Game Theory for Security: Past, Present and Future

CYBER 2020 Keynote Lecture

Eckhard Pfluegel[1]

[1]School of Computer Science and Mathematics
Kingston University
London, UK

October 19, 2020

Outline

Case Study 1

According to omnicoreagency.com, there are currently more than 31 million YouTube channels. Every year, thousands of them are compromised or receive unsolicited, misleading or illegal comments. Rather than purely relying on the account owners to spot and report these incidents, YouTube is employing dedicated staff to periodically screen randomly selected channels. Unfortunately, this screening process is time-consuming, staff need to be appropriately trained and might also need counselling. Hence, the question arises whether a strategy could be devised in order to replace the random selection process by a more informed one, taking into account the attacker's motivation.

Case Study 2

Alan is in charge of vulnerability management for the IT systems of his company. The security policy of his organisation only allows for critical patching of one software per day, due to issues with bandwidth and system downtime during the patching process. He has identified two different vulnerabilities that require patching. He downloads their CVSS scores but finds that they are very similar, making it difficult to decide on a priority ranking. He would like to use the information contained in some CVSS subscores in such a way that he follows game-theoretic principles, taking into account both the attacker's capabilities and the impact on the system. This would give him a different view and could potentially lead to a clearer prioritisation.

Approach Using Game Theory

- We would like to use the mathematical concept of game theory in order to address the previous two case studies.

- The usefulness of this approach depends on:
  - The level of sophistication of the specific game model;
  - The required assumptions that need to be made for the model to be valid;
  - The acceptance of the game theoretic model by the real-world players.

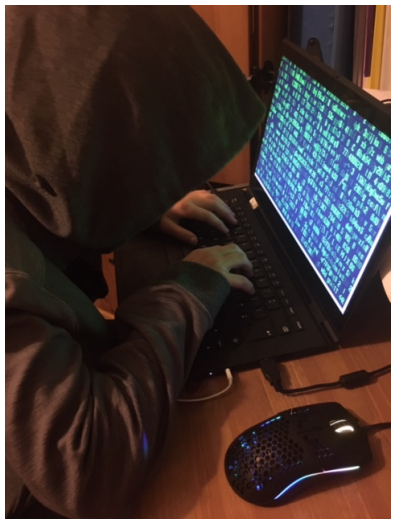
Outline

Concepts in Game Theory

- Game theory is "*the study of mathematical models of conflict and cooperation between intelligent rational decision-makers.*" [Wikipedia]
- Important concepts:
  - Game Type
  - Players
  - Strategies
  - Utilities (payoffs)
  - Solving a game
  - Nash Equilibrium

Security Games

- Security games model interaction between players:
  - attacker(s)
  - defender(s)



- They assume that players are *rational* and *non-cooperative*.
- All players seek to maximise their payoff (utility) functions.

- Our context: *two-player*, *bimatrix complete information* games.

Complete Information Security Games

- Complete information security games assume mutual knowledge of strategies, by both the attacker and defender.
- In real-world scenarios, this could correspond to a situation where the attacker might be able to gain some knowledge.
- For example, through:
  - insider information
  - information leakage
  - reconnaissance
- The defender might be aware of potential attackers and their motivation through security assessment and risk analysis.

Outline

Motivation

- Game-theoretic models that are motivated by real-world scenarios.
- Constraints in defense budget can be taken into account.
- Big security games are usually non-zero-sum games.
- Typically, they have 2 players but a large number of actions.
- A compact notation helps with efficiently storing and solving the game.

Notations

- Let us fix some notations:
  - $\mathcal{T} = \{t_1, \ldots, t_n\}$ – set of *targets* (assets under attack),
  - $\mathcal{R} = \{r_1, \ldots, r_m\}$ – set of *resources*, covering $m \leq n$ targets (implementing controls),
- Example: Case Study 1
  - $\{t_1, \ldots, t_n\}$ – targeted YouTube channels ($n = 31,000,000$),
  - $\{r_1, \ldots, r_m\}$ – YouTube employees ($m \ll n$).

Strategies – Attacker

- The attacker can choose between *pure* or *mixed* strategies.
  - Pure strategy space:
    - $s_A = \mathcal{T}$ – the set of targets,
  - Mixed strategy space:
    - $s_A = \{(q_1, q_2, \ldots q_n)\}$ where $0 \leq q_j \leq 1$ and $\sum q_j = 1$ represents the probability of attacking target t_i.
- Example: Case Study 1
  - The attacker can choose random YouTube channels as his targets, using an automated script. This implements a mixed strategy.

Strategies – Defender

- $\mathcal{S}$ – set of *feasible schedules*, representing specific allocations of resources to cover targets (respecting a finite budget constraint).
- There are $d = \binom{n}{m}$ such feasible schedules.
- Pure strategy:
  - This is a feasible schedule $s_D \in \mathcal{S}$, covering m out of n targets.
  - Notation: $s = \langle i_1, i_2, \ldots i_n \rangle$ $(i_j \in \{0, 1\}, \sum i_j = m)$.
- Mixed strategy:
  - $s_D = (p_1, p_2, \ldots p_d)$ where $0 \leq p_j \leq 1$ are probabilities of using a feasible schedule.
  - This induces a *coverage vector* $c = \langle c_1, \ldots, c_n \rangle$, expressing the probability of protection for each of the targets.

Utilities

- Compact notations: utility per use/attack of t_i
  - $u_D^c(t_i)$ – defender's utility when target t_i is covered by at least one resource,
  - $u_D^u(t_i)$ – defender's utility for uncovered target t_i,
  - $u_A^c(t_i)$ – attacker's utility for a covered target t_i,
  - $u_A^u(t_i)$ – attacker's utility, when t_i is uncovered.

Assumptions/Properties

- Applying resources to a target benefits the defender and hurts the attacker:
  - $\Delta u_D(t_i) := u_D^c(t_i) - u_D^u(t_i) > 0$ – the defender's utility reduction due to loss of coverage on attack,
  - $\Delta u_A(t_i) := u_A^u(t_i) - u_A^c(t_i) > 0$ – the attacker's utility gain when attacked target not covered.
- These assumptions are realistic, for the considered scenario.

Example: Case Study 1

- YouTube Game in sparse notation:

	t_1		t_2		t_3		t_4	
	c	u	c	u	c	u	c	u
D	7	3	5	1	4	3	3	0
A	0	1	1	4	0	5	2	3

- Using bimatrix game notation, we have:

$$A = \begin{pmatrix} 7 & 1 & 3 & 0 \\ 3 & 5 & 3 & 0 \\ 3 & 1 & 4 & 0 \\ 3 & 1 & 3 & 3 \end{pmatrix}, \; B = \begin{pmatrix} 0 & 4 & 5 & 3 \\ 1 & 1 & 5 & 3 \\ 1 & 4 & 0 & 3 \\ 1 & 4 & 5 & 2 \end{pmatrix}.$$

Case Study 1 (continued)

- The unique NE solution (x^*, y^*) to this game is: $x^* = (0, 0.39, 0.43, 0.17)$ and $y^* = (0, 0.16, 0.63, 0.21)$.
- The corresponding expected payoffs are 2.68 and 2.83.
- These are coverage vectors, from which suitable feasible schedules can be computed.

References

- Kiekintveld, C., Jain, M., Tsai, J., Pita, J., Ordez, F., and Tambe, M. (2009). Computing Optimal Randomized Resource Allocations for Massive Security Games. Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, 1, 689-696.
- Korzhyk, D., Yin, Z., Edu, Z., Kiekintveld, C., Conitzer, V., and Tambe, M. (2011). Stackelberg vs. Nash in Security Games: An Extended Investigation of Interchangeability, Equivalence, and Uniqueness. Journal of Artificial Intelligence Research, 41, 297-327.

Outline

Motivation

- Single-target game: we only consider one target. The focus is on the single asset that has a vulnerability.
- Most simple attacker defender scenario.
  - Rows corresponds to the strategies available to the defender.
  - Columns are the attacker's strategies.
- This game is suitable for modelling the vulnerability patching scenario (Case Study 2).

Payoff Notations

- $c^{\mathcal{D}}$ – the defense cost
- $l^{\mathcal{D}}$ – the defender's loss from an attack
- $c^{\mathcal{A}}$ – the attacker's cost
- $b^{\mathcal{A}} = l^{\mathcal{D}}$ – the benefit of the attacker.
- Assumptions:
  - *Principle of Adequate Protection*: $c^{\mathcal{D}} < l^{\mathcal{D}}$
  - *Principle of Easiest Attack*: $c^{\mathcal{A}} < l^{\mathcal{D}}$

Game Description

- Payoff Matrix:

$\mathcal{D} \downarrow \mathcal{A} \rightarrow$	s_a	s_{-a}
s_d	$-c^{\mathcal{D}}, -c^{\mathcal{A}}$	$-c^{\mathcal{D}}, 0$
s_{-d}	$-l^{\mathcal{D}}, b^{\mathcal{A}} - c^{\mathcal{A}}$	$0, 0$

- Strategies:
  - $S_{\mathcal{D}} = \{\text{patch, not patch}\} = \{s_d, s_{-d}\}$
  - $S_{\mathcal{A}} = \{\text{ attack, not attack }\} = \{s_a, s_{-a}\}$

Game Analysis

- **Theorem 1**. The security game $G(D, A)$ has no pure Nash Equilibrium strategy.
- **Proof**: By inspecting the game.
- **Theorem 2**. A mixed Nash Equilibrium strategy (s_D, s_A) is obtained, where $p = 1 - c^A/I^D$ and $q = c^D/I^D$ are the probability of defense and attack respectively. The resulting expected utilities, in this case, are $u_D = c^D$ and $u_A = 0$.
- **Proof**: Following Nash.

Example: Case Study 2

- The analysis of the previous game can be used for the vulnerability patching scenario.
- The goal is to find realistic values for the game payoff parameters I^D and c^A.
- This could be done for example using the *Common Vulnerability Scoring System* (CVSS).
- Information about the severity of the vulnerability is publicly available online.

Case Study 2 (continued)

- The attacker's cost c^A is proportional to the inverse of the CVSS *exploitability subscore*:
  - $c^A = \alpha \cdot {\mu_E}^{-1}$.
  - Here, α is a constant that needs to be suitably defined.
- The loss of the defender I^D is due to a threat event impact, exploiting the vulnerability and affecting the asset's CIA security requirements.
- Using the CVSS impact subscore, a vector V with numerical components is defined, depending on the security criticality of the asset.
  - This yields $I^D = \mu_{Imp,C} \cdot V_C + \mu_{Imp,I} \cdot V_I + \mu_{Imp,A} \cdot V_A$.
- Finally, by plugging this into the game solutions, a recommendation can be made for the decision to patch the vulnerability.

References

- Maghrabi, L., Pfluegel, E., Al-Fagih, L., Graf, R., Settanni, G., Skopik, F. (2017). Improved software vulnerability patching techniques using CVSS and game theory. In 2017 International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2017, Institute of Electrical and Electronics Engineering.

Outline

Summary

- In this talk, we have outlined some recent game-theoretical applications to security scenarios.
- We have explored the use of massive security games for optimal allocation of resources.
- We have also analysed a strategic security game in order to inform vulnerability patching with game theory.
- The usefulness of game theory depends on its acceptance amongst practitioners, including both defender and attacker.

Outlook

- Higher-dimensional matrix models for more realistic dilemma analysis and security scenario modelling need to be mastered:
  - Focus on order $n = 3$ first, then tackle higher orders.
  - Building on complete information games, devise incomplete information game models.
- The use of game-theoretic models in security standards would be an ultimate achievement:
  - Security management and assessment (OCTAVE, NIST RMF, ISO-27000),
  - Vulnerability scoring and assessment (CVSS).