NTNU
Norwegian University of
Science and Technology

Cyber Situational Security Awareness Architecture (CSSA) for Industrial Control Systems

Niclas Hellesen

01-06-2019

Master's Thesis
Master of Science in Information Security
30 ECTS
Department of Information Security and Communication Technology
Norwegian University of Science and Technology,

Supervisor: Prof. Stephen Wolthusen
Co-Supervisor: Dr. Vasileios Gkioulos

Preface

This is a Master thesis in information security at NTNU carried out during spring semester of 2019. The project was provided by KraftCERT through my supervisors at NTNU. The goal of the thesis is to lay the ground work for a cyber situational awareness architecture industrial control systems. It also contains a chapter on digital forensics relevant to the architecture and a chapter on penetration testing in industrial control systems. Penetration testing is meant to give a proactive effect, the architecture provides real time situational awareness and the digital forensics deals with post incident situational awareness.

The thesis is aimed at readers who own or control industrial control systems, or have interest in the topic.

01-06-2019

Acknowledgment

Thanks to my supervisors Stephen Wolthusen and Vasileios Gkioulos for their guidance during the master project. I also wish to extend my tanks to Lars Erik Smevold from KraftCERT for his guidance and to the expert from the power utility company that participated in the unstructured interview.

N.H.

Abstract

This thesis covers the ground work for a cyber security situational awareness architecture for industrial control systems. Furthermore, it answers research questions about requirements for capturing context and results from penetration testing in industrial control systems and requirements for forensics data collection relevant to the architecture. The purpose is to provide protection provided in different chronological stages. Penetration testing works as preventative work. The situational awareness architecture provides real time monitoring. Digital forensics provide additional understanding of an incident after it has occurred. The architecture will also be able to utilize stored historical data and data shared by external communities in the correlation process. In addition to uncover attacks against the organization, the information produced by the correlation phase should be useful in predicting trends. This means that the architecture should be able to contribute with more than real-time situational awareness. Normalization of information is a part of the architecture, where it transforms information into standard formats. Sharing of cyber threat intelligence with communities is important in order to create a proper threat overview and facilitating collaboration to uncover threats. The architecture utilizes technology to automate sharing of cyber threat intelligence in a secure manner with different communities without leaking sensitive information or reveal inner workings of the organization. After the information collected from internal and external sources are correlated, the results is presented in the visualization phase.

Contents

List of Figures

List of Tables

1 Introduction

The autumns research project planning course was a precursor phase leading to the masters project. The research project planning was therefore a preparation project which the master is built on.

1.1 Situational Awareness

Situational awareness is based on discovering, observing and understanding elements in an environment relevant to the situation and context of those who are conducting the observations. To use an example, imagine an operator entering an area of a factory and then observes in front of him or her a set of screens, a conveyor belt, and a set of buttons and levers belonging to a human machine interface (HMI). Observing these elements of the environment such as identifying them as screens and other machinery, their attributes such as placement relative to the other machines, and speed some machine parts are traversing a shop floor is what Dr. Mica Endsley would call level one situational awareness, which she refers to as Perception of Elements in Current Situation [5]. In her publication there exists three such levels. Level two is about observing and understanding patterns in the elements and is named Comprehension of Current Situation. The third level of situational awareness is about perceiving future states of the elements based on information gathered from the two prior levels.

Barford et al. defines a set of seven situational awareness requirements [4], later used by Mavridou et al. to create an architecture for situational awareness for Smart Grid [3]. Mavridou et al. work are described in the Related Works chapter 3.3. The situational awareness requirements is listed in figure 1 that Mavridou et al. compiled [3]. Barford et al. describes that cyber situational awareness consists of three processes starting with situation recognition, followed up by situation comprehension and then situation projection.

1.2 Keywords

Cyber security, situational awareness, architecture, industrial control systems, requirements for context and results of penetration testing in ICS, normalization, correlation, visualization, requirements for digital forensics data collection

1.3 Problem and purpose statement

It is difficult to gain understanding and overview of problem situations and even knowledge of their existence, especially when the problems exists within systems that in them selves are complicated pieces of machinery. And they may exist within even larger complex systems. The purpose of this thesis is to help increase the situational awareness by contributing with an architecture that increases cyber security situational awareness, as well as analyze penetration testing and digital

forensics on industrial control systems.

1.4 Topic covered by the project

This thesis is about obtaining cyber security situational awareness within industrial systems, however, its contents should in many cases be applicable to information systems in general. Since the project is provided by KraftCERT, which is the Norwegian computer emergency response team for the energy sector [6], the energy sector will put its color this project. Situational awareness in the context of information technology (IT) and operational systems (OT) deals with obtaining knowledge about the systems, how they are put together and how they can be misused by people with bad intent. It also means being able to anticipate how systems may react to different situations and how other processes are affected by those reactions. Situational awareness is also gained by obtaining a threat overview, by utilizing the information mentioned with how an attacker may proceed with causing as much harm as possible together with information from external organizations that own similar machines. In order to gain situational awareness it is therefore necessary to process as much relevant information as possible in a way that gives the good results. Therefore requirements for context and results of penetration testing as well as requirements for digital forensics must be in place.

In order to help solve the problem of obtaining situational awareness, the ground work of a cyber security situational awareness architecture (CSSA) was created with the processes needed to support it. The thesis contains a requirements analysis for digital forensics data collection relevant to the CSSA, attack scenarios and an analysis of requirements for capturing context and results of penetration testing in ICS.

The power grid is likely one of the most critical infrastructures a nation controls, which can be argued by stating that other critical infrastructures such as communication networks, water and waste water relies on electrical power.

1.5 Problem description

Attacks occur all the time against private citizens, businesses and governmental agencies. It appears to be an eternal cat and mouse between attacker and defender. The attackers may attack from any angle, both from the outside and the inside of the organization. Even though more resources are being used to defend the systems [7], the systems are in turn becoming more and more complicated, not only with new technology that gets developed to solve new problems, but they are in many cases integrated and have to work with older technology that already exists in the plant. Physical machines and software may experience problems which sometimes lead to the need of repair. When an operator or security personnel discovers such an event they need to make a decision if it could have been caused with malicious intent. Therefore is important that systems are designed to ensure that the right information with high integrity is communicated to the operators in a way that does not cause information overflow.

We all know the CIA triad that translates to confidentiality, integrity and availability. However, in industrial systems the triad is reversed resulting in availability being on the top [8]. The reason is that it is extremely important to prevent downtime. Think about systems producing electricity to a city, and how unfortunate it would be if an attack caused a power outage. An example of this could be the attack at Ukraine [9] that caused 225,000 people to loose electricity for up to three hours. A series of other examples could be imagined, such as there being a factory where many machines rely on other machines in order to fulfill their tasks, and if one the systems goes down it wont be able to produce what the next machine in the chain needs. Integrity which is the number two on the list is also important for many reasons. Systems and operators rely on the integrity of the data and if the data gets corrupted, maliciously altered or completely changed it could cause damage and harm. An example could be the STUXNET attack on the Natanz nuclear plant in Iran where the worm recorded legitimate sensor data so that it could replay it later in order to mask its malicious controlling of the plants uranium enrichment centrifuges [10, 11]. Lets also look at one more example from another industry as well. There was a case where a producer of cat and dog food mistakenly added too much magnesium in the food causing stomach upset for weeks in many animals where some had to be put down [12]. In this case it was a mistake rather than a security incident at the factory that led too much of one chemical to be added to the food which then lead to serious illness, according to the article. This helps shed light on how dangerous it can be if a machine at a factory did not follow the recipe correctly. Therefore, if an attacker managed to replicate a similar incident by attacking the integrity of the information the factory machines use to produce what we consume, it could possibly cause allot of harm.

1.5.1 Situational awareness in real time

In order to look closer at the problem of identifying and handling problems in real time allot can be said. One issue with identifying problems in real time is that large amounts of data is run through the correlation systems. It may therefore require allot of computing power and storage, and the processing methods may be complicated which adds more load. Displaying raw system and sensor data to operators would overwhelm them, and therefore it must be processed first. Both the raw and processed data can be stored and used later in analyses to help reveal larger patterns of malicious acts that was not revealed during real time computing. Therefore this stored data may be used to enhance decision making in the long term. Ultimately, this is about creating and then improving an ability to discover situations when they arise. Situational awareness, which this master thesis is about is the first half of the process in handling problems. First it is necessary to understand the problem, and then based on this knowledge a problem solution can be crafted.

In order to have the ability to make decisions in real time, it is important that for the most part only information relevant to the task is collected, because collecting too much information will make it more difficult to use it in real time in a good way. To collect the relevant information is not easy, and requires first of all understanding on what is relevant and how to reduce noise. The process of obtaining awareness deals with boiling the collected information down to something

specific that can be acted on. This also raises issues, as the process of reducing the size of the data should not inadvertently remove some important data that could have been used to reveal patterns of an attacker. The data should also be made human readable in both size and presentation. Further it is possible that the data that is being used in the real time processes might have had to go through construction and correlation from different data sources and in different data types before it could be used. This shows that there are numerous problems that has to be solved.

1.6 Justification, motivation and benefits

The purpose of this task is to provide security and safety in the power plants as well as contribute to reduced societal risk of loosing critical infrastructure and functions, increased safety at workplaces, reduced cyber warfare threat from other nations, reduced cyber threat from terrorists and reduced societal monetary costs by reducing amount of incidents.

1.7 Research questions

In this master thesis I am answering the following research questions:

1. What are the requirements for capturing context and results of penetration testing in ICS?
2. What are the requirements for digital forensics data collection relevant to CSSA?
3. Which are the architectural components, functionalities and communication patterns required in order to support CSSA within ICS?

1.8 Planned contributions

The planned contribution is to lay the ground work for an architecture for cyber sitautional awareness in industrial control systems. The ground work is meant to be helpful for the next master projects that are going to implement the different phases of the architecture. However, it is not meant to describe how each aspect of the technology is to be implemented, but contribute as ground work and describe research and technology used in similar use cases. The purpose is therefore to give a starting point other can build their research on. In addition to that it also provides discussions on penetration testing and forensics in industrial control systems.

2 Choice of Methods

The research approaches for this thesis will be mentioned with an explanation of choice. The methods are literature review, unstructured interview, experimental development, requirements engineering and security engineering.

2.1 Literature Reviews

A literature review of a collection of scientific publications was conducted. It was conducted in three iterations, one for each research question. However, more information on aspects of the architecture was desired and search related to correlation, normalization and visualization was conducted. Several scientific databases was searched. These where Springer, ACM, Science Direct, and IEEE while Lovdata was accessed to find Norwegian laws. In addition to that, searches was also done in Google Scholar. Included are also papers that was found in the reference lists of the selected literature collected from the scientific databases, as well as documents found on governmental websites.

2.1.1 Unstructured Interview with Expert

The unstructured interview was conducted as a discussion with prepared questions. It was planned to be set up in a way that the expert was not bound to the questions. The questions was there to be something the discussion could grow around.

Since this thesis was part of a larger project, it means that the implementation and further research on each section of the situational awareness architecture is a project by them selves. The unstructured interview has been conducted collaboratively with Kari Anette Sand [13], nevertheless, the focus and the questions where distinct. It was conducted with an expert from a utility company. Questions asked was meant to provide aid in the overall process of the project. The first question was if the amount of equipment and systems they had would hinder them in obtaining situational awareness. The answer was that it could be difficult at times, but they had people with deep knowledge of parts of the systems that worked as domain experts, and personnel that had less domain specific knowledge but possessed a large overview picture. However, he stressed the importance that as many as possible had a good overview picture. The next question was if there was any requirements for conducting penetration tests where the answer was that they did conduct tests but there was no requirements for them. While discussing this question they described what they considered highlights of a pentest. He start by mentioning that for the pentesters it would be to obtain user rights and privilegie escalation. To solve the problem may some times be a challenge since in their experience pentesters may forget the steps they did. This would also lead to a problem with reverting changes done to the system during the test. Then, of interest would be if the testers was able to obtain information from the systems, since these systems contains mostly information

that is not public. He describe that this information should be valuable for potential attackers. The next question was if they had any documented procedures for digital forensics. He answered that they had routines for incident handling and recovery. When an attack have happened they will try to analyze how it spread. They would isolate machines and let human operators take over control. The next question was about which machines it could be possible to obtain forensic information from, such as PLCs, RTUs or other machines one usually would not think about as a source of such information. He mentioned that syslogs are obtained from any machine possible as well as triggers that reads syslogs. And that they take a backup of config from network infrastructure each day. We then discussed ways to protect the systems. He start by describing that they register known vulnerabilities. All suppliers has to use their VPN system that utilizes account login with two way verification when they work on their systems. These accounts is only valid for eight hours, and if the supplier has completed their task before that time interval ends, they are required to notify the utility company in order for them to close the temporary account. The VPN system also requires the suppliers to connect from a list of known IP addresses.

2.2 Experimental Development

By using UNESCO's description, experimental development is a development methodology based on utilization of prior research and practice based knowledge [14]. Furthermore, this method should help to bring about new knowledge in the area with the intention of improving the processes of production of the relevant products as well as increasing their quality [14]. UNESCO has based their definition on the OECD Frascati Manual 2015: Guidelines for Collecting and Reporting Data on Research and Experimental Development [15]. This project will need to build on knowledge created by others, it will need a systematic process to develop the architecture, and then it will aim to generate value by proposing new knowledge. This therefore describes the process of experimental development, such that it appeared as a natural choice of method.

2.3 Requirements Engineering

A part of this project is to obtain requirements for both capture of context and results from penetration testing in industrial control systems and requirements for digital forensics data collection. A process to do this is called requirements engineering.

Nuseibeh et al. describes the core requirement engineering activities as eliciting, modeling and analyzing, communicating, agreeing, and evolving requirements [16]. They describe that it is important to identify system boundaries which in this context could be the boundaries for what the requirements should not cross, so that there are no requirements that are unnecessary for practical use when doing penetration testing or forensics.

Zave who discussed requirements engineering for software development writes that requirements engineering is a process that creates an overview and framework for further study [17]. She further writes that there are several dimensions of problems that has to be solved. The first dimension concerns the data collection, processing and development of strategies. In this case the strategy development should result in additional alternative strategies. One of the problems in this dimen-

sion is *overcoming barriers to communication* where she explains that the requirements engineers has to communicate with a large group of people with different professional backgrounds that may have different goals. There are a total of six problems that Zave described under this dimension [17]. In the next dimension she explains problems regarding software behaviour, followed up by the third dimension regarding problems with evolution of systems and their system families.

One of the first requirements for the architecture is that it should be a tool that can be utilized to uncover what potential threat lies in an event or a collection of events to the organization. It should be able to store data to enable generation of trends, to store information about threats and information that describes normal states. A examples of storing normal states are configuration files and hash values of ladder logic that can be used to discover changes. The architecture must be able to correlate large amount of data and information and provide useful cyber threat intelligence in a reasonable time frame that can be used by the organization to protect from both outside and inside threats. It also needs a way to provide useful presentations of threat information to personnel. It is likely that attackers will try to harm the situational awareness architecture or hinder its functionality. Therefore it should be robust and hardened.

There are also requirements for penetration testing and digital forensics which will be discussed in chapter 6, 7 and are summarized in the conclusion. These requirements are influenced by the need of availability in the industrial control systems.

2.4 Security Engineering

The architecture for cyber security situational awareness creates visibility and allows the operators to see what happens in the system as well as understanding the different situations. The design behind the architecture and its components and functionalities that realize the architecture could be a priority for an opponent to attack. Therefore every part needs to be designed with security in mind, and can be hardened with exercising of attack scenarios. Security engineering is based on creating systems that are designed to be as hardened as possible so that it can resist failures and attacks [18].

3 Laws, standards and related work

This chapter contains related work for the architecture and its components, penetration testing and forensics, as well as some laws and standards.

3.1 Standards

In this section standards important to the information security of power utilities which are relevant to the research questions will be mentioned. If one document overlap another one to a high degree, only that will be described, but both mentioned.

The ISO/IEC 27019 standard about security techniques and information security controls for the energy utility industry covers thirteen topics, and works as an extension to the ISO/IEC 27002 [19]. It describes three sources of requirements of the energy utility, those obtained from risk assessments, juridical and the societal. Since society is dependent on the continuous power supply, it therefore becomes a requirement that the energy utility organizations are able to maintain the delivery. Further, it puts weight on the importance of being in contact with relevant authorities, utilize network segmentation and segregation, physical access control and managing risks that such an organization may be exposed to. Such risks involve supply chain attacks, attacks against third parties that have access to the organizations systems, teleworkers and more. It suggests including all systems, software and services in an inventory of assets, which will help in detecting if a system has any known vulnerabilities. This will also help identify which systems is owned by who if some of them are stored in a location shared with one or more other energy utility organizations. It also touches upon legacy systems, and the secure handling of them. If known vulnerabilities exists in them, and that these cannot be removed, then security mechanisms should be implemented to prevent them from being exploited. When it comes to classification of information, they define energy utility specific classification criteria. Examples of these are assets, systems and information needed after blackout to start systems, that supports health and safety of the workers and more. It contains descriptions of securing control centers and equipment rooms from natural disasters, and describes a requirement to restrict unauthorized personnel. It recommends the use of anti virus (AV) on the control systems and if installing an AV is not possible, then it recommend other mitigation strategies that should be implemented, such as securing physical and logical interfaces, network isolation and segmentation, and more. However, the latter should always be done even when running an AV that is supported by the control systems, and network segmentation is already suggested throughout the document. When it comes to logging, it states that event logs may include operator actions, and that it could be a requirement from regulatory bodies to store these logs. It defines the logging of changes done to safety systems as a requirement. Other topics it covers are management of vulnerabilities by obtaining inventory information from system integrators when an upgrade, install

or change to a system has been done, the requirement of least functionality, technical compliance review, and telecommunication with suppliers regarding crisis and emergencies.

In addition there is the NIST SP 800-82 that provides comprehensive guidance on protecting industrial control systems [20]. IEC/TS 62351 is a series of standards on power system data communication security [21].

3.2 Requirements for capturing context and results of penetration testing

The Norwegian law of national security is a law from the Ministry of Defence, which recently experienced an update first of January 2019 [22]. The law describes requirements for preventative security for the ministries within their area of responsibilities and requirements for conducting inspections. The new version contains a paragraph with the Norwegian name §6-5 penetrasjonstesting av skjermingsverdige systemer, which is translated into penetration testing of systems with a high need of protection.

The Norwegian Water Resources and Energy Directorate (NVE) has the responsibility to manage Norway's water and energy resources [23]. They published a guidance on regulations and emergency preparedness which contains requirements for organizations with the role of being electricity suppliers [24]. Those responsible for implementing the emergency concept defined by NVE is defined to be followed by something called KBO units [25]. KBO is the power supply's emergency response organization and is made up of NVE and the organizations who produce electricity. The guidance from NVE has a chapter named *1.1.2 Extraordinary events* which states that assessments has to be undertaken to map vulnerabilities [24]. The findings is then to be used for decision making if additional measures should be taken.

Christiansson et al. argued back in 2007 for a European SCADA security testbed [26]. They wrote about the benefits of having a system to run penetration testing on that would not harm a physical plant. They mention that at the time of writing the paper, the US had such a facility, however the critical infrastructure facilities in Europe was different compared to America. They state that a test bed would help develop best practices in Europe. Considering this, having a testbed to do penetration testing of the ICS could allow companies large enough to afford it, or cooperation between companies to do extensive penetration testing on a testbed that simulates their own systems. If not the whole systems, simulate them part by part. This could therefore provide both information about software vulnerabilities as well as testing of attack scenarios.

The Norwegian Cyber Range was unveiled at NTNU the 4 of September 2018 and is an arena where actors of interest such as students, companies and others can conduct practical exercise [27]. This cyber range is likely to be utilized in the masters projects relevant to the CSSA architecture created in this thesis, as the cyber range offers access to relevant ICS SCADA devices.

A Sandia report from 2005 discusses correct approach for discovering networks, hosts and vulnerabilities in SCADA systems [28]. They start by mention the importance of identifying the goal of pentesting each device before listing preferred actions for SCADA systems compared to IT systems. These actions range from how to identify hosts, services and vulnerabilities within a service without putting the live SCADA system in danger. The main points of interest is to never test directly

on live SCADA systems but rather on a lab system that mirrors the live system. And if testing has to be done to a live system, they stress the importance of having operational personnel available and prepared to handle possible problems caused by the testing. When it comes to identifying hosts they mention the possibility of doing so by examining CAM tables on switches, router config files and route tables and conduct passive listening to network traffic. Hopefully the mirrored test and development systems contains the same services and topology as the live system, and can be used as a source of knowledge of the running systems services.

3.3 Architectural components, functionalities and communication patterns required in order to support CSSA within ICS

Work on creating architectures for obtaining situational awareness in different contexts and with the use of different technologies exists, and some have also tried to incorporate data fusion to their models. Many different projects will be described but first a description of what JDL Data Fusion is will be provided, followed up by an example of an situational awareness reference model that combines the use of Endsleys situational awareness with JDL Data Fusion is looked at in more detail.

JDL Data Fusion was designed to allow data from several sensors to be combined in order to obtain more information from that data, where an example from the JDL Lexicon are objects physical location in space [29]. The purpose of the JDL Data Fusion is to provide a structured way of combining data so that new information not available in the original data set are generated. Hall et al. described four JDL data fusion levels starting with object refinement, situation refinement, threat refinement and processing [30]. The object refinement level aims to obtain knowledge about the state of the object, by combining location, identity information and parametric. A situation can be made up by several objects and events, and situational refinement aims at describing the relationships between them. The threat refinement, which will be the third JDL level works on projecting the future, much like Endsleys situational awareness level three. The fourth level named processing works on monitoring performance and improving the three previous levels.

The first example looked at is the work of Tadda et al. who combined Endsleys situational awareness with JDL Data Fusion [31]. They created a situational awareness reference model and a process model. The reference model works as a set of definitions and collects the definitions based on context, and the process model shows the flow of a single process at given time interval. In the situational awareness reference model, level zero is the data and its sources, and level one is where the human mind understands that information from the world around him or her. Level two is information on self which is called “us”, where level three is then the understanding obtained on other objects called “them” including possible futures, impacts and threats. Level four appears to be the fourth of the JDL levels which as mentioned aimed at refining the whole process of the previous JDL levels, and in this case the levels used in their situational awareness reference model.

Taking a closer look at level one, Tadda et al. define a situation as: *“a person’s world view of a collection of activities one is aware of at an instance in time”* [31]. Each situation contains a set of activities where an activity is defined as an action or movement. A group is anything that can be grouped

together and obtain the metadata of belonging to that group. An event is a single occurrence of something that have happened. They define an entity as something material or non-material which is distinct and of separate existence. Followed up by an object which is a physical entity and finally a concept which Merriam Webster defines as an: *"abstract or generic idea generalized from particular instances"* [32].

Level one in their situational awareness reference model can therefore be understood as the pre-processing where metadata is applied to help categorize data into events, objects, entities and concepts as well as help group the data. It is then the level two and three that works as the situational assessment of the collected and pre-processed data.

Their situational awareness process model looks at a single activity of interest at a time with the goal of transforming the information into situational awareness for the user. The situational awareness process model starts with a set of observables, which is anything that can be observed. These observables is combined with a priori model knowledge to identify what activities they resemble. This leads to two possible streams of actions. The first is damage assessment and describes impacts that are already experienced. It is important to determine the damage caused. To determine damage it is necessary to have knowledge of which defines the importance of the assets and capabilities of the organization. This is information they call "knowledge of us". The other stream starts with the possible futures of these activities combined with known configuration data. The next two steps are more intertwined. They are plausible futures combined with knowledge of *them*, and potential impact/threat combined with knowledge of *us*. Therefore one can say it is an attempt to determine potential steps the attackers might do, by trying to figure out what possibilities the attackers may have to reach their goals, and also to obtain information about new data to collect that would help in providing situational awareness.

Shortly summarized, their situational awareness process model is divided into two paths, where the first tries to figure out the damage caused by existing events by combining knowledge about those events with "knowledge of us". The second path tries to obtain awareness about what the attackers might do next.

When it comes to measures of performance and effectiveness they substitute attack tracks with activity of interest and enables measurements of the systems ability to fuse evidence, create attack tracks and prioritize them. The four metrics they have is confidence, purity, cost utility, and timeliness. With confidence they mean how good the system is at detecting actual attacks. Purity looks at the quality of evidence behind the detection of an attack track. The cost utility in their case measures the amount of work needed to not only do the task, but to reach the task. This means that if there is a prioritization list of attacks that a human operator goes through, and the top attacks are false positives while attack number five is a true positive, then there will be four possible attacks that an operator has to process before he or she reach the true positive attack. The cost utility therefore sums up the work that the system operators has to go through. The cost utility helps with measuring a systems ability to identify the attack tracks necessary to thwart the attacks. Timeliness is described as both the time that the system gives information within the time frames required as well as the time it takes to make the operators gain awareness of what is going on. Overall they

write that their measures of effectiveness (MoE) is a measure on how good the system is at giving the operators situational awareness. And to add to this, they also describe something called Data-Information Ratio (DIR). Since the data collected is often allot more than what the human operators can process, as mentioned above, there is a need to compress the data before visually presenting it to the users. DIR therefore works as a measure on how much the data has been reduced in amount when presented to the users.

Work has also been done on situational awareness for detection of insider threats. Brancik et al. identified a high level architecture and mechanisms for both protection and detection of insider threats and puts extra effort on data exfiltration [33]. Their situational awareness architecture includes techniques to find the source of data, as well as use of data correlation. They list and describe the components of the architecture and points out that it works by processing large amounts of events. The components are event and anomaly collection, data analysis and correlation, E-discovery tools and security information systems tools. E-discovery tools is described as tools that collect evidence from e-mails, logs and more. The security information system tools are described to be systems that can take actions against attacks discovered by their architecture. They document that they collect data on network and services wherein the following is listed: security and audit logs, traces of network flows and service logs. As service logs they give email and web access as examples. They defined the complexity of the networks, applications, services and other factors as heterogeneity and state that this heterogeneity is a factor that allows attackers many potential attack vectors. And when it comes to insider threats, the attackers can hide information they exfiltrate inside legitimate traffic or with encryption. This will give an IDS problem with detecting the exfiltration. Their approach of correlating large amounts of data is designed to help counter these problems.

Their data analysis and correlation phase can be summarizes as using neural networks and associative memories (NAM) to identify normalcy benchmarks. If events don't fit with this benchmark, then they are flagged as suspicious.

Yen et al. describes a problem where human operators or analysts experiences a problem with obtaining cyber situational awareness due to the complexity of the situations as well as limitations of communication between the machine cyber world and the human mental world [34]. In order to aid in this situation they developed a framework that is supposed to close the gap between human mental model and the logic model of their tools. Decision makers looks back at their previous experience in order to see if they have experienced similar situations before, and if they have not, they describes that the decision maker may use something called story building. This is a process where observed information are linked together in order to construct a story that may explain the situation. To help with this they explain a model called Recognition-Primed Decision (RDP) that is a naturalistic decision-making model which contains the knowledge of how experts on specific domains gain situational awareness, and works by using recognition and evaluation phases. This combined with another model called Cognitive Agent Architecture (R-CAST) they create digital agents which is computerized intelligence's that is supposed to help the analysts by providing knowledge and insight. R-CAST allows the agents to obtain the ability for expectancy monitoring and story building.

They write that the digital agents are able to communicate to each others using Service-Oriented Architecture (SOA). They use hypergraphs to monitor relationships between events and hypotheses in several layers.

Deng et al. describe a distributed architecture for real time event capture, correlation and dissemination including an event correlation engine, a temporal-spatial event correlator mechanism and an automated publish and subscribe paradigm [35]. The events are attempts, successful or not, that an attacker disrupt a service or conducts other harmful acts. These acts leaves traces that can be used to gain situational awareness. An underlying idea of their event correlation is that analyzing single events may not help detect an attack, however correlating them may reveal the it. They explain the processing flow in four stages. The first stage is the metadata generation which works as pre-processing. This is followed up by event generation which receives input from two sources, system monitoring and the metadata generated in the first stage. When the events are generated they are run through their event correlation engine in the third stage. The results from this is sent to the fourth stage called event management and reaction.

The event correlation engine which resides in their SCADA master receives events from the Event Generators. These events are generated from field site components and is based on data from either physical processes or metadata. The events are then placed in a buffer awaiting processing which are connected to a priority policy module that helps prioritize the high-risk events in the event queue. The engine is also able to process the events in correlation with historical databases. The event queue leads to a Format Decoder module which they explain extract effective segments. The event correlator uses both the information from the format decoder as well as historical event data. The temporal-spatial event correlator mechanism improves event correlation accuracy and is a two dimensional correlation in time and space. Temporal correlation is monitoring of events over time, such as login attempts, and spatial correlation is cross process monitoring. Then the processed information is sent through a rearrange and output module to be prepared for the event management and reaction module which is the end-software and operators.

The publish and subscribe mechanism is made up by an event subscriber server that are able to subscribe to event publishers which is the event generators. The subscriber will only receive events from publishers it subscribed to which will reduce computing and network use. It is these events that are fed into the event correlation server. When the correlation engine has processed the information, the results are sent to the the event rearrangement and output module that adds statistical information to the output data.

When an incident has occurred in critical infrastructure, it is necessary to communicate this with other relevant stakeholders. Work has been done on topic by Hayretdin et al. who proposed a situational awareness framework for critical infrastructures that aims at providing decision support at national levels and help detect coordinated cyber attacks against critical infrastructures [36]. They also describe that it contains ability to correlate effects of attacks between different critical infrastructures. It is based on a pyramid of four different levels, corresponding to different roles. The roles are as following, from the first to the fourth: national, strategic, tactical, and operational. The framework is made up of three subsystems which is organizational situational awareness, national

situational awareness, and CI Honeypots.

Pournouri et al. used decision tree algorithm with four classifiers working on historical data about cyber attacks to provide estimates [37]. These estimates helps an organization obtain situational awareness. For example which organizations are most favourable for attackers based on target categories, and which threats are most likely to target a specific organization. They do however mention that their model did not do an acceptable job when it comes to these two predictions. The decision trees are written in R and the code is provided in their publication, which allows other researchers to learn from how it was done. They also suggest Naive Bayes and Neural Networks and mention that those might work better on their data set.

Work on obtaining situational awareness has also been done on the smart grid such as Mavridou et al. who created an situational awareness architecture for the Smart Grid [3]. With the help of the University of Tulsa they got access to a prototype of an electric power substation with 3KVA power transformers. The transformers communicate with a PLC over Ethernet using DNP3. Their architecture design in hardware has two sections each with three control devices. They are connected to the SCADA gateway through sensors, that they describe as having been placed at different strategic locations. From there the data is placed in a database before it can be accessed by the last section of their architecture, the Command Center. The command center is filled with programs that allows alert reporting, event correlation, trending, integrity checking and ability to launch commands back to the machines and systems below. Taking a closer look at the sensors, they run in promiscuous mode and can be interacted with in order to configure their network interfaces or generate reports. The SCADA gateway was tasked to convert the data formats it receives into other formats before storage. They write that their database stores system configuration, historical data, the set of possible critical states, and network reference and dynamic model. Having a set of critical states defined will allow software running in the command center to react if anything matches those descriptions. They stress the importance of time stamps on events and collected data.

They composed a table that lists NERC CIP reliability standards. With their architecture a topology is also generated that gives a visual image of the status of the devices existing in the system. They have a table that maps compliance between their architecture and NERC CIP standards. They also have a table that lists situational awareness requirements that they compiled based on Barford et al. as a source [4], as shown in figure 1.

The following part of this section will look more in detail at ways to do correlation and context management.

Vaarandi et al. wrote a rule based correlator in Perl that could run on several platforms named Simple Event Correlator (SEC), that other systems later would be built upon [38]. The rules SEC follow are determined by contexts and can be turned on or shut off by Boolean expressions at runtime. He also describes that SEC can associate events by context.

Morin et al. presented a formal model for security information representation and correlation called M2D2 [39]. It was able to utilize information about vulnerabilities and events. M2D2 takes the characteristics of the monitored information system into consideration and uses information

	Requirement	Description
1	Situation perception	Be aware of the current situation. Situation recognition and identification.
2	Impact assessment	Be aware of the impact of the attack. Vulnerability analysis.
3	Situation tracking	Be aware of how situations evolve.
4	Trend and intent analysis	Be aware of actor (adversary) behaviour.
5	Causality analysis	Be aware of why and how the current situation is caused.
6	Quality assessment	Be aware of the quality of the collected situation awareness information items.
7	Future assessment	Assess plausible futures of the current situation

Table 1: Mavridou et al. table for situational awareness requirements [3] based on Barford et al. [4].

about what security tools are being used in the monitoring process. They give three examples of correlation where the first one aggregates alarms referring to the same host, the second example identifies which host is vulnerable to the known vulnerabilities, and the last aims at reducing the number of false positive alarms. In order to map vulnerabilities to hosts, it is important to know what exists on that host, and to do that they defined a host configuration. A host system configuration consists of software such as OS, applications and services. This software was identified as products which is defined with product, version and vendor ID as well as product type. They wrote that vulnerabilities and product tables came from the ICAT database, which is today called the National Vulnerability Database (NVD) [40]. Users and files are concepts that was not added prior to the authors publication. They describe that M2D2 models alerts, scans, as well as IP, TCP, UPD, HTTP and HTTP log events but not OS level events. HTTP log was the HTTP server log. The alerts has a report name and contains alert generator information that identifies the IDS and information that describes what caused the alarm. The description of what caused the alarm could be either empty if that information is not available or it can be filled with several causes. The alarms are generated from events. A causal event was defined as an event causing an alarm, and that a single event can cause several alarms. Alarms that have at least one common causal event can be aggregated together. They also propose aggregating alerts based on vulnerability.

Chen et al. used support vector machine (SVM) in masquerade detection and used co-occurrence matrix to model users behaviours [41]. They also tested it with online update which allows the system to learn changes in users behaviour over time. A summary of previous approaches was provided including the false positive and hit rate percentages, sorted in a table.

Wei proposed a clustering algorithm for event correlation that looks at attribute similarity and an algorithm that sorts events according to their severity [42]. He describes that the correlation analysis is done in three parts, starting with formatting the security information, then matching of correlation rules and lastly the generation of security events. In order to sort events on severity, he takes the probability of successful attack, importance of target and priority of security event into consideration.

Sadighian et al. proposed an alert correlation framework called ONTIDS which is ontology-

based and context aware [43]. It works in four steps starting with collecting alerts, the second step is normalizing the data into a unified format and integrating context, third is populating context ontology's, and the last step is correlation. They wrote rules for filtering and correlation using Semantic Web Rule Language (SWRL)[1] and Semantic Query-enhance Web Rule Language (SQWRL) [44]. OWL-DL[2] was used to create the ontology's. In addition to using IDS alerts they also gathered context from sources such as configuration of networks and hosts, vulnerabilities and asset criticality, and other information that can provide context. The context ontology is split up into static and dynamic, where static can be OS, network architecture and other non-changing information, and dynamic those that do change such as time of day and system usage. Vulnerabilities are collected from vulnerability databases such as CVE and NVE. Common Attack Pattern Enumeration and Classification (CAPEC)[3] or expert knowledge was suggested as sources for attack scenarios and models. Examples of rules and their implementation was provided in their publication as well as two case studies.

Wang et al. proposed a correlation system using knowledge graphs named Knowledge Graph Based Intelligent Alert Correlation Framework (KGBIAC) [45]. The knowledge graph allows related information to be linked with edges, such that there is collections of entities and their relations. They describe that the information is stored in Resource Description Framework (RDF)[4] format. KGBIAC uses several knowledge bases, one for network infrastructure knowledge, vulnerability knowledge, cyber threat knowledge and alert knowledge. The first one contains all system information. Then, their knowledge graph correlator normalizes and extracts information, applies fusion, verifies alerts and correlates attack threats. Normalization and extraction is done by regular expression. They describe that alerts of the same event recorded within the same time slot and that contains the same attributes are fused before the verification stage filters all false alerts. Defining an alert as false is made possible by their knowledge graph since each host has a set of possible known vulnerabilities associated with it, and an alert against a vulnerability not existing in that set must therefore be false.

Cheng et al. created an ISMS to help telecom operators maintain a secure working state [46]. Therefore their research focus on helping telecom operators with securing their systems and network devices. The ISMS was built up by three parts starting with device log collection, then correlation analysis followed up by alerting of personnel both visually and over the phone or with email. Anomalies was ranked based on severity and attack types was ordered into classes. The classes are Device Access and Log in, Console Command Operation and Configuration Modification. Example from the first class are port scanning or unauthorized login attempt, from the second class is device log deletion and from the third an example is modification of NTP or ACL configurations. Their system uses both manual and automatic notifications, where manual refers to when personnel discovers a possible security event and reports it. The automatic notification system works on

[1] https://www.w3.org/Submission/2004/SUBM-SWRL-20040521/
[2] https://www.w3.org/TR/2012/REC-owl2-overview-20121211/
[3] https://capec.mitre.org/
[4] https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

rule based correlation. As mentioned they use network device log analysis, where the log server contains modules that handles normalization, filtering, search, analysis and more. The rules was described as being based on different domain knowledges.

Husák et al. surveyed literature on alert correlation and attack prediction using data mining and summarized it in their publication, and conducted experiments on sequential pattern and alert mining algorithms using a real data set containing sixteen million alerts [47]. The related work review was in addition to being described also summarized in table format listing authors, use case, approach, algorithm and what data set it was evaluated on. They used the SPMF library[5] to select candidate methods which is a large tree structure guiding the reader down the tree based on questions asked. Description of the seven sequential pattern mining and the three sequential rule mining methods they selected was provided for the readers. They scripted the processing of data sets and database creation, the running of algorithms and result formatting using Python, which were made available by the authors on GitHub [6]. They used, the open-source data mining mining library (SPMF)[7], and alarms was formatted in Intrusion Detection Extensible Alert (IDEA) format [48]. They had two evaluation criterion's, first was performance of the sequence mining approaches, and the second one was the usability and comprehensibility of output, however, many more findings not mentioned here can be read in their publication. They state that sequential rule mining works for alert correlation use cases and that sequential rule mining works for attack prediction use cases.

Lu et al. created an algorithm called Purpose-oriented Maximum Attack Sequence Patterns (PMASP) that was designed to generate attack sequences from clustered alarms [49]. Before the PMASP algorithm can be applied, they eliminate redundant alarms, and then cluster alarms into categories corresponding to classifications of attacks. This is when PMASP can be used to find frequent sequence sets. An attack sequence is steps an attacker does to reach their goal. A four part overview of their method starts with preparation followed up by initial attack sequence generation, frequent sequence set mining, and threat rules construction. Their alarm format contain four pieces of information starting with alarm number called sid, source IP, destination IP and a time stamp. Alarms originating within a specified time slot that have the same sid, source and destination IP are reduced to one alarm. They use the following four attack classifications; buffer overflow, DoS, web attack or as other attack.

Kenaza et al. propose an ontology based on OWL[8] called ONTO-SIEM and implemented it using Protégé [50, 51]. It consists of a conversion module that feeds alarms and contextual information into the ontology and a correlation module. They used Nmap[9] to obtain hosts and network topology, but writes that it is possible for operators to manually add information directly into their ontology. Having this manual ability is as discussed in this thesis very important when working with OT.

[5] https://www.philippe-fournier-viger.com/spmf/map_algorithms_spmf_data_mining097.png
[6] https://github.com/CSIRT-MU/SecAlertSeqMining
[7] https://www.philippe-fournier-viger.com/spmf/
[8] https://www.w3.org/TR/2012/REC-owl2-overview-20121211/
[9] https://nmap.org/

Nessus[10] was used to discover vulnerabilities and Snort[11] was used to discover attacks. Pellet was used as an OWL reasoner for their ontology. Events was run through a set of rules to decide if the event is relevant to consider or not. Each rule are displayed and described in their publication. Events was aggregated into meta-events describing a malicious activity and distinguished by host and network meta-events.

3.4 Work related to normalization languages and external communication

In order to share information on vulnerabilities, incidents and the actors involved a language is needed that can support this process in a structured and agreed upon manner. There are a wast amount of projects that has attempted to solve these problems, and two that could be useful for the purposes needed in this thesis are described here, while a third protocol and system called CYBEX will be discussed later in this chapter. These are STIX threat information expression language and the TAXII protocol, and VERIS event recording and incident sharing language.

It should be possible that computer systems can generate and read them, as well as they should be human readable. For the purpose of the architecture in this thesis it is important to store vulnerability data from vulnerability analyses and penetration tests in a local database, share CTI information with external partners and community in a safe manner, and to have a language that can provide descriptions to be used in the visualization phase of the architecture.

Beginning with STIX, which stands for Structured Threat Information Expression, is a well known language for sharing cyber threat intelligence (CTI) [52, 53, 54] and their documentation is divided into five main parts [54]. It is a markup language based on JSON[12] that allows the security analysts to describe CTI in a formalized manner. STIX uses Domain Objects (SDO) and Relationship Objects (SRO). The available SDOs are Attack pattern, Campaign, Course of action, Identity, Indicator, Intrusion set, Malware, Observed data, Report, Threat actor, Tool and Vulnerability. While the available SROs are Relationship and Sighting. The Relationship Object describes how two SDOs relate and the Sighting Object can be an indicator, malware or other CTI sighting.

STIX contains a language called STIX Patterning which is a proprietary correlation language [55]. IT can be as simple as doing comparisons on an IP address. The pattern is built up by one or more Observational Expressions. Each of these expressions is one or more Comparison Expressions and may contain Observation Operators and a Qualifier. The Comparison Expression defines a filter by for example having a simple IP address comparison. An Observation Expression may contain more than one Comparison Expression for example connected with a Boolean *OR*. Several Observational Expressions may be combined with Observational Operators such as *FOLLOWEDBY*. The pattern may end with a Qualifier such as *WITHIN* <*time interval*>. STIX is very object oriented, and can be summarized by saying that the JSON files consists of a series of objects. There are objects for malware, threat actors, objects defining identity, indicators, associations and objects that helps with controlling the use of indicators. The last mentioned is Granular Markings and Marking Definitions.

[10] https://www.tenable.com/products/nessus/nessus-professional
[11] https://www.snort.org/
[12] https://json.org/

Granular Markings enables STIX TLP use, which is a color description added to explain the degree to which an attribute is sensitive [56, 57, 58]. The colors range from Red, Amber, Green and White, where Red is most sensitive and white is the least sensitive and can be freely shared. The Marking Definitions allows copyrights to be mentioned in the objects.

After STIX it is natural to move over to TAXII which is an application layer protocol for exchanging CTI and is made to work with STIX [59]. TAXII stands for trusted automated exchange of intelligence information and works over HTTPS. It has clients and servers, where the clients are either publishers or subscribers. This means that the clients share CTI info with each others through TAXII Servers. The servers are therefore called Channels, as they distribute the information to all the subscribers clients.

Then there is VERIS, a language used when describing and sharing incident information which stands for vocabulary for event recording and incident sharing [60]. It too is based on JSON and has a database on GitHub[13] where the community can browse and publish incident information. VERIS uses four high level categories they call the four A's, which are Actor, Action, Asset and Attribute, where attribute can be an assets confidentiality level [61]. VERIS is very modular and has only a few circumstantial required variables, as given in their example, if hacking was present, the *hacking variety* is required [62]. Other required information is if the incident was an actual security event and description on how it was discovered. When writing this, their issues database has more than one thousand three hundred issues listed and the folder of validated files[14] contains more than seven thousand JSON files. VERIS also focuses on impact assessment and consequence measuring [63].

The difference between STIX and VERIS is that VERIS focuses on incidents that have happened and tries to document these where STIX describes for example threat actors, attack sets, campaigns and organizations sightings of these while drawing associations between those objects.

Rutkowski et al. publised an editorial in 2010 about the CYBEX project, that stands for The Cybersecurity Information Ex-change Framework [64]. There they describe the structure of the CYBEX protocol and the massive amount of standards that was introduced to support it. It is built up by five functional blocks. These blocks are as follows, information description, discovery, query, assurance and transport.

Vakilinia et al. proposed a system that enabled organizations to share CTI using CYBEX while remaining anonymous, as well as protected from organizations sharing corrupt or malicious information [65]. The organizations that wants to participate has to register to their system. Should any organization misbehave by sharing malicious content, then a dispute stage enables the system to ban the violating organization. In addition to anonymize the creators of CTI, it also anonymize information inside the CTI that might reveal the creators. To prevent leeching, they describe that it has a reward system for those that participate with CTI. The reward system works by giving tickets to those that contribute, which can be verified by their server for rewards. They proposed an ag-

[13] https://github.com/vz-risk/VCDB/issues
[14] https://github.com/vz-risk/VCDB/tree/master/data/json/validated

gregatable blind signature to keep the organizations anonymous when receiving tickets and getting rewards.

One year after Vakilina et al. proposed system, Sadique et al. made an automated system with focus on privacy, that can collect raw data from many sources in different formats and convert it to STIX information [2]. This is a CYBEX-P project [66] where the P stands for Privacy. They have four levels of data classification, starting at zero for least sensitive and level three as the most sensitive. Even though the system is real-time, it is also able to process previously collected data. The system consists of three modules, which are the collector, classifier and conversion module. They describe that the Data Classification module takes into account which format the receiver of the STIX information needs, which can depend on software and operating system. They made their Python parsing scripts available on GitHub[15]. The system supports horizontal scaling and distributed computing.

Sadique et al. published later a new paper where allot of additional work has been put into CYBEX-P [67]. They also they discuss some of the difficulties of sharing CTI where one example is that competitors may acquire information about underlying intelligence.

3.5 Work related to visualization of information security data

This related work section is related to the visualization part of the CSSA architecture. This section will focus on displaying security relevant information on screens for operators. There exists allot of research and technology on this topic. Therefore, collection of different ways this can be done will be presented.

Amico et al. produced several visualization *scenes* to present information security events as part of a DARPA project [68]. The system they made could display mission information, resources and assets they depend on, geographic location, network protocols and more in a 3D view. This data was fetched from a relational database, however, updates to this database would not synchronize to the view unless a user manually updated. The *scene* could utilize many different visual effects to rely information to the operators. Examples are using different geometrical objects such as spheres and squares, colorization's to describe severity of alarms and fading to represent time since alarm was made. Having 3D view gave them many opportunities, for example in one of their scenes it allowed them to represent layers from mission on the top, the tasks they consist of, the type of assets and their location on the bottom.

Yin et al. wrote a program that displayed network flows in order to provide information security situational awareness [69]. It used parallel axis view with three axes. The left axe represented external hosts sending information to internal hosts, where the external hosts was represented by the middle axes. The right axis was a symmetric representation of the left axis, and connections between the middle and the right axis represented information leaving an internal host. For an IP address to appear in the visualization, that is along one of the axis, it had to send or receive enough

[15] ttps://github.com/qclassified/cici

data to cross a defined threshold. Hosts that went below the threshold would be automatically cleared from the visualization. In order to prevent the screen to contain too much information, users could set a limit on the amount of hosts shown, and those with the highest amount of traffic would be selected by the program until the user defined limit was reached. Users could also sort on ports, and colors would then be used to distinguish the ports when drawing the lines in the view. The width of a line would represent the amount of data in the network flow.

Abdullah et al. developed a system that could fit two and a half class B IP address ranges with color coded alarms in one screen [70]. Their system uses logs generated from IDS's. In order to fit so much information into one screen, each IP address is represented by one pixel. The Y axis represents the addresses and the X axis represents time. The addresses are partitioned over eight columns where twenty addresses are represented by one pixel row, and the X axis twenty four hour intervals are repeated for each column. The pixels are normally black, but are lighted up with a color when an alarm is registered on it. They provided a zoom ability where the operator can select a single twenty four hour column, or zoom again and view twelve hours of data. During a zoom, time is represented on the top and the right column provides external IP addresses. Arrows connect internal addresses with corresponding external ones when applicable, depending on alarm. It also supports mouse over for details on alarm, and filtering on alarm severity. It is able to store images so that the operators can compare these images with images of normal activity. They describe is easier to compare the images than text based logs for large data sets.

Abdullah et al. also created a tool that could present traffic over ports visualized using histograms [71]. The Y axis represents packet count, but can also display packet size, while the X axis represents time. Important ports are shown in small ranges, while those not so important are displayed in larger ranges. They selected port 0 to 1023 as important. The operator can interact to display details about each range. The pillars will then contain the total amount of ports used for each time interval with the most active port range visually taking the largest portion of the pillar.

Koike et al. created a system that could generate logs from sensors, and process them all the way up to a visualization program they had created [72]. It used two 2D matrices, one to represent global IP addresses and another for local addresses. For the global address matrix, the Y axis represents first eight bits of an address while the X axis represents the following eight bits. It therefore covers the first two octets. This means that the matrix for local addresses covers the third and fourth octet. They selected nine points of information to use in their visualization, namely date, ID of network sensor source and destination IP and port number and alert frequency, name and ID. The operators can see both matrices side by side where each pixel in the matrix represents an IP, and each axis has a histogram display of relative number of attacks. Further they describe that they divided the two and a half thousand Snort alerts into eight groups that is represented by a color, such that each pixel will be colored during an attack, and a sector in either matrix can obtain a background color representing the most frequent attack. The operators will also see a rectangle below the matrices displaying IDS alarms textually. The visualization is interactive in the way that operators can select update intervals, rewind, fast forward and click a pixel for details. They suggest that some patterns can be predicted, and described two worms that used local and random IP

scanning, making it very likely that local addresses would quickly experience attack attempts.

Goodall et al. created a visualization system that utilized TCPdump data [73]. Its X axis displays user defined time intervals represented by the table columns, and can be compressed towards the edge of the screen to allow more space for those in center view. The Y axis is placed on the right hand side of the screen and displays IP addresses. The rows are hosts, which get a color code depending on amount of data communicated by that host in the given time interval. Coloring is also used to define if host is local, and colored lines to define connections. To obtain these lines, the operator had to expand a column into two new ones where the links could be displayed between them. This visualization therefore makes it appear as two new columns are created, but the middle column are empty, and is where the connection lines are drawn. They also implemented the ability to hover over a cell, described earlier as a host, the operators can obtain information about date and time, IP address, number of packets and ports the host is communicating over. Another table is displayed at the bottom of the screen that gives the same information as the hover over detail box, but in addition also offer protocol name. They conducted user tests to obtain feedback on the visualization.

Pearlman et al. created a system that could visualize services running on hosts in a network [74]. A differentiation was made between managed and unmanaged hosts by providing a more detailed view on the managed hosts. Managed hosts are hosts that are controlled by those who operate inside the network, and the unmanaged hosts can be hosts on the Internet. Each host which is a node in the visualization was represented with a pie chart where the sectors was divided into each service running. The size of the sectors in the pie chart visually represents the usage of that specific service represented by that sector. This means that if a service was used more than others on the node, it would have its sector on the pie chart grow to visually represent that. The size of the node represents the amount of activity. Colors was used in the representation. The system supports most of the features previously seen, such as details from hovering over with mouse. The nodes are placed in the center of the 2D plane and are moved according to the activities occurring inside them, and links are drawn between nodes that communicate. The operators have the ability to move the nodes manually. Their system allows the operators to define a model network for normalcy comparisons which worked as a comparison model to the real activity monitored. They have a numerical representation of anomaly ranging between zero to one where a higher number represents a larger gap between the actual network activity and the model. Rules to define abnormal behaviour also exists, as they described that some activity would change the score more than other. Some activity that was used as examples that would impact the anomaly factor was ICMP activity from machines not designated to send such activity, as well as user activity to websites with abnormal suffixes. Normal activity would slightly lower the factor back again. Nodes that had low anomaly factor would fade in the visualization. Their system allows new pie charts to grow outwards from old ones, giving an onion like structure where each circle inwards represents the activity of a historic time interval.

In 2007 Mukosaka et al. created a system that could create 3D visualizations of malicious activity happening on a large campus [75]. The visualizations gave geographic location of the involved

campus computers, similar to Amico et al. described earlier, but in this case it is represented as a map. The left wall is a time-diagram and the right wall is an IP matrix. They have a pillar at the very right end of the visualization which is a list of existing system ports. Lines are drawn between IP matrix and map during an attack. Their visualization system follows the trend with implementing more and more interactivity and also supports change of viewpoint with animated viewpoint.

Several ways to display information on a screen has been touched upon already, and there is a was amount of more literature on the topic. One example is Makanju et al. who created a visualization system to represents syslogs visually [76]. It was inspired by Shneiderman who back in 1992 used it to map files on a hard disk [77]. It displays a mesh where nodes are colored green, with a darker shade depending on the severity of the log event. In order to view logs from specific days, they implemented a slider that can be used to retrospectively visualize one month backwards in time. Users can do searches and when a result is found, those specific nodes will be colored red. The result is therefore an interactive green mesh with zoom and panning.

Bond proved the positive effect of creating visuals out of large amounts of firewall log data in a GIAC/SANS publication in 2009 [78]. He described the use of two tools called AfterGlow[16] and GraphViz[17]. He wrote a parser named ciscofw2csv.pl for the CISCO firewall logs. AfterGlow was used to create graph description language files containing source and destination IP, and a destination port field that included protocol, that GraphViz could create visualizations out of. This enabled the creation of maps over inbound and outbound blocked connections which is allot easier for humans to read than the firewall logs they where based on. In the case of outbound communication being blocked by the firewall, it is noted that this is of special interest to the operators, and the visualization of such occurrences will help point these events out from the large amount of logs.

Fink et al. conducted tests on cyber analyst to determine the usefulness of large screen areas when doing incident handling and analysis [79]. They created an eight screen setup with a total of 33 megapixel that the cyber analysis where to spend two hours analyzing incident data. They observed the analysts during the process and collected feedback not only related to the setup but also towards the analysis process as a whole. They also invited an ergo dynamics expert to provide her feedback on the large set of displays. From this process they obtained allot of important information including the benefits of the large available screen space, the importance of documenting steps conducted during an analysis, the importance of available communication channels between analysts while working and the possible eye strain from having that many screen. Regarding the last mentioned, they suggested having a black background and low screen lighting. The knowledge and feedback that was collected was used to create new work space prototypes and four design principles. They then collected feedback on the work space prototypes from the analysts that participated in the previous test. One of the prototypes displays a large black wall with images of user interfaces to programs the analyst have used while conducting an analysis where arrows displays the chronological order they where taken. The points of interest that the analyst found especially important was displayed as larger images. The prototype is meant to give the analyst a historical

[16] http://afterglow.sourceforge.net/about.html
[17] http://graphviz.org/about/

review and display the work flow during the analysis. This prototype was called History trees. Another in their set of prototypes allowed the users to have several cases in the screen space at the same time.

There also exists research on the processes itself of creating visualization tools. One of these works is by Langton et al. where they describe evaluation methods for information security visualization and provides advice on usage of the methods, steps an analyst must take and questions he or she must ask. They also provide a summary of research done on the topic [80]. They also describe that when creating visualization tools, they should be able to deal with large amounts of data and data origination from many different sources. There exists also allot of different types of data, ways to process them and ways that would best present those data to the operators. They also describe methods and advice for conducting user evaluations of cyber security visualization tools.

There is also requirements on skills and knowledge of the personnel conducting analyses and incident handling. Balarishnan describes three different domains of expertise needed [81]. These are hacking skills, math and statistics knowledge and substantive expertise. Last is explained as the security domain knowledge. He then goes on to describe the process of creating visualizations, starting with a five step circular process. The process begins with visualizing goals, that leads to data preparation, exploration, visualization and feedback. The feedback leads back to the first step again where the feedback can be used to improve the process. He also describes that the visualization process should be goal and case driven. The importance of asking the right questions, cleansing the data and normalizing the data formats is part of the data preparation stage. The explore stage contains a series of ten possible steps to work with the data. In the visualization phase he explains the two aspects of visualization theory, aesthetics and understanding of different available methods, and shows ways to select different tools such as graphs and charts. Time estimates for each step is provided. In the paper he also addresses the use and benefits of colors as we have seen used allot by the research described above.

4 Industrial Control System Overview

This chapter will give an overview of a SCADA reference architecture, industrial control systems and communication protocols.

4.1 Reference Architecture

Figure 1 represents the architecture used in both master thesises from kraftCERT, and is therefore a cooperation between me and Kari Anette Sand [13]. It is heavily based on ENISA [1]. It is divided into four levels where level one is the production and control process level. That layer consists of actuators 4.2.7 and monitor units (MU), and communication is channeled over IEC 61850 process bus [82]. The next layer is the supervision and monitoring layer, in there the controller LAN starts that contains PLCs and RTUs described in section 4.2.1 and 4.2.2. These devices communicate over a station bus where they can talk with SCADA servers 4.2.4 or other SCADA systems 4.2.3. Within the same level, but above the Controller LAN there is the Supervisory LAN where local HMI's 4.2.6 and a modem to communicate with the rest of the organization exists. The third level starts with the Operation DMZ. In this DMZ there may exist a series of servers that hosts applications, SCADA systems, Domain Controllers, Engineering stations and Historians 4.2.8. Above the Operation DMZ, but still inside level three there is the Internet DMZ and the Enterprise LAN, with potentially a large set of servers as well. Such servers can be web and mail servers that exists inside the demilitarized zone and authentication and business servers that exists within the Enterprise LAN. Descriptions of devices and protocols are provided in the following section 4.2.

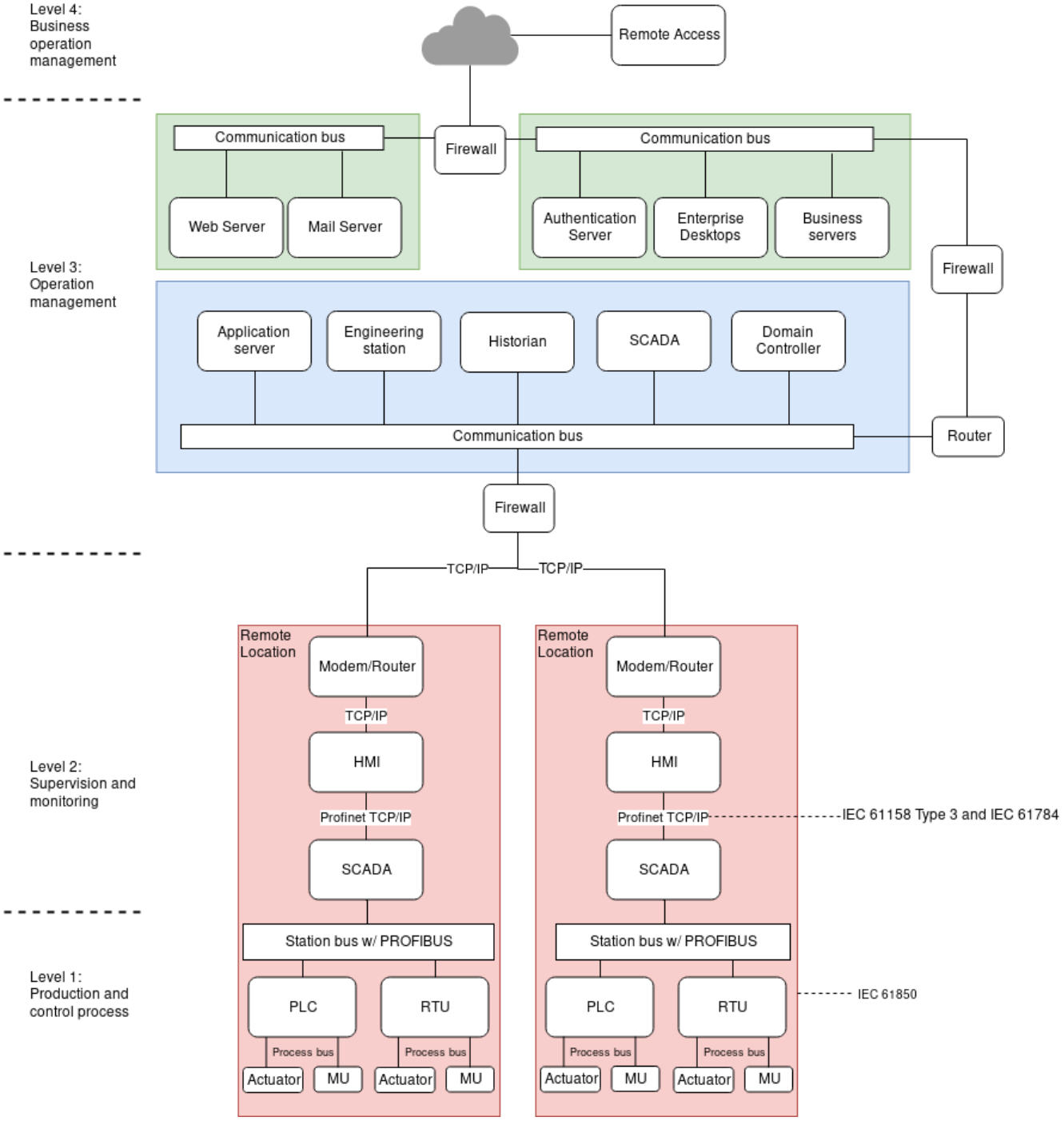

Figure 1: SCADA architecture based on ENISA [1]

4.2 Industrial Systems Components

In the following section descriptions of different industrial control systems and communication protocols are provided.

4.2.1 PLC

A programmable logical controller (PLC) is a controller that are used to automate manufacturing processes and therefore often used in industrial systems [83]. They are physically hardened in order to survive in many different and harsh environments. The PLCs has a solid state and allows the operators to program them to fit to the task at hand [20]. These controllers may run on ladder logic and have strict requirements on timeliness, since many industrial systems may require very time efficient responses. PLCs are programmed according to the IEC-61131-3 standard [83, 84].

4.2.2 RTU

A remote terminal unit (RTU) is a device that shares allot of its capabilities with a PLC, is environmentally hardened enough to be placed outdoors and can perform remote communication [83, 20]. A PLC can be seen as an RTU when it is given wireless communication ability [20].

4.2.3 SCADA

SCADA can refer to any SCADA systems, such as PLCs and IEDs. When used in an architecture the collection of devices in the labelled block depends on the needs of the specific tasks being run.

4.2.4 SCADA Server

A SCADA server is is one or more machines that works as servers between SCADA systems such as PLC's and RTU's and the other systems in use such as stand alone HMI's. The server acts as a master to the other devices [20].

4.2.5 IED

An intelligent electronic device (IED) is a device that can communicate with and control external devices [20]. Since an IED may operate in harsh environments they are often designed to be very hardened. When it comes to the difference between an IED and the two previously mentioned PLC and RTU is that it is often designed to do one specific function [83].

4.2.6 HMI

A human machine interface (HMI) works as a user interface with a screen and a physical input that operators can use to interact with machines. Such machines cab be the SCADA RTU's, PLC's, IED's or other machines operators would need to interact with. The HMI's can be either software based, embedded or a separate monitoring device [20].

4.2.7 Actuator

An actuator is the mechanism that can be used to move or control a system [20].

4.2.8 Historian

A historian is a centralized database [20] that can store large amounts of industrial data as well as alarms and events [85] from the machines and SCADA systems running in the plant.

4.3 Industrial Network Protocols

A handful of network protocols used in ICS are described below.

4.3.1 Profibus/PROFINET

PROFINET is a scalable fieldbus communication protocol, is described in IEC 61158 and IEC 61784 International Electrotechnical Commission standards, and is based on an open standard [83, 86, 87, 88].

4.3.2 Ethernet POWERLINK

The Ethernet POWERLINK has a master node to use cyclic polling of slaves using control messages over encapsulated Ethernet frames. The first period are *start of cycle*, second period is a period of waiting for the slaves to respond, and the third phase is an asynchronous period for communication of non critical data, which is explained to be non-critical [83].

4.3.3 DNP3

DNP3 is a secure variant of DNP [83]. The DNP user group writes that it provides communication between master stations and substations [89]. The DNP user group[1] is a non profit user group organization that works with maintaining and promoting the protocol. In electric substations Knapp et. al writes that an RTU (4.2.2) tend to be the master, and an IED (4.2.5) tends to be the slave [83]. DNP uses cyclic redundancy checks, time stamping and supports many standardized data formats, and DNP3 adds authentication [83].

4.3.4 Modbus

Modicon Communication Bus (Modbus) operates at OSI layer 7, is a request and response protocol that can allow up to 32 devices on one link [83]. Each command sent over Modbus will be received by all devices connected to the link but only the receiver with the matching address will respond. There is an organization called The Modbus Organization[2] that works on driving the evolution of the protocol it self and architectures for automated systems as well as trying to help simplifying the implementation of those who wants to utilize the protocol.

[1] https://www.dnp.org/About/DNP-Users-Group
[2] http://www.modbus.org/about_us.php

5 Attack Scenarios

The attack scenarios was written in cooperation with Kari Anette Sand who also works on a masters project from KraftCERT [13]. However, during the time each project has been on going, the attack scenarios have changed in pace with the work on each project.

The scenarios describes attacks spread over the different steps the attackers conduct, and suggested defensive responses. This is then followed up by additional description of the expected benefits of using the situational awareness architecture and how it should be of aid.

5.1 Attack Scenario 1

The first attack scenario involves an adversary that have managed to take control over an Engineering Workstation (EW) in Layer 3 and have successfully managed to privilege escalate by obtaining the user account credentials of an engineer. The adversaries display an ability to access the SCADA server on layer 2 where they proceed with their intention on causing disruption.

The premise of the adversary is to cause disruption in any way possible. In this case, some examples are made of how the adversary can work towards that goal. Disruption in this case is achieved by either shutting down PLCs and RTUs, called SCADA devices in attack defence tree (ADtree), or by terminating or manipulating the communication between those devices and the SCADA server in layer 2. An example of why manipulation of communication data could cause disruption is to manipulate information in such a way that may cause a PLC to not increase or decrease a correct operational level in machinery, causing those machines to not meet the correct demand of their output.

The following are the objectives of both parties:

1. **Attackers objective:** disrupt SCADA device or communication to it, by first attempting to shut device off, second by attempting to harm its availability, then lastly by attempting to attack integrity of its communication
2. **Defenders objective 1:** use network traffic to detect potentially dangerous as well as suspicious commands and report them for further evaluation
3. **Defenders objective 2:** obtain situational awareness based on reports from network traffic

Two figures will be used to illustrate the attack. The first is the ADtree shown in figure 2, which is divided in three parts, the first where the adversary is concerned with causing disruption by shutting down a SCADA device, in our example a PLC or an RTU. In the second part of the ADtree the attackers attempts to cause disruption by shutting down communication with SCADA devices and in the third part of the ADtree they attempt to apply malware in their effort to cause disruption. Figure 3 illustrates the adversary's movement in the reference architecture. The EW is red because

in this attack scenario it is assumed that the adversary has gained remote access over the EW. The adversary will do the following steps;

1. Use tools in the EW to send shutdown command to a PLC or an RTU
2. Terminate communication to SCADA device
    1. Log into the SCADA server in Level 2 using the access level of the engineering account and then terminate the process listening to the specific device
    2. Using software in the EW to change communication parameters to the device so that communication fails
3. Infect the SCADA server at Level 2 with malware that will manipulate communication channels through the server

The main purpose of the SCADA server is to facilitate communication from every PLC and RTU with the rest of the infrastructure. Even though some RTUs is set up with hardware allowing communication with other sites, this attack could still cause damage when the RTU is set to communicate through the SCADA server.

We will use a rectangle for the root node, rectangles with rounded of edges for attacks and hexagons connected with stipulated lines for defence.

5.2 Attack Scenario 2

The second attack scenario involves an adversary that have managed to take control over an Engineering Workstation (EW) in Layer 3 and have successfully managed to privilege escalate by obtaining the user account credentials of an engineer. This attack is considered to be a part of a bigger attack.

Objectives and description:

1. **Attacker objective:** change the set point variable in an RTU without raising alarms
2. **Defender objective1:** use network traffic to detect potentially dangerous as well as suspicious commands and report them for further evaluation
3. **Defender objective2:** obtain situational awareness based on reports from network traffic
4. **Set point definition:** A setpoint is a variable that the RTU uses to measure sensor data towards and is used as a baseline in order to make decisions on the machines it controls. An example of a setpoint can be a number that represents a temperature in Celsius where the RTU is programmed to do certain actions if a measured temperature exceeds or goes below the Setpoint value

Steps in the attack:

The two figures below illustrate the attack scenario. Figure 4 represents the steps with an ADtree where the root node is the attacker's objective, and then the child nodes are divided into two parts of the attack where the first is concerned with modifying the setpoint variable, and the second is

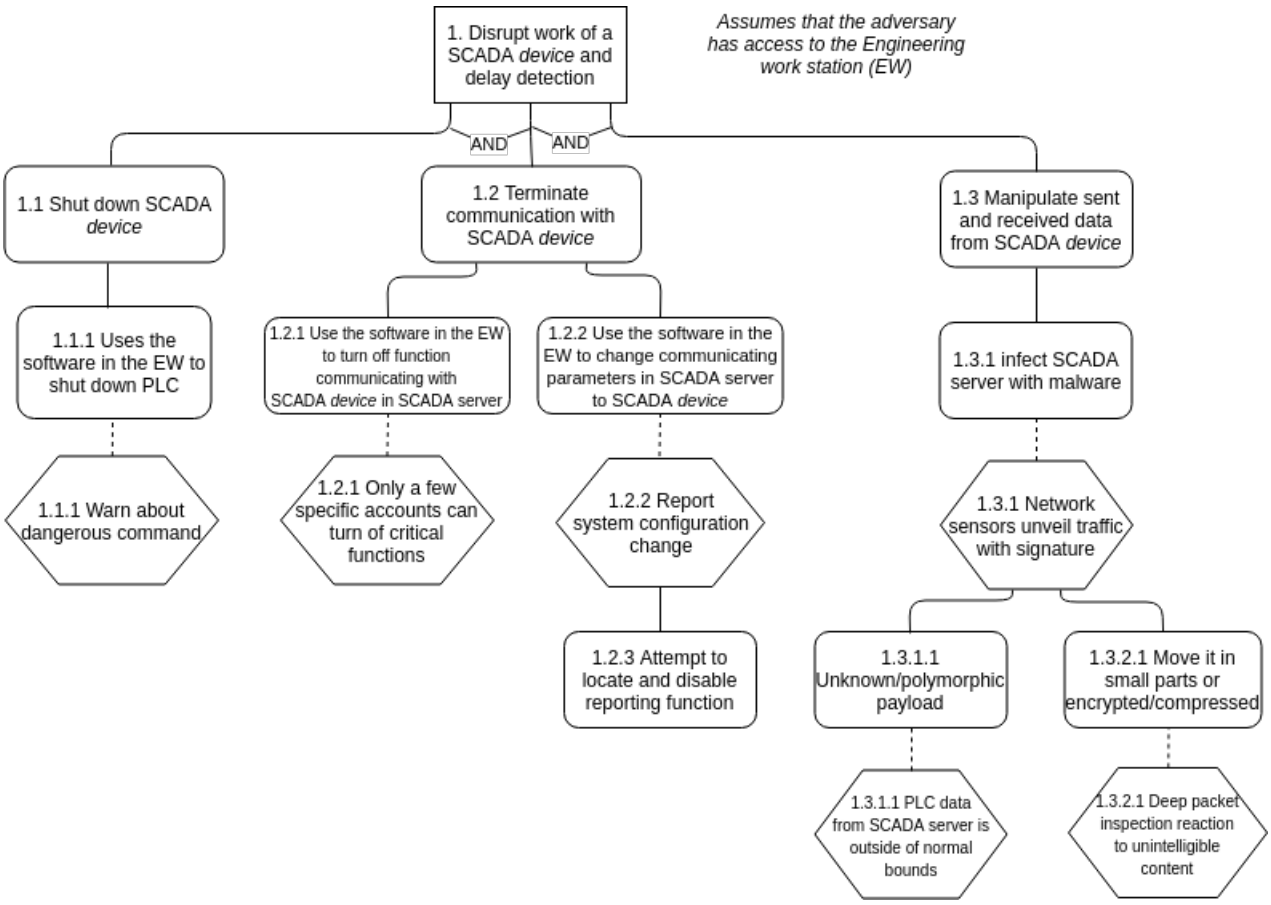

Figure 2: Attack Scenario 1

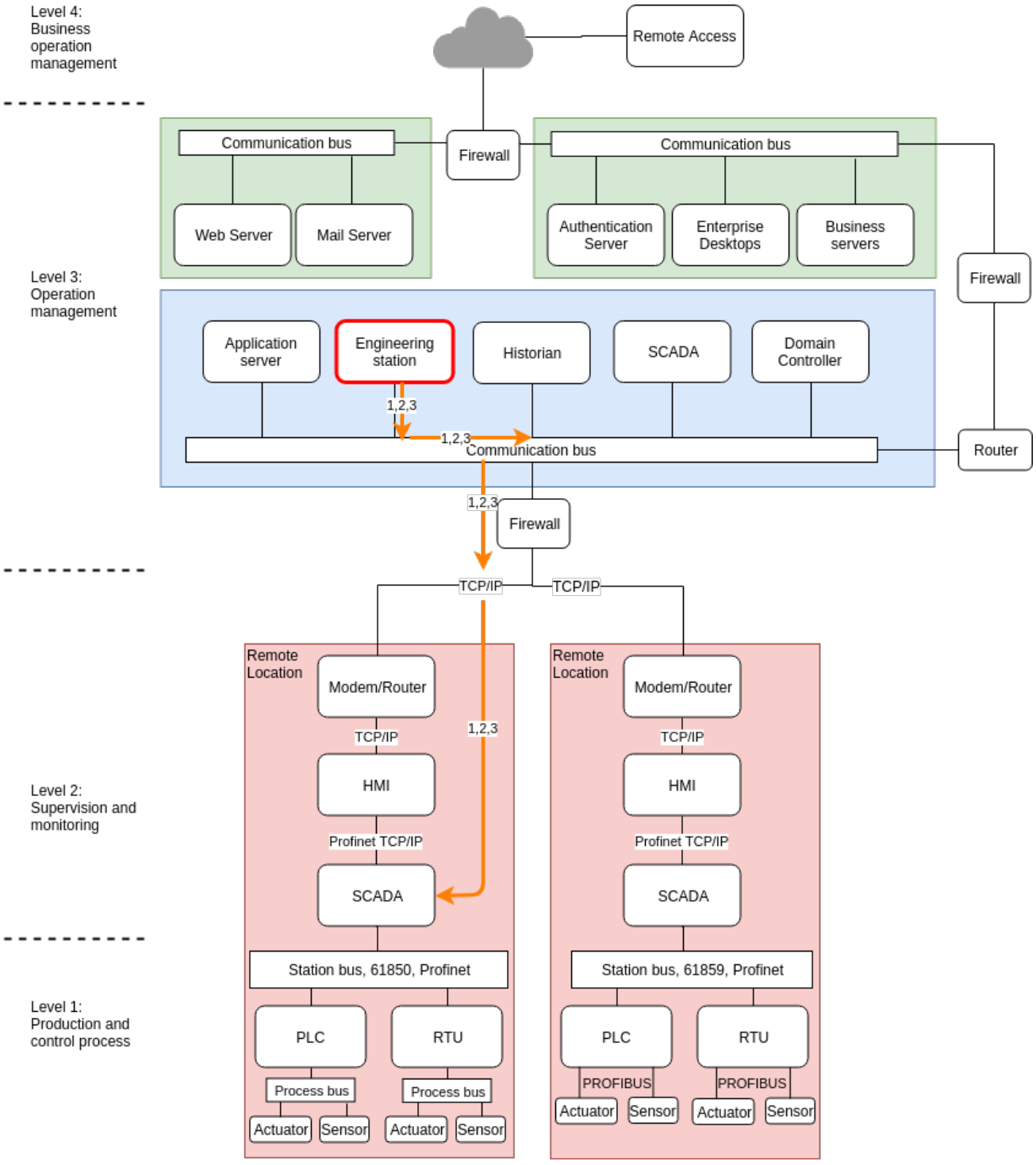

Figure 3: SCADA Architecture, Attack Scenario 1

concerned with preventing detection. Figure 5 illustrates the adversary's movement in the reference architecture. The EW is red in this attack scenario as well as it is assumed that the adversary has gained remote access over the EW. The adversary will do the following;

1. Log into the application server with a dormant account and deactivate alarms designed to log and react to RTU program changes
2. Log into the data historian with an inactive account and delete received processing data
3. The adversary modifies the setpoint variable of the RTU with the software on the EW

As shown in the data collection stage in the situational awareness architecture in chapter 8.1 data from machines, sensors and historians are collected. This gives the basis for information that can be used to defend the systems.

The sensor data reveals if a system such as the PLC has received a shutdown, reboot, halt, alter, or other possible commands relevant to the device type, programming on the device or type of work the device does. Such traffic to the devices will from now on be called suspicious traffic. The situational awareness system will from this point make an effort to determine if the command discovered by the sensors is to be considered suspicious traffic. It is also possible that some situations depending on systems programming and task, that a combination of commands which by themselves is harmless may be malicious when executed in a sequence. The correlation stage discussed in chapter 8.4 correlates the danger of those commands. Especially important aspects to look at will be to check if a shutdown command was sent to a PLC, where it originates from and if a shutdown was scheduled. A scheduled shutdown could be for maintenance, testing, or other purposes. To check for such scheduled events is a pre-processing stage that is part of the data normalization stage in chapter 8.2. In order to be able to obtain situational awareness, it is therefore very important to document such events where actions towards the systems that would be considered malicious are accounted for as legitimate. And such documentation should have its own storage location inside the organization as seen in chapter 8.3. The situational awareness architecture has the following main steps: generate data, normalize data, discover events, fetch external events and publish some events generated internally with external partners, correlate events and alarms, and then create visualizations to the personnel. Last step is the preparation for human use as well as preparing the results to systems that may need it.

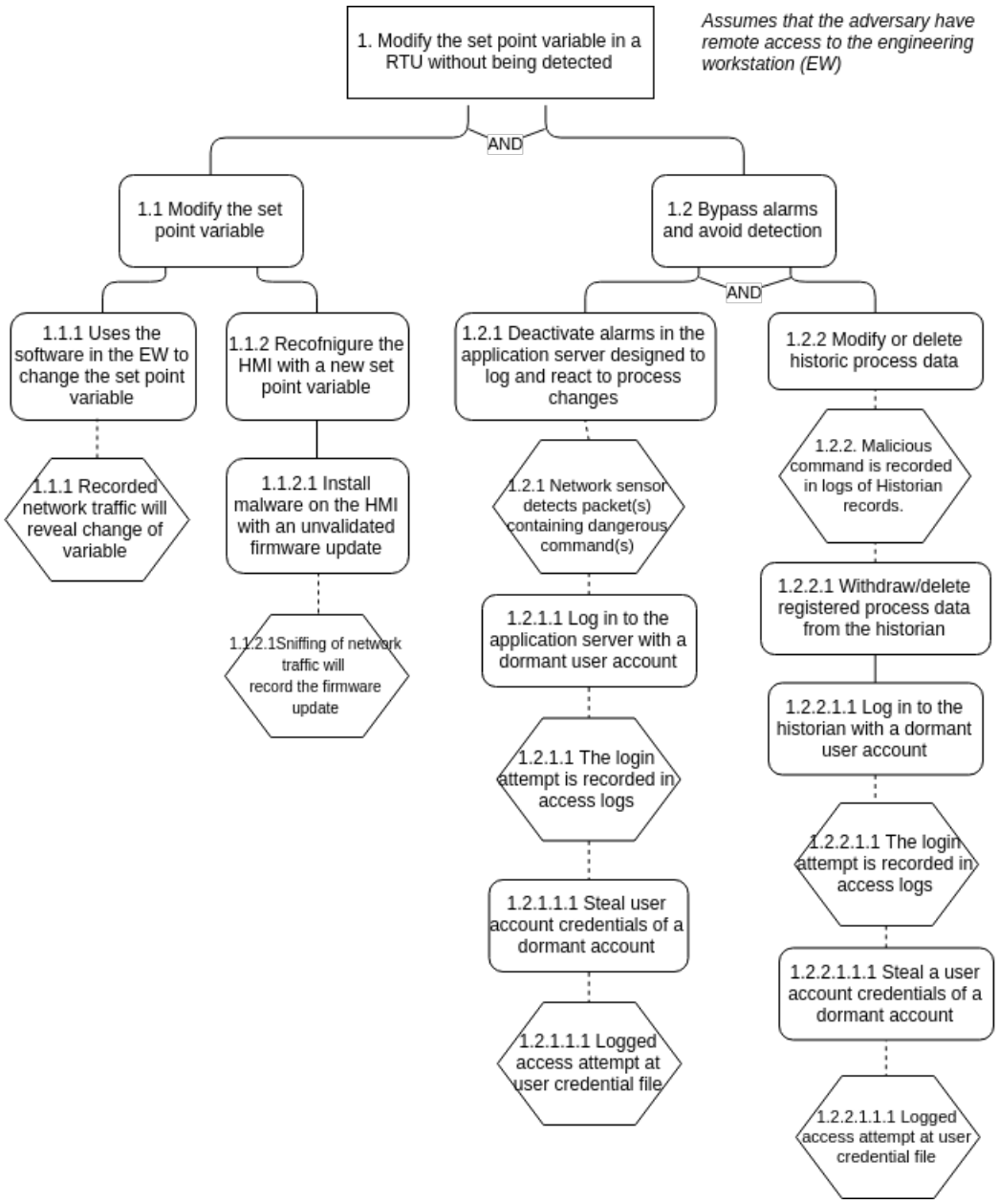

Figure 4: Attack Scenario 2

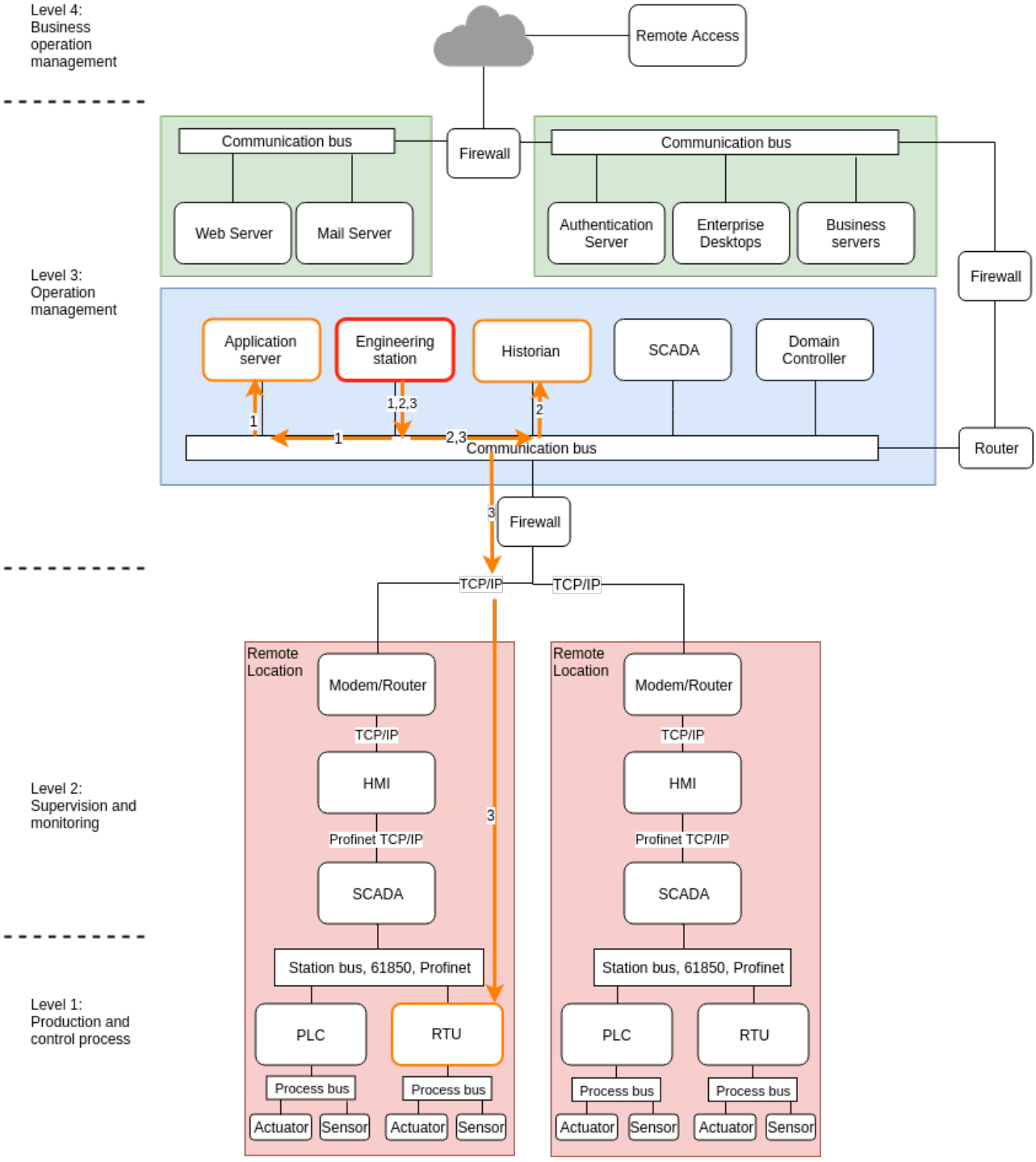

Figure 5: SCADA Architecture, Attack Scenario 2

6 Requirements Analysis for Penetration Testing in Industrial Control Systems

In order to get proper situational awareness it is important to understand our own vulnerabilities and weaknesses. According to Sun Tzu, *If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle*, Art of War, Attack by Stratagem, paragraph 6 [90]. Examination, risk assessment, vulnerability analysis and penetration testing helps towards knowing your self. Lets take a look at three different types of analyses. NIST SP 800-115 defines examination as a review of policies, procedures, security plans, asset inventory and other documentation, including system logs [91]. One benefit to do such an examination would be to gain an insight in how the system is supposed to operate from design, and can help uncover flaws, which are problems with design. A vulnerability analysis is according to SANS the search for weaknesses in the form of known security holes and bugs, insecure settings and misconfigurations [92]. The pentest on the other hand attempts to exploit the vulnerabilities and provide a proof of concept that one or more of them was exploitable, allowing the pentesters deeper access into the systems. However, it is important to note that pentesting in ICS differ from that on regular IT systems because of the resource constraints and high need of availability these systems have. Many of these systems may also not be design to handle unexpected situations caused by the pentesters. This will be discussed more in detail.

6.1 Context in ICS Penetration Testing

It is important to obtain the context of what you are looking before a penetration test, the context of what systems to touch and not to touch, and more. As can be read by Bodungen et al. in chapter four of their book [8], if an organization wants to get the most out of the money invested in penetration testing, then plans on what to test should be created rather than doing a "spray and pray". These plans should be based on risk scenarios from risk analyses so that systems which is critical and very important to protect are included in the test. There is also something called Rules of Engagement [8, 93]. The Rules of Engagement is a written document signed by both parties, one party is those who conduct the pentest and the other party are those who have ordered it. This document contains statements of what is to be tested, and what should be left out. This helps ensure that the pentest is not interrupting the continuous operation of the organization. This is especially important when working with operational technology (OT) such as industry equipment.

Lets discuss this in more detail. An IT system can be large and complex. Parts of the system may talk to other parts of the same system or other systems. Each of these systems are sub-parts of the larger context of the whole production system. Each system can contain a set of IT and

OT components as well as a set of physical machinery such as actuators 4.2.7 that conducts work in the physical world. This may lead to a factory consisting of a large set of systems, each of them dependant on each others in order to create the final product. To make an example of that statement, one can imagine a product being sent from system to system and elements are added to it on its way to be come a finished product. These systems are connected with control points and with servers and HMIs where operators may observe what is happening at the shopfloor and take manual control of some systems if necessary. If a pentest is conducted that puts any of these systems out of order for a brief moment, it could lead to problems with other systems dependant on them.

When doing penetration testing it is very important to follow a structured plan. And to understand and have experience with such plans will help with protecting the organization. In order to get a structured way of viewing an attack scenario, knowledge of the cyber kill chain is important [94]. The cyber kill chain was written by Lockheed Martin and contains 7 steps that an attack goes through. It starts with the attacker doing reconnaissance on the target. This is an information gathering phase which can be both active and passive. First looking at passive in its most strict context means that the attacker is not directly in contact with the victim such as their website or other infrastructure. An attacker in this situation tries to obtain as much information as possible about the victim organization by using OSINT information which stands for open source intelligence. Information that can be found in OSINT could be email addresses of employees which may also reveal naming policies that can be used to discover more email addresses, communications in forum threads made by employees and more. The email addresses can be used for spear phishing and online communications such as forum posts or even public social media communications from Facebook or other sources could reveal inner workings of the organization. An example of this could be that an employee wrote to another employee on a social media account that they where soon to receive some machines from a vendor, revealing vendor and system information. This is harmful because the attacker needs information on the target organizations systems in order to craft attacks. The purpose of this strictly passive phase is that the attacker leaves no traces the victim easily can detect. This is unless in the very hypothetical situation that the victim organization has sharing agreements of information with third parties such as site visitor searches. An example would be that one website experiences many, possibly automated searches on information about another organization on their own websites such as online forums within a small amount of time and warn their partner organization that this is suspicious. When it comes to active reconnaissance it means that the attacker directly is in contact with victims resources such as servers, personnel or other surfaces. The attackers may have seemingly friendly conversations with employees in social media, scrape or copy the victim organization web pages, attempt discovery of services behind internet facing network ports, and much more. Scraping a website means that the attacker runs an automated tool that checks every corner of a website for information they deem useful for their goal. The attacker might also take a copy of the whole site by for example using the httrack[1] tool which is provided in Kali Linux. The purpose of this would be to mine that information locally without making noise. Of

[1] https://www.httrack.com/

course it would make noise if the site was copied immediately or over a short time period, however the attacker may copy it slowly over time, and when a copy is made they may heavily mine their local copy without making any additional noise. When the attacker has collected enough information to start construct an attack, the next stage of the Kill Chain named weaponization will begin. With the information they have gathered from both outside and inside the organization they will craft an attack payload that will help them reach their goal. It is most likely something that can open a backdoor to their command and control (C2) servers but could also be something that damages or encrypts data when inside. The next stage will be to find a way to deliver the malware to its destination, and is the third stage of the Kill Chain named the delivery stage. A common way to deliver it could be through spear phishing where the attackers attempts to make an employee open an email with either a malicious link or file inside of it. A link may try to lead the victim to a website owned by the attacker that attempts to utilize a vulnerability in the browser, for example attempt to cause a drive by download which is a download that was not initiated by the user. The attacker could also have infected a website known very well by the victim such as a third party vendor site, which increases the victims trust towards the phishing attempt. An example where a third party vendor site was attacked causing the vendor to upload infected firmware was in the case of Havex [95]. When an attacker infects a site regularly used by its victims it is often called a water hole attack. However, the attacker could also use an infected file as mentioned. When this is the case the malware often needs the victim to open or run the file. The file would most likely be given a name that appears legitimate for the business that the organization conduct as well as the specific context of work for the targeted employee. The attacker would possibly make the email appear to be from a superior and imply time urgency on the request in order to increase the likelihood that the employee get stressed about opening it before thinking about the possibility that it is malicious, leading into the fourth stage named the exploitation stage. When opened the malware is released, where it most likely calls home for instructions. The malware has at least two possible design goals from this point of time which will lead into the installation stage. The first design goal would use the malware as a dropper that starts to download the actual malware and install it. The other design goal could be that when the attackers receive the call home, they can send updates that when installed modify or add functionality to the malware which will allow it to do new operations. Either way, the attackers gets to introduce their malicious code into the victims network and systems. New backdoors may be introduced in order to allow the attackers to become more resilient in case the victim discover and patch some of them, or should some of them dissapear in a system upgrade where the victim unknowingly kill one of the C2 connections by changing to new equipment and software in some of the systems. The attacker could also patch some vulnerabilities in the victim systems in order to reduce possible future competition from other attackers, as well as increase the victims trust in its own systems should they in the future check for such vulnerabilities after getting knowledge about them from for example common vulnerabilities and exposures (CVE) lists. The last stage of the Kill Chain is then actions on objectives which is where the attackers actively works towards their goals after gaining the foothold on the inside.

The attackers may now use this foothold inside the ICS infrastructure to obtain information

about it and the systems running inside it, as described by Assante and Lee from SANS [95]. They created the ICS Kill Chain which works as an extensions to the Lockheed Martin Kill Chain. It places more focus on the campaign around the attack and adds a second stage for developing and deploying ICS attacks after the attacker reached the foothold mentioned, inside the ICS infrastructure. In the second stage information about the victim ICS infrastructure will be used to develop an attack against it. The attackers will most likely use the information to create a lab that represents the same machines with the same software and software version in a scaled down representation. This allows them to develop and test the attack carefully before launching it. The main phases of the second stage is described as develop and test, followed by deliver, install or modify and then execute the attack.

6.2 Results in ICS Penetration Testing

To begin with, three quick starting points will be expanded. First of all, after obtaining a good context for the ICS pentesting, aiming for the results most relevant for obtaining the best preventative effect is the main goal. Since pentesting will work as preventative work, it is important that the pentest attempts to cover the most critical systems with highest probability of being targets. Considering that a vulnerability assessment might uncover several vulnerabilities, those with highest impact on critical business functions has the highest priority.

The second point is that the results from a penetration test will refer to the data that is obtained from the attempts at exploiting the vulnerabilities. These results can be anything from successful social engineering unveiling the need for more employee security exercising, weaknesses in the architecture or vulnerabilities in software or how its implemented. It very important to explain the steps that the pentester went through in order to successfully exploit, with textual explanations and pictures. This allows the exploit to be recreated, and may help with the development of patches. According to the interview conducted with the utility company, they explained yet another benefit with pentesters documenting what they did while inside the system, helps undoing any changes the pentesters did to the system. The last statement is relevant even though any action the pentesters did was not successful in actually exploiting anything, it may still have caused changes.

The third note is that in order to get as much value as possible out of the penetration test it is important to follow known standardized guidelines and use tools that are designed for the job, for example tools that has been built or modified to work with ICS. It is also important that the organization define security objectives to reach in order to have more solid defined requirements for results [91]. Examples of well known existing guidelines are OWASP [96], NIST800-115 [91] and more, as well as dedicated literature such as Hacking Exposed ICS [8]. There is a set of standards called the ISO family, where ISO/IEC 27001 [97] can be helpful with the pentest, as well as ISO/IEC 27019 [98], however the last mentioned aims at information security controls for the energy utility sector in general.

6.3 Methods of Penetration Testing in ICS

This section will cover documented best practices and standardized ways of doing penetration testing.

The NIST SP 800-115 starts by describing an overall security test and assessment process and splits it into three main phases namely planning, execution and post-execution [91]. It describes that the planning phase is to be considered as any other projects planning phase with everything from staffing to scope, limitations and much more. However, in this planning phase it is also described the need to list what assets should be assessed, the creation of an approach to do it as well as listing of potential threats towards these assets and the corresponding defensive controls. The execution phase then works towards discovering and exploiting vulnerabilities, which is where the discussion on pentesting occur in NIST SP 800-115 chapter 5, and the post-execution then covers the creation of a report with description of the causes of the detected vulnerabilities and mitigation's. When looking closer at their penetration testing process, it is divided into four phases. The phases are planning, discovery, reporting and attack, however not necessarily in this order. The planning phase is described as the phase where rules, goals and approvals are agreed upon. The discovery phase is a two phase process where first is testing and second is vulnerability analysis. Looking at the attack phase, it has four sub phases that naturally leads from one to the other and connects the whole attack phase back to the Discovery phase in a circular process. The main principle of the Attack phase is that the attacker have obtained enough information in order to attempt gaining access to a specific system. This then leads to the attacker attempting to escalate privileges after gaining access to the system. After successfully escalating privileges the attacker may try to pivot into other systems by using the current system as an entry point [99]. It is also possible to install tools on some systems as seen in their last sub-phase, which will then help the attacker on reaching further and possibly open new paths inside the organization. They also state that the reporting phase is a phase that runs along all other phases, such that any findings of significance during any of the phases gets reported. As mentioned, when interviewing a representative from a utility company it was important to require documentation of steps done by the pentest team. The explanation was that when the pentesters interact with the systems, they should document when necessary what they did, in order to help the process of cleaning up, that is undoing system changes after the test is concluded. Further, as discussed in detail in the section above, NIST also writes that it is important for the pentesters to focus on the situations that are the most likely and those that can cause the most harm to the organization. This was a look at the NIST SP 800-115 early chapter 5 on pentesting [91].

Bodungen et al. on the other hand is more specific towards ICS systems and the additional considerations that has to be taken in the OT environments [8]. In their book from 2017 about securing industrial control systems they focus on a thorough risk assessment and threat modeling to provide information on where vulnerabilities may reside, who the attackers are and how the attackers may conduct the attacks. This then leads to the creation of risk scenarios which then guides the pentest.

In their own words, "*the pentest should be performed as a part of the overall risk assessment process*" [8]. All this before describing the pentest practices within ICS. Therefore, their whole pentest process builds on the risk assessment and threat modeling results. Having risk scenarios to guide the pentest will help stream line the efforts invested and therefore reduce the chance that resources are spent on pentesting a device that most likely don't pose a reasonable threat. Benefits from this strategy is reduced cost from labour and an assurance that what is deemed important is given the attention by those conducting the pentest. In order to evade causing damage or disruption to operating ICS they also suggest the use of a lab where the systems mirror those in operation when it comes to hardware, software and configuration, architecture and so forth. They continue by describing how such a lab can be set up, and a short summary will be given. When it comes to the production control devices, they are needed in the lab and must be of the same type and configuration as those in the production environment. Virtualization was suggested for replicating the production servers and workstations in the lab, with VMWare vCenter Converter[2] as an example of a tool that can create a virtual version of a live system [8]. They also mention the Microsoft Sysinternals Disk2vhd, which is also able to create a virtual hard disk of a system while the system is live [100]. Looking at their pentesting strategies for ICS, they depend on the situation the tester is in, but all start with the reconnaissance phase which is equal for OT as it is for IT. This phase is the usual searching online and learning as much as possible about the organization, such as from their website, forums, social media, and so forth. Tools that the authors found especially useful for ICS pentesting was Discover Scripts[3], Maltego[4] and Shodan[5], and Google hacking databases. The authors did not mention explicit Google hacking databases, however, an example of such a database is the Offensive Security GHDB that they write is integrated into Exploit Database[6] [101]. The next part of their pentesting methods are external. They start by listing allot of tools for pentesting such as Kali Linux and tools within that distribution, as well as tools that might not be found inside it such as some disassembles and fuzzers. However, they also mention SamuraySTFU, now named ControlThings Tools which is a Linux distribution like Kali but specialized for ICS [102]. Two scenarios are described as possible outcomes. The first are the one where the testers are successful in gaining access to the business network from the outside. And a second case where the testers are not able to gain access, but are given access in order to allow the test to continue. They will now be able to attempt pivoting deeper into the systems with the goal of accessing ICS networks. The testers may exploit lack of network segmentation and the existence of dual-homed workstations. This is then followed up by and finished of with a long series of ways to exploit systems and protocols such as TCP hijacking and ICMP related attacks, as well as sniffing, accessing historians, MiTM attacks and more. Their chapter is then divided into two sections, one for ICS device pentesting strategies and one for ICS server and workstation strategies. The first suggestion is to look at robustness testing which is testing using crafted packets and fuzzing in their example. The robustness testing will then reveal if a device is

[2] https://www.vmware.com/products/converter.html
[3] https://github.com/leebaird/discover/
[4] https://www.paterva.com/web7/buy/maltego-clients/maltego.php
[5] https://www.shodan.io/
[6] https://www.exploit-db.com/

struggling with a packet of a specific size or content. Robustness testing is different to device hardening since hardening is based on reducing unnecessary functionality that can be exploited, and as explained by MITRE it is also about applying privilegie restriction and reduction of connections device are allowed to have to what it needs to operate [103]. Bodungen et al. then suggest attempting to apply MiTM attacks that can be used for replay attacks, and altering of the contents is the second suggestion. A replay attack was used in the Stuxnet attack on the nuclear power plant in Natanz in 2010 [10, 11] . It recorded real data before it started to alter the speed of the uranium enrichment centrifuges, and replayed the recording to mask that it manipulated the centrifuges speeds later on. A summary of some of the following strategies is testing remote communication and engineering applications from vendor, reverse engineer firmware in case the vendor has placed hard-coded keys or login credentials. They also mention attempting to create a trojanized firmware and then check if the pentester are able to both download and install it on a device.

7 Requirements analysis for digital forensics data collection in CSSA

Contents of this chapter includes problems with forensics in ICS systems, forensics data collection in ICS and how the situational awareness architecture helps with gathering data that can be used for digital forensics in such systems.

First of all it should be mentioned that ISO 27037 is a guideline for handling of digital forensics data [104]. Even though this guideline does not mention ICS specifically, the guideline should still play a big part of the forensics process. Evidence should be handled with care and all actions done to the evidence material should be documented according to the Chain of Custody [105]. The Norwegian government produced a report by the name New Criminal Procedure Act which is a document of type NOU which stands for Norwegian Official Report [106]. The report mentions proper evidence handling in chapter 13.2.2.2. The chapter is focusing on handling of digital evidence from a legal stand point. This document is not focusing on ICS, but it gives insight into thought processes of similar problem situations.

In order to obtain situational awareness it may be necessary to conduct digital forensics on devices in order to obtain a clearer picture of what happened during an incident, and how this can improve awareness and decision making. And in this project the digital forensics focuses is at discovering what have happened, how to prevent it and harden the systems rather than focusing on who is behind the attack. The reason for this is that those who attacks power plants could be seen as advanced threats. When talking about advanced threats there is a term called advanced persistent threat (APT) that describes an entity with high level of expertise and was amounts of resources [107]. It is unlikely that a highly skilled threat or an APT wont put allot of resources into hiding them selves from being detected. Disgruntled employees and insiders are also a possible threats, and they possess large amounts of knowledge and expertise concerning the systems running in the plant, and insiders have access to systems accounts. Therefore they start off with allot access to run malicious commands, which is an ability that an outside attacker would need to work hard to obtain.

However, when implementing data collection, storage and analysis procedures such as those suggested in this architecture, the organization will possess large amounts of process, network and event data, STIX and VERIS files as well as correlation information. This should provide aid to the personnel conducting the digital forensics analyses after an incident have happened. They should be able to utilize this information in order to obtain insight into the incident, where in the system it happened and aid in uncovering the scope. Spyridopoulos et al. describes minimum information that should be obtained, starting with network diagrams, followed up by configuration details, change logs if they exists, and then authentication credentials [108]. As sources they also mention

temporary file systems, physical configuration and process tables, which is not collected by the suggested situational awareness architecture in this thesis. It is however suggested that physical configuration is to be stored in the Internal Storage part 8.3. And when on that point, Wu et al. mention that ladder logic should get its hash taken which should be safely stored elsewhere and used for verification later, which should be the aforementioned Internal Storage.

A series of decisions has to be done by system and security experts when it comes to the path forward. If the machines, such as an affected PLC or other device are part of the power production, a decision should be made on how to isolate a spread, and how forensics data that is not collected by the situational awareness architecture shall be extracted without causing loss of availability. If it is possible, and if it does not cause problems for availability, then memory of the device could be read, starting with the highest volatility. There exists tools to help with this, just like ControlThings that was mentioned in chapter 6.3 which was a pentest distribution for ICS. One of these are the SCADA forensics toolkit by Stirland et al. that was built on Wu et al. SCADA forensics methodology [109, 110]. Stirland et al. describe however that while it is possible to obtain live information from a PLC it was only possible in some cases in a lab environment. Also, as a side note, one can imagine a situation other than a power plant where the systems are producing medicine or other consumables. Then decisions should be taken based on safety risks. In this case that is risk to the consumers. This was exemplified with the dog food case described in chapter 1.5, which was not caused by an attack but a fault, where a small change in the ingredients caused severe safety risks for the intended consumers resulting in a sick canine population.

There are additional difficulties with doing forensics on ICS systems than those mentioned. One of these problems is the high need of availability in ICS systems and that the forensic analysis should not interrupt the devices as described earlier in this thesis. And when it comes to live forensics, Wu et al. mentions that volatile RAM is forensically unsound because it cannot be verified [110]. Another problem is how many different devices that may exist, the quantity of the data and the numerous protocols which may be proprietary. Bos et al. addresses the challenges with the quantity of data and data types that may arise when conducting digital forensics [111]. They proposed the domain specific data description language (DDL) called DERRIC which will help describe data formats. Another problem is that the systems are very often proprietary, and Spyridopoulos et al. describes that cooperation with vendors may therefore be necessary [108]. They also describe that the file system of a historian may have experienced changes if vendors have updated the system, which may cause evidence corruption. Wu et al. also explains that the OPC server and historians may not be forensically sound since they are used for specific purposes, and especially for the historian that could have been accessed by external systems [110]. This is due to the possibility that the external system can have been compromised.

As seen, forensics in ICS is not easy due to how fragile many of these systems are for unexpected interaction that they where not designed to handle. It is also not easy because of their need of availability and difficulties caused by proprietary design in both the systems and network communication. Sources of forensics data can be system logs from devices supporting it, network logs, event logs from both anomaly and signature as well as results from correlation. In addition there

are change discoveries such as when someone takes the hash of the ladder logic of a controller in order to discover changes in it, and also checking for changes in configurations called diffing.

8 Cyber Security Situational Awareness Architecture (CSSA)

It is important to have the ability to collect the information that enables the organization to discover, understand and mitigate threats and if possible the actors behind. To enable that collection it is important to know what information to look for, which requires deep understanding of the systems that runs in the plant, how they operate and what differentiates an attackers activity from the normal legitimate activity in the systems. These logs should be used both in real-time and stored for use and processing in retrospective. Furthermore, one can say that in order to identify suspicious or malicious actions, the aforementioned understanding of the existing systems in use must be combined with knowledge of how to identify the data sources. Identifying a data source is, for example, knowing type and location of sensors. Sensor could be anything from IDS's, firewalls and other systems. At the same time, it is important to know the type of data the sensors should collect from the different data sources to provide as much knowledge and overview as possible. Not only network sensors, but system logs and data from data historians must be collected.

In addition it could be important to mention that looking for information that should not exist in the network is also important. An example could be the use of a communication protocol in an industrial control network where that protocol is not meant to be used. This is an indicator that something is wrong.

Since there are so many ways that systems can be attacked, the cyber situational awareness architecture needs to be extremely robust and hardened. The system has to be robust enough to resist attacks. The system has to be able to discover attack patterns even though attackers actively tries to hide them. And the third and last, it has to be able to condense allot of information so that the operators who view it are not overwhelmed by what they are presented, and equally important that details is not lost during any pre-processing. Of course, there are many steps in between those mentioned, which will be described within this chapter.

An overview of the architecture seen in figure 6 displays an abstraction of the architecture seen from above, but the figure does not contain all details such as every algorithm used. It displays the five main parts starting with data collection, followed up by two pre-processing stages. In this part of the pre-processing information that would be lost at higher levels in the architecture is added as metadata. The last part of the pre-processing stage is called the Normalization stage. In that part a normalized version of the collected data is generated, which means it is transformed into standardized formats. Now, the organization possess an additional version of the data in a normalized format. This phase directs the data in possible directions. First is an internal storage of known vulnerabilities in the organizations systems, that must be taken into consideration. The knowledge of these vulnerabilities is by itself situational awareness the organization needs and

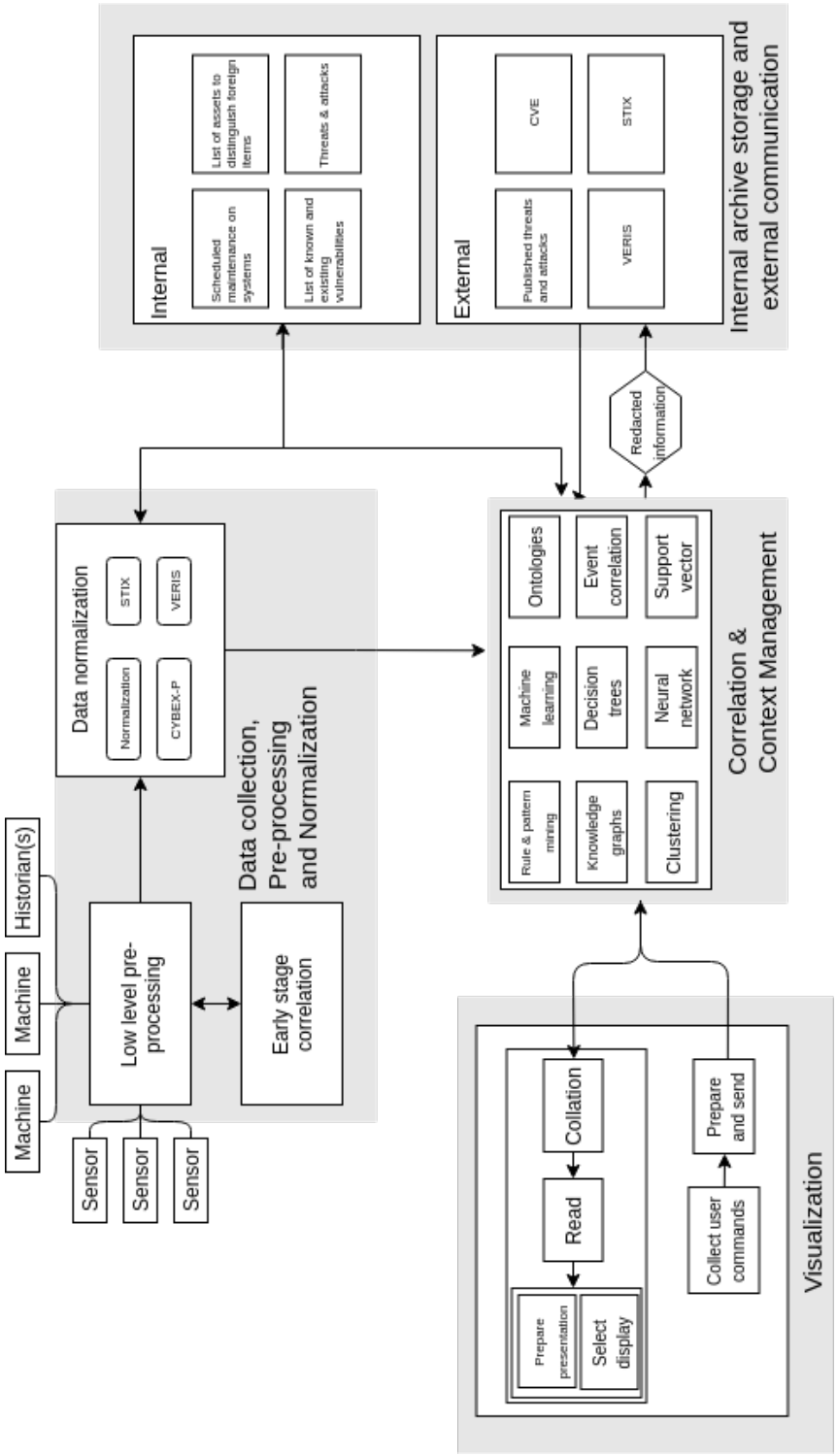

Figure 6: Summary overview of situational awareness architecture

helps determine the threat scenarios affecting the organization.

The other path the Normalization stage leads to is the Correlation and Context Management stage. It is important to note that this stage may need both normalized and the original data in pre-processed version. Pointing out again that pre-processing removes unwanted data adds metadata, and normalization transforms data into new formats. This phase correlates information and alarms from systems and produces new information and situational awareness. In addition to receiving collected alarms, it should be able to generate new alarms from collected network and system logs, depending on algorithms that ends up being implemented.

Information is then shared with confidentiality ensured with external partners, shown at the right side in the overview figure 6. At the left side is the Visualization stage which is the other direction the information can travel from the Correlation and Context Management stage. It displays the resulting information to the operators to provide situational awareness.

It is important that real-time information is fresh, which means that it is recent enough to allow quick response from operators. The information generated by the architecture should also have high reliance and trust. Stored information should be possible to be processed by the systems in retrospective.

8.1 Data collection

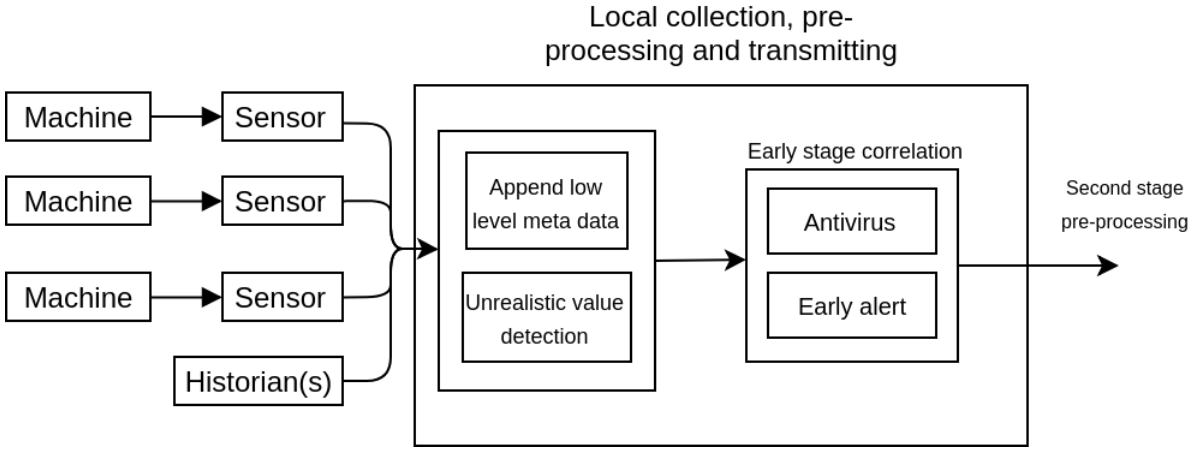

Figure 7: Data Collection and Early Pre-processing

This subsection will describe in short terms how the data collection phase of the situational awareness architecture works. It will be relevant to collect information from all levels of the reference architecture 4.1.

In order to do data collection it is first important to identify all sources needed. System generated data is the data generated by the machines and software that runs the plant. This data includes operational status of machines, devices and services running such as their health, logs on how

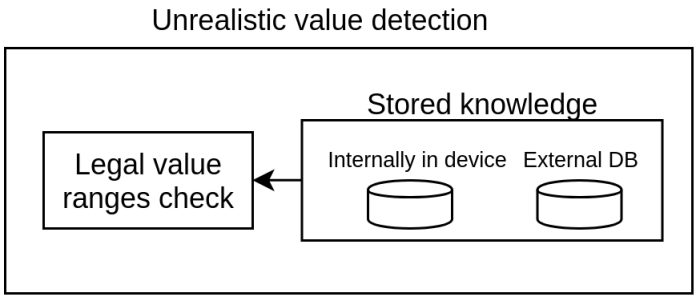

Figure 8: Filtering unrealistic values

they are running and logs describing errors experienced, notifications about received commands and more. Network data is data collected by network devices and sensors placed in the network. Examples are devices such as switches, routers, firewalls, IDS/IPS, sniffers. In addition there is vulnerability scanners and SIEM. Figure 7 shows input sources which are also part of the pre-processing. Pre-processing done at the collection stage has the purpose of appending metadata that would be lost at higher levels. This thesis does not cover the collection part as that is handled by another thesis [13]. However, the collection and the early stage pre-processing therein is part of the architecture, it will be touched upon architecturally from a high view point. As can be seen in figure 7, a general idea of the processes happening at the data collection phase is displayed. Input can be aggregated depending on its source and context and values that are deemed unrealistic should be marked to prevent it from being processed at a higher level in the architecture where such data might not be recognized as being wrong 8. A cause of unrealistic values could be a bug in software, a failing sensor or a transmission problem. A hypothetical example could be numeric values from measurements that are so high that they could never have happened, for example a temperature that is unrealistically high or low. Details about how to detect these values are looked at in more details shortly. The unrealistic data is also being sent to the early correlation stage so that statistics can be done to monitor the amount, frequency and origins of the errors. Then low level meta data gets appended to all the data collected. Immediately, packet inspection and scanning for patterns in the data flow representing known attacks is should be done, followed up by an early warning system that alerts other systems and operators if known fingerprints are found. Therefore the system that runs the scan must have the ability to notify operators or a communication path to the visualization phase must be created depending on how this is implemented.

Taking a closer look at the detection of unrealistic data, as shown in figure 8, there could be possible ways to do it. To detect if a value is unrealistic it could be compared against ranges of values expected to be the most extreme possible cases. If the device conducting the measurement is advanced enough to do this comparison it would help remove the need for communication with databases and other systems that otherwise would be needed, however this might not be advisable. The reason for this is that an attacker might try to target the system that runs these checks in

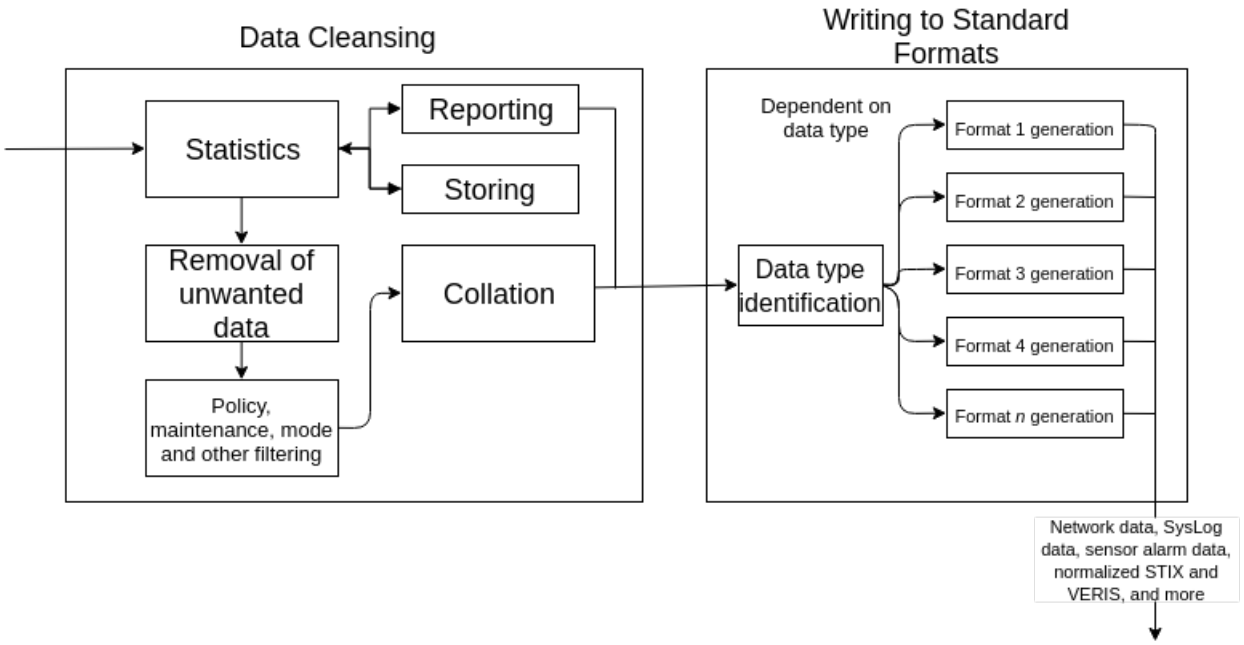

Figure 9: Data normalization

order to make it deem the values as acceptable or completely miss them. That way an attacker can conduct its activity without worrying about triggering a response in such checks. It is also important to not ignore or drop these unrealistic values after detection, but rather mark them with metadata before feeding them deeper into the system. This is because they can be used in statistics later, since they reveal that something went outside design such as a bug, that a sensor malfunctioned, or something malicious may have happened. Either way it hints at that something strange occurred or that a device might need maintenance. It is also important to read up on information about expected error rates from sensors and other systems that may have been published by vendors or could be found as published information on the Internet.

8.2 Data Normalization

There is a need to cleanse and normalize parts of the data in an effort to prepare it for the correlation systems and sharing with external community members. The architectural design for the data normalization stage is displayed in figure 9. However, this works as a concept displaying an idea of data flow through suggested processes. The data normalization stage purpose is therefore to remove unwanted data which may be any data deemed not necessary to process further, run it through filters and then collation helps order data based on time, size or other metrics. The data described as not needed will be filtered away. An example of data not needed is repeated information containing statements that everything runs as expected which many systems may regularly

produce. Then the information is sent through a set of to screens the incoming information against organizational policies, mode of operation, known maintenance's and lists of legal values. Parts of a plant may be in different modes of operation and therefore it is important that the system does not overflow the operators with warnings just because of a mode of operation change. Mode of operation can be seen in different ways such as day cycle of a power plant, the plant is in normal or emergency state, a state of recovery after an incident or disaster, part of the plant may be in a maintenance state and more. To make an example, if some machines are in maintenance state, and therefore parts of the production is down or running under manual control, it would be counter productive if the situational awareness system gave critical warnings about it, when the cause is known. Another example can be if a fire has made a machine at the plant unusable and repair has been ordered. Then the operators should be able to turn off warning about that machine not operating during the time it takes for repair to be conducted or a new machine to arrive and be installed.

In the next part of the normalization, the Data type identification will take the incoming data and send it to a Format Generator that can use it. The Data type identification step should work in some ways similar to a network switch, but instead of switching by MAC addresses it reads the metadata of the incoming traffic of information and redirect it towards the correct format engine. This engine will transform the incoming information into a standard format.

There exists already projects that have produced and to this day maintains languages for normalizing data into standardized formats such as STIX and VERIS, which are already widely accepted by the security communities [52, 60], and is described in related work chapter 3.4. There are many reasons for doing normalizing into standardized formats. By utilizing a standardized format it will be more practical to make sure that information shared with external parties does not contain information that should not be shared. It will also be more practical to handle anonymity, as cryptographic primitives can be used automatically by systems during conversion to new format. An example can be that parts of an information object becomes encrypted, so that only those with a key can read it.

The aforementioned languages are designed to be readable by both humans and machines and allows information to be written in a standardized way which helps support automation. And as seen in section 3.4 work has been done on automating the generation of STIX information from raw data by Sadique et al. through the CYBEX-P project [2, 66]. An example is shown in figure 11 where their research can be useful. In the example in the figure, DCPM means data collector plugin module, which Sadique et al. created to make CYBEX-P able to read and understand many different types of data, including raw data from systems. In this case, one can imagine the Data Identifier sending data into Format Generator 1, where that generator is running Sadique et al. CYBEX-P [2]. The Data Identifier would collect all incoming data that CYBEX-P had available plugins for and redirect it to the DCPMs.

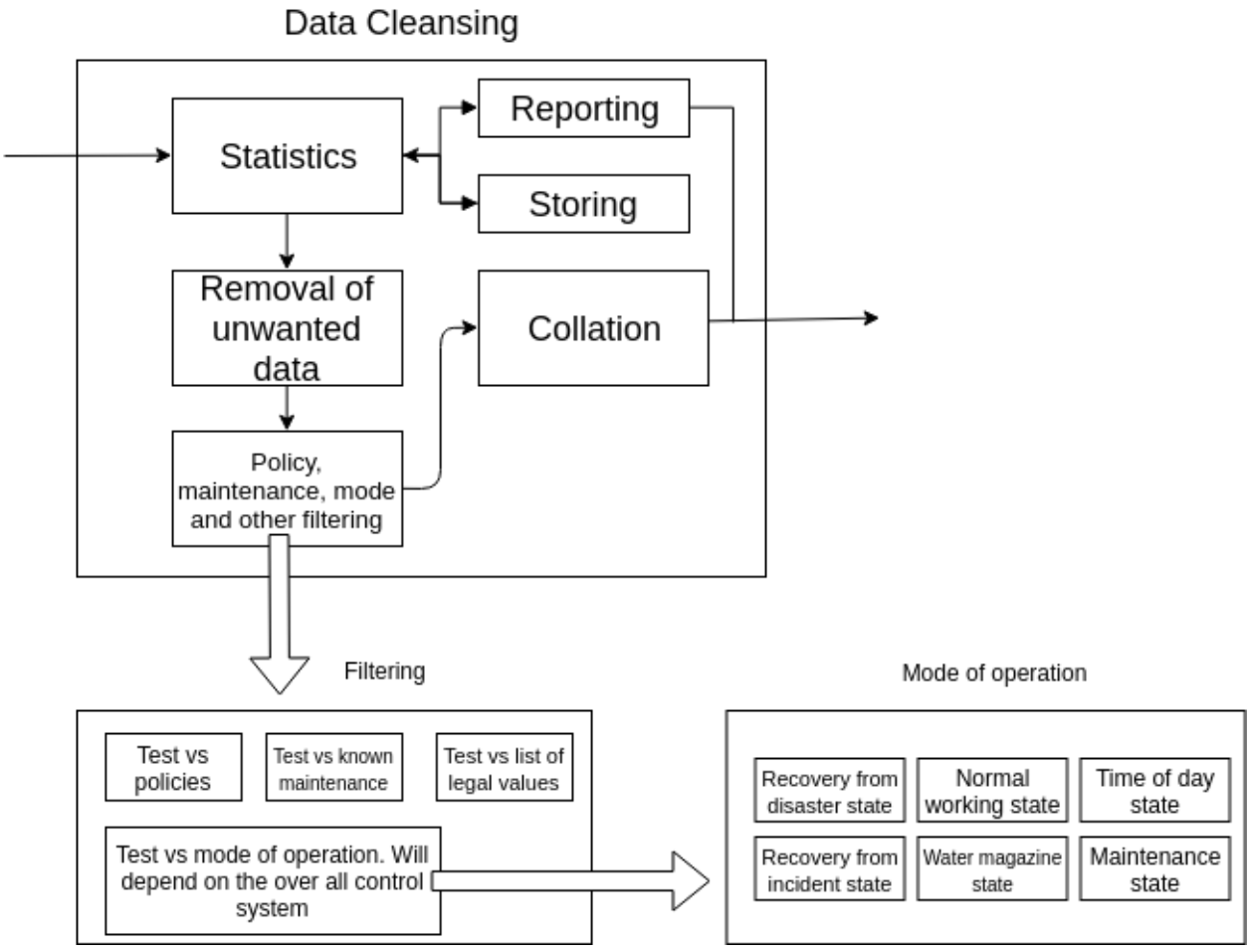

Figure 10: Filtering based on mode of operation, policies, maintenance and legal values

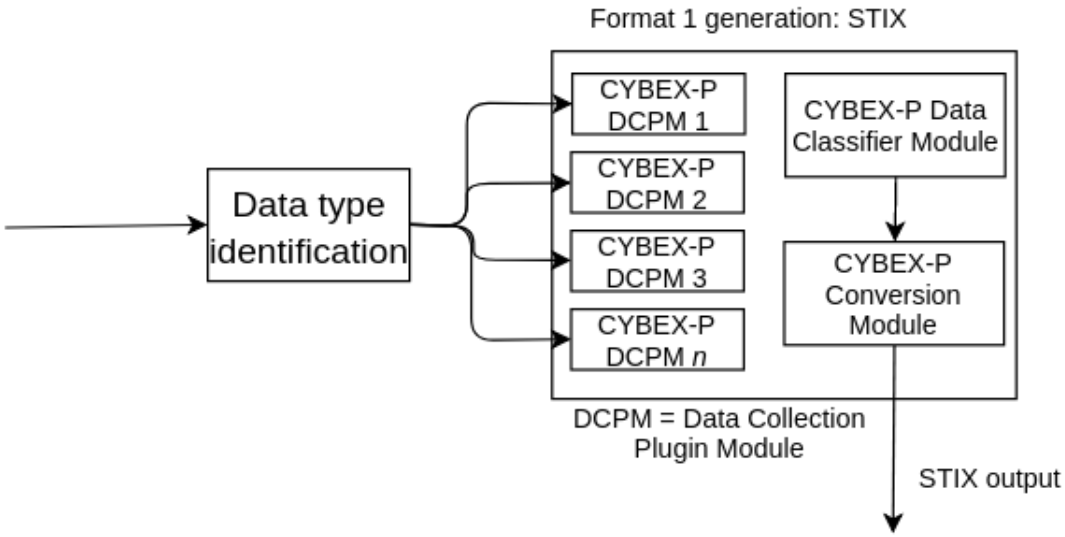

Figure 11: Example of using Sadique et al. CYBEX-P [2] as Format Generator 1

8.3 Internal Storage

The architecture contains internal storage shown in figure 12. It stores known malware signatures, attack patterns, assets owned and data that can be used to describe normal use of machines and systems. To expand on the latter, the industrial systems should not regularly be altered or modified after installation but will rather stay in operation within current design until modification of the plant is occurring. Therefore, learning normal working patterns of systems and networks should be more practical than in normal IT environments where there are more change and the usage of the systems is not as predictable.

The internal storage systems therefore must contains information of recorded data such as months of system history and actions done to the systems such as commands send by operators, in order to help track sources of potential attacks. This information is fed from the pre-processing and Normalization phase. The Correlation and Context Management phase will also need to communicate with this part of the architecture in order to fetch historical data and to store results found.

In addition to store system and network logs and information related to vulnerabilities both found internally and publicly known, it should contain information planned maintenance's. This includes knowledge about which systems are to receive maintenance or repair and the data of the occurrence. In addition to this the organization should keep lists of assets that it owns or operates in the plant. This is not only useful to enable the maintenance documentation but also when conducting penetration testing. As described in chapter 6 about pentesting in industrial control systems, a pentester cannot simply run ping sweeps to obtain information about existing systems as this could

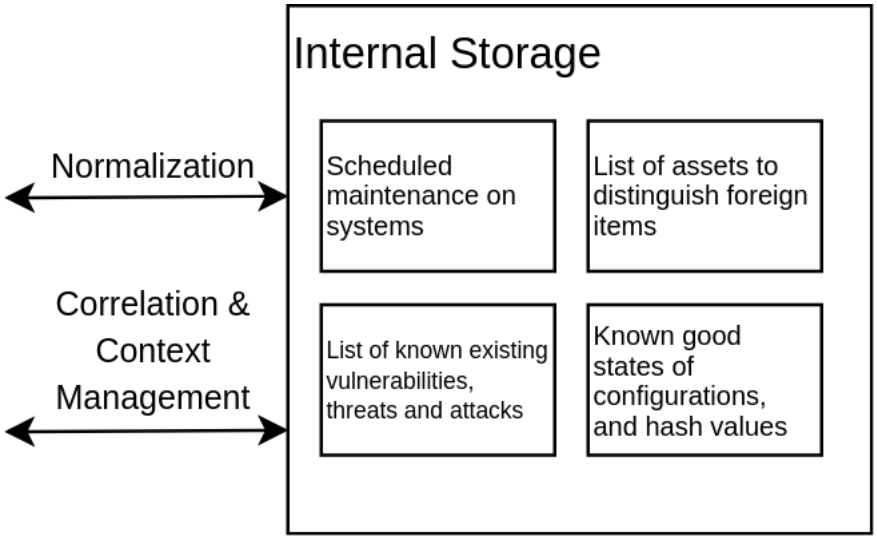

Figure 12: Internal Storage

cause problems with system availability. Having these lists of assets can help increase the awareness of the pentester on which systems exists.

Since the organization catalogues both vulnerabilities existing in their systems as well as vulnerabilities shared in communities, it will be a need to keep in mind the security of how this data is stored.

8.4 Correlation and Context Management

In this section the Correlation and Context Management part will be discussed. The goal is to be able to correlate both information from the pre-processing phase and information from the outside with aim at uncovering threats and attacks against the systems in the organization. The correlation is conducted with information from several different systems, sensors and historians. This is both historically collected as well as real time data. One reason why it is important to correlate with historical information is because it could be that the attackers has spread their activity over a time period in order to reduce noise and to make it more difficult to see the connections between the tracks the attacker leaves. This is a result of the attackers doing their best to hide their activities. This also leads the attackers to try to make their activity look as normal as possible in order to make them dissapear in the background noise of legitimate user activity.

An attack campaign, as described in SANS ICS Cyber Kill Chain, which is built on Lockheed Martin's kill chain shows that attackers may operate in different phases spread over a time period [95, 94]. This shows that it is important to have a good database which allows correlation of information collected over time, as well as good cooperation with external parties in order to uncover threats. If one organization discovers leads on something that might be a threat, but needs more

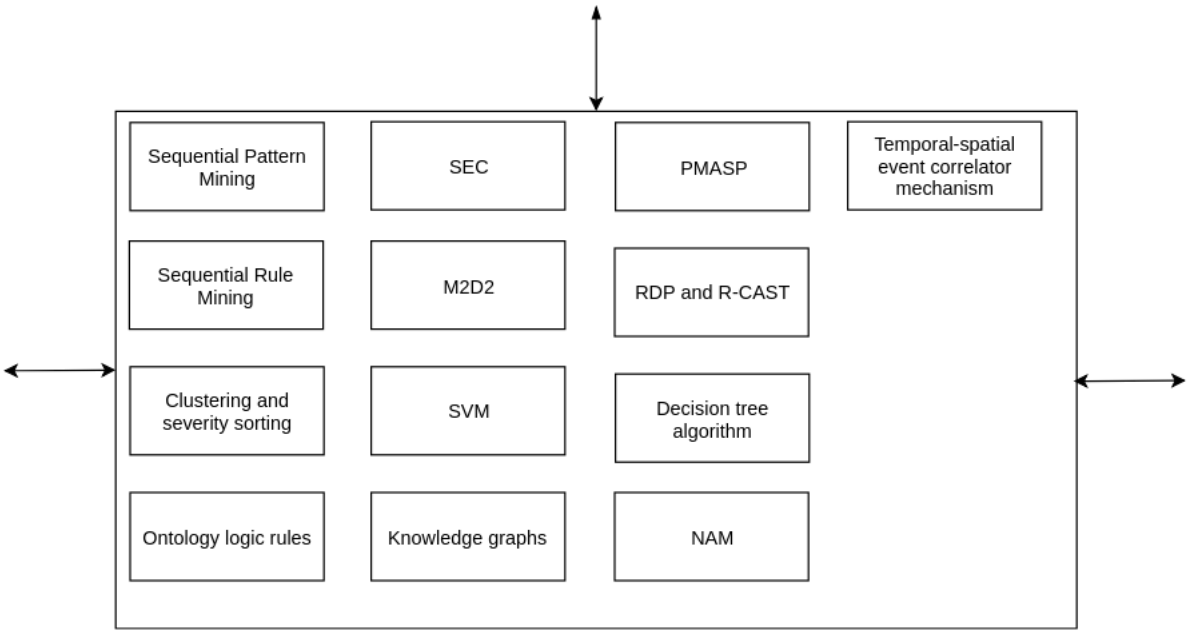

Figure 13: A sample of existing correlation technologies

information, it might be that another organization have done similar detection's in their own systems and therefore possess other leads of a similar attack or campaign. When cooperating on cyber threat intelligence, this information may be correlated together.

When it comes to the definite choice of correlation method, it is necessary to do this in a separate research project. There exist numerous systems and algorithms that are able to correlate alarms and system information. Many have been discussed in the related works chapter 3.3 and in this section.

The correlation phase will need to communicate with the pre-processing and Normalization phase in order to collect information about the organization. This phase also needs to have a two way communication with external parties in order to share and receive cyber threat intelligence, which is used in the correlation. Two way communication with the visualization phase is also important. This will allow information to be sent to that phase and be visualized for the operators as well as allow the operators to utilize interfaces to communicate back to the correlation phase. This is because their user interfaces must be able to provide functionality and receive queries from the operators. The architecture in this project does not describes how this would be in technical details in this project, when design and implementation of it is a project on its own. Figure 13 shows some technology that are able to run correlations, handle context or both.

Simple event correlator called SEC was an early rule based correlator from 2002 written by Vaarandi et al. [38] meant to be light weight and platform independent. M2D2 was a formal model for security information representation and correlation by Morin et al. that uses information about

hosts to determine if they are vulnerable to know vulnerabilities [39]. It also model alerts, scans and log events, identifies which system generated an alarm, what the cause of the alarm was, and more. PMASP which stands for Purpose-oriented Maximum Attack Sequence Patterns was created by Lu et al. as a system that could find frequent attack sequence sets from clustered data [49]. These clustering would cluster alarms into four classifications called buffer overflow, DoS, web attack or "other attack". Temporal-spatial event correlator mechanism was proposed by Deng et al. correlating events over several processes and systems over time [35]. Seven sequential pattern mining and three sequential alert mining algorithms was tested by Husák et al. where they evaluated the performance of the sequence mining approaches and the usability and comprehensibility of the output [47]. Since many methods exists it could be valuable to repeat that they used the SPMF library[1] to select candidate methods. RDP and R-CAST was used by Yen et al. to help operators gain situational awareness in complex situations and to help reduce limitations between human mental models and the cyber world [34]. RDP stands for Recognition-Primed Decision and is a model that helps domain experts relay how they think and gain SA. R-CAST is Cognitive Agent Architecture are computerized intelligence's that aids the domain experts and each agent can communicate with each others. Clustering and severity sorting was proposed by He Wei, where correlation rules are used and ranking on severity is done based on probability of successful attack, importance of target and and priority of security event [42]. SVM stands for support vector machine and was used by Chen et al. as masquerade detection [41]. Decision tree algorithm was used to for estimations regarding cyber attacks by Pournouri et al. [37]. In addition to collecting information and correlating it to obtain new information and insight it could be useful to have tools that may help estimate future trends based on whats known. They do mention that their model could be better at finding the most likely type of threat and type of target, but Pournouri et al. suggest that Naive Bayes and Neural Networks could provide good results on their data set. ONTIDS is an ontology-based and context aware rule based correlation method proposed by Sadighian et al.[43]. Knowledge graphs was used by Wang et al. in their proposed system called KGBIAC, which stands for Knowledge Graph Based Intelligent Alert Correlation Framework [45]. It makes it possible to link entities based on related information. A host with software that has known vulnerabilities will then be related with those vulnerabilities, and attacks that attempts to exploit vulnerabilities the host is not related with will be labeled as false. Attacks can also come from the inside, whether from full time employees or from third parties hired to do work for a period of time. Therefore it could be useful to have methods that help protect the organization from these threats. A whole architecture containing Neural networks and Associative Memories called NAM was proposed by Brancik et al. to detect insider threats [33]. NAM was used to identify normalcy benchmarks and flag anything outside these benchmarks as suspicious.

It is especially important that those working on implementing the context and management methods and algorithms are communicating with those responsible for the normalization phase

[1] https://www.philippe-fournier-viger.com/spmf/map_algorithms_spmf_data_mining097.png

and those handling the visualization and notification of operators and other employees. The correlation and context management phase becomes the connection point between the normalization and visualization phase.

There are many reasons to include information from external parties when doing correlation. First of all it is logical to assume that attackers has a strong interest of staying hidden. Therefore one cannot expect to find all traces of the attackers immediately and identify them. However, as previously mentioned in this chapter there exist kill chains that the attacker may follow. It is likely that a group that attacks a power provider would do this in a campaign that follow steps similar to the SANS ICS Cyber Kill Chain [95]. If the victim organization are able to detect some traces of an attack, while another organization detect other attacks from the same group, then collaboration between these two organizations will provide a larger set of data to work with. More data to use in correlation and context management will lead to better SA.

There also exists many external sources for information that can be used to correlate ones own systems with in order to gain awareness if they are affected in addition to partners the organization shares information with. Examples of databases of known vulnerabilities are as NVD[2] and CVE[3]. US-CERT[4] also has a list of current high impact security incidents that they keep updated.

8.5 Visualization

Visualization is about presenting the information generated to the operators in a way that allows them to observe it in an effective, fast and comprehensive way without loosing important details. The visualization system receives the information from the correlation and context management phase. The data needs to be compressed to fit the size of screens and other ways of displaying, such as projectors and notifications to hand held devices. It is important that the architecture provides the information the operators needs in order to safeguard the plant from malicious actors in a way that is most useful to them. It is therefore important that there are carefully planned goals with what the information presented to the users should solve and how this information will help them. The information must be condensed to provide an overview of the situation without loosing the ability to look at low level details. The operators and security personnel must therefore be able to zoom in on specific parts of the abstracted information in order to reveal details necessary to solves the situation and prevent it from happening again. Looking at the architecture in figure 14, the information this phase receives is first collated in order to let it be orderly accessed by the sub-phases. Then the type of information, for example if it is a push notification, is it information representing an attack path, is it an emergency alarm or many other possible types. Metadata helps describe the information received from the correlation phase. That can be metadata describing importance and miscellaneous meta data. Miscellaneous meta data can be other type of information that is considered important to the goals of the visualization phase but could not be classified as

[2] https://nvd.nist.gov/vuln
[3] https://cve.mitre.org/cve/search_cve_list.html
[4] https://www.us-cert.gov/ncas/current-activity

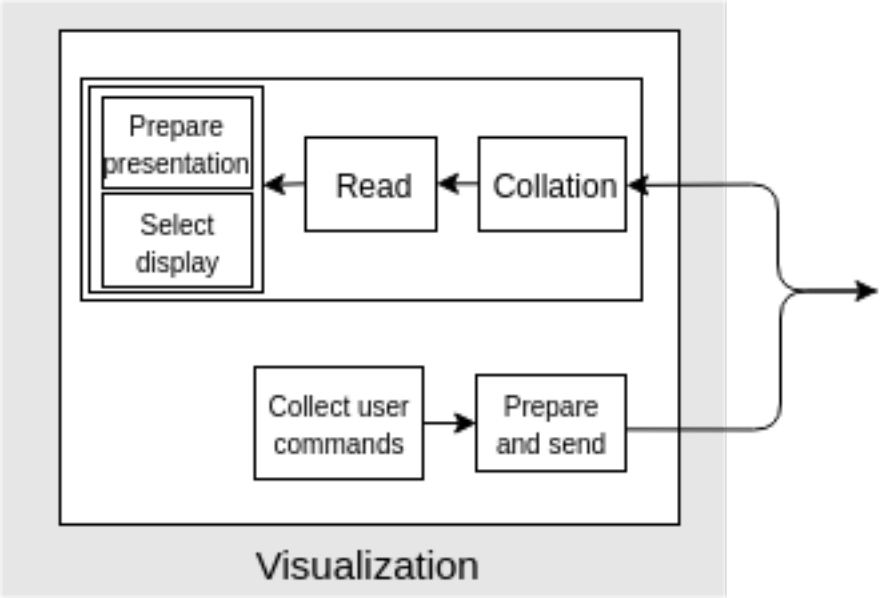

Figure 14: Visualization phase

one of the defined metadata values or be defined as an importance category when processed in the correlation and context management phase. The next sub phase is to decide how the information should be displayed and presented. For example, should it be presented on a large screen only or should notifications also be sent to hand held devices, should colors be used to emphasize the alarm such as red text or background, and more. The visualizations should also have the ability to display estimations of future trends computed in the correlation phase. For example, if a device experiences an increase in traffic over a period of time, then the system should be able to display a forecast for a time interval ahead based on this information. It is important that the operators can interact with the visualizations, and the user interface therefore needs to collect the user commands and send it to the correlation and context management stage.

9 Discussion

It is very important to secure industrial control systems. Critical infrastructure such as power production relies on it, and the requirement for availability in these systems are very high. The work to secure such systems require allot of knowledge on how they work. They have to be handled with care and testing should be done on a lab that represents the systems in operation. Conducting risks analyses should help determine the context of penetration tests out of best knowledge, and is discussed in chapter 6. However, when it comes to forensics, there will always be a strong wish to analyze the affected systems. But if these systems are running critical infrastructure it might be a risk to do live analysis in case it could inadvertently harm availability. Hopefully the cyber security situational awareness architecture can provide useful correlated and raw data to be used in a forensics analysis in addition to provide real time situational awareness.

There are also costs when implementing the architecture. Machines are needed to build the architecture enabling it to receive network data, system logs, and logs from firewalls, anti-viruses, intrusion detection systems and more. It is also possible that the architecture may introduce new unknown vulnerabilities as it introduces new components alongside existing systems. The information has to be transported from the sources and stored. Machines are required to conduct the correlation, sharing and visualization. However, allot of information is already stored in historians and can be accessed from there. The correlation phase will generate new information from the existing information which also has to be stored. It also requires personnel to monitor its results, and will require maintenance. However, it is important to take into consideration the amount of automation the architecture is based on. The architecture therefore facilitate automation of the collection, correlation, sharing of information and visualization. The results provides aid to anyone that works with security and safety in the organization, and the information generated may help in the detection of errors and faults in systems that not necessarily are security related.

Lets take a closer look at the requirements for the architecture and detection of threats against the organization. The architecture are able to collect information from many internal sources as mentioned above, as well as the external communities using VERIS and STIX. During correlation large amount of events from sensors and systems will be analyzed based on context and type depending on which algorithms are selected for implementation, in order to uncover what potential threat they represent. For example, it would help to answer the question if a specific event is part of a larger attack, rather than it for example being a single failed login attempt. Another requirement was that it should be able to store information relevant to its purpose. Therefore a region of the architecture was dedicated to store information received from two other phases of the architecture. The internal storage communicates with both the pre-processing and the correlation and context management phases for several reasons. First of all it is important to store information from the

pre-processing and normalization phase as this information represents data unaltered by the correlation algorithms. This could also prove useful for digital forensics as mentioned above. If some systems are set to only generate data necessary for them to function or for maintenance reasons, the architecture will provide additional systems data collected for the purpose of security when that is possible. However, when collecting data from systems it is important to make sure that the systems wont suffer under the strain of additional logging and communication. It is necessary to store information from the correlation and context management as well. This information will both aid with statistics as well as it will provide the possibility to correlate real time collected events with historical events. The benefit from correlating real time events with historical events is that it may help with uncovering campaigns that have happened over a period of time.

10 Conclusion

The thesis contains the groundwork for a cyber security situational awareness architecture for industrial control systems. It also contains information about penetration testing and digital forensics on the subject. It was natural to start with penetration testing when that is preventative work. To conduct a penetration test on operational technology (OT) such as industrial control system is different that on IT systems. OT systems are sensitive towards communication they are not designed to receive, which may happen during a penetration test. These systems are also required to have very high availability, and it is therefore undesirable to conduct a penetration test directly on systems in operation. The importance of availability is also problematic for digital forensics, which are conducted after an incident have happened. Context of a penetration test is important since it defines which systems to be included in the test, and helps the preventative security work and makes the test more economical. The context should be defined through a risk analysis that determines which assets have a high value to the organization and its ability to produce energy. Without a context defining the systems to be included in the test, the penetration testers may spend time testing systems of less importance. The results will then be if the personnel conducting the test are able to penetrate the systems. An organization may also list systems that is not to be part of the test, to make sure that the test does not harm availability of systems of critical importance. If the penetration testers is not able to gain access to parts of the systems, they can be provided that access to make sure all systems in the scenario are tested. The organization can create requirements for context of the test, and by that also what results they are looking for. Since it is highly undesirable to conduct tests on equipment in operation, therefore an additional requirement should be to create a test lab. The lab must have similar setup and machine configuration as in the operational system.

The components of the cyber security situational awareness architecture was data collection, pre-processing and normalization, internal storage, correlation, visualization and external sharing. The correlation phase receives information internally and externally and its results are presented in the visualization phase. The results are in real-time to allow operators to act on an incident when it happens. The visualizations should provide information in a condensed and readable manner without loosing important details.

When it comes to digital forensics it is important to have the ability to collect as much relevant information as possible in order to uncover what happened and how. As mentioned earlier it may be challenging to do forensics on operating machines as this might harm availability. These systems are also often proprietary and it may therefore be important to have collaboration with suppliers. Still, changes in the systems can be detected by comparing the configuration on a device with one that is stored safely, and as mentioned by Wu et al. a hash of ladder logic can be taken and used for comparisons [110]. It is therefore important that the architecture can provide as much information

about an attack as possible to facilitate a forensic analysis afterwards.

10.1 Future Work

As future work, rigorous test of the architecture and its suitability for acquiring and processing forensics information should be done. An analysis of which data that is needed and which algorithms that solves the specific problems must be done. It might not be easy to collect data that contains real incidents, however data with incidents could be produced in a lab environment. Then different algorithms should be tested on the data and the results should be handed over to experts. The experts can then give feedback on usefulness and value. Different visualization techniques should be tested on the results, and experts should be included again to test different variations of the visualizations and provide feedback.

In addition to that, there should be done research on what the architecture needs to include in the future. For example, if the organization decides to automate sales of electricity to organizations such as Nord Pool[1], the process that enables the automation should be monitored by the architecture. The architecture would now collect data from an additional set of systems and sensors in the data collection phase.

When it comes to digital forensics in ICS, research on actual hardware should be done in order to map which device it is possible to obtain useful information from. Literature review on the topic has been done such as Asmar et al. who created a literature review of research on SCADA forensics framework and methodologies, network forensics and device forensics [112]. They also provide a table listing research publications on these topics. The publications described there could be a starting point for more research.

[1] https://www.nordpoolgroup.com/

Bibliography

[1] ENISA. 2016. Communication network dependencies for ics/scada systems. https://www.enisa.europa.eu/publications/ics-scada-dependencies.

[2] Sadique, F., Cheung, S., Vakilinia, I., Badsha, S., & Sengupta, S. 11 2018. Automated structured threat information expression (stix) document generation with privacy preservation. URL: https://www.researchgate.net/publication/329465573.

[3] Mavridou, A. & Papa, M. 2012. A situational awareness architecture for the smart grid. In *Global Security, Safety and Sustainability & e-Democracy*, Georgiadis, C. K., Jahankhani, H., Pimenidis, E., Bashroush, R., & Al-Nemrat, A., eds, 229–236, Berlin, Heidelberg. Springer Berlin Heidelberg.

[4] Barford, P., Dacier, M., Dietterich, T. G., Fredrikson, M., Giffin, J., Jajodia, S., Jha, S., Li, J., Liu, P., Ning, P., Ou, X., Song, D., Strater, L., Swarup, V., Tadda, G., Wang, C., & Yen, J. *Cyber SA: Situational Awareness for Cyber Defense*, 3–13. Springer US, Boston, MA, 2010. URL: https://doi.org/10.1007/978-1-4419-0140-8_1, doi:10.1007/978-1-4419-0140-8_1.

[5] Endsley, M. R. 1995. Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1), 32–64. URL: https://doi.org/10.1518/001872095779049543, arXiv:https://doi.org/10.1518/001872095779049543, doi:10.1518/001872095779049543.

[6] Om kraftcert. https://www.kraftcert.no/om.html. Last accessed: 2019-02-17.

[7] International Data Corporation (IDC). Mars 2018. Worldwide spending on security solutions forecast. https://www.idc.com/getdoc.jsp?containerId=prUS43691018. Last accessed: 2019-02-17.

[8] 2016. *Hacking Exposed Industrial Control Systems: ICS and SCADA Security Secrets & Solutions*. McGraw-Hill Education Group, 1st edition.

[9] Lee, R. M., Assante, M. J., & Conway, T. March 2016. Analysis of the cyber attack on the ukrainian power grid. *SANS ICS, E-ISAC*. URL: https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf.

[10] Langner, R. May 2011. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security Privacy*, 9(3), 49–51. doi:10.1109/MSP.2011.67.

[11] Falliere, N., Murchu, L. O., & Chien, E. 2011. W32. stuxnet dossier. *White paper, Symantec Corp., Security Response*, 5(6), 29.

[12] 2018. Ramaskrik blant hundeeiere - mener hundene har blitt syke av norsk hundefor. https://www.vg.no/nyheter/innenriks/i/21AxzB/ramaskrik-blant-hundeeiere-mener-hundene-har-blitt-syke-av-norsk-hundefor. Last accessed: 2019-02-18.

[13] Sand, K. A. Incident handling, forensics sensors and information sources in industrial control systems. Master's thesis, Norwegian University of Science and Technology (NTNU), Gjovik, 05 2019. Not published.

[14] UNESCO. 2015. Experimental development. http://uis.unesco.org/en/glossary-term/experimental-development.

[15] OECD. 2015. Frascati manual 2015 guidelines for collecting and reporting data on research and experimental development, the measurement of scientific, technological and innovation activities, oecd publishing, paris. https://read.oecd-ilibrary.org/science-and-technology/frascati-manual-2015_9789264239012-en.

[16] Nuseibeh, B. & Easterbrook, S. 2000. Requirements engineering: A roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, 35–46, New York, NY, USA. ACM. URL: http://doi.acm.org/10.1145/336512.336523, doi:10.1145/336512.336523.

[17] Zave, P. December 1997. Classification of research efforts in requirements engineering. *ACM Comput. Surv.*, 29(4), 315–321. URL: http://doi.acm.org/10.1145/267580.267581, doi:10.1145/267580.267581.

[18] Anderson, R. Security engineering, second edition. *Wiley*. Chapter One: https://www.cl.cam.ac.uk/~rja14/book.html.

[19] for Standardization, I. O. Oct 2017. Iso/iec 27019:2017. https://www.iso.org/standard/68091.html.

[20] Keith Stouffer, Victoria Pillitteri, S. L. M. A. A. H. 2015. Nist special publication 800-82 revision 2. https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-82r2.pdf.

[21] Comission, I. E. May 2007. Iec ts 62351-1:2007. https://webstore.iec.ch/publication/6903.

[22] Lov om nasjonal sikkerhet (sikkerhetsloven). https://lovdata.no/dokument/NL/lov/2018-06-01-24/. Last accessed: 2019-02-20.

[23] The norwegian water resources and energy directorate. https://www.nve.no/english/. Last accessed: 2019-02-20.

[24] Skapalen, F., Ulsberg, H., Steen, R., Arnulf, R. C., & Sønsteby, T. Veiledning til forskrift om beredskap i kraftforsyningen. http://publikasjoner.nve.no/veileder/2011/veileder2011_01.pdf.

[25] Kraftforsyningens beredskapsorganisasjon (kbo). https://www.nve.no/damsikkerhet-og-energiforsyningsberedskap/energiforsyningsberedskap/organisering-av-energiforsyningsberedskap/kraftforsyningens-beredskapsorganisasjon-kbo/. Last accessed: 2019-02-21.

[26] Christiansson, H. & Luiijf, E. 2008. Creating a european scada security testbed. In *Critical Infrastructure Protection*, Goetz, E. & Shenoi, S., eds, 237–247, Boston, MA. Springer US.

[27] Norwegian cyber range. https://www.ntnu.no/ncr. Last accessed: 2019-03-02.

[28] Duggan, D., Berg, M., Dillinger, J., & Stamp, J. 2005. Penetration testing of industrial control systems. *Sandia national laboratories*. https://energy.sandia.gov/wp-content/gallery/uploads/sand_2005_2846p.pdf.

[29] White, F. E. 1991. Data fusion lexicon. URL: https://apps.dtic.mil/docs/citations/ADA529661.

[30] Hall, D. L. & Llinas, J. Jan 1997. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1), 6–23. doi:10.1109/5.554205.

[31] Tadda, G. P. & Salerno, J. S. *Overview of Cyber Situation Awareness*, 15–35. Springer US, Boston, MA, 2010. URL: https://doi.org/10.1007/978-1-4419-0140-8_2, doi:10.1007/978-1-4419-0140-8_2.

[32] Concept. https://www.merriam-webster.com/dictionary/concept. Last accessed: 2019-03-17.

[33] Brancik, K. & Ghinita, G. 2011. The optimization of situational awareness for insider threat detection. In *Proceedings of the First ACM Conference on Data and Application Security and Privacy*, CODASPY '11, 231–236, New York, NY, USA. ACM. URL: http://doi.acm.org/10.1145/1943513.1943544, doi:10.1145/1943513.1943544.

[34] Yen, J., McNeese, M., Mullen, T., Hall, D., Fan, X., & Liu, P. *RPD-based Hypothesis Reasoning for Cyber Situation Awareness*, 39–49. Springer US, Boston, MA, 2010. URL: https://doi.org/10.1007/978-1-4419-0140-8_3, doi:10.1007/978-1-4419-0140-8_3.

[35] Deng, Y. & Shukla, S. 2013. A distributed real-time event correlation architecture for scada security. In *Critical Infrastructure Protection VII*, Butts, J. & Shenoi, S., eds, 81–93, Berlin, Heidelberg. Springer Berlin Heidelberg.

[36] Bahşi, H. & Maennel, O. M. 2015. A conceptual nationwide cyber situational awareness framework for critical infrastructures. In *Secure IT Systems*, Buchegger, S. & Dam, M., eds, 3–10, Cham. Springer International Publishing.

[37] Pournouri, S., Akhgar, B., & Bayerl, P. S. 2016. Cyber attacks analysis using decision tree technique for improving cyber situational awareness. In *Global Security, Safety and Sustainability - The Security Challenges of the Connected World*, Jahankhani, H., Carlile, A., Emm, D., Hosseinian-Far, A., Brown, G., Sexton, G., & Jamal, A., eds, 155–172, Cham. Springer International Publishing.

[38] Vaarandi, R. Oct 2002. Sec - a lightweight event correlation tool. In *IEEE Workshop on IP Operations and Management*, 111–115. doi:10.1109/IPOM.2002.1045765.

[39] Morin, B., Mé, L., Debar, H., & Ducassé, M. 2002. M2d2: A formal data model for ids alert correlation. In *Recent Advances in Intrusion Detection*, Wespi, A., Vigna, G., & Deri, L., eds, 115–137, Berlin, Heidelberg. Springer Berlin Heidelberg.

[40] National vulnerability database general information. https://nvd.nist.gov/general. Last accessed: 2019-05-11.

[41] Chen, L. & Aritsugi, M. 2006. An svm-based masquerade detection method with online update using co-occurrence matrix. In *Detection of Intrusions and Malware & Vulnerability Assessment*, Büschkes, R. & Laskov, P., eds, 37–53, Berlin, Heidelberg. Springer Berlin Heidelberg.

[42] Wei, H. 2013. A correlation analysis method for network security events. In *Informatics and Management Science III*, Du, W., ed, 269–277, London. Springer London.

[43] Sadighian, A., Fernandez, J. M., Lemay, A., & Zargar, S. T. 2014. Ontids: A highly flexible context-aware and ontology-based alert correlation framework. In *Foundations and Practice of Security*, Danger, J. L., Debbabi, M., Marion, J.-Y., Garcia-Alfaro, J., & Zincir Heywood, N., eds, 161–177, Cham. Springer International Publishing.

[44] O'Connor, M. & Das, A. 2009. Sqwrl: A query language for owl. In *Proceedings of the 6th International Conference on OWL: Experiences and Directions - Volume 529*, OWLED'09, 208–215, Aachen, Germany, Germany. CEUR-WS.org. URL: http://dl.acm.org/citation.cfm?id=2890046.2890072.

[45] Wang, W., Jiang, R., Jia, Y., Li, A., & Chen, Y. 2017. Kgbiac: Knowledge graph based intelligent alert correlation framework. In *Cyberspace Safety and Security*, Wen, S., Wu, W., & Castiglione, A., eds, 523–530, Cham. Springer International Publishing.

[46] Cheng, W., Chuang, T., Yang, C., Lin, Y., Liu, M., & Yin, C. Sep. 2017. An integrated security monitoring system for digital service network devices. In *2017 19th Asia-Pacific Network*

Operations and Management Symposium (APNOMS), 118–122. doi:10.1109/APNOMS.2017.8094189.

[47] Husák, M., Kašpar, J., Bou-Harb, E., & Čeleda, P. 2017. On the sequential pattern and rule mining in the analysis of cyber security alerts. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ARES '17, 22:1–22:10, New York, NY, USA. ACM. URL: http://doi.acm.org/10.1145/3098954.3098981, doi:10.1145/3098954.3098981.

[48] Intrusion detection extensible alert. https://idea.cesnet.cz/en/index. Last accessed: 2019-05-12.

[49] Lu, X., Han, J., Ren, Q., Dai, H., Li, J., & Ou, J. Sep 2018. Network threat detection based on correlation analysis of multi-platform multi-source alert data. *Multimedia Tools and Applications*. URL: https://doi.org/10.1007/s11042-018-6689-7, doi:10.1007/s11042-018-6689-7.

[50] Kenaza, T., Machou, A., & Dekkiche, A. 2018. Implementing a semantic approach for events correlation in siem systems. In *Computational Intelligence and Its Applications*, Amine, A., Mouhoub, M., Ait Mohamed, O., & Djebbar, B., eds, 648–659, Cham. Springer International Publishing.

[51] Musen, M. A. June 2015. The protÉgÉ project: A look back and a look forward. *AI Matters*, 1(4), 4–12. URL: http://doi.acm.org/10.1145/2757001.2757003, doi:10.1145/2757001.2757003.

[52] Introduction to stix. https://oasis-open.github.io/cti-documentation/stix/intro. Last accessed: 2019-04-10.

[53] Oasis cyber threat intelligence (cti) tc. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=cti. Last accessed: 2019-04-10.

[54] Oasis standards stix. https://www.oasis-open.org/standards#stix2.0. Last accessed: 2019-04-10.

[55] Stix part 5 patterning. http://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part5-stix-patterning.html. Last accessed: 2019-04-10.

[56] Stix using granular markings. https://oasis-open.github.io/cti-documentation/examples/using-granular-markings.html. Last accessed: 2019-04-12.

[57] Stix 4.1.4 tlp marking object type. https://docs.google.com/document/d/1dIrh1Lp3KAjEMm8o2VzAmuV0Peu-jt9aAh1IHrjAroM/pub#h.yd3ar14ekwrs. Last accessed: 2019-04-12.

[58] Us cert cisa traffic light protocol (tlp) definitions and usage. https://www.us-cert.gov/tlp. Last accessed: 2019-04-12.

[59] Introduction to taxii. https://oasis-open.github.io/cti-documentation/taxii/intro. Last accessed: 2019-04-11.

[60] Veris, the vocabulary for event recording and incident sharing. http://veriscommunity.net/index.html. Last accessed: 2019-04-07.

[61] Getting started with veris. http://veriscommunity.net/howto.html. Last accessed: 2019-04-11.

[62] Veris overview. http://veriscommunity.net/veris-overview.html. Last accessed: 2019-04-11.

[63] Veris impact assessment. http://veriscommunity.net/impact.html. Last accessed: 2019-04-11.

[64] Rutkowski, A., Kadobayashi, Y., Furey, I., Rajnovic, D., Martin, R., Takahashi, T., Schultz, C., Reid, G., Schudel, G., Hird, M., & Adegbite, S. October 2010. Cybex: The cybersecurity information exchange framework (x.1500). *SIGCOMM Comput. Commun. Rev.*, 40(5), 59–64. URL: http://doi.acm.org/10.1145/1880153.1880163, doi:10.1145/1880153.1880163.

[65] Vakilinia, I., Tosh, D. K., & Sengupta, S. July 2017. Privacy-preserving cybersecurity information exchange mechanism. In *2017 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, 1–7. doi:10.23919/SPECTS.2017.8046783.

[66] National Science Foundation (NSF). Cici: Ce: Implementing cybex-p: Helping organizations to share with privacy preservation. https://www.nsf.gov/awardsearch/showAward?AWD_ID=1739032&HistoricalAwards=false. Last accessed: 2019-04-21.

[67] Sadique, F., Bakhshaliyev, K., Springer, J., & Sengupta, S. Jan 2019. A system architecture of cybersecurity information exchange with privacy (cybex-p). In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 0493–0498. doi:10.1109/CCWC.2019.8666600.

[68] D'Amico, A. & Salas, S. April 2003. Visualization as an aid for assessing the mission impact of information security breaches'. In *Proceedings DARPA Information Survivability Conference and Exposition*, volume 2, 190–195 vol.2. doi:10.1109/DISCEX.2003.1194964.

[69] and W. Yurcik, , Lakkaraju, K., & Abad, C. April 2004. Visflowconnect: providing security situational awareness by visualizing network traffic flows. In *IEEE International Conference on Performance, Computing, and Communications, 2004*, 601–607. doi:10.1109/PCCC.2004.1395108.

[70] Abdullah, K., Lee, C. P., Conti, G., Copeland, J. A., & Stasko, J. Oct 2005. Ids rainstorm: visualizing ids alarms. In *IEEE Workshop on Visualization for Computer Security, 2005. (VizSEC 05).*, 1–10. doi:10.1109/VIZSEC.2005.1532060.

[71] and C. Lee, Conti, G., & Copeland, J. A. June 2005. Visualizing network data for intrusion detection. In *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*, 100–108. doi:10.1109/IAW.2005.1495940.

[72] Koike, H., Ohno, K., & Koizumi, K. Oct 2005. Visualizing cyber attacks using ip matrix. In *IEEE Workshop on Visualization for Computer Security, 2005. (VizSEC 05).*, 91–98. doi:10.1109/VIZSEC.2005.1532070.

[73] Goodall, J. R., Ozok, A. A., Lutters, W. G., Rheingans, P., & Komlodi, A. 2005. A user-centered approach to visualizing network traffic for intrusion detection. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, 1403–1406, New York, NY, USA. ACM. URL: http://doi.acm.org/10.1145/1056808.1056927, doi:10.1145/1056808.1056927.

[74] Pearlman, J. & Rheingans, P. *Visualizing Network Security Events Using Compound Glyphs from a Service-Oriented Perspective*, 131–146. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. URL: https://doi.org/10.1007/978-3-540-78243-8_9, doi:10.1007/978-3-540-78243-8_9.

[75] Mukosaka, S. & Koike, H. Feb 2007. Integrated visualization system for monitoring security in large-scale local area network. In *2007 6th International Asia-Pacific Symposium on Visualization*, 41–44. doi:10.1109/APVIS.2007.329273.

[76] Makanju, A., Brooks, S., Zincir-Heywood, A. N., & Milios, E. E. Oct 2008. Logview: Visualizing event log clusters. In *2008 Sixth Annual Conference on Privacy, Security and Trust*, 99–108. doi:10.1109/PST.2008.17.

[77] Shneiderman, B. January 1992. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1), 92–99. URL: http://doi.acm.org/10.1145/102377.115768, doi:10.1145/102377.115768.

[78] Bond, T. 2009. Visualizing firewall log data to detect security incidents. *GIAC Certifications*. URL: https://www.giac.org/paper/gcia/1651/visualizing-firewall-log-data-detect-security/109883.

[79] Fink, G. A., North, C. L., Endert, A., & Rose, S. Oct 2009. Visualizing cyber security: Usable workspaces. In *2009 6th International Workshop on Visualization for Cyber Security*, 45–56. doi:10.1109/VIZSEC.2009.5375542.

[80] Langton, J. T. & Baker, A. June 2013. Information visualization metrics and methods for cyber security evaluation. In *2013 IEEE International Conference on Intelligence and Security Informatics*, 292–294. doi:10.1109/ISI.2013.6578846.

[81] Balakrishnan, B. December 2015. Metrics and visualization. *SANS Reading Room*. URL: https://www.sans.org/reading-room/whitepapers/metrics/paper/36387.

[82] 2019. Iec 61850:2019 ser series. *International Electrotechnical Commission*. https://webstore.iec.ch/publication/6028.

[83] Knapp, E. D. & Langill, J. T. 2015.

[84] plcopen. Introduction into iec 61131-3 programming languages. http://www.plcopen.org/pages/tc1_standards/iec_61131_3/.

[85] How do you manage your industrial time series and a&e data? https://www.ge.com/digital/applications/historian. Last accessed: 2019-05-22.

[86] Specification, P. 2015. Overview and guidance for profinet specifications. https://www.profibus.com/download/profinet-specification/.

[87] Comission, I. E. April 2019. Iec 61158-1:2014. https://webstore.iec.ch/publication/59890.

[88] Comission, I. E. Aug 2014. Iec 61784-1:2014. https://webstore.iec.ch/publication/5878.

[89] Overview of dnp3 protocol. https://www.dnp.org/About/Overview-of-DNP3-Protocol. Last accessed: 2019-05-22.

[90] The art of war. https://ctext.org/art-of-war/attack-by-stratagem. Last accessed: 2019-04-01.

[91] Karen Scarfone, Murugiah Souppaya, A. C. A. O. 2008. Sp 800-115 technical guide to information security testing and assessment. *NIST Computer Security Resource Center*.

[92] Cima, S. 2001. Vulnerability assessment. *SANS Reading Room*. https://www.sans.org/reading-room/whitepapers/basics/vulnerability-assessment-421.

[93] Simpson, N. 2001. Guidelines for developing penetration rules of behavior. *SANS Institute InfoSec Reading Room*.

[94] The cyber kill chain. https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html. Last accessed: 2019-03-25.

[95] The industrial control systemcyber kill chain. https://www.sans.org/reading-room/whitepapers/ICS/industrial-control-system-cyber-kill-chain-36297. Last accessed: 2019-03-25.

[96] OWASP. Penetration testing methodologies. https://www.owasp.org/index.php/Penetration_testing_methodologies.

[97] ISO. 2015. Information technology, securing techniques, network security. *International Organization for Standardization*. https://www.iso.org/isoiec-27001-information-security.html.

[98] ISO. 2017. Information security controls for the energy utility sector. *International Organization for Standardization*. https://www.iso.org/standard/68091.html.

[99] Engebretson, P. 2013. *The Basics of Hacking and Penetration Testing: ethical hacking and penetration testing made easy*. Elsevier, 2nd edition, Page 16.

[100] Russinovich, M. 2014. Disk2vhd v2.01. https://docs.microsoft.com/en-us/sysinternals/downloads/disk2vhd. Last accessed: 2019-02-10.

[101] Offensive Security. Google hacking database (ghdb). https://www.offensive-security.com/community-projects/google-hacking-database/. Last accessed: 2019-02-10.

[102] Controlthings i/o. https://www.controlthings.io/. Last accessed: 2019-02-11.

[103] MITRE. Device hardening. http://www2.mitre.org/public/industry-perspective/documents/04-ex-device-hardening.pdf. Accessed: 2019-02-11.

[104] October 2012. Iso/iec 27037:2012 information technology – security techniques – guidelines for identification, collection, acquisition and preservation of digital evidence. *International Organization for Standardization*. https://www.iso.org/standard/44381.html?browse=tc.

[105] Årnes, A. May 2017. Digital forensics. *John Wiley & Sons*.

[106] 2016. Nou 2016: 24 ny straffeprosesslov. https://www.regjeringen.no/no/dokumenter/nou-2016-24/id2517932/.

[107] advanced persistent threat (apt). https://csrc.nist.gov/glossary/term/advanced-persistent-threat. Last accessed: 2019-05-24.

[108] Spyridopoulos, T., Tryfonas, T., & May, J. Oct 2013. Incident analysis amp; digital forensics in scada and industrial control systems. In *8th IET International System Safety Conference incorporating the Cyber Security Conference 2013*, 1–6. doi:10.1049/cp.2013.1720.

[109] Stirland, J., Jones, K., Janicke, H., Wu, T., & Publications, S. 10 2014. Developing cyber forensics for scada industrial control systems. URL: https://www.researchgate.net/publication/266477470.

[110] Wu, T., Disso, J. F. P., Jones, K., & Campos, A. 2013. Towards a scada forensics architecture. In *Proceedings of the 1st International Symposium on ICS & SCADA Cyber Security Research 2013*, ICS-CSR 2013, 12–21, UK. BCS. URL: http://dl.acm.org/citation.cfm?id=2735338.2735340.

[111] van den Bos, J. & van der Storm, T. 2011. Bringing domain-specific languages to digital forensics. In *Proceedings of the 33rd International Conference on Software Engineering*, ICSE '11, 671–680, New York, NY, USA. ACM. URL: http://doi.acm.org/10.1145/1985793.1985887, doi:10.1145/1985793.1985887.

[112] Asmar, R., Beztchi, S., M. Smith, J., Lyles, B., & Prowell, S. 12 2018. Tools, techniques, and methodologies: A survey of digital forensics for scada systems. 1–8. doi:10.1145/3295453.3295454.