Modelling Cost-efficiency of Defenses in Industrial Control Systems

Andrew Fielder, Tingting Li, and Chris Hankin

Institute for Security Science and Technology,
Imperial College London, UK
{andrew.fielder, tingting.li, c.hankin}@imperial.ac.uk

Abstract. Industrial Control Systems (ICS) play a critical role in controlling industrial processes. Wide use of modern IT technologies enables cyber attacks to disrupt the operation of ICS. Advanced Persistent Threats (APT) are the most threatening attacks to ICS due to their long persistence and destructive cyber-physical effects to ICS. This paper considers a simulation of attackers and defenders of an ICS, where the defender must consider the cost-efficiency of implementing defensive measures within the system in order to create an optimal defense. The aim is to identify the appropriate deployment of a specific defensive strategy, such as defense-in-depth or critical component defense. The problem is represented as a strategic competitive optimisation problem, which is solved using a co-evolutionary particle swarm optimisation algorithm. Through the development of optimal defense strategy, it is possible to identify when each specific defensive strategies is most appropriate; where the optimal defensive strategy depends on the resources available and the relative effectiveness of those resources.

1 Introduction

Industrial Control Systems (ICS) are typically comprised of a set of supervisory control and data acquisition (SCADA) systems to control field actuators by monitoring the data of industrial processes. ICS can be found in various sectors of critical infrastructure. Disruption to such systems would lead to disastrous damage to the plants, environment and human health [26]. To promote efficient communication and high throughput, modern ICT technologies have been widely adopted into ICS, which makes them vulnerable targets for cyber criminals. ICS-CERT received 245 reports in 2014 by trusted asset owners, whilst there are still numerous incidents in critical infrastructure unreported [1].

Amongst the various cyber attacks against ICS, multi-stage Advanced Persistent Threats (APT) account for roughly 55% [1], and these are also the most threatening ones due to their long undetected persistence, sophisticated capabilities and destructive cyber-physical effects to ICS. We show a typical ICS architecture adapted from [26] in Fig. 1(a). In an APT attack, the attackers initially gain access to the target network, then propagate through the network by continuously exploiting chains of vulnerabilities of hosts, and eventually compromise operational field devices. A canonical example of

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

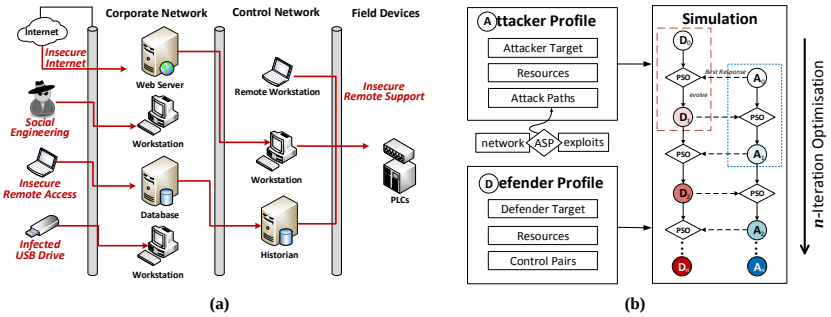


Fig. 1: (a) Typical ICS architecture threatened by APT attacks; (b)Simulation

such an attack is Stuxnet [6] acknowledged in 2010, which enabled cyber attacks to sabotage industrial plants. Stuxnet was introduced to the network by a removable flash drive, and propagated malware through the corporate and control network by exploiting zero-day vulnerabilities of hosts. Stuxnet eventually tampered the program controlling the field PLCs and disrupted the operation of ICS. According to the report [6] by Symantec, there were approximately 100,000 infected hosts across over 155 countries by September 2010. Another more recent example was reported by the German government[2] in December 2014. A cyber attack breached a steel mill through penetrating spear-phishing emails and resulted in massive damage to the whole plant.

In the wake of the increasing cyber attacks against ICS, the notion of *Defense-in-Depth* has been highly recommended as the best practice to protect critical infrastructures by numerous reports [26,14]. Defense-in-depth provides a multi-layer protection involving different security mechanisms such as a vulnerability management system, advanced firewalls with DMZ, intrusion detection, security awareness training and incident response. However, the high financial and managerial cost make defense-in-depth impractical and hard to fully implement [24]. Massive unnecessary efforts have been wasted on irrelevant attack vectors. Particularly smaller companies still struggle with finding the most cost-efficient way to deploy available controls. For this reason we look for alternative defensive strategies, and the most optimal implementation of them.

Most of the techniques involved at each stage of an APT can generally be defended by conventional security controls. A key question is how to allocate defensive resources and budget across the system to establish an effective protection against APT. In particular with limited low budget, our work here produces decision support tools to find *optimal defense strategies*, such as system-wide evenly spreading defense (i.e. Defense-in-depth) and focused defense on critical components. Unlike conventional analysis, we novelly consider the impact of varied *cost-efficiency* models of investments on deciding the most optimal defensive strategies, making the work more realistic and practical.

We use the notion of *attacker* to represent potential cyber attackers, and *defender* for the security manager who needs to deploy controls to protect an ICS. An attack graph is automatically generated by our logic-based reasoning engine, which chains various weaknesses of a given network that can be exploited by attackers. We model

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

the attacker and defender as a pair of competing agents and investigate their behaviours in a co-evolutionary process. *Particle Swarm Optimisation* (PSO) [11] is adopted to aid agents in finding the most optimal strategy to attack and defend under different circumstances. From this work, we discover that with limited low budget, the defensive effort should be generally focused on the critical targets rather than spreading over the system. However, when defending the critical assets becomes very inefficient, the most optimal strategies then favour defending other less valuable assets to form a defense-in-depth style strategy. The paper starts with a related work section where the work on attack modelling and agent-based co-evolutionary approaches are presented. The approach proposed in this paper is discussed in Section 3, which describes the modelling of the key elements and the development of the agent-based simulation. Three case studies extracted from the CSSP Recommended Defense-In-Depth Architecture [14] are described in Section 4 to demonstrate the effectiveness of our proposed tools. 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

A comprehensive introduction to the security issues of ICS is given in [26]. The generation of attack graphs has been studied extensively in the security community, and two of the most influential generators are MulVal [20] and NetSPA[16], both of which provide automatic generation of complete attack paths from scanned CVE vulnerabilities. [16] further provides a way of abstracting attack paths by classifying vulnerabilities in terms of CVE factors. Attack graphs have been widely applied to risk analysis. Noel et al. [19] measure the overall security of a network by simulating the propagation of multi-stage attacks and likelihoods of each single attack. Ma and Smith[17] provide a risk analysis for critical infrastructures to understand the impact of inter-dependency of CVE vulnerabilities on forming multi-step attack chains. This work particularly focuses on CVE vulnerabilities with effects of code execution and elevation of privileges as these vulnerabilities can serve as stepping-stone nodes to induce further attacks. In this paper, we generate attack graphs by common weaknesses of ICS, rather than specific vulnerabilities on each host. In this way, we lift our focus of defense to a more generic class of attacks, producing a more global view of deploying defense controls.

Antoine Lemay's 2013 thesis[15] presents an approach to defending SCADA components in electrical grids from APT style attacks. The author presents an Intrusion Detection System to protect ICS, with the aim of preventing the attacker from developing appropriate tools for exploiting the system. A more detailed view of the threats from APT models has been performed by Thomas Chopitea[2], and in a game theoretic manner by Pham and Cid[21]. One area of study that is of importance to this work is the concept of network hardening, which has been approached by studying vulnerabilities through attack graphs[18][28]. By studying the structure of networks and the vulnerabilities required to effectively exploit system, it is possible to identify the key areas where defensive measures are most effective, an approach that is taken in the representation of the problem in this work. Work undertaken by Fielder et. al.[7] uses a game theoretic approach to the optimal allocation of system administrator time to defensive

tasks. The results show that a greater emphasis of the limited administrator time should be placed on the most valuable assets, consistent with a critical component defense strategy. Extensive game theoretic work has been performed by Tambe and Kiekintveld looking at optimal security decisions for real-world scenarios using Stackelberg games. The work has covered scheduling of airport security[13], allocation of air marshals to flight paths[27] and border control[12]. The work has moved into cyber security, with a study of the use of honeypots[4]. While the origins of PSO was proposed by Kennedy and Eberhart in 1995 [5], a more up to date view of the field of PSO algorithms was presented by Poli, Kennedy and Blackwell in 2007 [23], which brought together many of the concepts developed over the previous 12 years. This overview was followed in a 2008 study by Poli [22], which identifies the application areas for PSO. This study identifies that very little work has gone into applying PSO algorithms to network security tasks, with only 1.3% of the literature covering the whole security field, with some work in security predictions [8], intrusion detection [25] and authentication [10].

3 Modelling and Simulation

The key components of the system are depicted in Fig.1(b). We define *Attacker Profile* and *Defender profile* to characterise attackers and defenders' behaviours and generate attack and defense strategies respectively. Attack strategies are decided by the attacking goals, available resources of the attacker and possible attack paths to launch and deploy the attack. The attack paths are generated by an automatic reasoning engine which uses a logic programming technique - *Answer Set Programming (ASP)* [9], given an established network and weaknesses in the network. For defenders, defensive preference of assets and available resource are the key to the decision making, as well as a set of control pairs defining the behaviours to form defense strategies. The most important part of the system is the agent-based simulation, as shown in the right part of Fig.1(b). In the simulation, attackers and defenders are modelled as a pair of competing agents, by which their behaviours are able to co-evolve to develop an optimal solution. PSO is adopted to encode each candidate strategy as a particle of a swarm and all such particles gradually move towards the best solution during each iteration of evolution.

3.1 Modelling and Representation

In this section, we introduce the representation of the pair of competing agents – *Attackers* and *Defenders*, in particular defining their possible behaviours and strategies. Other key components to establish the agent-based simulation are also presented.

We first represent a typical ICS architecture as a *network* graph, where each host or asset is identified as a *target* node $t_i \in \mathcal{T}$, a valid connection between a pair of targets is an *edge* $e \in \mathcal{E}$ and $\mathcal{E} := \mathcal{T} \times \mathcal{T}$. A compromised target produces certain gains for attackers $I^a : \mathcal{T} \rightarrow \mathbb{Z}^+$, while causing certain damages for defenders $I^d : \mathcal{T} \rightarrow \mathbb{Z}^-$. A sequence of single-step attacks constitutes an effective APT attack, and each single attack exploits a weakness of a targeted host. We define all such weaknesses as a set of ***attack methods*** $\mathcal{M} = \{m_1, \ldots, m_n\}$. We further distinguish different types of attack methods in terms of their origins: (i) *primary attacks:* gain initial access to the network,

such as social engineering and removable drives malware. (ii) *subsequent attacks:* penetrate further into the network, such as SQL-injection and LAN-based infection. (iii) *final attacks:* cause actual damage on the field devices.

An *attack path* p is derived by attaching an applicable attack method m to an edge, indicating a possible way to progress the attack from one target to another. $O(t_i)$ captures all outbound paths from the target t_i. Our reasoning engine can generate all such possible attack paths for a given network, which altogether render an *attack graph*. An example of such attack graph is given in Fig2(a). At each step of an APT, attackers probabilistically select an outbound attack path to exploit next.

Definition 1. *An* **attacker strategy** $a := \{(O(t_1), \Psi_1), \ldots, (O(t_n), \Psi_n)\}$*, where*

- $O(t_i) = [p_i^1, \ldots, p_i^k]$*, all outbound paths from the target* t_i*.*
- $\Psi_i = \{\psi_1, \ldots \psi_k\}$*, the probability distribution over* $O(t_i)$*,* $\sum_{j=1}^{k} \psi_j = 1,\ \psi_j \geqslant 0$*.*

The key task of *defenders* is to find a way of deploying defense controls with certain effectiveness to specifically combat APT-style attacks. Implementing a control with higher effectiveness generally requires more investment. We also define a cost-efficiency function in the general form of $\mathit{eff}_t(x) = \frac{ax}{bx+c}$ to compute the resulting effectiveness of a control at the target t with certain investment x, where $a, b, c \in \mathbb{N}$, $ax < bx + c$. The function will be instantiated to represent various cost-efficiencies in the examples later. Given a limited budget B, defenders need to decide the most cost-efficient way to deploy controls. Unlike attackers who have specific targets to stage attacks, defenders have to protect various targets across the network from numerous possible attacks and also stop the formation of APT.

Definition 2. *A* **defender strategy** $d := \{(c_1, x_1), \ldots, (c_m, x_m)\}$ *is a set of control pairs amongst the available controls* $\mathcal{C} = \{c_1, \ldots, c_m\}$*, and* x_i *denotes the number of allocated budget units to implement* c_i*, and* $\sum_{i=1}^{m} x_i \leqslant B$*, where* B *is the budget limit. The effectiveness of* c_i *implemented at the target* t *is then decided by a provided cost-efficiency function* $\mathit{eff}_t(x_i)$*.*

3.2 Simulation

We represent the problem of the calculation of optimal strategies as a competitive co-evolutionary process between an attacker and a defender. In this way we represent the problem as a system with two competing agents that aim to optimise their expected payoffs from either attacking or defending the system. In order to solve the co-evolutionary optimisation problem, we have developed a *Particle Swarm Optimisation (PSO)* algorithm. The PSO operates a number of rounds, where the attacker and defender attempt to create best response strategies to each other's actions. The PSO method creates an initial set of randomised strategies, called particles, for the attacker and the defender. The PSO algorithm explores the space of possible solutions by moving the particles to new positions in the search space aiming to find better solutions; this is done by applying a movement parameter to the particle, called a velocity. The velocity of a particle is a special form of the mixed strategy of the defender and the attacker, where the sum of all components must equal zero.

Each of the particles in the swarm must be evaluated to assess its performance; this is done by simulating the interactions between the two players in the environment.

Considering the nature of the kinds of attacks expected, the simulation represents the interactions of the players over a given fixed number of time steps. During each time step, the players have to make a series of decisions with regards to the actions that they perform, where the outcome of those actions are scored according to the amount of damage that successful attacks cause.

For each evaluation of the defense strategy, the defender sets a defense based on the mixed strategy, to prevent damage from an attacker who attempts to breach that defense over a period of time. At the start of each evaluation, the defender assigns the whole of the available budget to the system, where each target is assigned a portion of the budget based on the distribution defined by the strategy. The amount of units assigned to each control represents an amount of time and money used to protect the resource, with the effectiveness of the defensive measures based on the amount invested in defending them.

An attacker attacks through the system, starting from the node labelled EXT and attempts to advance through the network exploiting subsequent nodes until they have exploited a vulnerability on a device with no further connections in the network or the attack is halted by the defender. Specifically, the attacker selects an outward path from the current node based on their strategy and attempts to exploit a vulnerability on the connected node. If there is no defense assigned, then the attack is successful, however, if the defender has assigned defense to the node, then the attacker will exploit the vulnerability with probability $p_a > \mathit{eff}_t(v)$, where p_a defines the probability the attacker is able to successfully launch an attack and $\mathit{eff}_t(v)$ is the effectiveness of the defensive controls on node n, with an investment of v units from the budget.

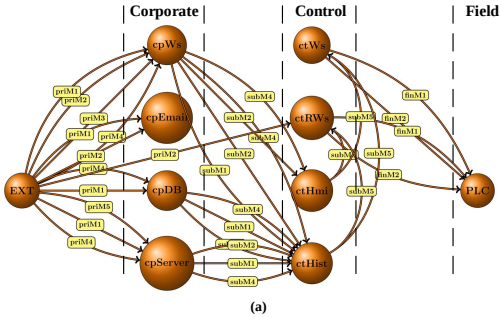
If the attacker is successful in exploiting a vulnerability, then they continue to attack selecting a further connecting node deeper in the system. In the event that there are no further outward connections, or the attack is halted by the defensive controls, then the attacker receives the maximum reward of the nodes exploited in the attack and the defender suffers damage in line with the value of that node.

4 Case Study and Experimental Settings

The network architecture of the case study is adapted from the common ICS architecture in [14][26], as shown in Fig. 2(a). It is a typical three-zone architecture of ICS, with *Corporate Network*, *Control Network* and *Field Devices*. The node *EXT* represents the external environment of the ICS. Four target nodes represent typical hosts in a corporate network and a control networks respectively. The rightmost node PLC symbolises the key control devices.

We also collect a set of attack methods, as listed in Fig. 2(b), from the *ICS Top 10 Threats and Countermeasures* 2014 [1] and *Common Cybersecurity Vulnerabilities in ICS* [3]. The corresponding set of defense controls is given in Fig. 2(c), derived from [1][14]. The attack methods countered by the controls are enumerated in the rightmost column of Fig. 2(c). A complete attack graph for the case study is then automatically generated by our ASP engine in Fig. 2(a). Each attack edge denotes a possible exploit of the system. In the example, control stations (*ctRWs* and *ctWs*) have direct access to PLCs, and thus all attacks aiming to PLCs have to pass through them. These control

workstations are not connected to any untrusted network, but can be infected by other hosts in the same network such as *ctHmi* and *ctHist*. Remote workstations (*ctRWs*) used for remote maintenance are threatened by the viruses infected by other external assets.



(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)

Fig. 2: Case study on ICS security management: (a) an attack graph; (b) Common ICS attack methods; (c) Common defense controls.

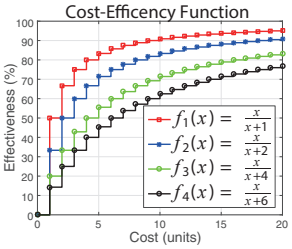
Furthermore, we provided four different cost-efficiency functions in Fig.3(b) to estimate the effectiveness of controls implemented by assigned budget. All functions constraint the effectiveness of controls as $\mathit{eff}(v) < 100\%$ by considering any potential zero-day attacks that cannot be defended by any implemented controls. Besides, as the investment increases, the improvement of the effectiveness gradually becomes minimum. Brief introductions to the three cases are given in Fig.3(a). The main differences amongst the three cases are the application of cost-efficiency functions for different zones, where eff_{cp} denotes the function adopted for all targets in the *Corporate Zone*, eff_{ct} for the *Control Zone* and eff_{d} for the *Field Zone*. Case 1 presents a standard ICS architecture with unique characteristics in each zone, and thus variable functions are applied [26]. A typical example of such ICS is Distributed Control Systems (DCS) that has high requirements on timeliness, availability and with very constrained resource, all of which make the defensive efforts in control and field layer less efficient. Unlike Case 1, Case 2 provides a comparative scenario where most IT and ICS components are located in business and commercial facilities with particular emphasis on gathering data by SCADA and hence a uniform function is adopted for all zones [26]. The last case further investigates the optimal allocation on controls to protect an isolated or remote distributed ICS system requiring extensive effort to deploy controls in *Field Zone*.

To run the simulations, we fixed a number of parameters that relate to the operation of the PSO Algorithm. The size of the swarm used was set at 100, which was large enough to reasonably represent the search space. This is given that the simulation happens over 50 moves per particle and for 50 generations of competition between the two agents. The weighting 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. The simulation operated over 20 time steps for the attacker and defender strategies and a particle would be evaluated 30 times to reduce variance from the non-deterministic nature of the simulated environment.

In the next section, we show the simulation results on the optimal defense strategies for all three cases. For the optimal solutions, we firstly analyse the variance of the

	Cost-efficiency Functions
Case 1	Scaled model for each zone: $\{eff_{cp} = f_1,\ eff_{ct} = f_2,\ eff_d = f_3\}$
Case 2	Uniform model for each zone: $\{eff_{cp} = f_1,\ eff_{ct} = f_1,\ eff_d = f_1\}$
Case 3	Scaled Low-efficient model for each zone: $\{eff_{cp} = f_1,\ eff_{ct} = f_2,\ eff_d = f_4\}$

(a)



(b)

Fig. 3: (a) Case Study Settings; (b) Cost-efficiency functions of investments

average expected damage per attack in Fig.4. Then we particularly look at the resource assignment on the critical component PLC in Fig.5 to find out the strategic change of defender from focused defense to defense-in-depth, and provide a comparison of both competing strategies.

5 Results

Fig. 4 shows the average expected damage per attack from an indifferent attacker with the three cost-efficiency models for defense.

The most noticeable result is the difference in expected damage between the uniform cost-efficiency and both forms of the scaled cost-efficiency. At low budget levels the difference in expected damage between the uniform and scaled methods is approximately 300. With more than 6 units of defense, this difference is reduced to an approximate range of 150-200. This reduction in difference is representative in a change in policy, where the defender switches from a heavier focus on protecting the PLC to performing a defense-in-depth strategy. By operating a similar strategy to those with the scaled cost-efficiency model, the difference in expected damage becomes equivalent to the difference in the efficiency. With the uniform cost-efficiency, we see that the first 5 units provide the most benefit of defense. The first unit reduces the damage by approximately 450 units, with the next 4 units reducing the damage by an average of 84 each. After 5 units, the average net gain in defense is reduced to 10 damage per unit. In contrast, for the scaled methods, the first unit for defense has a lower benefit, providing only a reduction in average expected damage by approximately 110. For the scaled model the next 4 units have a similar impact on the defense as the uniform model, reducing the expected damage by an average of 86 each, however this reduction is only 65 for the low cost-efficiency model. However each unit of defense after the fifth, reduces the expected damage by 24 for the normal scaled and 28 for the scaled with low cost-efficiency.

We see this change in the reduction of damage, because the effectiveness of the defense per unit placed on a single target is lower for every unit after the first. This means that after an initial investment to protect an asset, each further investment has a lower impact. For the uniform model, the initial investment of a single unit to protect the PLC is the most significant investment, with the next 4 units able to help protect the rest of

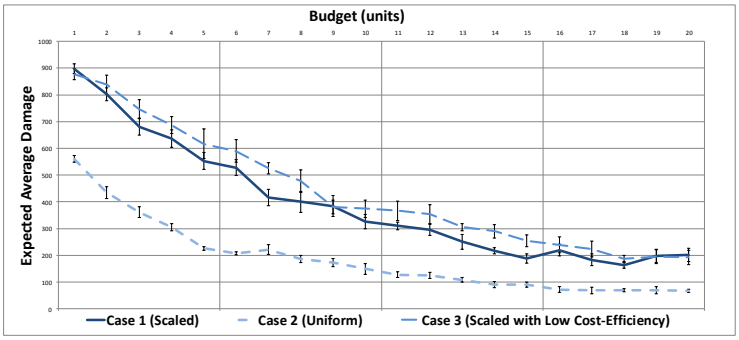


Fig. 4: Average Expected System Damage against the available defense budget

the system. Beyond this point each additional unit is used to protect a less risky component or add defense to an already defended component. For the scaled methods, this is not as consistent, since the first unit has a lower initial impact outside the assets at the corporate level. To identify the issue surrounding the type of defense that accounts for the optimal solutions, we need to identify the strategy of defense for the critical component, in this case the PLC. The graph presented in Fig. 5, shows the representation of the probability of defending PLC in the optimal solution across each of the runs. The uniform cost-efficiency method shows that the first unit of defense is almost always placed on the PLC, since there is no strategy that better defends the system. As the number of units increases, the probability of placing defense on the PLC decreases, but maintains the probability that at least 1 unit should be dedicated to the PLC. From 7 units, the defense tends towards a preference of defense-in-depth, with only 30% of the budget being allocated for the PLC. The remaining 70% of the budget is then split amongst all other controls, with an emphasis on the control layer. This is because with a larger budget available, it is no longer cost-efficient to focus all the defense on the PLC, but instead spread the defense to limit not only the damage to the PLC, but protect more of the network.

Unlike the uniform model, the scaled models initially reject placing the very limited resource in a place where it has the lower potential impact, which is at the PLC, favouring a more aggressive defense that has a higher variance. However, we see that as the budget increases, the ability to cover the PLC with some effectiveness is greatly increased, and so the scaled models adapt to this and assign resources accordingly. The low cost-efficiency scaled model assigns on average 13% less resources to the PLC than the standard scaled model, this is due to the lower impact that each unit has when applied to the node, with the initial uptake of non-incidental protection for the PLC occurring later. While the results may show that in general the optimal strategy for each of the cost-efficiency strategic tested converges to $0.175 < x_{\text{PLC}} < 0.25$ with a larger budget, the variance in those results indicates that there are a number of optimal strategies. It is accounted for in the high variance of the results associated with the implementation of defense on the PLC. We compare the outcomes of two different solutions in Table 1, which shows two competing strategies for the defender getting the same approximate outcome of 765. In one case minimal emphasis is placed on the PLC, whereas the other strategy implements defense at the PLC with a probability of 0.306 per unit of defense.

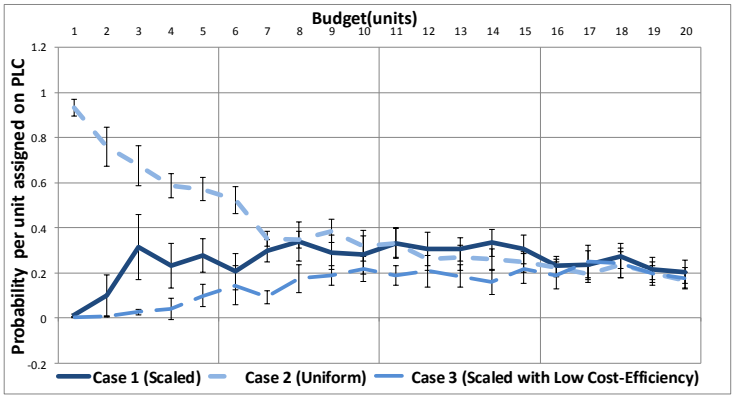


Fig. 5: Probability per unit assigned to PLC against the available budget

ctWs	*ctRWs*	*ctHist*	*cpDB*	*cpEmail*	*PLC*	*ctHmi*	*cpWs*	*cpServer*
0.320	0.008	0.634	0.010	0.002	0.008	0.005	0.006	0.007
0.009	0.104	0.011	0.007	0.009	0.306	0.007	0.541	0.007

Table 1: Comparison of Strategies with a Scaled Cost-efficiency and a Budget of 2

6 Discussion

One of the issues is what happens if the decision maker gets the cost-efficiency model wrong. When defining the optimal defense, the cost-efficiency model dictates how much effort should be placed onto any individual asset and so overestimating the cost-efficiency could prove disastrous. An overestimation could create a scenario where the defender believes they have one cost-efficiency model, but in reality they are operating under a different model. From the perspective of the results presented in Fig.4, that for a budget of 4, the average expected damage for a uniform model is approximately 305, with the scaled cost-efficiency only able to offer the same defense with a budget of 12. This means that in order to get the same coverage that was believed at a budget of 4, the defender would have to invest 3 times the amount. Additionally if the defender were to implement an optimal strategy from the uniform cost-efficiency with a scaled cost-efficiency model, the average expected damage would increase to 639, which is in the same range as the optimal strategy for the same budget under the correct cost-efficiency model. However, we see that there is a 109% increase in damage from the expected damage using the uniform model. While we have considered the impact of not as strongly defending the PLCs through a reduction in cost-efficiency, the results show that the defender should still place some emphasis on the PLC. However we know that there are some systems that have very restricted operational capacity, mainly concerning field controllers. To represent this within the model, we ran a special case of the variable cost-efficiency, where the cost-efficiency was set at $\frac{x}{x+14}$. With this low cost-efficiency, we see that the emphasis on protecting the PLC at high levels has an average probability of 0.08, where the emphasis of the defense is split across the rest of the devices with a higher emphasis placed on those in the the more valuable control layer.

7 Conclusions

In this work, we have developed a model studying the cost-efficiency of investments for defending components on a network. The model simulates an attacker attempting to breach a system against a probabilistic defense assignment of the defender. Using a particle swarm optimisation algorithm to simulate the behaviour of two agents, we have been able to identify the optimal strategy of a system defender in an ICS environment. The results show that as the cost-efficiency of protecting the most vulnerable node in a network decreases, the uptake of defense-in-depth style defenses increases.

To extend this work, we want to better represent the resources and decision making of the attacker. In this current model, we focus on the defender and the defense decisions, but in an APT style attack, the attacker is a more active player than currently considered. The attacker needs to better consider the paths and methods of defense utilised, where exploring the system is an action that is constrained by time. To represent this a proposed expansion of the model, would be to view the transitions between nodes as an event that occurs over time, creating a scenario, where the attacker must balance attacking the quickest paths against the defender's optimal strategy, that would ideally protect those paths more.

References

1. BSI. Industrial control system security top 10 threats and countermeasures 2014, mar 2014. "www.allianz-fuer-cybersicherheit.de/ACS/DE/_downloads/techniker/hardware/BSI-CS_005E.pdf".
2. T. Chopitea. Threat modelling of hacktivist groups organization, chain of command, and attack methods, 2012. "http://publications.lib.chalmers.se/records/fulltext/173222/173222.pdf".
3. U. S. Department of Homeland Security. Common cybersecurity vulnerabilities in industrial control systems, 2011. "www.ics-cert.us-cert.gov/sites/default/files/documents/DHS_Common_Cybersecurity_Vulnerabilities_ICS_20110523.pdf".
4. K. Durkota, V. Lisy, C. Kiekintveld, and B. Bosansky. Game-theoretic algorithms for optimal network security hardening using attack graphs. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1773–1774, 2015.
5. R. C. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science*, volume 1, pages 39–43. New York, NY, 1995.
6. N. Falliere, L. O. Murchu, and E. Chien. W32. stuxnet dossier. *White paper, Symantec Corp., Security Response*, 5, 2011.
7. A. Fielder, E. Panaousis, P. Malacaria, C. Hankin, and F. Smeraldi. Game theory meets information security management. In *ICT Systems Security and Privacy Protection*, pages 15–29. Springer, 2014.
8. K. Gao, L. Jianming, R. Xu, Y. WANG, and Y. LI. A hybrid security situation prediction model for information network based on support vector machine and particle swarm optimization. *Power System Technology*, 4:033, 2011.
9. M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and M. Schneider. Potassco: The Potsdam answer set solving collection. *AI Communications*, 24(2):107–124, 2011.

10. M. Karnan and M. Akila. Personal authentication based on keystroke dynamics using soft computing techniques. In *Communication Software and Networks, 2010. ICCSN'10. Second International Conference on*, pages 334–338. IEEE, 2010.
11. J. Kennedy. Particle swarm optimization. In *Encyclopedia of Machine Learning*, pages 760–766. Springer, 2010.
12. R. Klíma, V. Lisỳ, and C. Kiekintveld. Combining online learning and equilibrium computation in security games. In *Proceedings of the 2015 International Conference on Decision and Game Theory for Security*, 2015.
13. D. Korzhyk, V. Conitzer, and R. Parr. Complexity of computing optimal stackelberg strategies in security resource allocation games. In *AAAI*, 2010.
14. D. Kuipers and M. Fabro. *Control systems cyber security: Defense in depth strategies*. United States. Department of Energy, 2006.
15. A. Lemay. *Defending the SCADA Network Controlling the Electrical Grid from Advanced Persistent Threats*. PhD thesis, École Polytechnique de Montréal, 2013.
16. R. P. Lippmann, K. W. Ingols, C. Scott, K. Piwowarski, K. J. Kratkiewicz, M. Artz, and R. Cunningham. *Evaluating and strengthening enterprise network security using attack graphs*. Defense Technical Information Center, 2005.
17. Z. Ma and P. Smith. Determining risks from advanced multi-step attacks to critical information infrastructures. In *Critical Information Infrastructures Security*, pages 142–154. Springer, 2013.
18. S. Noel, S. Jajodia, B. O'Berry, and M. Jacobs. Efficient minimum-cost network hardening via exploit dependency graphs. In *Computer Security Applications Conference, 2003. Proceedings. 19th Annual*, pages 86–95. IEEE, 2003.
19. S. Noel, S. Jajodia, L. Wang, and A. Singhal. Measuring security risk of networks using attack graphs. *International Journal of Next-Generation Computing*, 1(1):135–147, 2010.
20. X. Ou, W. F. Boyer, and M. A. McQueen. A scalable approach to attack graph generation. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 336–345. ACM, 2006.
21. V. Pham and C. Cid. Are we compromised? modelling security assessment games. In *Proceedings of the 2012 International Conference on Decision and Game Theory for Security*, pages 234–247. Springer, 2012.
22. R. Poli. Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Applications*, 2008:3, 2008.
23. R. Poli, J. Kennedy, and T. Blackwell. Particle swarm optimization. *Swarm intelligence*, 1(1):33–57, 2007.
24. P. E. Small. Defense in depth: An impractical strategy for a cyber world. *SANS Institute, Bethesda*, 2011.
25. S. Srinoy. 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*, pages 186–192. IEEE, 2007.
26. K. Stouffer, J. Falco, and K. Scarfone. Guide to industrial control systems (ics) security. *NIST special publication*, 2011. "`http://csrc.nist.gov/publications/nistpubs/800-82/SP800-82-final.pdf`".
27. J. Tsai, S. Rathi, C. Kiekintveld, F. Ordez, and M. Tambe. *IRIS - A tool for strategic security allocation in transportation networks*, volume 2, pages 1327–1334. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 1 2009.
28. L. Wang, S. Noel, and S. Jajodia. Minimum-cost network hardening using attack graphs. *Computer Communications*, 29(18):3812–3824, 2006.