex distributed system more smoothly. The second limitation of our work lies in the fact that attack graphs do not consider 0-day exploits. This is also acknowledged by the community and works have been done to understand how these unknown vulnerabilities would change the shape of the attack graph [29, 31] and [5].

V. Conclusions

We have introduced HA-Grid, a methodology that enables to perform integrated safety and security analysis of smart grid infrastructures. HA-Grid is grounded in STAMP and STPA and starts with the definition of a set of hazards leading to accidents and losses. We study the impact of *tampering* and *spoofing* threats against the high-level safety model of the smart grid and identify sequences of *attack steps* that the attacker can execute to cause a hazard. Finally, we leverage a map between the safety model and the deployed architecture of the system to identify the attack paths that lead to violation of high-level safety properties. The operation helps security analysts to determine the weaknesses and vulnerabilities that should be addressed first to preserve safety. We have applied HA-Grid to discover safety critical attack paths on the EPIC [1] testbed. Future works involves automating the safety and threat modelling steps by reducing the amount of manual inputs required for the analysis.

Acknowledgements

This work has been partially supported by the EPSRC Centre for Doctoral Training in High Performance Embedded and Distributed Systems (Grant EP/L016796/1) and has received funding, in part, from the European Union's Horizon 2020 research and innovation programme under grant agreement No 830927 (Concordia).

References

[1] Sridhar Adepu, Nandha Kumar Kandasamy, and Aditya Mathur. Epic: An electric power testbed for research and training in cyber physical systems security. In *Computer Security*, pages 37–52. Springer, 2018.

[2] Sridhar Adepu, Nandha Kumar Kandasamy, Jianying Zhou, and Aditya Mathur. Attacks on smart grid: Power supply interruption and malicious power generation. *International Journal of Information Security*, 19(2):189–211, 2020.

[3] ArgusLab. Mulval documentation. URL https://people.cs.ksu.edu/~xou/argus/software/mulval/readme.html.

[4] BBC. Ukraine power cut 'was cyber-attack', 2022. URL https://www.bbc.co.uk/news/technology-38573074.

[5] Leyla Bilge and Tudor Dumitraş. Before we knew it: an empirical study of zero-day attacks in the real world. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 833–844, 2012.

[6] Luca Maria Castiglione and Emil C Lupu. Hazard driven threat modelling for cyber physical systems. In *Proceedings of the 2020 Joint Workshop on CPS&IoT Security and Privacy*, pages 13–24, 2020.

[7] Dragos. Chernovite's pipedream malware targeting industrial control systems (ics), 2022.

[8] Ivo Friedberg, Kieran McLaughlin, Paul Smith, David Laverty, and Sakir Sezer. Stpa-safesec: Safety and security analysis for cyber-physical systems. *Journal of information security and applications*, 34:183–196, 2017.

[9] Pudong Ge, Fei Teng, Charalambos Konstantinou, and Shiyan Hu. A resilience-oriented centralised-to-decentralised framework for networked microgrids management. *Applied Energy*, 308:118234, 2022.

[10] Greenbone. Openvas. URL https://www.openvas.org.

[11] Martin Higgins, Wangkun Xu, Fei Teng, and Thomas Parisini. Cyber-physical risk assessment for false data injection attacks considering moving target defences. *arXiv preprint arXiv:2202.10841*, 2022.

[12] Alfred Horn. On sentences which are true of direct unions of algebras1. *The Journal of Symbolic Logic*, 16 (1):14–21, 1951.

[13] Dragos Inc. Crashoverride analysis of the threat to electric grid operations 2022, 2022. URL https://www.dragos.com/wp-content/uploads/CrashOverride-01.pdf.

[14] Nandha Kumar Kandasamy, Sarad Venugopalan, Tin Kit Wong, and Nicholas Junming Leu. An electric power digital twin for cyber security testing, research and education. *Computers and Electrical Engineering*, 101: 108061, 2022.

[15] Rafiullah Khan, Kieran McLaughlin, David Laverty, and Sakir Sezer. Stride-based threat modeling for cyber-physical systems. In *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pages 1–6. IEEE, 2017.

[16] Shaharyar Khan and Stuart E Madnick. Cyber-safety: A system-theoretic approach to identify cyber-vulnerabilities & mitigation requirements in industrial control systems. *IEEE Transactions on Dependable and Secure Computing*, 2021.

[17] Shaharyar Khan, Stuart Madnick, and Allen Moulton. Cyber-safety analysis of an industrial control system for chillers using stpa-sec. 2018.

[18] Nancy Leveson. A new accident model for engineering safer systems. *Safety science*, 42(4):237–270, 2004.

[19] Nancy G Leveson. *Engineering a safer world: Systems thinking applied to safety*. The MIT Press, 2016.

[20] Gordon Fyodor Lyon. *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. Insecure. Com LLC (US), 2008.

[21] Jelena Mirkovic and Peter Reiher. A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2):39–53, 2004.

[22] Mitre. Blackenergy software s0089 mitre att&ck 2022, 2022. URL https://attack.mitre.org/software/S0089/.

[23] Xinming Ou, Sudhakar Govindavajhala, Andrew W Appel, et al. Mulval: A logic-based network security analyzer. In *USENIX security symposium*, volume 8, pages 113–128. Baltimore, MD, 2005.

[24] Xinming Ou, Wayne F Boyer, and Miles A McQueen. A scalable approach to attack graph generation. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 336–345, 2006.

[25] Adam Shostack. Experiences threat modeling at microsoft. *MODSEC@ MoDELS*, 2008, 2008.

[26] Orly Stan, Ron Bitton, Michal Ezrets, Moran Dadon, Masaki Inokuchi, Ohta Yoshinobu, Yagyu Tomohiko, Yuval Elovici, and Asaf Shabtai. Extending attack graphs to represent cyber-attacks in communication protocols and modern it networks. *IEEE Transactions on Dependable and Secure Computing*, 2020.

[27] Tenable. Nessus. URL https://www.tenable.com/products/nessus.

[28] John P Thomas IV. *Extending and automating a systems-theoretic hazard analysis for requirements generation and analysis*. PhD thesis, Massachusetts Institute of Technology, 2013.

[29] Lingyu Wang, Tania Islam, Tao Long, Anoop Singhal, and Sushil Jajodia. An attack graph-based probabilistic security metric. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 283–296, 2008.

[30] Qingyu Yang, Jie Yang, Wei Yu, Dou An, Nan Zhang, and Wei Zhao. On false data-injection attacks against power system state estimation: Modeling and countermeasures. *IEEE Transactions on Parallel and Distributed Systems*, 25(3):717–729, 2013.

[31] Mengyuan Zhang, Lingyu Wang, Sushil Jajodia, Anoop Singhal, and Massimiliano Albanese. Network diversity: a security metric for evaluating the resilience of networks against zero-day attacks. *IEEE Transactions on Information Forensics and Security*, 11(5):1071–1086, 2016.