Game Theory with Learning for Cybersecurity Monitoring

Keywhan Chung

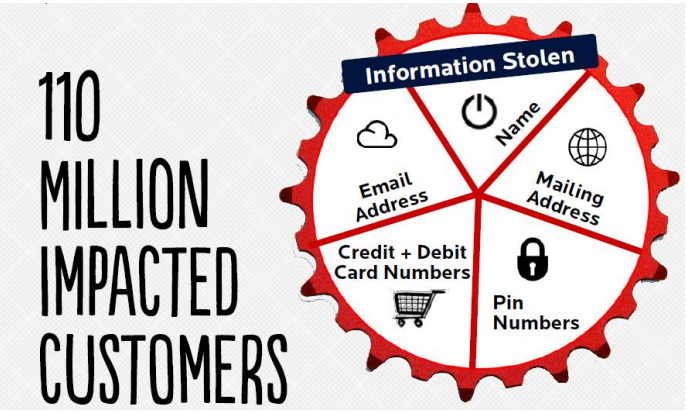
advised by : Professor Ravishankar Iyer
Professor Zbigniew Kalbarczyk

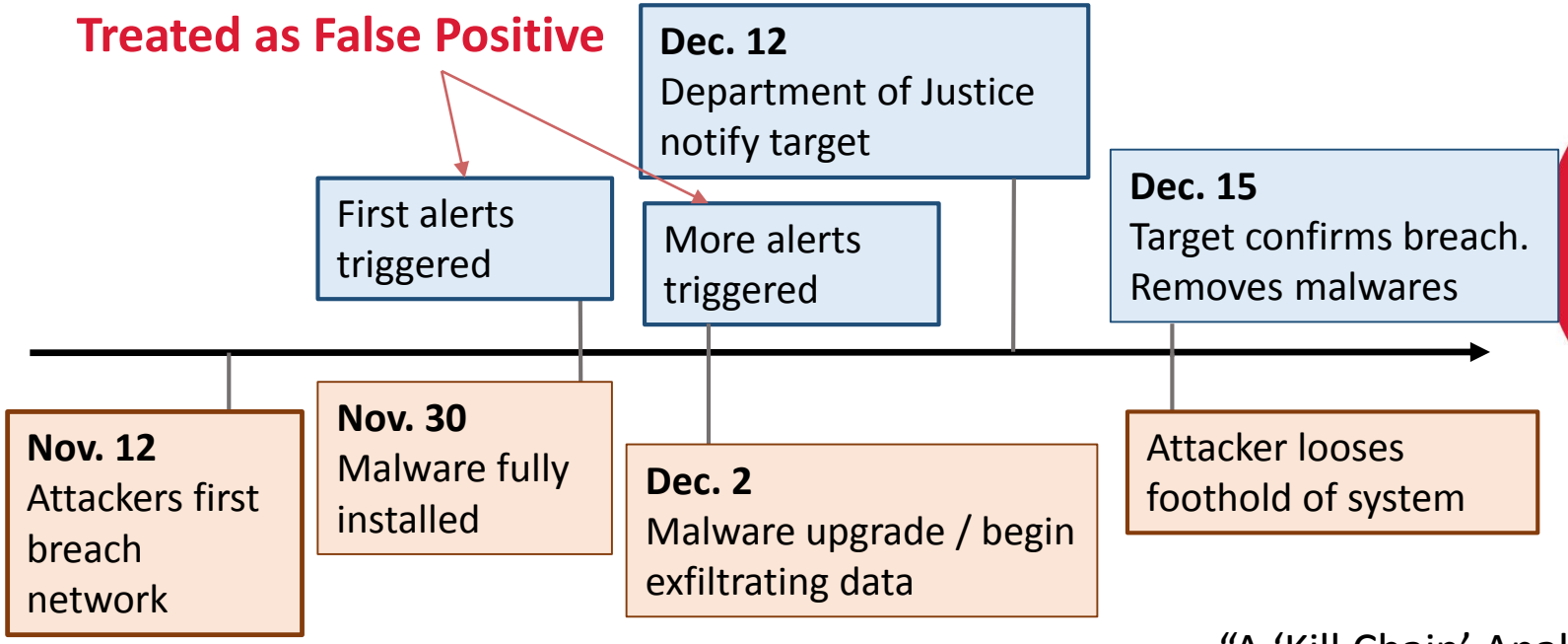
In collaboration with Dr. Charles Kamhoua, Dr. Kevin Kwiat at AFRL

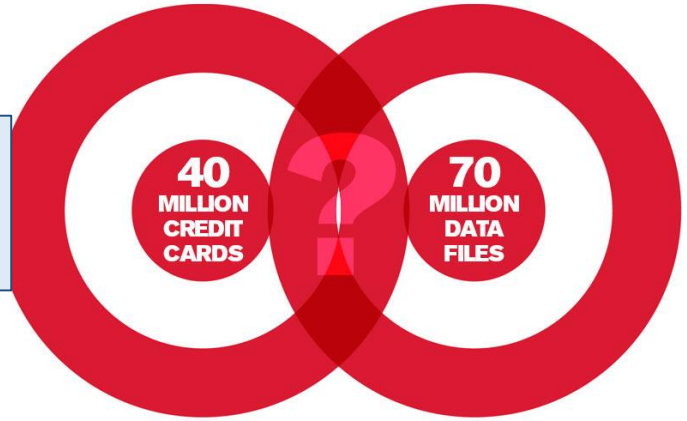
Dec. 3, 2014

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?

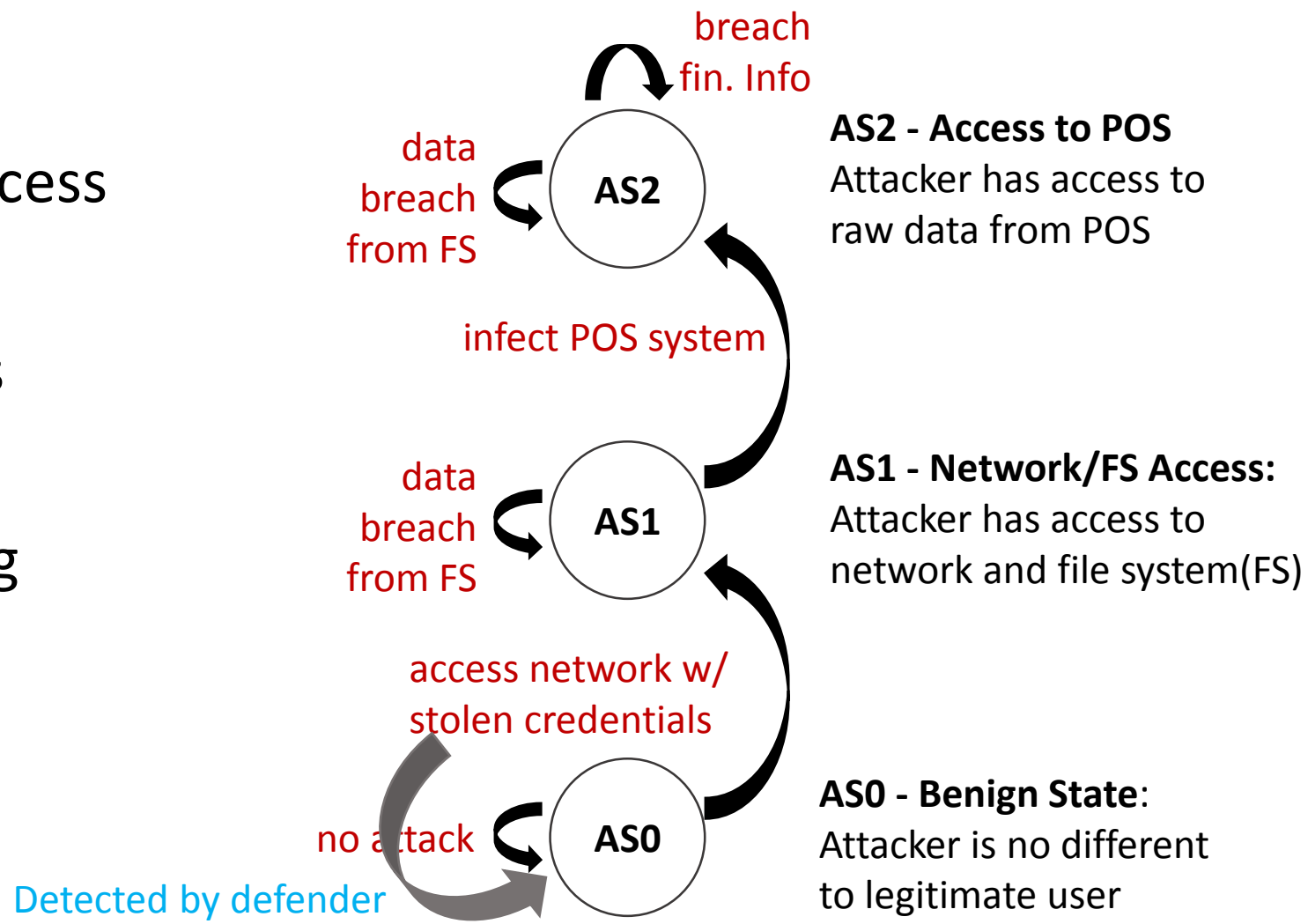
- Lack of integrated/automated detection methods
  - **Increased alarms <u>including FPs</u>**
  - Highly dependent on **human intelligence** for identifying the attack

- 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



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 Approach

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

Markov Game	**Quality of state**	expected reward by taking action pair at state and then following the optimal policy
	Value of state	expected reward when following optimal policy from the state
	Discount factor(γ)	player's intention on weighting between future and current rewards
Q - Learning	**Learning rate(α)**	player's intention on weighting between learning and rationality
	Exploration. Rate (exp)	degree of variation from the optimal policy (for learning through trial and error)

Markov Game vs Q-Learning

Markov Game

$$Q^{t+1}(s,a,o) = \underbrace{R(s,a,o)}_{\text{Immediate reward}} + \gamma \underbrace{V^t(s')}_{\text{Estimate of optimal future value}}$$

$\boldsymbol{\gamma} \rightarrow \mathbf{1}$: player strives for long-term high reward

Q - Learning

Minimax (MMQL)

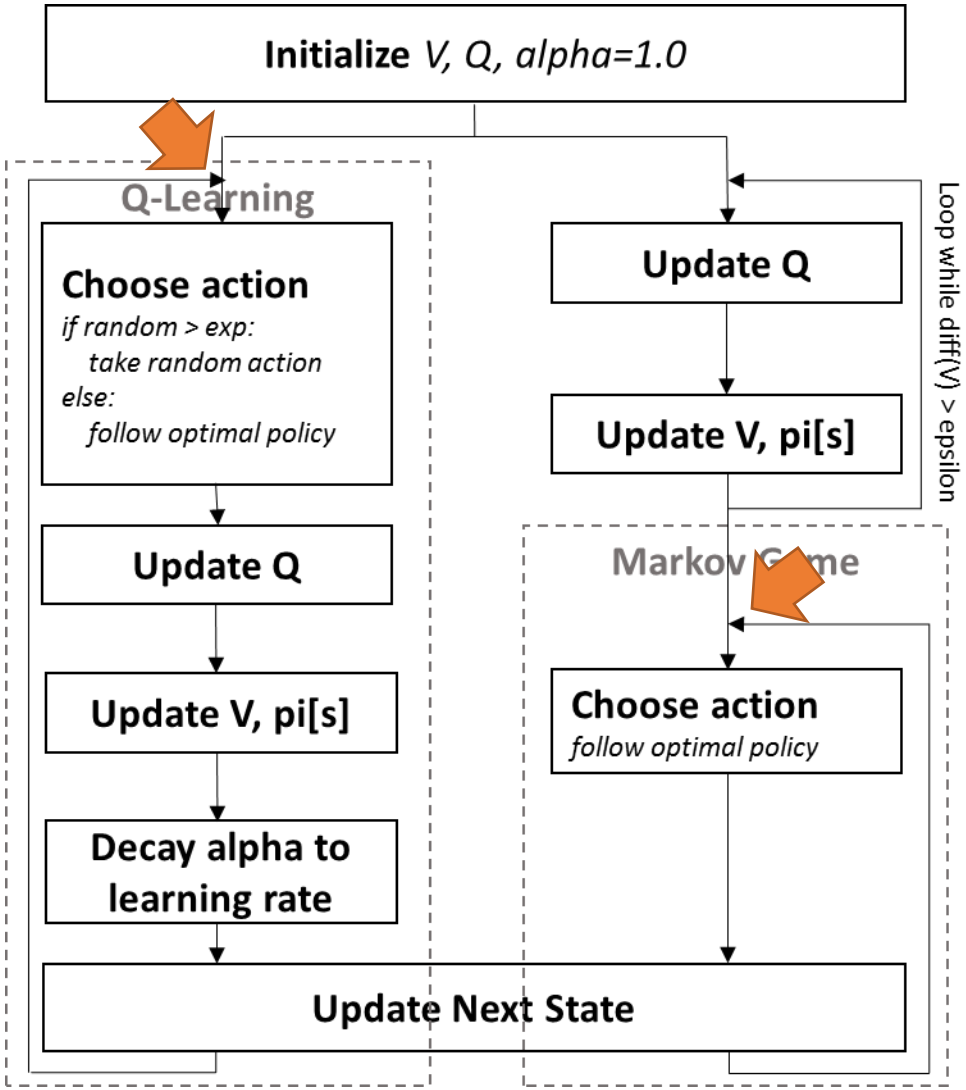
$$Q^{t+1}(s,a,o) = (\mathbf{1} - \boldsymbol{\alpha})\boldsymbol{Q^t}(\mathbf{s},\mathbf{a},\boldsymbol{o}) + \boldsymbol{\alpha}\big(R(s,a,o) + \gamma V^t(s')\big)$$

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

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

$\boldsymbol{\alpha} \rightarrow 1$: less learning

Experiment: Overall Structure

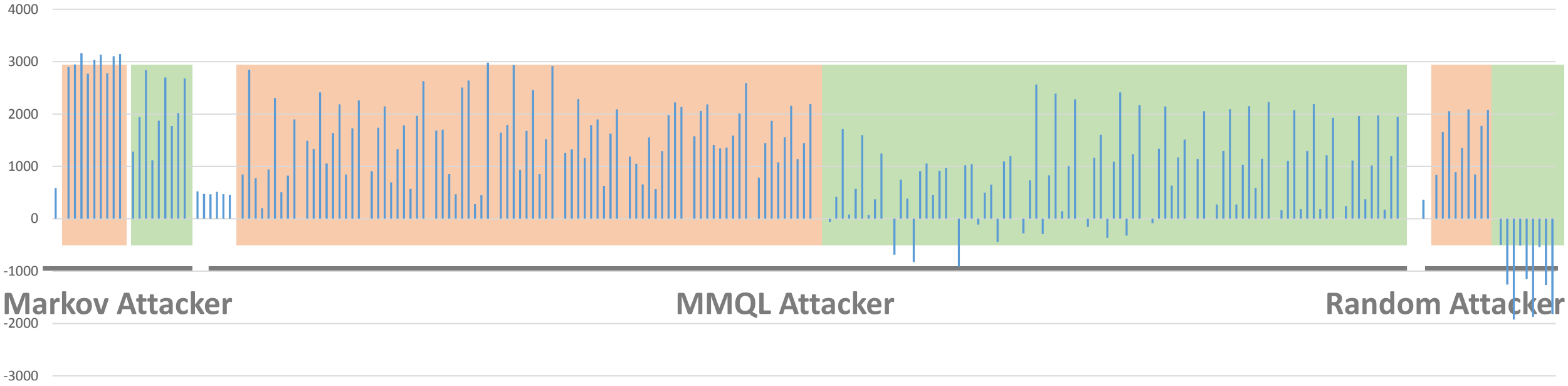


Q-Learning Strategy	Markov Strategy
Flexible	Fixed

Attacker		Defender
Random NQL MMQL Markov	⊗	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

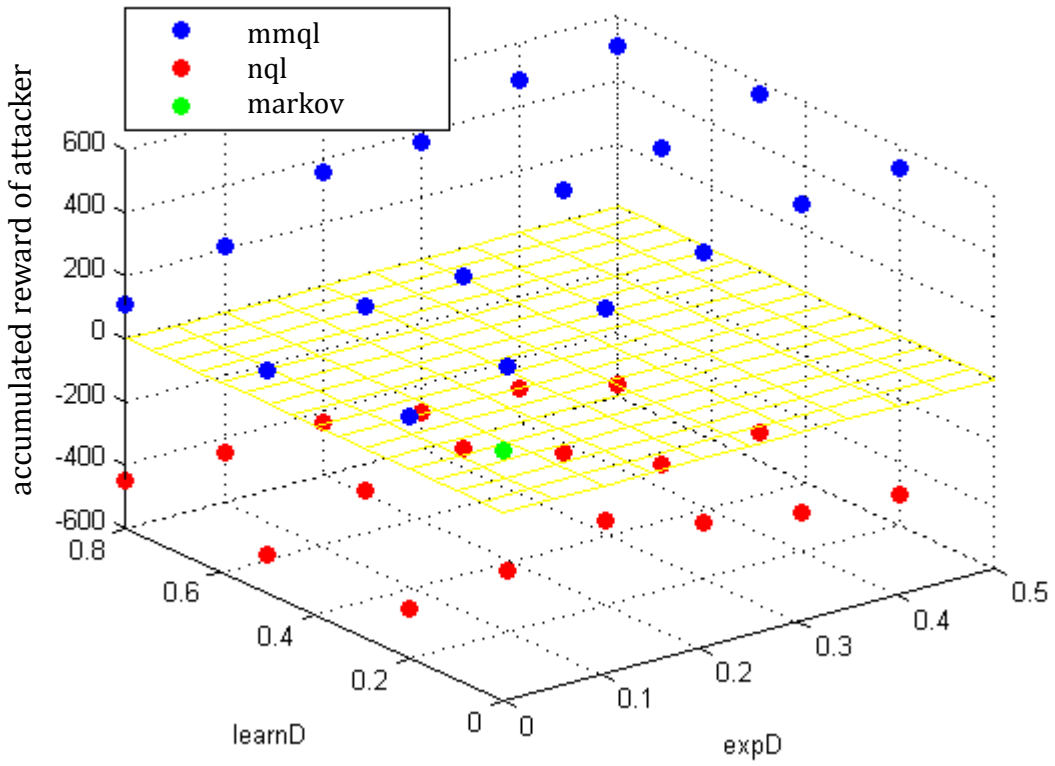
Markov Defender

MMQL Defender

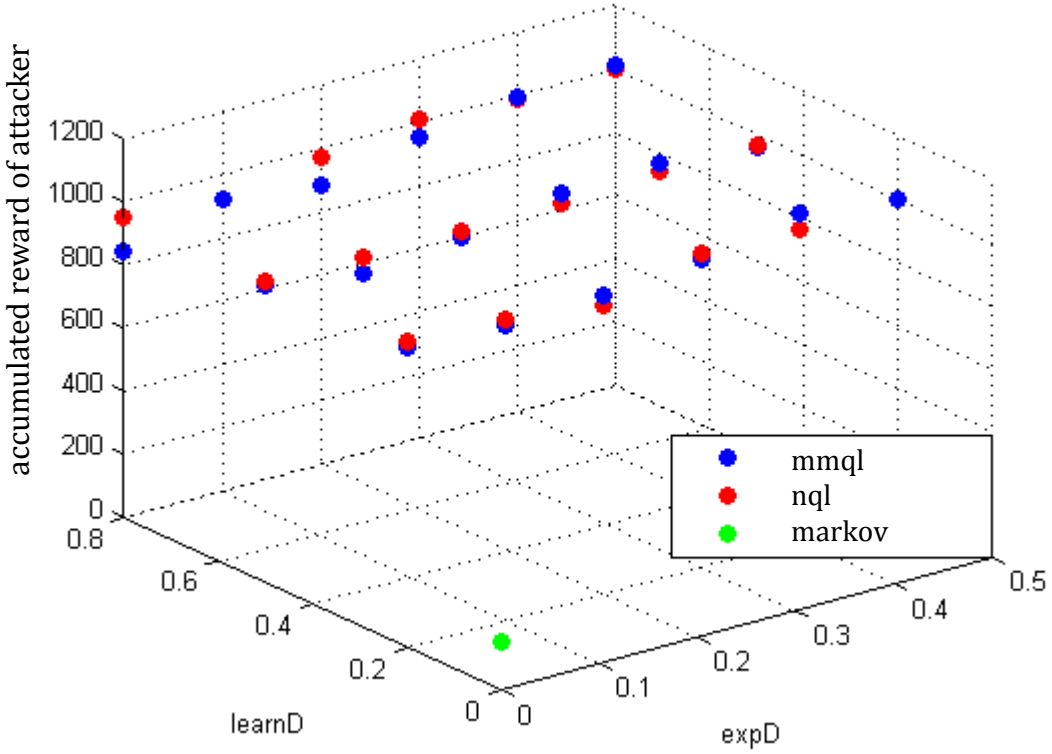
NQL Defender

Results: Accumulated Reward of Attacker

Random Attacker



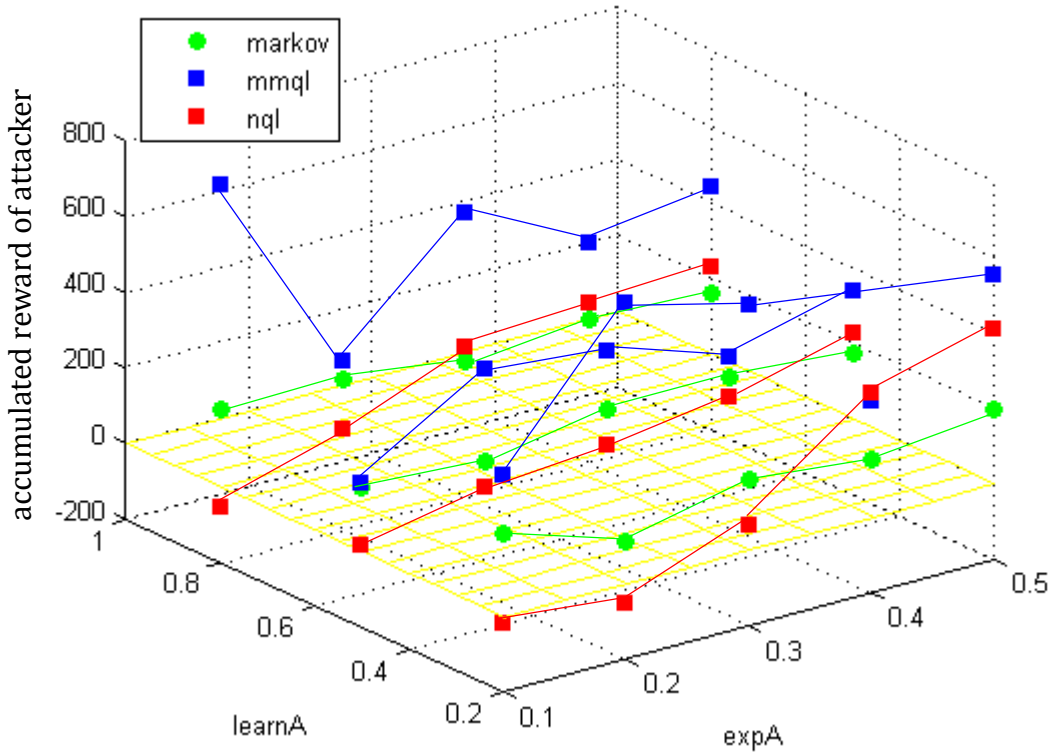
Markov Attacker



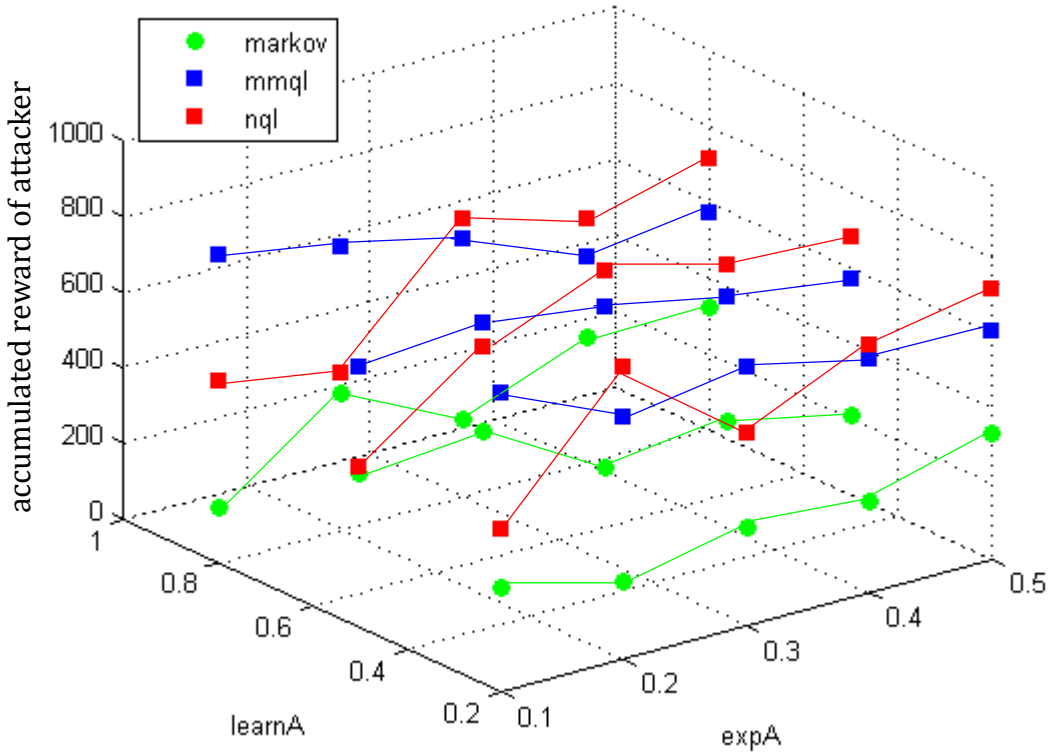
- Random Attacker: NQL > Markov > MMQL
- Markov Attacker: Markov >> MMQL ≃ NQL

Results: Accumulated Reward of Attacker

MMQL Attacker

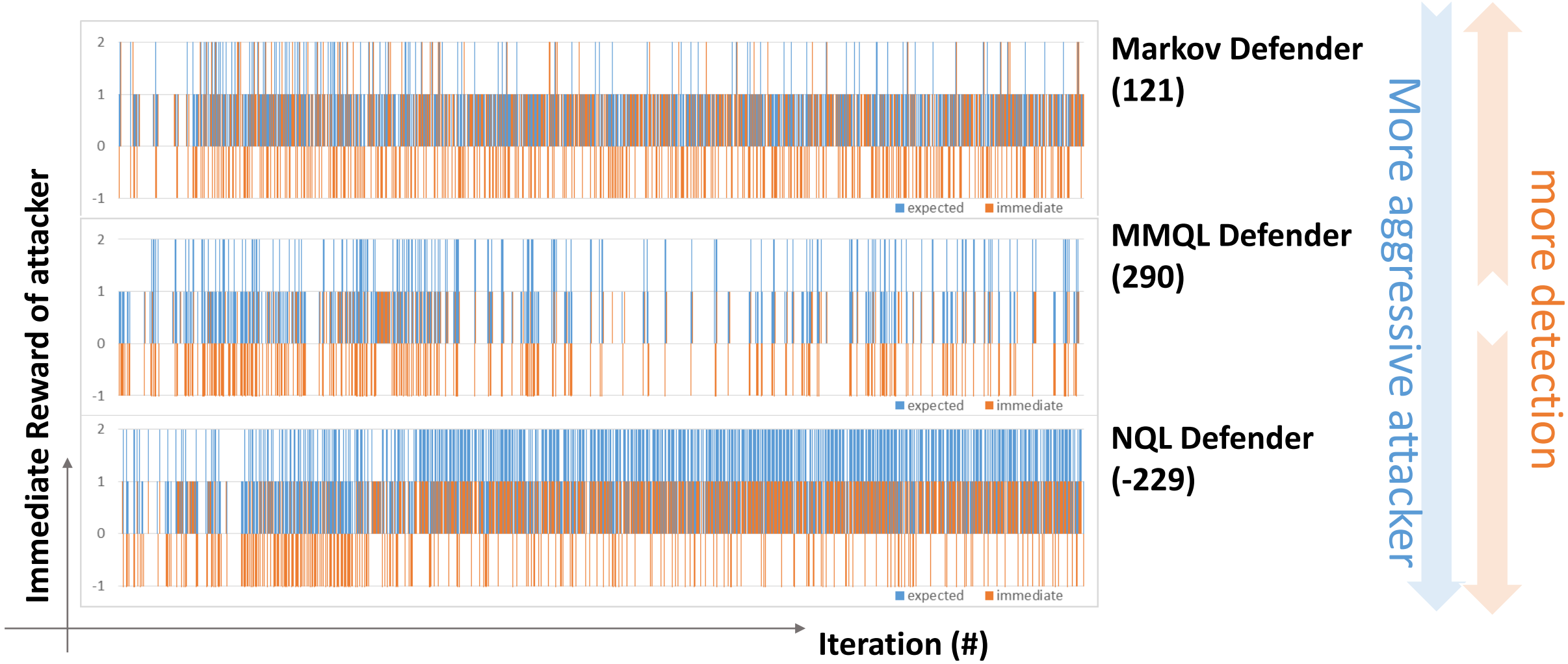


NQL Attacker



- NQL relatively weak against NQL opponent
- Exploration(exp) rate has different meaning to attacker and to defender

Results: Imm. Reward of the MMQL Attacker

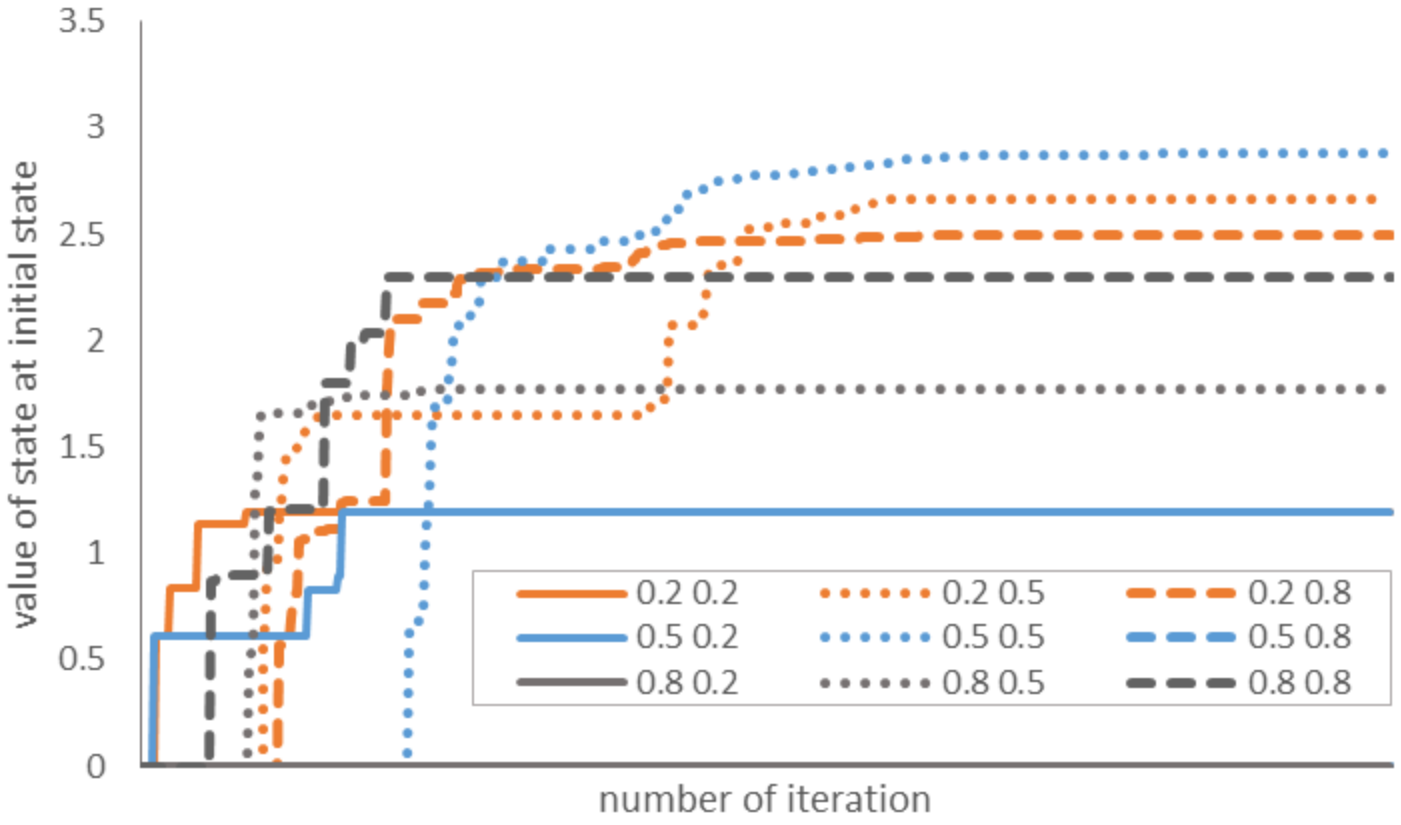


Results: Impact of Learning Rates

- Learning rate had no significant impact on accumulated reward

- Low α **(intensive learning)** likely to accelerate convergence
- V(s) bigger when $\alpha_{\text{attacker}} \leq \alpha_{\text{defender}}$

- **Better to keep α low**

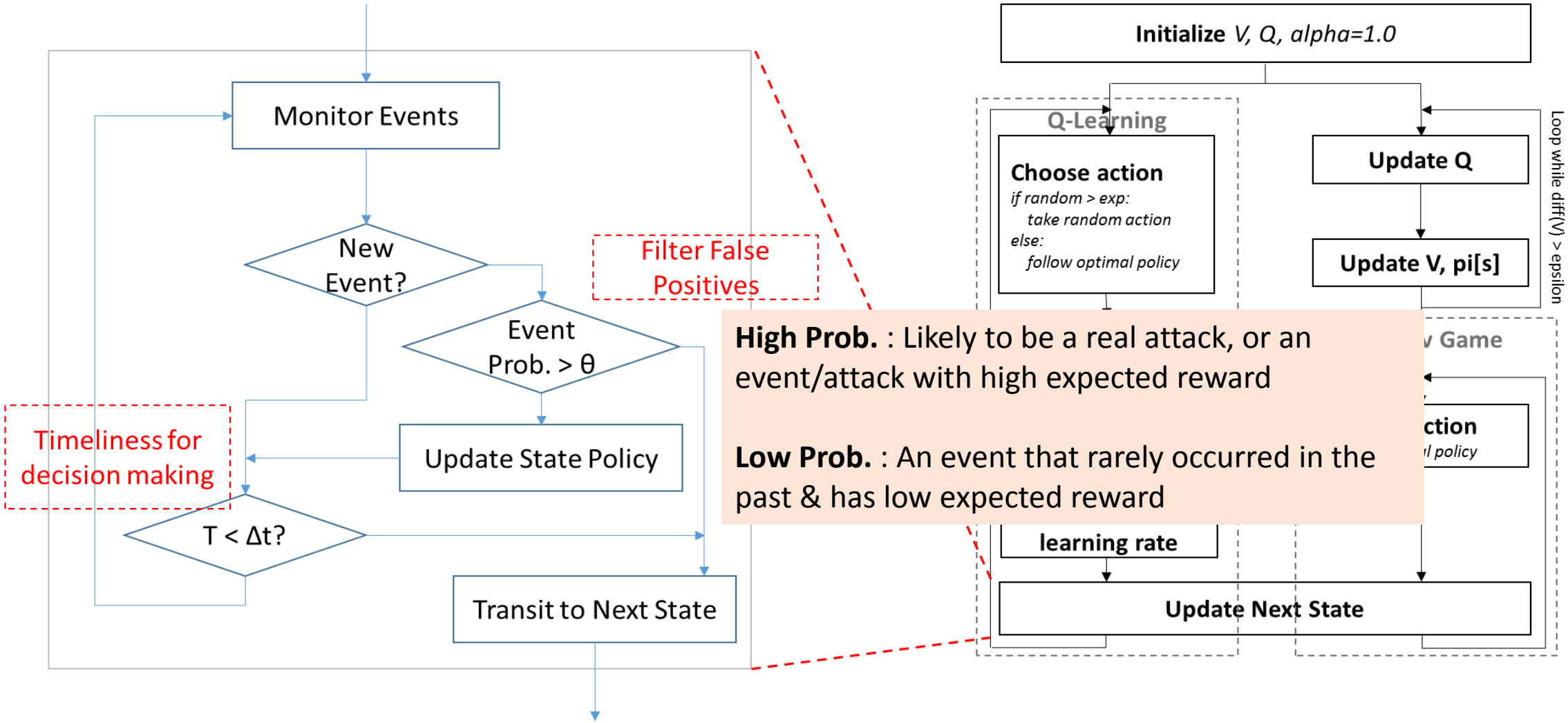


Note. Low α indicates intensive learning

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**?
  - $Q^{t+\boldsymbol{\Delta t}} = Q^{t} + \cdots$
    where Δt 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
  - $R(s, a, o, \boldsymbol{t}) = I(s, a, o) \times (t - t_s)$
    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



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

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?