Nonfunctional Requirements Validation-A Game Theoretic Approach

Vicky Papadopoulou and Andreas Gregoriades
Department of Computer Science and Engineering, European University Cyprus, Cyprus
Email: {v.papadopoulou, a.gregoriades}@euc.ac.cy

***Abstract*—Network Security requirements have recently gained widespread attention in the requirements engineering community. Despite this, it is not yet clear how to systematically validate these requirements given the complexity and uncertainty characterizing modern networks. Traditionally, network security requirements specification has been the results of a reactive process. This however, limited the immunity property of the software systems that depended on these networks. Security requirements specification prerequisite a proactive approach. Networks' infrastructure is constantly under attack by hackers and malicious software that aim to break into computers. To combat these threats, network designers need sophisticated security validation techniques that will guarantee the minimum level of security for their future networks. To that end, this paper presents a game-theoretic approach to security requirements validation. An introduction to game theory is presented along with a case study that demonstrates the application of the approach in a hypothetical network topology.**

I. Introduction

A computer network is defined as the purposeful interconnection of computer nodes for the efficient and effective interchange of information. *Network security* consists of the provisions made in an underlying computer network to protect the network and the network-accessible resources from unauthorized access.

Network security is of paramount importance to modern information systems and their applications in modern business environments. The recent growth of public networks such as the Internet made this requirement even more significant. However, the dynamic characteristics of contemporary networks combined with their increased size makes the vision of absolute network security almost impossible. Specifically, networks are vulnerable to infection by different types of electronic *attacks*. Typically, an *attack* by a virus, Trojan horse or eavesdropper exploits the loopholes in the security mechanisms of the network. Guaranteeing an acceptable level of security for a prospective system represents a common problem in systems engineering. More specifically network security, is defined as a non-functional property that is influenced by functional aspects of the system as a whole. This area of research has gained considerable popularity due to the implications it has on users satisfaction and business reputation. Therefore, being able to quantify the safety performance of a future network early in the design phase is of vital importance. The need to validate security requirements early has been addressed also by Lamsweerde [1] and Crook [3].

In this work, we apply game theory to validate the security NFR of a prospective network prior to its implementation. The assessed security NFR represents the minimum level of security guarantee for a prospective network, given a number of immunity requirements to be implemented in the network. These correspond to antivirus software and their location on the network. Specifically, in the problem scenario we address in this paper we assume that a number of harmful entities or *attackers* (or an upper bound of this number) may appear anywhere in the network. Attacks target nodes of the network. When, there is no information on the distribution of the placement of the attacks on the network nodes, one may assume that they follow a *uniform distribution*. The immunity functional requirements of the network describe its defence mechanisms and are expressed by a set of *defenders*; software security systems that can guarantee an acceptable level of security to a limited part of the network (a link, a path, or a subnetwork). Attackers damage targeted nodes unless these are guarded by a defence software. Lamsweerde in [3] also refers to the need to analyze the rational of the attacker in an attempt to become proactive rather than reactive in network security management. Lamsweerde refers to anti goals and anti requirements that define the attacker's strategies based on which the network designers specified functional requirements to tackle these.

Unlike functional requirements, which can be deterministically validated, NFRs are soft variables that cannot be implemented directly; instead, they are satisfied [40] by a combination of functional requirements. NFRs define the overall qualities or attributes of the resulting system and as such place restrictions on the software product being developed. Examples of NFR include safety, security, usability, reliability and performance requirements.

Typical approaches to validating NFRs include, formal methods, prototypes and system simulations [2].

The paper is organised as follows. Firstly, we describe the principles behind game theory. Next we describe our approach through a case study, before we conclude with a brief discussion.

II. Game Theory

Game theory attempts to mathematically model the *rational* behavior in strategic situations, in which an individual's success in making choices depends on the choices of others. Game Theory has been used to understand selfish rational behaviour

of complex networks, e.g. the Internet, of many "agents" (consisting the *players* of the game). In such domains, Game Theory models players with potentially different goals (*utility functions*or *payoffs*), that participate under a common setting with well prescribed interactions (*strategies*), e.g. TCP/IP protocols. More importantly, it helps finding the *best strategy* of each player that will guarantee the best result. The core concept of Game Theory is the notion of *equilibrium* that is defined as the condition of a system in which competing influences are balanced.

A *Nash equilibrium* [5] in a game refers to happy instances of it where each no player can benefit by changing his or her strategy while the other players keep theirs unchanged. Nash equilibria model well stables states of a network, since if the network reaches such a configuration, most probably it would remain in the same configuration, since none of the involving entities has a motivation to change his status in order to be more satisfied. Thus, identifying Nash equilibria configuration of a network and evaluating them has been the main approach in order to analyze, evaluate networks performance.

III. The Method

Towards the goal of assessing network security NFR, an increasingly popular approach is to express this problem the form of a *game* between attacker and defense entities [**?**]. When the designer starts thinking like an attacker, in essence he/she engages in a game with the attacker. Finding and evaluating equilibriums between the attackers and defenders' strategies provide a measurement of the network's security. Most importantly, this critical information can be provided during the design phase of a prospective network and hence, enabling the designer to opt the network features accordingly. Therefore, finding and evaluating Nash equilibria of prospective networks can be used to validate networks security NFR and this is the approach we describe in here.

However, validating security requirements early in the design phase, prerequisite that we capture the network's behaviour in *all* possible types of assaults. These combinations constitute a number of possible test *scenarios*. Therefore, to evaluate the security performance of a prospective network we need to assess it against each scenario. Scenarios became a popular method for validating NFR [2] where each corresponds to a set of situations that might occur during the operation of a system. Application of scenarios in requirements validation has been performed by a number of researchers. However, the main problem in requirements validation through scenarios is the specification of an adequate set of test cases. In particular, too many scenario variations is needed to validate NFRs, hence, this drowns the requirements engineers in excessive detail. On the other hand, automated support for the scenario generation proved to be a vexed problem due to the exponentially large set of possible variations that needs to be examined [2] for the NFR to be guaranteed.

An approach that makes this problem tractable is described in here and is based on the application of game-theoretic analysis. In particular, we manage to significantly reduce the number of scenarios needed to validate the NFRs by investigating only stable network states (configurations). Actually, our method is of polynomial time compared to the size of the proposed network. Stable configurations describe the most likely states that a network could be in. Thus, by validating security NFR in such states, we ensure the validity of the NFR almost always. Such states are very well captured through Nash equilibria profiles of the resulting game. Thus, we only utilize Nash equilibria in order to assess network security.

Our approach is composed of the following steps:

1) Initially the network designer specifies quantitatively the required level of security for the future network
2) Next we model the non-functional security requirement to be validated in the prospective network as a game played on graph. In particular, we adopt the security game introduced in [4]. According to this approach, the security threats and the potential defence mechanisms are realized as confronting players on a graphical game. Moreover, we assume that the prospective network satisfies some common topological properties. Furthermore, we make some typical assumptions on the attacks that may appear in the network. Trying to be as general as possible, we assume that we have no information on the distribution of the attacks is provided. That is, we have no prior knowledge on whether some parts of the network will encounter attacks more often than others. Thus, we assume that attacks on the network nodes follows a uniform distribution. Finally, as part of our game theoretic representation of the problem we need to define the organisation of defence mechanisms o the network. This constitutes a functional immunity requirement of the proposed network.
3) We utilize the Nash equilibria identified and evaluated in [4] in order to measure the security guarantee in the prospective network. Since Nash equilibria model well the stable configurations of the network validating NFR on them, we ensure the validity of the NFR in the most probable states of the network. Evaluating of the Nash equilibria of the resulting game [4] serves here a validation method of the security NFR of the prospective network.

IV. Case-study

The security NFR of a prospective network is initially defined as a percentage of the required level of security. Finding equilibria through algorithmic Game Theory enables the designer to identify "stable" network configurations that archive the required level of security. Our approach validates security requirements by initially specifying the required level of security quantitatively and subsequently tests through algorithmic game analysis whether the expected level of security will be achieved.

Our approach is based on the notion of *scenarios* or use cases [2]. Where, scenarios correspond to possible configurations of attackers and defenders on the network. The use of Game Theory enables us to reduce the complexity of this

process by analysing only scenarios that both attackers and the defender would choose given that they act *rationally*- they act in a way that aims to maximizes their benefit. In particular, given a required security level, game-theoretic analysis specifies strategies of both attackers and the defenders that maximize their individual benefits. Through this analysis we achieve an assessment of the networks security.

Next we illustrate the application of the method for a network that is characterized by a set of functional requirements.

A. Network and Security Requirement Specifications

1) Network Specification: The prospective network N consists of an arbitrary number of nodes, n and a set of communication links E between the nodes of the network. Moreover, the following topological property is satisfied by N: there exists a subset of the links $E' \subseteq E$ such that each node v of the network is "*hit*" (incident) to *exactly* one link of the set E'. Note that a network with this property can be build and identified (that has fulfills the property) in polynomial time (such a set is called a *Perfect Matching* of the network). We call such a network a *hit-all* network.

2) Security Specifications: We identify network security specification according to a common process utilized in critical systems specifications. The process consists of the following stages:

(a) *Asset identification*: The assets of the network are the nodes of the network. In the most general case, all nodes are of the same importance. A node is considered protected or *secure* if a security software is installed on that node. Otherwise it is considered vulnerable to attacks.

(b) *Threat analysis and assignment*: The prospective network may witnessed threats, such as viruses, Trojan horses and eavesdroppers which are described as *attacks* that target the nodes of the network. At any time there is a maximum number of *attackers*, α, that may be present in the network. Each of them damages nodes that are not protected. In the most general case, we have no information on the distribution of the attacks on the nodes of the network. So, we assume that attacks will follow a uniform distribution, which is quite common in such cases. We call such attacks *uniform attacks*.

(c) *Technology analysis*: One major security mechanism for protecting network attacks are the *firewalls*, that we refer to as *defenders*. Furthermore, in distributed firewalls the network that is protected includes the links spanned by the nodes that participate in the distribution of the defenders. The simplest case occurs when the subnetwork is just a single link with its two nodes. However, due to financial costs (e.g., the prohibitive cost of purchasing a global security software) or from performance bottlenecks (e.g., the reduced usability of the protected part of the network) distributed mechanisms are only able to clean a limited part of the network.
So, we assume (covering the most basic and simplest case) that the prospective network is supported by a *single* security software, denoted as d, which is able to clean a link between two nodes from possible attackers at the endpoints of that link.
The distribution of defenders on the networks nodes exploits the topological property of the network as presented in the specification. That is, there is a set of links E' in the network such that any node is hit by (exactly) one link of that set. In particular, we assume defense mechanism chooses one link among that set E' with the same probability, that is uniformly at random. We call this placement of the defense mechanism as *uniform-hit-all*.

B. Security Requirement Validation

Here, we present our method to measure the prospective network's security level and thus, validating the security requirement prior to implementation. We first introduce some necessary notions and the security measurement used. Then, we proceed with a theoretical modeling of the proposed network using Graph Theory.

1) Network Configurations: A *network configuration* or just a configuration s is specified by (*i*) the locations (i.e. nodes) of the attackers and (*ii*) the location of the defense mechanism (i.e. a link). The attackers and defenders may follow a probability distribution on a subset of the nodes or links respectively. That is, each attacker is targets more than one node according to some probability distribution and the defense mechanism protects more than one link according to another probability distribution. In such a case, we say that s is a *mixed* configuration. Otherwise, the configuration is said to be *pure*; one attacker on one node and the defender on one link.

Figure 1 illustrates a mixed configuration for sample network N of 8 nodes ($n = 8$). It can be seen that the network illustrated is a hit-all type. We assume that there exists 3 different attackers ($\alpha = 3$). According to our assumptions on the Network security specifications, the attacks are uniform; and hence, the probability of an attacker assaulting any node of the network is equal to $1/n$ which is equal to $\frac{1}{8}$. In Figure 1, each attacker is indicated by X.

Again, according to our assumptions on the network security specifications, the security software mechanism chooses one link among a subset E' of the links uniformly at random. It can be easily verified that the distribution of the placement of the security mechanism is uniform-hit-all. So, each links of this set is chosen with probability $\frac{1}{|E'|} = \frac{1}{4}$. The links, as well as their corresponding visiting probabilities, are indicated by Y.

2) Security Measurement: In order to evaluate network security of the prospective network we adopt the method described in [4], by measuring the security level obtained, in an arbitrary profile (i.e., a configuration) of the defined game.

Consider a pure network configuration s. Let s_d the edge selected to be protected by the security software and for each attacher $i \in [\alpha]$, let s_i the node in which the node is attacked. We say that the attacker i is *killed* by the security

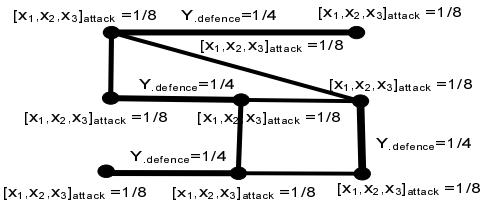

Fig. 1. A network configuration. Each attacker targets any node of the network with probability $\frac{1}{8}$. The security software chooses among a subset of links E' to clean them from possible attacks, uniformly at random. So, each link in the set is visited by the security software with probability $\frac{1}{4}$. The assessed security level of this scenario is equal to 25%.

mechanism if the node s_i is hit by the edge s_d selected by the security software. Then, the *defense ratio* [4] or *level of security* here of the configuration, r_s is defined to be: $r_s = \frac{\text{expected number of attackers killed in } s}{\alpha} \times 100$.

Based on the above, the optimal defense ratio of any network equals to 100 and it is achieved when the security software manages to kill all attackers. Then, in this case we specify that the network configuration obtains 100% security guarantee. The larger the value of r_s the greater the security level obtained.

In order to measure the security of the perspective network from all (exponential) configurations we consider only *stable* configurations. A network whenever reached such a configuration *tent* to remain in the same configuration. This is due to the fact that in such configurations the effects of the involving interacting entities even out. So, such configurations constitute the most probable states of the network. We identify such stable configurations to evaluate the network security on them. Thus, this measurement evaluates the security level of the prospective network.

3) A Game-Theoretic Modeling: We model both network and security specifications presented, using the graph-theoretic game of [4]. Specifically, we model the specification (1), (2.a) and (2.b) as a non-cooperative strategic game. The game is played on a graph G representing the network N. The players of the game are of two kinds: the *attackers* players and the *defender* players.

More importantly, we capture the stable configurations resulting from those specifications by the Nash equilibria found in the game of [4]. Thus, in order to evaluate network security we only need to evaluate the Nash equilibria of the game of [4]. Indeed they showed a result which is interpreted in our terms as follows:

Theorem 4.1 ([4]): Consider a network N with n nodes such that the network and security and requirement specifications given by (1), (2.a) and (2.b) are satisfied. Then the network contains a stable configuration (i.e. a Nash equilibrium) s with level of security expressed as: $r_s = \frac{2}{n} \times 100$.

The result implies that the network of Figure 1 has security level equal to $\frac{2}{n} \times 100 = \frac{2}{8} \times 100 = 25\%$, since $n = 8$. This designates that the level of security is 25% given the functional requirements specified in configuration s. This assessment however indicates that the initial NFR specified by the designer is not satisfied using the prescribed functional requirements of the network as is. Hence, the network specification needs to be revised.

V. Conclusion

Security requirements validation is typically performed through security-specific testing. This process is performed in addition to the traditional types of system testing. In this approach, test cases are usually based on abnormal scenarios that describe situations that the network will be called to face. This is analogous to test cases developed for use case based functional testing. These techniques however are mostly based on reactive a paradigm rather than proactive. Moreover, for these to be effective, it is required that a model of the prospective network is developed in a simulator based on which security can be validated. Most importantly, concentrating only on abnormal scenarios limits the effectiveness of our security validation process. Ideally, validation should be performed on all possible scenarios. However, examining all possible scenarios [2] in order to validating security requirements constitutes a highly complex (thus, inefficient) and sometimes infeasible task. In this work we manage to accomplish this process in only polynomial time. This is achieved by considering only stable configurations of the system, that we model using Nash equilibria. In this context, the method presented in this paper constitutes a novelty in validating security NFR through algorithmic game theory.

The approach presented in this paper constitutes a novelty in security requirements validation since it formally mimics the rationale of the network security problem in a game of attackers and defenders. The application of algorithmic Game Theory enables the identification of equilibria among the network's defenses and the attackers strategies and as a result enables the validation of a prospective networks security NFR using only a limited set of test scenarios. The method usage has been elaborated in a case study that explicitly demonstrates the core steps of the process for validating security requirements. The initial results of this work are encouraging and we are currently looking at techniques to automate the equilibria identification process through the application of systems thinking and system dynamics simulation.

References

[1] R. Crook, D. Ince, L. Lin and B. Nuseibeh, "Security requirements Engineering: When Anti-Requirements Hit the Fan, in *Proceedings of the 10th Anniversary IEEE Joint International Conference on Requirements Engineering*, pp. 203–205, 2002, IEEE Press.

[2] A. Gregoriades and A. Sutcliffe, "Scenario-Based Assessment of Non-Functional Requirements," *IEEE Transactions on Software Engineering*, Vol. 31, no. 5, pp. 392-409, 2005.

[3] A. van Lamsweerde, "Elaborating Security Requirements by Construction of Intentional Anti-Models", in *Proceedings of the 26th International Conference on Software Engineering*, pp. 148–157, 2004, IEEE Press.

[4] M. Mavronicolas, V. G. Papadopoulou, A. Philippou and P. G. Spirakis, "A Network Game with Attacker and Protector Entities," *Algorithmica*, Vol. 51, No. 3, pp. 315-341, July 2008.

[5] J. F. Nash, "Non-cooperative Games", *Annals of Mathematics*, 54(2):286–295, 1951.