

A Game Theoretic approach to Vulnerability Patching

Gabriele Gianini, Marco Cremonini, Andrea Rainini, Guido Lena Cota, Leopold Ghemmogne Fossi

Dipartimento di Informatica
Universita degli Studi di Milano
via Bramante 65, Crema (CR) 26013 - Italy

Abstract—Patching vulnerabilities is one of the key activities in security management. For most commercial systems however the number of relevant vulnerabilities is very high; as a consequence only a subset of them can be actually fixed: due to bounded resources, choosing them according to some optimal criterium is a critical challenge for the security manager. One has also to take into account, though, that even delivering attacks on vulnerabilities requires a non-negligible effort: also a potential attacker will always be constrained by bounded resources. Choosing which vulnerabilities to attack according to some optimality criterium is also a difficult challenge for a hacker. Here we argue that if both types of players are rational, wishing to maximize their ROI and aware of the two sides of the problem, their respective strategies can be discussed more naturally within a Game Theory (GT) framework. We develop the fact that the above described attack/defense scenario can be mapped onto a variant of GT models known as Search Games: we call this variant Enhanced Vulnerability Patching game. Under the hypothesis of rationality of the players, GT provides a prediction for their behavior in terms of a probability distribution over the possible choices: this result can help in supporting a semi-automatic choice of patch management with constrained resources. In this work we model and solve few prototypical instances of this class of games and outline the path towards more realistic and accurate GT models.

I. INTRODUCTION

Since decades, patching software vulnerabilities has been one of the key activities of an Information Security Management System and a task that every IT manager had to organise, schedule and include in management processes. The number of vulnerabilities affecting IT assets and the frequency of patch release from vendors are overwhelming and cannot be managed manually: the number of vulnerabilities classified in the CVE¹ has exceeded the threshold of 5000 per year in the last decade: of them, the High Risk go from 33% to 50%. This scenario has been recognised by many IT professionals as unmanageable. The reason is that patching a vulnerability requires a well structured procedure, since the installation of a patch could easily have undesirable side-effects (due to incompatibility with other libraries or due to bugs of the patch).

In many cases, for economic reasons, even the guideline prescription of patching all the High Risk CVE vulnerabilities is impossible to match. In short: an optimal management of the patches has to cope with the strong resource constraints and only a limited number of patches can really be actually deployed: choosing which vulnerabilities to address becomes

therefore a critical challenge.

The adequateness of an approach based only on the CVSS score² was challenged by a recent empirical investigation [2], showing that in order to achieve a high risk reduction the cost of patching should become huge. The reason is that typically only few hundreds vulnerabilities are targeted in a certain time-frame by actual malwares, therefore much of the patching efforts are useless. The authors suggest then that an effective patching method should first select the subset of vulnerabilities that are likely to be exploited by malware creators and then consider the CVSS rank order with respect to only to that subset. Still, this approach situates the problem within the domain of Decision Theory (DT).

In the present work not only 1) we take into account the observations in [2] and 2) argue that another major element to be modeled to account for limitations in the attacks spectrum is represented by the finite resources available to the hackers (buying and massively deploying *malware* is costly); moreover 3) we consider that in order to optimize their return on investment (ROI) hackers are likely to take their decisions *strategically*, by keeping into account the reasoning of the information security managers (for instance: patching mostly the "higher priority" vulnerabilities makes for the hackers more profitable attacking the "lower priority" ones). This leads us to the formulation of a more realistic model and the corresponding problem: optimally selecting a subset of vulnerabilities under the condition of selfishness and capability of strategic thought by both the hackers and the security manager. We call this problem the *Enhanced Patching Problem*.

The above elements lead quite naturally to represent the problem in terms of a Game Theory (GT) model. GT enables the modelling of interdependent decision landscapes consisting of two or more rational/selfish players with non-aligned interests (the degenerate case of a single decision taker against a non-strategic player corresponds to DT). By means of GT, one is able to predict the rational players' behaviour under specified circumstances and to calculate the solution of the game in terms of the so call game *equilibrium*: a collection of strategies (one for each player), from which no individual player has incentives to deviate unilaterally, since this would not increase its personal payoff.

The attack/defense scenario of the Enhanced Patching Problem can be mapped into a variant of a specific class of GT models, known as Search Games. In those games there are two types of players: the Searcher and the Hider; the Searcher tries

¹The Common Vulnerability Exposure (CVE) [6] is the official public reference for naming and classifying security vulnerabilities; it is paired with the National Vulnerability Database (NVD) a U.S. government repository. [10].

²The Common Vulnerability Scoring System (CVSS)[9] is the standard ranking algorithm adopted by NVD for classifying security vulnerabilities.

to *meet* the Hider, while the Hider tries to *avoid* the Searcher. The Hider receives a positive payoff when locating herself at a place which is not visited by the Searcher, while the Searcher receives a positive payoff when visiting the same place where the Hider is hidden. In the context of vulnerability patching, where the players are the Hacker and the Security Manager, the security manager role corresponds to the one of the Searcher (he benefits from patching the same vulnerability to be attacked by the Hacker) while the hacker's role corresponds to the one of the Hider (she benefits from attacking a vulnerability different from the one patched by the security manager).

Independently from IT applications the GT results related to Search Games have been summarized by Alpern and Gal [3] (only zero-sum games) and by Garnaev [7]. Quite a few works have tried to tackle some IT security problems with game theoretic approaches: Manshaei et al. [8] and Xiannuan et al. [11] provide comprehensive surveys of this area of research. Application of GT to vulnerability patching has been considered within the area of the so-called Interdependent Security Games (see [5] for a survey): those are games which focus on the interdependence of strategic defenders against either a malicious (thus non-rational) attacker. The presence of a single strategic attacker has been modeled by Chan et al. [4]. Also the work [1] posits the presence of a single attacker and – using as a model a nonzero-sum non-cooperative two-player Bayesian game – proposes a anti-malware that can shift between different security levels according to the assets value and the battery status of the resource-constrained device.

In the present work we do not consider interdependent defenders, but model (beyond the basic two-player case) the case of vulnerability patching with many attackers against a single defender (ideally the security manager of a system), and extend the analysis to a sequential move scenario.

Based on this background, we model the Enhanced Patching Problem in terms of an instance of the class of search games: we call it Vulnerability Patching Game. In this short paper, first we set stage of the game, then model and solve few prototypical variants (Section II) and finally outline the path towards obtaining more realistic GT models with increasing degrees of accuracy (Section III).

II. VULNERABILITY PATCHING MODELS

Game Theory (GT) is a branch of applied mathematics that models multi-person decision-making situations in order to account for interactions among strategies of rational decision makers. It is principally aimed at determining the preferred combination of strategies that will be adopted by rational agents trying to maximize their payoffs. A game is defined by a set of players, and, for each player, a set of possible strategies and a player's utility function mapping any possible state of affairs in the game into a payoff for the player.

For the purposes of the present work it is worth distinguishing between simultaneous single-round games, which are more efficiently represented in the so-called normal form, from the sequential games, which are more efficiently represented in the so-called extensive (or tree) form. First we model the Vulnerability Patching Game in its simplest form, a two-player simultaneous game in normal form over two vulnerabilities. Then we outline two generalizations, with respect to 1) the number of players (i.e. a many-player simultaneous game in

normal form) and to 2) the number of moves (i.e. a two-player sequential game with two rounds).

First we assume that two strategic players (one per type) are confronted: the Defender, denoted by G , and the attacker or Hacker, denoted by H . G would prefer to patch where there is going to be an attack, while H would prefer to attack a vulnerability where G has not put a patch.

The standard assumption (called *common knowledge*) is that each player has the same full knowledge of the rules of the game, of the strategies available to the opponents, of the payoffs for each state of affairs and of the utility function of all the players. Given this knowledge, the goal of every player consists in adopting the strategies maximizing his/her own payoff, by taking into account that they depend, also upon the other players' chosen strategies. The quest for the optimal strategy typically starts from the elimination of each player's *dominated strategies*. A player's strategy is dominated if for every strategy by her opponent she has another strategy with higher payoff. A mixed strategy is defined by a probability distribution on strategies. Mixed strategies can be compared based on their expected value to the player; they too can dominate one another. Comparing mixed to pure strategies is possible as well. A *Nash equilibrium* is a joint strategy solution that describes a static equilibrium condition of the game; it corresponds to an array of strategies (pure or mixed) – one for each player – (it is called *strategy profile*) such that no individual player would be better off by changing his own strategy unilaterally. The Nash Theorem states that in finite games (finite number of players and of pure strategies) at least an equilibrium always exists, either in pure strategies or in mixed strategies.

A. Simultaneous 2-player 2-vulnerabilities Patching Game

We consider the prototypical normal form of a two-vulnerabilities game (equivalent to a two-place search game): we call the two vulnerabilities A and B . The set of pure strategies for H consists in attacking at A or attacking at B – in short $S_H = \{A, B\}$; the set of pure strategies for G consists in patching at A and patching at B – in short $S_G = \{A, B\}$. A situation where the two players have to decide the single vulnerability to attack/patch before observing the move of the other player is equivalent to a single round simultaneous move game. Payoffs corresponding to the different states of the world take into account the different G 's costs for patching d_A, d_B and H 's costs for attacking c_A, c_B , as well as the losses u_A, u_B suffered by G from successful attacks and the gains s_A, s_B obtained by H from exploiting the different vulnerabilities. We include in the formalization also the finite probabilities α and β of success of an attack respectively on A and B , reflecting the intrinsic strength of the vulnerability. The corresponding payoff bi-matrix is shown in Table II. It consists in $(G, H^\top)$ the two payoff matrices G for player G and H for player H .

$$G = \begin{bmatrix} -d_A & -d_A - u_B\beta \\ -d_B - u_A\alpha & -d_B \end{bmatrix} \quad H^\top = \begin{bmatrix} -c_A & -c_B + s_B\beta \\ -c_A + s_A\alpha & -c_B \end{bmatrix}$$

It is straightforward to see that, in this game, the players' preferences – with the above defined payoff schema – have a circular structure: the parties cannot determine in advance which one is their own best pure strategy, and they will have to resort to a suitable randomization between the two choices. In other words there is no Nash equilibrium in pure strategies:

TABLE I. THE PAYOFF MATRIX FOR A TWO-PLAYER VULNERABILITY PATCHING GAME.

		Hacker (H)	
		attack A (with prob. q)	attack B (with prob. $(1-q)$)
Defender (G)	patch A (with probability p)	$(-d_A, -c_A)$	$(-d_A - u_B\beta, -c_B + s_B\beta)$
	patch B (with probability $(1-p)$)	$(-d_B - u_A\alpha, -c_A + s_A\alpha)$	$(-d_B, -c_B)$

all the pure strategy profiles have at least one player that would benefit from switching strategy unilaterally. As a consequence each party will have to adopt a *mixed strategy* (defined by a probability distribution over pure strategies) fulfilling the so called *indifference condition*: the mixed strategy of a player must be a mixing of his own pure strategies such that the other players expected payoff will not change whatever mix of his own pure strategy is adopted. An individual mixed strategy is represented by probability values. This joint mixed strategy will represent the Nash equilibrium of the game.

With two-players, two vulnerabilities, single attack and single patch, H decides an attack probability q on the vulnerability A (and an attack probability $(1-q)$ over the vulnerability B) and G decides a probability p for patching the vulnerability A (and a patching probability $(1-p)$ over the vulnerability B).

The solution of the game can be found by computing the pair (p, q) such that neither H can improve her expected payoff by deviating from q , nor G can improve his expected payoff by deviating from p . The solution strategy q for H can be worked out by solving the linear system imposing the indifference conditions to G $-d_Aq = (-d_A - u_B\beta)(1-q)$ and $(-d_B - u_A\alpha)q = -d_B(1-q)$ (which include also the probability normalization condition). Similarly the solution strategy p for G can be worked out from the indifference conditions to H $-c_Ap = (-c_A + s_A\alpha)(1-p)$ and $(-c_B + s_B\beta)p = -c_B(1-p)$. The solutions are

$$p^* = \frac{s_A\alpha - (c_A - c_B)}{s_A\alpha + s_B\beta} \quad q^* = \frac{u_B\beta - (d_B - d_A)}{u_A\alpha + u_B\beta}$$

One can see that, in the limit of negligible patch and attack costs, G patches A with a probability proportional to the relative importance it has to H, while H attacks A with a probability which increases with its unimportance to G.

B. Simultaneous 2-players m -vulnerabilities Patching Game

This approach can be extended to the case of a finite number m of vulnerabilities. Again consider a single round simultaneous game with a player of type G who has only a patch available and a player of type H who has the possibility of delivering only an attack. The general $(m \times m)$ payoff matrices for G and H in compact notation can be written

$$G = -D - \hat{U}^\top + \text{diag}(\hat{U}) \quad H = -C + \hat{S} - \text{diag}(\hat{S})$$

The mixed strategy equilibrium can be obtained by solving the linear system defined by the indifference conditions and the probability normalization condition. Let v_G and v_H be the unknown expected payoff at equilibrium for G and H respectively, one can show that the arrays of probabilities and expected payoffs at equilibrium $(p_1, \dots, p_m, v_G)$ and

$(q_1, \dots, q_m, v_H)$ are the solution of the two linear systems

$$\begin{bmatrix} G_{11} & \dots & G_{1m} & -1 \\ \dots & \dots & \dots & -1 \\ G_{m1} & \dots & G_{mm} & -1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ \dots \\ p_m \\ v_G \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} H_{11} & \dots & H_{1m} & -1 \\ \dots & \dots & \dots & -1 \\ H_{m1} & \dots & H_{mm} & -1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ \dots \\ q_m \\ v_H \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

C. Simultaneous r -attackers m -vulnerabilities Patching Game

The game, with r identical non-communicating attackers and 1 defender can be solved based on the previous one, taking into account the following considerations. In a simultaneous single-round game, the identical attackers cannot directly interfere with one another: they can attack and exploit the same vulnerability if it is not patched and will be equally unsuccessful when attacking a patched vulnerability; one attacker move has no consequence on another attacker's payoff. If they have identical utility functions, the *defender's strategy* that makes one attacker indifferent, makes all indifferent. In a sense, such an r -attackers game corresponds to r independent 1-attacker games. This translates also in the fact that an *attacker's equilibrium strategy* corresponds to the one of the 1-attacker game. What changes is the expected loss suffered by the defendant, which will be proportional to the number of independent games, i.e. to the number r of attackers. The solutions change when the attackers are not identical in available strategies and utility functions, since they will interact indirectly, through the response strategy of the defendant, which will take into account all of them.

D. Two Rounds 2-Players 2-Vulnerabilities Patching Game

A model with more than one round can be used to model the confrontation between a Hacker (H) and a Defender (G) where not only H gains knowledge from a previously delivered attack and uses it to sharpen a subsequent attack, but also G learns from the detected attacks to strengthen his own defences. While in a simultaneous move single round game a player knows nothing of the other players' chosen moves, in a sequential game players typically alternate their moves and can observe perfectly or at least partially the moves of the other players. This fact can be efficiently represented by using a game tree to account for the ordering of the moves: this tree is the basis of the so-called extensive form representation.

In our toy model of a sequential game Vulnerability Patching we assume that the players move in two turns, in the sequence (G,H,G,H). This might correspond to the following scenario: G chooses the first patch before going on-line; H gets to know the system and chooses whether to attack and where; G can detect or not the attack and can adopt some patching choice; finally H can try again to exploit the knowledge acquired so far. Each player at each turn can choose among doing nothing, acting on A, acting on B: $\{A, B, \emptyset\}$. We added the obvious constraints that a vulnerability cannot be patched twice nor attacked twice. Variants that can be devised

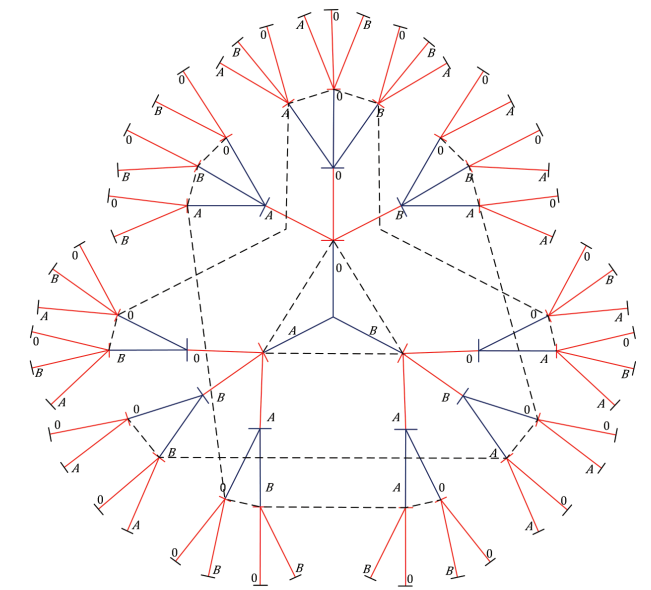


Fig. 1. Two round 2-players 2-vulnerabilities Patching Game in extensive form (top) and normal form (bottom). The 1st and 3rd generation of the tree represent (blue) the moves by the defender (G), the 2nd and 4th (red) represent the moves by the attacker (H). The dashed lines represent H's information sets (ISs). Nodes connected by an IS line correspond to states where the attacker has the same partial perception of the world (based on it she cannot tell at which node of the IS she is located). ISs for player G are drawn in Figure 2.

depending on the awareness by G of a received attack and the subsequent obligation to patch.

Since their rationality allows them to do so, the players can consider all the possible sequences of moves (*First, Second*) beforehand: each sequence of moves along the tree becomes the equivalent of a single pure strategy in a simultaneous decision game and can be represented in normal form. Extensive form and the normal form of the presently discussed game are given in Figure 1. The solution of the game in the form of a mixed strategy profile can be found as in the previous subsections by solving the corresponding linear system.

III. CONCLUSIONS AND OUTLOOK

In this work we observed that the problem of patching vulnerabilities can be formulated as an optimization problem within a strategic players' scenario: we called this formulation Enhanced Patching Problem to distinguish it from the Decision Theory formulation. We pointed out that this problem could be modeled by the tools of Game Theory. We mapped the problem onto a variant of the class of models known as Search Games and called this subclass (characterized by the semantics of the setting) Vulnerability Patching Games. We studied some prototypical instance of this class, with varied choices in terms of simultaneity or sequentiality, number of vulnerabilities and number of attackers. Those games provide some insight into the general features of the solutions.

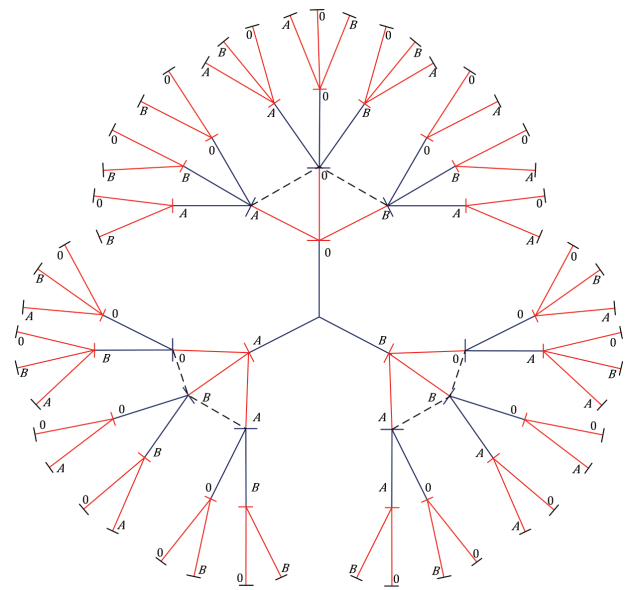


Fig. 2. Two round 2-players 2-vulnerabilities Patching Game in extensive form. See Figure 1 caption for further details. Here dashed lines represent the Defender's information sets. The two figures were realized by a purposely developed tool.

The accuracy of those models can be improved by introducing further elements such as the differences among attackers in terms of available strategies or utility functions, by extending the depth of the time dimension in sequential games. Further modeling steps toward realistic settings would include introducing probabilistic or possibilistic uncertainty in the parameters of the game, and lifting the assumption of complete knowledge of the game setting by the players.

Acknowledgements The work was partly supported by EU within the PRACTICE project (contract n. FP7-609611), by the Italian MIUR project SecurityHorizons (c.n. 2010XSEMLC) and by the CMIRA2014/AccueilPro and COOPERA program of the Region Rhone-Alpes, France.

REFERENCES

- [1] Al Housani, H.; Otrouk, H.; Mizouni, R.; Robert, J.-M.; Mourad, A., *Towards Smart Anti-Malwares for Battery-Powered Devices*, New Technologies, Mobility and Security (NTMS), 2012 5th International Conference on , vol., no., pp.1,4, 7-10 May 2012
- [2] Luca Allodi and Fabio Massacci, *My Software has a Vulnerability, should I worry?*, arXiv preprint arXiv:1301.1275, 2013.
- [3] Steve Alpern and Shmuel Gal, *The theory of search games and rendezvous*, Springer Science & Business Media, vol. 55 (2003)
- [4] Hau Chan, Michael Ceyko, and Luis E. Ortiz, *Interdependent defense games: Modeling interdependent security under deliberate attacks*, arXiv preprint arXiv:1210.4838 (2012).
- [5] Aron Laszka, Mark Felegyhazi and Levente Buttyán, *A survey of interdependent security games*, CrySyS 2 (2012)
- [6] Mitre, *Common Vulnerability Exposure (CVE)*, cve.mitre.org, 2015.
- [7] Andrey Garnaev, *Search games and other applications of game theory* Vol. 485. Springer Science & Business Media (2000).
- [8] M.H. Manshaei et al. *Game theory meets network security and privacy*. ACM Computing Surveys (CSUR) 45.3 (2013): 25.
- [9] Mell, P., Scarfone, K., and Romanosky, S., *A Complete Guide to the Common Vulnerability Scoring System Version 2.0*, Forum of Incident Response & Security Teams (FIRST), www.first.org/cvss/cvss-guide
- [10] National Institute for Standard and Technology, *National Vulnerability Database - NVD*, https://nvd.nist.gov/home.cfm, 2015.
- [11] Liang, Xiannuan, and Yang Xiao. *Game theory for network security*. Communications Surveys & Tutorials, IEEE 15.1 (2013): 472-486.