Cyber Security: A game-theoretic analysis of defender and attacker strategies in defacing-website games

Palvi Aggarwal[1] Antra Grover[1] Saumya Singh[1] Zahid Maqbool[1] V.S.C Pammi[2] Varun Dutt[1]

[1]Applied Cognitive Science Laboratory

Indian Institute of Technology, Mandi, India – 175005

[2]Centre of Behavioural and Cognitive Sciences

University of Allahabad, Allahabad – 211002

Motivation

- Cyber-attacks, i.e., the disruption of normal functioning of computers and loss of private information through malicious network events, are becoming widespread (Johnson, 2011).
- In the absence of real defenders and attackers for laboratory studies, one solution is to use computational cognitive modeling.
- Both the attacker's and defender's cognitive limitations seem to influence the defender's timely and accurate detection of cyber-attacks (Dutt, Ahn, & Gonzalez, 2013).
- To understand the cognitive aspect of defender and hackers, cyber security may be studied as a non-cooperative dynamic games (Dutt, Ahn, & Gonzalez, 2013; Arora & Dutt, 2013; Kaur & Dutt, 2013; Gonzalez, 2011). We formulated a Defacing Website Game to do cognitive analysis.

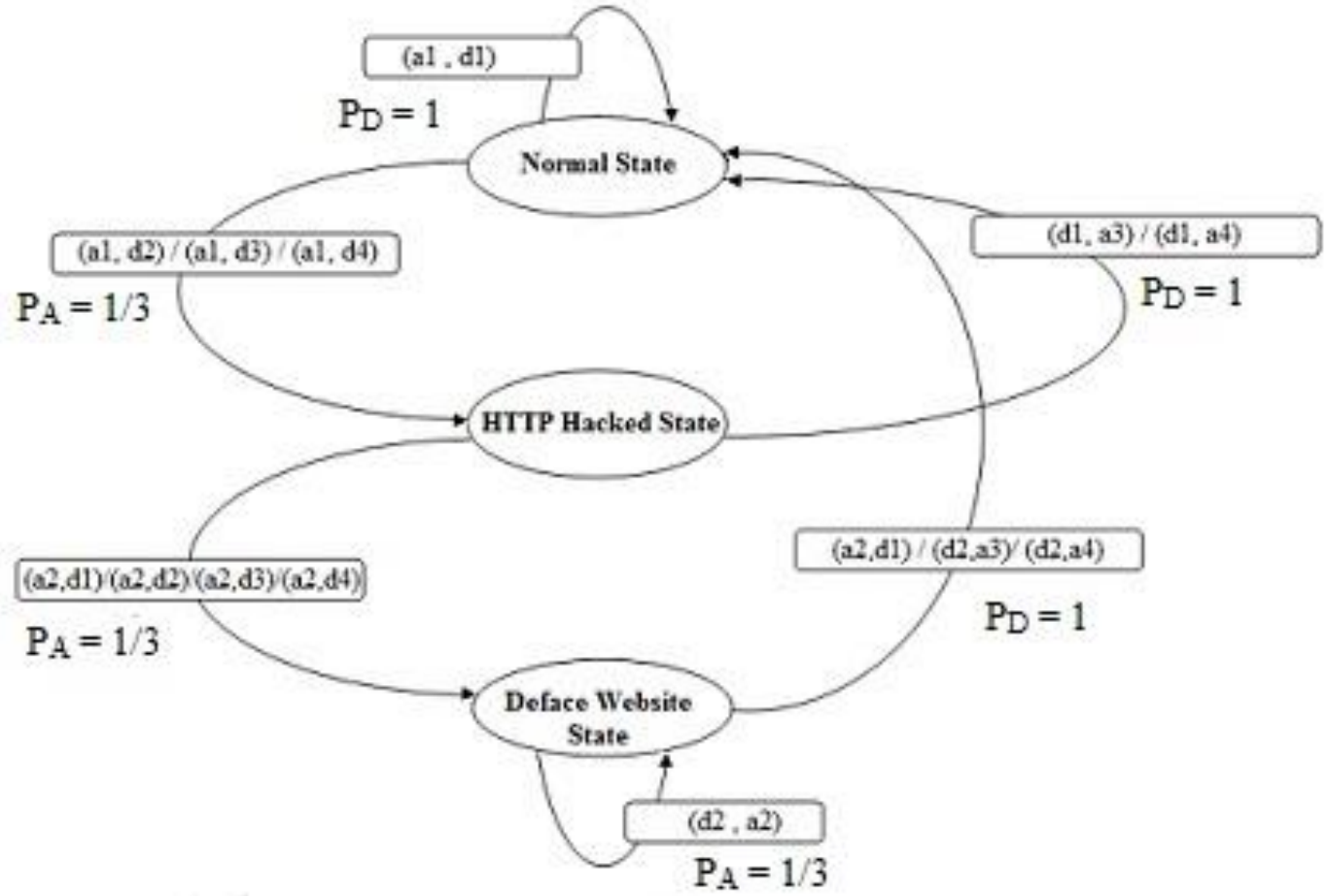
What is a Game?

- A game consists of
  - a set of players
  - a set of strategies for each player
  - the payoffs to each player for every possible list of strategy choices by the players.

- A game with just two players is a two-player game.

- We will study only games in which there are two players, each of whom can choose between only two strategies.

Attack Scenario for Defacing Website Game



Actions of Attacker

a1: Attack_Httpd

a2: Deface_Website

a3: No_Move

a4: Any_Other_Move

Actions of Defender

d1: Defend_Httpd

d2: Restore_Website

d3: No_Move

d4: Any_Other_Move

Here, P_A represent the probability of success of Attacker and P_D probability of success of Defender.

Cost Matrix: Normal Network State

NORMAL STATE COST MATRIX (scoreD, scoreA)				
Attacker → Defender ↓	AttackHttpd	DefaceWebsite	NoMove	AnyMove
DefendHttpd	(+5,-5): success (-1,-1): failure	(-1,-1)	(-1,0)	(-1,-1)
RestoreWebsite	(-1,+5): success (-1,-1): failure	(-1,-1)	(-1,0)	(-1,-1)
NoMove	(-5,+5): success (0,-1): failure	(0,-1)	(0,0)	(-1,-1)
AnyMove	(-1,+5): success (-1,-1): failure	(-1,-1)	(-1,0)	(-1,-1)

Attack matrix shows (scoreD, scoreA)

scoreD : score of Defender
scoreA: score of Attacker

As highlighted in table, When attacker chooses action AttackHttpd and defender chooses DefendeHttpd in normal network state, Score +5 shows reward for defender and -5 shows penality for attacker.

Cost Matrix: Http Hacked Network State

HTTPD HACKED STATE COST MATRIX(scoreD, scoreA)				
Attacker → Defender ↓	AttackHttpd	DefaceWebsite	NoMove	AnyMove
DefendHttpd	(-2,+5)	(+4,+6): success (+4,-1): failure	(+4,0)	(-2,-1)
RestoreWebsite	(-2,-1)	(-2,+6): success (-2,-1): failure	(-2,0)	(-2,-1)
NoMove	(-2,-1)	(-2,+6) : success (-2,-1): failure	(-2,0)	(-2,-1)
AnyMove	(-2,-1)	(-2,+6) : success (-2,-1): failure	(-2,0)	(-2,-1)

Attack matrix shows (scoreD, scoreA)

scoreD : score of Defender
scoreA: score of Attacker

As highlighted in table, When attacker chooses action DefaceWebsite and defender chooses RestoreWebsite in Http Hacked State, scor -2 shows penalizing action for defender and +6 shows rewarding action for attacker.

Cost Matrix: Deface Website Network State

DEFACE WEBSITE STATE COST MATRIX(scoreD, scoreA)				
Attacker → / Defender ↓	AttackHttpd	DefaceWebsite	NoMove	AnyMove
DefendHttpd	(-3,-1)	(-3,-1)	(-3,0)	(-3,-1)
RestoreWebsite	(+3,-1)	(-3,+4): success (+3,-1): failure	(+3,0)	(+3,-1)
NoMove	(-3,-1)	(-3,-1)	(-3,0)	(-3,-1)
AnyMove	(-3,-1)	(-3,-1)	(-3,0)	(-3,-1)

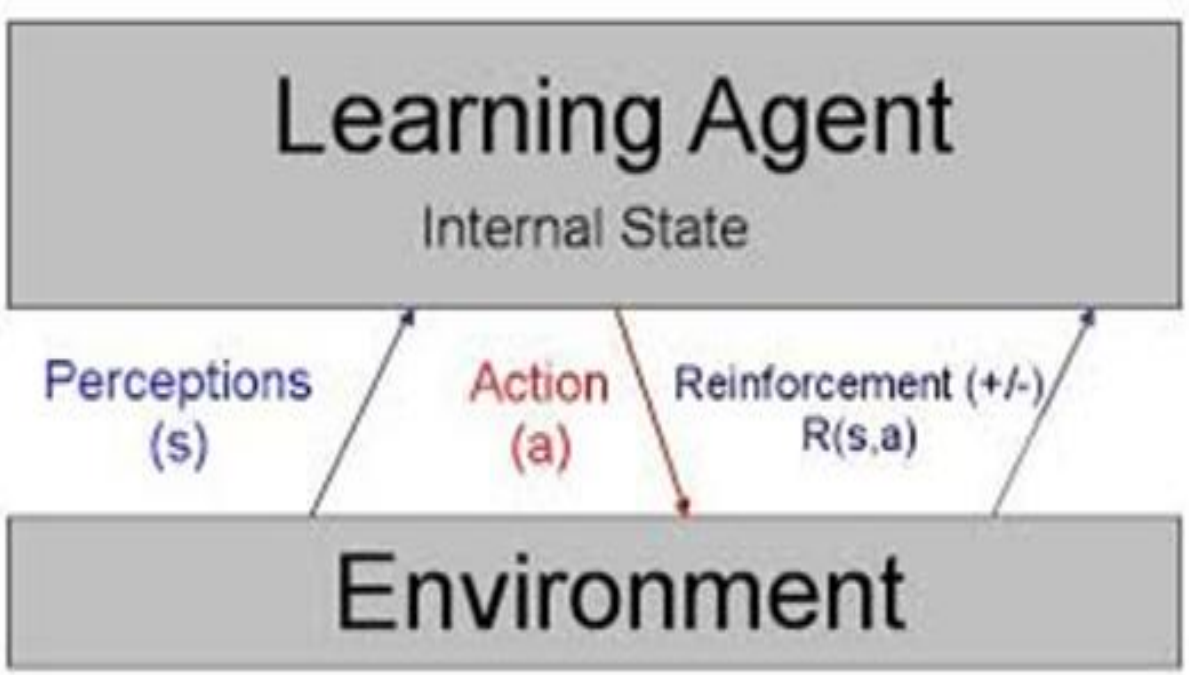
Attack matrix shows (scoreD, scoreA)

scoreD : score of Defender
scoreA: score of Attacker

When attacker chooses action DefaceWebsite and defender chooses RestoreWebsite in Deface Website State, score -3 shows penalizing action for defender and +4 score shows rewarding action for attacker.

The penalty for the attacker remains -1 across all the matrices; while, the penalty for the defender keeps on increasing from -1 to -3.

REINFORCEMENT-LEARNING (RL) MODEL



REINFORCEMENT-LEARNING (RL) MODEL

- Initially, the players have an equal expectation to choose any action

- This measure keeps getting updated as the players get rewards/penalties based on the action-pair and the final network state as per the following rule:

$$\textbf{Expect}_j\textbf{(t+1)} = \textbf{Expect}_j\textbf{(t)} * \textbf{(1-w)} + \textbf{O}_j\textbf{(t)} * \textbf{(w)}, \text{ if option j was selected in trial t.}$$
$$= \textbf{Expect}_j\textbf{(t)}, \text{ otherwise.}$$

Where,

$Expect_j$ (t+1) = updated expectation value of the action j chosen

$Expect_j$ (t) = previous expectation value of the action j chosen

O_j(t) = reward/penalty factor obtained on choosing action j

w = weight parameter ranging from [0, 1]

Method

We chose a Reinforcement Learning (RL) model to represent a simulated attacker and a defender in a 2x4 cyber-security game, where each of the 2 players could take up to 4 actions.

Each of the scenario is characterized by one attacker-defender pair playing against each other; the initial state of network in each case being Normal State.

Each of the players in a particular simulation were allowed to make 30 moves each.

We manipulate the value of weight parameter ‘w’ for both attacker and defender. We would be using two weight parameters corresponding to each player i.e. wA(attacker) and wD(defender).

Method

A high value of ‘w’ implies higher attention to the immediate costs of the actions, which further implies higher exploration and tendency to try different moves.

High w = 0.9

A low value of ‘w’ implies lesser significance of the learnings from the cost values and more reliance on previous experiences.

Low w = 0.1

We averaged our analysis over 1000 attacker-defender pairs which is a reasonable number to cover almost all network attack-defend strategies human mind can come up with.

RESULTS

Percentage of simulations for higher scores

Score Type	wD high	wD low
Higher score for defender	67%	64%
Score tie	02%	02%

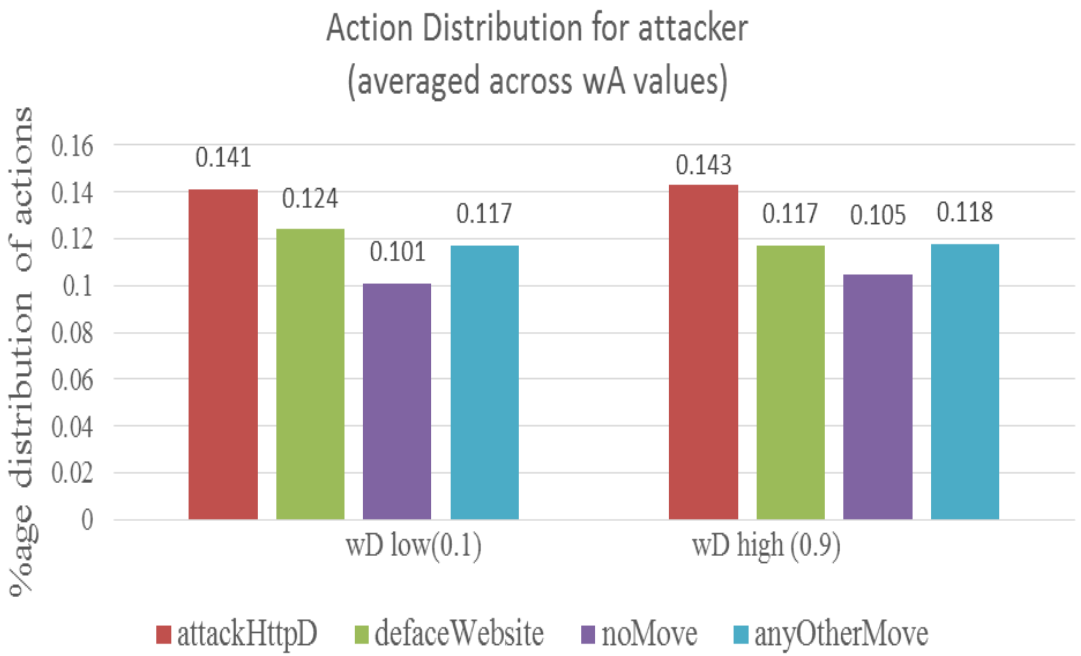
Score Type	wA high	wA low
Higher score for attacker	30%	35%
Score tie	03%	01%

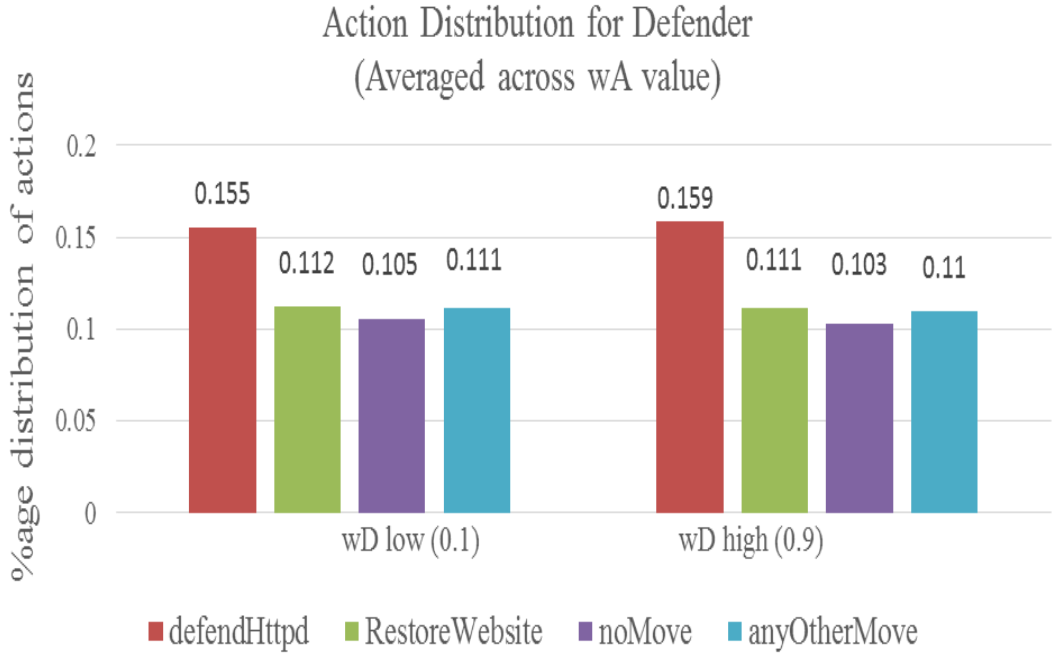
Percentage wins for the Attacker

wA	Percentage wins
High	0.15
Low	0.00

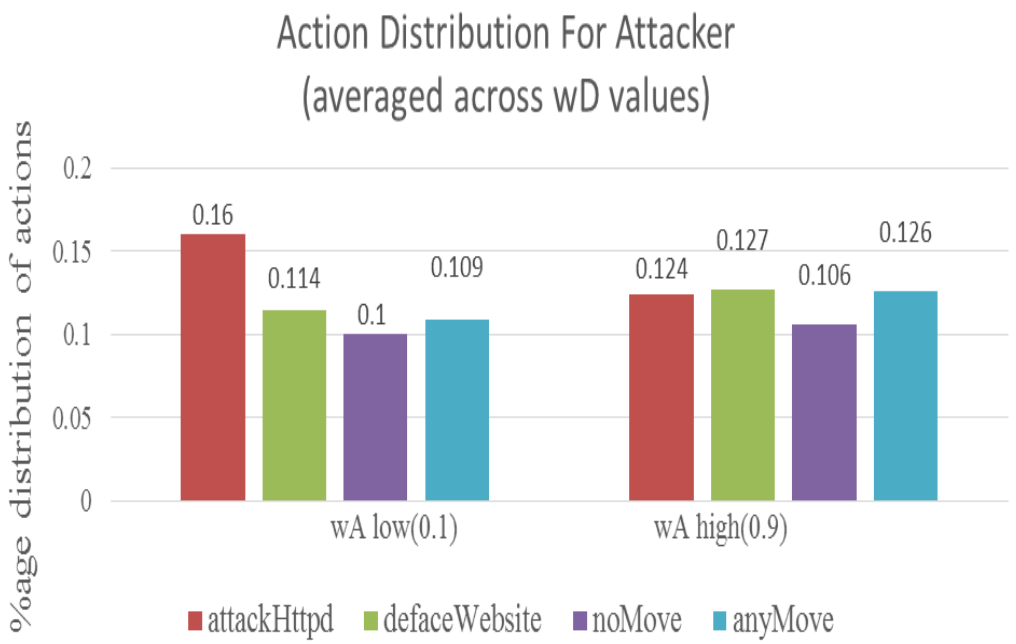
wD	Percentage wins
High	0.05
Low	0.1

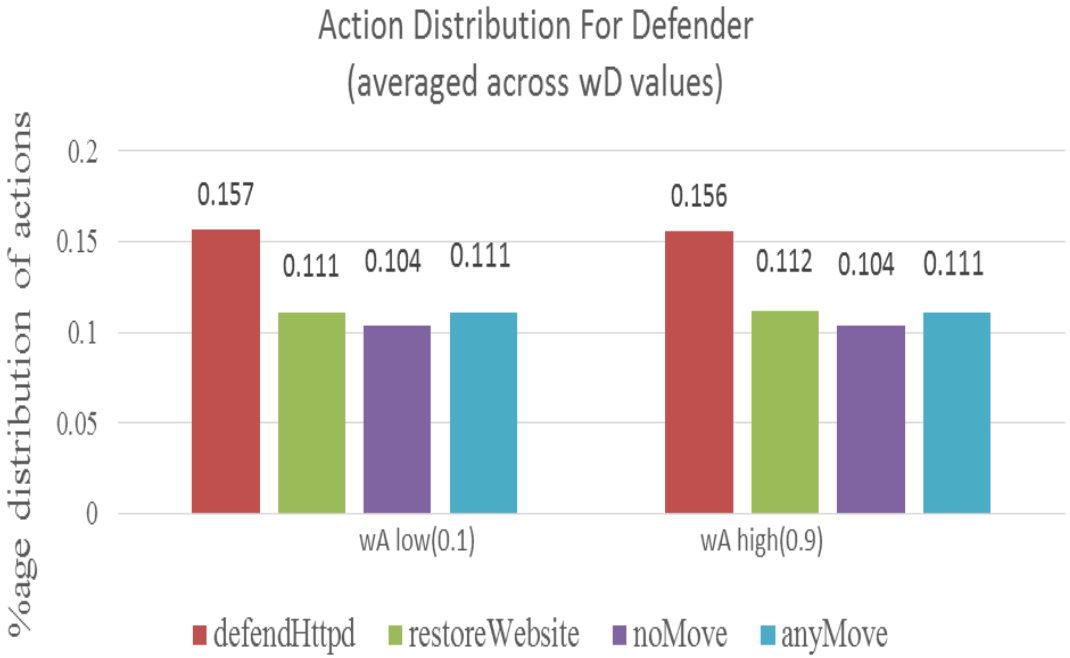
Effect of weight parameter on the actions of Attacker and Defender





Effect of weight parameter on the actions of Attacker and Defender





Summary of Results

- First, we found that greater attention to current outcomes led the attacker to win more games compared to lesser attention to current outcomes. The reason is that a higher wA value means that the attacker has a greater propensity of repeating a rewarding action and migrating away from a penalizing action.

- Further, it was found that relative percentage of simulations for which the defender's score is more than the attacker's score is approximately two times than that of the attacker. This suggests that our implementation is essentially defender biased, although the penalty for the defender was more in the higher network states.

- Finally, we found that the network state for the most time remains in the Normal State only. This result is clear by the fact that the Defaced_Website state is seldom reached since the probability of attacker's attack being successful and hence win is very low as per the results from the model.

CONCLUSION

- As we have seen, results of our simulations are inline with reinforcement learning model.
- Overall, it was observed that if attacker pays more attention to recent outcomes, then he is more likely to perform attack actions; whereas, paying more attention to recent outcomes did not influence defender's actions.
- These results may be used to understand how rewards and costs for attackers and defenders play a role in shaping actions in the cyber security domain.

References

Johnson, N.B. (2011). Cyber attacks up 40%, report says,Federal Times, Retrieved on , April 3, 2011 from http://www.federaltimes.com/

Arora, A., & Dutt, V. (2013). Cyber Security: Evaluating the Effects of Attack Strategy and Base Rate through InstanceBased Learning. In Paper presented at the 12th International Conference on Cognitive Modeling. Ottawa, Canada.

Dutt, V., Ahn, Y. S., & Gonzalez, C. (2013). Cyber situation awareness modeling detection of cyber attacks with instance-based learning theory.*Human Factors:* The Journal of the Human Factors and Ergonomics Society,55(3), 605-618.

Kaur, A., & Dutt, V. (2013). Cyber Situation Awareness: Modeling the Effects of Similarity and Scenarios on Cyber Attack Detection. In Paper presented at the 12th International Conference on Cognitive Modeling. Ottawa, Canada.

Barto, A. G. (1998). Reinforcement learning: An introduction. MIT press.

Lye, K. W., & Wing, J. M. (2005). Game strategies in network security. International Journal of Information Security, *4*(1-2), 71-86.

THANK YOU

Contact us:

Palvi aggarwal

Email: palvi_aggarwal@students.iitmandi.ac.in

Dr. Varun dutt

Email: varun@iitmandi.ac.in

Applied Cognitive Science Laboratory

Indian Institute of Technology, Mandi, India – 175005