Defense-in-depth vs. Critical Component Defense for Industrial Control Systems

Andrew Fielder
Institute for Security Science and Technology
Imperial College London
London, UK
andrew.fielder@imperial.ac.uk

Tingting Li
Institute for Security Science and Technology
Imperial College London
London, UK
tingting.li@imperial.ac.uk

Chris Hankin
Institute for Security Science and Technology
Imperial College London
London, UK
c.hankin@imperial.ac.uk

Originally designed as self-contained and isolated networks, Industrial Control Systems (ICS) have evolved to become increasingly interconnected with IT systems and other wider networks and services, which enables cyber attacks to sabotage the normal operation of ICS. This paper proposes a simulation of attackers and defenders, who have limited resources that must be applied to either advancing the technology they have available to them or attempting to attack (defend) the system. The objective is to identify the appropriate deployment of specific defensive strategy, such as *Defense-in-depth* and *Critical Component Defense*. The problem is represented as a strategic competitive optimisation problem, which is solved using a co-evolutionary *Particle Swarm Optimisation* problem. Through the development of optimal defense strategies, it is possible to identify when each specific defensive strategies is most appropriate; where the optimal defensive strategy depends on the kind of attacker the system is expecting and the structure of the network.

Industrial Control Systems, Defense-in-depth, Defensive Strategy, Agent-based Modelling

1. INTRODUCTION

Industrial Control Systems (ICS) play a crucial role in supervising industrial processes and production. Disruption to ICS might lead to disastrous damage to the plant, environment and even human health (Stouffer et al. 2011). ICS were originally designed as isolated self-contained systems, which nowadays have evolved to become increasingly interconnected with IT systems and other complex networks. It greatly improves the efficiency of communication and control of ICS, but has left ICS exposed to cyber threats. Modern ICS thus have to be tolerant of accidental malfunctions, as well as intentional cyber attacks. ICS-CERT received 295 reports in 2015 by trusted asset owners[1]. In particular multi-stage Advanced Persistent Threats (APT) account for roughly 55% amongst the various cyber attacks against ICS [2]. ICS-targeted APT often start with gaining access to the target network, propagate through the network by continuously exploiting chains of vulnerabilities, and eventually disrupt the operation of ICS. We outline common ways to stage APT targeting a typical ICS in Figure 1. The most well-known example of such attacks is Stuxnet in 2010(Falliere et al. 2011), which was introduced to the targeted network by a removable flash drive and eventually infected approximately 100,000 hosts across over 155 countries until September 2010 according to the Symantec report by Falliere et al. (2011). More recent accidents are the German steel mill breach in December 2014[3] and the Ukrainian power outages in December 2015[4] .

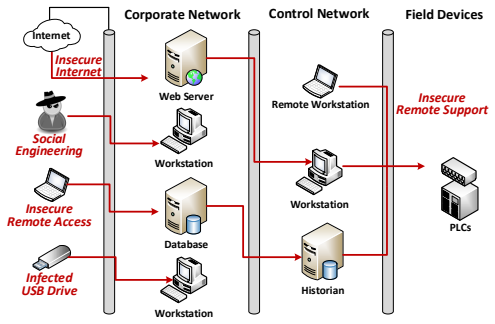


Figure 1: *Typical ICS architecture threatened by APT*

[1]ICS-CERT: Nov.2015 - Dec. 2015. https://ics-cert.us-cert.gov/monitors/ICS-MM201512

[2]ICS-CERT: Sep. 2014 - Feb. 2015. www.ics-cert.us-cert.gov/monitors/ICS-MM201502

[3]SANS ICS Case, 2014. https://ics.sans.org/media/ICS-CPPE-case-Study-2-German-Steelworks_Facility.pdf

[4]ICS-CERT Alert, 2016. https://ics-cert.us-cert.gov/alerts/IR-ALERT-H-16-056-01

© *The Authors. Published by BISL.*

To mitigate the increasing cyber threats targeting ICS, many government advisory reports and industrial standards propose *Defense-in-Depth* as the best practice to defend ICS (Stouffer et al. 2011; Kuipers and Fabro 2006). Defense-in-depth aims to protect a system by establishing a multi-layer defense by combining various defensive controls, such as advanced firewalls with DMZ, security awareness training programs, a vulnerability management system, intrusion detection with effective access policies, and incident response mechanisms. However, as there are numerous controls involved, it needs a deployment plan to optimally distribute the defensive resources over those controls before such practice could produce the most effective protection with limited resources. Otherwise the high financial and managerial cost make defense-in-depth impractical to fully implement (Small 2011), because massive unnecessary efforts might be wasted on irrelevant attack vectors and security activities. With limited resources, system wide defense-in-depth provides only a wide but low-level defense across the network, which might be able to stop sophisticated attackers. Therefore, we produce a decision support tool to help with better understanding of defense for ICS, and discover the optimal defensive strategy for use in ICS.

The concept of *attacker* denotes the possible cyber attackers attemping to breach an ICS, while *defender* acts as the security manager who needs to deploy available defensive controls to protect the ICS. Given an established network, we first generate the underlying attack graph for it by using our logic-based reasoning engine. The attack graph chains various weaknesses of the network that can be exploited by attackers to stage an APT attack. The attacker and defender are then modelled as a pair of competing agents in a co-evolutionary simulation. *Particle Swarm Optimisation* (PSO) (Kennedy 2010) is adopted to aid agents in finding the most optimal strategy to attack and defend given different conditions and profiles. Agents' gains and losses are qualified in iterated games, by which an overall score can be produced to evaluate the performance of the chosen strategy. Eventually both the attacker and defender would co-evolve to their best strategies against each other. From this work, we discover that the decision on defensive strategies should rely on the type of attacks we are combating and the network layout including network topology and distribution of valuable targets. Particularly we find out that system wide defense-in-depth is viable in protecting the system from greedy attackers, where lower-level conventional attacks are generally used. However, defense-in-depth is less capable of defending against more sophisticated attacks, in which case the defensive effort should be focused on the critical targets in the system. Furthermore, we run extended experiments to investigate the role of network topology in deciding defensive strategies. Specifically we look at the performance of defending bottleneck nodes of a network to produce effective protection with minimised defensive efforts.

The paper starts with a related work section where the work on attack modelling of ICS and APT attacks, agent-based coevolutionary approaches and PSO-related topics are presented. The approach proposed in this paper is discussed in Section 3.1, including the modelling of the key elements (e.g. attacker methods, defense controls, network architecture) and the development of the agent-based simulation. A case study extracted from CSSP Recommended Defense-In-Depth Architecture (Kuipers and Fabro 2006) is designed in Section 4 to demonstrate the effectiveness of our proposed tools in finding optimal defense deployment. Five different scenarios are provided to capture attackers and defenders with different profiles. Relevant results are presented in Section 5 and discussed in Section 6. The paper concludes with a summary and discussion of further directions of research in Section 7.

2. RELATED WORK

Stouffer et al. (2011) provided a comprehensive introduction to the key ICS-specific cyber threats. Automatic generation of attack graphs based on CVE vulnerabilities has been well developed, such as the tool MulVal by Ou et al. (2006) and NetSPA by Lippmann et al. (2005). However, such complete CVE-based attack graphs are often very complex to understand and analyse. Thus our work produced attack graphs based on common weaknesses of ICS, rather than specific vulnerabilities on each host, by which we can lift our focus of defense to combat more generic classes of attacks and hence produce a broader view to deploy defensive controls. Lippmann et al. (2005) also abstracted complete attack graphs by classifying vulnerabilities in terms of CVE factors. Attack graphs have been applied effectively to assess the potential risks of a network. Noel et al. (2010) introduced a metric for quantifying the security of a network based on attack graphs with assigned likelihoods of each attack edge. Using attack graphs for risk analysis of critical infrastructures was reported by Ma and Smith (2013). Attack graphs are able to help with finding effective defensive measures by studying the network structure and required vulnerabilities to comprise a system. Thus we also adopt attack graphs for the initial representation of the problem.

Another closely related area to this work is the concept of network hardening. Fielder et al.

(2014) present a game theoretic approach to the optimal allocation of system administrator time to defensive tasks. One important finding of the work shows that a greater emphasis of the limited administrator time should be placed on the most valuable assets, which is conceptually consistent with the Critical Component Defense strategy discussed in this paper. Game theoretic approaches using Stackelberg games to finding optimal security decisions for real-world scenarios have been extensively studied, such as scheduling of airport security (Korzhyk et al. 2010), allocation of air marshals to flight paths (Tsai et al. 2009) and deployment of honeypots (Durkota et al. 2015).

PSO was originally proposed by Eberhart and Kennedy (1995), and a more up-to-date review of this area was provided by Poli et al. (2007). Poli (2008) identified active applications of PSO, and also pointed out that very little work has undertaken by applying PSO to cyber security problems, with only 1.3% of the literature covering the whole security field, such as security predictions (Gao et al. 2011), intrusion detection (Srinoy 2007) and authentication (Karnan and Akila 2010). A similar PSO-based simulation was also employed to investigate the impact of cost-efficiency of defence on deciding the optimal defence for ICS in our previous work (Fielder et al. 2016) .

3. MODELLING AND SIMULATION

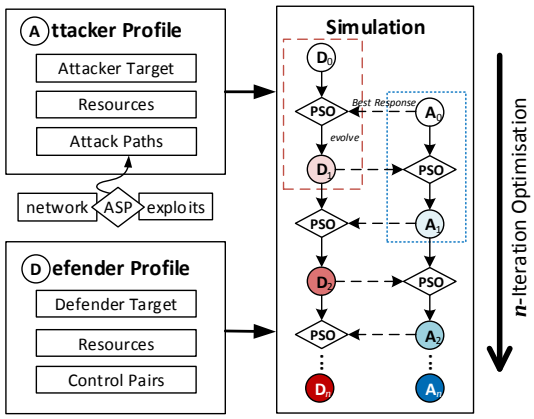


Figure 2: *system overview*

Figure 2 shows the schematic diagram of the simulation. The *Attacker Profile* and *Defender Profile* collect key elements to decide attackers and defenders' strategies and actions respectively. Attackers' strategies are decided by attack targets (e.g. espionage or sabotage), resources available for attackers and all possible attack paths. Particularly the attack paths are generated by an automatic reasoning engine by a logic programming technique - *Answer Set Programming (ASP)* by Gebser et al. (2011) analysing an established network and exploits on each host in the network. Defenders also have to decide the preference of targets to defend, based on the available resource and control pairs. Formal definitions of the above notions above are given in Section 3.1. The right part of Figure 2 depicts the development of optimal solutions by using our simulation. Following the optimization algorithm *PSO*, each candidate strategy is encoded as a particle of a swarm and all such particles are gradually moving towards the best solution. In each iteration of evaluation, both attackers and defenders generate the best response to compete with each other, and eventually co-evolve to optimal solutions. Details of the simulation and optimisation process are discussed in Section 3.2.

3.1. Modelling and Representation

We start with the representation of the pair of competing agents – *Attackers* and *Defenders*, including the key components for defining their strategies. We model a typical ICS architecture as a *network* graph, where each host/asset is identified as a *target* node $t_i \in \mathcal{T}$, an edge $e \in \mathcal{E}$ indicates a valid connection between a pair of targets. Each target has different impacts on attackers and defenders. A compromised target produces certain gains $I^a : \mathcal{T} \to \mathbb{Z}^+$ for attackers, while causing certain damage for defenders $I^d : \mathcal{T} \to \mathbb{Z}^-$.

The notion *attack methods* $\mathcal{M} = \{m_1, \ldots, m_n\}$ collects all weaknesses of the targets in a network and we mainly have three types of attack methods in terms of their origins: (i) *primary attacks:* allow attackers to gain initial access to the network, such as Internet malware, social engineering, and removable drives malware. (ii) *subsequent attacks:* help attackers penetrate further into the network, such as SQL-injection, weak authentication bypass and other LAN-based infection. (iii) *final attacks:* cause actual damage on the field devices, such as buffer-overflow and man-in-the-middle attacks.

An *attack path* $\langle (t_i, t_j), m_k \rangle$ is derived by attaching an applicable attack method m_k to an edge in the network, indicating a possible way to progress the attack from one target t_i to another t_j. For this paper, we develop a logic reasoning engine to generate all possible attack paths for a given network, which altogether render an *attack graph*. An example of such an attack graph is given in Fig3(a). $O(t_i)$ has all outbound paths from the target t_i. At each step of an APT, attackers decides either to *upgrade* an attack method, or to perform an *attack* aganist a target. In the cases of upgrading, attackers have to further select the attack method to upgrade, while in the cases of actual attacking, attackers have to decide which outbound attack path to proceed.

Definition 1 *An* **attack mixed strategy** $a := \langle \alpha, \beta, \Phi, \Psi \rangle$*, where*

- $\alpha \in [0,1]$*, probability of upgrading a method.*
- $\beta \in [0,1]$*, probability of launching attacks, and* $\alpha + \beta = 1$*.*
- $\Phi = [\phi_1, \ldots \phi_n]$*, probability distribution over* $\mathcal{M}$ *and* ϕ_k *denotes the probability of upgrading the attack method* m_k *and* $\sum_{i=1}^{n} \phi_i = 1$*,* $\phi_i \geqslant 0$*.*
- $\Psi = \{\psi_1, \ldots \psi_m\}$*, a set of probability distributions over outbound paths from all targets, and* $\psi_i = [\varphi_i^1, \ldots, \varphi_i^k]$ *denotes the probability distribution over all outbound paths from the target* t_i*, denoted by* $O(t_i) = [p_i^1, \ldots, p_i^k]$ *and* $\sum_{j=1}^{k} \varphi_i^j = 1,\ \varphi_i^j \geqslant 0$*.*

Upgrading attack methods is likely to increase the chance of success, while only attacking brings actual impacts. Both actions consume resources. A **greedy attacker** spends most effort on performing attacks rather than upgrading, but such attacks would not succeed on heavily defended targets. A **methodical attacker** tends to launch infrequent attacks with more advanced attack methods. Finding an optimal distribution between upgrading attack methods and actual attacking is not an easy task, particularly since the chosen attack strategy has to be evaluated against unknown defender's strategy.

The notion of defenders characterises the role of a security manager who needs to find a way of deploying defense controls to protect an ICS. We define a set of ***defense controls*** $\mathcal{C} = \{c_1, \ldots, c_m\}$, that are available for an defender to employ and $D(c_i)$ is the set of attack methods c_i counters. Defenders also have two actions to comprise a defense strategy – *advancing* a defense control or actually *deploying* a control. Unlike attackers who have specific targets to plan attacks, defenders have to protect a number of targets at different levels of a network from numerous possible attacks. Particularly, to combat APT-like attacks, defenders are required to consider not only defending one particular attack, but also stopping the exploit and formation of a complete attack route reaching the most valuable targets.

Definition 2 *A* **defense mixed strategy** $d := \langle \gamma, \delta, \Theta, \Omega \rangle$*, where*

- $\gamma \in [0,1]$*, probability of advancing the defense level of a control.*
- $\delta \in [0,1]$*, probability of deploy a control, and* $\gamma + \delta = 1$*.*
- $\Theta = [\theta_1, \ldots \theta_n]$*, probability distribution over* $\mathcal{C}$*.* θ_i*, probability of advancing* c_i *and* $\sum_{i=1}^{n} \theta_i = 1, \theta_i \geqslant 0$*.*
- $\Omega = [\omega_1, \ldots \omega_m]$*, probability distribution over targets* $\mathcal{T}$ *and* ω_j *is the probability of deploying controls on target* t_j*, and* $\sum_{j=1}^{m} \omega_j = 1, \omega_j \geqslant 0$*.*

Finding the most optimal defense strategy against various unknown attack strategies is challenging. Several well known defensive strategies are given as follows: (i) system wide *defense-in-depth* (DID); (ii) focusing the defense on the critical components (CCD) in the network; (iii) defend the *bottleneck* targets through which all attacks have to pass; or (iv) mixture of the above. It is not easy to tell which one is the most appropriate strategy combating various types of attackers, and therefore we provide an agent-based solution to this problem.

3.2. Simulation

We first model an attacker and a defender as a pair of competing agents in a co-evolutionary process, where both agents aim to produce the optimal mixed strategies to maximise their payoffs. A PSO algorithm is developed to solve such a problem. It starts with an initial set of randomised mixed strategies for the attacker and defender, represented as particles in a swarm. The PSO then runs a number of iterations in order to generate the best response strategies to compete with each other. In each run, the algorithm moves the particles towards a better solution, which is achieved by applying a movement parameter (called a velocity) to the particles. The velocity of a particle is a special form of the mixed strategy, where the sum of all components must equal zero.

The evaluation of a particle is conducted by simulating the interactions between the attacker and the defender over a fixed number of time steps. In terms of the damage that successful attacks cause, the payoff of chosen actions at each step are scored, by which an overall score can be produced to evaluate their performance and the chosen strategy. At each time step the defender chooses an action, corresponding to advancing a control with probability γ and deploys new defenses to a network component with probability δ. If advance is selected, then the defender increases the level of a control c_i by 1 based on a single selection from amongst the probabilities defined by Θ. If the defender chooses to deploy a defense, then they select a device given by the distribution and apply all available controls for that device up to the maximal level advanced so far. In a similar manner, the attacker advances the methods which they are able to exploit the system with probability α and attempt to attack the system with probability β. When an attacker chooses to attack, the attacker starts at the node labelled EXT and selects a node to attack, and an associated method to advance to that node. An attack can only be considered successful if the level of the attack is greater than the level of defense at that node, the success of an attack is given by comparing the level of the method of attack used and the highest

level of control implemented on that node. If an attack is considered successful, then the attacker selects a new node from that location and a new method to attack, and the process is repeated until either an attack is unsuccessful or, there are no more outward paths from the current node. When an attack is halted, the players each receive a score associated with the last node that has been successfully reached.

4. CASE STUDY

In this section, we demonstrate the effectiveness of our tools with a case study on ICS security management. The case study uses a typical three-zone ICS architecture extracted from Kuipers and Fabro (2006); Stouffer et al. (2011). As given in Figure 3(a), there are three zones in the architecture, *Corporate Network*, *Control Network* and *Field Devices*. Each circle represents a common type of host in each specific zone. For instance, *cpWs* stands for a workstation in the corporate network, and *ctRWs* for a remote workstation in the control zone. *EXT* and *PLC* are two special nodes, representing the external untrusted environment and the key control units respectively. Besides, a number of attack methods are gathered in Figure 3(b) from *ICS Top 10 Threats and Countermeasures*, BSI (2014) and *Common Cybersecurity Vulnerabilities in ICS*, Department of Homeland Security (2011). The corresponding set of defense controls is given in Figure 3(c), derived from BSI (2014) and Kuipers and Fabro (2006).

A complete attack graph in Figure 3(a) is then generated by our ASP reasoning engine for the case study. Each exploitation of a weakness of the system is represented by an edge in the graph. For instance, all attacks aiming to *PLC* have to at least comprise either *ctRWs* or *ctWs* which has direct access the *PLC*, and then launch attacks such as Buffer Overflow (*finM1*) and Man-in-the-middle (*finM2*). These control workstations are generally not connected to any untrusted network, but they can be infected by other hosts in the same control network such as *ctHmi* and *ctHist* in the example. Besides, remote workstations used by remote maintenance contractors are threatened by viruses infected by other external assets.

We design five scenarios for different analysis purposes. Table 1 highlights the key settings of the cases. *Case 1* aims to find out the optimal defense for protecting a single high-valued target *PLC*, while *Case 2* increases the values of *ctRWs* and *ctWs* to create a larger defense coverage to consider. *Case 3(a)* and *3(b)* implement paired strategies for both the attacker and the defender, to investigate

Table 1: *Case study settings*

Case 1	Single high-valued target (*e.g. PLC*)
Case 2	Multiple high-valued targets
Case 3(a)	DID defender *vs.* Greedy attacker
Case 3(b)	CCD defender *vs.* Methodical attacker
Case 4	DID defender*vs.* Methodical attacker
Case 5	Bottleneck-node Defense

the interactions between different strategies. *Case 4* particularly studies the the performance of defense-in-depth against a methodical attacker. A complementary *Case 5* is designed to find out effective defensive strategies to protect a network with bottleneck nodes.

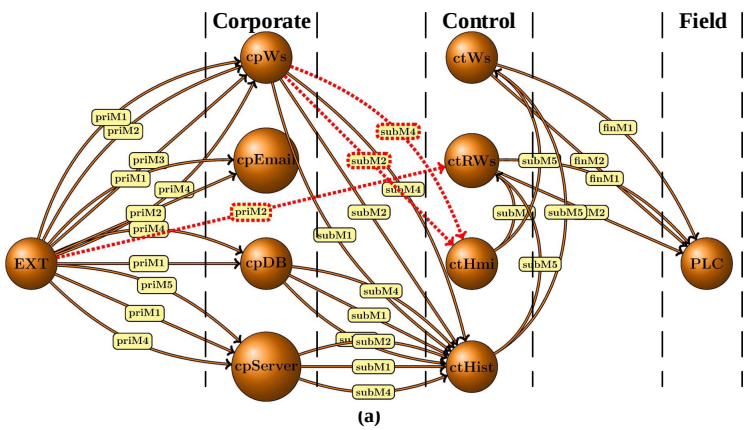
The PSO algorithm sets the size of a swarm at 200, which was sufficient to represent and explore the search space. The simulation operates over 100 moves per particle and for 100 generations of competition between the two agents. The w values for all factors contributing to the velocity were set at 0.05, this was set so as to allow for better exploration of the strategy space, by not favouring a single component. It runs over 50 time steps for the attacker and defender strategies and a particle is evaluated 30 times to reduce variance from the non-deterministic nature of the simulated environment.

5. RESULTS

The results of the simulations are represented as combined effort heat maps of the network, shown in Figure 4. The results represent the combined effort that should be applied to the asset in the network. For the defender, the combined effort is the probability of upgrading the defense of a node, added to the probabilities associated with advancing the controls that are relevant to defending that node. Additionally, the attackers effort is represented by the lines connecting the node, where the combined effort is the probability of using that attack path and the probability of using and advancing the attack methods relevant to that line. In Figure 4, the higher the combined effort, the darker the colour of either the line or node.

In this initial case 1, where we consider that each player only has interest in the *PLC*, we see that in Figure 4(a) that the defender chooses to apply a large proportion of effort to the *PLC* Node.

The results present a case where the defender chooses to split effort evenly between advancing and upgrading, but chooses with $p > 0.95$ to advance control *c6* and applying that defense to the *PLC* node. While the defender chooses to upgrade and implement *c6* in the results presented, *c5* is equally preferable in this scenario since it also protects *PLC*.



(a)

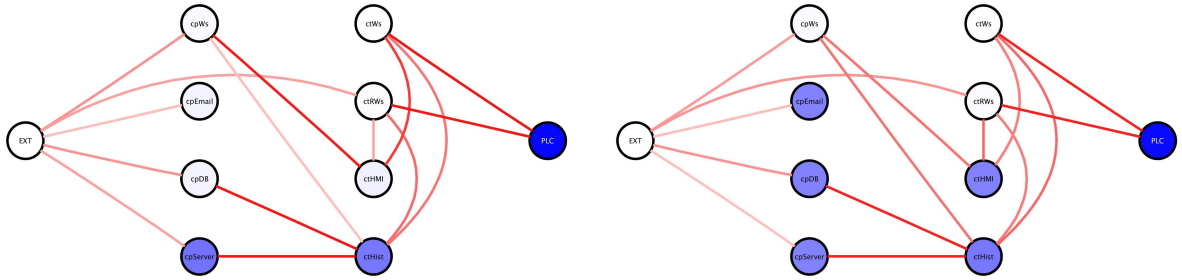
Attack	Attack Description	Attack	Attack Description
priM1	Internet Malware	subM2	Authentication Bypass
priM2	Removable Drive Malware	subM3	Removable Drive Malware
priM3	Social engineering	subM4	Abuse of Access Authority
priM4	Malicious Remote Access	subM5	LAN-based injection
priM5	Cross-site Scripting	finM1	Buffer overflow
subM1	SQL Command injection	finM2	Man-in-the-middle

(b)

Control	Control Descriptions	Attack Methods Defended
c1	Removable media checked before use	priM2
c2	Employees security training	priM3
c3	Fine-grained remote access	priM4
c4	Anti-virus Software	priM1, priM2, subM5
c5	DMZ and Firewalls	priM1, priM4,subM2,subM4, finM1, finM2
c6	Network data validation & integrity check	priM5, subM1,subM3, finM1, finM2
c7	Fine-grained access controls	priM4, subM2, subM4

(c)

Figure 3: *Case study on ICS security management: (a) attack graph with all attack paths(dashed edges are removed for* ***Case 5****); (b) attack methods from top weaknesses of ICS; (c) Common defense controls for ICS and attack methods defended.*



(a) **Case 1**: one valuable target only

(b) **Case 2***:* mutiple valuable targets

(c) **Case 3a***:* greedy attack *vs.* defense-in-depth

(d) **Case 3b***:* methodical attack *vs.* critical component defense

Figure 4: *Simulation Results*

In addition to this, it is in the best interest of the defender to upgrade only a single control and not attempt to use both *c5* and *c6*, since the effort would be split between the two, but would not achieve the same level of defense, since we consider the defensive level to be equal to the highest of the controls implemented.

Since the only damage in the system is represented at the *PLC* node, this reduces the expected damage to near zero levels. Since the damage reaches a near zero level, the attacker has relatively few strategic choices, with the primary method being to be as aggressive as possible, with the aim of exploiting the system before the defender has established an effective defense.

As part of this, we see some incidental effort that is applied to both *cpServer* and *ctHist*, this is because control *c6*, has some protective capability for both of those nodes, however the actual level of upgrade applied to those nodes is very minimal.

The second case in Figure 4(b) represents a scenario, where there are 3 nodes in the system that are considered valuable to both the attacker and the defender.

Unlike Case 1, the defender in this scenario chooses to use control *c5* instead of *c6*. The reason for this is that in Case 1, there is no reason to try and defend any other nodes, because the one valuable node is fully protected, however with additional nodes that are valuable there may be a justification for applying defense elsewhere, which would warrant the use of

a control that is effective against the highest number of additional vulnerabilities.

In this case slightly more emphasis has been placed by the attacker on attacking *ctWs* over *ctRWs*, since the former node has a slightly higher value for the attacker if it is compromised. While the defender does play a defense-in-depth strategy, that would allow for the attacker to attack *ctWs* with $p = 1$ to get a higher score, the defender would be able to shift defensive strategy to that node to reduce the damage even further, as such it is in the best interest of the attacker to spread the attack between *ctWs* and *ctRWs*.

Case 3 presents a scenario where there is value to both players across the whole of the system. Unlike in other cases, we see that there is no single strategy that provides the best defense for the defender. In this case we see that there are two possibilities for both the attacker and the defender.

Figure 4(c) shows the first pair of strategies, which represents a defense-in-depth and a greedy attacker.

The heat map displayed in Figure 4(c) shows a low effort value on all of the nodes. The reason for this apparently low effort, is that the defensive actions are spread across all the nodes and controls. This defensive strategy aims to provide a basic defense on all of the nodes as quickly as possible and then tries to slowly upgrade the whole defense over time. This is a simple strategy that will prevent most conventional attacks, but would not provide an effective mechanism against sophisticated attacks. The attacker we see in Figure 4(c) is similar to the attackers in both cases 1 and 2, which is considered a greedy attacker, since it attacks frequently to maximise score by launching lots of unsophisticated attacks. In this case the attacker is able to potentially score quickly as there are a number of nodes that the attacker can exploit early.

Counter to this strategy, we have a defender employing critical component defense and a more methodical attacker in Figure 4(d).

The methodical attacker, still prefers to attack, since continually advancing the attack methods does not impact their score directly, however unlike the previously seen greedy attackers, the methodical attacker attempts to advance attack methods to be able to reliably overcome low level defenses that might be protecting the outer components of the system. This is represented in in Figure 4(d) as a lower attack effort across all lines.

The defender is the same critical component defense style defender that we saw in case 2, where despite the system having a number of nodes with value the defender chooses to defend only the most important. We have seen in Case 3a that this is not the only possible choice for the defender and is relevant to the attack strategy.

It is important to note, that both the defensive strategies shown for case 3 in Figure 4 are the counter to the paired attacker strategy. A greedy attacker who is using low level attacks is countered by a strategy that employs basic defenses across the whole of the network, i.e. defense-in-depth. However a methodical attacker will be able to beat this strategy, since they advance the attack methods over the basic controls that are present in the system wide defenses, allowing them to more frequently exploit the system to reach the high valued *PLC* node. Against this methodical attacker, the defender is best advised to play a critical component defense strategy, since the attacker attacks less frequently, so the average expected loss is reduced and the *PLC* is not normally compromised. In this case the attacker is more incentivised to launch numerous low level attacks that are able to get past either basic or non-existent defense, which is akin to the initial greedy attacker.

Table 2 shows the distribution of optimal strategies for each player amongst the two main actions. The defender's actions in Cases 1, 2 and 3b have the same optimal probabilities, in which the defender splits the effort in a near even manner between advancing the level of the control and implementing the upgrades. Likewise in Cases 1, 2 and 3a, the attacker chooses to attack with probability $p(A) > 0.8$. This represents the consistent behaviour of an aggressive attacker and a critical component defender. However if we consider the patient attacker in case 3b, we see that they attack with a lower probability, increasing the the likelihood that their attacks are more sophisticated. The defender in case 3a upgrades the defense more frequently than the other defenders, where the defender advances the controls rarely, opting to implement lower level controls where available, this is consistent with a defense-in-depth style approach.

Table 3 shows the distribution of emphasis on upgrading the defense at each of the nodes in the network. Case 3b shows that there is little emphasis placed on any node other than data, with a value of 0.1 on each node except for Data, which has a upgrade probability of 0.92. This is in contrast to case 3a, where Data only has an upgrade probability of 0.22. The rest of the probability is then distributed amongst the rest of the node. This is also reflected in the effort place on controls, where in case 3b control c5 is used with $p(c5) = 0.97$, but in case 3a

Table 2: *Optimal Distribution in Strategies*

		Case 1	Case 2	Case 3a	Case 3b
Defender	Upgrade	0.47	0.47	0.77	0.47
	Advance	0.53	0.53	0.23	0.53
Attacker	Attack	0.9	0.83	0.85	0.61
	Advance	0.1	0.17	0.15	0.39

Table 3: *Distribution of Upgrade Effort by Node*

Node	cpWS	cpEmail	cpDatabase	cpServer	ctWS	ctRW	ctHmi	ctHist	Data
Case 3a	0.05	0.01	0.13	0.01	0.18	0.18	0.04	0.17	0.22
Case 3b	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.92

the highest distribution for any control is c3 with a $p(c3) = 0.25$.

6. DISCUSSION

The results present a scenario where we see that there is a rationale to consider that critical component defense is the most viable defensive strategy given a limited amount of resources for defending a system. This appears to hold true in the case of industrial control systems with a single asset or set of assets that are considered particularly valuable compared to the rest of the system. With a single point of failure, the defender is incentivised to defend that node very heavily, even in the face of other low value attacks. As the value of the damage across the system rises, in comparison to the node with a single point of failure, the defender now has more incentive to protect more of the nodes if possible. The results of case 3 identify that there is a certain point where the value of the system causes the defender to look at the trade-off between multiple low level threats and single highly sophisticated attacks. While the specific view of an ICS in this paper focusses on those with highly valuable field controllers, however not all ICS systems have these single high value nodes; it is ICS with this kind of property that with limited resources it would be best to employ a critical component defense style system.

To demonstrate this we considered a further case, where each of the nodes has a similar value for successful compromise, with the outermost nodes all valued at 100 with nodes at each successive depth worth an additional 10 to the attacker. This removes a single critical point of failure for the defender, which would spread the damage out across the system more evenly. Without a single point of failure for the network, the defender is offered two main strategies, a system wide defense-in-depth strategy, that protects the slightly more valuable inner nodes, while providing basic security across the network, or a perimeter defense strategy, where the defender attempts to protect each of the outer nodes as highly as possible.

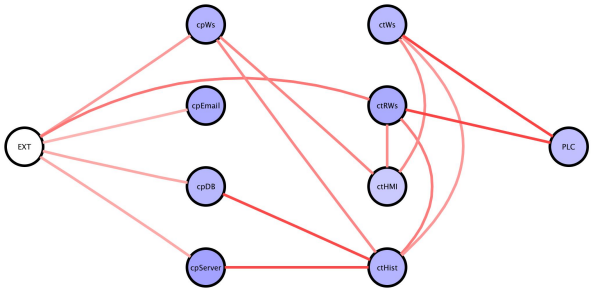


Figure 5: Case 4*: methodical attack* vs. *defense-in-depth*

The results presented in Figure 5 show that by applying a more even spread of value across the network, the defensive strategy is consistent with the strategy presented in case 3 for the defense-in-depth style defender. Unlike case 3 we are presented with only a single attacker and defender strategy, where the attacker is a methodical attacker. The reason for this is that since the defender has equal value across their system, there is no incentive to try and defend a single point, given that any targeted defense strategy would be met with an attacker strategy that would obtain better results by ignoring that node. It is also for this reason that means given a methodical attacker, the attacker has no incentive to change to a less effective greedy approach, which dictates that the same cyclical nature of attack and defense strategies presented in the results of case 3 do not occur.

While exploring the concept of defense-in-depth, we identified that this strategy does not make sense when there is a single focal point to defend and multiple routes to reach that node. We considered that if there was a defensive point earlier in the attack graph, that had the same defensive properties as the single valuable node, then it might be preferable to defend at that point instead. To this end we considered that if we could create a bottleneck for the attacker, then it would be more preferable for the

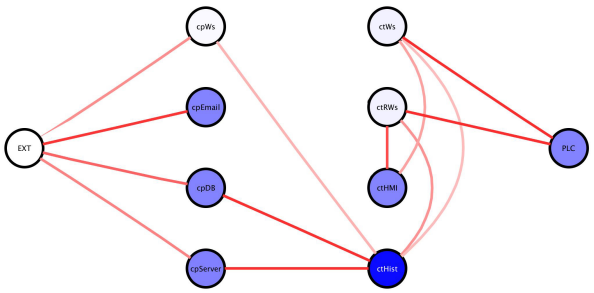


***Figure 6: Case 5**: defending bottleneck*

defender to not protect the vulnerable node directly, but instead protect the system at a node causing a bottleneck earlier in the system. The reason for this, is that while defending at the bottleneck, the defender is able to protect not only the most vulnerable node, but all nodes after the bottleneck in the attack graph. The rationale for this is similar to that of perimeter defense.

To test the idea of defense in bottlenecks, we have designed an additional Case 5, where all of the attacks must pass through the node *ctHist* in order to reach the PLC node. We modified the scenario for case 3, by removing three attack paths (highlighted in dashed lines in Figure 3(a)) from the attacker.

The results shown in Figure 6, demonstrate that by creating a bottleneck of attacks at an earlier stage of the attack, it is better for the defender to protect at *ctHist*. Since *ctHist* is protected from attacks, then *ctWs*, *ctRWs* and *PLC* are all effectively covered as well, which is better than the alternative critical component defense scenario, which would leave *ctHist*, *ctWs* and *ctRWs* uncovered. The system wide defense presented in case 3a would also be less efficient in a number of cases, since spreading the defense would potentially open the attacker to higher value nodes if the attacker were to implement a more methodical approach.

From the results, the defender uses control *c5* to defend the node, since it covers the vulnerabilities at *ctHist*. Much like in cases 2 and 3, the defender chooses c5 as the primary control to upgrade, because it gives the best opportunities for covering additional nodes with minimal additional investment of time.

Conceptually we believe that critical component defense makes sense, but as a special case of the wider concept of defending bottlenecks. As such, a system that represents defense in bottlenecks considers that the defense should be centred around the set of minimal elements that all attacks must pass through, and if there are multiple minimal sets of nodes, then the defense should be applied at the set of nodes that occur earliest in the collection of attack paths. It should be noted that this applies when the value of assets increases with depth of the attack or with a single level of highly valuable nodes, such as those in an industrial control system.

7. CONCLUSIONS

This work shows that when a defender has relatively few valuable assets in a system, the best use of resources and effort available is to apply the defense to the most valuable nodes. This kind of defense is particularly important in a number of ICS, where there is a single critical point of failure for the organisation. This strategy does not work effectively if the system has a wide spread of valuable targets, since there is no single place to focus the defensive strategy to protect the system. Under these circumstances, defense-in-depth is generally preferable, since a set of controls across all nodes of the network will reduce the viable attack surface. The defense-in-depth strategy works most effectively against volumes of attacks, not sophistication, when we consider limited resources.

When there is a network with a single point of failure, but also containing other valuable assets, the optimal defensive strategy must consider the kind of attacker the system faces. Where we consider that critical security controls is best applied to those scenarios where the attacker develops sophisticated attack methods and attacks less frequently. However, when there is a hyper aggressive attacker that is attempting to attack the system frequently using relatively low effort attacks, then a defense-in-depth strategy is preferred.

Additionally, we have also introduced the notion that a more general form of the critical component defense strategy is to consider a defensive strategy that operates at bottlenecks in the system by considering at a location that provides the minimal attack surface. It should be noted that not all defenses in ICS are equal, and defending at certain nodes, such as the PLC, might not be viable due to a reduction in performance, such as the battery life of controllers in the field. In addition to considering performance metrics for implementing defensive strategies in a system, further work would be carried out looking at alternative network structures that would be capable of testing the idea of defense at bottlenecks further. Coupled with this we aim to look at the relative amount of resources that each player has available to identify if the strategies diversify and what causes the diversification.

REFERENCES

BSI (2014, mar). Industrial control system security top 10 threats and countermeasures 2014. "www.allianz-fuer-cybersicherheit.de/ACS/DE/_downloads/techniker/hardware/BSI-CS_005E.pdf".

Department of Homeland Security, U. S. (2011). Common cybersecurity vulnerabilities in industrial control systems. "www.ics-cert.us-cert.gov/sites/default/files/documents/DHS_Common_Cybersecurity_Vulnerabilities_ICS_20110523.pdf".

Durkota, K., V. Lisy, C. Kiekintveld, and B. Bosansky (2015). Game-theoretic algorithms for optimal network security hardening using attack graphs. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pp. 1773–1774.

Eberhart, R. C. and J. Kennedy (1995). A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science*, Volume 1, pp. 39–43. New York, NY.

Falliere, N., L. O. Murchu, and E. Chien (2011). W32. stuxnet dossier. *White paper, Symantec Corp., Security Response 5*.

Fielder, A., T. Li, and C. Hankin (2016). Modelling cost-efficiency of defenses in industrial control systems. In *Computer Safety, Reliability, and Security (SAFECOMP)*. Springer.

Fielder, A., E. Panaousis, P. Malacaria, C. Hankin, and F. Smeraldi (2014). Game theory meets information security management. In *ICT Systems Security and Privacy Protection*, pp. 15–29. Springer.

Gao, K., L. Jianming, R. Xu, Y. WANG, and Y. LI (2011). A hybrid security situation prediction model for information network based on support vector machine and particle swarm optimization. *Power System Technology 4*, 033.

Gebser, M., R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and M. Schneider (2011). Potassco: The Potsdam answer set solving collection. *AI Communications 24*(2), 107–124.

Karnan, M. and M. Akila (2010). Personal authentication based on keystroke dynamics using soft computing techniques. In *Communication Software and Networks, 2010. ICCSN'10. Second International Conference on*, pp. 334–338. IEEE.

Kennedy, J. (2010). Particle swarm optimization. In *Encyclopedia of Machine Learning*, pp. 760–766. Springer.

Korzhyk, D., V. Conitzer, and R. Parr (2010). Complexity of computing optimal stackelberg strategies in security resource allocation games. In *AAAI*.

Kuipers, D. and M. Fabro (2006). *Control systems cyber security: Defense in depth strategies*. United States. Department of Energy.

Lippmann, R. P., K. W. Ingols, C. Scott, K. Piwowarski, K. J. Kratkiewicz, M. Artz, and R. Cunningham (2005). *Evaluating and strengthening enterprise network security using attack graphs*. Defense Technical Information Center.

Ma, Z. and P. Smith (2013). Determining risks from advanced multi-step attacks to critical information infrastructures. In *Critical Information Infrastructures Security*, pp. 142–154. Springer.

Noel, S., S. Jajodia, L. Wang, and A. Singhal (2010). Measuring security risk of networks using attack graphs. *International Journal of Next-Generation Computing 1*(1), 135–147.

Ou, X., W. F. Boyer, and M. A. McQueen (2006). A scalable approach to attack graph generation. In *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 336–345. ACM.

Poli, R. (2008). Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Applications 2008*, 3.

Poli, R., J. Kennedy, and T. Blackwell (2007). Particle swarm optimization. *Swarm intelligence 1*(1), 33–57.

Small, P. E. (2011). Defense in depth: An impractical strategy for a cyber world. *SANS Institute, Bethesda*.

Srinoy, S. (2007). Intrusion detection model based on particle swarm optimization and support vector machine. In *Computational Intelligence in Security and Defense Applications, 2007. CISDA 2007. IEEE Symposium on*, pp. 186–192. IEEE.

Stouffer, K., J. Falco, and K. Scarfone (2011). Guide to industrial control systems (ics) security. *NIST special publication*. "http://csrc.nist.gov/publications/nistpubs/800-82/SP800-82-final.pdf".

Tsai, J., S. Rathi, C. Kiekintveld, F. Ordez, and M. Tambe (2009, 1). *IRIS - A tool for strategic security allocation in transportation networks*, Volume 2, pp. 1327–1334. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS).