Game Theory with Learning for Cybersecurity Monitoring

Keywhan Chung
advised by: Professor Ravishankar Iyer
Professor Zbigniew Kalbarczyk
In collaboration with Dr. Charles Kamhoua, Dr. Kevin Kwiat at AFRL

Dec. 3, 2014
Target Data Breach

- A Data Breach on Target customer data
- Attack Lasted for a month (Nov-Dec, 2013)
- ~110M customers impacted

“A ‘Kill Chain’ Analysis of the 2013 Target Data Breach”
Insights on Attacks

• Earlier Attacks:
  • Cause damage to the system by getting in and out as quickly as possible
  • E.g., DoS, Ransomware, etc.

• Recent Attacks:
  • Attacks are getting sophisticated and hard to detect
  • Goal is not only to obtain access, but also maintain the foothold without discovery
  • Consists of multiple stages that are hard to differentiate from legitimate operations
  • E.g. Target Data Breach, STUXNET, etc.
What is The Problem?

• Lack of integrated/automated detection methods
  • Increased alarms including FPs
  • Highly dependent on human intelligence for identifying the attack

• Shortage in security specialists!

Need for an automated decision making process that can be applied to cybersecurity monitoring
Modeling the Attack

- Attack as a decision process
- Consists of attack states
- Attacker chooses among available actions

**AS0 - Benign State:**
Attacker is no different to legitimate user

**AS1 - Network/FS Access:**
Attacker has access to network and file system (FS)

**AS2 - Access to POS**
Attacker has access to raw data from POS

- Detected by defender
- No attack
- Data breach from FS
- Access network with stolen credentials
- Infect POS system
- Data breach from FS
- Breach fin. Info
Our Approach

• Interest of attacker and defender(players) conflicts w/ each other
  • Attacker wants to intrude into the system + breach data
  • Defender wants to protect the system and data
• Players making rational decisions
A Game Theoretic model

• Markov Game (Stochastic Game) for repeated games on a MDP
  • Assumes rational player with ‘Complete Information’
  • Derives optimal policy for maximum gain (reward)

• Complete Information
  Players have all information of its own and the opponent

• Rationality
  Players play to maximize their gain assuming a rational opponent

• Full Rationality & Complete Information are not realistic
Guess 2/3 of the Average

• Lets say we have a competition
• Everyone in the room chooses a real number between 0 and 100
• Player who chooses the number closest to 2/3 average wins the game
• Your guess?

A game theoretic approach
- You can easily assume that any number above 66.67 is unlikely to win
- Others would also think in this manner
- Better to choose from \([0, 66.67 \times \frac{2}{3}]\)
- Again others will think in this manner so repeat!
- Resulting to 0 (theoretical)

An experimental result
- Competition over 19,196 people
- Winning value 21.6

- http://twothirdsofaverage.creativitygames.net
Security Games

• In Security Games
  • **Incomplete information** about the attacker and his/her strategy
  • What attacks can the attacker perform?
  • What is the reward function of the attacker?
  • **Players learn** about the opponent
    • Attackers probe the system to exploit vulnerabilities
    • Attackers dynamically optimize their attack

• Learning has been a common method for security problems (IDS etc.)
  • Learn from history
  • Limited rationality and knowledge of the system

Q-Learning for Security Games : rationality + learning
# Markov Game v/s Q-Learning

<table>
<thead>
<tr>
<th>Markov Game</th>
<th>Q-Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Quality of state</strong></td>
<td>expected reward by taking action pair at state and then following the optimal policy</td>
</tr>
<tr>
<td><strong>Value of state</strong></td>
<td>expected reward when following optimal policy from the state</td>
</tr>
<tr>
<td><strong>Discount factor ($\gamma$)</strong></td>
<td>player’s intention on weighting between future and current rewards</td>
</tr>
<tr>
<td><strong>Learning rate ($\alpha$)</strong></td>
<td>player’s intention on weighting between learning and rationality</td>
</tr>
<tr>
<td><strong>Exploration Rate ($\exp$)</strong></td>
<td>degree of variation from the optimal policy (for learning through trial and error)</td>
</tr>
</tbody>
</table>
**Markov Game vs Q-Learning**

**Markov Game**

\[ Q^{t+1}(s, a, o) = R(s, a, o) + \gamma V^t(s') \]

Immediate reward \hspace{1cm} Estimate of optimal future value

\[ \gamma \rightarrow 1: \text{player strives for long-term high reward} \]

**Q-Learning**

**Minimax (MMQL)**

\[ Q^{t+1}(s, a, o) = (1 - \alpha)Q^t(s, a, o) + \alpha(R(s, a, o) + \gamma V^t(s')) \]

**Naïve (NQL):** decision making with no information on opponent

\[ Q^{t+1}(s, a) = (1 - \alpha)Q^{t+1}(s, a) + \alpha(R(s, a, o) + \gamma V^t(s')) \]

\[ \alpha \rightarrow 1: \text{less learning} \]
Experiment: Overall Structure

Initialize $V, Q, \alpha = 1.0$

Q-Learning

Choose action
if random > exp:
take random action
else:
follow optimal policy

Update Q

Update V, $\pi[s]$

Decay alpha to learning rate

Update Next State

Loop while diff$(V) > \epsilon_{\text{precision}}$

Q-Learning Strategy
Flexible

Markov Strategy
Fixed

Attacker
Random
NQL
MMQL
Markov

Defender
NQL
MMQL
Markov

Random attackers to represent attackers with low information/rationality e.g., script kiddies
Results: Accumulated Reward of Attacker

- Markov Defender >> Naïve Q-Learning Defender > Minmax Q-Learning defender
Results: Accumulated Reward of Attacker

- Random Attacker: NQL > Markov > MMQL
- Markov Attacker: Markov >> MMQL ≈ NQL
Results: Accumulated Reward of Attacker

- NQL relatively weak against NQL opponent
- Exploration(exp) rate has different meaning to attacker and to defender
Results: Imm. Reward of the MMQL Attacker

- Markov Defender (121)
- MMQL Defender (290)
- NQL Defender (-229)

More aggressive attacker leads to more detection.
Results: Impact of Learning Rates

- Learning rate had no significant impact on accumulated reward

- Low $\alpha$ (intensive learning) likely to accelerate convergence

- $V(s)$ bigger when $\alpha_{\text{attacker}} \leq \alpha_{\text{defender}}$

- Better to keep $\alpha$ low

Note. Low $\alpha$ indicates intensive learning
Adding reality to the model

• **Timeliness** missing in the model!
  
  • $Q^{t+1} = Q^t + \cdots$
    
    • **Till when** should the player make the decision?
    
    • Can the model make the right decisions **in a timely manner**?
  
  • $Q^{t+\Delta t} = Q^t + \cdots$
    
    where $\Delta t$ comes from the attack data at NCSA
  
  • Attacker event **modeled from the data**

• A more realistic **reward model**
  
  • $R(s, a, o) = N$
    
    • Reward cannot always be static.
      E.g., Data Breach: longer the remaining, more the reward
  
  • $R(s, a, o, t) = I(s, a, o) \times (t - t_s)$
    
    where $I(s, a, o)$ is the unit reward (previously $R(s, a, o)$) and $(t - t_s)$ is the time interval spent at the state
Adding reality to the model

- **New Event?**
  - **Event Prob. > θ**
    - **Update State Policy**
    - **T < Δt?**
      - **Transit to Next State**
  - **Filter False Positives**
  - **High Prob.**: Likely to be a real attack, or an event/attack with high expected reward
  - **Low Prob.**: An event that rarely occurred in the past & has low expected reward

- **Q-Learning**
  - **Choose action**
    - if random > exp:
      - take random action
    - else:
      - follow optimal policy

- **Update Q**
- **Update V, π[s]**
- **Update Next State**

- **Initialize V, Q, alpha=1.0**

- **Game State**
  - **Monitoring**
  - **Decision Making**
  - **Timeliness for decision making**
Applications & Limitations

• Can be used to derive the risk on alarms
  • Expert knowledge to build attack graph
  • QL method assigns priorities on alarms with high risk
  • Differentiate True alarms from False alarms
  • Predict the next alarm or attack state, given a set of alarm from a combination of expert knowledge + history

• Limitations
  • Not applicable for detecting new attacks, more about decision making
  • Dependent on sensor performance that generates alarms and graphical representation of the attack
Conclusion

• Need for an **automated decision making process**
• **Q-Learning to emulate rationality + learning**
  • Generally, not as good as Markov Game
  • Markov Game not applicable for **incomplete information**
• **Naïve Q-Learning:** tempting solution given **limited information**
  • Outperforms Markov Game against **less rational players (random, MMQL)**
  • NQL defender leads MMQL attacker to mal-perform (through interaction)
  • Study on effects of parameters

• Future Work
  • A more realistic reward model?
  • What is a real attacker’s decision process like?
  • How to tune the parameter after the opponent?