SAND2013-3734P

Optimal Interdiction of Attack Plans

Joshua Letchford
Duke UNIVERSITY

Yevgeniy Vorobeychik
Sandia National Laboratories

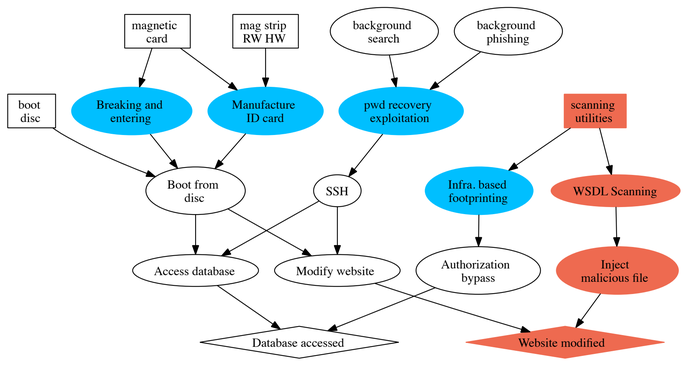
Framework

We propose a Stackelberg game model of security in which the defender chooses a mitigation strategy that interdicts potential attack actions, and the attacker responds by computing an optimal attack plan that circumvents the deployed mitigations. Our starting point is actually an extension to the classical planning framework termed *partial satisfaction planning* (PSP), which assigns each goal literal a value, and each plan action a cost.

Cyber Security Example

Initial state: initial attacker capabilities (possession of a boot disk and port scanning utilities)

Actions: physical actions (breaking and entering and booting a machine from disk) and cyber actions(performing a port scan to find vulnerabilities)



Blue: Defended actions, Red: Attacker's plan

IP for Classical Planning

The problem of finding an optimal plan for PSP given a fixed number of time-steps has a known integer programming (IP) formulation which will provide the basis for our own techniques. The planning IP introduces a number of meta-variables to capture how actions modify the state of each literal l at every time step t. As an example, computing $x_{l,t}^{pa}$ (which is 1 iff an action is executed in timestep t that has l as a precondition but does not delete it) requires the following constraints:

$$\forall_{l,t} \sum_{a \in pre_l \setminus del_l} y_{a,t} \geq x_{l,t}^{pa}$$
$$\forall_{l,t,a \in pre_l \setminus del_l} y_{a,t} \leq x_{l,t}^{pa}$$

where pre_l represents the set of actions which have as l a pre-condition and del_l the set of actions where l is deleted as a post-condition. Each of the other meta-variables is computed with a similar set of constraints.

$$\max_{y_{a,t}} \sum_l V_l s_l - \sum_{a,t} C_a y_{a,t}$$
s.t. :
constraints from metavariables
$$\forall_{l,t}\ x_{l,t}^{pa} + x_{l,t}^{m} + x_{l,t}^{pd} \leq x_{l,t-1}^{add} + x_{l,t-1}^{pa} + x_{l,t-1}^{m}$$
$$\forall_{l,t}\ x_{l,t}^{pa} + x_{l,t}^{m} + x_{l,t}^{pd} \geq x_{l,t-1}^{add}$$
$$\forall_{l,t}\ x_{l,t}^{pa} + x_{l,t}^{m} + x_{l,t}^{pd} \geq x_{l,t-1}^{pa}$$
$$\forall_{l,t}\ x_{l,t}^{pa} + x_{l,t}^{m} + x_{l,t}^{pd} \geq x_{l,t-1}^{m}$$
$$\forall_{l}\ x_{l,|T|}^{add} + x_{l,|T|}^{pa} + x_{l,|T|}^{m} \geq s_l$$
$$\forall_{l}\ x_{l,|T|}^{add} \leq s_l$$
$$\forall_{l}\ x_{l,|T|}^{pa} \leq s_l$$
$$\forall_{l}\ x_{l,|T|}^{m} \leq s_l$$
$$\forall_{l}\ x_{l,0}^{add} = \begin{cases} 1 \text{ if } l \in I \\ 0 \text{ otherwise} \end{cases}$$

Deterministic Plan Interdiction

A *deterministic plan interdiction problem (DPIP)* is described by a tuple $\{P, M, V_l^D, V_l^A, C_m^D, C_a^A\}$, where:

- P is the planning problem for the attack in the absence of any mitigations
- M is the set of mitigation strategies for the defender
- V_l^D and V_l^A are the utilities of the defender and attacker respectively when the attacker achieves a goal literal $l \in G$
- C_m^D is the cost of mitigation $m \in M$ to the defender
- C_a^A is the cost of action $a \in A$ to the attacker

While a mitigation strategy $m \in M$ can potentially protect against a subset of attack actions a or remove literals from the initial state I, without loss of generality, we assume that m only has an effect on attacker actions. Although we prove this problem to be $PSPACE$-Complete, when we restrict ourselves to a fixed number of time-steps we can formulate $DPIP$ as a (very large) integer program.

IP for DPIP

$$\max_{D_a, D_m, y_{a,t}, \delta_p} \sum_{l \in L} V_l^D s_l - \sum_{m \in M} D_m C_m^D$$
s.t. :
$$\forall_a\ D_a \leq \sum_m D_m A_{m,a}$$
$$\forall_{m,a}\ D_a \geq D_m A_{m,a}$$
$$\forall_{a,t}\ y_{a,t} \leq (1 - D_a)$$
$$\forall_{p,a}\ \delta_p \geq D_a$$
$$\forall_p\ \delta_p \leq \sum_{a \in p} D_a$$
$$\forall_p \sum_{l \in L} V_l^A s_l - \sum_{a,t} C_a^A y_{a,t} \geq U^A(p) - Z\delta_p$$
constraints from Planning IP

Scaling Up with Constraint Generation

The number of feasible plans, and consequently the number of constraints are exponential in the number of actions. To manage this problem we develop several constraint generation approaches. Below is one algorithm for doing this, where $\hat{\mathcal{P}}$ corresponds to the subset of plans we are currently considering. We can use the IP for Classical Planning as an oracle to generate the plans we optimize against.

Algorithm for Constraint Generation

$\hat{\mathcal{P}} = \emptyset$;
$\hat{U} = 0$;
$U = \infty$;
while $\hat{U} < U$ **do**
  $(\hat{M}, D_a, \hat{U}) = DPIP\_MASTER(\hat{\mathcal{P}})$;
  $A_{\hat{M}} = \emptyset$;
  for $a \in A$ **do**
    if $D_a = 0$ **then**
      $A_{\hat{M}} = A_{\hat{M}} \cup a$;
    end
  end
  $(p, U) =$optimalPlan$(A_{\hat{M}})$;
  if $U > \hat{U}$ **then**
    $\hat{\mathcal{P}} = \hat{\mathcal{P}} \cup p$;
  end
end

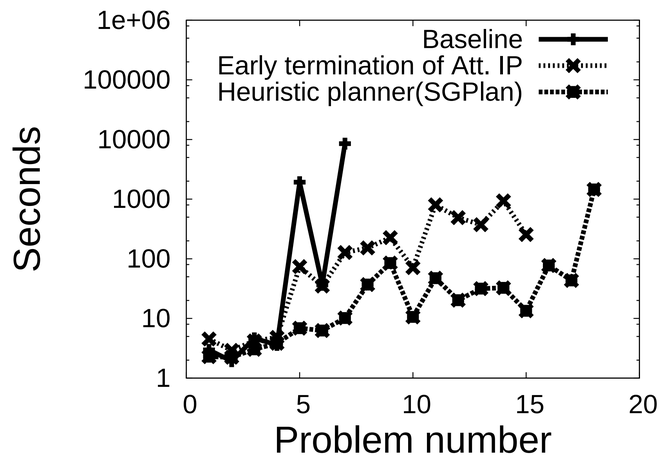
Optimistic Constraint Generation

(3rd Col) Observe that to make progress in each iteration of Algorithm 1 we need only to generate a plan with a higher utility than any in $\hat{P}$, and not necessarily an optimal plan. We consider two natural candidates for such optimistic constraint generation: the planning IP (above) but to cut off the solver after a fixed time limit and using an off-the-shelf state-of-the-art heuristic planner (SGPLAN).

Experiments

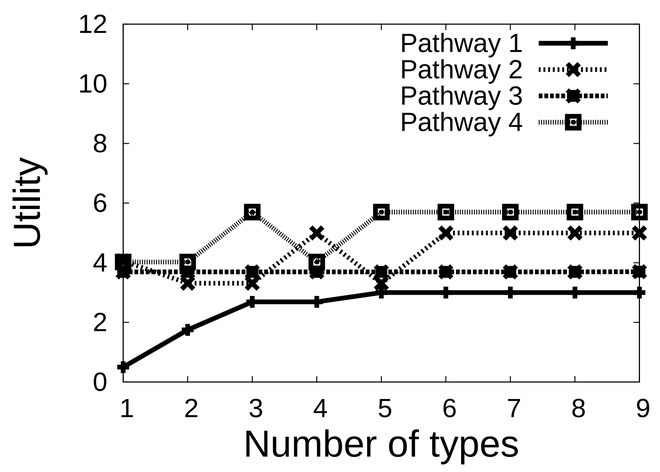
For the experimental evaluation, we used the pathways planning domain from the 2006 international planning competition (IPC). To formulate these as interdiction problems, we let the set of mitigations correspond to the set of plan actions (thus, each mitigation m blocks exactly one attack action). The problems that we ran our experiments on ranged in size from 46 possible attacker actions and one potential goal (problem number 1) to 490 attacker actions and 27 potential goals (problem number 20), with a general trend of a larger index having a larger number of possible attacker actions and goals.

Effect of Optimistic Constraint Generation on Runtime



Adding Uncertainty

We can relax the assumption that the attackers capabilities, goals, and action costs as known to the defender in a relatively standard way by using a Bayesian Stackelberg game model. Below we focus on uncertainty in attackers capabilities (uncertainty about goals and costs can be handled similarly). For the figure below, we first generated 9 attacker types corresponding to initial capabilities, and then incrementally abstracted these to obtain smaller numbers of types. We discuss other types of uncertainty in the paper.



Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND number: 2012-6612 P.