An Optimal Decision-Making Method for Cyberspace Countermeasure Based on Game Theory

Tian Chen[1,a], Ming Xian[1], Jun Gu[1,b], Huimei Wang[1], Ruixiang Du[1]

[1]State Key Laboratory of Complex Electromagnetic Environment Effects on Electronic and Information System, National University of Defense Technology, Changsha, 410073, China

[a]email:1402305267@qq.com, [b]email:54185011@163.com

Keywords: Decision-Making; Game Theory; State Tree; Computer Network; Course of Actions

Abstract. The traditional Decision-Making methods leverage the awareness from one side of operators and evaluate the state of environment and devices statically. However, as the correlations of cyberspace's actions become more and more complex and the cyberspace confrontation is always changing, it cannot meet practical requirements. To solve these problems, a tree-based search algorithm is put forward in the paper, which exploits the correlations between actions and scores all plausible courses of actions dynamically, and gives optimal tactics advice for both the defenders and attackers. Additionally, it can reduce the cost of resources for operators to decide their actions. Experimental results show the feasibility and effectiveness of our proposed Decision-Making method.

Introduction

Decision-Making is regarded as the cognitive process resulting in the selection of a belief or course of action among several alternative possibilities [1]. Game theory is the study of mathematical models of conflict and cooperation between intelligent rational decision-makers [2]. In recent years, it has been applied to the domain of computer network to study the security thereof. In the wake of computer network (CN) attack technology development, the intellective, coordinated and diversified attack methods are in vogue, at the same time, the defense technology is booming accordingly. This is a game between CN attacker and defender. How to find an optimal tactic to effectively reduce security risk or increase the rate of attack success and cut down the cost during the action is the key problem for player [3].

RoyS et al. [4] have token a review of game theory's application in CN security domain according to different game models. YanFen et al. [5] described the CN situation awareness based on game theory. They define the game parameter by the manager's evaluation of the network node importance, lead to a little subjectivity of the method, and what's more, they paid too much attention to the side of defense. As the environment of CN is dynamical and the opponent moves are unpredictable, WangChunlu et al [6] put forward a Random Game model combining the random Petri Net model, have resolved sophisticated and dynamical network confrontation. Allen Ott et al [7] introduced a mathematical search method—Themistocles engine, which solved the problem of time relative and error-tolerant of game models.

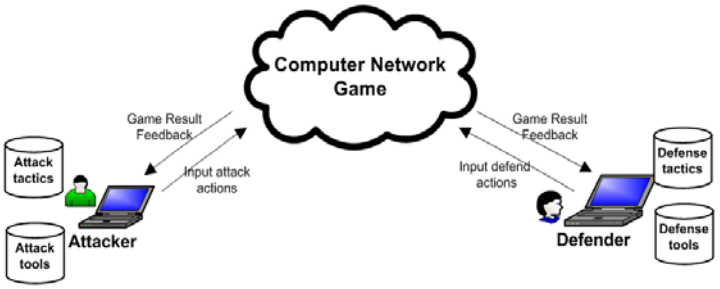


Figure.1 Diagram of a Brief View of Game Process between Two Players

In this paper, a tree-based search method based on game theory has been introduced to optimize tactics in network confront move for both attacker and defender dynamically. This method can

advise a particular player select future course of actions based upon their estimate of current state.

The remainder of this paper is organized as follows. Section 2 presents the structure and the key search algorithms for optimal move. Next, the section 3 gives an example of the algorithm's employment and analysis the result. Finally, we conclude in section 4.

Structure and Algorithms

The Figure.2 shows the structure of our Decision-Making methods, The "Game" starts when Human/Computer selects the game action queues. The action queue searching methods and scoring algorithms are the key component of our work. We output every plausible COAs and their corresponding scores. After all plausible COAs are ergodic, the game over, and we can get a recommended COA that may be useful for realistic decision maker.

Definitions.

State: State is the set of all variables and their associated values needed to identified the particular device situation in a given time (e.g. device on/off state, FTP service available/unavailable, Patch version, risk, etc.). During the interest time the state changed.

Move: A relatively small set of steps that can execute on physical device (e.g., port scan, restore system, etc.). We generate a move after the system given corresponding advice for operators.

Course of Action (COA): Consist of one or more individual moves taken by each player at a given stage of game, starting from their estimate of the current game state S(n) at time n.

Interesting Time: The time when a move is executing or a move is detecting by opponent.

State estimation: To estimate own and opponent's future state based on the existing information and expert experience.

Count function: A count function is a utility function for player to count the state-related score for a predictable action.

Score: The judgment standard of resource cost. The move score valued by previous experience, while the COAs score valued by algorithms.

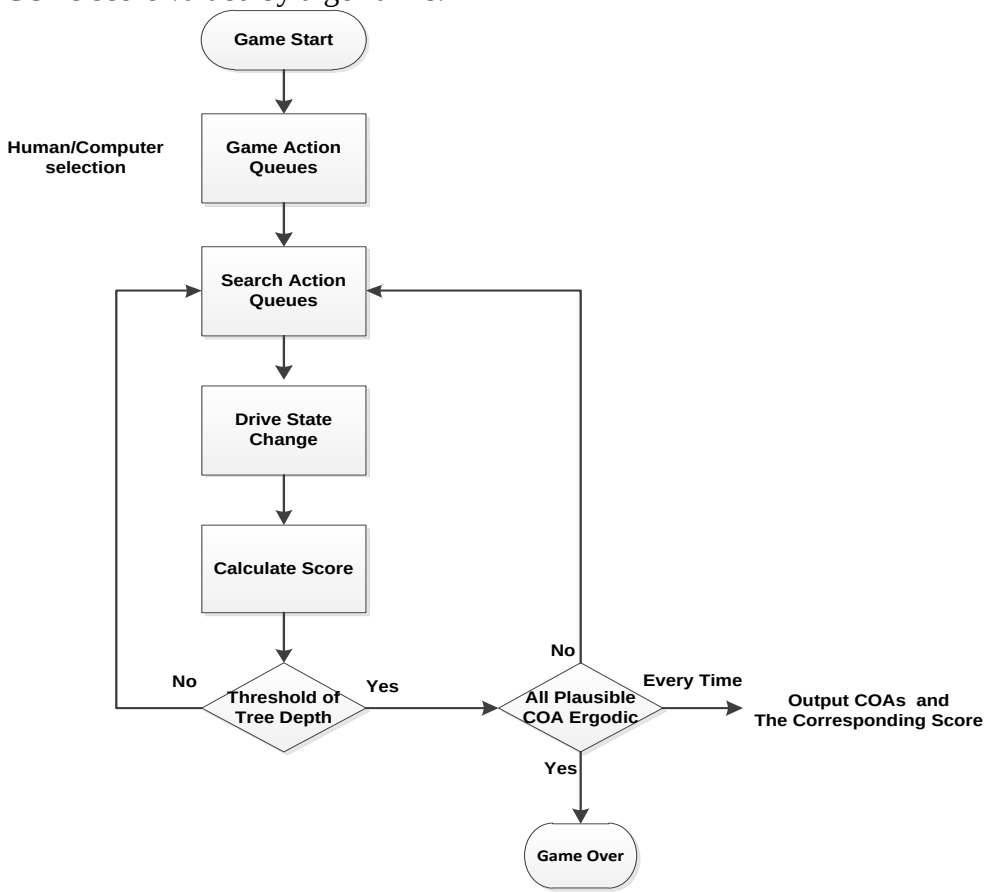


Figure.2 Diagram of Optimal Decision-Making Method Architecture

A Tree-Based search method.

As it mentioned above, the action queue searching method and the scoring algorithms is critical, we studied a tree-based search method based on game theory. The Tree-Based search method is main to give advice for player as to the selection of future moves based upon their estimate of current state S(n). The count function counts the scores which is mapped from the state variable values for a particular move as well as the corresponding impacts on operators and their opponents. These scores are expected in the scale on the interval[-100,100], the negative value represent for the cost of system state value while the positive value represent for the benefit .While we consider the COA score we should normalize them back into unit values[0,1].

In Figure.3, the search starts with the root node at the state S(n) at time n, and the Child node $M^{d}_{i,j}$, as d=1,2,D, where D is the max depth of the tree which depend on the scale of state value interval, i indexes the child nodes at a given level k (and k=1 refers to the root node) , while j indexes i. We suppose the state utility value for red attacker is $\mathrm{P}\left(S^{r}_{i,j}(n+d)\right)$, so the normalized unit value $\mathrm{U}\left(S^{r}_{i,j}(n+d)\right)$ can be calculated as follow:

$$\mathrm{U}\left(S^{r}_{i,j}(n+d)\right)=\frac{\mathrm{P}\left(S^{r}_{i,j}(n+d)\right)}{\sum_{j}\mathrm{P}\left(S^{r}_{i,j}(n+d)\right)} \tag{1}$$

The corresponding nodes' score $F^{r}_{i,j}(n+d)$ can be calculated as follows:

$$F^{r}_{i,j}(n+d)=\begin{cases}\mathrm{U}\left(S^{r}_{i,j}(n+d)\right), d=1\\ F^{r}_{i,j}(n+d-1)\mathrm{U}\left(S^{r}_{i,j}(n+d)\right), d>1 \; and \; F\geq F_{threshold}\end{cases} \tag{2}$$

Where $F_{threshold}$ refers the threshold of a COA score. According to the formula above, we noticed the value $\mathrm{U}(\cdot)\leq 1$, so $\mathrm{F}(\cdot)$ will not increase, on the contrary, decreased at most of the time. Therefore, the threshold is smaller, the time to search a particular COA is longer.

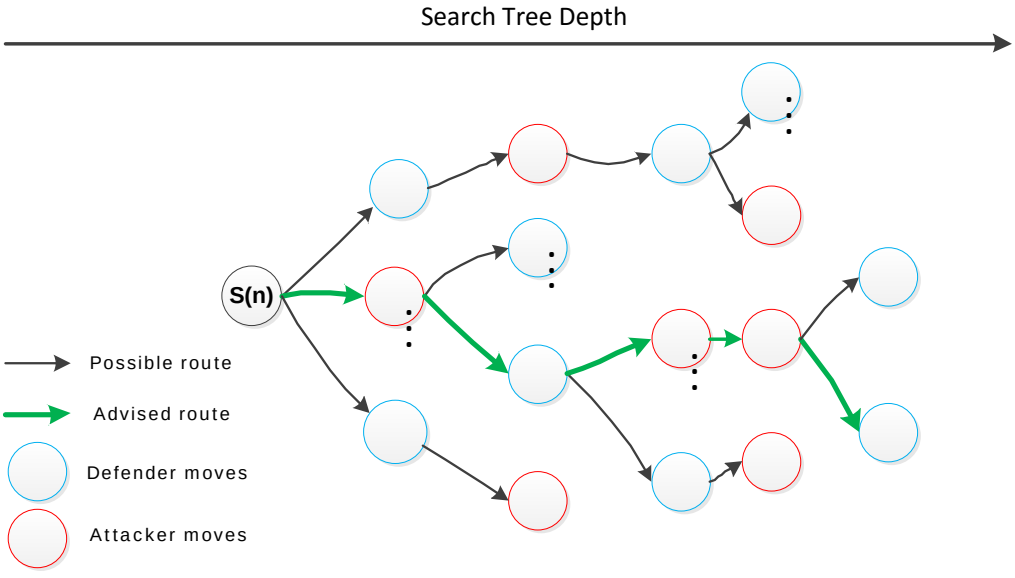


Figure.3 Diagram of a search tree for the advised optimal attack/defend routes

To test the effects of each move and manage the time clock, we introduce the action queue. Once a particular move is added to the action queue, the resulting state is calculated and stored. We note this search tree can score all plausible paths for actions, therefore, we can choose the highest scoring leaf node as the advised COA and can provide choice for Blue defender and Red attacker to optimize their tactics.

Time Sequence Processing.

As there are two players, when the tree-search algorithm runs, the system will have two interesting time queues for the attacker and defender accordingly. There need a synchronization

when the moving generated. In traditional game, the operators alternate moves, however, moves may be executed at the same time. To solve this problem, we use an untraditional methods mentioned in [8], which serialize the simultaneous chosen moves and will not take any action before an interesting time finished.

Moves choices for attacker and defender.

Table.1 shows some typical moves and their impacts of the red offensive player and blue defensive player.

Table.1 The Red Player and Blue Player typical moves and their impacts

Attacker Moves	Move Effect	Impacts	Defender Moves	Move Effects	Impacts
Modified Data	Corrupt data in database	Tripwire alert	**Analyze System Logs**	Logs viewed	None
Port Scan	Determine the host IP on subnet	Traffic analyzer alert	**Restore System**	Back to standard start state	Lost conn-ection to Red Player
Setup Bot	Takeover a machine	None	**IP filter**	Blocks a given IP address	SYN flood stop working
SQL Injection	Gain root privilege	None	**Investigate shutdown**	Determine legitimate of Host shutdown	None
SYN Flood	DDoS by sending lots of TCP SYN packets	Network slow or service unavailable	**Notify Security**	Security team on alert	More extreme counter moves

The moves initial impact score is assigned by experts.

Examples

Scenario setting:

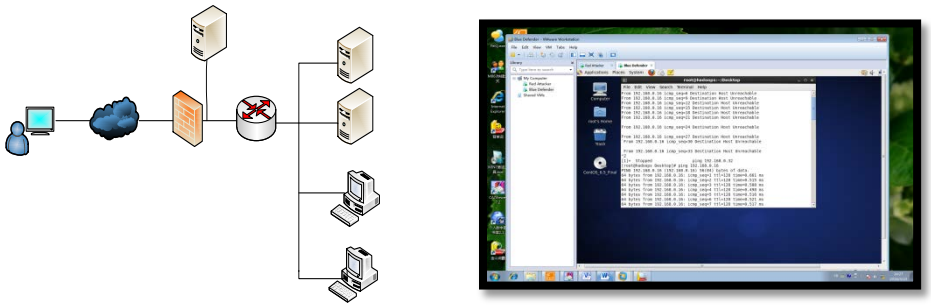
This example is about a two-player game. The Red player represents for the attacker while the Blue player represents for the defender. The Red player is intend to exfiltrate data from the Blue player's host computer, and monitoring the Blue player's Screen while not be discovered. The Red player's task is preventing the red player's attack, and reverse osmosis when the player consider it necessary. The depth of search tree is 10.

Table.2 The Red Player and Blue Player Move Setting

Players	Moves Setting
Red Attacker	Setup external proxy, FTP Scan, Exploit FTP, Upload Hostile Software, Login via backdoor, Ping subnet internal, Install rootkit, Exfiltrate data, Modify data, etc.
Blue Defender	Analyze system logs, Analyze data logs, Install INDS, Scan subnet for vulnerability, Harden system, Deploy honeypot, Apply patches, etc.

Environment Setting:

Windows OS 2007, VMware Virtual Machine Interface, CentOS-6.5-x86_64.



(a) (b)

Figure.4 The figure (a) shows the environment topology of network and the figure (b) shows that VMMs can communicate normally in experiment.

Result:

The recommended course of actions searched by the method this paper designed can see Table.4:

Table.4 The recommended course of action by the Decision-Making method

Number	player	Actions	Number	player	Actions
1	Red	FTP Scan	1	Red	FTP Scan
2	Blue	Analyze system logs	2	Red	Upload Hostile Software
3	Red	Exploit FTP	3	Red	Remain Under Cover
4	Red	Ping subnet internal	4	Blue	Analyze system logs
5	Red	Login via backdoor	5	Blue	Analyze data logs
6	Blue	Scan subnet for vulnerability	6	Red	Upload Hostile Software
7	Blue	Analyze data logs	7	Blue	Install INDS
8	Red	Upload Hostile Software	8	Blue	Harden system
9	Blue	Deploy honeypot	9	Red	Exfiltrate data
10	Blue	Restore System	10	Red	Monitoring

Analysis:

In Table.4 shows two recommended COAs for the red attacker and blue defender respectively, the left column is for the blue player while the right column for the red player. The blue player restore the system because it realized intrusions by deploying the honeypot, so all possible hostile software will be cleared, the blue player's task failed. The right column shows a minimum cost COA for the red player, it recommend upload hostile software once finding an available port, and remain under cover for opportunity, the COA recommended completes the task in costless way finally.

Conclusion

To reduce the risk of computer network security and give optimal tactics for the operators, the paper suggests a Decision-Making method based on game theory to find an optimal scheme for cyberspace confrontation.

According to the analysis of methods and examples, the Decision-Making method based on game theory proposed in the paper shows a well performance in advising cyberspace confrontation course of action. The attacker and defender can make optical tactic in realistic world depend on the

scenarios built by the method, and it is reduce cost of resource.

However, the realistic world cannot simply map into two-player game and the score setting is not accurate for a particular move, so the future work should pay more attention to multi-player game in cyberspace and should consider a self-adaption for scoring system based on the course of actions applications in realistic world.

Reference

[1] Triantaphyllou, Evangelos. Multi-criteria Decision Making Methods: a comparative study. Applied Optimization. Dordrecht, Netherlands: Kluwer Academic Publishers. 2000

[2] Myerson, Roger B. Game Theory: Analysis of Conflict[M], Harvard University Press, 1991.pp.1.

[3] Liu Gang, Zhang Hong, Li Qian mu, Network Security Optimal Attack and Defense Decision-Making Method Based on Game Model[J], Journal of Naming University of Science and Technology, 2014, 38 (1).

[4] oyS, EllisC, ShivaS, et al. A survey of game theory as applied to network security [C]. Proceedings of the 43rd Hawaii International Conference on System Sciences. Washington DC, USA: IEEE, 2010.pp 1-10.

[5] YANFen, YINXinchu, HuangHao. Research on Establishing Network Intrusion Based on MLL-AT [J]. Journal on Communications, 2011, 32 (3).

[6] WangChunlu, WangYaneheng, Dung Yi ni and Zhang Tianle. Novel Comprehen- sive Network Security Assessment Approach[C]. InProc. of IEEE ICC, 2011.

[7] Allen Ott, Alex Moir and John T. Ricka. A Game Theoretic Engine for Cyber Warfare[J]. Studies in Computational Intelligence , 2014.

[8] Ronald R. Yager Marek Z. Reform at Naif Alajlan. Intelligent Methods for Cyber Warfare [M]. Springer. 2014, pp119-123.