Can Non-Cooperative Game Theory Assist Modeling Security Implementation for Back-End Web App Development?

T. Koetting

E–mail: *tkoettin@stumail.jccc.edu*

Abstract

In active development environments of web-based applications, back-end engineers and developers are constantly doing battle with bad actors probing for exploits. We explore the potential of non-cooperative game theory as a framework for enhancing back-end web development security and resource management. Most traditional strategies implemented by companies focus on static defense and reactive methods. By adopting a perspective influenced by game theory, we delve into the back and forth nature of the development timeline. We assume that the two exclusive contributors to the game in this framework are the development team and the bad actors. We finally explore whether or not the model being implemented should have characteristics aligned with flexibility or rigidity, and what influence their implementation environment has on the model's final structure.

Key words. Game Theory – Web-App Development – Development Models

1. INTRODUCTION

Game theory has found myriad uses in security and infrastructure planning within organizations. Many aspects of non-cooperative game theory have been explored in network and IT with the two prominent model categories being security attack-defense analysis and security measurement [**1**]. For most purposes within security, most of these models are decidedly competitive and therefore non-cooperative due to the fact that both attackers and defenders are working to maximize their own payoff at the expense of their adversary [**13**]. There is also the given assumption depending upon the model used of known information, with the most realistic and computationally expensive models being Bayesian Games, which fall under the category of games of incomplete information, but Bayesian games also allow for knowledge of the payoffs of the adversary on both sides of the game, a situation which parallels active security efforts and offensive plans by the defending networks and attackers alike [**14**]. Through an evaluation function the central defender can make decisions about which nodes and plans of action to take in terms of defensive strategy. Within a network architecture, there is a need for multiple end point defense with moving attacks targeting different network nodes required distributed responses and strategies that are capable of choosing the most formidable defense while expending the least resources. These scenarios are also highly applicable to the exploitation prevention measures taken by defending web applicable developers, however, security research has sparsely touched game theory's applications to software level environments and situations [**2**], with the bulk of the research focusing on active threats within IT infrastructure and the active reactions taken by the defending player and by the attacker [**3**]. The planning of infrastructure, whether classical or IT infrastructure, can be shown to have some applicability to development, in the sense that given a significant element of software where exploits or vulnerabilities may exist given inability to completely close off offensive opportunities due to budget constraints, where, for example given [**3**] and reinforced power lines capable of being disconnected by y_c cyber-attacks, we get the resulting equation:

$$p_{P|R} = \frac{f_P}{1 + N_L[y_c - x_c]_+} \tag{1}$$

Where N_L is the number of power lines, and where x_c are the respective reinforcement of the component being attacked. So for components given the model derived from [**3**], we can understand the components as web application elements, where y_c could stand for a non-specific attack particular deployment

of Laravel, and where x_c would be the respective development hours spent on security and fortification implementations. Extensive work has also been done in the field dealing with software business competition, licensing dynamics [**4**], and development expenditures in relation to business decisions, with its main focus trending towards management science rather than security [**5**]. The most applicable model within the management frameworks is that proposed by [**4**] on the game theoretical models applying to software functionality considerations in monolpoly and duopoly markets, where higher software functionality has decreasing return and greater strain depending on the development team size and budget. When given the functionality Q and its second order derivative Q" :

$$Q^{"} = \frac{2}{5M}((V+M) + \sqrt{(V+M)^2 - 5VM}) \quad (2)$$

Where the payoff of Q" is given by π" where:

$$\pi^{"} = \frac{2M-8V}{25M}((V+M)+\sqrt{(V+M)^2-5VM}) - \frac{V}{5} \quad (3)$$

Where in a competitive market the given payoff is dependant on not only the other market participant but software functionality and the consumer demand. While we are not necessarily concerned with consumer demand, we can model prospective returns from security by altering the payoffs to instead of profit, of conditionally decrementing vulnerability to attacking players. Web-app development has concerns with active IT infrastructure but also software development management, where neither school of thought adequately handles the unique issues presented particularly by the use of Ruby on Rails, Microsoft's .NET, Java, or other programming languages to deploy web applications, which are vulnerable to IT based attacks as well as exploits found within the programs themselves [**15**]. Parallels can however be found within the different fields, where the [**6**]. approach presents a firms optimal security investment as satisfying Equation (3):

$$(\frac{\delta s(v,z,c_h(z))}{\delta z})_{z=z^*_{SE}} = -\frac{1}{L} \quad (4)$$

Where z^*_{SE} is the optimal investment level, with $s(v,z,c_h(z))$ denoting the investment level decisions. Within the model the main determination of resource allocation depends upon the expectation of hacker effort. However, this model assumes static vulnerability and static developer input, with changes in security investment reflecting broadly IT-focused applications. As for management applications utilizing game theory models, within situations such as price wars or contract bidding [**7**], where the assumption is that all games are zero sum. The zero sum assumption conflicts though directly with a model depicting debugging resources, where a nuanced approach is required to reflect the complexity of the model's inherent real time back-and-forth interactions between defending players and attackers. This is in contrast with the model proposed by [**6**], which only concerns itself with the shifting allocation strategy given expected offensive coordination by the attackers, where as [**7**] delivers a model that emphasizes highly dominant strategies even if the the attacker is currently pursuing a dominated strategy.

2. Resource Management and Software Development

However, a consideration must also be factored in which applies mostly to software features, software optimization, and functionality provided to the end user [**16**]. Within a model with dense features and functionality, more variables must be considered when reviewing potential security risks, and therefore the payoff and cost of features [**4**]. When deciding whether to include function x, the decision maker should consider the strategic important of x not just in the marketability of the final product but also in the cost to secure x's implementation in the final product would be and what the cost of securing x would be on top of the cost of adding it in the first place. While function rich web applications are taxing on development resources, they would drain resources quicker given the proper security and optimization, which would inevitably increase the size of the development team and thereby the cost [**18**]. Total cost reductions as a result of factoring inefficiencies, θ_i^B θ_i^C, and effort as a variable of wages and working hours, the resulting equation from [**19**] is:

$$(1 - \sum_{i=1}^{540}(\frac{\theta_i^B}{\theta_i^C} * \frac{\textit{Actual Effort for ith Project}}{\textit{Total Effort}})) \quad (5)$$

Given the ability to estimate inefficiencies in an abstract manner, and integration into a broader model of calculating strategic investment in a software development team, or that of a team working on development of a web-application, is needed. However, based upon the development team then, a law of diminishing returns is applied such that scale and budget of the corporation and development team creates a higher ceiling able to be reached by adequately securing web applications before diminishing returns on the budget expenditures are reached [**19**] where the base diminishing returns model (BDRM) is shown as follows:

$$\text{MinZ} = \text{OH}x_n + \sum_{ij \in A}(c_{ij}r_{ij}) + \sum_{ij \in A} F_{ij} \quad (6)$$

given the specific activity-on-arc environment outlined in [**19**], [**8**]. Establishing the diminishing re-

turns of the respective model and integrating it into a functional model of security implementation is ideal, since overcompensating for security weakness by excessive investment does not yield in the same level of security enhancement at an instantaneous point over time. The factoring of the other variables within development costs also requires expansion of the dimensions of the Nash Equilibrium, where each budgetary decision has an impact on the corresponding level of investment and vulnerability to attackers of the web applications, whether directly concerning security and optimization or not [**17**].

3. Moving Target Defense

These abstract models lack the defensive tactics needed to competently repulse attackers in a web application setting, they however allow for calculation of risk and reward of resource usage in management and distribution of related strategies, but are too broad to be properly implemented to have any positive affect on the security of the application. For web applications, a distributed and highly malleable framework is needed. Works have provided frameworks for web application development to implement systems that utilize moving target defense to combat not only static vulnerabilities but active vulnerability reconnaissance by the attackers [**9**]. Much work has been carried out in terms of the utility of a Moving Target Defense (MTD) for web applications alongside general security research for web application development and deployment. However as outlined by [**10**] the recurring oversight by pre-existing models is neglectful of the switching costs when altering configurations of the defense. Within the allocated resource perspective, while the switching cost may be computationally expensive, it is factor-able into the final development cost based upon the cost and time required to create an efficient and scalable algorithm which implements an MTD framework. In calculating the incentive reward structures for the MTD game theoretical model.

For simultaneously anticipating not only direct attack moves but also reconnaissance, where the switching can be modeled as a Stackelberg competition [**11**], where the leading party, the attacker, can perceive and understand the reaction of the switching of the defensive configuration for each attacking move. Calculating the Stackelberg equilibrium thus becomes a difficult task requiring of almost unimaginable computational power due to the variety and diversity of attacker strategies and sources. Where the Therefore optimization and efficiency in programming of the models are necessary and the optimization problem is given in Equation 2 using the Decomposed Optimal Bayesian Stackelberg Solver, but again such programming would require increasing costs in development to decrease costs in server usage and runtime concerns. [**10**]

$$max_{x,n,v} \sum_{c \in C} \sum_{\theta_{2i} \in \theta_2} \sum_{a \in A_{\theta_{2i}}} P_{\theta_{2i}} R^D_{a,\theta_{2i},c} x_c n_a^{\theta_{2i}} - \alpha \sum_{i \in C} \sum_{j \in C} K_{ij} x_i x_j \tag{7}$$

Again because of the inverse relationship between optimizing time in development and we should expect Nash Equilibrium models to include multiple variables on both axes, assuming that the game is played with two player classes of defender and attacker, represented respectively in Equation (2) by x and n. The strategy of determining MTD configurations from the Bayesian Stackelberg model presented above is thus more effective in most simulations which have switching costs from configuration alterations as a significant factor within the modeling process.

4. Active Defense

Within the models of game theory proposed for digital security, most focus on modeling non-cooperative games is based upon the idea that it will emulate non-cooperative games where one player is an enterprise network structure and its administrators and the other player is an attacker. Most models unfortunately lack the assumptions of restricted information access, instead choosing to either games which are static where information is incomplete, or games which are dynamic which have perfect information [**12**]. Neither is realistic, with real development cycles and active maintenance as well as web application penetration and exploitation being dynamic synchronous activities where neither party is fully aware of all information. This can lead to serious problems with applying game theory to digital development with static models, since the opponent has no time constraints, no movement constraints on a turn-by-turn basis, and payoffs and goals are not a consistent Nash Equilibrium since payoffs are different depending on which side of the game the player is on. So while network security models may lack this trait, they are more deployment ready than the models based purely upon investment which takes a purely economic and strategic approach without having innate reactivity or flexibility. Static models can be useful given varying model scenarios, where non-static strategies or models may result in exponential growth of computational complexity from the synchronous action of all game participants as well as the nuanced factors and inputs given by both players which factor into varying payoffs. Within a software development scenario, static models can be applied given the active repair of websites after discovery of exploits, however this may not cover all necessary situations, since in early development cycles (i.e. beta, alpha releases) and post-formal-release, there may be a non-static non-cooperative game taking place with the two players being the defending software developers and the attacking hackers. [**20**] Given the at-

tacker's and defender's mixed strategy:

$$(\delta_i, 1-\delta_i) = \sum_{T=1}^{\infty} \sum_i (\frac{P_a - C_a^T - C_w^T}{P_a - C_i^T + U_a}, \frac{U_a - C_w^T}{P_a - C_i^T + U_a}) \quad (8)$$

$$(\sigma_i, 1-\sigma_i) = \sum_{T=1}^{\infty} \sum_i (\frac{C_m^T}{C_i^T}, \frac{C_i^T - C_m^T}{C_i^T}) \quad (9)$$

Payoff of the game model presented in [**20**] is:

$$\mathrm{U}_{game} = \sum_{T=1}^{\infty} \sum_{i=1}^{\lambda} U_I = \sum_{T=1}^{\infty} \sum_{i=1}^{\lambda} (C_i^T \theta_i \sigma_i + U_i - C_m^T \theta_i - C_i^T \sigma_i) \quad (10)$$

Given the set algorithm and pattern of the strategies implemented by the defenders or attackers, and the thus predictable payoffs, we can conclude that while useful in their ability to be implemented at scale on large web applications which have high resource use, the lack of adaptability and mobility may become a hindrance rather than an asset given some scenarios of highly mobile attackers traversing large swathes of a web application in a matter of seconds.

5. Further Research

Within the field we have seen significant investment into network infrastructure planning and development using game theory and algorithmic models implemented in web service platforms to mitigate risk and create greater protection efficiency. However, these models lack an integration of managerial game theory which we have discussed, with the focus being primarily on the implemented software and models rather than the actual resources expended in developing them and the various payoff considerations that may be a result of this. We anticipate a large possible field of study where managerial and resource allocation decisions can be adaptable within a game theory framework using different model styles and techniques. Further research may also be conducted specifically to study the game theoretical implications of applying these aforementioned management models in high-tech and digital industries where most resources expended in the pre-product development phase are on software developers and related costs, and which security implementation would factor in as a large cost factor due to its use of these developers. Payoffs of a fiscal nature are to be considered alongside the security payoffs, with the emulation of a complex business environment allowing for innovation in niche models which follow game theoretical principles.

References

1

X. Liang and Y. Xiao, "Game Theory for Network Security," in IEEE Communications Surveys & Tutorials, vol. 15, no. 1, pp. 472-486, First Quarter 2013,
2

Julie E. Kendall, Kenneth E. Kendall & Matt Germonprez (2016) Game theory and open source contribution: Rationale behind corporate participation in open source software development, Journal of Organizational Computing and Electronic Commerce
3

N. S. V. Rao, C. Y. T. Ma, F. He, J. Zhuang and D. K. Y. Yau, "Cyber-physical correlations for infrastructure resilience: A game-theoretic approach," *17th International Conference on Information Fusion (FUSION)*, Salamanca, Spain, 2014, pp. 1-8.
4

Kai Lung Hui, Kar Yan Tam (2002)Software Functionality: A Game Theoretic Analysis,Journal of Management Information Systems,19:1,151-184,
5

Mattos, E., Vieira, M., Schmitz, E. A., & Alencar, A. J. (2014). Applying game theory to the incremental funding method in software projects. *Journal of Software*, *9*(6), 14-35.
6

Cavusoglu, H., Mishra, B., & Raghunathan, S. (2004). A model for evaluating IT security investments. *Communications of the ACM*, *47*(7), 87-92.
7

Shubik, M. (1955). The uses of game theory in management science. *Management Science*, *2*(1), 40-54.
8

Bang, S. K., Chung, S., Choh, Y., & Dupuis, M. (2013, October). A grounded theory analysis of modern web applications: knowledge, skills, and abilities for DevOps. In *Proceedings of the 2nd annual conference on Research in information technology* (pp. 61-62).
9

Lei, C., Zhang, H. Q., Tan, J. L., Zhang, Y. C., & Liu, X. H. (2018). Moving target defense techniques: A survey. *Security and Communication Networks*, *2018*.
10

Sengupta, S., Vadlamudi, S. G., Kambhampati, S., Doupé, A., Zhao, Z., Taguinod, M., & Ahn, G. J. (2017, May). A Game Theoretic Approach to Strategy Generation for Moving Target Defense in Web Applications. In *AAMAS* (Vol. 1, pp. 178-186).
11

Li, T., & Sethi, S. P. (2017). A review of dynamic Stackelberg game models. *Discrete & Continuous Dynamical Systems-B*, *22*(1), 125.
12

Roy, S., Ellis, C., Shiva, S., Dasgupta, D., Shandilya, V., & Wu, Q. (2010, January). A survey of game theory as applied to network security. In *2010 43rd Hawaii International Conference on System Sciences* (pp. 1-10). IEEE.
13

S. N. Hamilton, W. L. Miller, A. Ott, and O. S. Saydjari, "The role of game theory in information warfare," Proc. 4th information survivability workshop (ISW-2001/2002),
14

R. Gibbons, Game Theory for Applied Economists. Princeton University Press, 1992
15

Chaudhuri, A., & Foster, J. S. (2010, October). Symbolic security analysis of ruby-on-rails web applications. In *Proceedings of the 17th ACM conference on Computer and communications security* (pp. 585-594).
16

Sohrabi, M. K., & Azgomi, H. (2020). A survey on the combined use of optimization methods and game theory. *Archives of Computational Methods in Engineering*, *27*(1), 59-80.
17

Jorgensen, M., & Shepperd, M. (2006). A systematic review of software development cost estimation studies. *IEEE Transactions on software engineering*, *33*(1), 33-53.
18

Pendharkar, P. C., & Rodger, J. A. (2009). The relationship between software development team size and software development cost. *Communications of the ACM*, *52*(1), 141-144.
19

Deckro, R. F., & Hebert, J. E. (2003). Modeling diminishing returns in project resource planning. *Computers & industrial engineering*, *44*(1), 19-33.
20

Wang, K., Du, M., Yang, D., Zhu, C., Shen, J., & Zhang, Y. (2016). Game-theory-based active defense for intrusion detection in cyber-physical embedded systems.