

Towards Visualising Security with Arguments

Stefano Bistarelli^{1,2}, Fabio Rossi¹, Francesco Santini^{2,*}, and Carlo Taticchi¹

¹ Dipartimento di Matematica e Informatica, Università di Perugia, Italy
[bista,rossi,taticchi]@dmi.unipg.it

² IIT-CNR, Pisa, Italy

francesco.santini@iit.cnr.it

Abstract. Abstract Argumentation has been proved as a simple yet powerful approach to manage conflicts in reasoning with the purpose to find subsets of “surviving” arguments. Our intent is to exploit such form of resolution to visually support the administration of security in complex systems. For instance, in case threat countermeasures are in conflict (also with assets) and only some of them can be selected.

1 Introduction and Motivations

An *Abstract Argumentation Framework (AAF)*, or System, as introduced in a seminal paper by Dung [6], is simply a pair $\langle A, R \rangle$ consisting of a set A whose elements are called arguments and of a binary relation R on A , called “attack” relation. An abstract argument is not assumed to have any specific structure but, roughly speaking, an argument is anything that may attack or be attacked by another argument. The sets of arguments (or *extensions*) to be considered are then defined under different semantics, which are related to varying degrees of scepticism or credulousness.

In this work, our goal is to start developing a tool to visualise security threats and related countermeasures as arguments, as if security was a continuous dynamic discussion between the administrator and the surveilled system. Existing automated tools to defend a system from such security threats are one potential solution, but a completely automated approach could undervalue the strong analytic capabilities of humans, particularly in problematic situations that require vigilant human oversight.

We measure the strength of subsets of arguments and single arguments in accordance with Argumentation Theory. We print such strength degrees in different colours with the purpose to immediately catch the attention of the Security Administrator on what is going on in his system, and help him to take a decision on the set of countermeasures to be considered.

2 Preliminaries

In this section we briefly summarise the background information related to classical Abstract Argumentation Frameworks (AAF) [6].

* The author is supported by MIUR PRIN 2010XSEMLC “Security Horizons”.



Fig. 1: An example of AAF.

Definition 1 (AAF). An Abstract Argumentation Framework (AAF) is a pair $F = \langle A, R \rangle$ of a set A of arguments and a binary relation $R \subseteq A \times A$, called the attack relation. $\forall a, b \in A$, aRb (or, $a \rhd b$) means that a attacks b . An AAF may be represented by a directed graph whose nodes are arguments and edges represent the attack relation. A set of arguments $S \subseteq A$ attacks an argument a , i.e., $S \rhd a$, if a is attacked by an argument of S , i.e., $\exists b \in S. b \rhd a$. An argument $a \in A$ is defended (in F) by a set $S \subseteq A$ if for each $b \in A$, such that $b \rhd a$, also $S \rhd b$ holds.

Argumentation semantics [6] characterise a collective “acceptability” for arguments. Respectively, *adm*, *com*, *prf*, and *stb* stand for admissible, complete, preferred, and stable semantics.

Definition 2 (Semantics [6]). Let $F = \langle A, R \rangle$ be an AAF. A set $S \subseteq A$ is conflict-free (in F), denoted $S \in cf(F)$, iff there are no $a, b \in S$, such that $a \rhd b$ or $b \rhd a \in R$. For $S \in cf(F)$, it holds that

- $S \in adm(F)$, if each $a \in S$ is defended by S ;
- $S \in com(F)$, if $S \in adm(F)$ and for each $a \in A$ defended by S , $a \in S$ holds;
- $S \in prf(F)$, if $S \in adm(F)$ and there is no $T \in adm(F)$ with $S \subset T$;
- $S \in stb(F)$, if for each $a \in A \setminus S$, $S \rhd a$;

We also recall that the requirements in Def. 2 define an inclusion hierarchy on the corresponding extensions, from the most to the least stringent: $stb(F) \subseteq prf(F) \subseteq com(F) \subseteq adm(F)$. Moreover, $\sigma(F) \neq \emptyset$ always holds for each considered semantics σ (except for the stable one).

Definition 3 (Arguments acceptance-state). Given one of the semantics σ in Def. 2 and a framework F , an argument a is i) sceptically accepted if $\forall S \in \sigma(F), a \in S$, ii) *a* is credulously accepted if $\exists S \in \sigma(F), a \in S$ and *a* is not sceptically accepted, and iii) *a* is rejected if $\nexists S \in \sigma(F), a \in S$.

Consider $F = \langle A, R \rangle$ in Fig. 1, with $A = \{a, b, c, d, e\}$ and $R = \{a \rhd b, c \rhd b, c \rhd d, d \rhd c, d \rhd e, e \rhd e\}$. In F we have $adm(F) = \{\emptyset, \{a\}, \{c\}, \{d\}, \{a, c\}, \{a, d\}\}$, $com(F) = \{\{a\}, \{a, c\}, \{a, d\}\}$, $prf(F) = \{\{a, d\}, \{a, c\}\}$, and $stb(F) = \{\{a, d\}\}$. Hence, argument a is sceptically accepted in $com(F)$, $prf(F)$ and $stb(F)$, while it is only credulously accepted in $adm(F)$.

3 A Visualisation Example

Consider a small research and development company. This company cooperates with other (often large) enterprises for the development of complex goods. Such

company possesses high-tech knowledge which has to be protected from competitors. The company needs to efficiently use its resources with the purpose to survive in a highly competitive market. In short, the company has the goal (i.e., asset) of ensuring the productivity of operations (QoS).

In this small example, the security-system administrator has identified the following threats and related security controls (in square brackets): hacker penetration (HP) [host IDS (HI), network IDS (NI)] (where IDS stands for Intrusion Detection System), employee abuse (EA) [monitoring functionality (MF), audit procedures (AP)], and compromise of communication channel (CCC) [virtual private network (VPN), encrypted line (EL)].

We would like to emphasise that abstract arguments have no internal structure, and are not “directly linked” to classical logic. For this reason, we can consider multiple sources of information but and belief, such as case law, common sense, and expert opinion. We can consider information coming from multiple network-sensors, in the form of logs, warnings, and errors. Facts and beliefs can be also taken from internal policy documents, and standard documents as well. For instance the *Standard of Good Practice for Information Security*, is a business-focused, practical and comprehensive guide to identifying and managing information security risks in organizations and their supply chains. The 2011 Standard is aligned with the requirements for an Information Security Management System (ISMS) set out in ISO/IEC 27000-series standards, and provides wider and deeper coverage of ISOIEC 27002³ control topics, as well as cloud computing, information leakage, consumer devices and security governance.

To work on our example we use *SecArg*⁴ (Security with Arguments). *SecArg* is based on *ConArg* [4, 5] (ARGumentation with CONstraints), which is an Abstract Argumentation reasoning-tool using the *Gecode* library⁵, an efficient C++ environment where to develop constraint-based applications. The input (text) file passed to *SecArg* contains the list of arguments partitioned into *countermeasures*, *threats*, *assets*, and attacks between them: for instance, *countermeasure(HI)*, *threat(HP)*, *att(HI,HP)* (hacker penetration is prevented by a host IDS). *SecArg* visually represents the different nature of arguments with different colours: green for countermeasures, red for threats, and yellow for assets.

A more extended example is represented in Fig. 2. In such AAF we have that executing a host IDS and a monitoring functionality on the same machine (i.e., HI&MF) impacts on its QoS. Hence, we pose an attack between them, and we also consider not having HI (NotHI) or MF (NotMF). Moreover, we have some countermeasures in conflict, i.e., EL or VPN, and MF.

We obtain three stable extensions (we use the stable semantics because it is the most sceptical one, see Sec. 2): *i*) {AP, VPN, EL, HI, NI, NotMF, QoS}, *ii*) {AP, VPN, EL, HI, NI, HI&MF}, and *iii*) {AP, VPN, EL, NI, NotHI, NotMF, QoS}. In this case, reasoning in terms of stable or preferred semantics is the same, since they both returns the same three extensions. Reasoning on the scep-

³ ISO, ISO, and I. E. C. Std. “ISO 27002: 2005.” Information Technology-Security Techniques-Code of Practice for Information Security Management. ISO (2005).

⁴ <http://www.dmi.unipg.it/secarg>

⁵ <http://www.gecode.org>

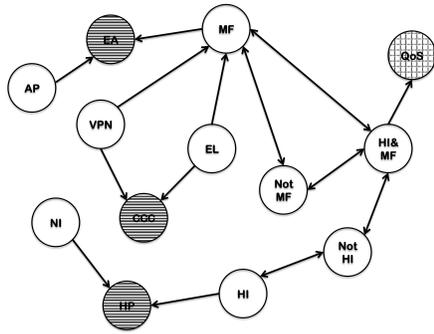


Fig. 2: The AAF with controls, threats (horizontal filling), and *QoS* asset.

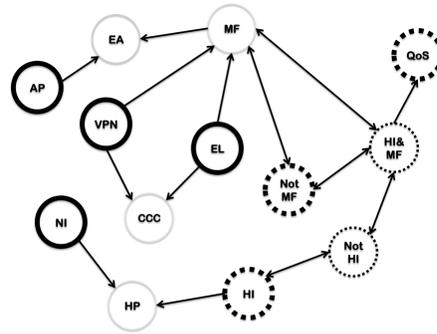


Fig. 3: Sceptically (thick), credulously (dotted) accepted, rejected (grey).

tical acceptance of arguments in such three extensions, we obtain that AP, VPN, EL, NI are sceptically accepted (i.e., “always”). This means that, for the attack/countermeasure scenario we have depicted, having audit procedures, a virtual private network, an encrypted line, and a network IDS is always considered a valid argument. Therefore, they correspond to a strong suggestion for the security administrator. On the other hand, there are some other arguments that are rejected (see Def. 3), that is they never appear in such extensions; for instance EA, HP, MF, and CCC. All three threats are successfully “avoided”, in the sense that adopted security countermeasures always prevent all of them. Moreover, also adopting the monitoring functionality countermeasure is not a good idea given this scenario, since it is rejected as well. Finally, the remaining arguments appear sometimes but not always in such three extensions (they are credulously accepted, according to Def. 3): NotHI (in 1 extension), HI&MF (1), HI (2), NotMF (2), QoS (2). The number of times they appear is visually highlighted in SecArg by filling arguments with different shades of grey, and also returning the appearance ratio, e.g., 66.6% for QoS and 33.3% for NotHI. This can be interpreted as a strength-score for these arguments: for instance, having an host IDS beats not having it (2 to 1): hence the administrator is recommended to use it. For the sake of presentation, in Fig. 3 we use thick continuous circles for sceptically accepted arguments, thin/thick dotted circles for credulously accepted ones (respectively for lower/higher ratio of appearance, e.g., QoS is thicker than NotHI), and light-grey circles for rejected arguments.

4 Related and Future Work

Since the application of Argumentation to Cybersecurity-related issues is relatively a new field (or, at least, not deeply investigated), there is a few related work to be mentioned. A bunch of works applying Argumentation-based conflict-resolution to the specific case of firewall rules are [1–3]. In our approach, however, we would like to provide a general reasoning-tool.

In [8] the authors suggest the use of Argumentation to provide automated support for Cybersecurity decisions. Three different tasks where Argumentation can contribute are surveyed in the paper: first, the establishment of a security policy, drawing from a range of information on best practice and taking into account likely attacks and the vulnerability of the system to those attacks. Secondly, the process diagnosis to determine if an attack is underway after some apparent anomaly in system operation is detected; the final goal is to decide what action, if any, should be taken to ensure system integrity. At last, Argumentation can be used to reconfigure a security policy in the aftermath of a successful attack: this reconfiguration needs to ensure protection against future similar-attacks, without creating new vulnerabilities.

In [7] the authors propose how arguments can support the decision making process: the aim is to help the system security administrator to react (or not) to possible ongoing attacks. For instance, a decision can be taken either to disable traffic through port 80 or not to disable it.

In the next future we would like to extend SecArg from both the theoretical and practical point of view by *i)* interactively changing the AAF with a new node or attack and immediately see how much such modification impacts on the strength of arguments; *ii)* selecting a subset S of arguments and get the minimal amount of change to the AAF that transforms S into an extension satisfying a given semantics (e.g., preferred).

References

1. Applebaum, A., Levitt, K.N., Rowe, J., Parsons, S.: Arguing about firewall policy. In: Verheij, B., Szeider, S., Woltran, S. (eds.) COMMA. Frontiers in Artificial Intelligence and Applications, vol. 245, pp. 91–102. IOS Press (2012)
2. Bandara, A.K., Kakas, A.C., Lupu, E.C., Russo, A.: Using argumentation logic for firewall policy specification and analysis. In: State, R., van der Meer, S., O’Sullivan, D., Pfeifer, T. (eds.) DSOM. LNCS, vol. 4269, pp. 185–196. Springer (2006)
3. Bandara, A.K., Kakas, A.C., Lupu, E.C., Russo, A.: Using argumentation logic for firewall configuration management. In: Integrated Network Management. pp. 180–187. IEEE (2009)
4. Bistarelli, S., Rossi, F., Santini, F.: Benchmarking hard problems in random abstract AFs: The stable semantics. In: Computational Models of Argument - Proceedings of COMMA. FAIA, vol. 266, pp. 153–160. IOS Press (2014)
5. Bistarelli, S., Rossi, F., Santini, F.: A first comparison of abstract argumentation reasoning-tools. In: ECAI 2014 - 21st European Conference on Artificial Intelligence. FAIA, vol. 263, pp. 969–970. IOS Press (2014)
6. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artif. Intell.* 77(2), 321–357 (1995)
7. Martinelli, F., Santini, F.: Debating cybersecurity or securing a debate? - (position paper). In: Foundations and Practice of Security - 7th International Symposium, FPS 2014. LNCS, vol. 8930, pp. 239–246. Springer (2014)
8. Rowe, J., Levitt, K., Parsons, S., Sklar, E., Applebaum, A., Jalal, S.: Argumentation logic to assist in security administration. In: Proceedings of the 2012 Workshop on New Security Paradigms. pp. 43–52. NSPW ’12, ACM (2012)