Critical Infrastructure Cyber-security Risk Management

Theodoros Spyridopoulos[a,1], Konstantinos Maraslis[b], Theo Tryfonas[b] and George Oikonomou[b]

[a] *University of the West of England*

[b] *University of Bristol*

Abstract. Traditional IT cyber-security risk management methods are based on the evaluation of risks calculated as the likelihood of cyber-security incidents occurring. However, these probabilities are usually estimations or guesses based on past experience and incomplete data. Incorrect estimations can lead to errors in the evaluation of risks that can ultimately affect the protection of the system. This issue is also transferred to methods used in Industrial Control Systems (ICSs), as they are mainly adaptations of such traditional approaches. Additionally, conventional methods fail to adequately address the increasing threat environment and the highly interdependent critical nature of ICSs, while proposed methods by the research community are as yet far from providing a solution. The importance of securely managing ICS infrastructures is growing, as they are systems embedded in critical national infrastructure (e.g. city traffic lights controls) and thus a potentially attractive target for organized cyber-criminals and terrorists. In this Chapter we present a novel approach that combines Stafford Beer's Viable System Model (VSM) with Game Theory in order to develop a risk management process that addresses the above issues. The model we develop provides a holistic, cost-efficient cyber-security solution that takes into account interdependencies of critical components as well as the potential impact of different attack strategies.

Keywords. Industrial Control Systems, Cyber-security, Risk Management, Game Theory, Viable System Model

1. Introduction

Various cyber-security risk management tools have been applied to Industrial Control Systems (ICS), all of which are based on the ISO 27005:2011 structure. Their main objective is to assess and minimize cyber-security risks associated with the operation of the system. However, due to the fact that they are mainly adaptations of Enterprise IT-specific tools there are some inherent deficiencies. In particular, they cannot capture the multitude of interdependencies within the ICS, they may not take into account security costs (impact of applying security to the system), which in ICSs can be significantly high, and also they are based on cyber-attack likelihood estimations [1, 2], for which current information on past ICSs cyber-incidents is not sufficient to provide statistically safe assumptions [3, 4, 5, 6, 7].

In this chapter we develop a cyber-security risk management framework that takes into account the particular needs of an ICS providing the means for an evaluation

[1] Corresponding Author.

process that does not rely on expected frequencies of attacks, like conventional methods. We use the principles of the Viable System Model (VSM) [8] in order to provide a way to evaluate the viability of an ICS exploring dependencies between its components. Our systemic approach towards an assessment of the system's *viability* enables us to take into account more complex attack methods (zero-day attacks, attacks with complex multi-target strategies etc.) that cannot be addressed by traditional risk management techniques or lack of relevant experience. Using Game Theory we identify the defenses that should be applied to the system in order to protect it against an attack regardless of its type, taking into account the security costs and the attack impact. The game results in pure strategies Nash Equilibria (NE) that describe the optimal strategies for both attacker and defender. The strategies described by the NE represent the moves from which if either player deviates they will always get less of a payoff [9].

The chapter is structured as follows: Section 2 provides background knowledge on the VSM. In Section 3 we introduce the proposed ICS risk management model, which is based on the VSM and game theory. Section 4 includes the application of our model to a case study scenario. Thoughts for further work and conclusions are presented in Section 5.

2. Background – The Viable System Model

We chose to utilise Stafford Beer's Viable System Model (VSM) [8] in order to model an ICS from a view that would reveal all those essential functions and components that make the system 'survivable', i.e. maintaining the provision of core capabilities. According to the VSM, every organisation can be divided in three parts, Operational, Management and Environment. Each part is composed of subsystems that interact with each other and through their collective operation they contribute to the viability of the system as a whole (Figure 1). Furthermore, operations (operational units) within the Operational part of the VSM form recursively a VSM by themselves, including a management and an operational part. The existence of a VSM's subsystems along with their interdependencies and exchange of information with the corresponding environment, maintains the viability of the system. In more detail:

System 1 includes the operational units of the VSM. Those are the units that are actually doing something (e.g. a Programmable Logic Controller (PLC) that controls a motor). Each one of them serves a purpose within the system and can exchange data with other operational units or with the environment. Their actions are coordinated by System 2, controlled by System 3 and audited by System 3*.

System 2 manages the coordination of the operational units within System 1. It enables the interconnection of the different elements in such a way as to ensure the harmony of their function. The level of efficiency is then reported to System 3. In ICSs System 2 can be translated into the networking device that interconnects the various operational units and reports the state of the network to the control centre. Possible disruption or destruction of System 2 can disturb the balance within the system.

System 3 controls the units of System 1 and also provides the synergies among them. It both receives and transmits data from and to the operational units in order to control their functions. Its controlling capabilities are based on the input from System 2, System 3* and System 4. It realises the control changes that are based on research made by System 4, decided by System 5 and communicated to System 3 again through System 4. It also receives input from System 2, which, as mentioned earlier, reports the result of the operational units' coordination to it. Lastly, it receives input from System 3* that audits the operational units of the system and informs S3 of their state.

System 3* audits the operational units within System 1. Its purpose is to identify anomalies in their operation, i.e. deviations from the operational plan issued by System 3, and communicate them to System 3 in order to operationalise the corresponding control measures.

System 4 is responsible for the adaptation of the whole system to the ever changing environment. It communicates with the environment capturing its changes and transfers the additional knowledge in the form of external Situational Awareness within the system, communicating it to System 5. It also transfers System 5's decisions to System 3. In an ICS, System 4 can represent the IT security manager.

System 5 represents the upper level of management within the System. It embodies the decision maker who, based on the knowledge gathered from System 4 and the general role of the system in its environment, decides the changes that need to take place and communicates them to System 3 through System 4. It also monitors the homeostasis between System 4 and System 3.

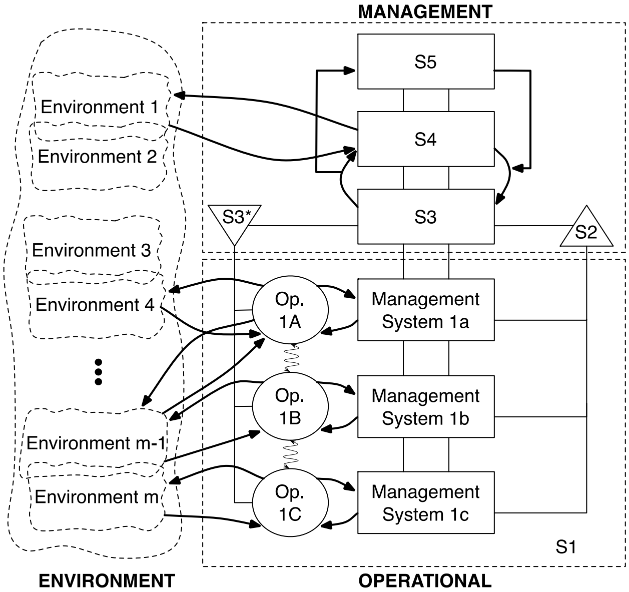


Figure 1. Beer's Viable System Model [9].

3. Proposed Model

Our model approaches an ICS as a VSM. Each component within the system is represented as part of a VSM and has the characteristics and connections that correspond to its particular purpose within the system. Thereby, the system can be modelled as an aggregation of interdependent VSM components with specific characteristics that contribute to the system's viability. The viability of the system is defined through a system of weighted components connected through weighted links.

The weights in both cases reflect the purpose of the element in the VSM or its importance to the system, in other words. Subsequently, we define the strategies of the two adversaries, extending the work of Levitin and Hausken [10], maintaining their systemic nature so that we can provide defence strategies against unknown threats. Eventually, we develop a two-player, zero-sum game with pure strategies. The defender's objective is to minimise the impact of a cyber-attack while minimising the security costs regardless of the attacker's move. This is achieved by following a strategic plan that represents the Nash Equilibrium (NE) of the game.

3.1. The VSM of ICS

Figure 2 presents an architectural block diagram of a typical ICS. The system is divided into three sections. The first part consists of the field devices, including devices used to control mechanical processes or transfer data from and to other devices (e.g. programmable logic controllers, PLCs, that control the speed of a motor, Remote Terminal Units, RTUs, that exert wireless control on operations etc.) The second part forms the control centre, which exerts control on the field devices. It communicates with the field devices and includes operator workstations also known as Human Machine Interfaces (HMIs), data historians, databases etc. Finally, the third part of an ICS represents the outer world with which the system communicates.

In order to show how the various parts of an ICS can collectively form a VSM, we examine a simplified ICS version which includes three operations: 1) the control and monitoring of the speed of a motor, 2) the remote control and monitoring of a waste disposal unit and 3) the voltage control on a specific instrument used within the ICS. Figure 3 shows how this example can be presented as a VSM. As we see, the three operations are managed by a PLC, an RTU, an Intelligent Electronic Device (IED) and their corresponding sensors. Those elements form the components/operational units of System 1. Those components are controlled by the Control Centre which is an aggregation of machines (i.e. data historians, shared resources, HMIs, etc.) that help controlling the various field devices. Therefore, the Control Centre forms System 3. The communication between the Control Centre and the field devices is managed through the control network and the use of a Supervisory Control And Data Acquisition (SCADA) server. Therefore, the SCADA server forms System 2. The HMIs within the Control Centre are responsible for auditing the system. Thus, they play the role of System 3*. Lastly, System 4 and 5 are realised through the Forward Planning Direction of the organisation and the Management Board respectively.

3.2. Measuring Viability

According to the VSM the viability of the system depends on its subsystems; the performance of each subsystem affects the whole system's viability. Therefore, to measure the viability of the system we need to calculate the performance of each element (S2, S3, S3*, S4, S5 and operational units within S1 are considered as elements) within the system identifying (based on the VSM representation in Figure 3) the way interdependencies affect it.

We consider the performance of each element on a scale from 0 (the element has stopped functioning) to x (the higher the value of x the better the performance of the element). Its value depends on the element's functional capability, which refers to

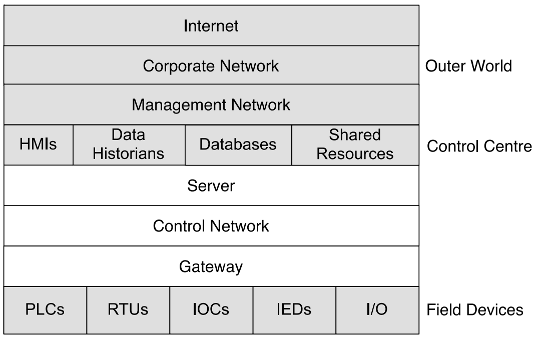


Figure 2. Simplified ICS architecture [9].

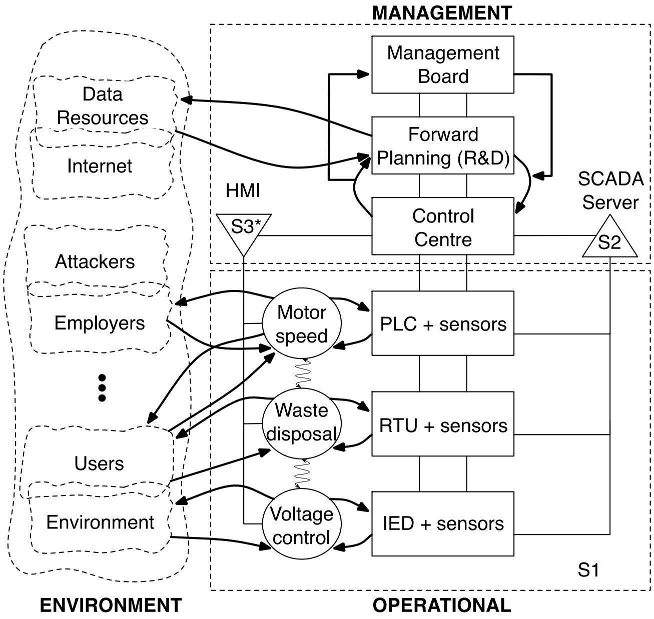


Figure 3. The VSM of an example ICS.

its performance before we take into account interconnections (normally this is equal to 1, “optimal performance”; values less than 1 would indicate some malfunction), and its connection to other elements (each connection is weighed from 0 to 1 according to its significance to the element’s performance).

3.2.1. Performance of Operational Units within System S1

Figure 4 emphasises the interdependencies between an operational unit and the other systems within the VSM (it has to be noted that there is no communication between operational units in this level). A weight is assigned to each connection according to its significance to the unit’s performance. Additionally, a performance value is assigned to each connected system. Based on that, Eq. 1 calculates the performance of the unit taking into account its functional capability, its dependencies on other systems (S2, S3, S3* and the environment) within the VSM and the performance of each connected system.

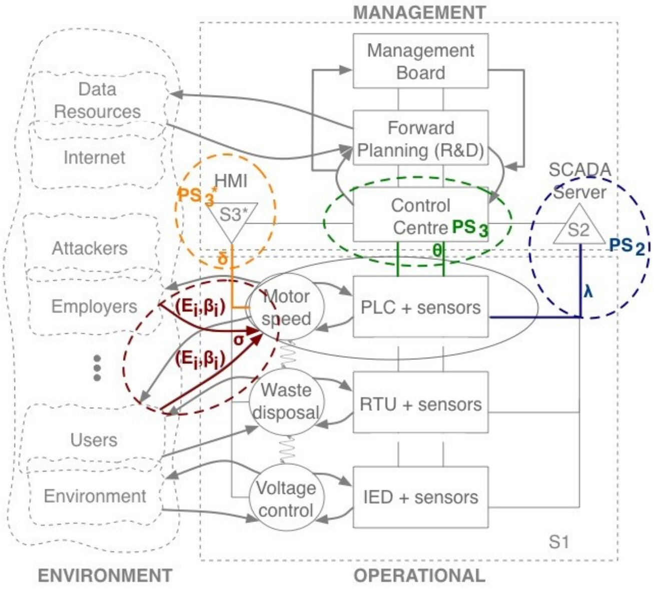


Figure 4. Dependencies of an operational unit.

$$PU = FCU \cdot f\left(\sum_{i=0}^{n}(\beta_i \cdot E_i) \cdot \sigma, \lambda \cdot PS_2, \theta \cdot PS_3, \delta \cdot PS_3^*\right) \tag{1}$$

PU stands for the unit's performance and FCU represents its functional capability. E$_i$ represents the input from environmental groups (under normal conditions this is considered equal to 1; values less than 1 would indicate issues in connection with the environment), β_i is the weight assigned to the specific environmental input, σ is the significance of the total environment to the unit and n is the total number of environmental groups that communicate with the unit. PS$_2$ represents the performance of System 2 that coordinates the unit's communication within the VSM with weight λ, PS$_3$ stands for the performance of System 3 that controls the unit with weight θ and PS$_3$* is the performance of the System 3* that audits the unit with weight δ. The coordination, control, and audit weights denote the significance of those functions to the performance of the unit. All weights take values from 0 to 1.

Taking into account that S3 and S3* cannot communicate with S1 without S2, and also that S3 (for control) and S3* (for monitoring) are essential for S1's operation, Eq. 1 can be simplified in:

$$PU = FCU \cdot \left(\sigma \sum_{i=0}^{n}(\beta_i E_i) + \lambda PS_2 \cdot \theta PS_3 \cdot \delta PS_3^*\right) \tag{2}$$

3.2.2. Performance of System 2

Since System 2 is responsible for the connection of operational units in S1 to the rest of the VSM. Its performance does not depend on other systems; it relies solely on the element's functional capability and is described by Eq. 3

$$PS_2 = FCS_2 \tag{3}$$

where $0 \leq FCS_2 \leq 1$ represents the functional capability of System 2.

3.2.3. Performance of System 3

The ability of System 3 to monitor and control operations within System 1 depends on its connection to S1 as coordinated by S2 and audited by S3*, and its communication with S4. Thus, the performance of System 3 can be described by Eq. 4,

$$PS_3 = FCS_3 \cdot f(\upsilon \cdot PS_2, \omega \cdot PS_3^*, \chi \cdot PS_4) \tag{4}$$

where FCS_3 is the functional capability of System 3, PS_2 corresponds to the performance of System 2, PS_3* corresponds to the performance of the System 3*, PS_4 is the performance of System 4 and υ, ω and χ the respective weights that indicate the systems' significance to S3's performance. All weights take values from 0 to 1.

Taking into account that S2 and S3* are essential for S3's operation (monitoring and control of S1's operations), and that although S4 adds to S3's performance it cannot be considered as vital, we simplify Eq. 4 as:

$$PS_3 = FCS_3 \cdot \upsilon PS_2 \cdot \ \omega PS_3^* \cdot (1 + \ \chi PS_4) \tag{5}$$

3.2.4. Performance of System 3*

System 3* is responsible for auditing operations within S1. Its performance is therefore based on its connection to S1 which is realised through S2. Thus, the performance of System 3* is:

$$PS_3^* = FCS_3^* \cdot f(k \cdot PS_2) \tag{6}$$

where FCS_3* is the functional capability of System 3*, PS*2* the performance of S2 and $0 \leq \kappa \leq 1$ its weight depending on its significance to S3*.

Given the fact that without S2 there is no communication between S3* and S1 Eq. 6 can be simplified in:

$$PS_3^* = FCS_3^* \cdot kPS_2 \tag{7}$$

3.2.5. Performance of System 4

The performance of System 4 depends on its connection to S3, S5 and the environment. Thus it can be calculated as:

$$PS_4 = FCS_4 \cdot f(\alpha \cdot EI, \varepsilon \cdot PS_5) \tag{8}$$

where FCS_4 is the functional capability of System 4, EI represents the interaction with the environment (if such an interaction exists then EI = 1, otherwise EI = 0), PS_5 stands for the performance of System 5 and α and ε are the corresponding weights. All weights take values from 0 to 1.

Given the fact that both S5 and EI are essential for the operation of S4, the equation can be simplified in:

$$PS_4 = FCS_4 \cdot \alpha EI \cdot \varepsilon PS_5 \tag{9}$$

3.2.6. Performance of System 5

System 5's functionality is based on the knowledge it receives from System 4, and therefore its performance that represents its speed of decision is based on S4's functionality as shown in Eq. 10,

$$PS_5 = FCS_5 \cdot f(\mu \cdot PS_4) \tag{10}$$

where FCS_5 is the functional capability of System 5 (we consider this equal to 1 since we do not take into account ill management practices), PS_4 is the performance of System 4 and $0 \leq \mu \leq 1$ the corresponding weights.

Since, in case S4 is missing, S5's decisions cannot be applied in the lower levels of the system the equation can be simplified as:

$$PS_5 = FCS_5 \cdot \mu PS_4 \tag{11}$$

3.2.7. Total Performance

Modelling each element's performance according to their contribution to the VSM gives us an insight into how interconnections affect the performance of the system. From the equations provided above, we can observe that operations within System 1 depend on the performance of each element of the system. To ensure viability we have to ensure that elements (operational units) within S1 perform maximally.

Since the investigation of the effect of ill management and poor planning practices on system performance are not in the scope of this work, we can consider PS_4 and PS_5 to be constant; for simplicity we assume PS_4=PS_5=1. Thus, combining Eq. 2, 3, 5, 7, 9 and 11 we have:

$$PU = FCU \cdot \left(\begin{array}{c} \sigma \sum_{i=0}^{n} (\beta_i E_i) + \lambda FCS_2 \cdot \theta FCS_3 \cdot \upsilon FCS_2 \cdot \omega FCS_3^* \cdot \\ \kappa FCS_2 (1 + \chi) \cdot \delta FCS_3^* \cdot \kappa FCS_2 \end{array} \right) \tag{12}$$

Since PU refers to the performance of one operational unit within S1, and due to the fact that operational units are independent of one another, the total performance can be calculated as:

$$P_{total} = \Phi_1 PU_1 + \Phi_2 PU_2 + \cdots \Phi_k PU_k \tag{13}$$

where k is the total number of operational units and ϕ_k is the importance of each unit to the functionality of the whole system.

3.3. Defining Strategies

In order to overcome likelihood estimations of conventional risk management approaches we use game theory. By deploying a game between the attacker (cyber-threat actor) and the defender (ICS operator), both of which are considered as rational players (i.e. they both want to maximise their payoff taking into account the incurred cost), we can identify strategies for the defender that will return the optimal outcome (here defined as maximum system performance under the minimum cost) regardless of the attacker's strategy.

We consider two types of defence methods for the defender. The first defence method we use is redundancy. In particular, the ICS operator needs to find the elements within the system to which redundancy should be applied in order to maximise the total performance while minimising costs. The cost of redundancy depends on the element (e.g. in an example where legacy systems are used within S1 while S3*, the HMI, has been upgraded with modern systems, the application of redundancy is much easier in S3* than S1). The second type of defence is patching. In the same way as with redundancy, the ICS operator needs to identify the elements within the system which should be patched in order to maximise performance while minimising costs. The cost of patching depends again on the element (e.g. remote, inaccessible operational units and legacy systems are more difficult to patch). By 'patching' we mean an essential software update that mitigates known vulnerabilities.

From the attacker's point of view, the strategies involved depend on the selection of the element that should be compromised (e.g. compromising the SCADA server - S2 - may return a larger payoff compared to compromising a single operational unit within S1) and the complexity of the attack that should be used (e.g. complex Advanced Persistent Threats, APTs, that include previously unseen 'zero-day' attacks, or exploit common vulnerabilities).

3.4. Deploying the Game

Our game is based on Eq. 13 and Eq. 12. For the players' strategies we make the following assumptions:

- Attacks on elements are binary: successful (decrease the element's functional capability to 10%) or unsuccessful (the element's capability is not affected) - see Figure 5-8.
- Attacks can be deployed against multiple elements.
- Available attacks per element: common or zero day (one attack per element; no mixed attacks)
- Available defences per element: redundancy or patching (one defence per element; no mixed defences)
- Patching renders a common attack unsuccessful.
- A zero-day attack is successful against patching.
- Redundancy renders both zero-day attack and common attack unsuccessful.
- The cost of an attack-strategy depends on the number of elements to attack and the type of attack ($\text{Cost}_{\text{zero-day}} > \text{Cost}_{\text{common}}$).
- The cost of a defence strategy depends on the element type (e.g. applying redundancy or patching to S3 may be more costly than applying them to S2) and the type of defence, which in turn depends on the ICS Implementation (e.g. redundancy may seem more costly but patching may require system reboot that - especially in the case of legacy systems - can also be costly).

Figures 5, 6, 7 and 8 provide the attack/defence trees for all elements on which the game is played. It should be noted that Figure 5 represents one element within S1; since we have k elements within S1 (k operational units in S1) we also have k trees similar to Figure 5 (i.e. one for each element/operational unit).

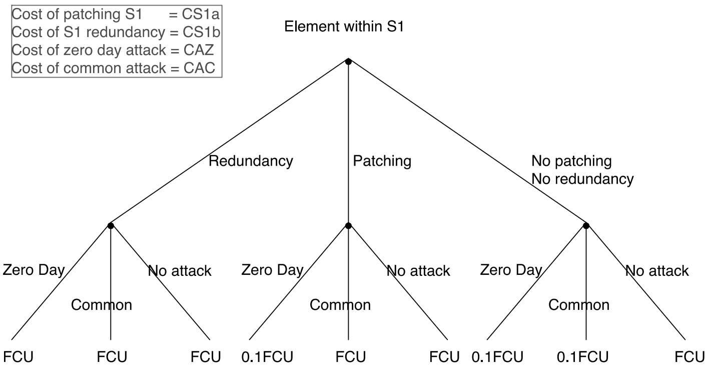


Figure 5. "Attack/Defence on element within S1" tree (one tree for each element within S1).

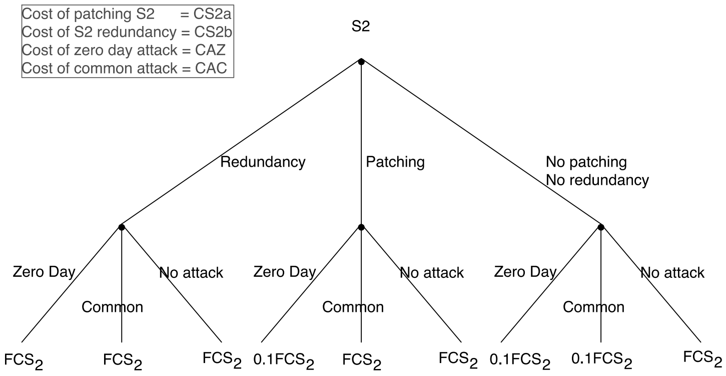


Figure 6. “Attack/Defence on S2” tree.

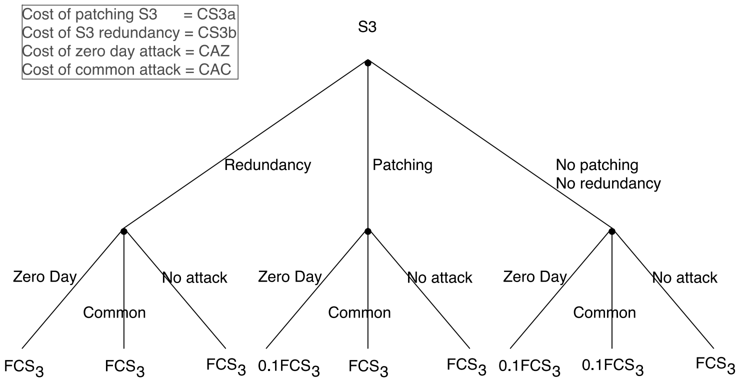


Figure 7. “Attack/Defence on S3” tree.

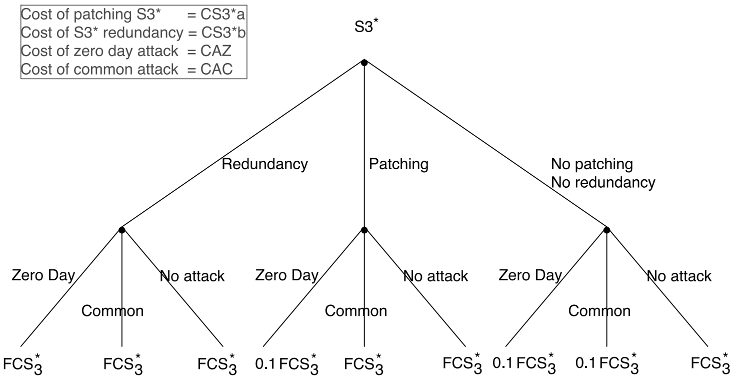


Figure 8 “Attack/Defence on S3∗” tree.

Considering the game as a zero-sum game (since the defender’s loss is the attacker’s gain and vice versa), the players’ payoffs are calculated as:

$$\begin{aligned} DPayoff &= P'_{total} - DCost + ACost \\ APayoff &= -P'_{total} + DCost - ACost \end{aligned} \tag{14}$$

where P'_{total} is calculated based on Eq. 13 and Eq. 12 using the attack/defence trees, $DCost$ is the total cost of defence and $ACost$ is the total cost of attack.

4. Model Application - Case Study

In this section we apply our model to the ICS of Figure 9. As shown, there are six elements to attack/defend, including the PLCs in the field level (that correspond to changes to the FCU_1, FCU_2 and FCU_3), the control server (FCS_2), the HMI (FCS_3^*) and the engineering workstations (FCS_3) within the control center. The game revolves around these elements and depends on the way their functionality changes as a result of the opponents' chosen strategies.

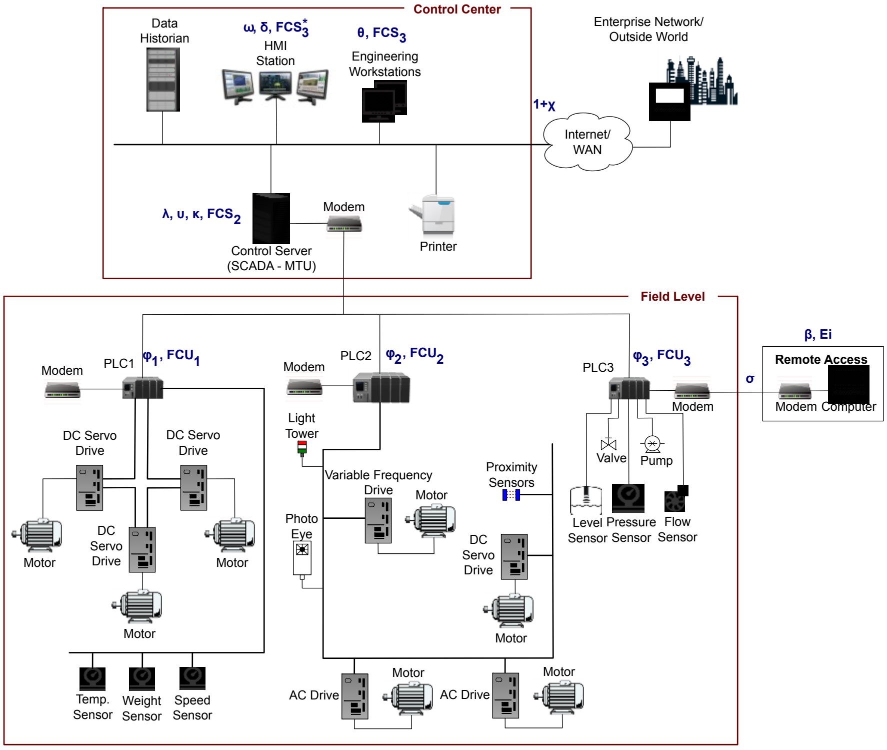


Figure 9. ICS Example

We consider that initially all elements perform optimally ($FCU_1 = FCU_2 = FCU_3 = FCS_2 = FCS_3^* = FCS_3 = 100\%$). Additionally, we assume that the HMI (S_3^*), the workstations (S_3) and the SCADA server (S_2) are equally important to the system ($\omega = \delta = \theta = \lambda = \upsilon = \kappa = \chi = 1$). Furthermore, since there is only one connection of the field level to the environment (remote access to PLC3), which is used for maintenance rather than control purposes, we consider $\sigma = 0.5$ as the weight for the connection to the total environment and $\beta = 1$ as the weight to the remote connection in particular. For the purposes of our illustration we also assess the importance of the field level devices (PLC1, PLC2 and PLC3) based on the processes they control. In particular, PLC2 controls four motors (the highest number of devices compared to the other PLCs), therefore we consider ϕ_2 = 1. PLC1 controls three motors, thus ϕ_1 = 0.9. Finally, since PLC3 controls only one process (the valve) we consider ϕ_3 = 0.7.

In a real world scenario, these values would derive from the asset evaluation process where the ICS operator would identify and assess all system assets.

The most challenging part of the model application is the cost evaluation. As we mentioned in the previous section, the available moves for the defender include patching, redundancy and "no security". The cost for the latter is D_{cost} = 0. However, the cost for the patching and redundancy strategies is based on the ICS implementation and operator's budget, which are difficult to simulate. In our experiment we consider that the ICS operator uses legacy devices (PLCs) in the field level that are difficult to reboot or replace (in some cases legacy devices may not be available on the market) and modern machines within the control center. Thus, the cost of patching or redundancy for the field-level elements is much higher than the cost of securing the elements within the control center. Additionally, in a modern system it is easier to patch than deploy redundancy. In short, we assume the following values for the defender's costs (as shown in Figure 5-8):

- When deploying redundancy for elements within S1 (field-level elements): CS1b = 10^7.
- When patching elements within S1 (field-level elements): CS1a = 10^5.
- When deploying redundancy for the HMI (S3*): CS3*b = 10^4.
- When patching the HMI (S3*): CS3*a = 10^3.
- When deploying redundancy for engineering workstations (S3): CS3b = 10^4.
- When patching the engineering workstations (S3): CS3a = 10^3.
- When deploying redundancy for the SCADA server (S2): CS2b = 10^3.
- When patching the SCADA server (S2): CS2a = 10^2.

From the attacker's point of view there are only two costs, the cost of deploying a zero-day attack and the cost of deploying a common attack. We assume the following values for the attacker's costs:

- Cost of zero-day attack: CAZ = 10^5.
- Cost of common attack: CAC = 10^2.

Based on this information we can now construct the attack/defence trees in Figures 10, 11, 12, 13, 14 and 15.

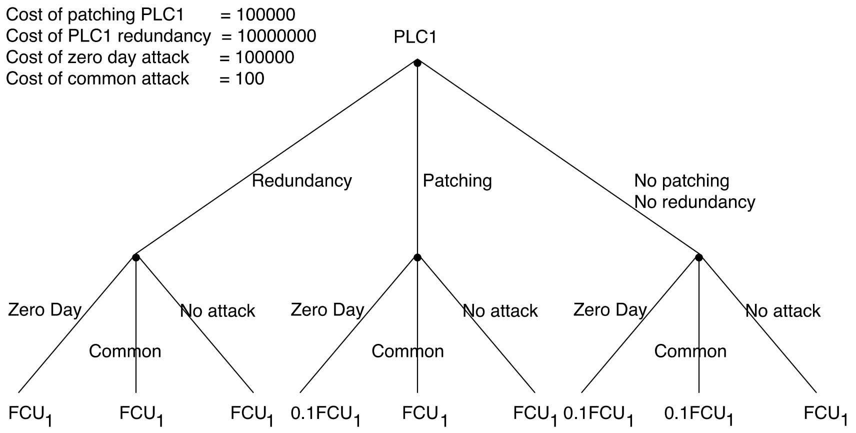


Figure 10. Attack/Defence tree for PLC1

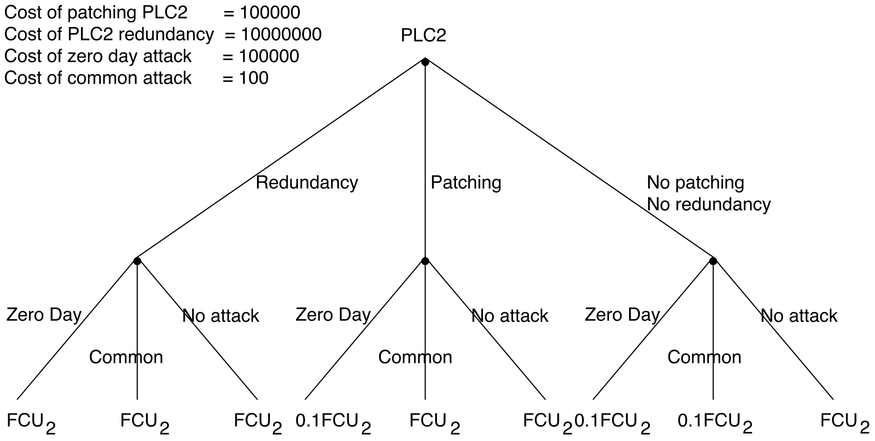


Figure 11. Attack/Defence tree for PLC2

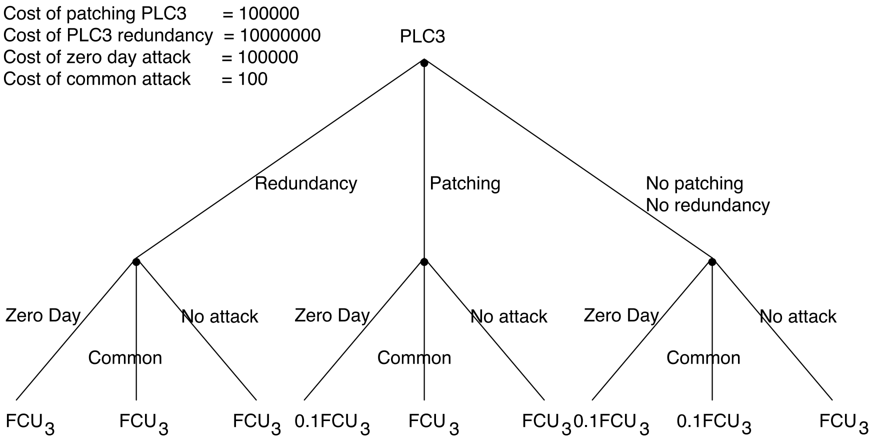


Figure 12. Attack/Defence tree for PLC2

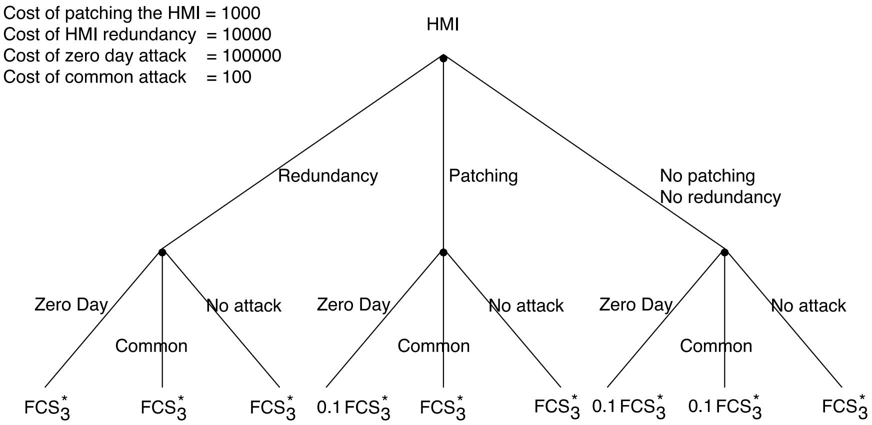


Figure 13. Attack/Defence tree for HMI.

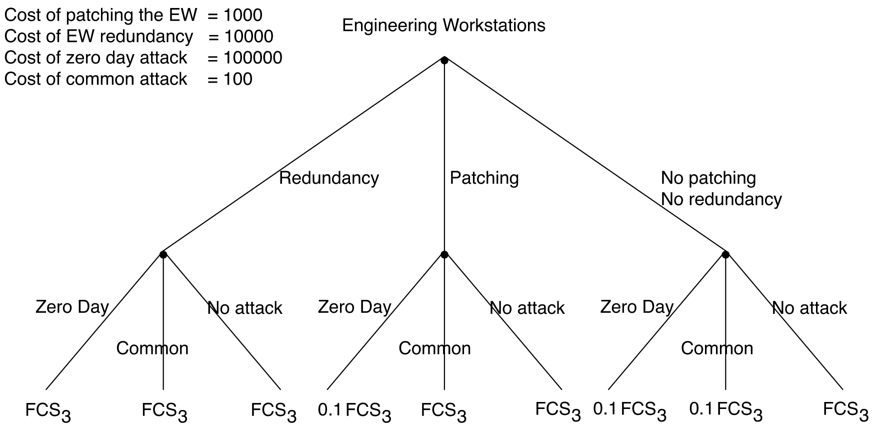


Figure 14. Attack/Defence tree for Engineering Workstations.

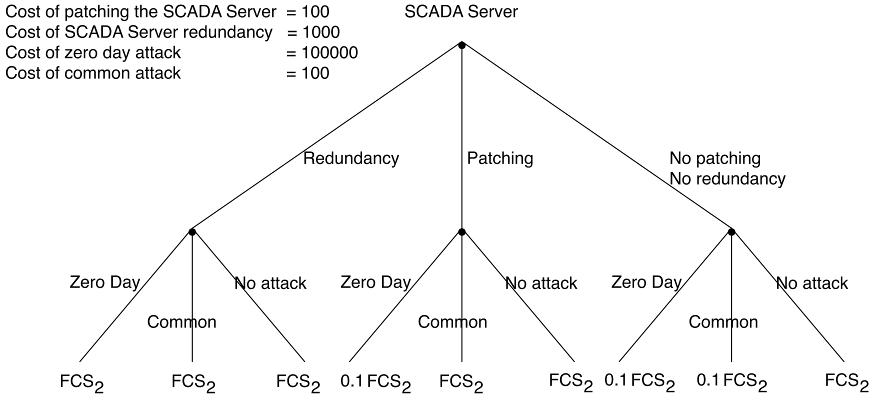


Figure 15. Attack/Defence tree for SCADA server

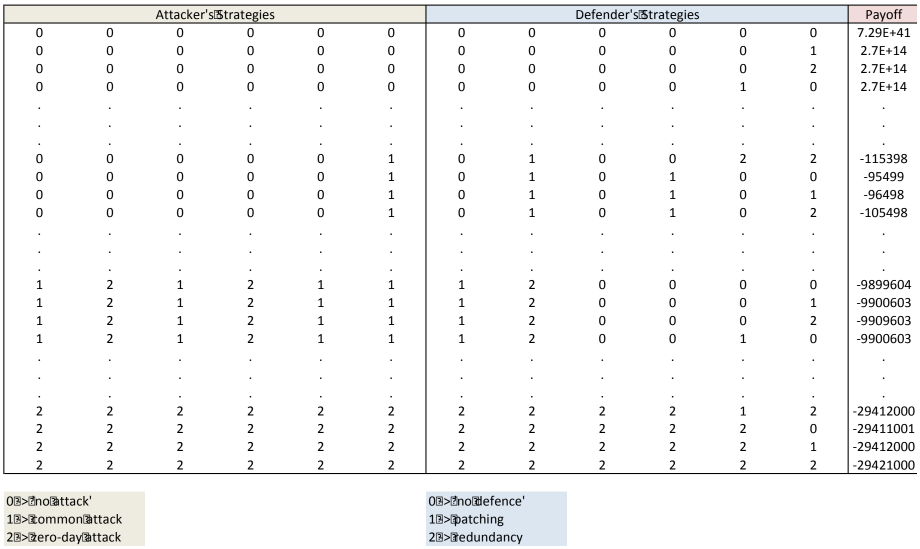
Based on the attack/defence trees we can identify all possible scenarios in the game. Since the defender can apply either patching, redundancy or "no security" to each of the six elements, the number of available defence strategies is 3^6. Additionally, since the attacker can choose between zero-day, common attack or "no attack" for each element, the total number of attack strategies is 3^6. Figure 16 shows part of all available pair of strategies along with the defender's corresponding payoff calculated based on Eq. 14. To find the Nash Equilibria of the game we apply the maximin algorithm where the attacker tries to maximise her minimum payoff. The algorithm is also known as *low risk algorithm* and can be described with the following two steps:

- Defender calculates the minimum payoffs for each of her strategies, based on the fact that for each of her 3^6 strategies Attacker would choose a strategy that minimises Defender's payoff (at the end of this step the defender has 3^6 minimums).
- Among those 3^6 minimums, Attacker chooses the strategy that returns the highest minimum payoff. This corresponds to the Nash Equilibrium.

Figure 17 plots the minimum payoffs for each of the defender's strategies. As seen there are four areas that return maximum minimums. In particular, the defender's strategies that correspond to the Nash Equilibrium are:

- Strategy no.16: 000120 (patching S2 and applying redundancy to S3)
- Strategy no.93: 010102 (patching PLC2, patching S2 and applying redundancy to S3*)
- Strategy no.257: 100111 (patching PLC1, patching S2, patching S3 and patching S3*)
- Strategy no.343: 110200 (patching PLC1, patching PLC2 and applying redundancy to S3)

These are the optimal cost-efficient defence strategies.



Attacker's Strategies						Defender's Strategies						Payoff
0	0	0	0	0	0	0	0	0	0	0	0	7.29E+41
0	0	0	0	0	0	0	0	0	0	0	1	2.7E+14
0	0	0	0	0	0	0	0	0	0	0	2	2.7E+14
0	0	0	0	0	0	0	0	0	0	1	0	2.7E+14
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
0	0	0	0	0	1	0	1	0	0	2	2	-115398
0	0	0	0	0	1	0	1	0	1	0	0	-95499
0	0	0	0	0	1	0	1	0	1	0	1	-96498
0	0	0	0	0	1	0	1	0	1	0	2	-105498
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
1	2	1	2	1	1	1	2	0	0	0	0	-9899604
1	2	1	2	1	1	1	2	0	0	0	1	-9900603
1	2	1	2	1	1	1	2	0	0	0	2	-9909603
1	2	1	2	1	1	1	2	0	0	1	0	-9900603
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
2	2	2	2	2	2	2	2	2	2	1	2	-29412000
2	2	2	2	2	2	2	2	2	2	2	0	-29411001
2	2	2	2	2	2	2	2	2	2	2	1	-29412000
2	2	2	2	2	2	2	2	2	2	2	2	-29421000

0 -> 'no attack'
1 -> common attack
2 -> zero-day attack

0 -> 'no defence'
1 -> patching
2 -> redundancy

Figure 16. Available Strategies

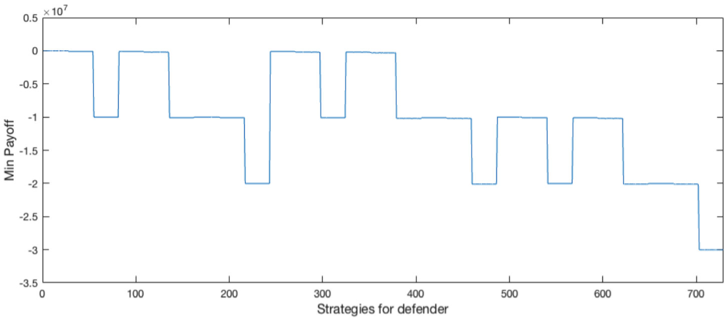


Figure 17. Defender's minimum payoffs for each of her strategies.

5. Conclusion

In this Chapter we combined two classic Systems Analysis techniques, VSM and Game Theory, in order to model an ICS through a living analogy. Unlike the traditional risk analysis methods, where emphasis is on calculating probabilistic measures of risk based on perceived likelihoods of threat occurrences, our approach allowed us to compose a set of formulae that describe the level of service provision of an ICS and can provide the basis for an impact analysis through the lens of viability (defined as the ability to maintain a core level of functionality, as deemed necessary for critical infrastructure). Based on the perceived significance of interacting components and an estimate of the impact of their compromise we set up typical scenarios of attack and defence in ICSs as games between rational players and computed Nash Equilibria for varying strategies of redundancy and immunisation. This approach can be used to design defences against unknown attacks, with reference to the system architecture only.

References

[1] S. Klipper, ISO/IEC 27005, *In Information Security Risk Management*, 63–97, Springer, 2011.

[2] G. Stoneburner, A. Goguen, and A. Feringa, Risk Management Guide for Information Technology Systems, Technical Report, *NIST special publication*, SP 800-30, 2002.

[3] L. Rajbhandari and E. A. Snekkenes, Mapping between Classical Risk Management and Game Theoretical Approaches, *In Communications and Multimedia Security*, 147–154, Springer, 2011.

[4] M. Cheminod, I. C. Bertolotti, L. Durante, P. Maggi, D. Pozza, R. Sisto, and A. Valenzano, Detecting chains of vulnerabilities in industrial networks, *IEEE Transactions on Industrial Informatics*, **5** (2009), 181–193.

[5] G. Digioia, C. Foglietta, S. Panzieri, and A. Falleni, Mixed Holistic Reductionistic Approach for Impact Assessment of Cyber Attacks, *In European Intelligence and Security Informatics Conference (EISIC)*, 123–130, IEEE, 2012.

[6] K. Hausken, Probabilistic Risk Analysis and Game Theory, *Risk Analysis*, **22** (2002), 17–27.

[7] G. Levitin, Optimal Defense Strategy Against Intentional Attacks, *IEEE Transactions on Reliability*, **56** (2007), 148–157.

[8] B. Stafford, The Viable System Model: Its Provenance, Development, Methodology and Pathology, *Journal of the Operational Research Society*, **35** (1984), 7–25.

[9] T. Spyridopoulos, K. Maraslis, T. Tryfonas, G. Oikonomou, and S. Li, Managing Cyber Security Risks in Industrial Control Systems with Game Theory and Viable System Modelling. *In 9th International Conference on System of Systems Engineering (SOSE)*, 266–271, IEEE, 2014.

[10] G. Levitin and K. Hausken, Redundancy vs. Protection vs. False Targets for Systems Under Attack, *IEEE Transactions on Reliability*, **58** (2009), 58–68.