A Survey on Open Forensics in Embedded Systems of Systems

From Automotive Considerations to a Larger Scope

Robert Altschaffel, Kevin Lamshöft, Stefan Kiltz, Mario Hildebrandt, Jana Dittmann
Advanced Multimedia and Security Lab
Otto-von-Guericke-University
Magdeburg, Germany
email:Robert.Altschaffel|Kiltz|Mario.Hildebrandt|Jana.Dittmann@iti.cs.uni-magdeburg.de; Kevin.Lamshoeft@st.ovgu.de

Abstract—**Embedded systems form the foundation for most electronic systems in our everyday environment. Complex systems of embedded systems, usually connected via communication buses, are fundamental for a broad range of applications like industrial control systems (ICS) or transportation. While the complexity of these systems of systems is ever increasing, our understanding of these systems is decreasing. This not only leads to problems in designing safe and secure systems - but also in reconstructing the chain of events that led to an unwanted outcome regarding the system of systems in question. While previous work gave a survey on the means to perform event reconstruction (hence, forensic investigations) into automotive environment, this work extends the point of view to include other systems of embedded systems connected using communication bus technologies. Hence, the main contribution is a survey on the field of forensics in generalized control systems. This includes the identification of implications for forensics in ICS, the discussion of approaches and tools already in existence and those still needed to perform a forensic investigation in ICS environments.**

Keywords- automotive; computer forensics; embedded systems; forensic processes; industrial control systems; safety & security.

I. INTRODUCTION

Most complex electronics-based systems rely on embedded systems connected to actuators and sensors to perform their given task. They often implement a number of (open/closed) control loops using sensors as input and actuators as means to influence their physical environment - earning them the alternative designation as cyber-physical (cp) systems. Prime examples for such control systems are automotive systems [1] and industrial control systems. In these domains, embedded systems control even fundamental, safety-critical functions. Hence, problems with the security of such control systems can affect its safety - e.g., a vehicle might crash or an industrial robot might hurt a worker if something goes awry.

While the paramount objective is to prevent such incidents from happening altogether, this objective seems to be out of reach forever. Hence, in the case of such an incident, (computer) forensic investigations are necessary in order to identify the course of events that led to a given outcome. The results of this investigation can help to fix vulnerabilities or act as a foundation for an assessment of legal responsibility. Especially in the latter case, it is necessary that such an event reconstruction follows generally accepted, scientific and well-proven principles. These principles are referred to as a forensic process. A forensic process requires traces used for event reconstruction to be gathered and analyzed in an authentic fashion (originating from the subject of the investigation) and with integrity (unaltered by external influences or during the course of the investigation) assured. Further, the whole process is required to be comprehensively documented. The process also needs to ensure protection of personal data in accordance to applicable regulations and laws. Further, the process needs to respect regulations concerning the collection of data, especially in consideration of privacy laws and human rights.

In the beginning of an investigation it is very often unclear if an incident arises from an error or an attack or if an investigation will be escalated to include legal authorities. An investigation thus should follow the same principles without regards to the starting hypothesis of the investigator.

Previous work [1] discussed this topic with a scope on automotive systems while this article contributes an extended scope with a generalized view on control systems.

In this broader scope, the same principal challenges as within the automotive domain are apparent: there is a lack of forensic processes focused on (industrial) control systems (so called ICS) that are openly discussed and peer-reviewed in the scientific community. This holds particularly true for the basic process control, where often non-standard IT-equipment is used [2], and affects the identification, acquisition, investigation and analysis of potential case-relevant data. While isolated solutions are applied, these are often hidden behind heavy regulations regarding intellectual property. Hence, this article aims at discussing the possibility of applying well-researched approaches from the domain of desktop-IT forensics to control systems.

In order to achieve this, this article is structured as follows: Section II gives an overview on the technical background of control systems and forensics. Section III discusses the forensic process in the context of control systems. Currently available tools, which might support the forensic process in ICS and their suitability for forensic use, are discussed in Section IV. Section V discusses the requirements for tools geared towards supporting forensics in ICS, while Section VI concludes this article.

II. Technical Background

This section gives a brief overview on the topic of forensic in classical desktop IT and a basic understanding of (industrial) control systems in order to bring these topics together in the following sections. In order to compare forensics in the ICS domain to the automotive domain, a brief recap of automotive IT is given as well.

A. Forensics in Desktop IT

The forensic process aims at finding traces that support the reconstruction of an event. In order to increase the validity of the reconstruction, these traces have to be gathered in a way to preserve authenticity (trace origin) and integrity (trace is unaltered). Additional challenges arise from the need to protect personal data in accordance to applicable regulations and laws as well as respecting regulations concerning the collection of data, especially in consideration of privacy laws and human rights.

To ensure this, a range of models for the forensic process exist, both for classical crime scenes [3], as well as for digital forensics in Desktop IT [4]. These models are often practitioner driven and usually break down the forensic process into distinct phases. For this article, we use the forensic process from [5], as it contains both the practitioner's and the computer scientist's view (see [6]), the latter often being omitted in an attempt to provide guidelines for practitioners only. This model includes investigation steps (practitioner's view), data types (computer scientist's view) and methods for data access (computer scientist's view). Thus, by adhering to this model, both the research aspect as well as the implementation of forensic procedures in practice is supported.

For this first survey on automotive IT forensics we rely on the investigation steps:

- **Strategic preparation** (*SP*) represents measures taken by the operator of an IT-system, prior to an incident, which support a forensic investigation.
- **Operational preparation** (*OP*) represents the preparation for a forensic investigation after a suspected incident.
- **Data gathering** (*DG*) represents measures to acquire and secure digital evidence.
- **Data investigation** (*DI*) represents measures to evaluate and extract data for further investigation.
- **Data analysis** (*DA*) represents the detailed analysis and correlation between digital evidence from various sources.
- **Documentation** (*DO*) represents the detailed documentation of the investigation.

The forensic process is furthermore also divided into live forensic and post-mortem forensics. Live forensics covers the part of the forensic examination performed while the system under investigation is still active. Post-mortem forensics covers all the part of the forensic examination while the system under investigation is powered-off. Live forensics offers the possibility to find traces in highly volatile areas such as main memory but often comes with the implication of substantially altering the state of the system under investigation - either by letting it perform its current operations or by querying the system for certain information from the main memory, which actively alters the state of the system. Post-mortem forensics allows access to lesser volatile mass storage and analyze it in ways ensuring integrity of the mass storage device (typically by using write-blocking devices) but cannot gain insight into the main memory contents. The consideration when to power off a system under investigation and switch from live forensics to post-mortem is to be decided on a case-by-case basis and represents a crucial decision in every forensic examination.

B. Automotive IT

As discussed in previous work [1], modern cars consist of components with fixed logic (or none at all) and components with (re-)programmable logic. The latter often include embedded systems and thus are more important for this article, although being only useful in conjunction with electronic devices with fixed or no built-in logic. Of particular relevance for our discussion are:

- **Sensors** measure the conditions of the vehicle's systems and environment (e.g., pressure, speed, light levels, rain intensity etc.) as well as user input.
- **Actuators** are electrically operated and manipulate their environment in non-electric aspects (e.g., mechanics, temperature, pressure, etc.).
- **Electronic Control Units** (ECUs) electronically process input signals acquired via sensors and relay commands to actuators. Some units control critical systems, such as the engine or safety-critical systems like ESC (Electronic Stability Control) or SRS (Supplemental Restraint System), while others control comfort functionality (e.g., door control units). ECUs are custom-tailored compact, embedded systems. Due to high cost constraints in the automotive industry, they operate on a minimum set of resources regarding CPU computing power, mass storage and main memory. Common exceptions are ECUs that handle multimedia functionality. The number of ECUs embedded with a vehicle is still rising - while a luxury car in 1985 contained less than 10 ECUs, the numbers increased to more than 100 in 2010 [7].
- **Direct analogue cable connections** connect sensors and actuators directly to a specific ECU.
- **Shared Digital Bus Systems** are used for communication among ECUs [8]. In modern cars, several different technologies for digital automotive field bus systems are used with different capabilities, requirements and cost factors. The most common automotive field bus system, often forming the core network of vehicle systems communication, is the Controller Area Network (CAN) [9]. This CAN network is often divided into sub-networks such as powertrain/engine, diagnostics, comfort or infotainment. ECUs are connected to the sub-network and these sub-networks interconnect using a CAN Gateway ECU, which handles the routing of messages to different sub-networks. The CAN

message consists of several flags, the CAN ID and the payload. The CAN ID represents the type of a message and implies a certain sender and receiver for the message. It is assumed that a message with the corresponding ID is sent by the ECU normally responsible for this message. In addition, the CAN ID serves as priority. A lower CAN ID corresponds to a higher priority.

The above implement essential instrumentation and control circuits for the functionality of today's vehicles.

C. Industrial Control Systems (ICS)

On a fundamental level, ICS consist of the same basic building blocks found in automotive IT. Again, sensors collect information about the environment while actuators manipulate the environment. Just like ECUs in automotive systems, so called Programmable Logic Controller (PLCs) process input taken from sensors and relay commands to actuators. The components are usually arranged in more or less complex hierarchies, as shown in Fig. 1. Since these actuators could be used to manipulate hazardous substances or objects (e.g., hot steam, poisonous gas, heavy loads), some of them might be regarded as safety-critical. Some ICS might also control critical infrastructures [10] such as emergency services or traffic control. Communication with users and/or supervisors is performed using dedicated human-machine-interfaces (HMI) or supervisory control and data acquisition (SCADA) control systems. PLCs need to communicate with each other using various communication buses, i.e., field buses such as PROFIBUS [11] or industrial Ethernet such as PROFINET [12] or Modbus TCP [13].

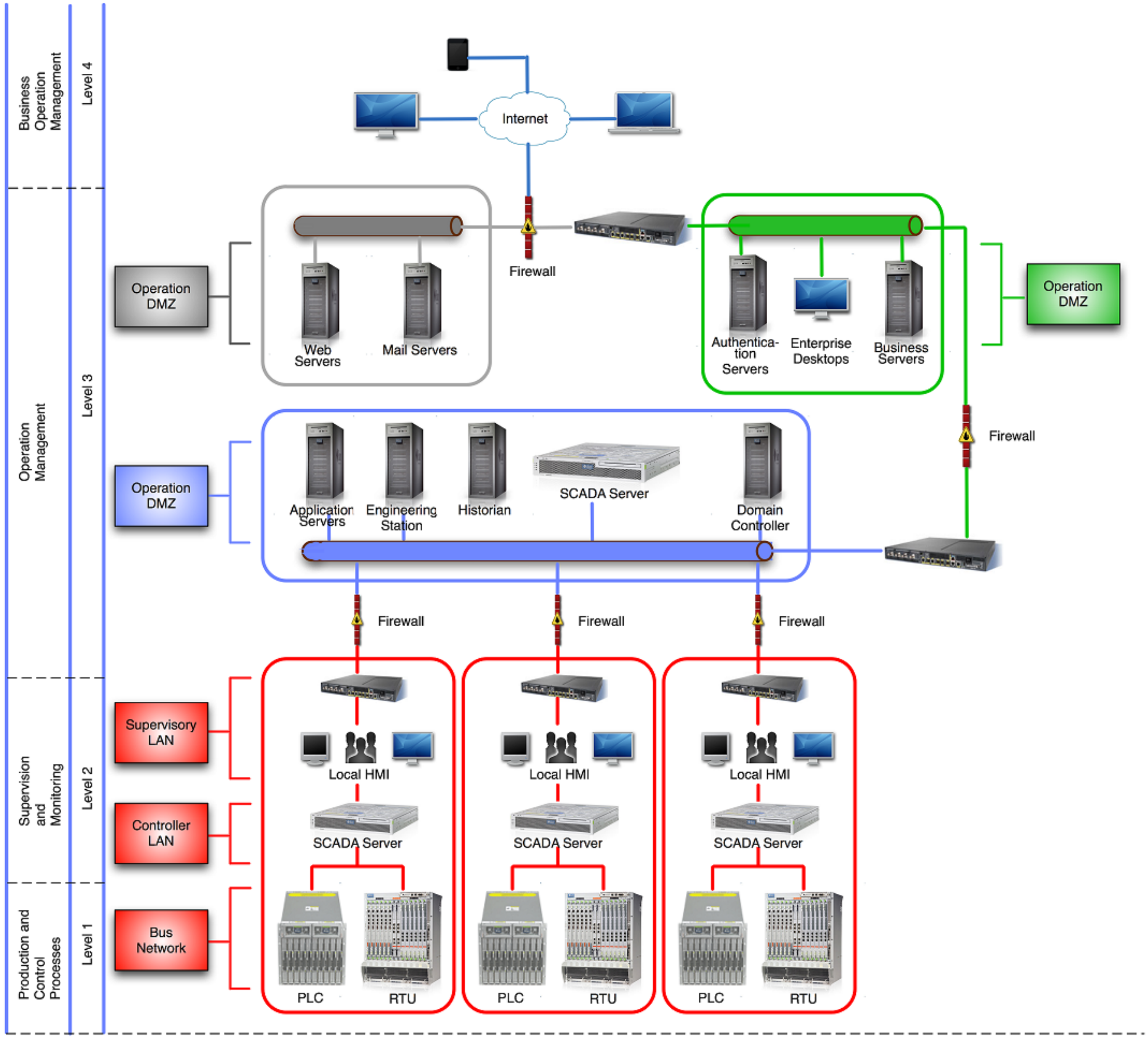

Figure 1. Exemplary Industrial Control Systems, based on [15]

Fig. 1 shows a typical hierarchy for ICS. This figure also shows, that ICS often form a part of larger company networks. Hence, ICS are often connected to classical IT.

A model for these hierarchical networks is given by the Purdue Enterprise Reference Architecture [14]. Today this architecture is used in various iterations, for example, the ISA99 variation.

According to [15], this model defines three different zones within an enterprise network, as pictured in Fig. 2. The zones serve as an indicator for communication flows - communication is only possible within a given zone or to the neighboring zone. That means that there is no direct communication between Enterprise Zone and Cell/Area Zone.

The specific levels are defined in [16]:

- *Level 0 — The physical process — Defines the actual physical processes.*
- *Level 1 — Intelligent devices — Sensing and manipulating the physical processes. Process [sic] *sensors, analyzers, actuators and related instrumentation.*
- *Level 2 — Control systems — Supervising, monitoring and controlling the physical processes. Real-time controls and software; DCS, human-machine interface (HMI); supervisory and data acquisition (SCADA) software.*

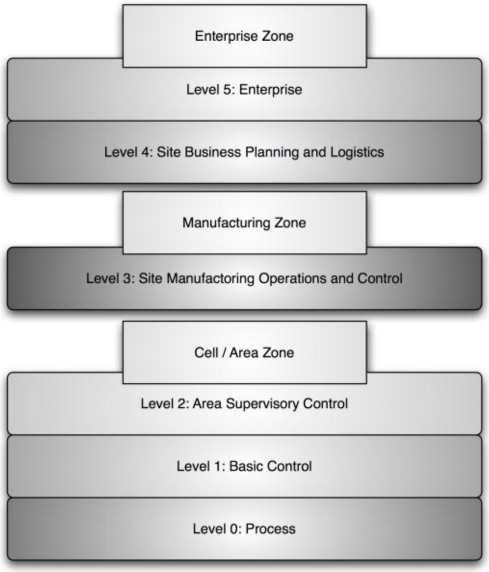
Enterprise Zone
Level 5: Enterprise
Level 4: Site Business Planning and Logistics
Manufacturing Zone
Level 3: Site Manufactoring Operations and Control
Cell / Area Zone
Level 2: Area Supervisory Control
Level 1: Basic Control
Level 0: Process

Figure 2. Purdue Model in the variant of ISA99, according to [15]

- *Level 3 — Manufacturing operations systems — Managing production workflow to produce the desired products. Batch management; manufacturing execution/operations management systems (MES/MOMS); laboratory, maintenance and plant performance management systems; data historians and related middleware. Time frame: shifts, hours, minutes, seconds.*
- *Level 4 — Business logistics systems — Managing the business-related activities of the manufacturing operation. ERP is the primary system; establishes the basic plant production schedule, material use, shipping and inventory levels. Time frame: months, weeks, days, shifts.*

According to this, activities on level 4 and above are not the scope of this work since they basically represent the classic IT domain - Industrial Control Systems correspond to the components on levels 0 to 3 of the hierarchy defined in [16] for the scope of this article. Hence, ICS contain PLCs, sensors, actuators, HMIs, and SCADA.

III. Review of the forensic Process in Control Systems

This section discusses the implications originating from the involvement of ICS in a forensic process. After Section II covered the fundamental structure of ICS, this section will start with an overview of the data present in this structure. Following this, an overview on the nature of concurrent attacks targeting cyber-physical systems is given in order to identify components and data affected by these attacks. After that, implications for the forensic process will be formulated, based on these observations and the specifics of ICS.

A. Data Streams in Cyber-physical Systems

Components in ICS share the same possible locations of data as classical IT systems or components in automotive IT. These possible locations of data are referred to as data streams [1]. In general, three data streams can be identified in components:

Communication describes the data transmitted over networks. In an ICS environment this network might consist of field bus systems or dedicated serial lines.

Volatile memory represents the non-persistent part of a component's memory. In the context of ICS, this would include the main memory of PLCs.

Non-Volatile memory is the persistent part of a component's memory. It consists of mass storage integrated in the PLC as well as additional memory cards.

B. Attacks on Cyber-physical Systems

In recent years, cyber-physical systems are subject to a growing number of attacks. In order to show the challenges when investigating cyber attacks, a brief review of these cyber attacks is necessary. While these attacks all focus on ICS, they carry wildly different implications for the forensic process. These implications are discussed in Section III.C.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

1) *Dragonfly (aka Energetic Bear)*

Dragonfly was an attack aimed against energy suppliers [17]. While BlackEnergy disrupted, Dragonfly gathered information by infecting the targeted systems with the Dragonfly Remote Access Trojan horse (RAT). This remote access enabled attackers to use the specialized Havex malware on the systems. According to [18]: "*Havex used an OPC malware scanning module to gather information about ICS devices and send that data back to Command and Control (C&C) servers used by the Dragonfly group.*

The malware used an industrial protocol scanner to find networked devices on TCP ports 44818, 102 and 502. Automation companies such as Siemens and Rockwell Automation use these ports for ICS system communication. The industrial processes using the protocols are found in consumer goods manufacturing and packaging applications."

This quote shows that while this attack was aimed towards ICS, it did not touch the ICS in question. This attack played out solely above level 3 on the hierarchy of ICS discussed in Section II.C.

2) *BlackEnergy*

BlackEnergy started out as a popular crimeware (malware designed to automate criminal activities) [19] but is mostly known for its use during the cyber attacks on Ukrainian power grids in late 2015. These attacks are a prime example of the cyber kill chain [20]. The attackers used an approach with various, distinguishable steps [21]. In a preparation stage, access to the network was obtained by using spear-fishing attacks deploying the BlackEnergy 3 malware. An extensive reconnaissance stage followed, allowing the tailoring and planning of further actions. When the final stage of the attack started, the attackers sent the signal to open the power breakers, overwrote the firmware of serial-to-ethernet-converters (thereby destroying the link between SCADA and the PLC), disabled the uninterruptable power supply and, finally, wiped the hard drives of the SCADA systems. The last three steps aimed at making recovery from this cyber attack more difficult. However, the attack did not aim at 'hacking' the ICS in question - the attackers essentially took over the SCADA system, sending perfectly legitimate commands to the ICS in question. Hence, this attack can be described as a case of 'SCADA Hijacking' [22].

3) *Stuxnet*

Stuxnet is a major example for an advanced persistent threat (APT). This highly sophisticated attack followed a multi-step approach. This approach consisted of infecting systems in the target network, then infecting the programming environment for the ICS and finally, the ICS in question [23]. In essence, the final stage of the Stuxnet attack was the injection of malicious logic into the ICS code while using rootkit techniques to hide this logic from the programmer. The modified ICS code (containing the malicious logic, still hidden from the programmers' view) was then loaded on the ICS and executed - leading to the breakdown of actuators (permanent physical destruction).

4) *PLC- Blaster*

PLC-Blaster [24] is a malware executed directly on a PLC. This malware is a worm, which resides within a PLC and during execution scans the attached network for possible targets. Once targets have been identified, the malware infects the PLC in question. The infection takes place using network connections of the PLC. PLC-Blaster initiates a transfer of software to the PLC in question and basically updates the PLC with the malicious code. The implementation demonstrated in [24] shows various possible malicious functions, which might be implemented in order to illustrate the possible impact of such a PLC-resident malware.

C. *The Nature of Forensic Investigations into ICS*

These four examples given under Section III.B all have in common that all attacks have an impact on the security of ICS. However, while ICS have always been the target, they played varying roles in the attacks itself.

1) *Dragonfly (aka Energetic Bear)*

In attack 1 (Dragonfly) the ICS within the system was not even touched - there were no unauthorized, malicious messages or anything outside of the normal operating procedures. From the point of view of the ICS, no attack happened at all. The forensic implication is that the forensic investigation in this case has to concentrate on the surrounding IT-infrastructure. Thus, the gathering, investigating and analyzing data is covered by conventional Desktop-/Server-IT. The first challenge (as part of strategic preparation) is to provide access to network, main memory and mass storage data streams to a potential broad range of Desktop/Server IT Systems. The next challenge, in operational preparation, is to decide, which data streams from which systems need to be considered.

2) *BlackEnergy*

Attack 2 (BlackEnergy) is different in that, at the final stage, a malicious command was transmitted to the ICS in question. While the command itself seemed authentic to the ICS, it altered volatile (main) memory and caused actions. The forensic implication is the need to consider both, a conventional Desktop-IT system and an ICS, in this forensic investigation. Here the complex interchange using networks communication towards the ICS is the primary challenge. The mass storage, main memory and network data streams have to be gathered, investigated and analyzed using conventional Desktop-IT forensics. Additionally, access at least to the network data stream of the ICS system is necessary for the subsequent forensic investigation of the ICS system. The later involves non-standard data gathering, investigation and analysis techniques.

3) *Stuxnet*

Attack 3 (Stuxnet) went further - here the ICS in question was directly infected with malicious code. This not only altered the volatile memory and caused action, but also altered non-volatile (mass) memory. Similarly to Attack 2, the forensic implication is, that the forensic investigation in this case has to consider both a conventional Desktop-IT system and an ICS.

Additional to the access to mass storage, main memory and network data streams of the Desktop-IT system and the network data stream towards the ICS system, also access to the main memory and mass storage streams of the ICS are needed.

4) PLC-Blaster

Attack 4 (PLC-Blaster) was even more extreme in that only the PLCs were altered at all. Any attached control systems where totally left out of the loop. The forensic implication of this attack is, that forensic capabilities to gather, investigate and analyze data in ICS are necessary to detect the attack at all. This includes access to all data streams (mass storage, main memory, network), involving non-standard investigation techniques.

In order to clarify the scope of this article, it is necessary, to have a closer look on the forensic examinations to be performed to investigate these attacks.

1) Dragonfly (aka Energetic Bear)

In the case of attack 1 (Dragonfly), no forensic investigation into the ICS targeted by the attack takes place. Although, obtaining information on ICS is the ultimate goal of the attacker, all systems attacked belong to the classic IT domain. A forensic investigation would therefore take place in this domain. In addition, it would be utterly impossible to find any traces of such attacks inside the ICS systems in question.

2) BlackEnergy

Attack 2 (BlackEnergy) is a borderline scenario - again, all systems attacked belong to the classic IT domain. However, in this case, a command is transmitted to the ICS, leaving potential traces in the memory of the ICS.

3) Stuxnet

In this complex attack, various systems are attacked, including the ICS itself. Hence, the ICS is infected by malicious code, leaving traces.

4) PLC-Blaster

In attack 4 (PLC-Blaster) the PLC in question is the only component attacked and possibly containing forensic evidence.

As shown in these diverse steps needed to investigate these different attacks, all these attacks led to wildly different locations of possible traces during a forensic investigation. Table I maps these possible traces to the hierarchy levels in automation and the different data streams (see Section II.C for further information).

1) Dragonfly (aka Energetic Bear)

In this case, no traces are left on level 1 and level 2. Communication on level 3 and level 4 will be altered and could lead to possible traces.

2) BlackEnergy

BlackEnergy will cause traces on level 2, since volatile memory, non-volatile memory and communication of level 2 systems is altered. Level 3 and level 4 might offer traces of the spear fishing campaign used to access level 2. However, the PLC in question is not altered and it is very unlikely that it will contain any useful forensic traces if the communication (including the valid, but malicious, requests from the level 2 systems) is not completely captured.

3) Stuxnet

While Stuxnet is a highly advanced malware, it leaves (well-hidden) traces at anything it touches. Stuxnet alters the software on level 1 and level 2. Hence, volatile memory and non-volatile memory of the PLC in question offers another source of forensic data.

4) PLC-Blaster

PLC-Blaster only alters the PLC in question and the communication between various PLCs. Since level 2 is sometimes used in order to enable various PLCs to communication with each other, a complete capture of the communication on level 2 might serve as a source of forensic traces. However, the main source will be the volatile and non-volatile memory of the PLC in question.

These examples show the diversity of ICS-centered attacks and that these are often accompanied by attacks on classical IT systems. This is based on the fact that attackers first need to gain access to a system connected to the targeted ICS. If the ICS in question were, for example, directly connected to the internet (which, at the time of writing, happens dangerously often [25]), this step would be necessary. Hence, most forensic investigations in ICS will have points of contact with forensic investigations into classical IT in order to identify used attack vectors.

TABLE I. LOCATION OF POSSIBLE TRACES FOR DIFFERENT TYPES OF ICS-CENTERED ATTACKS

Attack	**Possible Traces in ...**			
	Level 1	***Level 2***	***Level 3***	***Level 4***
Dragonfly	no	no	Communication	Communication
BlackEnergy	Communication	Communication Volatile Memory Non-Volatile Memory	Communication Volatile Memory Non-Volatile Memory	Communication Volatile Memory Non-Volatile Memory
Stuxnet	Communication Volatile Memory Non-Volatile Memory	Communication Volatile Memory Non-Volatile Memory	Communication Volatile Memory Non-Volatile Memory	Communication Volatile Memory Non-Volatile Memory
PLC-Blaster	Communication Volatile Memory Non-Volatile Memory	Communication	no	No

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

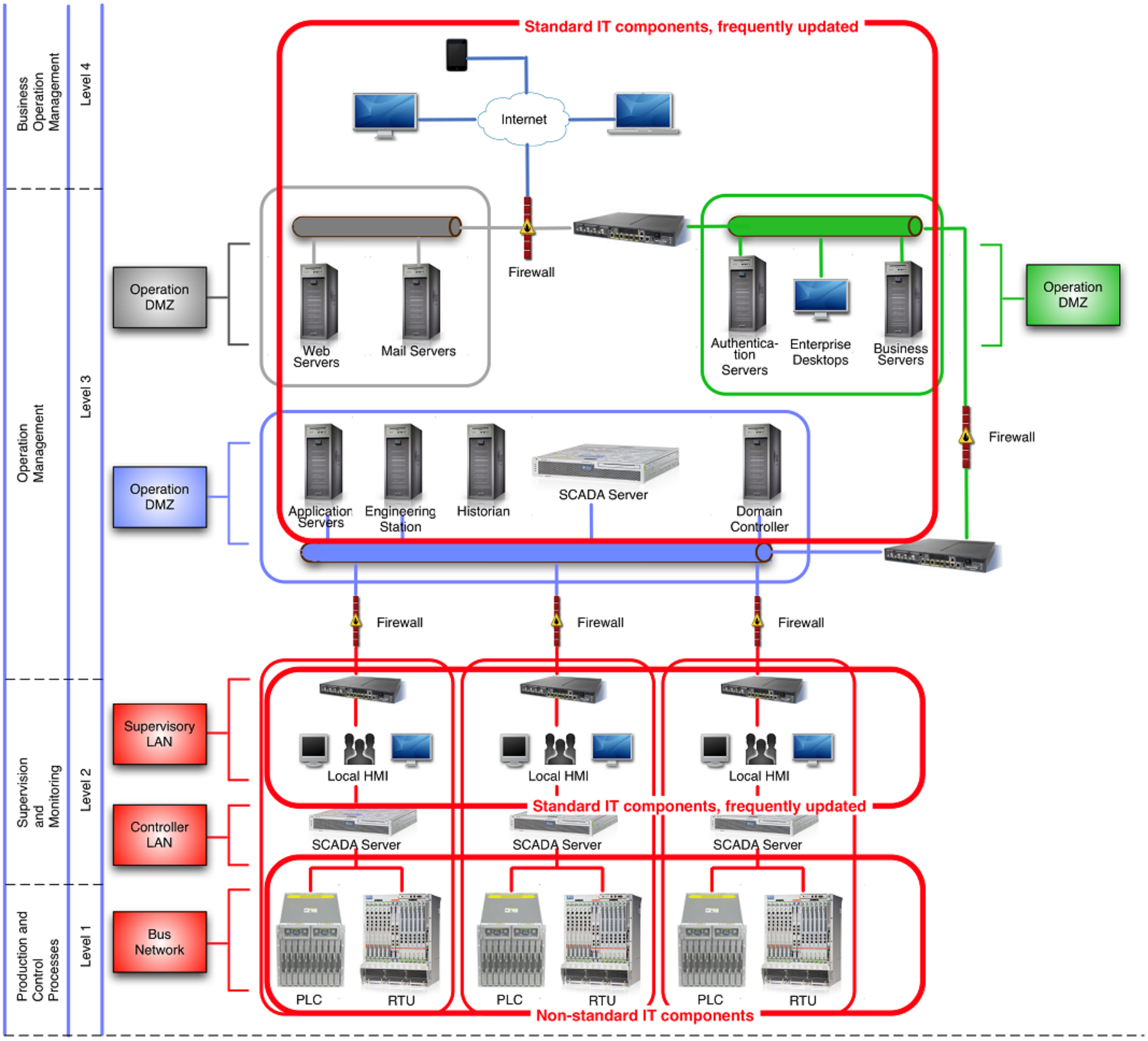

Figure 3. Relationship between ICS forensics and the traditional domain of digital forensics, according to [25]

This relation is shown in Fig. 3, taken from [26]. ICS are usually connected to classical IT environments. There is a clear overlap if ICS employs standard IT components for isolated solutions. Hence, ICS Forensics is a new forensics domain with strong ties and relations to the traditional (IT-) forensic domain.

D. *The Nature of Forensics in Automotive Systems*

The points discussed in the previous two sections hold also true for other instances of cyber-physical systems. This includes automotive systems, which have been the primary focus of previous work [1]. In the automotive domain, there are examples of direct attacks and manipulations on specific automotive components (e.g., odometer manipulation [27]), attacks on the classical IT systems forming the back-end in order to forge apparently authentic commands to the automotive components and approaches in between (e.g., the complex attacks demonstrated by [28] or [29]).

In this, automotive systems are similar to ICS and hence procedures learned for automotive forensics also hold true for the nascent field of ICS forensics.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

E. Implications on Forensic Investigations in ICS

Various factors have implications on the forensic process regarding ICS. These originate from the nature of components, structures and processed used in ICS. Previous work [1] discussed the implications of automotive IT on the forensic process and some of these changes hold true for the ICS domain. In addition, related work ([26], [30], [31], [32], [33]) identifies some constraints. This section aims at summarizing these constraints and discusses the implications. These challenges include:

- The field devices usually have a low storage capacity. This includes storage capability for events, errors or log files. Sometimes fault codes are implemented in a ring buffer where older fault codes are frequently overwritten with newer ones - sometimes field devices do not have any logging mechanisms at all [33]. In addition, it has been found that on devices where extensive logging is supported, this feature is often disabled, or the devices lack sufficient capacity to store enough data to allow analysts to meet forensics requirements [26].
- The often process-critical nature of ICS makes it more unlikely that a system will be powered off for a forensic investigation [31]. This leads to a heavier emphasis on live forensics.
- The nature of communication (network traffic) in ICS differs from classical IT environments. While classical IT networks contain mainly user-generated traffic, SCADA traffic is routine and predictable [32]. In addition, the amount of traffic is - by modern standards - relatively low. This can simplify network forensics in ICS systems considerably.
- ICS systems are built to last for a long time without any update or upgrade. As [33] states: "*It is common for an ICS system to run for 20 or 30 years without update or upgrade*". This leads to an abundance of legacy hardware, which is connected over legacy communication systems. In consequence, ICS tend to be even more heterogeneous that automotive IT. This includes hardware, software, interfaces and communication protocols used. Hence, specialized knowledge is needed to access, obtain and analyze the data obtained in ICS. This knowledge is often hard to access, since vendors rely heavily on their intellectual property to protect their business - reverse engineering is a common (and time-consuming) occurrence in this field of forensics.
- ICS are not geared towards security but towards safety [26] [33]. Most mechanisms aim at achieving availability. This means, that authenticity-centered mechanisms are not that common in ICS. This needs to be kept in mind during forensic investigations.
- Access to mass storage is more complicated compared to Desktop IT. In Desktop IT, mass storage generally can be easily separated from the system under investigation and attached to a forensic workstation. Here, write-blockers are utilized to prevent all write-operations on the mass storage. This guarantees integrity of the data. In ICS mass storage is often part of the MCU silicon itself, rendering the access a very complex issue. However, sometimes the program executed by the PLC in question is stored on a removable memory card. This card can be removed and investigated using read-only hardware, achieving a similar result to classical Desktop IT in terms of integrity. In general, accessing the mass storage requires the stopping of the PLC in question. However, alterations at runtime in main memory of the PLC are possible, which, of course, are not reflected on the program stored on the removable memory card.
- Access to volatile memory is only possible by using built-in diagnostic functions. If such functions are available at all, using them carries a high structural impact. In addition, they are usually only able to access a very limited amount of the volatile memory of the PLC in question. These built-in diagnostic functions might be the target of attackers. In the case of Stuxnet [23], the attackers altered the diagnostic functions in order to deceive the operator (or a potential investigator) by only delivering back information about the volatile memory purged of any traces of potential wrongdoing.
- ICS might also control critical infrastructures [10] such as emergency services, traffic control or power generation. In such cases, the consideration for powering off the given ICS in order to perform a thorough forensic investigation will often favor keeping the systems in question active. On the other hand, there might be some procedures required by law in the case of an incident. This could be the reporting of critical events or suspected attacks.
- Some ICS might feature redundant systems. This could be implemented by performing calculations in multiple PLCs in order to detect failures and hence to increase robustness. Another possible approach would be the inclusion of fallback devices in case one PLC fails. This might allow for a forensic investigation into one of the PLCs while the fallback devices keep the system running.

The limitations discussed in the previous section have a strong impact on the forensic process employed in ICS environments. While related work ([26], [30], [31], [32],[33]) mostly focuses on the classical IT part of SCADA systems, this section will discuss the forensic process aimed at the specific field devices found on level 0 and level 1 of the automation levels.

F. Data Streams in ICS Forensics

As discussed in Section III.A, three data streams can be identified in field devices: *communication*, *volatile memory* and *mass storage*. Forensic traces can be extracted from these three data streams. After discussing the implications of an ICS architecture on forensics in general, it is worthwhile to investigate the impact on the gathering and interpretation of these three distinct data streams.

Communication investigations can be described as network forensics since it encompasses data transmitted over network. In an ICS environment, this network might be field bus systems or dedicated lines. Communication can only be observed at the moment it occurs. Hence, traces originating from the communication data stream can only be gathered during the moment the communication is performed. As [31] puts it: "*Network forensics cannot be performed without mechanisms that systematically capture relevant traffic and state information throughout the network*". The fundamental question is therefore how to access the carrier mediums of the communication in question and how to analyze the captured data. Accessing the carrier medium requires either physical or logical access to the carrier medium or one of the devices involved in the transmission. Physical access would mean tapping directly into the carrier medium, while logical access would imply a device attached to the medium forwarding the communication to the investigator (or, more likely, a tool employed by him). Physical access would be the preferred method, since it allows for a more thorough control of the integrity and authenticity of the capture communication. Also, capturing communication in this manner is a purely passive affair, causing no structural impact on the system in question.

The analysis of the captured communication is generally more complicated compared with a conventional IT environment. Various protocols with proprietary extensions are used, increasing the need of manufacturer cooperation or reverse engineering. On the other hand, the predictive nature and relatively low volume of communication eases the identification of untypical events in the communication stream.

Access to *main memory* in general is only possible by sending requests to the respective PLCs. The accessible data is limited by the diagnostic functions of those PLCs. These diagnostic functions might be extensive in theory but are usually very limited or not available at all. This type of data gathering carries the same implications as in Desktop IT - sending these requests alters the state of the system under investigation (structural impact). Hence, it alters the communication on the field bus system transferring the requests to (and the answer from) the PLC and the specific PLC. While these implications seem grave, it might still be worth acquiring this data when the investigators take these implications into account during the discussion of the conclusiveness of the traces. Hence, the investigator should have an idea of what specific data should be requested in order to keep these implications low and predictable.

Mass storage in ICS consists of mass storage integral to the PLC silicon itself and, optionally, additional memory cards. These memory cards are used to store the executable programs while the integral storage stores the runtime environment. Access to the integral memory in general is only possible by sending (diagnostic) requests to the respective PLC. This carries the same implications and limits as with using (diagnostic) requests to access the main memory: structural impact cannot be avoided as the data gathered is limited by the availability of diagnostic functions.

Memory cards can easily be removed and investigated using write-blockers in order to maintain the integrity of the trace in question. As noted, these memory cards usually contain the executable program in question as well information about the hardware configuration and the project the transferred program belongs to [34]. However, potential alterations to the program after its transfer to the internal main memory of the PLC are impossible to detect with this method.

A serious drawback is the fact that all access to mass storage in ICS environments is only possible if the PLC is in stop mode. This carries the drawbacks of post-mortem forensics (the respective PLC is not available for operation) without offering the increased protection of the integrity of the mass storage since most data gathering will still be performed by sending request to the PLC. It might be possible to circumvent stopping the complete ICS if the system in question is sufficiently redundant. This might allow for the investigation of single PLCs while the system, as a whole, stays operational.

IV. Survey of existing tools and their applicability to the forensic process in ICS

Forensic Investigations in ICS explore a new domain of forensics. They need to be supported by tools in order to gather, evaluate and analyze forensic traces. This section gives an overview of tools and approaches usable during forensic investigations into ICS systems. It discusses the merits and pitfalls of the tools and identifies additional measures needed in order to employ them in a forensically sound manner. While this collection of tools focuses mainly on Siemens systems, many of the observations can be transferred to tools designed to access other hardware.

This survey is structured along the lines of the forensic process detailed in Section II.A and aims at identifying means to acquire the forensic traces identified in Table 1.

C. Strategic Preparation (SP)

This phase represents measures taken by the operator of an IT-system, prior to an incident, which support a forensic investigation. These measures often increase the possibilities available to the investigator during Data Gathering.

The foundation of a forensic investigation is obtaining a deep understanding of the respective system. This includes collecting any documentation of the electronic and electrical system in question. Wiring schemes and electronic parts catalogues, as well as repair manuals, are a vital source of information to decide on further steps to strategically prepare for a forensic investigation. In addition, they can form a solid foundation to make decisions during the later stages of a forensic investigation. This information is especially important during the Operational Preparation (OP).

A common type of SP is the activation of logging mechanisms already available in the specific components. While this requires only minor reconfiguration, further work might be needed to introduce adequate storage in order to record the resulting logs. This includes addressing the amount of data stored as well as supporting the integrity and authenticity of the data in question. The latter can be done by including cryptographic hashes and using a fixed, reliable time base for the generated logs.

More dedicated means of SP introduce interfaces to specific components that can be used during DG, if appropriate. This would include the installation of data taps for future access (Section IV.D discusses this topic in more detail).

SP also encompasses the installation of additional logging mechanisms geared towards the use during event reconstruction. One such example will now be presented in further detail.

1) *Forensic Agents*

The introduction of so-called 'Forensic Agents' into the SCADA architecture is discussed in [32]. This forensic agent represents data taps accessing the communication inside the SCADA networks. These taps are attached on level the levels 0, 1 and 2 of the standard Purdue Model.

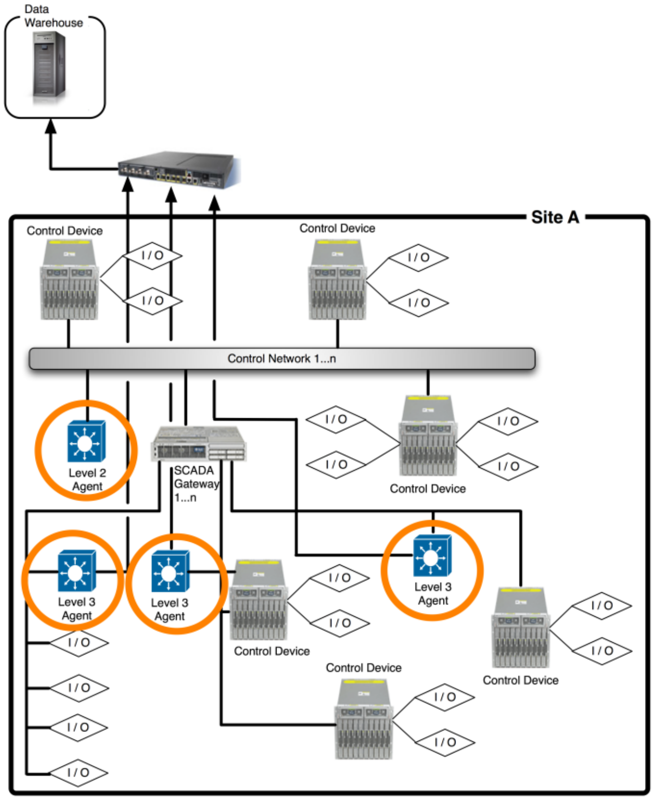

Figure 4. Placement of the Forensic Agents in Industrial Control Systems, according to [32]

Hence, they allow access to the communication inside the Cell/Area zone. The placement of these 'Forensic Agents' is shown in Fig. 4. The captured communication is then stored using data warehouse technology, making the events available for the use as forensic evidence. While this approach addresses the need for storage space, further refinement is needed in order to ensure authenticity and integrity of the respective captures.

This could be achieved by also storing cryptographic hashes generated over specific events (or timeframes) using an algorithm considered secure against collision attacks. Further, the inclusion of timestamps originating from a reliable time source is invaluable in event reconstruction.

Despite this additional effort, the introduction of forensic agents can greatly improve forensic capabilities.

In practice, a scaled down approach using only a subset of data taps or stores only a subset of events could also be viable.

Based on such agents it is advisable to implement means of intrusion detection. Such techniques could help to discover potentially anomalous system behavior as symptoms for initiating a forensic process. Moreover, the approach could be combined with the data reduction strategies, i.e., the complete traffic of all data taps is just recorded if it is justified by the symptom.

D. *Operational Preparation (OP)*

The Operational Preparation starts after an incident has been observed. In this phase, the fundamental decisions of the forensic investigation are made. The major question is whether to stop the suspect ICS or to keep it active. In an ICS environment, the system might have a critical function e.g. controlling critical infrastructure (emergency services, traffic control, power generation) or ensuring the safety of a plant environment. While in a classical desktop environment the decision on keeping the system productive or performing an extensive forensic investigation is often driven by the interest of stakeholders, ICS scenarios might well be influenced by the need of public safety. On the other hand, ICS scenarios are unlikely to contain much private data. Hence, they are less troublesome for the forensic investigator with regard to ensuring accordance to applicable data collection regulations and laws.

After the decision on keeping the system active or shutting it down (for forensic investigation and offline recovery), useful data sources are identified. As with the decision on the shutdown of the system, these considerations need to rely on the system understanding achieved during the SP. The availability of information on possible consequences as well as available traces greatly increases the ability to perform a well-informed OP.

In addition, there might be legal requirements to report incidents or suspected attacks on ICS systems. This might especially be the case in critical infrastructure domains.

E. Data Gathering (DG)

As discussed in Section III.F, three distinct data streams (*communication*, *volatile memory* and *non-volatile memory*) can provide traces for the forensic investigation.

Communication needs to be captured at the moment it occurs and requires physical access to the carrier in question. Given that access, there are tools for several of the communication bus systems used in ICS environments.

1) PBMaster

The PBMaster project offers an open software implementation of Profibus DP (Process Field Bus Decentralized Peripherals, see [11]) as presented in [35]. This project supports a wide range of hardware integrating UART (Universal Asynchronous Receiver/Transceiver) and RS-485 output. On the hardware-side it supports RS-232/RS-485 converters for Desktop computers, PCI based cards and ARM based boards. The software runs on Linux, FreeBSD, NetBSD and ARM based embedded systems. A Linux Live CD with all required components is provided as well. The project consists of a Profibus FDL master/slave station implementation, FDL/DP frame analyzer, FDL programming interface, a Live Linux CD and TCP/IP server for remote analysis of Profibus network traffic.

This tool does not provide any mechanisms to ensure integrity or authenticity of the gathered data. This can be addressed by using external mechanisms, like cryptographic hashes, to ensure authenticity and integrity of the gathered data. The passive reading access does not come with a structural impact.

However, this useful tool also serves as an example of the highly proprietary and intellectual property-protected tools in the ICS domain. Due to patent violations, distribution of the software is prohibited and limited to members of the Profibus International Organization and therefore not usable for an independent forensic investigator.

As discussed before, the access to *volatile memory* and *non-volatile memory* is indirect, since it relies on querying the PLC is question. In the following, two tools available to perform these queries for contents of *volatile* and *non-volatile memory* are presented.

1) NodeS7

NodeS7 [37] is a Node.js [36] library geared for communication with Siemens S7 PLCs. It provides functions to query values of variables as well as the possibility to write values. In order to work with the S7 1200 and 1500 series, an option called "Enable GET/PUT Access" must be set, which opens the PLC to third party software. This tool can be used to gather information about the current status of the PLC. Hence, it allows access to volatile and non-volatile memory.

To use it in a forensic environment, write operations should be disabled in order to minimize the structural impact. Additionally, no mechanisms to ensure integrity or authenticity of the gathered data are provided. This needs to be addressed by using external mechanisms.

2) Snap7

Snap7 [38] is an open source C++ suite for communication with S7 PLCs. The suite is able to read and write valuable information such as DataArea, DB, IPU, IPI, Merkers, timers, counters and variables. It can list, download, upload and delete blocks of data. Moreover, it is possible to retrieve detailed information on the PLC state, such as IDs, information on the CPU. It also offers the means to start and stop the PLC. The modification of this suite in a way that disables write operations would greatly increase the usefulness during forensic investigations. This could reduce structural impact. As with the other tools presented in this section, no mechanisms to ensure integrity or authenticity of the gathered data are provided. The integrity and authenticity of the gathered data needs to be addressed by using external mechanisms.

3) Soft-Update

Recent research into the behavior of Siemens S7 1516-F PN/DP PLCs lead to the discovery off an unknown behavior which might be useful for forensic investigations [39]. If the CPU of the PLC is in the 'RUN'-state during the loading of a new program, it briefly stops, loads the new program and restarts. During this process, the contents of the non-volatile memory are not overwritten. While this process has been shown in a proof of concept, a dedicated software solution for gathering this data is missing at the point of writing. Additional research into other PLCs might identify further PLCs where this approach of data gathering is applicable.

F. Data Investigation (DI)

Data Investigation represents measures to evaluate and extract data for further investigation. This includes data reduction and the identification of relevant data. It also includes the interpretation from raw data to (human-) readable information. In the classical Desktop domain, file reconstruction would be a prime example for a method used during DI. Another example is the dissection of captured raw communication. In general, tools that are able to 'make sense' (add semantics) of captured raw data by interpreting them are used in this section. One tool, that interprets captured raw data containing *communication* is presented and discussed here:

1) Wireshark S7Comm Dissector

The Wireshark S7Comm Dissector [40] is a tool able to interpret the raw communication between Siemens S7 PLC, the attached HMIs and the control system (the TIA - Totally Integrated Automation Portal, see [41]). This communication relies on a proprietary protocol using ISO-on-TCP packets. The application of this tool allows gaining meaningful insight into the communication between ICS components. While it is always best to use tools on copies of the captured traces, this tool performs no unannounced modifications of the captured trace. However, for a sound forensic process, the usage of external means to ensure integrity and authenticity of the processed traces is strongly advised.

The tool itself comes as a plug-in for the Wireshark [42] network dissector. The dissector for the S7Comm protocol, which is used by older models of the S7-300 and S7-400 series, is included in current Wireshark versions. The dissector for the newer S7 Comm Plus protocol, used by the newer series S7-1200 and S7-1500, needs to be installed as a plug-in (for Windows) or compiled with Wireshark itself (for unix-based Systems).

G. *Data Analysis (DA)*

Data Analysis brings the different traces gathered in a forensic investigation together. In this step, information is aggregated, correlations found and chains of events identified. While dedicated tools for the use in ICS environments are missing, some of the more generalized tools from the Desktop IT domain can be adopted for use during an investigation into ICS. This refers to tools, which help to create and organize chains of events, like Zeitline (see [43]).

In general, violating the authenticity and integrity of the traces processed in this step can be avoided by using copies of the original traces. The authenticity and integrity of the achieved results should be ensured. While Zeitline has functionality for this, most methods might require the use of external tools to achieve authenticity and integrity.

H. *Documentation (DO)*

The documentation consists of two parts. First, there is the process of accompanying documentation, which maintains an account of all the actions taken by the examiners. This process should ideally be highly assisted by software, recording all parameters and selected menu items. For desktop IT a range of dedicated IT forensic suites exist (e.g., X-Ways forensics [44]). In the field of non-standard forensic environments such a tool is missing. Hence, the investigator needs to rely on a mostly manual process involving screenshots, digital photographs, etc.

The results of the investigation are then compiled to a final examination report. This report describes the examination process and the results as well the most likely chain of events according to the reconstruction from traces. Usually, no dedicated tools are used for this process - besides a word processor.

V. Design recommendations for future Tools aimed towards forensics in Non-Standard IT environments

As depicted in the prior section and in previous work [1], there is a lack of tools geared towards the use in forensic investigations into non-standard IT. Major challenges are the heterogeneity of the domain (including hardware, software and communication protocols) and the reliance on proprietary and intellectual property-protected solutions.

Major work needs to be done to develop usable interfaces to access communication, volatile and non-volatile memory in order to acquire the forensic traces as identified in Table 1.

Especially for mass storage data streams, i.e., non-volatile memory, on levels 1 and 2 access today is either downright impossible (debug fuses set) or incomplete (e.g., only access to external memory chips) or at best very difficult using debug interfaces with undocumented parameterization and protocols (e.g., JTAG). Full access to the non-volatile memory should be provided, preferably using a serial high-speed interface and measures to ensure integrity and authenticity ensured using up to date cryptographic techniques. The same requirements should be placed towards the data gathering on volatile memory. Additionally, due to the volatile nature of the main memory content, a measure to halt the CPU register and RAM states (e.g., using non maskable interrupts pointing towards an integrity and authenticity ensuring dump routine) would add a new descriptive power to the traces gathered in memory forensics.

Further work is needed in order to interpret the traces gathered from these data streams with regards to semantics in order to support event reconstruction in more detail. This applies to both data investigation (allowing for a data reduction by excluding case irrelevant data) and data analysis (supporting the piecing together of the traces within the respective data stream to get a global picture of events).

For some isolated solutions, tools are available. However, these tools are not geared towards usage in forensic scenarios. Specialized solutions are needed here.

Previous work [1] already discussed criteria for the design of future forensic tools. Further considerations can be found in [45], giving the following requirements:

- the collected/processed data should be useful for the forensic process
- ensure the integrity, authenticity and confidentiality of the collected/processed data
- have a minimized and well-known structural impact, ensuring the integrity of the source data as best as possible
- document the actions performed
- the frequency of possible errors during processing should be known.

VI. Conclusion

This article presents the challenges of forensic investigation into potential security incidents in non-standard IT on the example of ICS and automotive environments. The growing interconnectivity in this domain comes at the price of an increased number of incidents - some of them caused by malicious attacks. This carries the need for forensic investigations into these incidents.

However, this article shows that forensic investigations in ICS environments still have significant shortcomings. The field is hampered by a severe lack of adequate tools owing to the heterogeneity of the ICS domain and the high barriers laid out by proprietary and intellectual property-protected solutions prevalent in ICS.

Approaches from the classical Desktop-IT domain can be adapted in order to preserve authenticity and integrity of the forensic evidence used during an investigation. The same holds true for the need of documentation of the whole

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

forensic process. Preferably, the tools themselves support the investigator in retaining authenticity and integrity while achieving a thorough documentation. Lacking that, tactics from the classic IT domain, which do not rely on internal tool support, can be applied.

In addition, the protection of personal data in accordance to applicable regulations and laws as well as adhering to regulations concerning the collection of data, especially in consideration of privacy laws and human rights, cannot be solved by the usage of tools alone - strong policies for the gathering and use of forensic evidence are needed.

The main contribution of this paper is the identification of 'white spots' where tailored and adequate solutions are needed in order to perform forensic investigations and giving guidance on the creation of such tailored solutions.

Acknowledgment

This document was produced with the financial assistance of the European Union. The views expressed herein can in no way be taken to reflect the official opinion of the European Union.

In addition, we like to thank our students working on automotive and industrial forensic topics.

References

[1] R. Altschaffel, K. Lamshöft, S. Kiltz and J. Dittmann, "A Survey on Open Automotive Forensics", Securware 2017, 2017.

[2] European Union Agency for Network and Information Securit y, "Communication network dependencies for ICS/SCADA Systems", ISBN: 978 - 92 - 9204 - 192 - 2, doi: 10.2824/397676, 2016.

[3] K. Inman and N. Rudin, "Principles and Practises of Criminalistics: The Profession of Forensic Science," CRC Press LLC Boca Raton Florida, USA, ISBN 0-8493-8127-4, 2001.

[4] M. Pollit, "Applying Traditional Forensic Taxonomy to Digial Forensics," IFIP International Federation for Information Processing, Volume 258; Advances in Digital Forensics IV, pp. 17-26, DOI: 10.1007/978-0-387-84927-0_2, 2008.

[5] S. Kiltz, J. Dittmann, and C. Vielhauer, "Supporting Forensic Design - a Course Profile to Teach Forensics," IMF 2015.

[6] S. Peisert, M. Bishop and K. Marzullo, "Computer forensics in forensis", In SIGOPS Operating Systems Review, Volume 42, Issue 3, pp 112-122, ACM, DOI=10.1145/1368506.1368521, 2008.

[7] T. Sugimura, "Junction Blocks Simplify and Decrease Networks When Matched to ECU and Wire Harness," Encyclopedia of Automotive Engineering. 1–7.

[8] A. Hillier, "Hillier's Fundamentals of Automotive Electronics Book 2 Sixth Edition," Oxford University Press, 2014.

[9] Robert Bosch GmbH, "CAN Specification 2.0, 1991" http://esd.cs.ucr.edu/webres/can20.pdf, (18/05/2018), 1991.

[10] Presidential Policy Directive/PPD-21, "Critical Infrastructure Security and Resilience," https://obamawhitehouse.archives.gov/the-press-office/2013/02/12/presidential-policy-directive-critical-infrastructure-security-and-resil (23/05/2018), 2013.

[11] PROFIBUS and PROFINET International: "PROFIBUS," https://www.profibus.com/technology/profibus/ (23/05/2018), 2018.

[12] PROFIBUS and PROFINET International: "PROFINET," https://www.profibus.com/technology/profinet/ (23/05/2018), 2018.

[13] Modbus Organisation: "Modbus," http://www.modbus.org/ (23/05/2018), 2018.

[14] T. J. Williams: "The Purdue enterprise reference architecture: a technical guide for CIM planning and implementation," Research Triangle Park, NC: Instrument Society of America, 1992.

[15] SANS Institute: "Secure Architecture for Industrial Control Systems," https://www.sans.org/reading-room/whitepapers/ICS/secure-architecture-industrial-control-systems-36327 (18/05/2018), 2015.

[16] T. J. Williams: "The Purdue enterprise reference architecture," Computers in industry Vol 24 (2). p. 141-158. 1994.

[17] Symantic: "Dragonfly: Cyberespionage Attacks Against Energy Suppliers," https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/Dragonfly_Threat_Against_Western_Energy_Suppliers.pdf (18/05/2018), 2014.

[18] N. Nelson: "The Impact of Dragonfly Malware on Industrial Control Systems," https://www.sans.org/reading-room/whitepapers/ICS/impact-dragonfly-malware-industrial-control-systems-36672 (18/05/2018), 2016.

[19] F-SECURE LABS: "BLACKENERGY & QUEDAGH - The convergence of crimeware and APT attacks," https://www.f-secure.com/documents/996508/1030745/blackenergy_whitepaper.pdf (23/05/2018), 2014.

[20] LOCKHEED MARTIN: "The Cyber Kill Chain®," https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html (23/05/2018), 2015.

[21] P. A.L. Ducheine, J. van Haaster, R. van Harskamp: "Manoeuvering and Generating Effects in the information Environment," in Netherlands Annual Review of Military Studies 2017: Winning Without Killing, 2017.

[22] R. M. Lee, M. J. Assante, T. Conway: "Analysis of the Cyber Attack on the Ukrainian Power Grid," https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf (23/05/2018), 2016.

[23] N. Falliere, L. O Murchu, E. Chien: "W32.Stuxnet Dossier," https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf (18/05/2018), 2011.

[24] Ralf Spenneberg, Maik Brüggemann, Hendrik Schwartke: " PLC-Blaster: A Worm Living Solely in the PLC," https://www.blackhat.com/docs/us-16/materials/us-16-Spenneberg-PLC-Blaster-A-Worm-Living-Solely-In-The-PLC-wp.pdf (23/05/2018), 2016.

[25] S. Gallagher: "Vulnerable industrial controls directly connected to Internet? Why not?," https://arstechnica.com/information-technology/2018/01/the-internet-of-omg-vulnerable-factory-and-power-grid-controls-on-internet/ (23/05/2018), 2018.

[26] M. Fabro, E. Cornelius: "Recommended Practice: Creating Cyber Forensics Plans for Control Systems," https://ics-cert.us-cert.gov/sites/default/files/recommended_practices/Forensics_RP.pdf (23/05/2018), 2008.

[27] European Parliament: "Odometer manipulation in motor vehicles in the EU," http://www.europarl.europa.eu/RegData/etudes/STUD/2018/615637/EPRS_STU(2018)615637_EN.pdf (29/05/2018), 2018.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[28] C. Miller, C. Valesek: "Remote Exploitation of an Unaltered Passenger Vehicle", Black Hat USA, 2015.

[29] Keenlab: "Experimental Security Assessment of BMW Cars: A Summary Report," https://keenlab.tencent.com/en/Experimental_Security_Assessment_of_BMW_Cars_by_KeenLab.pdf (29/05/2018), 2018.

[30] T. Spyridopoulos, T. Tryfonas, J. May: "Incident Analysis & Digital Forensics in SCADA and Industrial Control Systems," 2013.

[31] J. Stirland, K. Jones, J. Janicke, T. Wu: "Developing Cyber Forensics for SCADA Industrial Control Systems," 2014.

[32] T. Kilpatrick, J. Gonzalez, R. Chandia, M. Papa, S. Shenoi: "An Architecture for SCADA Network Forensics," 2006.

[33] P. Van Vliet, M-T. Kechadi, Nhien-An Le-Khac :"Forensics in Industrial Control System: A Case Study," https://arxiv.org/ftp/arxiv/papers/1611/1611.01754.pdf (23/05/2018), 2016.

[34] Siemens: „Structure and Use of the CPU Memory,"https://cache.industry.siemens.com/dl/files/101/59193101/att_897341/v1/s71500_structure_and_use_of_the_PLC_memory_function_manual_en-US_en-US.pdf (29/05/18), 2016.

[35] D. K. Tran, P. Pisa, P. Smolik: "An Open Implementation of Profibus DP," https://static.lwn.net/images/conf/rtlws11/papers/proc/p29.pdf (23/05/2018), 2009.

[36] Node.js, https://nodejs.org/en/ (23/05/2018).

[37] node7s, https://github.com/plcpeople/nodeS7 (23/05/2018).

[38] snap7, http://snap7.sourceforge.net/ (23/05/2018).

[39] (german) Oliver Keil: "Forensik in Automatisierungsystemen - Konzept zur Identifation und Erhebung verschiedener Datenquellen," Bachelor Thesis at Otto-von-Guericke University Magdeburg, 2018.

[40] s7commwireshark, https://sourceforge.net/projects/s7commwireshark/ (23/05/2018).

[41] Siemens: "Your gateway to automation in the Digital Enterprise Totally Integrated Automation Portal," https://c4b.gss.siemens.com/resources/images/articles/dffa-b10161-00-7600.pdf (23/05/2018), 2016.

[42] wireshark: https://www.wireshark.org/ (23/05/2018).

[43] F. Buchholz, C. Falk: "Design and Implementation of Zeitline: A Forensic Timeline Editor," http://www.dfrws.org/sites/default/files/session-files/paper-design_and_implementation_of_zeitline_-_a_forensic_timeline_editor.pdf (23/05/2018), 2005.

[44] X-Ways Software Technology AG, "X-Ways Forensics: Integrated Computer Forensics Software," http://www.x-ways.net/forensics/ (23/05/2018).

[45] M. Hildebrandt, S. Kiltz, J. Dittmann: "A Common Scheme for Evaluation of Forensic Software," In Proceedings of the 6th International Conference on IT Security Incident Management and IT Forensics (IMF2011) Stuttgart, Germany, 10.05.-12.05.2011, ISBN 978-0-7695-4403-8, pp. 92-106, 2011.