Trustworthy Computer Security Incident Response for Nuclear Facilities

Mislav FINDRIK[1], Ivo FRIEDBERG[1], Ewa PIATKOWSKA[1], Paul SMITH[1], and Jae-Gu SONG[2]

1. Center for Digital Safety and Security, AIT Austrian Institute of Technology, Donau-City-Straße 1, Vienna 1220, Austria ({firstname.lastname}@ait.ac.at)

2. I&C Human Factors Research Division, Korea Atomic Energy Research Institute, Daejeon, Korea (jgsong@kaeri.re.kr)

Abstract: New forms of advanced computer security threat are targeting critical infrastructures, including nuclear facilities. These threats use sophisticated and stealthy methods to target a specific infrastructure, with the aim of causing operational consequences. For nuclear facilities, this could involve compromising Instrumentation and Control (I&C) systems that underpin nuclear security and safety functions. In this context, effective and rapid incident response is necessary to mitigate and contain the potential effects of a cyber-attack. Incident response includes a detection and analysis phase, wherein incidents are identified, and their effects are understood. This phase involves reasoning about the state of systems, i.e., whether they are compromised or not, based on potentially unreliable sources of information. In this paper, we motivate and present a high-level architecture to support reasoning for incident response, based on unreliable detection capabilities. With an example nuclear-relevant scenario, we indicate how its reasoning component can be realized with the use of Evidential Networks – a graph structure that represents knowledge about a target domain, and supports inference using unreliable information sources.

Keyword: Computer Security, Incident Response, Machine-based Reasoning

1 Introduction

Nuclear Power Plant (NPP) Instrumentation and Control (I&C) systems are being increasingly digitalized. This has many benefits, but introduces potentially new computer security threats. Recent incidents, notably in the Ukraine in December 2015[1], have demonstrated that so-called Advanced Persistent Threats (APTs) can result in disruptions to physical systems and processes – in the Ukrainian case, a major blackout. Arguably, the same potential exists for NPPs, whereby computer security threats could result in the compromise of nuclear security and safety objectives.

In this context, it is important that members of a Security Operations Center (SOC) at an NPP can rapidly and effectively respond to Indicators of Compromise (IoCs) – evidence that a threat is taking place. To achieve this goal, appropriate systems need to be deployed that can monitor, detect, and determine the root cause of IoCs. Typically, Intrusion Detection Systems (IDSs), which are coupled with a Security Information and Event Management (SIEM) solution, support this functionality. These systems have been extensively deployed, e.g., in enterprise environments; however, they have several shortcomings, and have seen limited use, for application at NPPs. These shortcomings include a limited capacity to leverage data from I&C systems equipment, such as Programmable Logic Controllers (PLCs), and nuclear processes; potentially high false positive and negative rates, which reduces the trustworthiness of such systems; a strong focus on the correlation of events, as opposed to determining causality, which is critical for effective incident response; and a lack of guidance on deployment and usage strategies for NPPs.

In this paper, we present research towards a system for trustworthy computer security incident response for NPPs. To motivate our research, we discuss the nature of modern advanced computer security threats that could target NPPs. To detect and respond to these threats, it is necessary to reason about multiple IoCs, to determine whether they result in systems being compromised. We present a high-level architecture whose purpose is to enable this form of reasoning. Building on the architecture, we propose Evidential Networks (ENs)[2] as an approach to reasoning about IoCs when detection capabilities are known to be unreliable. An example threat that is targeting an I&C system, which controls reactor cooling, is used to demonstrate how ENs can be applied to reasoning.

2 Computer Security Threats

Here, we discuss a specific type of modern computer security threat that targets critical infrastructures, such as nuclear facilities. The nature of these threats forms part of the motivation for our research.

In recent years, several cyber-attacks have occurred that target a specific organization, and use technically sophisticated and stealthy means of realizing a malicious intent. They are often executed over extended periods of time, such as several months, and implement several attack steps. These attacks are widely referred to as *Advanced Persistent Threats (APTs)*[3]. Normally, the main goal is to commit data theft for espionage or fraud reasons – an information security concern.

An increasing number of high-profile computer security incidents have shown how APTs can be used to compromise the physical processes that are controlled by Supervisory Control and Data Acquisition (SCADA) and Instrumentation and Control (I&C) systems. Prominent incidents include the Stuxnet virus[4], which resulted in damage to a centrifuge at a nuclear fuel processing facility, and a major power blackout in the Ukraine[1]. In both cases, the attackers implemented a so-called *ICS Cyber Kill Chain*[5] – a "standard" APT that includes additional steps to compromise Industrial Control Systems (ICSs) that manage physical processes.

To the best of our knowledge, aside from the Stuxnet virus, there have been no such incidents, i.e., cyber threats that directly manipulate physical processes using compromised I&C systems, in the nuclear sector. To achieve this, we anticipate that a threat source will need to realize an attack that incorporates both cyber *and* physical activities, as was the case for the Stuxnet attack. (We suggest this because of the particularly stringent computer security requirements for nuclear facilities that restrict digital communication with critical I&C systems[6].) For example, this can involve social engineering to trick or coerce an insider (e.g., an employee or contractor) to deploy malicious software, e.g., using removable media, that gives an attacker a cyber presence within a facility. Moreover, this could involve compromising Physical Protection Systems (PPSs) using cyber means.

As the steps of an APT are executed, *Indicators of Compromise (IoCs)* – i.e., evidence of the presence of an intrusion – should become observable. For the threats discussed herein, IoCs can take several forms: in the cyber domain, antivirus signatures, malicious Internet Protocol (IP) addresses, etc. can indicate the presence of an intrusion. Erroneous or anomalous process behaviour could be used to evidence the existence of an intrusion. Additionally, PPSs could provide indications that a physical incursion is taking place, as part of an attack that incorporates cyber and physical steps.

A key aim for incident response is to detect these IoCs and relate them to a specific APT. Moreover, this should be done as early as possible in the kill chain, such that an attacker is not able to compromise I&C systems that can be used to manipulate processes that are nuclear safety and security critical.

3 Incident Response Architecture

In this section, we present a high-level architecture to support incident response in nuclear facilities, which is depicted in Fig. 1. The aim of this architecture is to support operators as they respond to the computer security incidents that are described in Section 2. The architecture is shown in the context of the five-step incident response strategy from the National Institute for Standards and Technology (NIST) in Special Publication 800-61[7]. It is intended to primarily support the *Detection and Analysis* and *Containment* incident response activities. The aim of *Detect* activities is to identify an incident, such as a cyber-attack, based on IoCs. Furthermore, having detected an incident, *Analysis* activities aim to understand the nature of an attack and how much damage it has caused. With an understanding of an ongoing incident, the next major activity is to *Contain* it, such that (further) damage cannot be caused. Another containment goal is to limit the capability of an attacker to propagate further. In this work, we do not focus on *Eradication and Recovery* activities, which aim to remove the traces of a cyber-attack, patch systems, and return to normal operation. In what follows, we provide a brief introduction to the main components of the architecture presented in Fig. 1: *Detectors*, the *Reasoning Engine*, *Containment Engine*, and *Remedial Actions*.

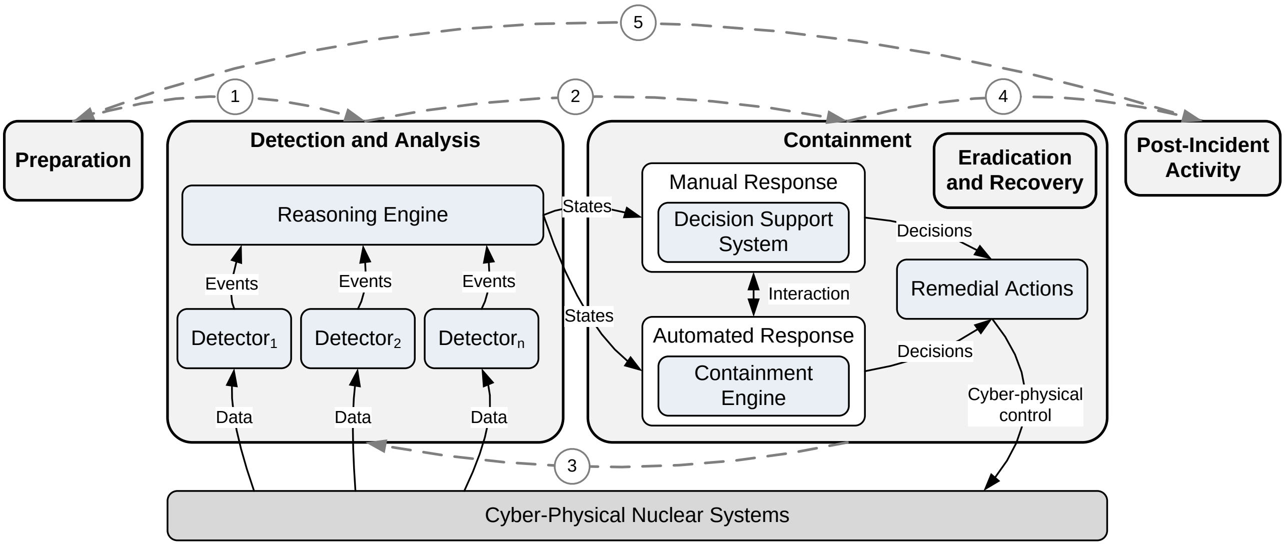

Fig. 1 Overview of the incident response system in relation to the phases of a five-stage incident response cycle

3.1 Detectors

A nuclear facility is a large cyber-physical system of systems. These systems generate *data* that can be used as a basis to detect a cyber-attack. For example, data can be collected from computer networks about the traffic that is being communicated over it. Other digital assets, such as Personal Computers (PCs), Programmable Logic Controllers (PLCs) and computer networking equipment, generate logging data that can be used to detect attack behaviour. Similarly, data regarding the physical processes that are under control in a nuclear facility can be used to indicate the presence of a cyber-attack. Such data can be gathered from digital sensors and control systems, for example.

This data can be used by *Detectors* to identify the presence of a cyber-attack. In short, detectors take *data* as input and generate an *event*, if malicious behaviour is determined to be evident in the data. For computer security, such detectors often take the form of an Intrusion Detection System (IDS)[8]. There are several approaches to intrusion detection that differ in the location that data are collected (e.g., *host* versus *network* data) and the approach used for detection (*signature* or *anomaly*-based detection), for example. Recent research has investigated the use of data that is collected from physical processes to detect the effects of a cyber-attack[9]. We propose that a combination of detection systems that identify malicious behaviour in cyber and physical systems is important to detect the advanced cyber-attacks that are discussed earlier.

A common characteristic of all these detection systems is uncertainty about the correctness of their results. More specifically, the uncertainty of a detector’s correctness can be characterized by a *false positive* and *false negative* rate. The former defines the rate at which a detector incorrectly identifies benign behaviour as malicious, whereas the latter describes the rate that malicious behaviours are not correctly detected. This is a well-understood problem, and presents a major challenge for operators that make use of these technologies.

In many cases, an operator is tasked, either explicitly or implicitly, with configuring the parameters of detectors to manage these two rates. The aim is to find a trade-off between a manageable false positive rate, such that operators are not overwhelmed with false alarms, and an appropriate false negative rate that does not leave a system exposed. Moreover, a major challenge is determining the significance of events that are generated by detectors, given that they are often numerous and potentially unreliable. A part of this challenge is reasoning about detection events to determine whether they characterize the existence of an APT or are caused by other phenomena, such as faults.

3.2 Reasoning Engine

In our architecture, we propose the use of a *Reasoning Engine* to derive insights about the root cause of events that are generated by *Detectors* – these insights are expressed as system states (e.g., whether systems are in a compromised, faulty or normal state) that are described by the detected events. The aim is to support operator-driven or automated decision making regarding suitable containment actions.

Contemporary computer security tools, such as Security Information and Event Management (SIEM) solutions, offer similar – but different – functionality. For example, the Open-Source Security Information Management (OSSIM) SIEM includes a *Correlation Engine*. To use this engine, an operator defines *Correlation Directives* that describe the way that events from a variety of sources should be correlated to identify a threat. As more events, of the same or different type, are correlated, increased confidence about the existence of a threat is gained, leading to an alarm being generated for operator consideration. Thresholds are used to determine whether the number of observed events are significant and should be escalated in importance. In this way, the number of false positive events from detectors, which an operator must attend to, can be reduced.

In contrast, the proposed *Reasoning Engine* yields hypotheses, including a belief in their correctness, about a system's state, using the events that are generated by *Detectors*. The belief in hypotheses aims to account for detector uncertainty (and any uncertainty about the knowledge that has been modelled and is used by the *Reasoning Engine*). We suggest this difference can be important, as APTs typically do not generate numerous detectable events, and using simple thresholds could result in attacks not being brought to the attention of an operator. Instead, providing a set of hypotheses about system state and a belief in their correctness is more suited to managing and representing *Detector* uncertainty. Providing a belief in the certainty of a hypothesis can help decision makers reason about whether to initiate potentially expensive or risky remedial actions – policies can be written that guide decision making, based on the certainty of results from the *Reasoning Engine* and the criticality of the systems affected. Furthermore, by focusing on inferring system states, rather than identifying threats, we suggest that short-term remedial containment actions can be more readily derived. (A deep understanding of the threat is arguably more important for the *Eradication* and *Recovery* incident response steps.)

We anticipate that a combination of approaches – based on the correlation of events to identify noteworthy threats and reasoning about causal factors of system states – should be applied by practitioners. Understanding the relationship between these approaches, and how they can be used in conjunction, is a matter for further investigation. To implement the *Reasoning Engine*, we are considering the use of evidential networks, which are presented in Section 4.

3.3 Containment Engine

Having detected an incident and understood how it affects system state, it should be contained in a timely and effective manner. There may be several containment actions that can be applied, depending on the incident and the consequences associated with their use. For example, the nuclear security and safety consequences of executing (or not) containment actions must be evaluated against operational availability requirements that could be affected by shutting down systems, for example.

Normally, these decisions are made by plant operators. There may be circumstances, considering computer security threats, that some automated decision making regarding which containment actions to implement may be desirable. For example, if it understood that nuclear safety critical systems have been compromised, automatic decisions to immediately shutdown systems or block certain actions could be necessary. The purpose of the *Containment Engine* is to determine what course of action to take, given the hypothesis set and beliefs that is generated by the *Reasoning Engine*.

We are exploring the use of *multi-criteria decision making*[10] as means to realize this component. In short, given a system state, potential containment actions and a payoff associated with their use, multi-criteria decision making determines the most beneficial actions to apply. These criteria must be evaluated and programmed a priori, so that run-time decisions can be made. One decision that could be taken is to notify an operator, rather than initiating containment actions.

The relationship between manual and automated decision making for containment is an important topic for further research.

3.4 Containment Actions

The actions that the containment engine will select are primarily intended to limit the damage associated with a computer security incident. For nuclear I&C systems, the nature of the containment actions will depend on the criticality of the nuclear functions that are affected. For example, for systems that realize safety-critical functions, such as reactor regulation and cooling systems, the presence of malicious software should result in a reactor trip. The behaviour of systems when compromised by a cyber-attack is non-deterministic and could result in a nuclear or radiation accident. Different containment actions could be implemented for non-safety and security critical functions and systems. For example, adaptation of an I&C system's control behaviour and network configuration, for example, could be used to limit the capacity for an attacker to realize their objectives.

4 Reasoning with Evidential Networks

In this section, we introduce *Evidential Networks (ENs)*[2] – an approach to implementing the *Reasoning Engine*, discussed in Section 3.2. To demonstrate how an EN can be used to reason about the state of I&C systems, which could be compromised by a sophisticated cyber-attack, we present a threat scenario and show how an EN can be constructed to reason about the state of a system, based on the events that are generated by *Detectors*.

4.1 Evidential Networks

An evidential network is a graph structure for knowledge representation and inference. Nodes of the evidential network are system variables and their relationships are expressed by belief functions. Formally, an evidential network is defined as a tuple:

$$EN = \{V, \Theta_V, M_V, \oplus, \downarrow\},$$

where $V = \{x_1, x_2, \dots, x_n\}$ is a set of all the variables in the system and $\Theta_V = \{\Theta_x : x \epsilon V\}$ is the set of frames of these variables. Frame Θ_x defines a finite set of possible values of variable x. Elements of the frame are assumed to be mutually exclusive and exhaustive.

The set $M_V = \cup \{M_D : D \subseteq V\}$ is a collection of all the mass functions that describe relations between system variables. A mass function $m: 2^{\Theta_\mathrm{x}} \rightarrow [0,1]$ is a function mapping the frame Θ_x into the interval $[0,1]$, that satisfies $\sum_{A \subseteq \Theta_\mathrm{x}} m(A) = 1$.

The beliefs about the actual value of the variable x can be expressed on the subsets of its frame Θ_x, thus a mass function is defined on 2^{Θ_x}, i.e., the power set of Θ_x, which includes all possible subsets of Θ_x. This provides a richer description of the variables and allows uncertainty to be represented.

Inference within the evidential network is achieved by two operators, called *combination* and *marginalisation*, denoted as $\oplus$ and $\downarrow$, respectively.

Combination is defined by Dempster's rule of combination to describe the aggregation of evidence from multiple independent sources. Let $m_1^{D_1}$ be defined on a domain $D_1 \subseteq V$, and $m_2^{D_2}$ be defined on domain $D_2 \subseteq V$. If domains $D_1 \equiv D_2 = D$, the combination can be performed directly using the Dempster Shafer rule, as follows:

$$\left(m_1^{D_1} \oplus m_2^{D_2}\right)(A) = \frac{\sum_{B \cap C = A} m_1^D(B) m_2^D(C)}{1 - \sum_{B \cap C = \emptyset} m_1^D(B) m_2^D(C)}$$

The $A, B, C \in \theta_D$ denote subsets of the frame that are defined by the Cartesian product of the variables in D. However, if domains D_1 and D_2 are different, then prior to applying the combination rule, the mass functions need to be extended to the joint domain $D_1 \cup D_2$. This operation is called vacuous extension, denoted by $\uparrow$ and defined as:

$$m_1^{D_1 \uparrow (D_1 \cup D_2)}(C) = \begin{cases} m_1^{D_1}(A) & if\ C = A \times \Theta_{D_2}, A \subseteq \Theta_{\mathrm{D}_1} \\ 0 & otherwise \end{cases}$$

Marginalisation is a projection of a mass function that is defined on domain D onto a mass function defined on a coarser domain $D' \subseteq D$. It is the inverse operation of extension (but extension is not the inverse of marginalization). Marginalisation is formalized as follows:

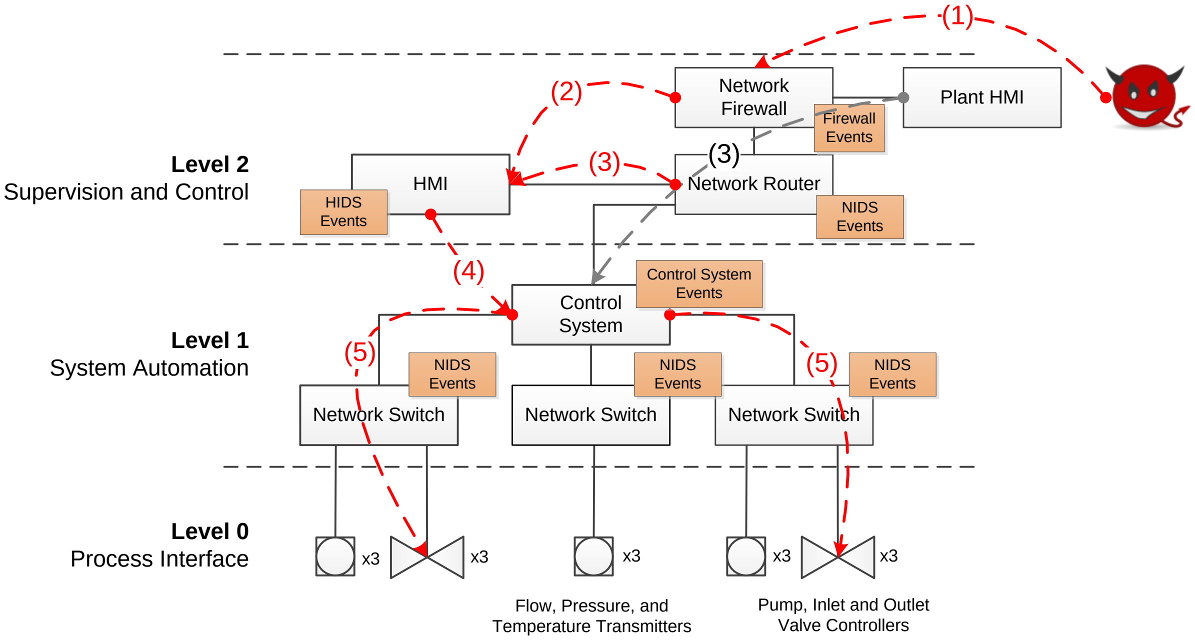

Fig. 2 Overview of the I&C system for reactor cooling, showing the steps taken by an attacker to issue malformed commands

$$m^{D\downarrow D'}(A) = \sum_{B\downarrow A} m^D(B)$$

Configurations in B reduce to the configurations in $A \subseteq \Theta_{D'}$ by the elimination of variables in $D \setminus D'$.

Evidential networks describe causality by the relationships between the system variables, through implication rules, e.g., *if-then* rules. Dempster Shafer (DS) theory[11] allows relation implication rules with uncertainty measures to be assigned. Suppose there are two disjoint domains $D_1$and D_2, with frames Θ_{D_1} and Θ_{D_2}. Then the implication rule can be formalized, as follows:

$$A \subseteq \Theta_{D_1} \Rightarrow B \subseteq \Theta_{D_2}$$

We can associate a certain degree of confidence in this rule, for example:

$$\rho \in [\alpha, \beta], 0 \leq \alpha \leq \beta \leq 1$$

The probability measure that is used for decision making on the domain of interest D within evidential networks is defined by pignistic probability distribution. The pignistic transform of a mass function m^D is defined for every element of the frame $\theta \epsilon \Theta_D$

$$etP(\theta) = \sum_{\theta \in A \subseteq \Theta_D} \frac{m_D(A)}{|A|},$$

where$|A|$ denotes the cardinality of set A (the number of elements in A).

Evidential networks infer knowledge about higher system states based on the evidence that is available from *Detectors* (*variables*) and their relationships (*mass functions*). Having a domain of interest $D^0 \subseteq V$, of variables that system states (used in state inference task). The information is derived by computing $(\oplus M)^{\downarrow D}$, where $\oplus M$ is the combination of all mass functions in the network.

4.2 Computer Security Threat Scenario

In this section, a computer security threat scenario is presented that will be used to demonstrate how ENs can be applied to reason about system states. An overview of the scenario is presented in Fig. 2. The scenario relates to the compromise of an I&C system that is supporting reactor cooling – the system that is responsible for removing heat from the reactor to the steam generator. For the purposes of this example, some details of the I&C system have been omitted.

Table 1 Attacker steps for the threat scenario

Step	Description
1	From their foothold in the infrastructure, which was initially obtained non-cyber means, the attacker exploits a remotely exploitable vulnerability on the *Network Firewall* that is located on the perimeter of the I&C system that supports reactor cooling. There are several ways that network firewall vulnerabilities can be exploited[12]. We assume the attacker can execute malicious programs on the firewall that support the next steps in the attack.
2	With control of the *Network Firewall*, the attacker scans the local network to identify other devices that are present. This can be realized using tools, such as *nmap* (https://nmap.org/) or *prads* (https://gamelinux.github.io/prads/). The result of this activity is that two devices are discovered: the *Local HMI* and *Control System*. Unable to directly compromise the *Control System* – there are no software vulnerabilities and remote services are protected with strong passwords – they successfully guess the password for a management interface on the *Local HMI*. This process can be supported by password cracking tools, such as Brutus (http://sectools.org/tool/brutus/).
3	From the *Local HMI*, the attacker then conducts a Man-In-The-Middle Attack (MITM) between the *Plant HMI* and the *Control System*. To achieve this, they perform an ARP spoofing attack[13], which results in all the traffic from the *Plant HMI* that is destined for the *Control System* being sent to the *Local HMI*. Tools, such as Ettercap (https://ettercap.github.io/ettercap/), can be used for the MITM attack.
4	After an operator has remotely connected to the *Control System* from the *Plant HMI* using their credentials, the attacker learns of the username and password for the *Control System*. (It is assumed the communication within the facility is not encrypted, which is typical.) They use these credentials to log onto the *Control System*.
5	Now that they can configure the *Control System*, the attacker could attempt to change the configuration of the pump, and the inlet and outlet valve controllers, e.g., by placing them in a potentially unsafe state. Furthermore, they could attempt to change the configuration of the control algorithm that is implemented on the *Control System*, e.g., by changing setpoints.

6.2.1 I&C System Description

The (sub-)systems that are part of the scenario can be summarized, as follows:

Plant HMI: A Human Machine Interface (HMI) that can be used to monitor and control the behaviour of the reactor cooling system. This is primarily achieved by changing the parameters that the *Control System* (see below) uses. The HMI is in the facility control room.

Network Firewall: The *Network Firewall* manages the communication between other systems (at different Security Levels) in the facility and those in this security zone. For example, according to IAEA NSS 17, a reactor cooling system would be assigned to Security Level 2, which implies restrictions on network communication:

> *"Only an outward, one way networked flow of data is allowed from level 2 to level 3 systems. Only necessary acknowledgment messages or controlled signal messages can be accepted in the opposite (inward) direction (e.g. for TCP/IP)."*

In this scenario, it is assumed that the *Plant HMI* is also assigned to Security Level 2, enabling remote access to the *Control System* from the control room. The firewall is configured to implement the control that is required under IAEA NSS 17 and to manage communication between systems at the same Security Level.

Network Switch and Router: Network devices that enable OSI Layer 2 and Layer 3 connectivity, respectively. Devices on the same broadcast domain use the Address Resolution Protocol (ARP) to map L3 identifiers (i.e., IP addresses) to L2 identifiers (i.e., MAC addresses). ARP is a request-response protocol, wherein an ARP query, to identify an L3-L2 mapping, is broadcast on the local network. The device that "owns" the L3-L2 mapping replies to the query. The results of this exchange are typically cached on the corresponding devices.

Local HMI: The *Local HMI* is used by operators that are on-site to observe the state of the reactor cooling system. Data are collected from the *Control System* and presented on this HMI. In this example scenario, an operator *cannot* change the behaviour of the *Control System* from this device.

Control System: The *Control System* takes input from transmitters, which measure flow, pressure and temperature from a cooling loop, and issues commands to actuators that control pump speeds, and the position of inlet and outlet valves. The purpose of the *Control System* is to manage the behaviour of the reactor cooling system.

6.2.2 Threat Scenario

In this scenario, an attacker intends to manipulate the reactor cooling system to cause a reactor trip. To achieve this, they can use several strategies to obtain a cyber presence in the nuclear facility. These include the use of social engineering or exploiting a third-party contractor's equipment. The intention is that the attacker can remotely access computer systems in the facility. Having gained a foothold, they can discover systems and traverse the infrastructure, by exploiting software vulnerabilities, to reach the target I&C system that manages the reactor cooling function. Having identified the system, the steps an attacker could take to exploit it are presented in Fig. 2 and Table 1.

In short, the attacker exploits a vulnerability on the *Network Firewall*. With a presence on this device, the attacker probes the network to discover other systems. The *Local HMI* is then exploited using a brute force password cracking attack. At this point, from the *Local HMI*, they perform a Man-In-The-Middle (MITM) attack between the *Plant HMI* and the *Control System* to obtain the credentials that allow them to manipulate the behaviour of the *Control System*. The attacker is then able to change the configuration of the *Control System*, to send commands to controllers that cause operational problems with the reactor cooling system. These could lead to a reactor trip.

6.3 An Evidential Network to Support Reasoning on I&C Incidents

The evidential network that can be used to reason about the state of the *Control System* is shown in Fig. 3. The variables V and frames Θ_V are given in Table 2. Meanwhile, the mass functions M_V, which are associated with the variables V, are given in Table 3.

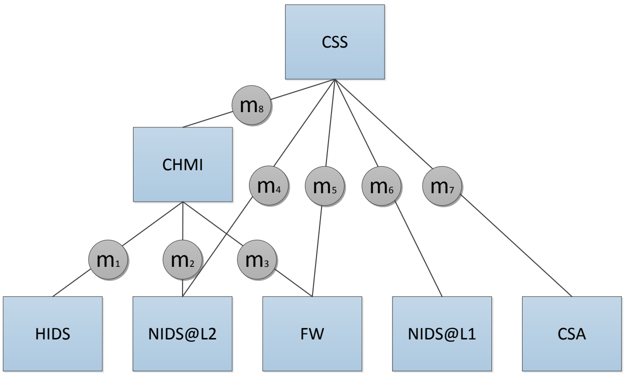

Fig. 3 Overview of the evidential network. Rectangles depict variables (Table 2), while circles depict mass functions (Table 3)

Table 2 Variables and Frames of the EN

Variables	Frame	Description
Control System State (CSS)	{0,1,2}	0-normal, 1-erroneous 2-compromised
HIDS alarms (HIDS)		
Password Brute Force (BF)	{0,1}	0-none 1-detected
NIDS alarms (NIDS@L1/L2)		
Malformed Packets (MP)	{0,1}	0-none 1-detected
Network Scanning (NS)	{0,1}	0-none 1-detected
Network Connection (NC)	{0,1}	0-none 1-detected
Firewall alarms (FW)		
Packet Drops (PD)	{0,1}	0-none 1-detected
Control system alarms (CSA)		
System Errors (*SA*)	{0,1}	0-none 1-detected
Intermediate Variable		
Compromised HMI *(CHMI)*	{0,1}	0-false 1-true

The variables that are shown in Table 2 are grouped, based on the type of detector they belong to. To reason about the *Control System* state, the following detectors are used:

1. *Cyber Security Detectors*: A Host-based Intrusion Detection System (HIDS), deployed on the *Plant HMI* for detecting malicious activity, such as the brute force password guessing attacks (BF); a Network Intrusion Detection System (NIDS), deployed at the System Level 1 and 2 network switches (MP, NS, and NC); and packet drop notifications from the *Network Firewall* (PD).
2. *Control System Alarms*: Detectors that are used for displaying errors in an automation system, e.g., station failures or I/O module errors (SA).

The evidential network has one intermediate variable *Compromised HMI (CHMI*), which is used for reasoning about the top level variable *CSS*. The *Control System* variable (CSS) is our domain of interest for the problem $D^0 = \{CSS\}$. This variable is used for reasoning about the state of the C*ontrol System*, which is assumed to be either in a normal (0), erroneous (1), or compromised (2) state. The variables are combined using the mass functions that are specified in Table 3, to reason about the state of the *Control System*. The confidence in a rule can be specified using a linguistic scale for probability values[14]. In this example, we have used a four-element scale with the following mapping to the probabilities: probable (99%), likely (67%), possible (33%) and unlikely (1%).

Using the mass functions, a domain expert can specify knowledge about incidents that result from cyber threats, as well as those resulting from faults that might occur in the control system under observation. Most of the detectors in our reactor cooling system example provide information which are relevant evidence for the detection of cyber-threats, such as the attack that is described in Table 1. For example, Step 1 involves cyber-attack techniques that could be detected by the *Packet Drop* (*PD)* alarms. However, the *PD* alarm could be raised not only by an attack, but also incorrectly configured devices on the plant bus network, which will cause this alarm to be trigger (i.e., rule m_5: $(PD = 1) \Rightarrow (CSS = 0)$).

Evidential networks allow the specification of uncertainties in similar situations in which events that are generated by *Detectors* might be fault positives. In this case, they are used to reason about CSS states, and become more relevant when they are fused with evidence from other *Detectors*. For example, the likelihood that *CSS* = 2 (i.e., the *Control System* is compromised) is negligible if only *PD* = 1, but becomes significant once it is combined with evidence from other *Detectors*, e.g., *NC* = 1. Thus, the evidential network should be constructed such that it correctly captures uncertainty in detector alarms, due to the noisy environment or the detector's inherent (in)accuracy, and encapsulates the right set of evidence that are necessary for accurate reasoning about the control system state (CSS).

Table 3 The EN mass functions and corresponding rules

Mass Functions	Rules
m_1	$(BF = 1) \Rightarrow (CHMI = 1)$ **with conf. between probable and 1;** $(BF = 0) \Rightarrow (CHMI = 1)$ **with conf. between unlikely and possible;**
m_2	$(NC = 1, NS = 1) \Rightarrow (CHMI = 1)$ **with conf. between probable and 1;** $(NC = 0, NS = 1) \Rightarrow (CHMI = 1)$ **with conf. between likely and probable;** $(NC = 0, NS = 0) \Rightarrow (CHMI = 0)$ **with conf. between likely and probable;**
m_3	$(PD = 1) \Rightarrow (CHMI = 1)$ **with conf. between unlikely and likely;** $(PD = 0) \Rightarrow (CHMI = 0)$ **with conf. between unlikely and possible;**
m_4	$(MP = 1) \Rightarrow (CSS = 2)$ **with conf. between possible and likely;** $(NC = 1) \Rightarrow (CSS = 2)$ **with conf. between possible and probable;** $(NS = 1) \Rightarrow (CSS = 2)$ **with conf. between possible and probable;** $(MP = 1, NC = 1, NS = 1) \Rightarrow (CSS = 2)$ **with conf. between probable and 1;**
m_5	$(PD = 0) \Rightarrow (CSS = 0)$ **with conf. between possible and likely;** $(PD = 1) \Rightarrow (CSS = 2)$ **with conf. between likely and probable;**
m_6	$(NC = 0) \Rightarrow (CSS = 0)$ **with conf. between possible and likely;** $(NC = 1) \Rightarrow (CSS = 2)$ **with conf. between probable and 1;**
m_7	$(SA = 0) \Rightarrow (CSS = 0)$ **with conf. between possible and probable;** $(SA = 1) \Rightarrow (CSS = 1)$ **with conf. between possible and likely;** $(SA = 1) \Rightarrow (CSS = 2)$ **with conf. between possible and likely;**
m_8	$(CHMI = 0) \Rightarrow (CSS = 0)$ **with conf. between likely and probable;** $(CHMI = 0) \Rightarrow (CSS = 1)$ **with conf. between likely and probable;** $(CHMI = 1) \Rightarrow (CSS = 2)$ **with conf. between probable and 1;**

8 Conclusion

Operators of critical infrastructures, including nuclear facilities, are facing an increasingly sophisticated computer security threat[15]. These new threats use sophisticated attack methods and are targeted. Moreover, there are more cyber-attacks that result in operational consequences, such as blackouts in the energy sector.

In this context, operators of nuclear facilities require an effective computer security incident response capability. At the core of incident response is the ability to *detect*, *analyse* and *contain* threats. To detect cyber-attacks, monitoring and detection systems need to be deployed, such as intrusion detection systems. Based on the events that are generated by these systems, an operator must analyse and reason about the presence and nature of an incident, and its (potential) effect on a target environment. This is a challenging task. The events generated by detection systems can be numerous and are known to be unreliable (i.e., they generate false positives and negatives).

In this paper, we have motivated and outlined a high-level architecture that can be used to support incident response, given untrustworthy detection sources. Central to this architecture is a reasoning engine that generates hypotheses about the state of systems and a belief in their correctness. The measure of belief accounts for the uncertainty associated with detection system performance (caused by false positives and negatives). We propose the use of evidential networks to realise this reasoning engine, and present an example threat scenario.

Future work will focus on developing the example scenario and evidential network that is presented in the paper. The aim is to ascertain its capacity to determine system states for different scenarios, e.g., in the presence of threats and faults, for example. To achieve this, we will develop a representative I&C test system that manages primary reactor cooling for a Pressurized Water Reactor (PWR).

Acknowledgement

The research leading to these results has received funding from the IAEA as part of the CRP J02008 on Enhancing Computer Security Incident Analysis at Nuclear Facilities.

References

[1] R.M. Lee, M.J. Assante and T. Conway, Analysis of the Cyber Attack on the Ukrainian Power Grid, SANS ICS and E-ISAC Whitepaper, March 2016.

[2] P. P. Shenoy, A valuation-based language for expert systems, International Journal of Approximate Reasoning, Vol. 3, pp. 383–411, 1989.

[3] M. Auty, Anatomy of an advanced persistent threat, Network Security, Vol. 2015, No. 4, 2015, pp. 13-16, ISSN 1353-4858.

[4] R. Langner, Stuxnet: Dissecting a Cyberwarfare Weapon, IEEE Security & Privacy, Vol. 9, No. 3, pp. 49-51, May-June 2011.

[5] M.J. Assante, R.M. Lee, The Industrial Control System Cyber Kill Chain, SANS Institute Whitepaper, October 2015.

[6] International Atomic Energy Agency (IAEA), Computer Security at Nuclear Facilities, IAEA Nuclear Security Series No. 17, December 2011.

[7] P. Cichonski, T. Millar, T. Grance, K. Scarfone, Computer Security Incident Handling Guide, NIST Special Publication 800-61 R2, August 2012, http://dx.doi.org/10.6028/NIST.SP.800-61r2

[8] F. Sabahi and A. Movaghar, Intrusion Detection: A Survey, 2008 Third International Conference on Systems and Networks Communications, Sliema, 2008, pp. 23-26.

[9] M. Iturbe, J. Camacho, I. Garitano, U. Zurutuza and R. Uribeetxeberria, On the Feasibility of Distinguishing Between Process Disturbances and Intrusions in Process Control Systems Using Multivariate Statistical Process Control, 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W), Toulouse, 2016, pp. 155-160.

[10] E. Triantaphyllou, Multi-criteria decision making methods: a comparative study, Vol. 44. Springer Science & Business Media, 2013.

[11] G. Shafer, A Mathematical Theory of Evidence, Princeton, NJ, USA: Princeton University Press, 1976.

[12] S. Kamara, S. Fahmy, E. Schultz, F. Kerschbaum, M. Frantzen, Analysis of vulnerabilities in Internet firewalls, Computers & Security, Vol. 22, No. 3, 2003, pp. 214-232.

[13] C. L. Abad and R. I. Bonilla, An Analysis on the Schemes for Detecting and Preventing ARP Cache Poisoning Attacks, 27th International Conference on Distributed Computing Systems Workshops (ICDCSW '07), Toronto, Ont., 2007, pp. 60-68.

[14] I. Friedberg, X. Hong, K. McLaughlin, P. Smith, P. Miller, Evidential Network Modeling for Cyber-Physical System State Inference, IEEE Access, Vol. 5, pp. 17149-17164, 2017.

[15] US CERT, Alert (TA17-293A) Advanced Persistent Threat Activity Targeting Energy and Other Critical Infrastructure Sectors, https://www.us-cert.gov/ncas/alerts/TA17-293A October 2017.