Security Stress: Evaluating ICT Robustness Through a Monte Carlo Method

Fabrizio Baiardi[1(✉)], Fabio Corò[1], Federico Tonelli[1], Alessandro Bertolini[1], Roberto Bertolotti[1], and Luca Guidi[2]

[1] Dipartimento di Informatica, Università di Pisa, Pisa, Italy
{baiardi,fcoro,tonelli}@di.unipi.it
[2] ENEL Ingegneria e Ricerca SpA, Pisa, Italy
luca.guidi@enel.com

Abstract. The security stress is a synthetic evaluation of how an ICT infrastructure resists to attacks. We define the security stress and show how it is approximated through the Haruspex suite. Then, we show how it supports the comparison of three versions of an industrial control system. Haruspex is a suite of tools that apply a Monte Carlo method and support a scenario-based assessment where in each scenario intelligent agents compose attacks to reach some predefined goals.

Keywords: Risk assessment · Intelligent agent · Robustness

1 Introduction

We consider the risk assessment of an ICT infrastructure under attack by intelligent agents that achieve some predefined goals through complex attacks, e.g. sequences of attacks. A complex attack escalates the privileges, e.g. access rights, of an agent till it owns all the privileges in one of its goals.

The *security stress* is a synthetic evaluation of how an infrastructure resists to the agents. This measure can assess an infrastructure or support the comparison of alternative infrastructures from a robustness perspective. Given an agent and a goal, the security stress plots, for each time t, the probability that the agent reaches the goal within t. We refer to the curve as a stress one because it shows how the infrastructure resists to the force due to an agent for increasing times. After discussing the security stress and its approximation through the Haruspex suite, we generalize it to any number of agents and of goals.

This paper is structured as follows. Section 2 briefly reviews the Haruspex suite and security metrics. Section 3 introduces the stress curve and shows how it supports the comparison of distinct systems. Section 4 use the stress to compare three versions of an industrial control system. Lastly, we draw some conclusions.

2 Related Works

We briefly recall the Haruspex suite and related works on metrics to evaluate the robustness of an infrastructure.

© Springer International Publishing Switzerland 2016
C.G. Panayiotou et al. (Eds.): CRITIS 2014, LNCS 8985, pp. 222–227, 2016.
DOI: 10.1007/978-3-319-31664-2_23

The tools in the Haruspex suite support the risk assessment of an ICT system by applying a Monte Carlo method to simulate a scenario where some intelligent, goal oriented agents attack the system. [1,2] outline the tools of the suite to build the models of interest and apply the Monte Carlo method. Three tools are the kernel of the suite: the *builder*, the *descriptor* and the *engine*. The first two tools build models of, respectively, the system and an agent. The *engine* uses these models to simulate the agent attacks. This tool builds a statistical sample to support an assessment by applying a Monte Carlo method and collecting a sample in each simulation.

The metrics in [3–5] evaluate the robustness of an ICT infrastructure under attack without integrating the proposed metrics with the simulation of the attacks. The metric in [6] is focused on the discovery of zero-day vulnerabilities. [7–9] review alternative security metrics. [12] is similar to security stress as it considers the amount of work to attack a system.

3 Security Stress of an ICT Infrastructure

The stress $Str^S_{ag,g}(t)$ at t of an infrastructure S is the cumulative probability distribution that the agent ag reaches g within t. Being a probability distribution, $Str^S_{ag,g}(t)$ is monotone non decreasing in t and $Str^S_{ag,g}(0) = 0$.

To justify the adopted definition, let us denote by t_0 the lowest time where $Str^S_{ag,g}(t)$ is larger than zero and by t_1 the time, if it exists, where it is equal to 1. If we consider ag as a force aiming to change the shape of S, then this force is ineffective till t_0. Then the shape of S changes due the attacks of ag and S cracks after t_1, because ag is always successful for larger times. $t_1 - t_0$ evaluates how long an infrastructure can, partially, resist to the attacks of ag before cracking.

We believe $Str^S_{ag,g}$ is a proper synthetic evaluation of the robustness of S because its shape is related to several attributes of S. t_0 depends upon both the time to execute an attack and the length of the shortest complex attacks to reach g. For each attack at in the sequences to achieve g, t_1 depends upon $succ(at)$ that determines the average number of executions of at. $t_1 - t_0$ depends upon both the standard deviation of the number of attacks to reach ag and their success probabilities. Because of these relations, $Str^S_{ag,g}(t)$ returns a more accurate evaluation of the robustness of S than metrics that consider just one value, such as the average time or the average number of attacks to reach g.

A lower bound on t_0 is the minimum of the set produced by mapping each complex attack ag can implement to reach g into the sum of the execution times of its attacks. This is the best case for ag where no attack fails. Computing this bound is not trivial because the size of the set increases exponentially in the number of the components of S [2].

To evaluate the robustness of S in a predefined time interval, we plot $Str^S_{ag,g}(t)$ in the considered interval. Obviously, $Str^S_{ag,g}(t)$ may be lower than 1 in the interval.

To generalize $Str^S_{ag,g}$ to a set of goal Sg, we assume that ag stops its attacks after reaching any goal in Sg. Under this assumption, $Str^S_{ag,Sg}(t)$ is the probability that ag is idle after t. To generalize to a set of agents Sag, we consider

the most dangerous agent in Sag. This is the agent, if it exists, with the highest stress curve. As an alternative, $Str^S_{ag,g}(t)$ is the weighted sum of the stress due to each agent in Sag where the weigh of an agent evaluates its contribution to the overall impact. Further generalizations are possible but, in this paper, we focus on one agent aiming to achieve any goal in a predefined set.

$Str^S_{ag,g}$ is the inverse of a survival function [10] as it plots the probability of a success of ag instead than the one that S survives ag attacks.

We approximate $Perc^S_{ag,g}(t)$ as the percentage of samples collected in an *engine* experiment where ag reaches g before t. The experiment simulates ag for the time interval of interest.

3.1 Supporting a Comparison

To evaluate the relative fragility of two infrastructures, $S1$ and $S2$, with respect to an agent ag trying to achieve a goal we analyze $Str^{S1}_{ag,g1}(t)$ and $Str^{S2}_{ag,g2}(t)$. $g1$, the goal of ag in $S1$, may differ from $g2$, the goal in $S2$, because the same high level goal, e.g. read an information, may involve distinct rights in the two infrastructures.

The two stress curves show the time interval when an infrastructure better resists to the attack of ag, e.g. it has a lower stress. We say that $S1$ is more robust, or less fragile, than $S2$ if $Str^{S1}_{ag,g1}$ is always lower than or equal to $Str^{S2}_{ag,g2}(t)$, e.g. $Str^{S1}_{ag,g1}$ lies in the space bounded by $Str^{S2}_{ag,g2}$. This implies that, at any time, the amount of deformation in $S1$ is always lower than in $S2$. This condition is violated if $0 < Str^{S1}_{ag,g1}(t_x) = Str^{S2}_{ag,g2}(t_x) < 1$ for some t_x. However, even this comparison may return useful information. Suppose that, initially, $Str^{S1}_{ag,g1}$ is lower than $Str^{S2}_{ag,g2}$ but then two curves cross. In other words, initially, the deformation in $S1$ is lower than in $S2$ but, for values of t larger than t_x, the situation changes. This happens when ag can reach its goal in $S1$ only through complex attacks that require a long time either because they compose a large number of elementary attacks and/or because the time of to implement these elementary attacks is large. Hence, the lowest time to successfully attack $S1$ is larger than for $S2$ but, if all the success probabilities of the attacks against $S1$ are close to 1, the difference $t_1 - t_0$ will be small and $S1$ will quickly crack provided that ag has enough time available. The slower increase of $Str2$ may be due to the lower success probabilities of attacks against $S2$.

4 Comparing Distinct Version of an Infrastructure

This section applies the stress to compare three versions of a system to supervise and control power generation that is segmented into four types of subnets: Central, Power Context, Process and Control. Users of the intranet run the business processes of power generation through the nodes in a Central subnet. The plant operators interact with the SCADA servers through the nodes in a Power Context subnet. The SCADA servers and the systems to control power production

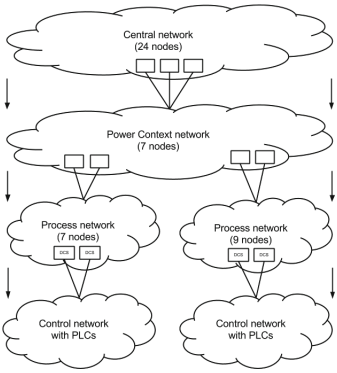

Fig. 1. First version of the infrastructure

belongs to a Process network. Finally, the PLCs in a Control subnet control the devices in the plant.

Figure 1 shows the first system version [11] with 49 nodes segmented into six subnet. The Central subnet includes 24 nodes, the Power Context includes 7 nodes. Then, Process subnet 1 and 2 include, respectively, 9 and 7 nodes. Each Process subnet is connected to a Control subnet with a PLC device. Three nodes of the Central subnet have a connection with the Power Context subnet. Two pairs of nodes in the Power Context network are connected to nodes in one Process subnet, Lastly, two nodes in each Process subnet are connected to the corresponding Control subnet.

We compare this version against two other ones. In the first one, we double the number of nodes by replicating each node without altering the number of connections between subnets. Also the third version includes 98 nodes as the second one, but the Central subnet is segmented into two subnets with 24 nodes each. Furthermore, all the nodes connected to the Power Context subnets belong to just one of the resulting subnets, as in Fig. 1. We consider four classes of agents and assume that any agent initially owns some rights on a node in the Central subnet and it aims to control the PLC devices. In particular, agent in the first class, $T1$, aim to control both devices and those in the second class, $T2$, aims to control any of the devices. Agents in the two last classes, $T3$ and $T4$, aim to control a distinct PLC device. Agents need to scan each node to discover its vulnerabilities. To cover alternative strategies to select the complex attack to a goal, each class includes seven agents. For each version, Figs. 2, 3, and 4 show the stress curves of the most dangerous agent in each class. The figures show that in the first version, the most dangerous agent in the $T2$ class reaches its goal in about twelve hours while an agent of another class reaches its goal in about fourteen hours, i.e. about two hours later. In the second version, the most dangerous agent belongs to the $T2$ class and it reaches its goal in about 21 h. Other agents take one more hour. Then, in the third version of the infrastructure, the time to reach the goal is a bit larger than in the second one. Indeed, the

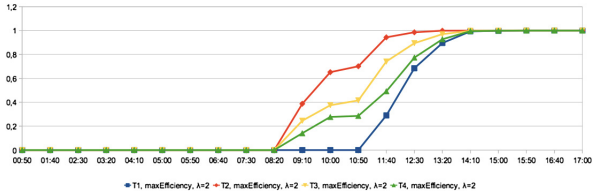

Fig. 2. First version: stress curve of the most dangerous agents

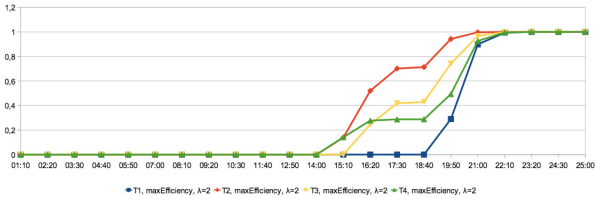

Fig. 3. Second version: stress curve of the most dangerous agents

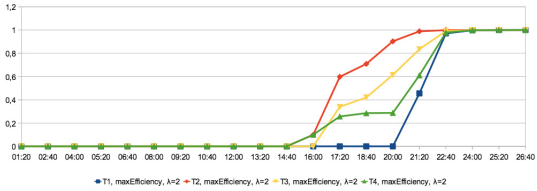

Fig. 4. Third version: stress curve of the most dangerous agents

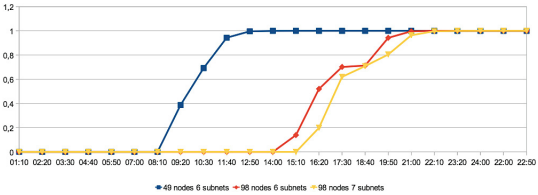

Fig. 5. Robustness of the three versions

class $T2$ agent reaches the goal only 20 min later than in the second version but the remaining agents reach their goal after more than two hours. Lastly, Fig. 5 compares the robustness of the three versions. Each curve refers to the most dangerous agent for the considered version. As expected, the first version is the most fragile one because its number of nodes reduces the number of attacks to reach a goal. The number of nodes in the second version confuses the agents and increases the time to reach their goal. Finally, the third version is the least fragile one because the larger numbers of nodes and of subnets increase the number of attacks and the time to reach a goal.

5 Conclusion

The stress curve is a synthetic evaluation of the robustness of an infrastructure with respect of complex attacks by intelligent agent. It simplifies the comparison of distinct infrastructures or of alternative versions of the same one and it is approximated through the Haruspex suite. We have applied this measure to compare three versions of an infrastructure and discussed how the stress curve changes according to the number of nodes or of subnets.

References

1. Baiardi, F., Sgandurra, D.: Assessing ict risk through a monte carlo method. Environ. Syst. Decisions **33**(4), 1–14 (2013)
2. Baiardi, F., Corò, F., Tonelli, F., Guidi, L.: Gvscan: Scanning networks for global vulnerabilities. In: First International Workshop on Emerging Cyberthreats and Countermeasures, Regensburg, Germany (2013)
3. Vaughn Jr., R.B., Henning, R., Siraj, A.: Information assurance measures and metrics - state of practice and proposed taxonomy. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences, 2003, p. 10 (2003)
4. Schudel, G., Wood, B.: Adversary work factor as a metric for information assurance. In: Proceedings of the 2000 Workshop on New Security Paradigms. NSPW 2000, pp. 23–30. ACM, New York (2000)
5. Langweg, H.: Framework for malware resistance metrics. In: 2nd ACM Workshop on Quality of Protection, pp. 39–44. ACM, New York (2006)
6. Wang, L., Jajodia, S., Singhal, A., Cheng, P., Noel, S.: k-zero day safety: A network security metric for measuring the risk of unknown vulnerabilities. IEEE Trans. Dependable Sec. Comput. **11**(1), 30–44 (2014)
7. Jaquith, A.: Security Metrics: Replacing Fear, Uncertainty, and Doubt. Addison-Wesley Professional (2007). ISBN:0321349989
8. Payne, S.C.: A guide to security metrics. SANS Institute (2006)
9. Swanson, M.: Security metrics guide for information technology systems. Technical report, NIST, US Department of Commerce (2003)
10. La Corte, A., Scatà, M.: Failure analysis and threats statistic to assess risk and security strategy in a communication system. In: ICSNC 2011, The Sixth International Conference on Systems and Networks Communications, pp. 149–154 (2011)
11. Nai Fovino, I., Masera, M., Guidi, L., Carpi, G.: An experimental platform for assessing scada vulnerabilities and countermeasures in power plants (2010)
12. Pamula, J., Jajodia, S., Ammann, P., Swarup, V.: A weakest-adversary security metric for network configuration security analysis. In: Proceedings of the 2nd ACM Workshop on Quality of Protection. QoP 2006, pp. 31–38. ACM, New York (2006)