# Network Security Supported by Arguments

Fabio Martinelli, Francesco Santini[*], and Artsiom Yautsiukhin
IIT-CNR
Via Moruzzi 1, 56124 Pisa, Italy
Email: firstname.lastname@iit.cnr.it

***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 support the administration of security in complex systems, e.g., in case threat countermeasures are in conflict with non-functional requirements. The proposed formalisation is able to find the required security controls and explicitly provide arguments supporting this selection. Therefore, an explanation automatically comes as part of the suggested solution, facilitating human comprehension.**

## I. Introduction and Motivations

An *Abstract Argumentation Framework* (*AAF*), or System, as introduced in a seminal paper by Dung [1], 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" relations. 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. Suppose two arguments $a$ and $b$ are mutually in conflict, i.e., $a$ attacks $b$, and $b$ attacks $a$, then either $\{a\}$ or $\{b\}$ can be presented as possible acceptable extensions. Argumentation is an important subject of research in Artificial Intelligence and it is also of interest in several disciplines, such as Logic, Philosophy and Communication Theory (see [2, Ch. 1]).

Argumentation allows us to link different arguments on a topic, highlight counterarguments and reason about a balance of the opinions, thus reasoning in presence of conflict. In other words, this kind of reasoning under uncertainty ($\{a\}$ or $\{b\}$?) can be used in security areas where different opinions should be taken into account before a decision is made. Abstract Argumentation finds subsets of "collectively acceptable" arguments about the security of a system, and then show why a specific decision was taken and counterarguments rejected/accepted.

In this work we focus on application of Abstract Argumentation for decision-making during risk management of a system. We assume, that the owner of the system has already identified the main threats and would like to check whether the security countermeasures installed in the system are sufficient to maintain the risk level at minimum. Such analysis may be used separately, or as a part of the overall risk assessment process (see, for example [3], [4], [5], [6], [7]).

The main goal of any risk assessment is to assure that security goals are achieved (e.g., protect critical assets). Moreover, security should not also violate the core business goals for operation of the system. For example, productivity of the system should not be impacted by security measures significantly. Moreover, some security countermeasures may conflict with each other, reducing the overall security level. We see argumentation logic as a formal way to structure the control analysis and selection process and help the analysts to weight arguments supporting installation of new controls.

In this work we provide two version of the approach. The first one is a high level representation of the risks, security controls, goals and relations between them. Such a model is easier to build but it lacks of specific details and its quality and usefulness heavily depends on the analysis. The second version is based on the network topology of the organisation and explicitly indicates the weaknesses in the security system.

The mains contribution of this paper is an approach for analysis of security system with abstract argumentation framework. In particular, the approach: *i)* describes how to build abstract argumentation framework out of general description of the system or using network topology, *ii)* visualises the existing arguments supporting and attacking each other, *iii)* supports decision making by analysis of the framework and checking whether considered countermeasures are able to protect the system properly and satisfy the business goals.

The paper is structured as follows: in Sec. II we introduce the necessary background notions related to AAFs. In Sec. III we introduce our case example. Sec. IV and V propose a high level and topology-aware versions of our approach. Finally, Sec. VI compares the paper with other similar approaches and Sec. VII wraps up the paper and proposes some future work.

## II. Abstract Argumentation Frameworks

In this section we briefly summarise the background information related to classical Abstract Argumentation Frameworks (AAFs) [1].

*Definition 2.1 (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. $\forall a, b \in A$, $aR\,b$ (or, $a \rightarrowtail b$) 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., $S \rightarrowtail a$, if $a$ is attacked by an argument of $S$, i.e., $\exists b \in S.b \rightarrowtail a$.

*Definition 2.2 (Defence):* Given $F = \langle A, R\rangle$, an argument $a \in A$ is defended (in $F$) by a set $S \subseteq A$ if for each $b \in A$, such that $b \rightarrowtail a$, also $S \rightarrowtail b$ holds.

The "acceptability" of an argument can be defined under different semantics $\sigma$, depending on the frequency of its membership to some sets, called *extensions*: such semantics characterise a collective "acceptability" for arguments. In Def. 2.3

[*] Author supported by PRIN 2010XSEMLC "Security Horizons".

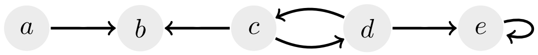

Fig. 1. An example of AAF.

we only report the original semantics given by Dung [1] (successive proposals can be found in the literature [2, Ch. 2.5]: $\sigma = \{adm, com, prf, stb, gde\}$, which stand for admissible, complete, preferred, stable, and grounded semantics.

*Definition 2.3 (Semantics [1]):* 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 $a \rightarrowtail b$ or $b \rightarrowtail a \in R$. For $S \in cf(F)$, it holds that *i)* $S \in adm(F)$, if each $a \in S$ is defended by $S$; *ii)* $S \in com(F)$, if $S \in adm(F)$ and for each $a \in A$ defended by $S$, $a \in S$ holds; *iii)* $S \in prf(F)$, if $S \in adm(F)$ and there is no $T \in adm(F)$ with $S \subset T$; *iv)* $S \in stb(F)$, if for each $a \in A \backslash S$, $S \rightarrowtail a$; *v)* $S = gde(F)$ if $S \in com(F)$ and there is no $T \in com(F)$ with $T \subset S$.

We also recall that the different requirements in Def. 2.3 define an inclusion hierarchy on the extensions: from the most to the least stringent we have $stb(F) \subseteq prf(F) \subseteq com(F) \subseteq adm(F)$. For such reason, this hierarchy also defines a degree of credulousness (conversely, "strength") for a considered subset of arguments, e.g., the stable semantics is the least credulous (strongest) among all, and this is why it will be extensively used in the rest of the paper. The grounded extension is the minimal fixed point (on complete extensions) of a framework: it minimises the arguments that are taken in.

Moreover, we can also define a strength level for each argument. A sceptically accepted argument proves to be stronger than a credulously accepted one.

*Definition 2.4 (Arguments acceptance-state):* Given one of the semantics $\sigma$ in Def. 2.3 and a framework $F$, an argument $a$ is *i)* sceptically accepted in iff $\forall S \in \sigma(F), a \in S$, *ii)* $a$ is credulously accepted if $\exists S \in \sigma(F), a \in S$ and $a$ is not sceptically accepted.

*Example 2.1:* Consider $F = \langle A, R \rangle$ in Fig. 1, with $A = \{a, b, c, d, e\}$ and $R = \{a \rightarrowtail b, c \rightarrowtail b, c \rightarrowtail d, d \rightarrowtail c, d \rightarrowtail e, e \rightarrowtail e\}$. In $F$ we have $adm(F) = \{\emptyset, \{a\}, \{c\}, \{d\}, \{a, c\}, \{a, d\}\}$, $com(F) = \{\{a\}, \{a, c\}, \{a, d\}\}$, $prf(F) = \{\{a, d\}, \{a, c\}\}$, $stb(F) = \{\{a, d\}\}$, and $gde(F) = \{a\}$. Hence, argument $a$ is sceptically accepted in $com(F)$, $prf(F)$ and $stb(F)$, while it is only credulously accepted in $adm(F)$.

In Def. 2.5 we introduce Preference-based Argumentation, which will be usedE.g. in Sec. IV-A to refine extensions.

*Definition 2.5 (Preference-based Argumentation [8]):* A preference-based argumentation framework is a triplet $\langle A, R, Pref \rangle$ where $Pref$ is a partial pre-ordering (reflexive and transitive binary relation) on $A \times A$. The notion of defence (see Def. 2.2) changes accordingly: let $a$ and $b$ be two arguments, we define $b \rightarrowtail a$ iff $R(b, a)$ and not $a > b$.

In order to find extensions, in the following of the paper we exploit *ConArg*[1] [9], [10], [11] (ARGumentation with CONstraints), which is a reasoning-tool based on Argumentation.

## III. Setting the Scene

Consider a small research and development company (SME). 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 cooperation with other enterprises is achieved through the Internet. Finally, the company needs to use its resources efficiently with the purpose to survive in a highly competitive market. In short, the company has the following security goals:

✓ protect the knowledge from external attackers (PKE);
✓ protect the knowledge from internal abuse (PKI);
✓ protect the communication channels (PC);
✓ efficiently use the available resources (Cost);
✓ ensure productivity of the operations (QoS).

The administrator of the SME has identified the following threats (items) and related protective security controls:

✠ hacker penetration (HP):
  ∇ host IDS (HI),
  ∇ network IDS (NI),
  ∇ penetration testing (PT),
  ∇ proxy firewall (PF);
✠ employee abuse (EA):
  ∇ monitoring functionality (MF),
  ∇ audit procedures (AP);
✠ compromise of communication channel (CCC):
  ∇ VPN (virtual private network) (VPN),
  ∇ encrypted line (EL);

The questions for the administrator are: *Is the current protection enough? What can improve the security system?* We provide the answers to these questions using our approach.

## IV. Modelling without topology

In this section we describe how to build an AAF. In this initial application, our aim is to reason on the goals, threats, and security controls (similar to the ones given in Sec. III).

We start with an empty framework $F = \langle A, R \rangle$, where $A = \emptyset$ and $R = \emptyset$. First we add the main goals of the evaluated organisation to the graph: let $A^g$ be a set of arguments stating that the goals are fulfilled. Then, we add possible threats $A^t$ (considered as arguments as well), and arguments for possible security controls, i.e., $A^{sc}$. Then, we add the attacks $R$ describing threats endangering the successful fulfilment of goals $R^{t-g} : A^t \times A^g$. The next step is to add attacks stating that specific security controls are effective against specific threats: $R^{sc-t} : A^{sc} \times A^t$. We should not forget about possible conflicts between security controls $R^{sc-sc} : A^{sc} \times A^{sc}$, and security controls and the goals $R^{sc-g} : A^{sc} \times A^g$. Finally, we find the stable extensions of $F$ and check that they do not contain arguments for threats $A^t \cap stb(F) = \emptyset$ (i.e., threats are eliminated) while main goals do belong to the set $A^t \subset stb(F)$.

Finally, the resulting abstract argumentation graph for analysis of the considered system is: $F = \langle A, R \rangle$, where $A = A^g \cup A^t \cup A^{sc}$ and $R = R^{t-g} \cup R^{sc-t} \cup R^{sc-sc} \cup R^{sc-g}$.

*Example 4.1:* By considering the scenario in Sec. III, first we collect

[1] http://www.dmi.unipg.it/conarg/

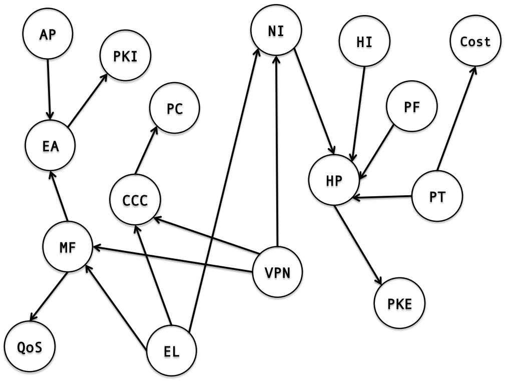

Fig. 2. HI = Host IDS, NI = Network IDS, PF = Proxy firewall, PT = Penetration test, CCC = Compromise of communication channel, VPN = Virtual private network, MF = Monitoring functionality, EL = Encrypted line, EA = Employee Abuse, HP = Hacker penetration, AP = Audit procedures, PC = protect communication channel, PKI = protect knowledge from internal abuse, PKE = protect knowledge from external abuse QoS = Good QoS, Cost = efficient use of resources.

$$\begin{aligned} A^g = &\ \{PKE, PKI, PC, QoS, Cost\},\\ A^t = &\ \{HP, EA, CCC\},\\ A^{sc} = &\ \{HI, NI, HO, VS, PT, PF, MF, AP, VPN, EL,\\ &\ LL, SCA, FA, SSA, SSP\}. \end{aligned}$$

Their union consists in our set of arguments $A$. Then we can focus on attacks: $R^{sc-t}$ can be simply derived from the list of items in Sec. III. We add an attack between each security control and the corresponding threat they mitigate: for instance $HI \rightarrowtail HP$, $MF \rightarrowtail EA$.

For what concerns $R^{t-g}$, hacker penetration may result in a confidentiality violation of protected data ($HP \rightarrowtail PKE$), employee abuse may lead to leakage of data ($EA \rightarrowtail PKI$), while a compromised channel may reveal the data in transmission ($CCC \rightarrowtail PC$). Note that some of these controls may conflict with each other, leading to $R^{sc-sc}$ attacks. For example, the power of monitoring procedure and a network-based IDS may be reduced with the use of encrypted line or VPN ($VPN \rightarrowtail MF$ and $EL \rightarrowtail MF$ and $VPN \rightarrowtail NI$ and $EL \rightarrowtail NI$). Moreover, security controls often affect some of the qualities of the system in a negative way: for instance, monitoring functionalities can affect the responsiveness of software ($MF \rightarrowtail QoS$), and penetration testing may be considered too costly ($PT \rightarrowtail Cost$).

The overall AAF is represented in Fig. 2. From such framework, by using ConArg (see Sec. II) we obtain one stable extension only, which is also the single preferred one, and grounded (which is always unique, see Sec. II): $\{AP, VPN, EL, HI, PF, QoS, PKI, PKE, PT, PC\}$. Hence, we obtain that, as desirable, all the threats ($HP$, $EA$, $CCC$) are prevented by countermeasures: they never appear in all the considered semantics. However, we realise that some security-controls may be dropped as well: monitoring functionality and network-based IDS go in conflict with the use of a VPN ($MF$ and $NI$ do not appear in the solution). Finally, not all the goals have been achieved, since having a penetration test impacts on the cost: argument $Cost$ does not belong to the solution, even if is not in conflict with any other security control.

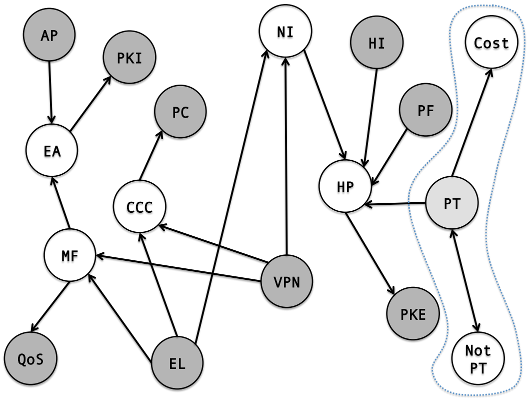

Fig. 3. Stable extension $\{AP, VPN, EL, HI, PF, QoS, PKI, PKE, PT, PC\}$.

Therefore, by gathering the information concerning our network in the form of arguments, we can use Argumentation semantics to visualise where noteworthy issues are. The Security Administrator can be suggested to find an alternative to $PT$ with the purpose to improve the use of available resources, so to bring argument $Cost$ back in. Not having $Cost$ is due to having $PT$ as always (sceptically) accepted. For this reason, the Security Administrator can be suggested to add a $NotPT$ argument (i.e., not having a penetration test countermeasure in the network) that attacks $PT$, and viceversa: these two attacks are clearly mutually excludable. By extracting again all stable extensions we now find two equally strong solutions

- $\{AP, VPN, EL, HI, PF, QoS, PKI, PKE, \mathbf{PT}, PC\}$,
- $\{AP, VPN, EL, HI, PF, QoS, PKI, PKE, \mathbf{NotPT}, \mathbf{Cost}, PC\}$,

which are respectively depicted in Fig. 3 and Fig. 4. The two extensions differ on arguments in bold, which are also graphically grouped in both Fig. 3 and Fig. 4. Consequently, sceptically accepted arguments are $AP$, $VPN$, $EL$, $HI$, $PF$, $PC$, $QoS$, $PKI$, $PKE$, while credulously accepted arguments are $PT$, $NotPT$, and $Cost$.

The minimal core of strong arguments is represented by the (single) grounded extension (see Sec. 2.1), which is $AP$, $VPN$, $EL$, $HI$, $PF$, $QoS$, $PKI$, $PKE$, and $PC$.

### A. A Refinement of Extensions

Since all the extensions satisfying a given semantics are equally strong, all the returned results (i.e., Fig. 3 and Fig. 4) are valid "truth versions". For this reason, we can use Preference-based Argumentation (see Def. 2.5) to restrain the number of solutions. In order to accomplish this, we establish a preference relation between arguments as it follows: we

- prefer a goal over a security control. e.g., $Cost > PT$;
- prefer a goal over a threat, e.g., $Cost > HP$;
- prefer a security control over a threat, e.g., $Cost > HP$;
- prefer not having a security control, than having that security control, i.e., $NotPT > PT$. It is clearly better

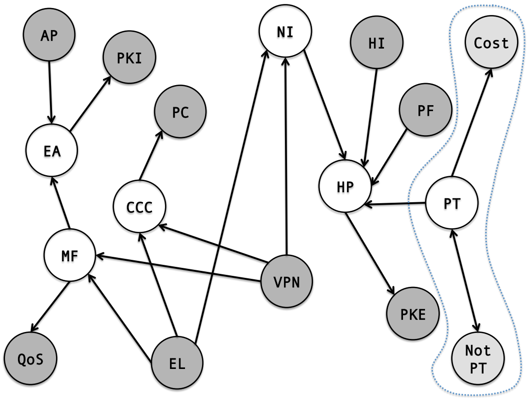

Fig. 4. Stable extension $\{AP, VPN, EL, HI, PF, QoS, PKI, PKE, NotPT, Cost, PC\}$.

not to introduce a security control whenever possible, in order to reduce the administration task.

By also coding such preferences we obtain only the solution in Fig. 4, thus including all the arguments representing a goal. The Security Administrator is then unequivocally supported in taking a final decision that does not include penetration-test tools (as Fig. 3 would have suggested instead).

## V. Modelling with Network Topology

The approach proposed in Sec. IV is simple to apply, but has some important limitations. First of all, relations between threats and goals are established only with the knowledge of an expert. In general, we are not able to say whether hacker penetration reaches the secrete information. Instead, an attacker is able to compromise computers which do not contain this data. Moreover, if we would like to separate different effects of hacker penetration (e.g., attack on one server or on another one), we can do this only due to the expert knowledge (an expert should mark threats and goals for different services differently). Different controls may be in conflict only when applied to the same element of the system. What is more important, is that the high level model does not guarantee that the applied control is effective against the considered threat (without a proper control of the expert, of course). For example, a host-based IDS may be installed on a wrong server and leave an important machine unprotected. Finally, many penetrations in the system require several steps to prevent such penetrations, there is no need to prevent all these steps (which may be costly), but to break the chain of steps at some point. Firewalls and Network-based IDS often work with this principle in mind. Naturally, an expert may predict the effect of such controls, but we would like to reduce the subjectivity of the analysis.

In order to improve our approach, we propose to consider the topology of a system, and specify attacks/goals/countermeasures for specific components of this topology.

For the sake of presentation, we focus on a simplified network topology-model. We consider three types of elements: devices, channels and networks. Channels in this work are virtual connections of computers that have to be considered separately, and which usually work over usual networks. For example, a channel between two partners over the Internet.

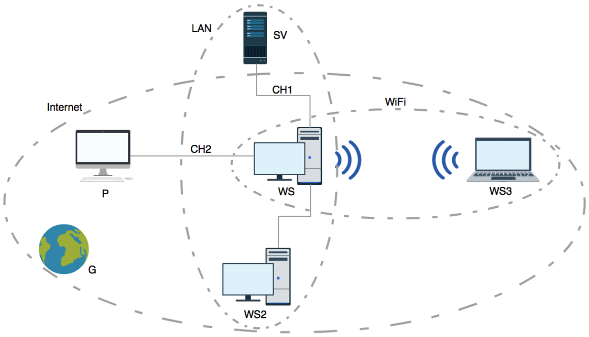

Fig. 5. Example of network topology.

Let $E$ be a set of elements $e$ with a predicate $type : e \mapsto TYPE$ defined over it. $TYPE$ is a set $\{d(evice), n(etwork), c(hannel)\}$. We formalise topology as a graph $TG = \langle E, AR \rangle$, where the elements of the considered topology are nodes and the arcs indicate direct connection between the elements, i.e., between a device and a network, a network and a channel, a network and a network, and a channel and a device.

*Example 5.1:* The enterprise has one server ($SV$) and three workstations ($WS$, $WS2$, $WS3$). Only workstation WS has access to the sensitive information from server $SV$. Also the partners $P$ have access to the WS through the Internet. The network of the enterprise consists of local access network ($LAN$), between $WS$, $WS2$, and $SV$, and a Wi-Fi network ($WiFi$), between $WS$ and $WS3$. Finally, $WS$ and $WS2$ have access to the Internet directly (not via LAN). There are also two channels which require special consideration: $CH1$, between $WS$ and $SV$ and $CH2$, between $WS$ and $P$.

All named elements are nodes in the graph $TG = \langle E, AR \rangle$, where $E = \{WS, W2, W3, SV, LAN, WiFi, CH1, CH2, INT, P\}$ and relations can be derived from the description above. Special considerations required only for networks. $WiFi$ is connected to the Internet, when $LAN$ is not. Therefore, there is an arc in the first case, and there is no arc in the second[2]. See Fig. 5 for the network topology and Fig. 6 for the corresponding $TG$.

In order to connect different elements of the topology as arguments/attacks of an AAF, we consider three steps: *i)* enter to an element trough a network/channel, *ii)* pass through the element, *iii)* exit from an element to a network/channel.

Every step is considered as two arguments: *protective* (positive) argument and *risky* (negative) argument, while the first is attacking the second one. Now, if we need to indicate the propagation from a network to an element we should:

- add two (positive and negative) arguments for the network;
- add two arguments for a connection between the network and the element;

[2] Someone may consider such topology not optimal, but we use it to show different cases for application of our methodology

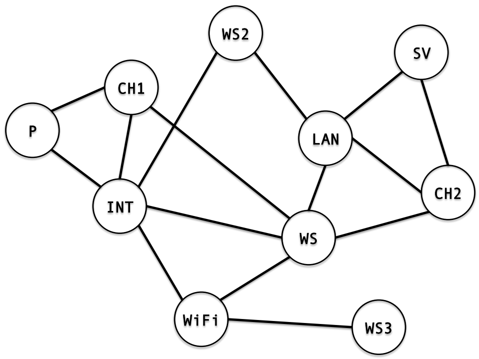

Fig. 6. Example of network topology model.

- connect the negative argument from the first couple to the positive one from the second couple;
- add two arguments for the element;
- connect the negative argument from the second couple and the positive argument from the third one.

See Fig. 7. As a result we get a chain of six arguments starting with a positive argument and ending with a negative one. Note, that such a trick may help us to join different elements of the same network, and even elements of connecting networks. For example, we may indicate how the same network may affect its elements (see Fig. 8), or how one element affects another one through the network.

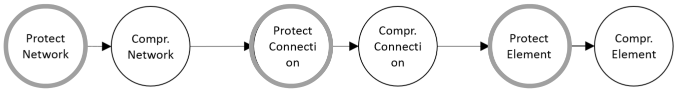

Fig. 7. From network to an element: protective arguments are represented by grey circles, negative arguments by black ones.

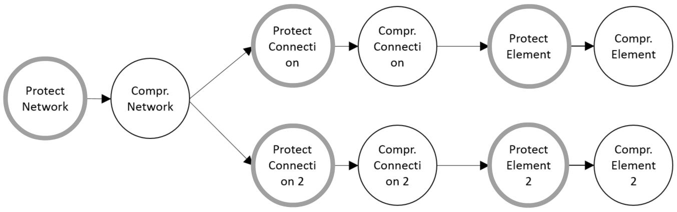

Fig. 8. From a network to elements: protective arguments are represented by grey circles, negative arguments by black ones.

Chains of arguments separate the effect of a threat and its cause. Every threat now should be marked with two arguments: 1) what is the result of the threat and 2) what is the initial element of the topology where the adversary starts an attack. For example, $HP(INT, SV)$ shows that a hacker penetration initiates in the internet and targets SV. Naturally, the result of an attack is the last argument of the chain (which then affects a business goal). This node is always protected by construction. In other words, our considered network is vulnerable but we do not consider this threat unless there is someone to exploit it. In order to add the threat to our model, we need to show where this threat comes from, i.e., at which part of the chain the attacker starts its malicious activity. In most cases, hacker may be considered attacking the protecting argument of the Internet (modelling the unsafe Internet case). Through the chain, the attacker is able to reach her goal.

For simplicity, we denote protection argument for an element $e$ as $P^q(e)$, while $C^q(e)$ states for compromising argument for the same element. Transition elements are $P^q(e_1-e_2)$ and $P^q(e_1-e_2)$. We would like to note, that there are different types of threats, and different types of protection respectively. For example, availability of a node may be considered as a different argument with respect to its integrity. Naturally, hacker penetration target integrity of a node, when vandals launch a DoS attack and affect availability of the element. In order to take this different nature of threats into account, we mark protecting and compromising arguments with type $q \in Q$, where $Q$ is a set of all types of penetration.

Let $e_1$ and $e_2$ be two elements of a network. Then we propose a notation $e_1 \overset{q}{\to} e_2$ to express a graph $F' = \langle A', R' \rangle$ formed with the six arguments, similar to Fig. 7, such that: $A' = \{P^q(e_1), C^q(e_1), P^q(e_1 - e_2), C^q(e_1 - e_2), P^q(e_2), C^q(e_2)\}$, $R' = \{P^q(e_1) \rightarrowtail C^q(e_1), C^q(e_1) \rightarrowtail P^q(e_1 - e_2), P^q(e_1 - e_2) \rightarrowtail C^q(e_1 - e_2), C^q(e_1 - e_2) \rightarrowtail P^q(e_2), P^q(e_2) \rightarrowtail C^q(e_2)\}$. We also use a similar notation to show that an element may attack an argument $ARG : e_1 \xrightarrow{a_1} ARG$ ($ARG \in A$) to express a graph $F'' = \langle A'', R'' \rangle$ formed with the three arguments, such that: $A'' = \{P^q(e_1), C^q(e_1), ARG\}$, $R'' = \{P^q(e_1) \rightarrowtail C^q(e_1), C^q(e_1) \rightarrowtail ARG\}$. Finally, we write $e_1 \overset{q}{\to} e_2 \Rightarrow F$ to indicate, that we add all arguments and transitions to $F$.

Let $TG = \langle E, AR \rangle$ be a topology graph. Let $TF = \langle N, ED \rangle$ be an abstract argumentation graph constructed from $TG$ in the following way: for every edge $\langle e, e' \rangle \in AR$, add $e \overset{\bar{q}}{\to} e' \Rightarrow TF$ and $e' \overset{\bar{q}}{\to} e \Rightarrow TF$, where $\bar{q}$ is the type of the attack to be instantiated later.

There is no need to indicate how security controls affect the external threats. Similar to threats, security controls also should be marked with the element to which the control is applied. For example, $HI(LAN)$ shows that the network-based IDS is installed for network LAN. The security controls are simply result in arguments attacking insecure states of the system (negative arguments). For example, a firewall does not stop a hacker from trying her penetration attempt, but protects the connection between the Internet and the internal network from such attempts. Similarly, a host-based IDS does not stop a hacker from penetrating to the network, but protects a specific node from being compromised as a result of this attempt.

### A. Methodology for analysis with topology

Our methodology for the security analysis with topology starts with $TG = \langle E, AR \rangle$ and corresponding $TF = \langle N, ED \rangle$ and can be seen as follows:

1) The first step is to add all security goals to as arguments $A^g$ to $F$ the graph $A^g \Rightarrow F$.
2) Then, we add the topology to the AAF: copy $TF$ to $F$.
3) We add the arguments for the threats to the model linking arguments for goals with argument for elements.
   a) Substitute all $\bar{q}$ in $F$ with $q = “int''$, i.e., seizing control threats

b) For every threat $a_t(e, e')$ of type $q$.

i) First, for all elements $e' \in E$, which compromised by a threat type $q$ impact goal $a_g \in A^g$, we add $e^q \xrightarrow{q} a_g \Rightarrow F$. $e^q$ is $e'$ if $q$ is of type “$int''$, and is a twin for $e'$ required for the analysis of threats of type $q$ only.

ii) Second, for all elements $e'' \in E$, which are connected to $e'$, we add $e'' \xrightarrow{q} e^q \Rightarrow F$.

iii) Third, we add the threat $a_t \in A^t$ itself and its connection to the topology element $e$: $a_t \xrightarrow{q} e \Rightarrow F$.

4) Add arguments $A^{sc}$ for security controls to $F$ graph and connect them to the graph according to the predefined description of the controls. Every control should specify which node it may protect and how it is connected to the topology. Similar to threats, security controls may define to which element of the system (or connection), they are applied, as well as the type of threat they protect from. For example, a network-based IDS can be seen as $NI(NW, int)$, where NW is the name of the network element and $q =$ “$int''$ is a type of threat which results in attacker getting the possibility to send and receive messages from a network or a node. A monitoring functionality, can be defined as $MF(WS-INT, abuse)$, indicating that the monitoring is performed for communication going from WS to the Internet, preventing possible employee abuse.
We propose to add also two more arguments between the security control argument and the attacked argument (see top four arguments in Fig. 9 for an example). The reason for doing that is that adding a control to the system could be impeded by some parts of the system. Thus, although the control is present in the system it cannot provide its functionality properly. The considered case of conflicts of security controls is an example of such situation (e.g., VPN applied to the same channel as a Network-based IDS impedes work of the latest, see Fig. 9).
Note, that every security control should be added according to its functionality and depending on the topology. For example, a Network-based IDS protects the network, when, in fact, it protects all elements of this network (including other sub-networks).

5) Add attacks of security controls on security goals $R^{sc-g}$ and on other security controls $R^{sc-sc}$.

6) Now we are able to analyse AAF and find all stable extensions. In this work we analyse separately every set of threats heaving the same origin of attack, i.e., the same attacker. Thus, at step 3b we add to same $F$ only threats with the same initial element $e$. Now, for every built $F$ we look whether the stable extension includes all business goals. Moreover, we may conduct a what-if analysis to optimise the network configuration.

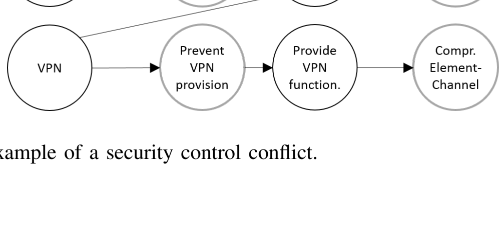

Fig. 9. Example of a security control conflict.

Note, that steps 4 and 5 require special consideration, depending on the type of the threat and security controls considered. These are rules which have to be specified by experts for every threat and can be simply applied in every specific analysis.

### B. An Example with Topology

The first step of our methodology is similar to the previous example and $A^g = \{PKE, PKI, PC, QoS, Cost\}$. On the other hand, we should specify threats with more details:

- hacker penetration (to SV): $HP(INT, SV)$;
- employee abuse of the user of WS (through networks Internet, WiFi): $EA(WS, INT), EA(WS, WiFi)$;
- compromise of communication channel (*CH1* and *CH2*): $CCC(INT, CH1), CCC(INT, CH1)$.

Note, that every threat consists of the name, source and target of the threat (e.g., $HP(INT, SV)$ shows that hacker penetration originates from the Internet and targets the server). Now, it is clear, that for Step 3b(i) of our methodology it is required to add $SV \xrightarrow{“int''} PKE$ to $F$, where $q =$ “$int''$ refers to getting control over $SV$. Step 3b(ii) for $q =$ “$int''$ does not change the graph. In order to realise step 3b(iii) we should add $HP \xrightarrow{“int''} INT$ to $F$. Naturally, here the hacker does not get full control over the whole Internet, but is able to send and modify messages through it. Similarly, other threats are specified to attack security goals. Note, that these parts are disjoint now. This means, that the goals are defended.

For the sake of simplicity we assume that HP and CCC are both of type “$int''$ and we simply re-use the graph built from the topology ($TF$) updated with the threat related arguments, as it is stated in steps 3b(i) and 3b(iii). The result is shown in Fig. 10.

Employee abuse does not require seizing power of elements, thus next to adding attacks for compromising the security goal (step 3b(i)) $PKI$ $INT^a \xrightarrow{“abuse''} PKI$ and $WiFi^a \xrightarrow{“abuse''} PKI$ and the threat itself (step 3b(iii)) $EA \xrightarrow{“abuse''} WS$, we also need to add attacks indicating the fact of abuse (step 3b(ii)) $WS \xrightarrow{“abuse''} INT^a$, $WS \xrightarrow{“abuse''} WiFi^a$ and also $P \xrightarrow{“abuse''} INT^a$, $WS2 \xrightarrow{“abuse''} INT^a$, $CH1 \xrightarrow{“abuse''} INT^a$, $P \xrightarrow{“abuse''} WiFi^a$, $WS2 \xrightarrow{“abuse''} WiFi^a$, $CH1 \xrightarrow{“abuse''} WiFi^a$. Although, we assumed that only $WS$ can abuse the data, the graph already shows that instead of sending secrete data directly to the untrusted channels, a dishonest employee may find a different path for the data leakage. Assume that this employee knows about possible monitoring his/her interactions through the INT or WiFi. He/she may compromise a computer of a colleague (e.g., *WS2*) and redirect the secrete data through it to an untrusted channel. Finally, we would like to note that arguments for $INT^a$ and $INT$ ($WiFi^a$ and $WiFi$) are not the same: although they are related to the same element of the network, but $INT$ ($WiFi$) can be used for further attacks, when $INT^a$ ($WiFi^a$) attack only the one goal $PKI$.

Now we are able to add the security controls installed in the system. These controls may be defined as follows:

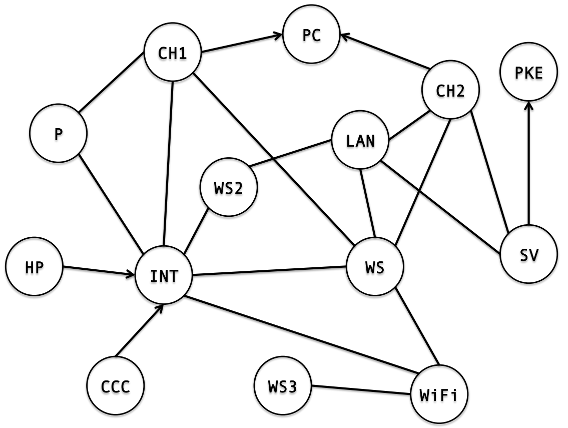

Fig. 10. Example of AFF for analysis of HP and CCC threats.

- Network-based IDS: $NI(LAN, int)$
- Host-based IDS: $HI(SV, int)$
- Proxy Firewall: $PF(INT, WiFi, int)$
- Penetration testing: $PF(LAN, int)$
- Monitoring functionality: $MF(WS, LAN, abuse)$, $MF(WS, CH1, abuse)$, $MF(WS, CH2, abuse)$, $MF(WS, Int, abuse)$, $MF(WS, WiFi, abuse)$
- VPN: $VPN(WS, CH1, int)$, $VPN(P, CH1, int)$, $VPN(NW, CH1, int)$, $VPN(WS, CH2, int)$, $VPN(SV, CH2, int)$, $VPN(LAN, CH2, int)$,
- Encrypted line: $EL(WS, CH1, int)$, $EL(P, CH1, int)$, $EL(NW, CH1, int)$, $EL(WS, CH2, int)$, $EL(SV, CH2, int)$, $EL(LAN, CH2, int)$,

Every security control applied not only to the specified elements of the system, but also some others, regarding to the rules specified for the control. For example, Network-based IDS protects not only the primary network, but also all networks and channels based on the primary network. Similarly, the penetration testing protects not only the network, but also all elements of this network from being seized by a hacker.

Finally, we should specify all negative effects of security countermeasures. These negative effects also rooted in the parameters of the security control. For example, monitoring functionality reduces productivity of WS node only, leaving other elements of the network unaffected. The negative effect of a VPN can be seen when we consider the channel encrypted: CH1, which is also protected by a network-based IDS (CH1 is affected since it is a part of LAN primarily affected by NI). The similar negative effect VPN has on MF functionality.

For the sake of simplicity we consider only HP and CCC cases. The stable extension related to this example includes two goals that are consequently protected ($PC$ and $PKE$), while $Cost$ is not in because it is in conflict with $PT$, as in Fig. 4. By introducing $NotPT$, as in Fig. 3, we can make $Cost$ appear again in a second stable extension. However, we also note that $PF$ is not included in the stable extension: we can deduce that having such counter measure is not useful for this kind of attack, given the topology of this network.

In addition, we can see what happens if we remove two security controls from the configuration, i.e., $HI$ and $PT$. We obtain the following stable extension $\{C\_INT, C\_INT\_WS2, C\_WS2\_INT, C\_INT\_WS, C\_WS\_INT, P\_LAN\_WS2, C\_WS2\_LAN, P\_LAN\_WS, C\_WS\_LAN, P\_LAN\_CH2, P\_CH2\_LAN, P\_CH2\_WS, P\_SV\_CH2, P\_CH2\_SV, P\_SV\_LAN, P\_LAN\_SV$, **C_WS2**, **C_WS**, $P\_SV, P\_CH2, \underline{PC}, \underline{PKE}, HP, CCC, NI, P\_NI\_LAN, VPN, P\_VPN\_WS\_CH2, P\_VPN\_LAN\_CH2, \underline{Cost}, P\_VPN\_SV\_CH2\}$. We now notice that $Cost$ is in (since no more in conflict with $PT$), thus all three goals are satisfied. Moreover, we see that this happens even if two machines $WS$and $WS2$ (in bold) are compromised. Therefore, the remaining security controls are still enough to satisfy all the goals, because the critical point is the server $SV$. Finally, note that although we have network-based IDS installed together with conflicting VPN, we still get all goals satisfied, since the conflict relates to CH2 only, where VPN provides adequate protection, leaving NI protect the LAN.

## VI. Related 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 [12], [13], [14]. In our approach, however, we provide a general reasoning-tool, not focused on firewall rules only, but applicable to network security in general.

In [15] the authors formalise the reasoning about access control using a planning theory formalised in Dung's abstract argumentation framework [1]; such planning is based on an adaptation of Dung's notion of defence. Their formal argumentation framework allows arguments about the backward derivation of plans from objectives and policy rules (abduction), as well as arguments about the forward derivation of goals from general objectives. Parties negotiate to find an agreement about which policy to apply, even though there may be more than one way to achieve a security objective.

A first general and introductory work on Argumentation and Cybersecurity is proposed in [16]. There 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.

The work in [17] introduces an approach for the enforcement of security requirements based on argumentative logic; the aim is to reason about activation or deactivation of different security mechanisms under certain functional and non-functional requirements. The framework is applied to an automotive on-board system. Differently from this work, in [17] the authors take advantage of Argument-based Logic Programming (see [2, Ch. 8]), and not Abstract Argumentation (see Sec. II).

In [18], some of the authors of this paper 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. The work in [18] represents a first step along the line presented here; however, it does not consider the topology of a network.

## VII. Conclusion

We have shown how abstract argumentation framework may be helpful in analysis of security of a system. There are different ways how an analyst may use the proposed framework. The most straightforward way is to analyse the existing system. Then, the analyst may start conducting what-if analysis, considering different alternatives (e.g., as it is shown in Example 4.1, when option of having and not having penetration testing is considered). We have shown, that although well-defined high level model can be good representation of reality and used for analysis, the topology-aware version of the approach is less dependent on the correct specification by an analyst and provide much more details, allowing wider range of analysis types.

There are a number of ways to improve the work and make the analysis more powerful. First, we did not exploited the full range of analysis techniques provided by the argumentation logic. We would like to explore deeper these techniques in order to widen the possible analysis. One possible direction is to help an analyst to select the minimal set of countermeasures which still satisfy business goals. We can also consider the requirements for compliance with a standard (e.g., 2700x [19]).

Our approach clearly depends on the input of analysts: e.g., goals, threats, possible countermeasures, and attacks need to be provided. However, note that some tools already automatically suggest fixes for known vulnerabilities.[3] On the other hand, the approach is structured in a way to automatise many of the most boring steps. For example, if a formalised network description is available the dependency between nodes can be specified by a tool. Moreover, this tool may include the database of threats, goals, possible countermeasures, and dependencies between them. For example, new countermeasures may be added to the model automatically, if properly formalised. Implementation of such tool is also our future goal.

In this work, we considered cost and productivity requirements in a simplistic way. In the future, we would like to consider these and other constraints in a quantitative way, e.g., consider cost of every security control separately and aggregating the cost of all controls at the end, keeping it below some threshold. Also the preferences of arguments can be seen in quantitative or qualitative way in order to compare the effects of arguments on the system and prioritize stable extensions. This approach should help the analysis to select the most appropriate configuration.

Another possible future direction of our work is taking into account dynamicity of the system. The proposed method so far is static but it can be improved for a fast re-analysis when a part of the system or strategic business goals change.

## References

[1] P. M. Dung, “On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games,” *Artif. Intell.*, vol. 77, no. 2, pp. 321–357, 1995.

[2] I. Rahwan and G. R. Simari, *Argumentation in Artificial Intelligence*, 1st ed. Springer Publishing Company, Incorporated, 2009.

[3] S. A. Butler, “Security attribute evaluation method: a cost-benefit approach,” in *Proceedings of the 24th International Conference on Software Engineering (ICSE’02)*. ACM Press, 2002, pp. 232–240.

[4] G. Stoneburner, A. Goguen, and A. Feringa, “Risk management guide for information technology systems,” National Institute of Standards and Technology, Tech. Rep. 800-30, 2001.

[5] C. Alberts, A. Dorofee, J. Stevens, and C. Woody, “Introduction to the octave approach,” Software Engineering Institute, Carnegie Mellon University, Tech. Rep., 2003.

[6] ITC, *MAGERIT Version 1.0  Risk analysis and management methodology for information systems  Procedures Handbook*, Information Technology Council (Consejo Superior de Informatica), 2000.

[7] B. Karabacak and I. Sogukpinar, “Isram: information security risk analysis method,” *Computers & Security*, vol. 24, no. 2, pp. 147–159, 2005.

[8] L. Amgoud and C. Cayrol, “On the acceptability of arguments in preference-based argumentation,” in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI’98. Morgan Kaufmann Publishers Inc., 1998, pp. 1–7.

[9] S. Bistarelli, F. Rossi, and F. Santini, “Enumerating extensions on random abstract-AFs with argtools, aspartix, conarg2, and dung-o-matic,” in *Computational Logic in Multi-Agent Systems - 15th International Workshop, CLIMA XV*, 2014, pp. 70–86.

[10] S. Bistarelli, F. Rossi, and F. Santini, “A first comparison of abstract argumentation reasoning-tools,” in *ECAI 2014 - 21st European Conference on Artificial Intelligence*, 2014, pp. 969–970.

[11] S. Bistarelli, F. Rossi, and F. Santini, “Benchmarking hard problems in random abstract AFs: The stable semantics,” in *Computational Models of Argument - Proceedings of COMMA*, 2014, pp. 153–160.

[12] A. Applebaum, K. N. Levitt, J. Rowe, and S. Parsons, “Arguing about firewall policy,” in *COMMA*, ser. Frontiers in Artificial Intelligence and Applications, B. Verheij, S. Szeider, and S. Woltran, Eds., vol. 245. IOS Press, 2012, pp. 91–102.

[13] A. K. Bandara, A. C. Kakas, E. C. Lupu, and A. Russo, “Using argumentation logic for firewall configuration management,” in *Integrated Network Management*. IEEE, 2009, pp. 180–187.

[14] A. K. Bandara, A. C. Kakas, E. C. Lupu, and A. Russo, “Using argumentation logic for firewall policy specification and analysis,” in *DSOM*, ser. LNCS, R. State, S. van der Meer, D. O’Sullivan, and T. Pfeifer, Eds., vol. 4269. Springer, 2006, pp. 185–196.

[15] G. Boella, J. Hulstijn, and L. W. N. van der Torre, “Argumentation for access control,” in *AI*IA*, 2005, pp. 86–97.

[16] J. Rowe, K. Levitt, S. Parsons, E. Sklar, A. Applebaum, and S. Jalal, “Argumentation logic to assist in security administration,” in *Proceedings of the 2012 Workshop on New Security Paradigms*, ser. NSPW ’12. ACM, 2012, pp. 43–52.

[17] T. Bouyahia, M. S. Idrees, N. Cuppens-Boulahia, F. Cuppens, and F. Autrel, “Metric for security activities assisted by argumentative logic,” in *ESORICS workshops DPM/SETOP/QASA 2014, Revised Selected Papers*, ser. LNCS, vol. 8872. Springer, 2015, pp. 183–197.

[18] F. Martinelli and F. Santini, “Debating cybersecurity or securing a debate? - (position paper),” in *Foundations and Practice of Security - 7th International Symposium, FPS 2014*, ser. LNCS, vol. 8930. Springer, 2014, pp. 239–246.

[19] ISO/IEC, *ISO/IEC 27001:2013 Information technology  Security techniques  Information security management systems  Requirements*, 2013.

[3] For instance, *Nessus*: http://tinyurl.com/q6buj5d.