Australian Government
Department of Defence

Australian National University | RESEARCH SCHOOL OF COMPUTER SCIENCE

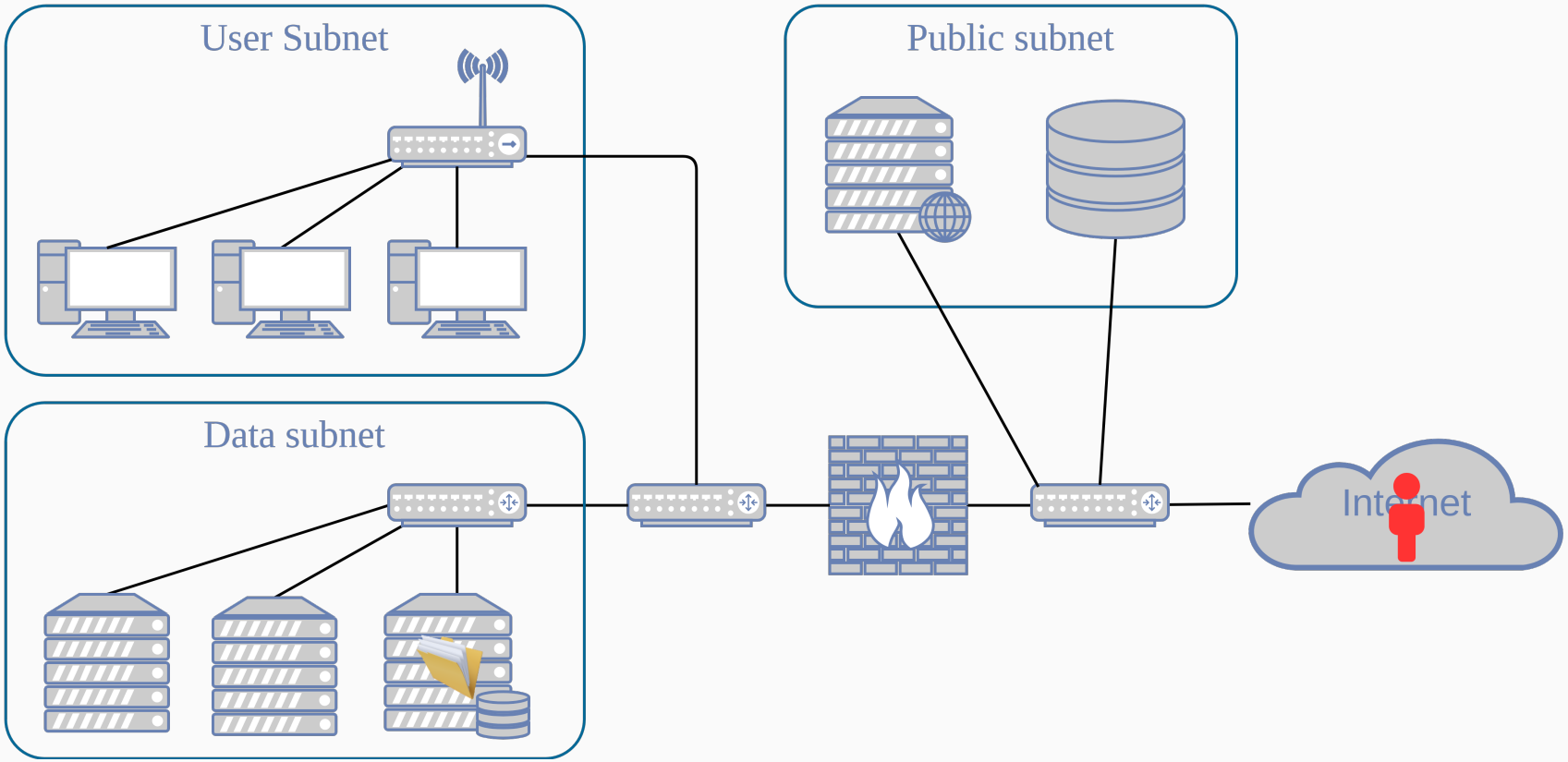
Information Decay + POMDP

Incorporating Defender's Behaviour in Autonomous Penetration Testing

Jonathon Schwartz[1], Hanna Kurniawati[1], and Edwin El-Mahassni[2]

[1] Research School of Computer Science, ANU
[2] Defence Science and Technology Group, Australian Department of Defence

Network Penetration Testing



Autonomous Penetration Testing

- We can view penetration testing as a sequential decision problem.
- Three sources of uncertainty:

1. Partial observability

2. Unreliable attack tools

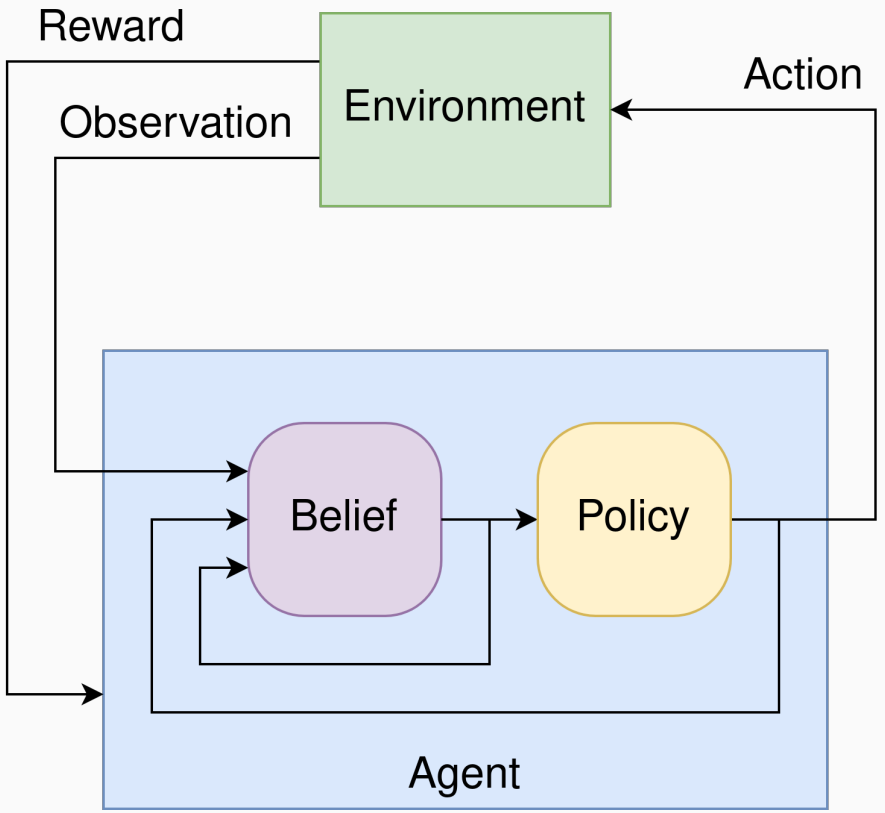
3. The defender

Current state of autonomous pen-testing

Method	Partial observability	Unreliable actions	Defender
Attack planning *(Lucangeli et al, SecArt '10)*	no	yes	no
POMDP *(Sarruate et al, AAAI '12)*	yes	yes	no
Stochastic game *(Lye and Wing, IJIS '05)*	no	yes	yes
This work	**yes**	**yes**	**yes**

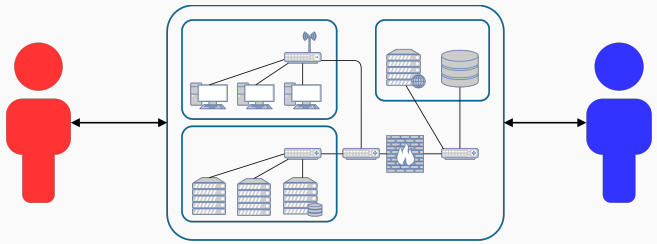
POMDPs

Partially Observable Markov Decision Process $\langle S, A, T, O, Z, R, \gamma \rangle$

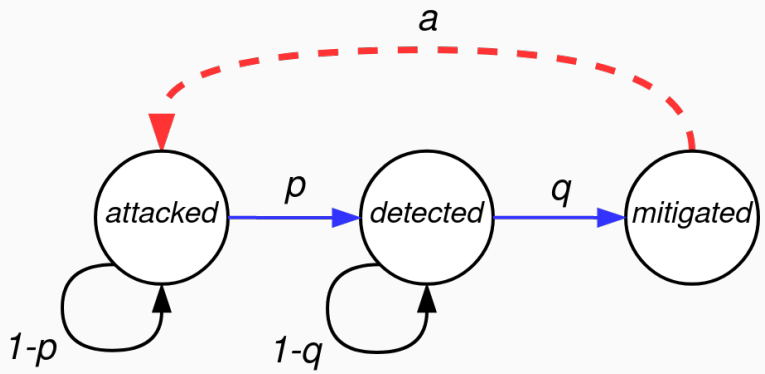


Idea for incorporating the defender

Pen-tester and defender can only infer each other via observed changes to the network state.

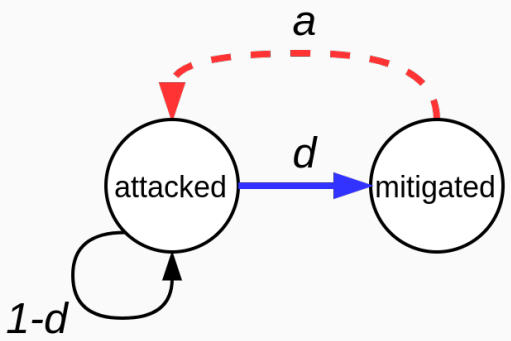


Our proposed idea: Model defender's behaviour as a Markovian Arrival Process (MAP)



Information Decay

This work: **Bernoulli process.**



Model defender by single parameter: the *information decay factor d*.

Intuitively, *d* is probability that the defender mitigates the pen-tester's action

For each system property we assume the same process and that each process is IID.

Information Decay + POMDP: D-PenTesting

Given $\mathcal{P} = \langle S, A, T, O, Z, R, \gamma \rangle$

Let $\mathbb{I}(a)$ be the affected set of $a \in A$, where $i \in \mathbb{I}(a)$ iff state variable s_i is changed or observed by a

Define transition T_d for state variable s_j:

$$T_d\left(s'_j \mid s_j, a\right) = \begin{cases} T\left(s'_j \mid s_j, a\right) & j \in \mathbb{I}(a) \\ d \cdot \frac{1}{|S'_j|-1} & j \notin \mathbb{I}(a) \text{ and } s'_j \neq s_j \\ 1-d & \text{otherwise.} \end{cases}$$

Requires knowing d beforehand.

Learning the defenders model: LD-PenTesting

Given D-PenTesting POMDP $\mathcal{P}_d = \langle S, A, T_d, O, Z, R, \gamma \rangle$

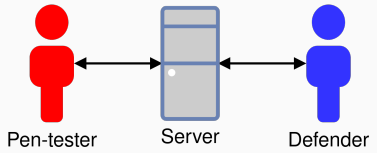
Define LD-PenTesting POMDP $\mathcal{P}_{ld} = \langle S_{ld}, A, T_{ld}, O, Z_{ld}, R_{ld}, \gamma \rangle$, where:

- A, O, γ are unchanged from $\mathcal{P}_d$
- $S_{ld} = S \times D$, where
  - D represents possible values of d
  - D discretised to resolution δ
  - Increases $|S|$ by $\frac{1}{\delta}$ fold
- $Z_{ld}(\langle s, d \rangle, a, o) = Z(s, a, o)$
- $R_{ld}(\langle s, d \rangle, a) = R(s, a)$
- $T_{ld}(\langle s, d \rangle, a, \langle s', d' \rangle) = T_d(s, a, s') \cdot \Delta_{dd'}$
  - where $\Delta_{dd'}$ is the Kronecker Delta (identity) function
  - T_d is transition function with decay factor d

Experimental Scenarios

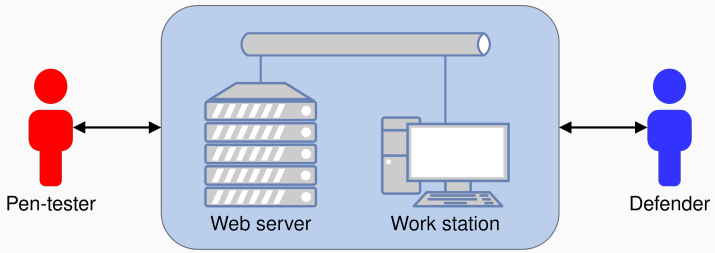
Scenario 1

Extends original scenario proposed for POMDP pen-testing by *Sarruate et al (AAAI '12)* to include a defender.



Scenario 2

Extends stochastic game scenario proposed by *Lye and Wing (IJIS '05)* to partially observable setting.



Experimental setup

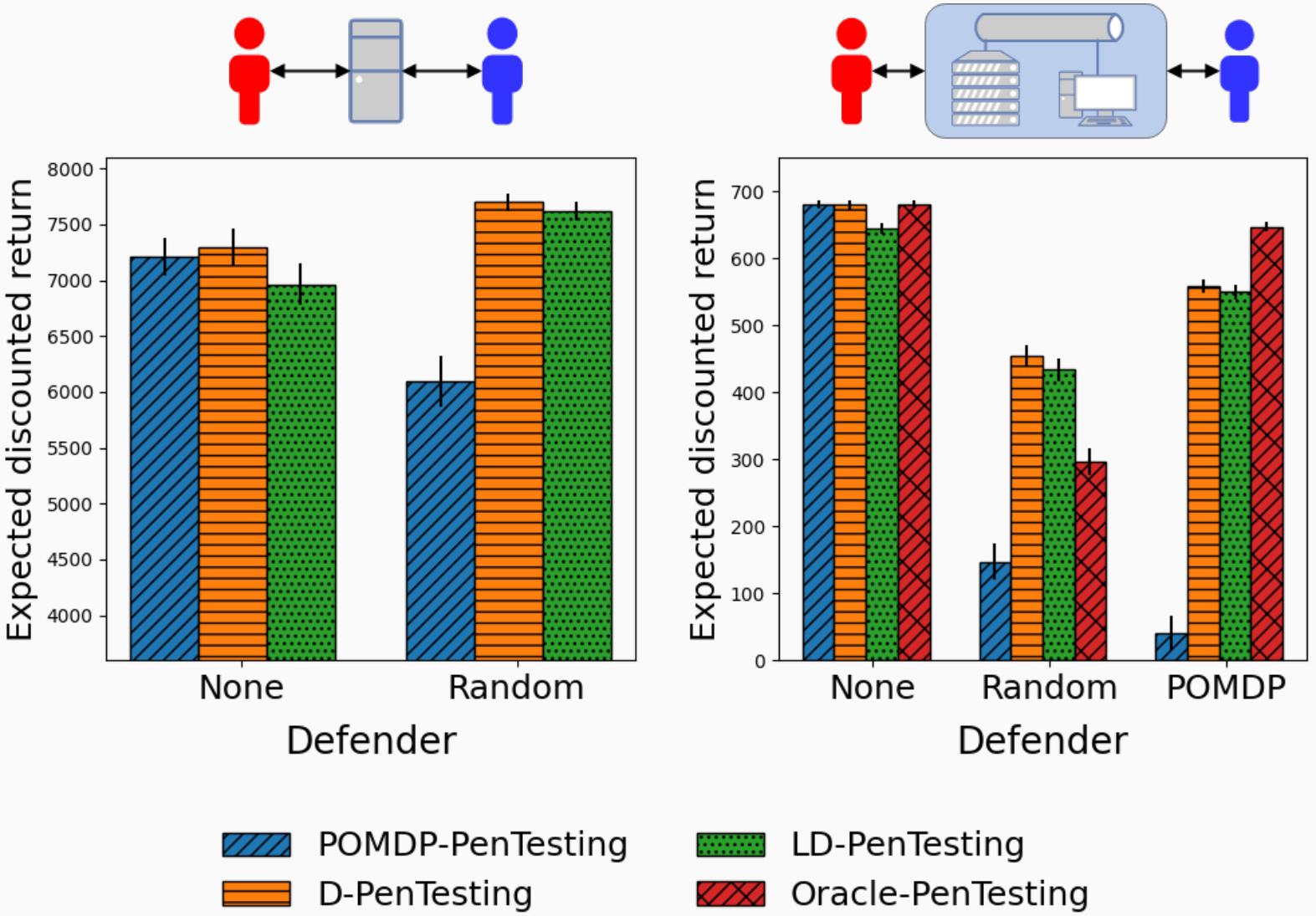
Planning

- Planning using SARSOP offline POMDP solver *(Kurniawati et al, RSS '08)*
- D-PenTesting and LD-PenTesting given no knowledge of defender during planning

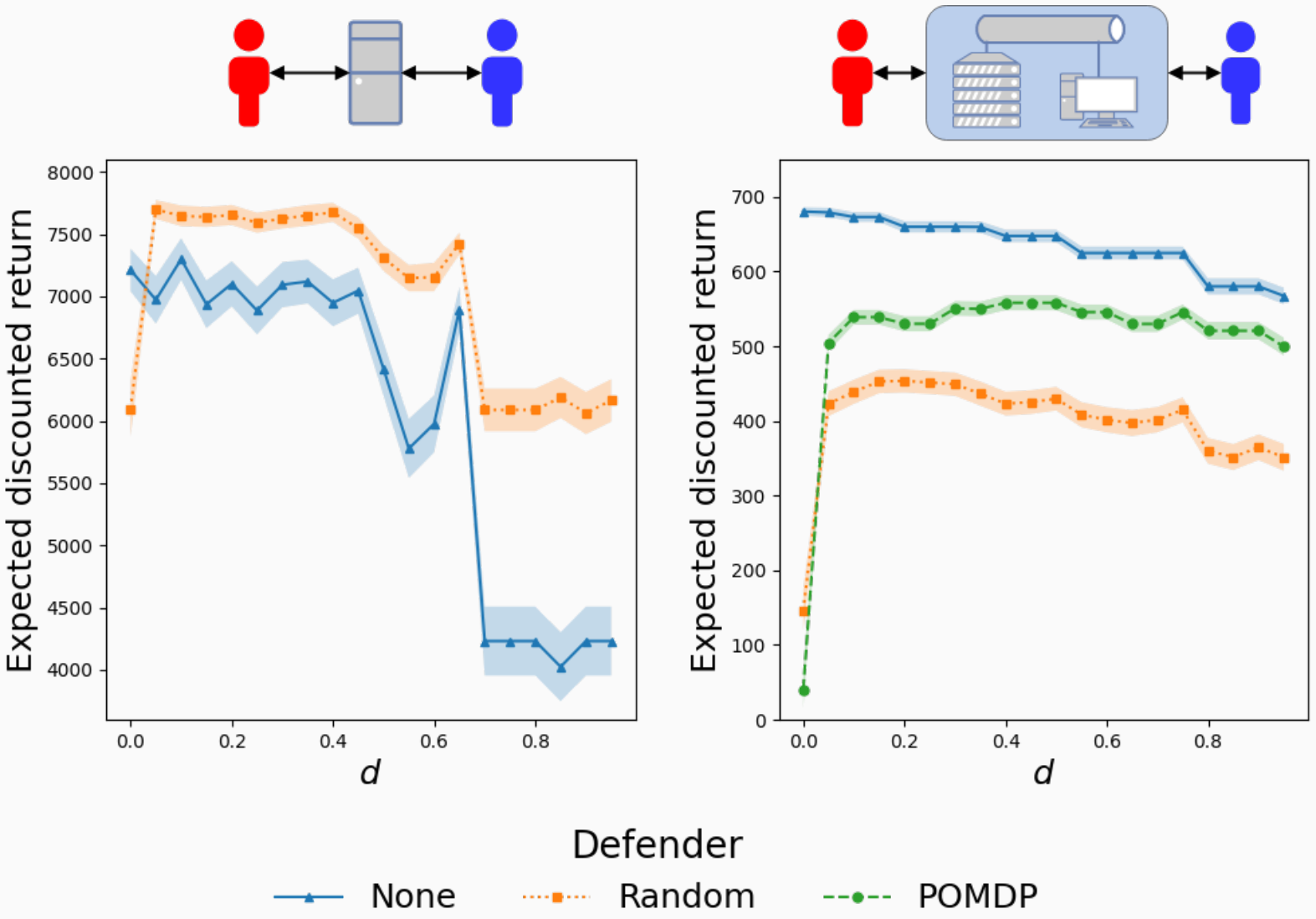
Simulation

- Tested each pen-tester agent against different defenders in simulation

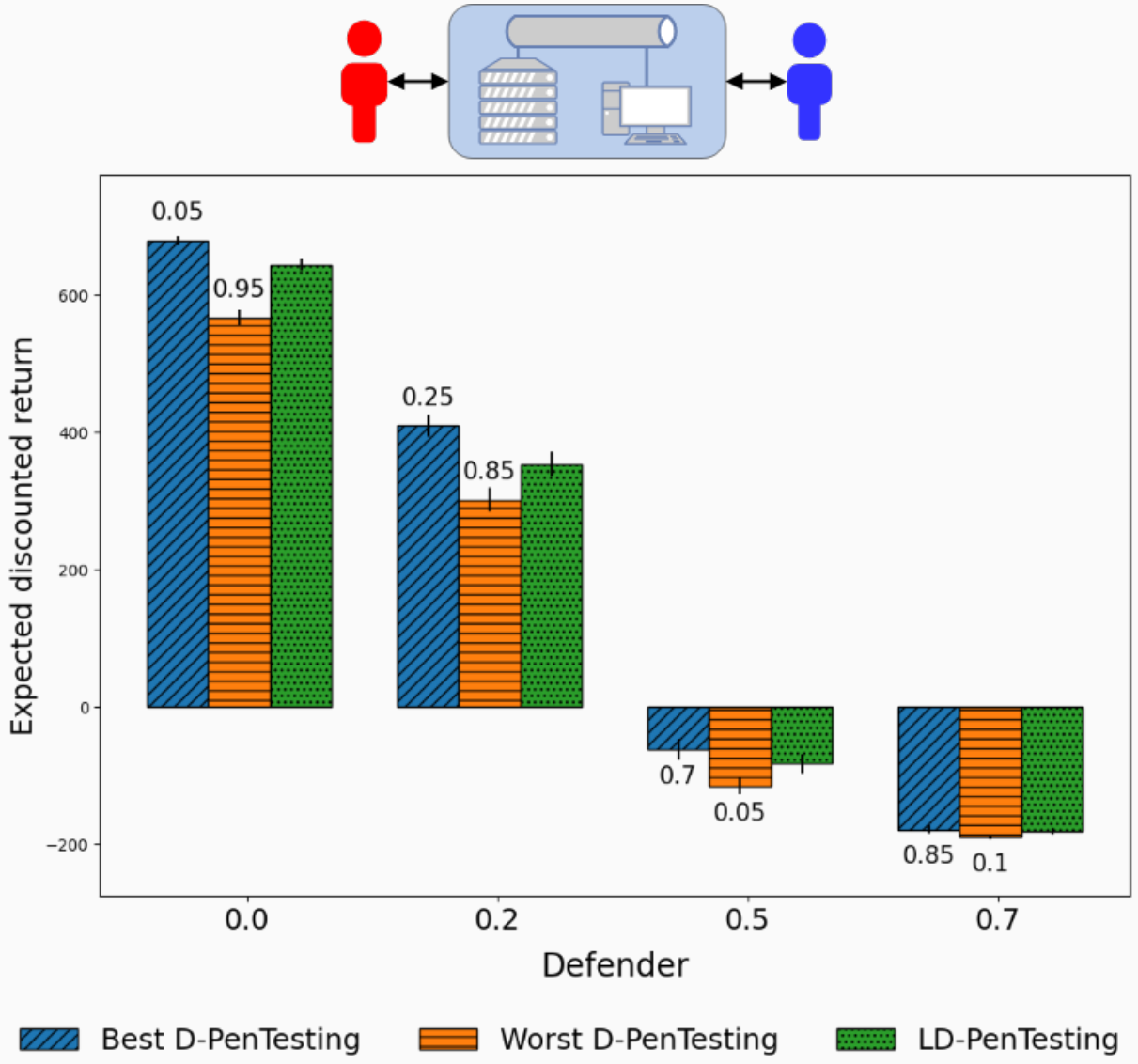
POMDP-PenTesting vs D-PenTesting vs LD-PenTesting



D-PenTesting performance



D-PenTesting vs LD-PenTesting



Conclusion

- Presented efficient abstract defender model based on MAP
- Incorporated this model to create D-PenTesting and LD-PenTesting.
- Our approach can handle the three main sources of uncertainty:
  1. partial observability,
  2. unreliable attack tools, and
  3. the defender

Thank you for listening. Questions?