A Systems Theoretic Approach to the Security Threats in Cyber Physical Systems: Applied to Stuxnet

Arash Nourian
Stuart Madnick

Working Paper CISL# 2014-13

September 2014

Composite Information Systems Laboratory (CISL)
Sloan School of Management, Room E62-422
Massachusetts Institute of Technology
Cambridge, MA 02142

A Systems Theoretic Approach to the Security Threats in Cyber Physical Systems Applied to Stuxnet

Arash Nourian and Stuart Madnick, *Member, IEEE*

***Abstract*—The emerging smart technologies, while benefiting customers and companies, also provides adversaries including insiders with powerful tools to affect the physical world. Using traditional IT systems in cyber physical systems (CPS) unfortunately provides potential attackers with many new opportunities to disrupt the services provided by CPSs. In this paper, we examine Stuxnet and utilize a system-theoretic approach taking both physical and cyber components into account to address the threats posed by Stuxnet. We show how such approach is capable of identifying cyber threats geared towards CPSs and provide practical recommendations that can be utilized by CPS designers in building a secure CPS.**

***Index Terms*—security of cyber physical systems, Stuaxnet, CPS**

1. Introduction

THE increased challenges of today's life such as energy scarcity, require the integration of computing intelligence into physical world. Cyber physical systems (CPS) [1] such as industrial control systems are examples of such integration where the effects on physical world are controlled through the use of smart technologies created by computers [2].

In recent years, most of the computing systems used in CPSs are based on commercial-of-the-shelf (COTS) components. COST systems integration not only provide fine grained level of control but also improve reliability and lower deployment and operational costs in comparison to the traditional vendor-specific proprietary and closed-source systems.

However, the rapid growth of using COTS products and IT-based systems in CPSs, have made CPSs more available target for attackers [3]. Attackers can take advantage of vulnerabilities in COTS to take control of a CPS. With a physical manifestations in the real world, attacks on CPSs can cause disruption to physical services or create a national disaster. As a cyber physical system requires a tight coupling between the physical and cyber controlling components, it is crucial to ensure that the system is secure for all the cyber and physical processes. Therefore, protecting the CPSs' against cyber attacks is of paramount importance.

Traditional IT security methods can be applied to protect a CPS, such as a critical infrastructure system, against cyber threats or threats imposed by malicious insiders. However, due to the unique characteristics of a CPS, traditional IT security strategies and approaches are not sufficient enough to address the security challenges of a CPS [3] [4] [5] [6] [7] [8]. For example, installing security patches or numerous system updates that require taking the system offline is difficult, not economically justifiable, and often not feasible. Also, new updates or security patches may create other problems such as in a case where a nuclear power plant accidentally was shutdown after a software update [9]. Recently, it has been shown that attackers can take control of air planes by having access to Wi-Fi services provided by the planes [10].

Traditionally, industrial control systems were considered secured as long as they are air-gapped, not connected to outside world. This notion is not valid anymore as more and more industrial control systems are connecting to outside of their perimeter for vari-

A. Nourian and S. Madnick are with the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA.
E-mails: nourian@mit.edu, smadnick@mit.edu

ous reasons such as providing better services similar to smart grids or updating their softwares. However, having a direct connection to outside world is not necessary to make a CPS vulnerable to cyber attacks. Cases like Stuxnet has shown that even without direct connections to outside cyber world, cyber physical systems are still vulnerable.

In this paper, we utilize a system theoretic framework to evaluate and enhance the security of CPSs. The framework can be used in CPS attack modeling and threat assessment as well as diagnosis methods for stealthy attacks against a CPS. We evaluate the effectiveness of our proposed framework in terms of finding vulnerabilities and protecting a CPS by applying it on the Stuxnet case.

The rest of the paper is organized as follows. Section 2 discusses the traditional approaches for evaluating safety and security in CPSs. In section 3, we review how Stuxnet works and infects the CPSs. Section 4 contains a thorough application of proposed security analysis scheme on Stuxnet. Section 5 summarizes the results of our analysis.

2. Related Techniques for Safety and Security Analysis in CPS

Traditionally, several approaches are available for safety analysis in CPS. Among the most popular ones are Fault Tree Analysis (FTA) [11], Failure Mode and Effects Analysis (FMEA), Hazard Analysis and Critical Control Points (HACCP), and Hazard and Operability Study (HAZOP) [11] [12].

Although traditional approaches provide somewhat effective way to address and analyze the safety and security of a complex systems, they fail to consider new issues in modern complex systems such as numerous interactions among different components, heterogeneity of the networks, and cyber connections.

FTA and FEMA methodologies use the decomposition approach on safety and security. One of the issue of this approach is that it assumes any failure is the result of a linear chain of undesired events that are caused from a single random component failure. However, most security threats in CPS happens when the system is compromised without any evident failure. For example, due to lack of authentication for control parameter modifications, an attacker is able to modify the control parameters within the safe range. In this case, no failure happens but the system's security is compromised.

Another issue with the traditional approaches is that they consider safety or security as a reliability issue. For example, they consider an absence of failure in systems as a sign of safe or secure system. However, they system can be under attack without any sign of component or system failure [13].

Furthermore, none of these traditional techniques are geared towards addressing the security of a CPS since they consider individual components but not the interaction among components in addressing the safety of a CPS. In addition, since these approaches are mainly designed for safety analysis, they can not be used effectively to address the security concerns in a CPS as safety and security are different in nature. A system may be safe but not secure. For example. a system can allow unauthorized modifications of the control parameters within the safe range, creating undesirable output.

Recently, a new system based approach, Systems Theoretic Accident Model and Process (STAMP) [13] has been introduced to address the need for an effective approach for addressing security in complex systems such as a CPS.

A. System Theoretical Accident Model and Process (STAMP)

The System Theoretical Accident Model and Process (STAMP) is a new system-based approach to safety and security. Figure 1 shows the STAMP model modules. The fundamental differences between STAMP and other traditional approaches is that STAMP looks at systems as dynamic systems rather than static and consider safety and security of a system as a control problem not a reliability issue.

According to STAMP, the individual components inside a system require control by enforcing a set of constraints. STAMP assumes that the inadequate enforcement of the required constraints in all levels

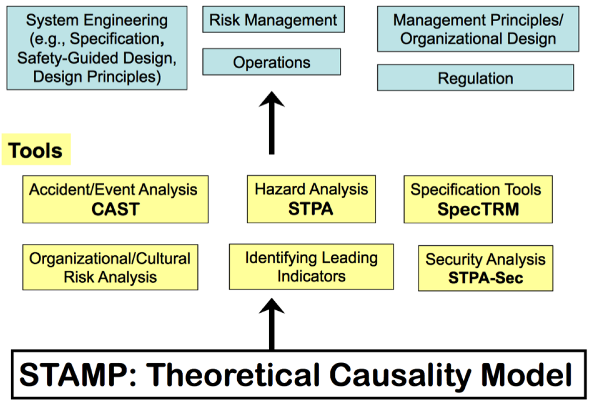

System Engineering (e.g., Specification, Safety-Guided Design, Design Principles)
Risk Management
Operations
Management Principles/ Organizational Design
Regulation
Tools
Accident/Event Analysis CAST
Hazard Analysis STPA
Specification Tools SpecTRM
Organizational/Cultural Risk Analysis
Identifying Leading Indicators
Security Analysis STPA-Sec
STAMP: Theoretical Causality Model

Figure 1: Modules of STAMP model [13]

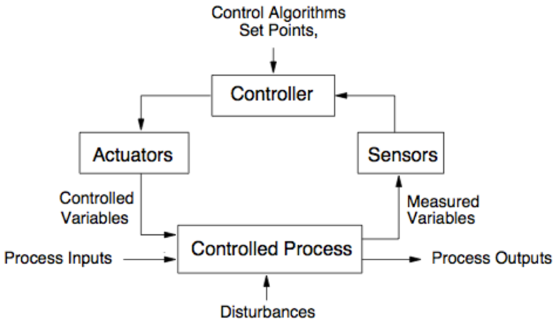


Figure 2: Simple Control loop [13]

including design and development can lead to a failure or an accident. Any undesired events that lead to system failure without component failure or miss interactions among components are called an accident in STAMP.

STAMP analyzes the hierarchical control structure by monitoring how the contextual control structures (i.e. all control structures in different system levels) interact to have a safe and secure state. STAMP analysis helps in finding the mitigations of the detected unsafe state, control loops, and their interactions, which were not possible in the traditional approaches.

Having a holistic system thinking approach that is looking at the whole system and interaction among components rather than just individual isolated component, STAMP also not only allows the analysis of failures and unsafe states but also those that are related to organizational, cyber, and environmental failures. STAMP methodology is based on the following three pillars [13]: (i) safety control structure, (ii) safety constraint, and (iii) process model. Safety control structure represents the hierarchy of all control loops in the system from higher levels to lower levels [13]. Figure 2 shows a standard control loop with Controller, Actuators, Controlled Process, and Sensors as its building blocks. A controller runs the control algorithm for the received commands from the operator or other controllers. The generated command signal changes the state of the controlled process in the actuator. After executing the command, related control variables are sent to the controlled process by the actuator. The sensors verify the system state using the measurement variables and send the result back to the controller. Finally, the controller compares the system state with the desired state and determines the subsequent actions.

Safety constraints are used to identify the safe and unsafe states of a system. They are derived from hazards that are defined in the system specifications. The successful design and enforcement of safety constraint increases system safety. In STAMP, these constraints are used to generate the system requirements that are mandatory to maintain the system safety. Causal Analysis based on STAMP (CAST) [13] is an application of STAMP for accident analysis that we utilize in this paper for the analysis of Stuxnet. The core of CAST is to investigate the control structure dynamics for accident analysis [13]. This investigation begins by looking at safety constraints and show how a constraint violation can lead to a system failure by providing its hierarchical cascading effects on the overall system control structure.

3. Overview of the Stuxnet Case

Stuxnet was first discovered by the VirusBlock-Ada company in June 2010 after they received a request for help from one of their Iranian customers that their Windows-based system was rebooting with the famous blue screen. Further investigation of the problem led to the detection of Stuxnet. Stuxnet

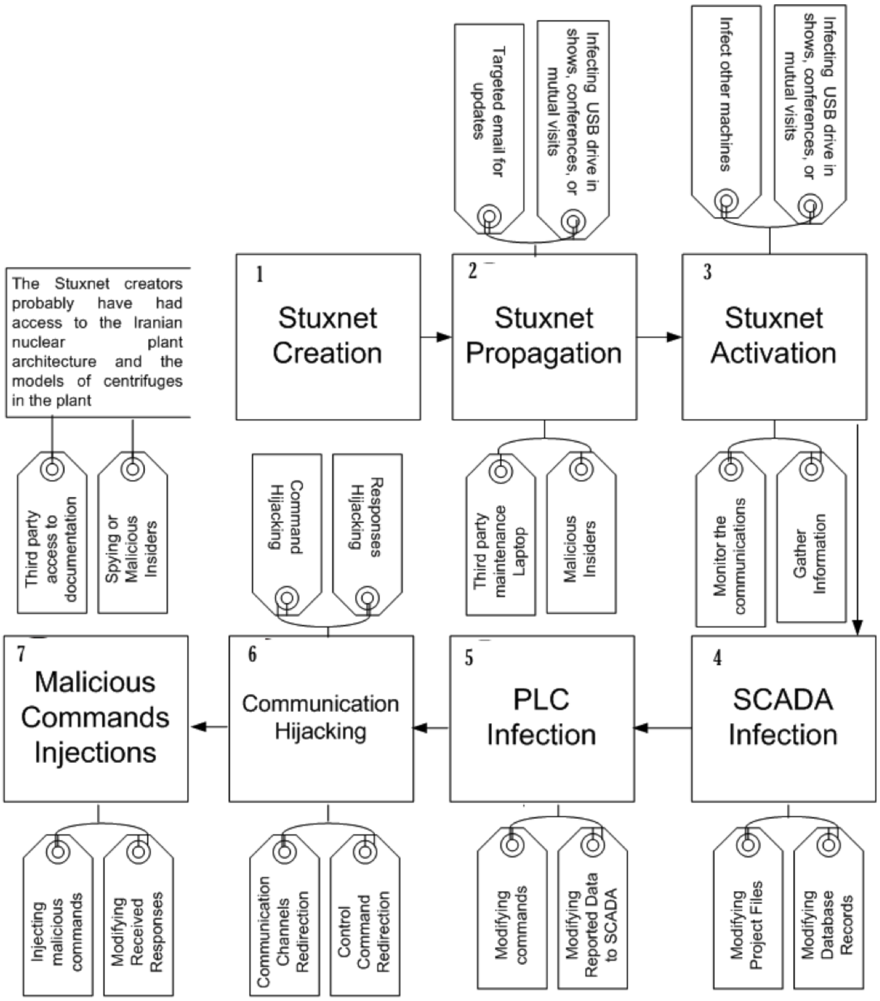

Targeted email for updates
Infecting USB drive in shows, conferences, or mutual visits
Infect other machines
Infecting USB drive in shows, conferences, or mutual visits
The Stuxnet creators probably have had access to the Iranian nuclear plant architecture and the models of centrifuges in the plant
1
Stuxnet Creation
2
Stuxnet Propagation
3
Stuxnet Activation
Third party access to documentation
Spying or Malicious Insiders
Command Hijacking
Responses Hijacking
Third party maintenance Laptop
Malicious Insiders
Monitor the communications
Gather Information
7
Malicious Commands Injections
6
Communication Hijacking
5
PLC Infection
4
SCADA Infection
Injecting malicious commands
Modifying Received Responses
Communication Channels Redirection
Control Command Redirection
Modifying commands
Modifying Reported Data to SCADA
Modifying Project Files
Modifying Database Records

Figure 3: Stuxnet Attack Process (the numbers indicate the step-number in the attack process)

infected computers all around the world. However, the majority of the computers were in Iran [14].

In the design of Stuxnet, several complex techniques have been used, making it one of the most complicated malwares targeting a CPS [15]. The process of infecting, activating, and launching the attack was carefully designed and probably had been tested on similar plant architecture for high degree of effective impact since Stuxnet did not create any damage on other infected uranium enrichment facilties. Figure 3 shows the overall Stuxnet's attack vector both before and after activation.

PLCs are responsible for controlling centrifuges inside a uranium enrichment infrastructure. As each PLC is configured uniquely, the configuration documentations are needed for any type of targeted attacks. In the case of Stuxnet, possible ways of accessing these documents can be either by an insider, third party contractors or even snooping malwares that are designed specifically to gather information about an ICS in order to reverse engineer the actual architecture.

As the targeted uranium enrichment infrastructure were air-gapped, propagation of Stuxnet was probably done via an insider whether through a USB drive or a maintenance laptop. Once the infected USB was connected to the maintenance laptop, Stuxnet was activated and infected all the network devices particularly SCADA (supervisory control and data acquisition) systems, DCS (distributed control system), and PLCs (program logic controller) , sensors/network adapters firmwares, printers, computers, database servers, and application servers. As shown in Figure 3, the original data flow from controllers to centrifuges was modified by the Stuxnet and these modification were not detected by security measures in place.

4. Stuxnet CAST analysis

Traditionally, bottom-up approaches are used to evaluate the safety of a system. However, as discussed in Section 2 some hazards and threats were not identified by standard practices and that caused the breakdown of most centrifuges. This shows why applying a linear traditional approach to a non-linear complex system was not enough. The security of a nonlinear system is not solely directly proportional to the security of individual components. Therefore, a new approach that utilizes a system-thinking approach such as STAMP is required. The intent of our analysis is show whether the STAMP methodology, in particular CAST could have discovered the hazards that led to the centrifuges break down in the Stuxnet case. If those hazards were identifiable using STAMP, its recommended mitigations could have been applied in the design phase to prevent the same hazards to happen in the new or current systems. Also, we show hazards identified by CAST that could not be found by traditional methodologies such as FMEA. Thus, our analysis confirms the advantage of applying a system model in security analysis that can improve the overall safety and security of complex systems.

In CAST each individual component of a complex CPS is analyzed in terms of safety to form a safety perception. Such analysis considers parameters such as incoming data, its source, and interactions with other components inside the operational system. The involved components in the analysis are then linked together to form larger sub-systems until a complete system is formed. However, the interactions between components as depicted in Figure 4 are usually not considered in other approaches, making them insufficient to address the security needs of a CPS. Each link between two components in a loop is labeled with the first letter(s) of the originating component followed by the first letter(s) of the terminating component as shown in Figure 4.

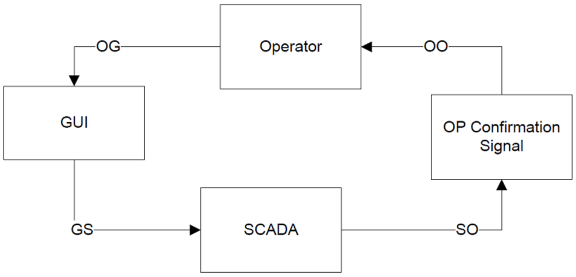


Figure 4: Control loop

In the Stuxnet case the system (i.e. uranium enrichment infrastructure) is operated as follows. The operator may either issue a command to the centrifuges or other controlling components through SCADA or load a predefined operation configuration file that issues the previously defined operations sequences. Once the requested operation is performed within the desirable timeframe, the results are sent back to user for its verification. If the average turn-around time for the requested operation is delayed, then the system may go into a hazardous state.

The system allows the operator to either manually check the correctness of the results or use an automatic verification algorithm that runs a specific simulation for each operation. The algorithm compares the result of simulation with that of the received results for verification purposes. The operator is also able to monitor centrifuges status, PLC's status, as well as other users activities.

After the operator or the automatic verification module verifies the correctness of the requested operation, the system automatically resets itself by performing the required readjustment process for the next new requested operation or the next operation in the sequence.

Traditionally, such a system undergoes serious risk analysis using traditional methodologies such FMEA to not only find the possible hazards caused by the specific system design but also implement the recommended mitigations derived from the analysis [16]. The case system probably had followed the same process as a standard practice recommended for all uranium enrichment infrastructure.

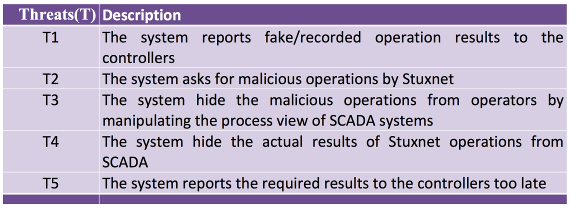
The user interacts with system using the graphical user interface that records the user's commands as well as showing the user the result of its requested operations. Figure 4 shows the typical operation loop in ICS. Lack of properly controlling such a loop as well as other system-wide loops were the main reasons that the Stuxnet attack went through as we show later in this section.

In the Stuxnet case, as described in the previous section, the interactions among operators, SCADA systems, PLCs, and sensors were intercepted and used to launch the malicious operations. As we later show by analyzing all the control loops within the system boundary, lack of authentication and result verification on feedback loops was also evident in the system architecture that made the system vulnerable to threats imposed by Stuxnet.

A. System threat identification

As discussed in Section 2, the first step in CAST is to define the system and hazards related to the accident. The system is the uranium enrichment infrastructure controlled by a set of automated tools such as SCADAs, PLCs, Sensors, and a communication network.

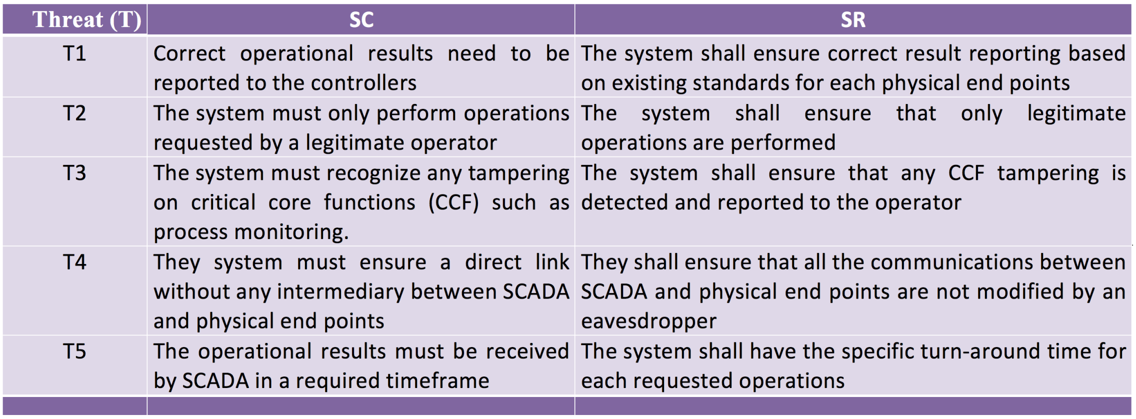
We define threats by extending the definition of hazards in STAMP as explained in Section 2 to consider states that are not hazardous but are undesirable by the users. These states are caused mainly by attackers who circumvent the security measures to execute their control actions with parameters within the safe range. Using the definition of threats and the Stuxnet case analysis discussed in Section 4, most of the relevant threats within the studied system's boundary are listed in Figure 5. These threats are identified based on our analysis of missing controls and the threats posed by Stuxnet. The description of each threats is as follows:



Threats(T)	Description
T1	The system reports fake/recorded operation results to the controllers
T2	The system asks for malicious operations by Stuxnet
T3	The system hide the malicious operations from operators by manipulating the process view of SCADA systems
T4	The system hide the actual results of Stuxnet operations from SCADA
T5	The system reports the required results to the controllers too late

Figure 5: System Threats

1) The T1 threat of reporting fake results to the controllers is highly dangerous that can led to issuing undesired operations from the controllers with a physical manifestation. As discussed in Section 4, the reported fake results to SCADAs led to not recognizing the actual damages to the centrifuges by the operators.
2) T2 is the threat where the system executes the requested operations by Stuxnet rather than that of the operators. Running centrifuges with the highest speed and switching their speed to the lowest speed without considering the speed requested by SCADA or the operator is an example such a threat. These threats are not recognized by the controllers in the system as such attacks hides the actual situation from the controllers, imposing another threat- T3.
3) T3 is the threat where malicious operations such those explained in T2 are concealed from the process view of controllers such as SCADAs. Since the design intent of the system was that always the correct results are available to the SCADAs, no proper controller verification step was used in the original design to address such flaws.
4) T4 is the threat where the whole system was blind on the actual operations that were happening within centrifuges. Usually the actual results are reported by the centrifuge sensors



Threat (T)	SC	SR
T1	Correct operational results need to be reported to the controllers	The system shall ensure correct result reporting based on existing standards for each physical end points
T2	The system must only perform operations requested by a legitimate operator	The system shall ensure that only legitimate operations are performed
T3	The system must recognize any tampering on critical core functions (CCF) such as process monitoring.	The system shall ensure that any CCF tampering is detected and reported to the operator
T4	They system must ensure a direct link without any intermediary between SCADA and physical end points	They shall ensure that all the communications between SCADA and physical end points are not modified by an eavesdropper
T5	The operational results must be received by SCADA in a required timeframe	The system shall have the specific turn-around time for each requested operations

Figure 6: System security requirements and constraints

to SCADAs. The original design intent did not consider result verification and reporting authentication to address this issue.

5) T5, the threat of delayed reporting, was not directly exploited by Stuxnet but the system was susceptible to such a threat by Stuxnet as it was sitting as a middleware between controllers and physical devices, in this case centrifuges and were able to delay the reception of results by SCADAs. This may lead to launching undesired operations by SCADAs due to lack of results.

B. System Security Constraint and Security Requirements

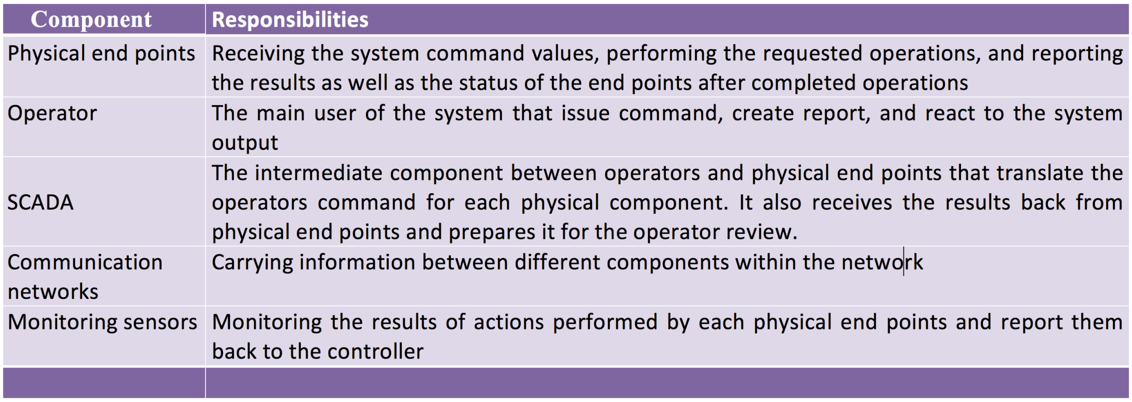
The second step in the CAST analysis is to define the security constraints based on hierarchical control systems. Also, security requirements associated to each security constraints should also be defined to ensure that the security constraints are not violated. The security constraints and security requirements of Stuxnet case is shown in Figure 6

As it is shown in Figure 6, a security constraint is defined for each identified threat shown in Figure 5. For example, for T1, the defined security constraint indicates the receiving of the correct results by the controllers. As mentioned earlier, failure to enforce such constraint led to the T1 in the Stuxnet case. The security requirements that addresses this constraint is to ensure that always the correct results are reported to the controllers. Without the correct results, the operators are blind to the centrifuges status and are unable to react properly as happened in the Stuxnet case. Therefore, there is a need for a controller for result verification from the system level down to the devices to avoid such threats. This security requirement was neither included nor enforced in the original design of the case system.

The centrifuges should spin with a desirable speed requested by PLCs. Therefore, there is a need for a controller that checks whether the desired operations are performed. The security constraint and security requirement associated with such threat (i.e. T2) is shown in Figure 6. The ensuring requirements addresses this threat by making sure that only the legitimate operations are performed. Other security constraint and requirements for other identified threats are also shown Figure 6. The system should be able to identify all operation tampering or communication tampering to avoid T3 or T4. Addressing these threats require immediate intervention undesired damage to the system.

C. System Control Structure

After identifying threats, security constraints and requirements, the next step is to investigate the hierarchical control structure of the system for lack of controls. In the Stuxnet case the physical system is the uranium enrichment infrastructure that needs



Component	Responsibilities
Physical end points	Receiving the system command values, performing the requested operations, and reporting the results as well as the status of the end points after completed operations
Operator	The main user of the system that issue command, create report, and react to the system output
SCADA	The intermediate component between operators and physical end points that translate the operators command for each physical component. It also receives the results back from physical end points and prepares it for the operator review.
Communication networks	Carrying information between different components within the network
Monitoring sensors	Monitoring the results of actions performed by each physical end points and report them back to the controller

Figure 7: System components

to be investigated. The critical components of the case system and their functionalities are shown in Figure 7. It is noteworthy that there are many other components. However, we show only the critical components related to the Stuxnet case.

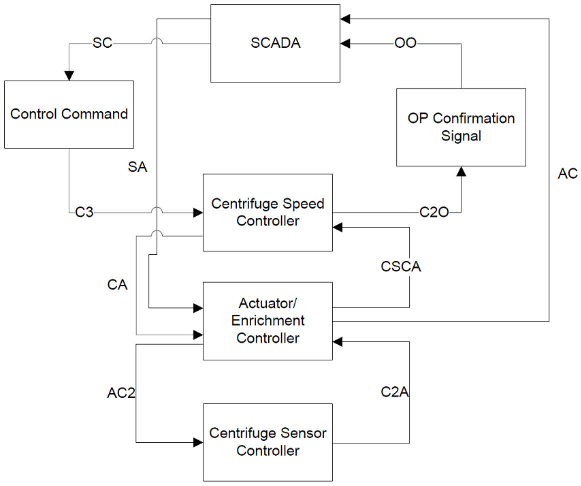

SC
SCADA
OO
Control Command
OP Confirmation Signal
SA
AC
C3
Centrifuge Speed Controller
C2O
CSCA
CA
Actuator/ Enrichment Controller
AC2
C2A
Centrifuge Sensor Controller

Figure 8: Hierarchical Internal Control loops

The system can be decomposed into three core subsystems: the operator subsystem that contains all the user interfaces, control algorithms, and verification systems, the control subsystem that contains all SCADAs, PLCs, and device controllers, and the communication subsystem that contains all network communications among different entities in the system.

The system is complex since it contains numerous components within many layers. Thus, we start by the first control loop at the top level with the operator that is shown earlier in Figure 4. This is the operator control loop that is present in almost all CPS. It shows how the operator interacts with the system. The GUI enables operators to request operations such as centrifuge speed increase, insert initial values, changes centrifuges or PLCs settings, and capture the reported results. The GUI sends the requested commands to SCADA that needs to be preformed. The verification of the requested operations are sent back to the user.

The full control loop is referred to by putting all the labels together. For example, OG-GS-SO-OO refers to the basic control loop showed in Figure 4.

After showing the top level control structure, the components within that structure is further decomposed. In this paper, as an example, we only decompose one of the critical components in the top level that is SCADA. Similar process can be applied to other components as well. The SCADA decomposition in the control structure of the case system is shown in Figure 8. At this level, SCADA becomes a controller for the three lower level controlled processes: Centrifuge speed controller, Enrichment con-

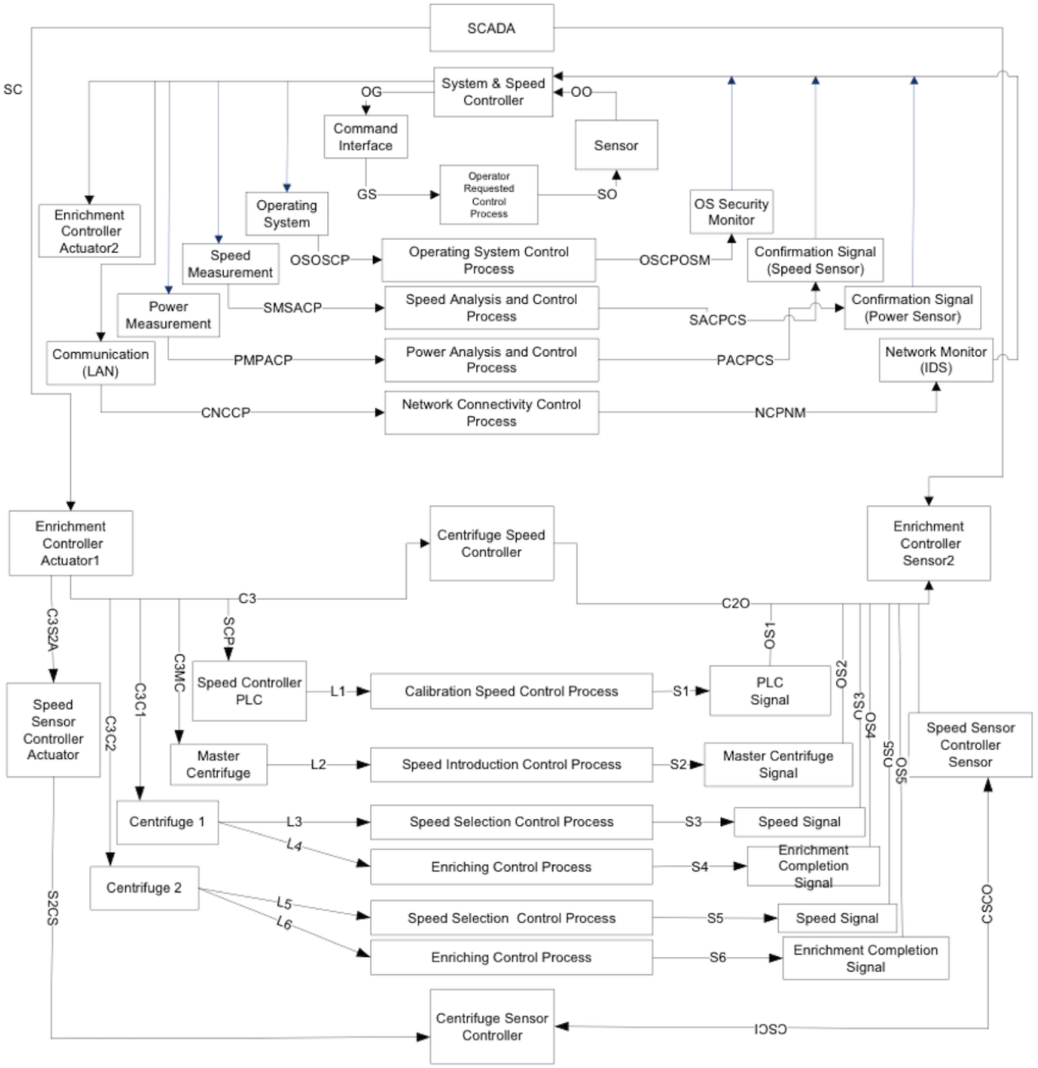
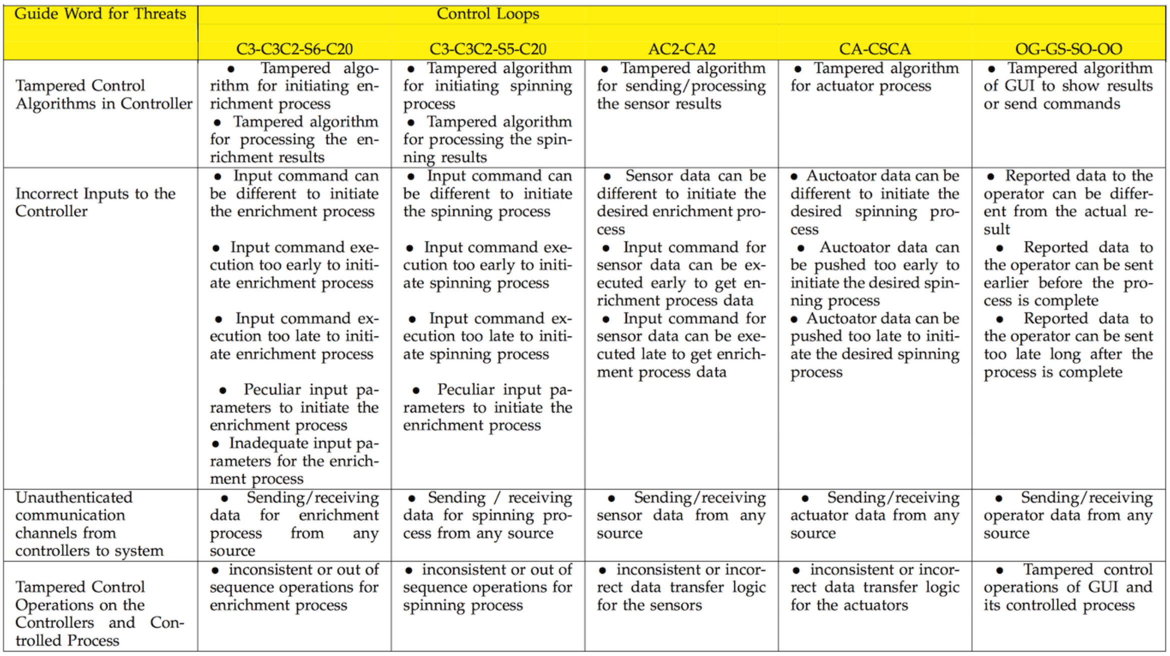

SCADA
SC
System & Speed Controller
OG
OO
Command Interface
Sensor
GS
Operator Requested Control Process
SO
Enrichment Controller Actuator2
Operating System
OS Security Monitor
Speed Measurement
OSOSCP
Operating System Control Process
OSCPOSM
Confirmation Signal (Speed Sensor)
Power Measurement
SMSACP
Speed Analysis and Control Process
SACPCS
Confirmation Signal (Power Sensor)
Communication (LAN)
PMPACP
Power Analysis and Control Process
PACPCS
Network Monitor (IDS)
CNCCP
Network Connectivity Control Process
NCPNM
Enrichment Controller Actuator1
Centrifuge Speed Controller
Enrichment Controller Sensor2
C3
C2O
C3S2A
SCP
C3MC
C3C1
C3C2
OS1
OS2
CS3
OS4
CS5
OS5
Speed Sensor Controller Actuator
Speed Controller PLC
L1
Calibration Speed Control Process
S1
PLC Signal
Speed Sensor Controller Sensor
Master Centrifuge
L2
Speed Introduction Control Process
S2
Master Centrifuge Signal
Centrifuge 1
L3
Speed Selection Control Process
S3
Speed Signal
L4
Enriching Control Process
S4
Enrichment Completion Signal
Centrifuge 2
L5
Speed Selection Control Process
S5
Speed Signal
L6
Enriching Control Process
S6
Enrichment Completion Signal
S2CS
CSCO
Centrifuge Sensor Controller
CSCI

Figure 9: Inter layer system decomposition

troller, and the centrifuge sensor controller. The centrifuge speed controller maintain the desired speed of the centrifuges. The enrichment controller monitors the level of desired enrichment. The centrifuge sensor controllers captures the centrifuges sensor data.

Finally, we decompose the above three controllers to show the interactions among controllers. Figure 9 shows the detailed decomposition of the three critical controllers. As shown in Figure 9, all of these three controllers are interacting with each other creating the final desired operation by the system. Such functional decomposition is critical to identify the lack of control or inadequate control among the critical components that interact with each other. The next step is to investigate the control loops. The main purpose of analyzing control loops is to find vio-



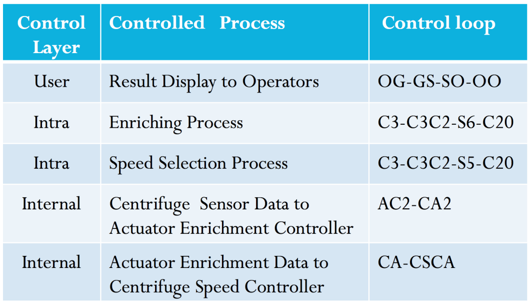
Guide Word for Threats	Control Loops				
	C3-C3C2-S6-C20	C3-C3C2-S5-C20	AC2-CA2	CA-CSCA	OG-GS-SO-OO
Tampered Control Algorithms in Controller	• Tampered algorithm for initiating enrichment process • Tampered algorithm for processing the enrichment results	• Tampered algorithm for initiating spinning process • Tampered algorithm for processing the spinning results	• Tampered algorithm for sending/processing the sensor results	• Tampered algorithm for actuator process	• Tampered algorithm of GUI to show results or send commands
Incorrect Inputs to the Controller	• Input command can be different to initiate the enrichment process • Input command execution too early to initiate enrichment process • Input command execution too late to initiate enrichment process • Peculiar input parameters to initiate the enrichment process • Inadequate input parameters for the enrichment process	• Input command can be different to initiate the spinning process • Input command execution too early to initiate spinning process • Input command execution too late to initiate spinning process • Peculiar input parameters to initiate the enrichment process	• Sensor data can be different to initiate the desired enrichment process • Input command for sensor data can be executed early to get enrichment process data • Input command for sensor data can be executed late to get enrichment process data	• Auctoator data can be different to initiate the desired spinning process • Auctoator data can be pushed too early to initiate the desired spinning process • Auctoator data can be pushed too late to initiate the desired spinning process	• Reported data to the operator can be different from the actual result • Reported data to the operator can be sent earlier before the process is complete • Reported data to the operator can be sent too late long after the process is complete
Unauthenticated communication channels from controllers to system	• Sending/receiving data for enrichment process from any source	• Sending / receiving data for spinning process from any source	• Sending/receiving sensor data from any source	• Sending/receiving actuator data from any source	• Sending/receiving operator data from any source
Tampered Control Operations on the Controllers and Controlled Process	• inconsistent or out of sequence operations for enrichment process	• inconsistent or out of sequence operations for spinning process	• inconsistent or incorrect data transfer logic for the sensors	• inconsistent or incorrect data transfer logic for the actuators	• Tampered control operations of GUI and its controlled process

Figure 10: CAST Results for the Control Loops

lation of security constraints that may be caused by other interacting control loops. Based on the overall control structure and the three decomposition levels as depicted in Figure 4, 8, and 9, the critical control loops that are interacting with each other are in the table shown in Figure 11.

The identified control loops should be investigated for the factors causing the identified threats as shown in Figure 5. In CAST there are several classifications of control loops that can cause unsafe states [13]. Using traditional classifications in CAST and the control loops in the table shown in Figure 11, the threats are listed in Table 10.

The key to the design of Stuxnet was that the malware would be able to interact with the system components as a legitimate entity in the systems. Since the were no component authentication mechanisms in place as evident in Figure 9, Stuxnet took advantage of this design flaw in order to launch its malicious operations. The authentication mechanisms should be in place among each interacting components of Figure 9 to avoid malicious injections of commands or parameters. Once all the core system components are infected, Stuxnet then issues malicious operations from each infected components.



Control Layer	Controlled Process	Control loop
User	Result Display to Operators	OG-GS-SO-OO
Intra	Enriching Process	C3-C3C2-S6-C20
Intra	Speed Selection Process	C3-C3C2-S5-C20
Internal	Centrifuge Sensor Data to Actuator Enrichment Controller	AC2-CA2
Internal	Actuator Enrichment Data to Centrifuge Speed Controller	CA-CSCA

Figure 11: Critical control loops of the system

From Figure 9, we can also notice that the actual sensors results are not passed securely to the controllers since there is no secure channel between sensors and controllers. Therefore, the results can be modified by Stuxnet along the way. There is no controller to check the validity of the results. There can be result verification controller that runs the simulated version of the requested operation and compares the received results with that simulated ones to predicted any tampering with results.

Table 10 shows the 35 threats associated with the control loops in Figure 11. Detailed analysis of control loops and their components can reveal threats that are directly related to the Stuxnet case. 35 potential threats were generated for all the analyzed control loops that most of them were directly related to the Stuxnet case. For example, a contributing factor to T2 can be identified in each of the control loops that is “lack of input verification associated with each operation/process”. Similarly, “Lack of results verification/validation module” is a contributing factor to T1. This could lead to the situation that all the received data can be considered trusted and may have undesired impact on the other interacting control loops. Our analysis shows that STAMP can be useful to identify threats in complex systems that are mainly caused by uncontrolled interactions, something that is missing in the standard practices such as FMEA or FTA.

D. Result Discussion

As it is shown in Table 10, 35 threats were identified based on the analyzed control structure. These threats can be categorized into the following broad categories: (i) lack of control in verifying inputs and outputs for each individual components in the control loops, (ii) lack of control in verifying the source command issuer and destination command received, (iii) lack of control in predicting emerging effects created by the lower-level or upper-level control loops, (iv) lack of control in verifying the authenticity of the software pieces used in system components such as SCADAs, PLCs, and devices’ firmwares, and (v) lack of control in creating secure tunnel for communication between the components in the network

Although sixteen control loops within the system boundary were identified, the five loops that are shown in Figure 11 are the major contributors that had a direct impact to the identified threats. The combination of the identified threats led to the ultimate goal of Stuxnet- disrupting the complete uranium enrichment process. Our CAST analysis found the threats associated with the involved control loops that could be utilized to put required measures to avoid threats imposed by Stuxnet.

As it is shown in Figure 4, the control loop OG-GS-SO-OO, is the highest control loop in the system that requires the correct operation result reported to the operator in order to maintain the correct sequence of operations. Violation of such constraint can be led to undesired operations. Therefore, having a result verification controller can protect the system against such threat.

As another example, the control loop C3-C3C2-S5-C20 could not detect the malicious speed request coming from an authorized source. An analysis of FMEA could not detect such a threat as a potential threat because based on such analysis as long as a sensor is healthy and works properly (getting the requests and responds to them), the functionality is not disrupted and hence the system could be considered safe. However, such a threat could be identified by CAST and proper mitigations could be placed accordingly. Operation result verification (ORV) at lower-levels can be done easily as the number of involved parties are less in comparison to upper-level control mechanisms, improving the accuracy of final results reported to the operators. In addition such ORV can monitor the physical components’ (such as sensors) integrity and performance.

Additionally, even with the presence of an OVR, there is no verification for the sequence of results reported from lower-level loops to the higher-level loops in the hierarchical control structure. For example, a malware such as Stuxnet can report the results (fake results) to the higher-level control loops before the lower-level control loops could verify the results. Therefore, the higher-control loops take actions based on the received results that are not the actual expected results. This is an example of not defining the appropriate behavior of the system that makes the process model incomplete and it is one of the frequent forms of deficiencies that occurs due to incomplete process model [13]. To address such threats, the process model of the controller should either perform a source verification for any received results by utilizing a light-weighted public/private crypto system or use a secure communication tunnel with its components such as secure socket tunneling protocol (SSTP).

Our CAST analysis facilitated the process of understanding a complex control structure such as a uranium enrichment infrastructure and the relationship among its control loops. As we showed in our analysis, even though some of the threats were the result of insufficient access control at lower-level loops, most of them were the result of inadequate control over the interactions among the system components and their associated control loops.

The lesson learned from our CAST analysis can be used to prevent threats in other CPSs. For example, cars are becoming more intelligent these days and numerous components have to interact with each other to accomplish a task. It is estimated that intelligent cars have as much/more code than a fighter jet in near future [17]. Attacks like Stuxnet can cause the car's motor to overspeed similar to the Iranian centrifuges, creating a catastrophic event. Therefore, system designers can utilize the STAMP framework to identify threats in a complex environment that runs mostly through complex interactions among its numerous components.

5. Conclusions

The design of security for cyber-physical systems must take into account several characteristics common to such systems. Among these are interactions between the cyber and physical environment, distributed management and control, real-time requirements, and geographic distribution. This paper discusses these characteristics and suggests a design analysis approach that better integrates security into the core design of the system. We applied CAST on a sample case study. Numerous threats were identified that highlights some of the missing design requirements pieces needed in the original design intent to avoid security threats imposed by the studied case.

References

[1] N. S. Foundation, "Cyber physical systems," 2014.

[2] R. Poovendran, K. Sampigethaya, S. K. S. Gupta, I. Lee, K. V. Prasad, D. Corman, and J. L. Paunicka, "Special issue on cyber-physical systems [scanning the issue]," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 6–12, 2012.

[3] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: Risk assessment, detection, and response," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '11, 2011.

[4] US-CERT, "Control systems security program," 2008.

[5] V. M. Igure, S. A. Laughter, and R. D. Williams, "Security issues in scada networks," *Computers & Security*, vol. 25, no. 7, pp. 498–506, 2006.

[6] E. Johansson, T. Sommestad, and M. Ekstedt, "Issues of cyber security in scada-systems-on the importance of awareness," in *Electricity Distribution-Part 1, 2009. CIRED 2009. 20th International Conference and Exhibition on*. IET, 2009, pp. 1–4.

[7] H. Christiansson and E. Luiijf, "Creating a european scada security testbed," in *Critical Infrastructure Protection*. Springer, 2007, pp. 237–247.

[8] M. HADLEY, N. Lu, and A. DEBORAH, "Smart-grid security issues," *IEEE Security and Privacy*, vol. 8, no. 1, pp. 81–85, 2010.

[9] B. Krebs, "Cyber incident blamed for nuclear power plant shutdown," 2008.

[10] "Planes are at risk of cyber attack through their wi-fi and entertainment systems, says hacker, prompting fears for aircraft security," http://www.dailymail.co.uk/sciencetech/article-2715964/Cyber-hacker-figured-hack.html, 2014.

[11] U. N. R. Commission, "Nrc: Fault tree handbook (nureg-0492)," 1981.

[12] C. Ericson, *Hazard analysis techniques for system safety*. Wiley-Interscience, 2005.

[13] N. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*. MIT Press, 2011.

[14] "Stuxnet expert: Analysis shows design flaw, not vulnerability sunk siemens," http://threatpost.com/stuxnet-expert-langner-analysis-shows-design-flaw-not-vulnerability-sunk-siemens-011912/76115, 2012.

[15] K. Research, "Kaspersky lab provides its insights on stuxnet worm," 2010.

[16] B. M. Tashjian, "The failure modes and effects analysis as a design tool for nuclear safety systems," *Power Apparatus and Systems, IEEE Transactions on*, vol. 94, no. 1, pp. 97–103, 1975.

[17] D. MCCANDLESS, "Visualization of how many millions of lines of code go into various products," http://www.informationisbeautiful.net/visualizations/million-lines-of-code/, 2013.