Markov Game Modeling of Moving Target Defense for Strategic Detection of Threats in Cloud Networks

Sailik Sengupta*,
Subbarao Kambhampati

Yochan AI Lab

*Ankur Chowdhary**,
Dijiang Huang

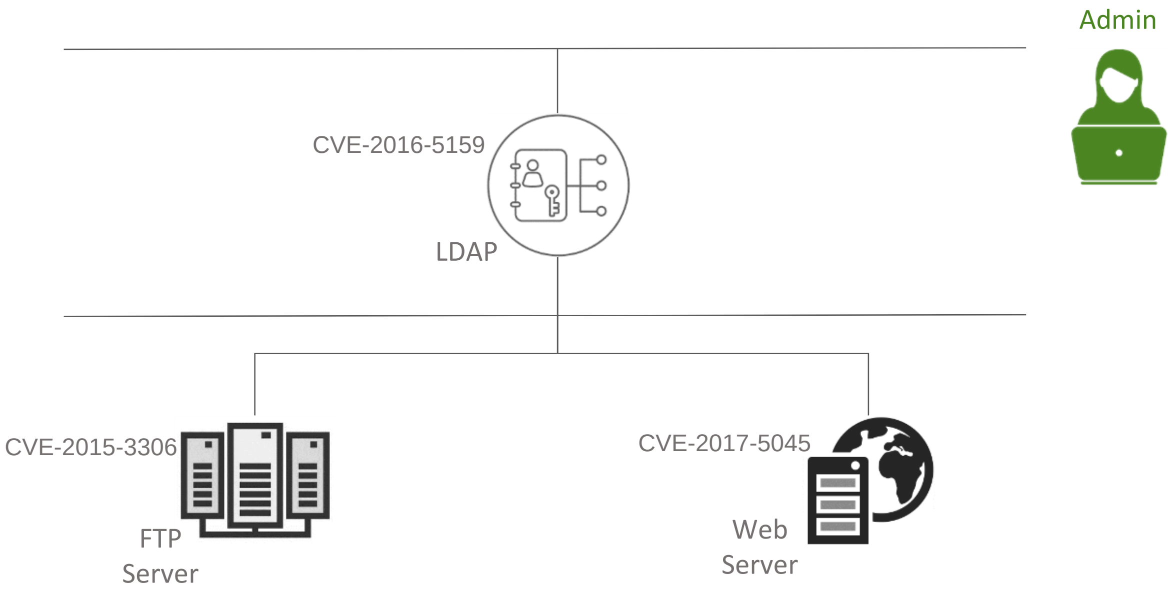
SNAC

Secure Networking and Computing Lab

Artificial Intelligence for Cyber Security (AICS), 2019

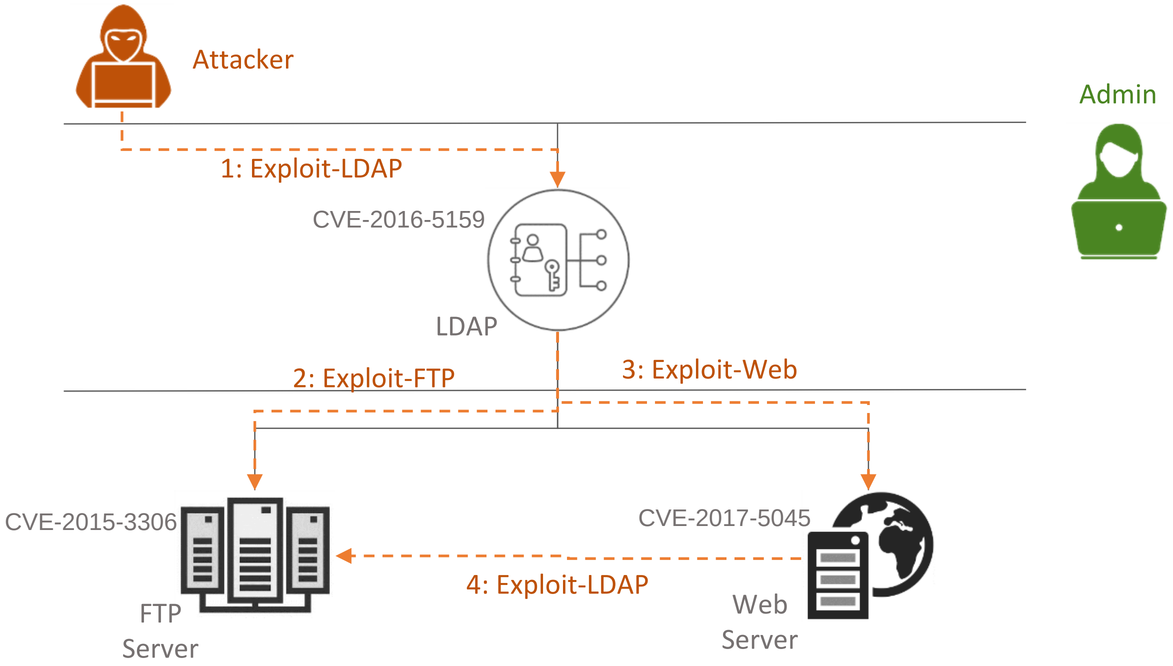
Markov Game Modeling of Moving Target Defense for Strategic Detection of Threats in Cloud Networks

- Cloud service providers provide computing and network resources to third parties for business.
- Attackers seek to attack such systems leading to a loss of Confidentiality, Availability and/or Integrity.
- Defenders can choose to monitor attacks on these systems using intrusion detection systems.



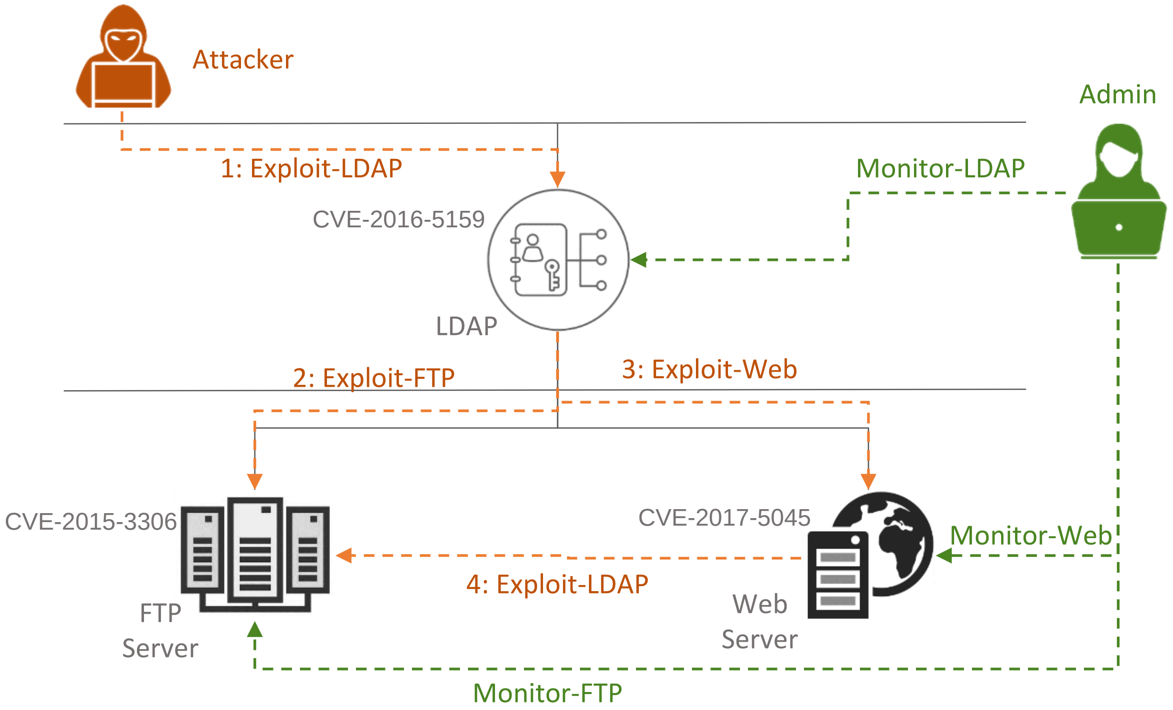
Markov Game Modeling of Moving Target Defense for Strategic Detection of Threats in Cloud Networks

- Cloud service providers provide computing and network resources to third parties for business.
- Attackers seek to attack such systems leading to a loss of Confidentiality, Availability and/or Integrity.
- Defenders can choose to monitor attacks on these systems using intrusion detection systems.



Markov Game Modeling of Moving Target Defense for Strategic Detection of Threats in Cloud Networks

- Cloud service providers provide computing and network resources to third parties for business.
- Attackers seek to attack such systems leading to a loss of Confidentiality, Availability and/or Integrity.
- Defenders can choose to monitor attacks on these systems using intrusion detection systems.

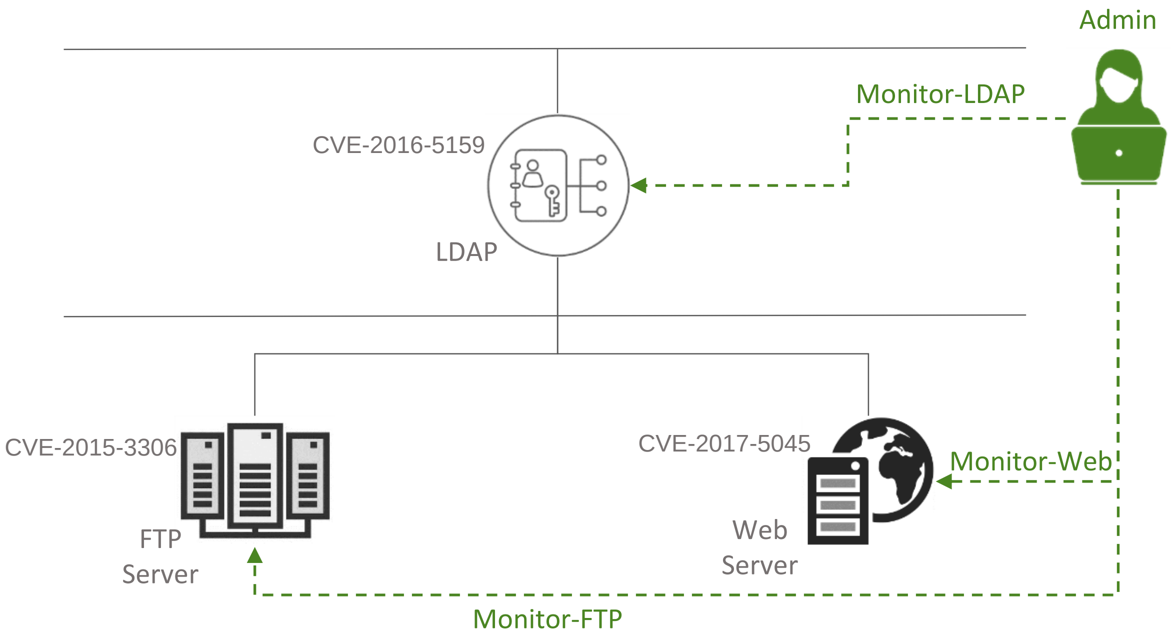


Markov Game Modeling of Moving Target Defense for Strategic Detection of Threats in Cloud Networks

- Place all possible Network and Host-Based Intrusion Detection Systems.

☺ Every known attack can be detected.

☹ Network Performance and Computing Resources are used up for security leading to lower Quality of Service (QoS) for actual customers.



Venkatesan, S et al. 2016, Sengupta et. al. 2018

Markov Game Modeling of Moving Target Defense for Strategic Detection of Threats in Cloud Networks

Attack + Exploration Surface Shifting
Zhuang et. al. 2014
Venkatesan 2016
Lei et al. 2017

Exploration Surface Shifting
Al-Shaer et. al. 2013
Jajodia et. al. 2018

Hot topic for physical security

Attack Surface Shifting
Manadhata et. al. 2013
Zhu and Bashar 2013
Carter et. al. 2014
Prakash and Wellman 2015
Sengupta et. al. 2016, 2017
Chowdhury et. al. 2016
B. Bohara 2017

Detection Surface Shifting
Venkatesan et. al. 2016
Sengupta et al. 2018
Chowdhury* et al. 2019

Prevention Surface Shifting

Uses Stackelberg Security Games.
-- Attacks are either successful or detected with 100% accuracy.
– Do not model multi-stage attacks..
-- Attacker has capability to attack any node on the system as opposed to planning an attack path.

Uses centrality based measures.
-- Higher centrality node sees more attack traffic.
-- Strategy optimizes performance by moving IDS between HCNs.

ARIZONA STATE UNIVERSITY

Markov Game Modeling of Moving Target Defense for Strategic Detection of Threats in Cloud Networks

Two-Player Markov Games

Markov Game (Shapley 1953) for two players P_1 and P_2 can be defined by the tuple $(S, A_1, A_2, \tau, R, \gamma)$ where,

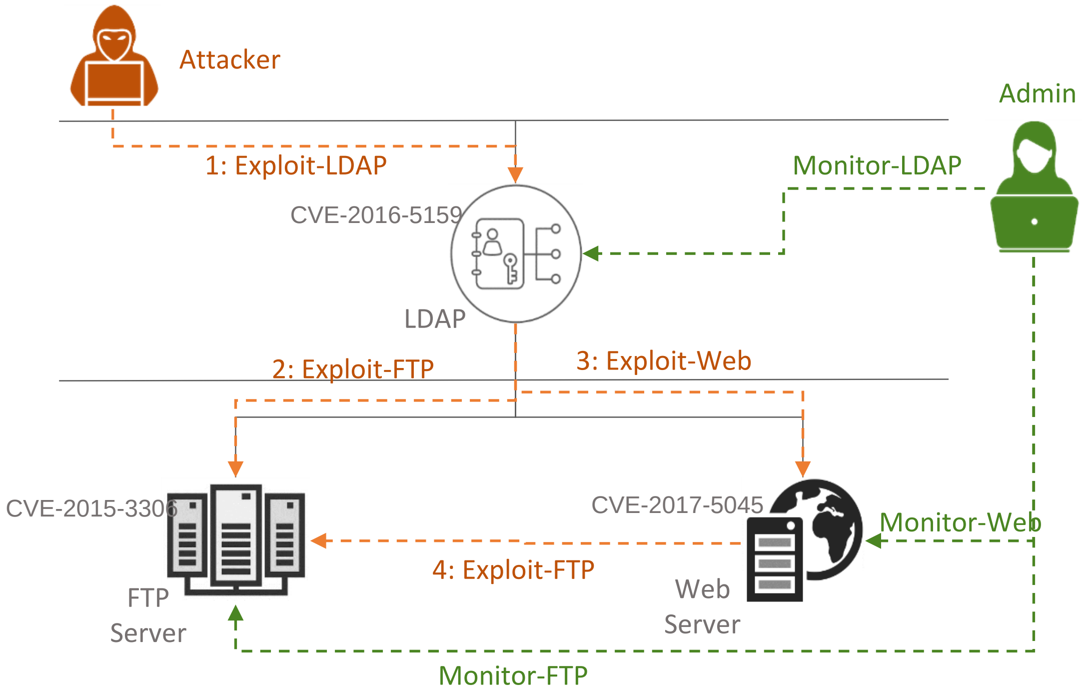
- $S = \{s_1, s_2, s_3, \ldots, s_k\}$ are finite states of the game,
- $A_1 = \{a_1^1, a_1^2, \ldots, a_1^m\}$ represents the possible finite action sets for P_1,
- $A_2 = \{a_2^1, a_2^2, \ldots, a_2^n\}$ are finite action sets for P_2,
- $\tau(s, a_1, a_2, s')$ is the probability of reaching a state $s' \in S$ for state s if P_1 and P_2 take actions a_1 and a_2 respectively,
- $R^i(s, a_1, a_2)$ is the reward obtained by P_i if in state s, P_i and P_{-i} take the actions a_1 and a_2 respectively, and
- $\gamma \mapsto [0, 1)$ is discount factor for future discount rewards.

$$Q(s, a_1, a_2) = R(s, a_1, a_2) + \gamma \sum_{s'} \tau(s, a_1, a_2, s') \cdot V(s')$$

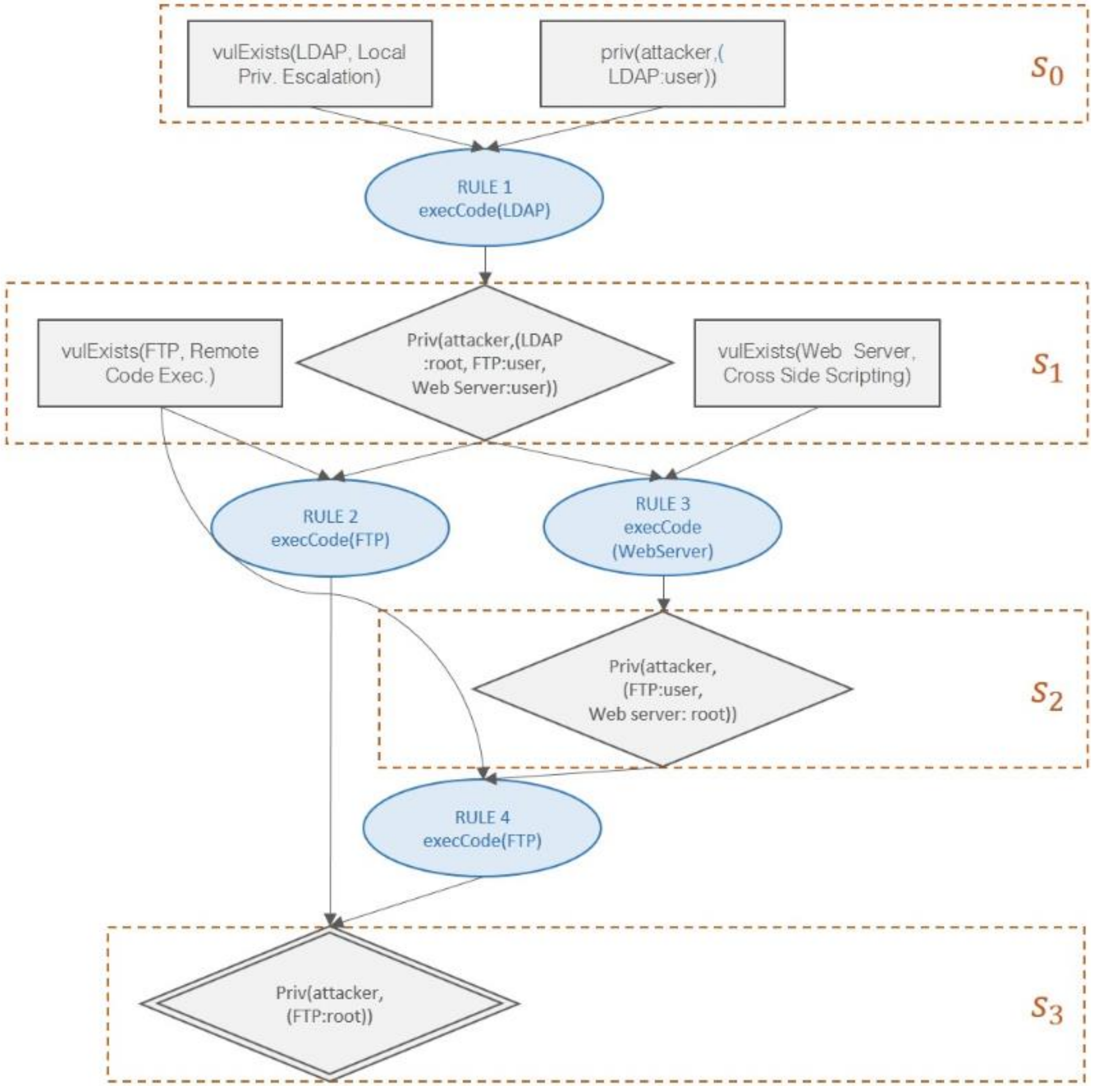
$$V(s) = \max_{\pi(s)} \min_{a_2} \sum_{a_1} Q(s, a_1, a_2) \cdot \pi_{a_1}$$

Can be solved using a Linear Program when updating the Value in every iteration.

States



Sample Network scenario

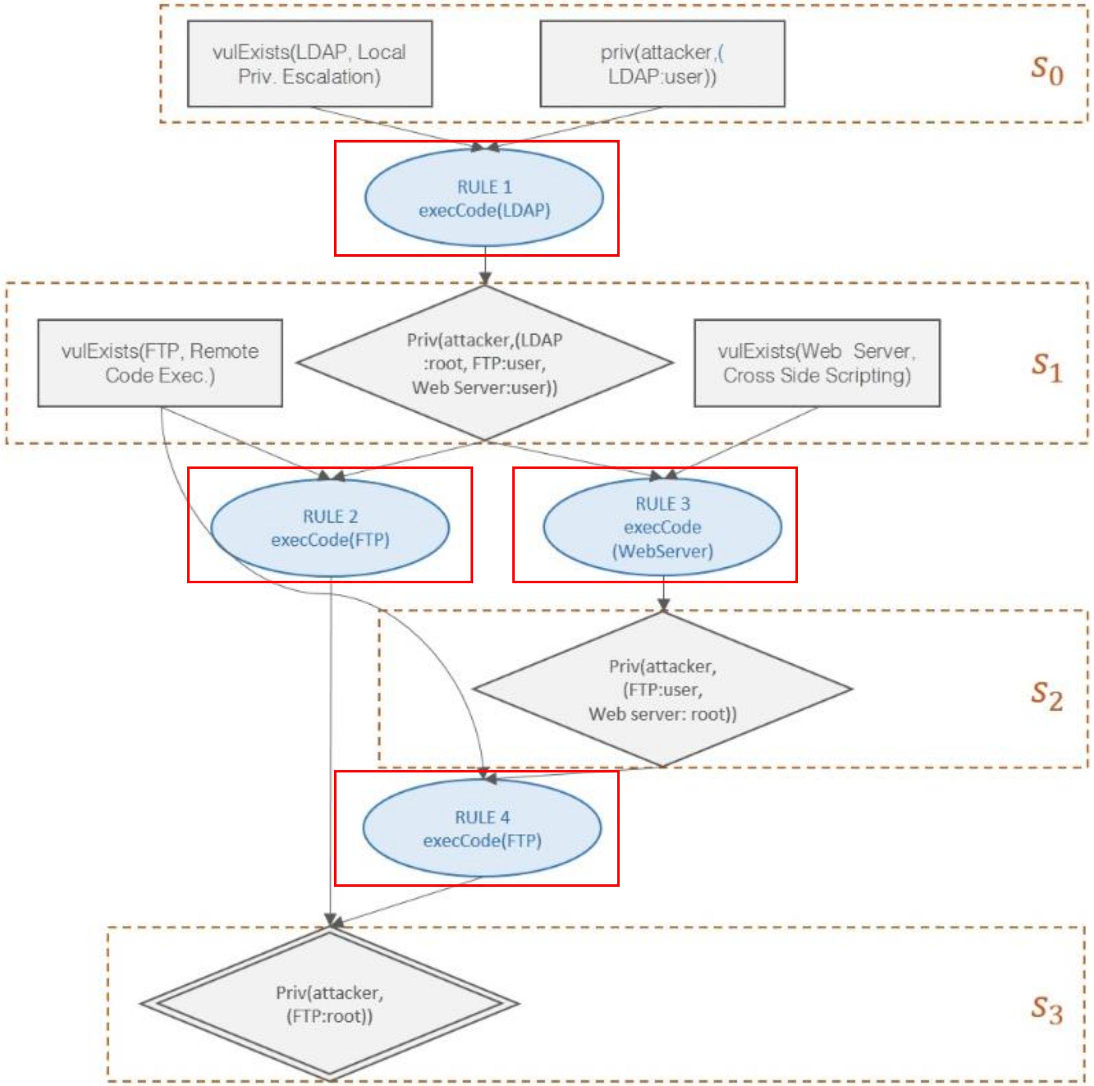


Corresponding Attack Graph

Actions

	no-mon	mon-Web	mon-FTP
no-act	$0, 0$	$2, -2$	$3, -3$
exp-Web	$7, -7$	$-5, 5$	$10, -10$
exp-FTP	$10, -10$	$10, -10$	$-7, 7$

s_1



Defender and Attacker Actions

Corresponding Attack Graph

Reward and Transition Model

Common Vulnerability Scoring Service (CVSS)

Impact Score of a Common Vulnerability and Exposures (CVE)

Assumes that the rewards are zero-sum structure.

- Note that this may not be true since attacker does not care about defenders performance metrics or QoS to legitimate users.

How to find a value for the effect on QoS given that a monitoring system is deployed.

- Venkateshan et. al. 2016 and Sengupta et. al. 2018 uses centrality measure of the nodes as a heuristic to estimate this value.

- We feel that a better estimate can be found by testing the impact on bandwidth and measuring increase in CPU usage and using these value by scaling it appropriately w.r.t the CVSS scores.

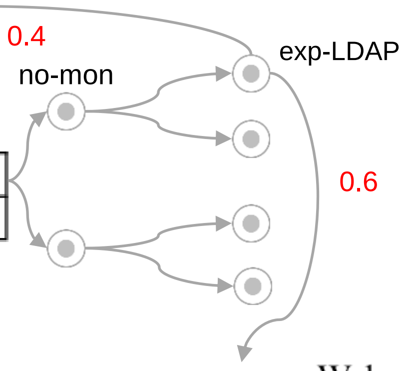
Exploitability score of a Common Vulnerability and Exposures (CVE)

Assumption is based on the fact that a random attacker is more likely to succeed if the attack is easy to exploit.

- Chung et. al. 2013 shows how Exploitability Scores can be used in attack graphs for calculating the probability of an attacker being able to successfully exploit an attack.

	no-mon	mon-LDAP
no-act	0, 0	3, −3
LDAP	5, −5	−5, 5

s_0



	no-mon	mon-FTP
no-act	0, 0	2, −2
exp-FTP	10, −10	−8, 8

CVE-2015-3306

s_2

	no-mon	mon-Web	mon-FTP
no-act	0, 0	2, −2	3, −3
exp-Web	7, −7	−5, 5	10, −10
exp-FTP	10, −10	10, −10	−7, 7

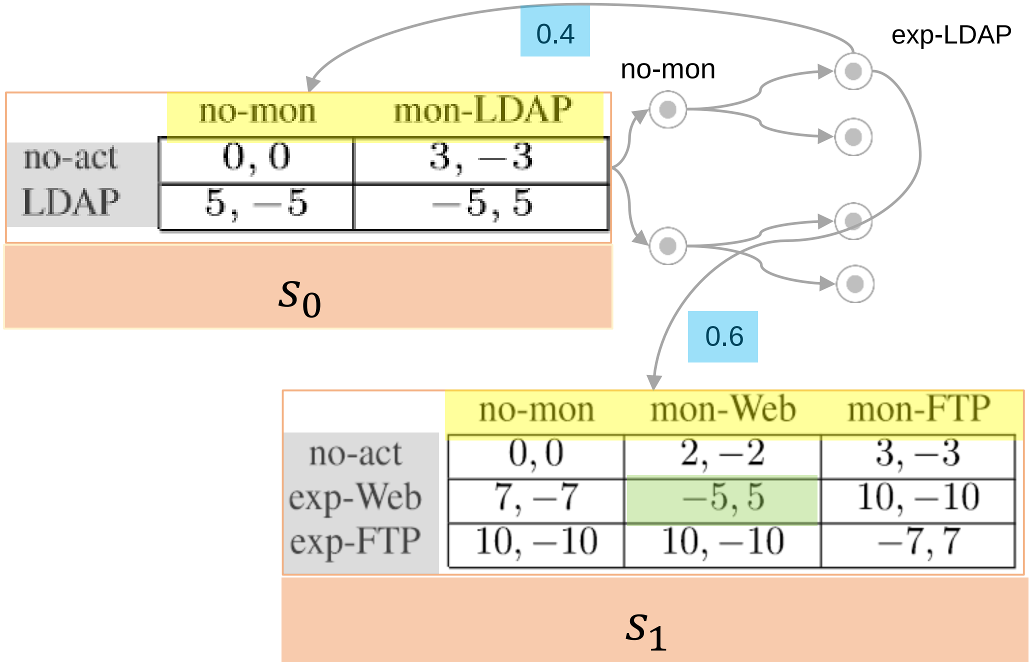
s_1

Markov Game Modeling of Moving Target Defense for Strategic Detection of Threats in Cloud Networks

Two-Player Markov Games

Markov Game (Shapley 1953) for two players P_1 and P_2 can be defined by the tuple $(S, A_1, A_2, \tau, R, \delta)$ where,

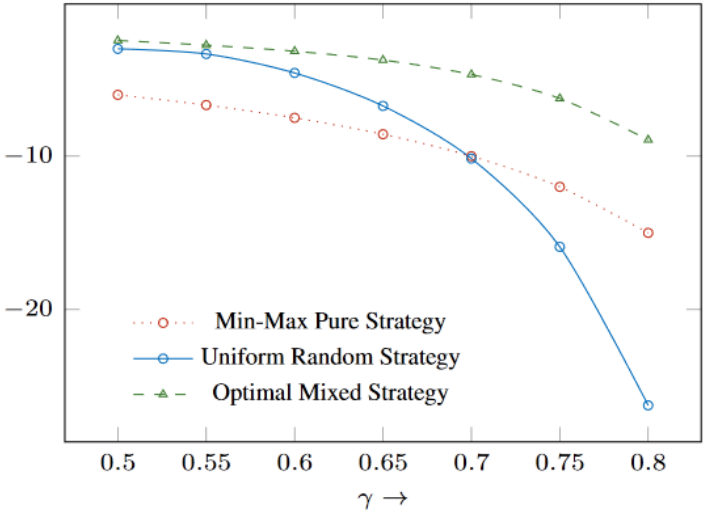
- $S = \{s_1, s_2, s_3, \ldots, s_k\}$ are finite states of the game,
- $A_1 = \{a_1^1, a_1^2, \ldots, a_1^m\}$ represents the possible finite action sets for P_1,
- $A_2 = \{a_2^1, a_2^2, \ldots, a_2^n\}$ are finite action sets for P_2,
- $\tau(s, a_1, a_2, s')$ is the probability of reaching a state $s' \in S$ for state s if P_1 and P_2 take actions a_1 and a_2 respectively,
- $R^i(s, a_1, a_2)$ is the reward obtained by P_i if in state s, P_i and P_{-i} take the actions a_1 and a_2 respectively, and
- $\gamma \mapsto [0, 1)$ is discount factor for future discount rewards.

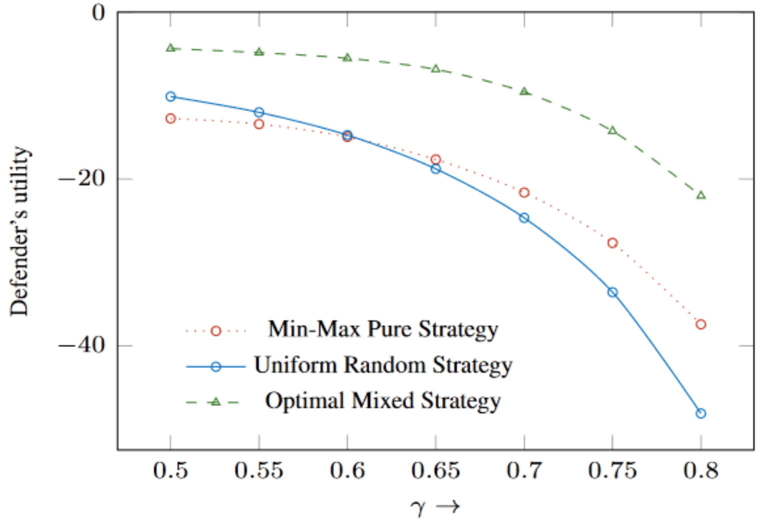


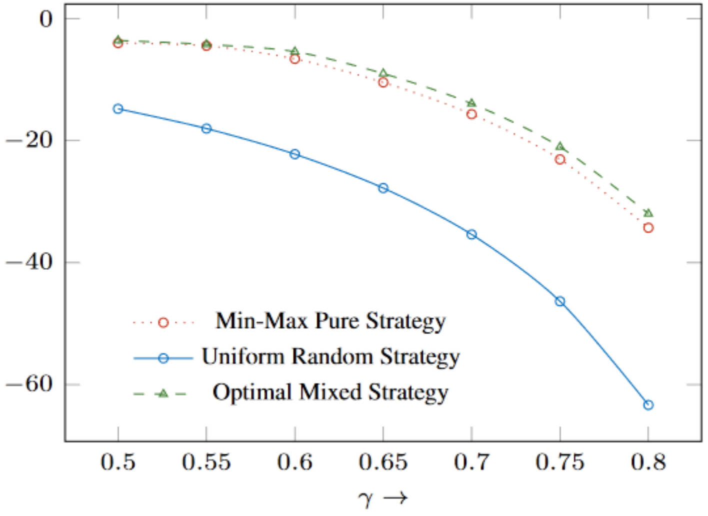
	no-mon	mon-LDAP
no-act	0, 0	3, −3
LDAP	5, −5	−5, 5

	no-mon	mon-Web	mon-FTP
no-act	0, 0	2, −2	3, −3
exp-Web	7, −7	−5, 5	10, −10
exp-FTP	10, −10	10, −10	−7, 7

Results

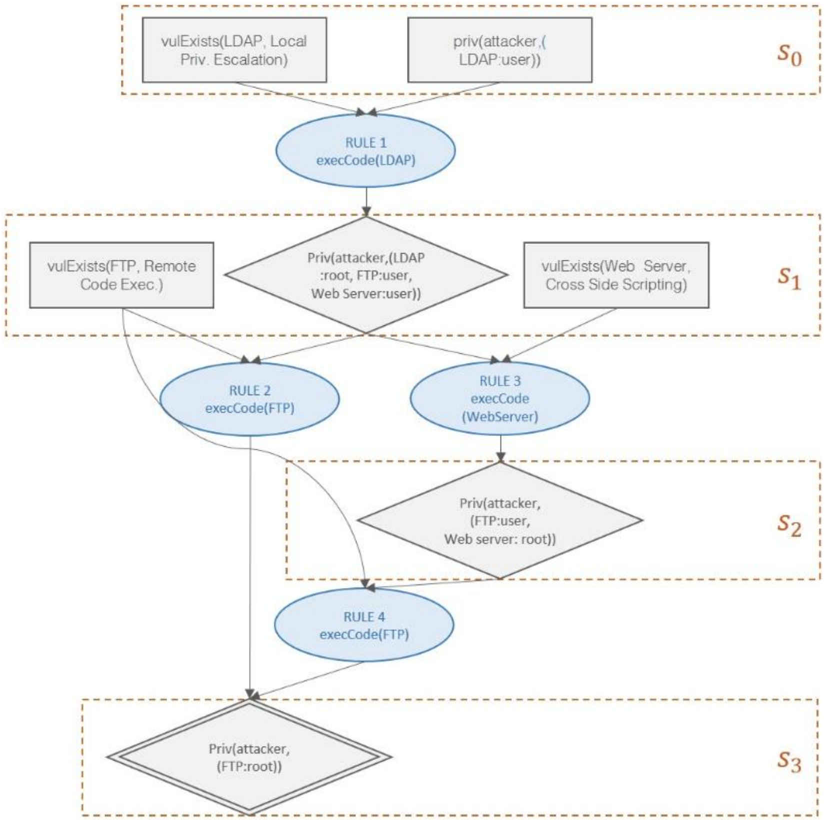






$$\pi(s_1) = [\quad 0.0, \quad 0.547, \quad 0.453 \quad]$$

	no-mon	mon-Web	mon-FTP
no-act	$0, 0$	$2, -2$	$3, -3$
exp-Web	$7, -7$	$-5, 5$	$10, -10$
exp-FTP	$10, -10$	$10, -10$	$-7, 7$



Results

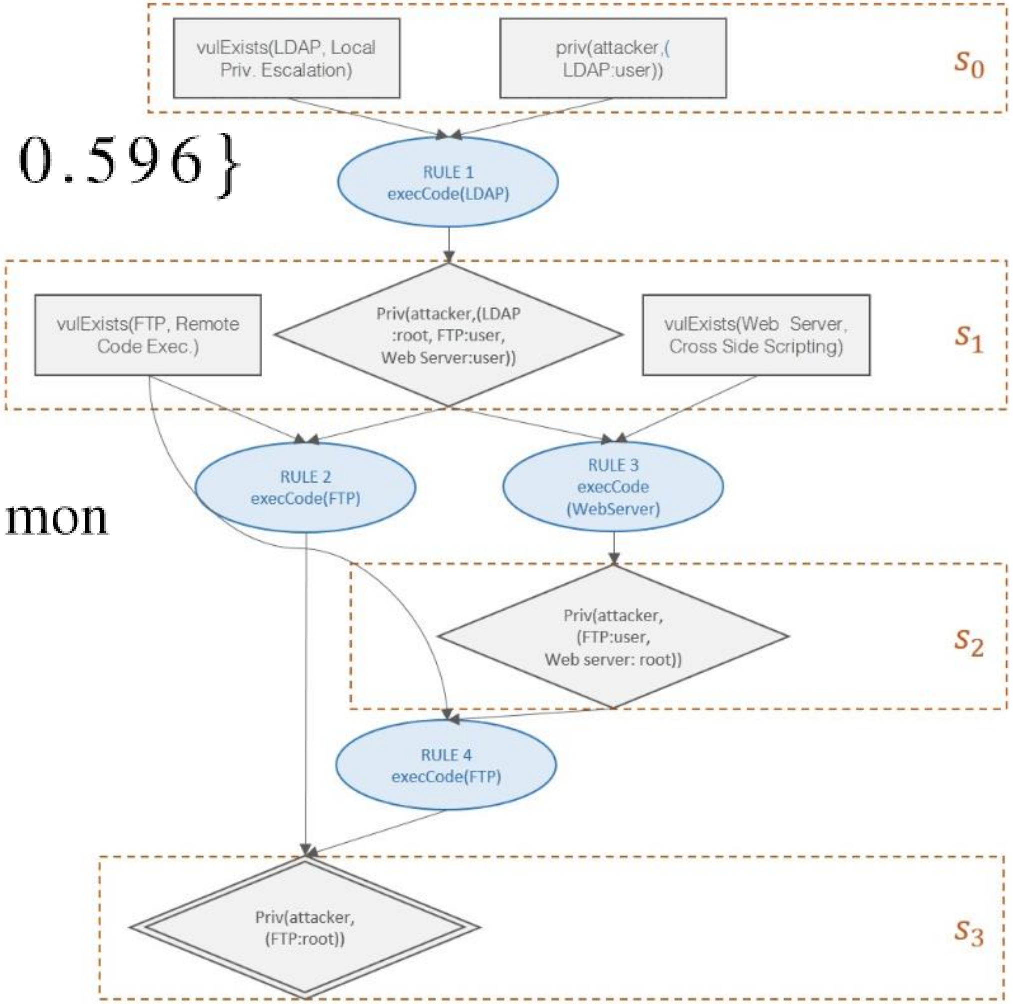
$$\pi(s_0) : \{\text{no–mon}: 0.404, \text{mon–LDAP}: 0.596\}$$

For states further away from the goal, don't need to monitor at times to enhance performance QoS.

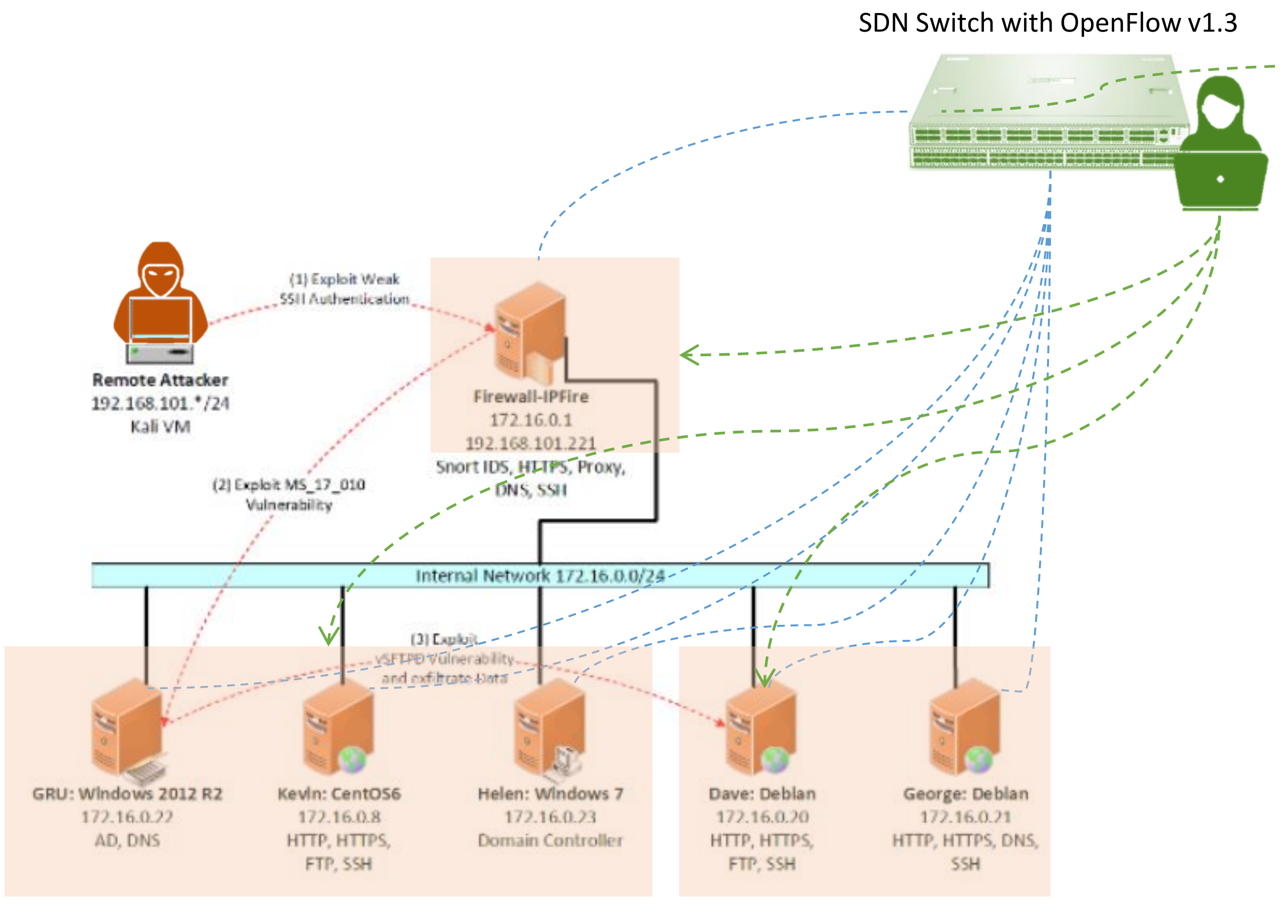
$$\pi(s_1) : \{\text{no–mon}: 0.0, \text{mon–Web}: 0.547, \text{mon–FTP}: 0.453\}$$

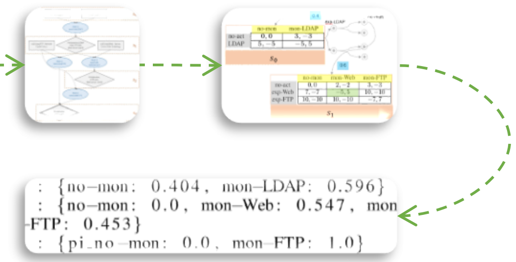
$$\pi(s_2) : \{\text{pi\_no–mon}: 0.0, \text{mon–FTP}: 1.0\}$$

For states closer to the goal, not monitoring is not an option. Security becomes more important that performance.



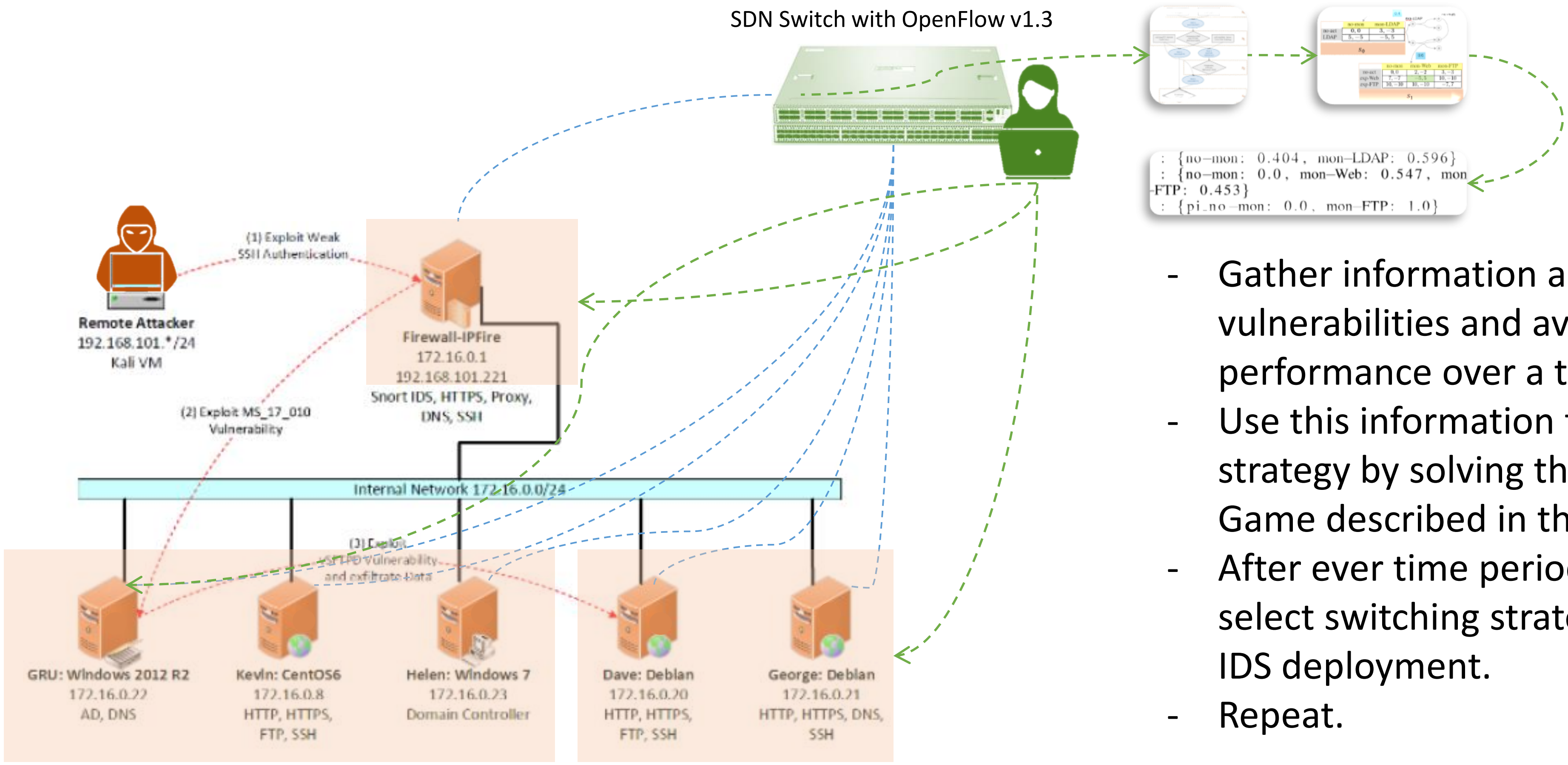
Implementation in the Real World





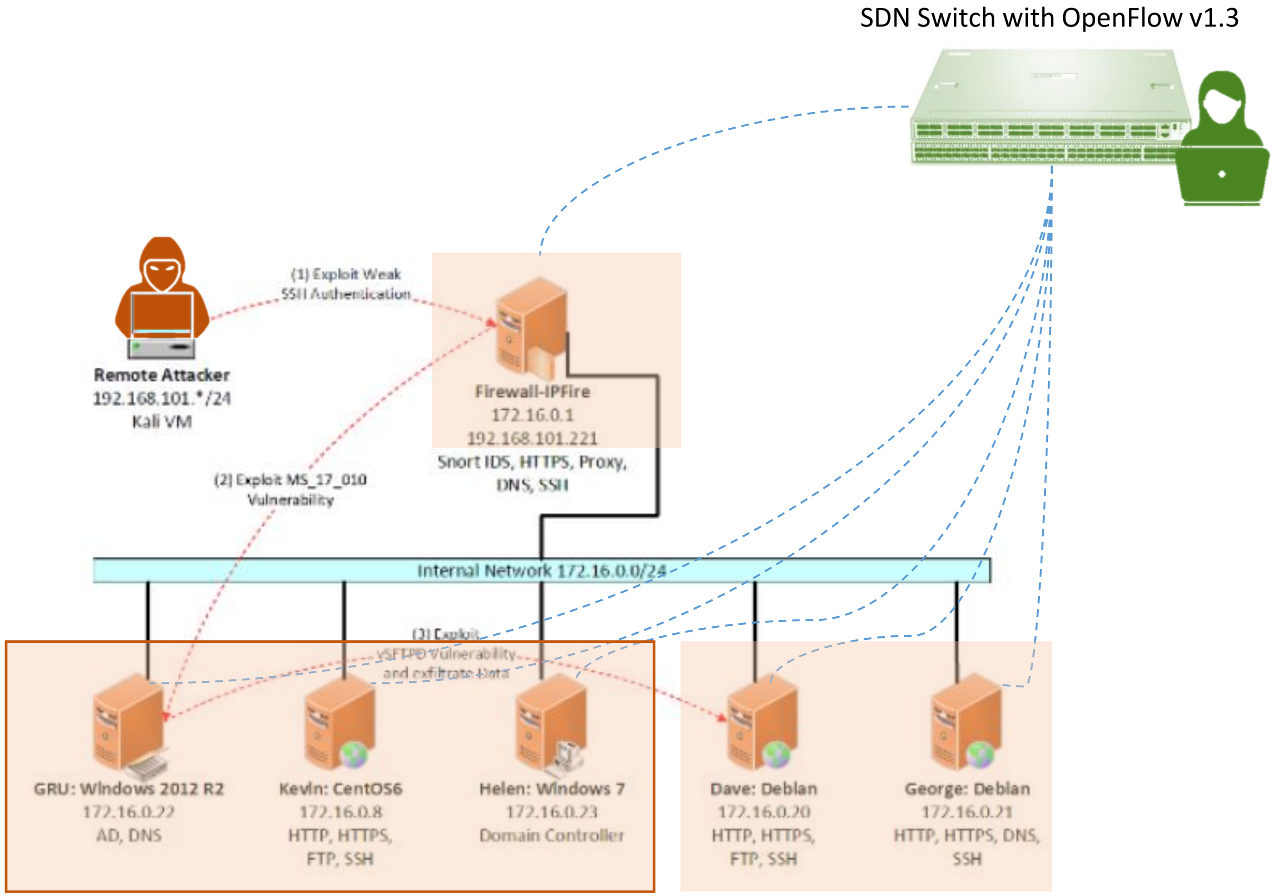
- Gather information about new vulnerabilities and average network performance over a time period *T*
- Use this information to precompute a strategy by solving the formulated Markov Game described in this work.
- After ever time period *t* << *T*, randomly select switching strategy and change the IDS deployment.
- Repeat.

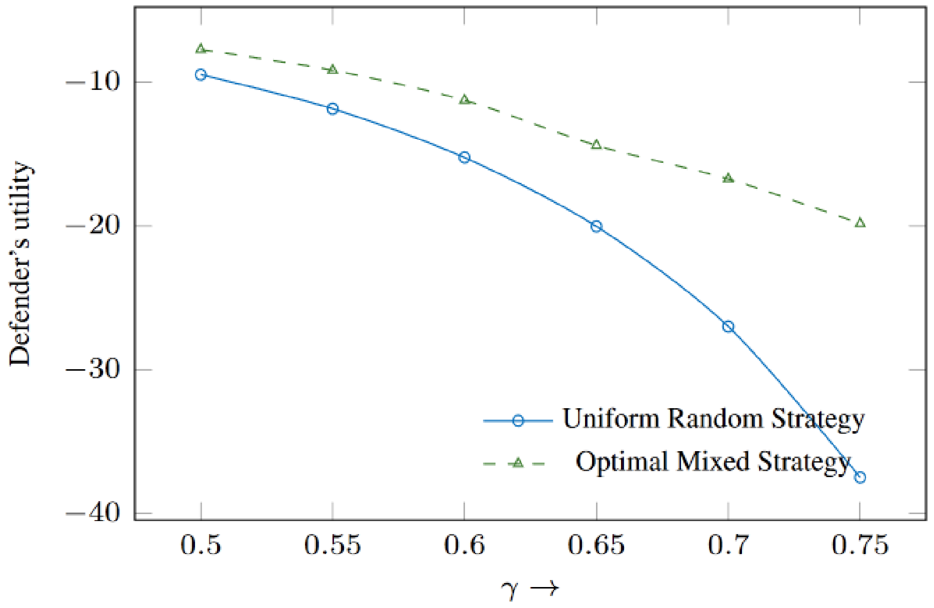
Implementation in the Real World



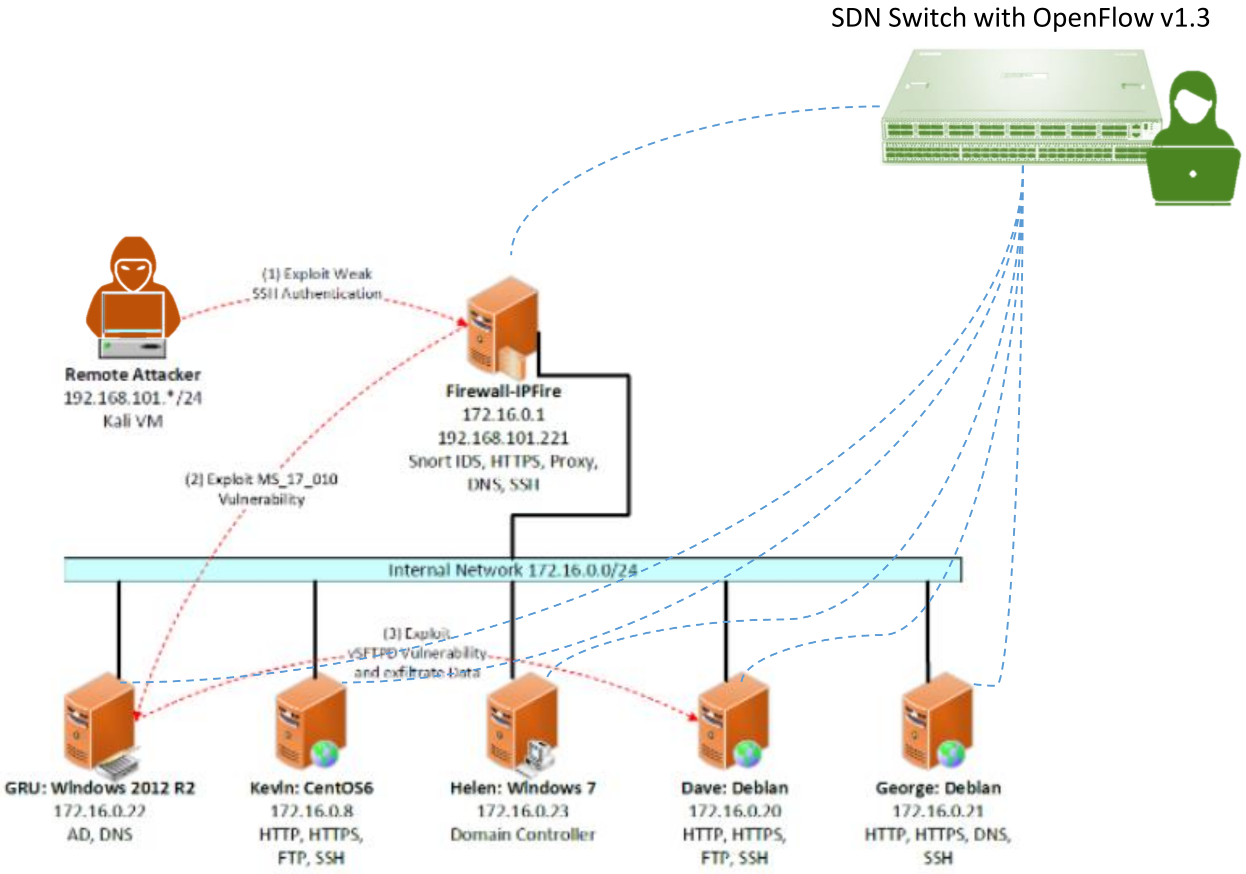
- Gather information about new vulnerabilities and average network performance over a time period *T*
- Use this information to precompute a strategy by solving the formulated Markov Game described in this work.
- After ever time period *t* << *T*, randomly select switching strategy and change the IDS deployment.
- Repeat.

Implementation in the Real World





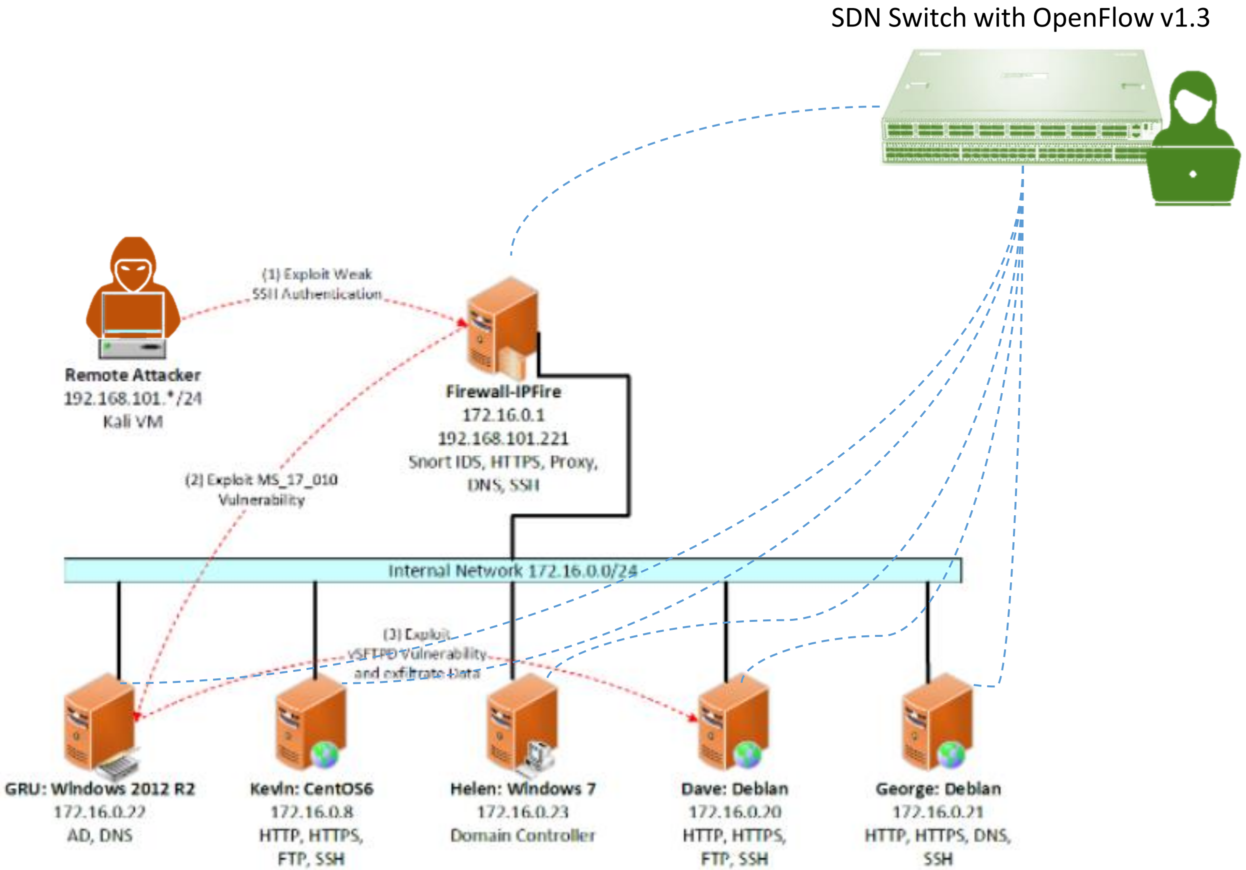
Conclusion & Future Work



- We formulated the placement of IDS systems in the cloud as a Markov Game. We found strategies for efficient detection surface shifting which allows the defender to trade-off between security and Quality of Service.

- We hope to relax a set of assumptions we made in this work in the future—
  - Zero sum game?
  - Game states are visible to both the players?
  - What happens when this is simulated in a real world cloud network?
  - How does the incomplete knowledge of existing attacks and irrationality of attackers affect the quality of solution?
  - How does one reason about the zero day attacks – incomplete knowledge of the defender about the attacks?

Conclusion & Future Work



- We formulated the placement of IDS systems in the cloud as a Markov Game. We found strategies for efficient detection surface shifting which allows the defender to trade-off between security and Quality of Service.

- We hope to relax a set of assumptions we made in this work in the future—
    - Zero sum game?
    - Game states are visible to both the players?
    - What happens when this is simulated in a real world cloud network?
    - How does the incomplete knowledge of existing attacks and irrationality of attackers affect the quality of solution?
    - How does one reason about the zero day attacks – incomplete knowledge of the defender about the attacks?