Keep spending: Beyond optimal cyber-security investment

Yunxiao Zhang, Pasquale Malacaria

IEEE CSF 2023

what is this work about?

Game solutions (e.g. Nash equilibrium) are fundamental concepts in Game theory.

- A solution is an Optimal point.
- We argue however that for some games (e.g. security games) this optimality is suboptimal (i.e it can be improved) in the real world.

So the question this work aim to address is:

how can we improve this optimal (yet real-world suboptimal) game solution?

Security games are Stalkeber games (leader-follower games)

- The leader is the defender who selects an optimal portfolio of security controls, e.g. 2FA, encryption etc
- The follower is the attacker, who observes the leader's defense and select the sequence of steps most likely to reach some objective (e.g. become root user)
- The game solution here is for the leader to choose the strategy that minimizes the security risk, taking into account the most powerful attack (it is a bi-level optimization)

what is this work about?

Security games are Stalkeber games (leader-follower games)

- optimal game solution here means selecting a set of controls minimizing the security risk
- each control has associated a risk reduction coefficient (effectiveness), a cost (the cost of implementing it) and an indirect cost (negative cost, e.g. how much it may degrade productivity)
- the games are over probabilistic attack graphs (attack graph = threat model)

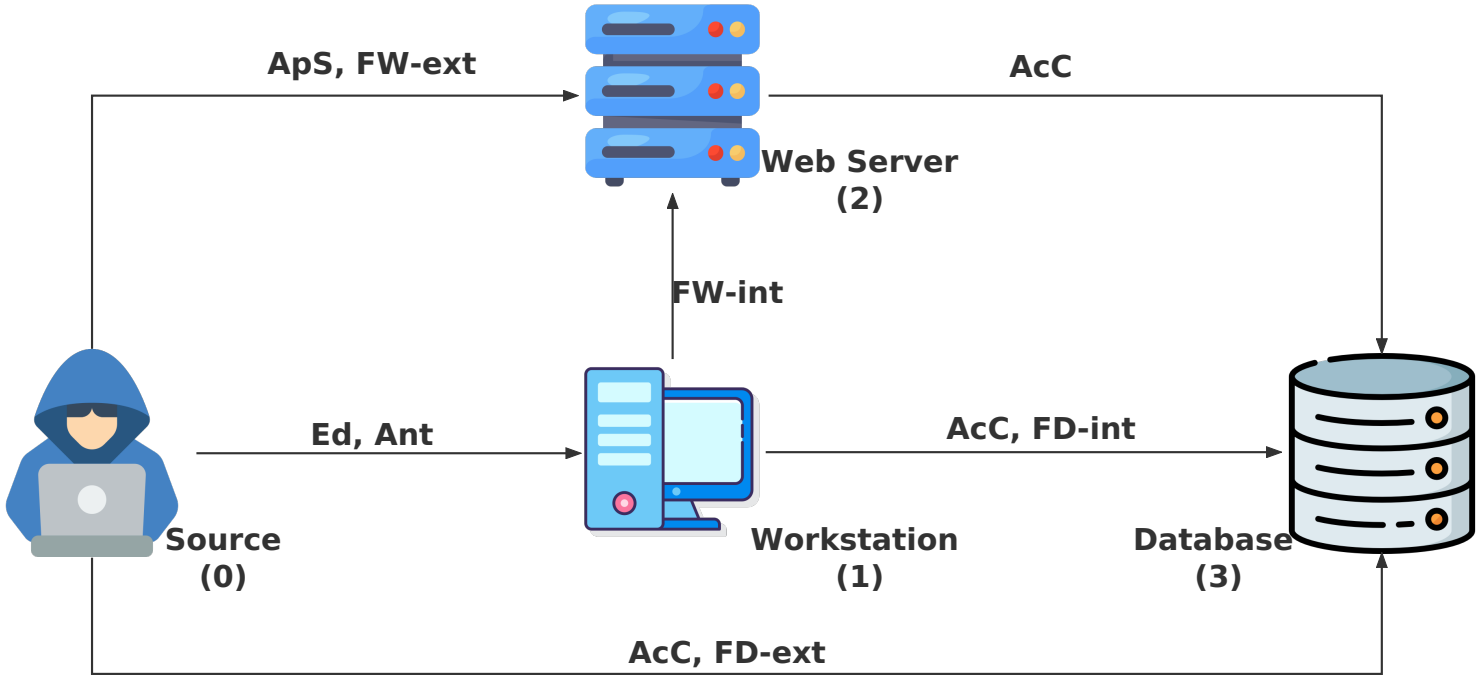


Figure: A simple attack probabilistic graph; it shows the attacker's possible steps (edges) to reach the target (database) and the defender possible controls (labels on the edges, e.g. FW-ext = external firewall)

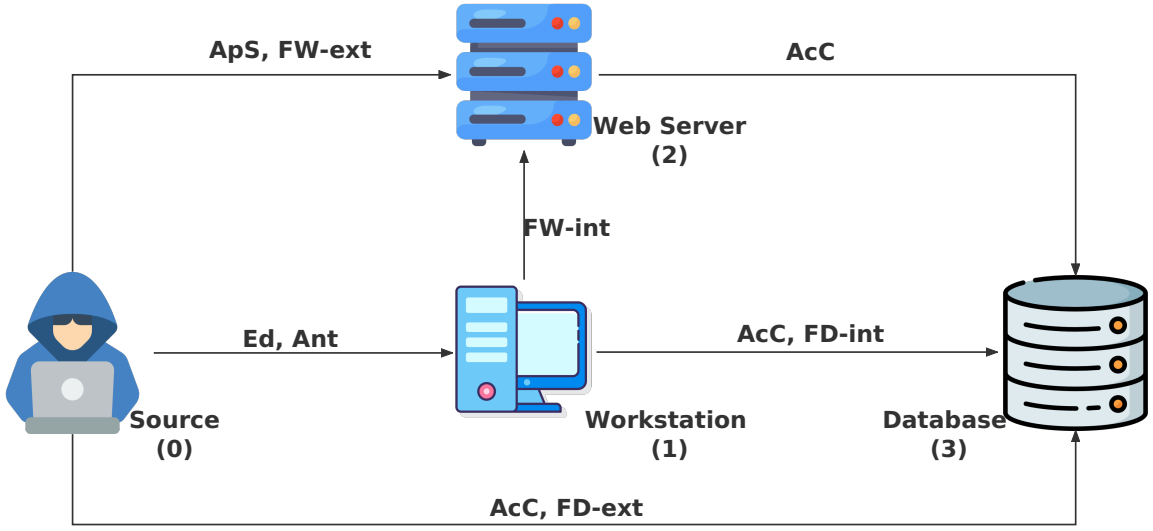


Figure: Controls: *Ed*=user education, *Ant* = anti-malware, *ApS* = application isolation, *AcC* = access control. *FW-int*, *FW-ext*, *FD-int*, *FD-ext* = internal, external firewalls for the web server and the database. Effectiveness of controls: Ed = L, Ant = M, ApS = M, AcC = H, FW-int = M, FW-ext = M, FD-int = M, and FD- ext = H. (L = 0.7, M = 0.5, and H = 0.2.)

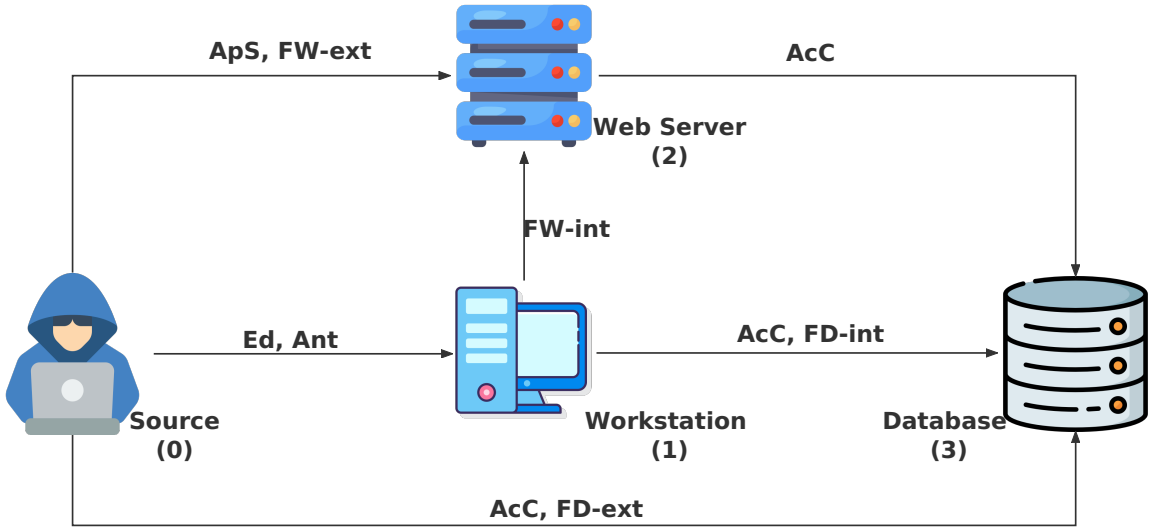


Figure: Suppose all controls have cost 1 and the budget is 8, then the game solution is $[Ed, Ant, ApS, AcC, FW\text{-}ext, FD\text{-}ext]$ which minimises the risk to 0.0125 corresponding to the path $0 \to 2 \to 3$. **The game solution is under budget: we could still invest more!**

- Investing beyond the optimal solution doesn't help against a rational attacker, but humans are often irrational.
- Game theoretical experiments (e.g. in the game of poker and in security games) have shown that when optimal strategies are played against humans, humans typically respond suboptimally (bounded rationality).
- so it makes sense to extend the optimal game solution, in a principled way: protect from the most rational attacker, then from the second most rational, then ... until budget exhausted or controls exhausted

possible principled options

- parallel: change the game solver to invest all budget in one optimization step
- iterative 1: solve the game and after the solution remove the "weakest path" (= rational attacker's choice); then solve the modified game and so on
- iterative 2: solve the game and after the solution add blocking constraints on edges of the "weakest path"; then solve the modified game and so on

possible principled options

- the parallel solution seems the most appealing but: 1) it may not include the optimal solution so it is not an extension of the game solution and 2) it may not be possible to set it as an optimization
- iterative 1: too restrictive (easy to find examples were good suboptimal solutions are excluded)
- iterative 2: this (we argue) is the most principled, and is the one developed in the paper

high-level description

the iterative solution (N-Solution) can be described as follows[1]:

1. solve the game and add to the solution set the controls in this game solution,
2. eliminate from the set of controls the controls (up to the levels) selected in the game solution,
3. add constraints forcing the attacker to use at least one new edge wrt the previous attacks,
4. if remaining budget>0 and |controls| >0 go to 1 else return solution

[1] some hand-waving here to avoid technicalities

- The game solver used in step 1 was introduced in M. Khouzani, Z. Liu, and P. Malacaria, “Scalable min-max multi- objective cyber-security optimisation over probabilistic attack graphs” (European Journal of Operational Research EJOR 2019)
- That solver uses properties of total unimodular matrices to achieve exact LP relaxation and dualisation, This result in a very efficient MILP solver, e.g. it returns the optimal solution for attack graphs with 20,000 nodes in less than four minutes typically.
- as paths can be exponential in the number of nodes these are huge games!!

- An important thing to prove is that extending the (EJOR) solver with the constraints introduced by the iterative algorithm doesn't brake the property of total unimodularity
- This is proven in the paper.
- hence an efficient and exact duality solution (using KKT conditions) can be derived for the iterative algorithm (some caveat here: at iteration N the number of constraints added is $\leq L^N$, L being the length of the longest attack path)

.

evaluation: a smart home case study

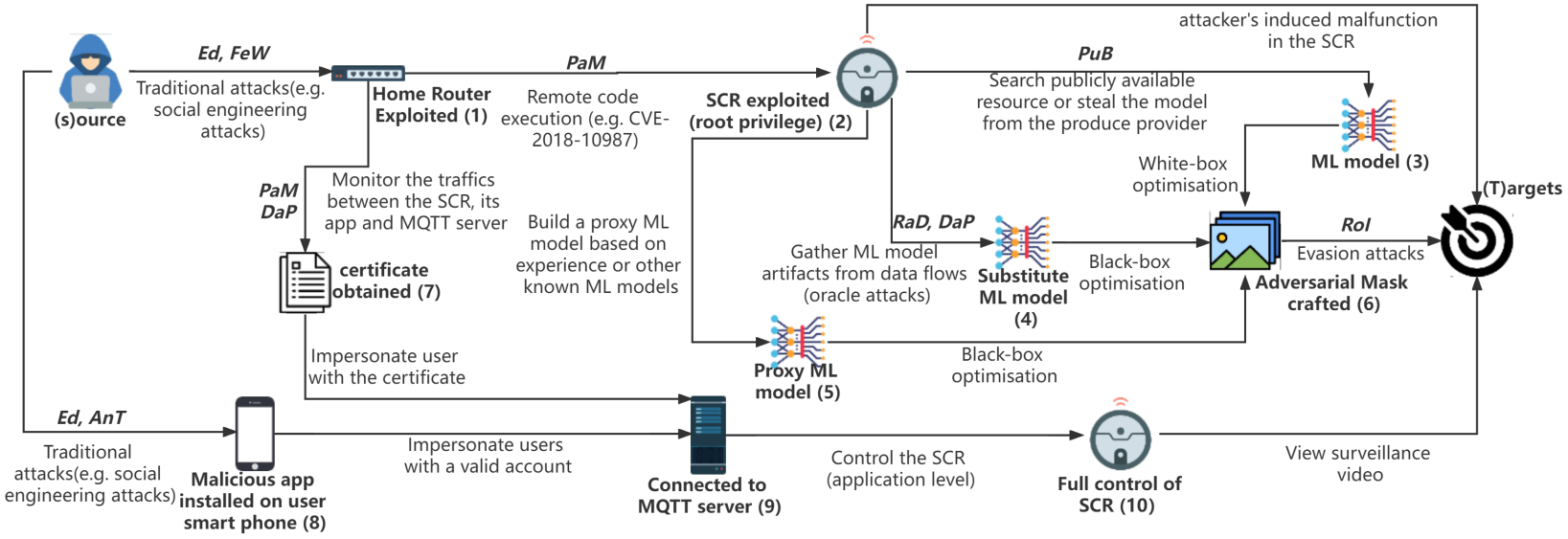


Figure: In this scenario the attacker's aims are to cause a smart cleaning robot (SCR) to malfunction or take control of the robot camera. Attacker may use conventional or AdvML attacks on IoT devices. Details of case study in the paper.

evaluation 1: a smart home case study

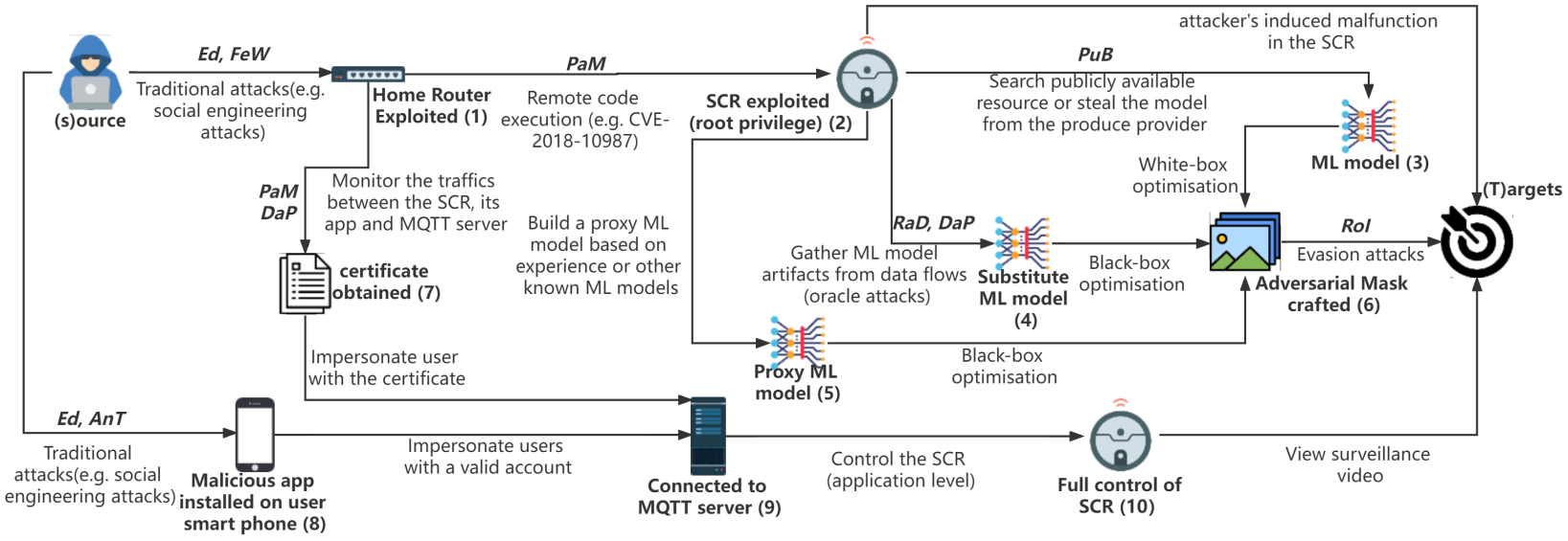
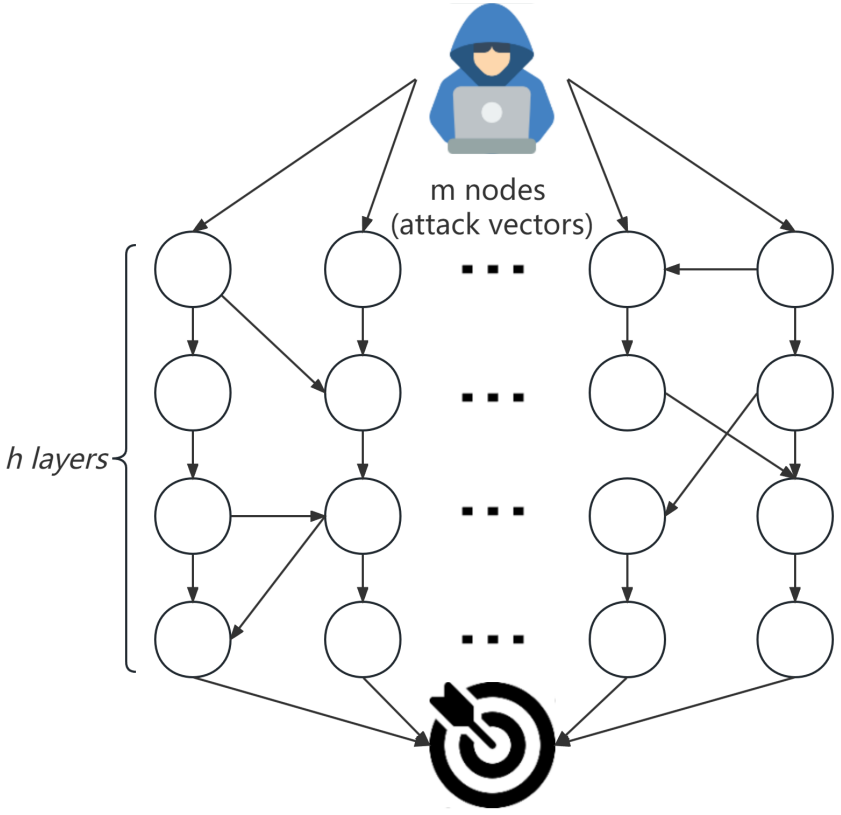


Figure: game solution=1-Solution= [Ed 2, FeW 2, PaM 1, AnT], 2-Sol: replace PaM 1 with PaM 2, 3-Sol: add DaP, 4-Sol: add RoI. Total=4-Solutions=[Ed 2, FeW 2, PaM 2, AnT, DaP, RoI]

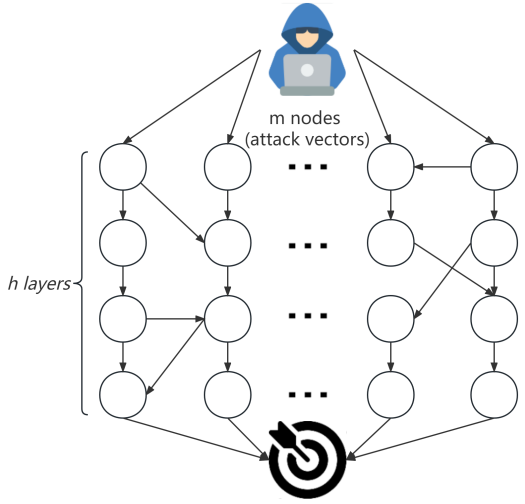
Evaluation 2: scalability

- to illustrate the scalability of the N-Solutions we use random attack graphs with a similar topology to real attack graphs. The topology is inspired by the ATT&CK Enterprise Matrix, where multi-stage attacks have 14 layers (from Reconnaissance to Impact).



Evaluation: scalability

- the (1,2,3),4-Solutions algorithm is run on random graphs with up to 1250 nodes. Solutions are found within reasonable time (a few minutes at most)
- Experiments show that the number of layers, more than the number of nodes is the factor affecting performance (details in the paper)



Conclusions

Keep spending:

- if the game solution is under budget, it makes sense to invest beyond the optimal solution up to budget exhaustion
- the optimal solution can be augmented in a way that improve defence against suboptimal attackers
- as humans are suboptimal this augmented defence is an improvement wrt real world attackers
- the algorithm presented here is principed i.e. at each round it deals with the most rational attacker not dealt with in the previous rounds
- the algorithm, thanks to total unimodularity and exact duality, is efficient.