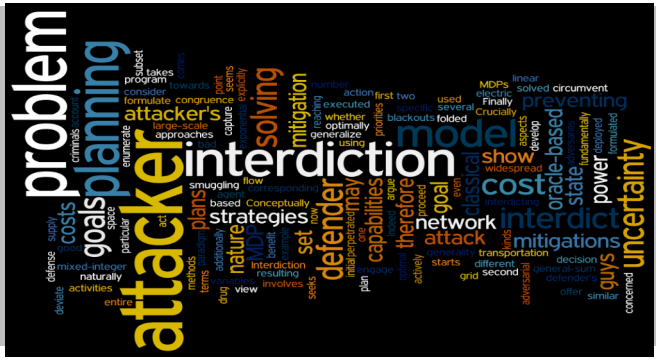
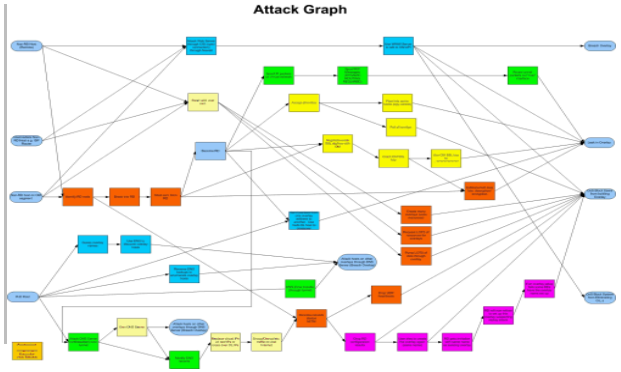
SAND2013-0538P

Exceptional service in the national interest

Sandia National Laboratories





Cyber Games

Yevgeniy Vorobeychik

Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY | NNSA Nuclear National Security Administration

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 NO. 2012-10175C

The Cyber threat

Sandia National Laboratories

- *Recent DDOS attacks on US banks (from the Middle East, according to media)*
  - *Unprecedented in scale/speed*
  - *Using "weak servers" as launch pads (weak = poor security, bad passwords, etc)*
- *Shamoon attack on Saudi state oil company*
  - *wiped 30,000 computers*
  - *the most destructive attack on the private sector to date*
- *Stuxnet/conficker*
  - *sophisticated combination of cyber attack tactics, including zero-days*
  - *Stunxnet: cyber-physical attack targeting critical infrastructure*
- *Corporate America loses between $100 billion to $1 trillion / year from theft of proprietary information (The Economist, Babbage blog, May 11, 2012)*
- *Advanced persistent threats: attackers are highly sophisticated, often have (direct or tacit) support of a state, very highly rewarded for success, and pay attention to what we are doing!*
  - *By focusing on responding only to current threats/attacks/vulnerabilities, we just weed out the incompetent attackers*

Conficker vs. The World

Sandia National Laboratories

- Conficker:
  - an infected machine generates 250 random domain names from 5 TLDs each day, and if one resolves, it is used to download the payload (seeded with date to synchronize)
- The World:
  - block all 250 domain names each day to counteract this
  - anti-virus signature, windows update
- Conficker:
  - increase number of TLDs to 8
  - blocks auto-update
  - attaches to removable media
- The World:
  - Conficker working group continues to shut down the generated domain names
- Conficker:
  - generate 50,000 random domain names from 8 TLDs
  - disables safe mode; kills anti-malware products

Security as Chess

Adversarial: only one winner ("zero-sum")

Opponents must anticipate each other's moves

- *strategy should account for how the opponent will respond*

Extremely complex: impossible to enumerate the set of all states in the game

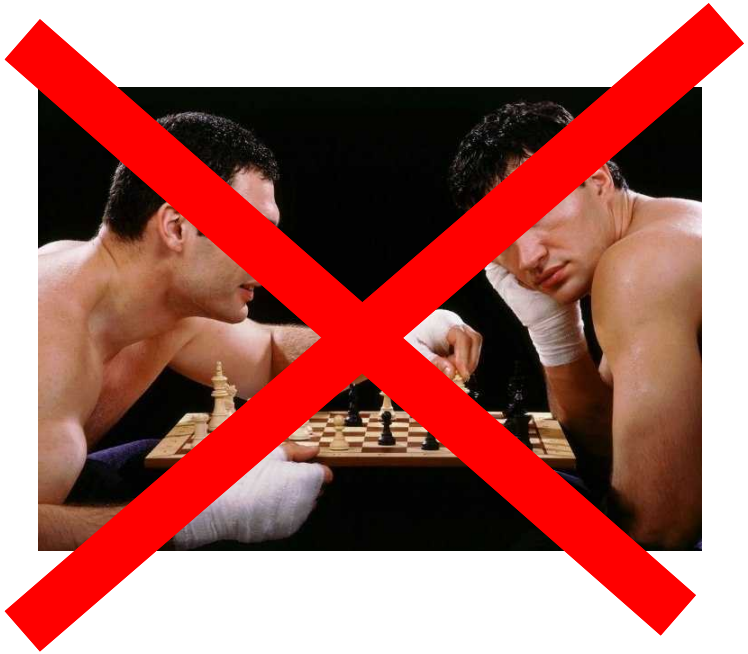
- *difficult to rule out vulnerabilities in a given position*
- *player strategies highly sophisticated*

Cyber as a Game

Sandia National Laboratories

- *Attackers* aim to do damage, deny service, exflitrate information, etc
- *Defenders* wish to protect their systems and information
- *Attackers act to promote their interests/goals, using information about defense countermeasures*

How is Cyber NOT like Chess?



Chess has *complete information*:
- well-known rules
- clear what the opponent's objective is
  - *Cyber*: attacker' specific objectives are uncertain

Chess has *perfect information*:
- can observe all previous moves
  - *Cyber*: attacker's can be difficult to detect

Chess has *no uncertainty* of any kind:
- consequences of moves are deterministic
  - *Cyber*: systems complex, non-deterministic

Chess has *only two players*
- *Cyber*: many players

Salient Issues in Cyber Games

Sandia National Laboratories

Adversarial: (like chess) but (often) with more than two players

- many defenders
- many (potential) attackers
- not necessarilty “zero-sum” (attackers and defenders have different goals/costs)

Goals and capabilities unknown

- even if we pin down an attacker, uncertain about their capabilities and what they are ultimately after

Strategic landscape extremely complex

- attacks can exploit known and zero-day vulnerabilities
- exploits can be chained together in complex ways to achieve goals

Lots of noise in the system

- effects of attacks/mitigations non-deterministic (e.g., system can go down, software upgraded, etc)

Systems are highly interdependent

- different system components have complex dependencies (e.g., supply chain dependencies)

Game Theoretic Model

Sandia National Laboratories

- *Defenders move first*:
  - *jointly choose defense policies (possibly randomized)*
- *Attackers learn about the defenders' decisions*
  - *respond by choosing optimal attack policies*

- Notes:
  - I will assume that only one attacker attacks at any given time (simultaneous attacks are rare)
  - Uncertain about *which* attacker will attack

Outline

- Complexity of Attack Space
  - Model attacker as a planning agent
  - Optimal plan interdiction problem
- Uncertainty about attackers (capabilities, goals); noise
  - Bayesian plan interdiction problem
  - Interdicting an attacker MDP
- Interdependencies among valuable assets (e.g., critical infrastructure sectors, supply chains)
  - Blending game theory, network science, and OR
- Decentralization in defense (i.e., many defenders)
  - Blending physics and game theory
  - Benefits and costs of decentralization

Sandia National Laboratories

OPTIMAL INTERDICTION OF ATTACK PLANS

AAMAS, 2013, to appear

Threat/Adversary Modeling

Sandia National Laboratories

- Basic picture in threat modeling and mitigation:
  - *Attacker*:
    - has a set of capabilities
    - can choose from a collection of "actions" (activities, exploits, attacks)
      - actions can have costs (time, likelihood of detection/capture, etc)
    - has a collection of goals
      - goals may have different importance to the attacker/defender
      - may be content to achieve a subset of goals
    - constructs a plan (a sequence of actions) to achieve goals starting with capabilities
  - *Defender*:
    - system of interest has initial state (vulnerabilities, NW configuration, etc)
    - *mitigations*: can block attack actions, initial capabilities, patch vulnerabilities
      - mitigations can be costly

AI + Planning

Sandia National Laboratories

- One of the main branches of AI is (formal/logic-based) planning
  - Model the planning problem using formal logic
  - Numerous heuristic planning tools and some well-accepted planning languages (STRIPS, PDDL)
- (*Deterministic/Classical*) Planning problem:
  - The world = a set of logical variables = *state* (of the world)
  - Start: initial state of the world (variables that are **true** at time 0)
  - Goal(s): variables (or boolean expressions) that the planner wants to satisfy
  - Actions: actions/steps a planner can take towards the goal
    - preconditions: variables that must be true for the action to apply
    - effects: variables that become true/false as a result of the action
  - A plan: a partial order of actions that achieve goals starting from the initial state

Attacker as a Planner

Sandia National Laboratories

- Initial state = attacker capabilities (expressed as logical variables), system vulnerabilities, initial NW config, etc
- Goals = attacker's goals (expressed as logical variables)
  - Data X exfiltrated
  - Root access to machine Y
  - Different goal variables may vary in importance
- Actions = attack actions

Example: Initial capabilities

Sandia National Laboratories

initial attacker capabilities

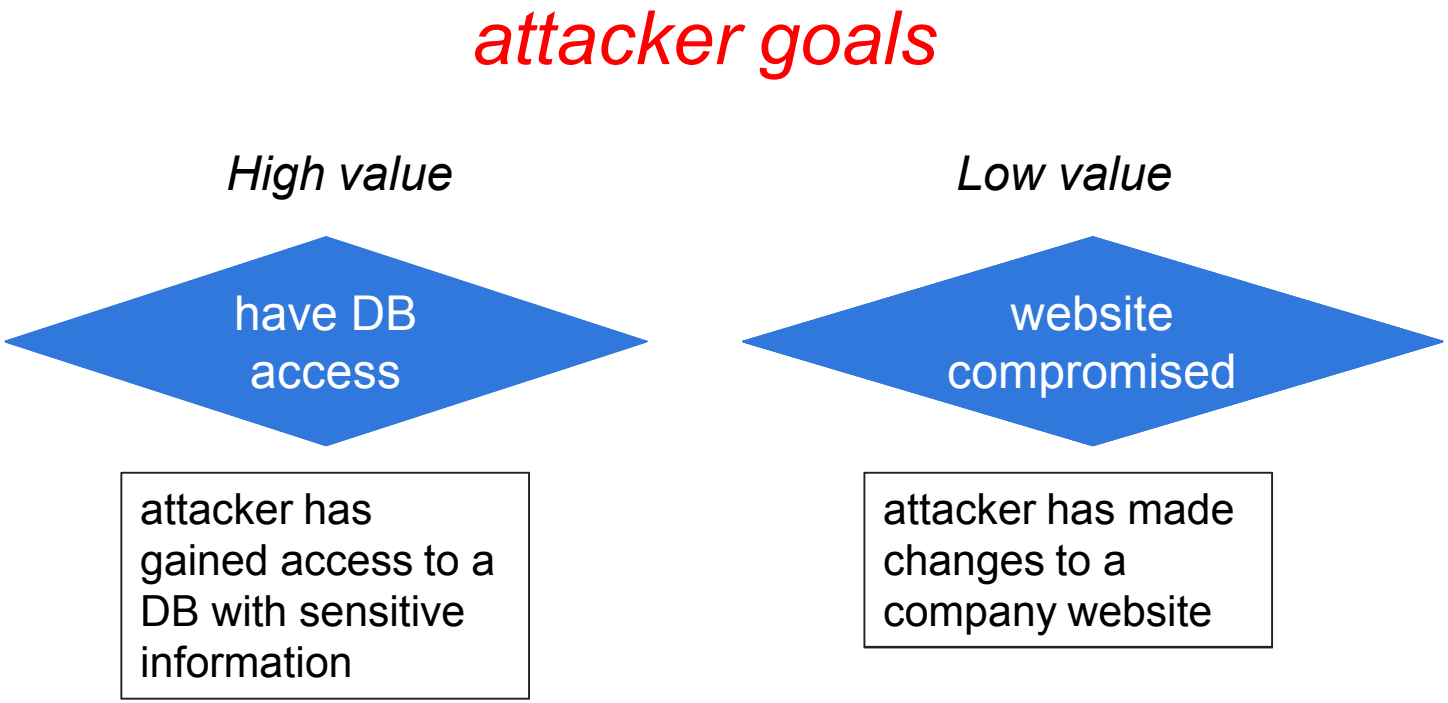
magnetic card

attacker has access to employee's magnetic card

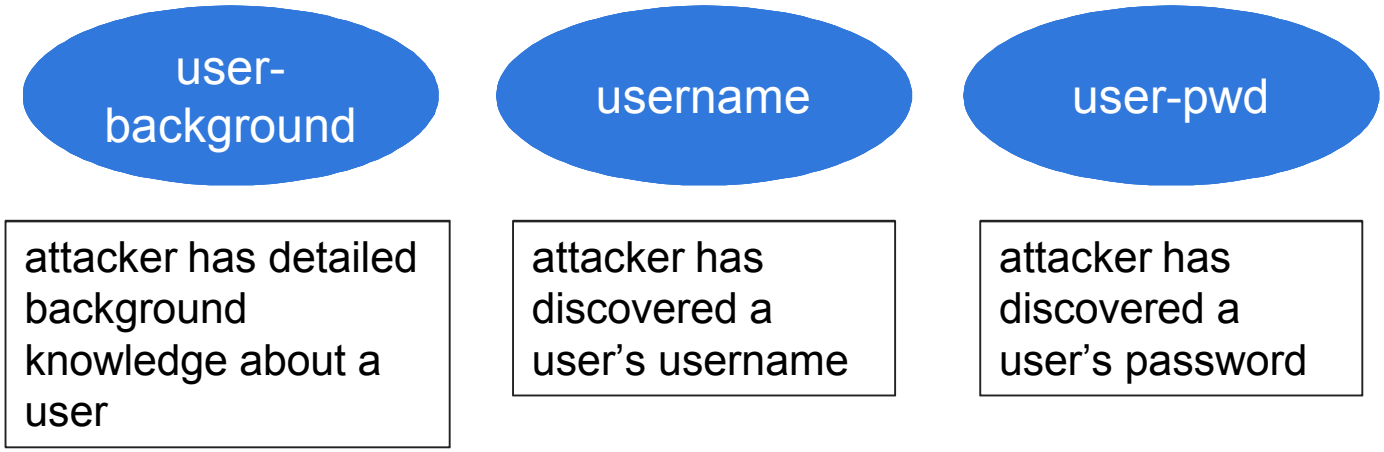
boot disk

attacker has a boot disk

Example: Attacker Goals



Example: State Variables



Example: Attack Action

Sandia National Laboratories

preconditions *actions* *add effects*

Common vulnerabilities & exploits: http://capec.mitre.org

A Few Words on Attack Graphs

Sandia National Laboratories

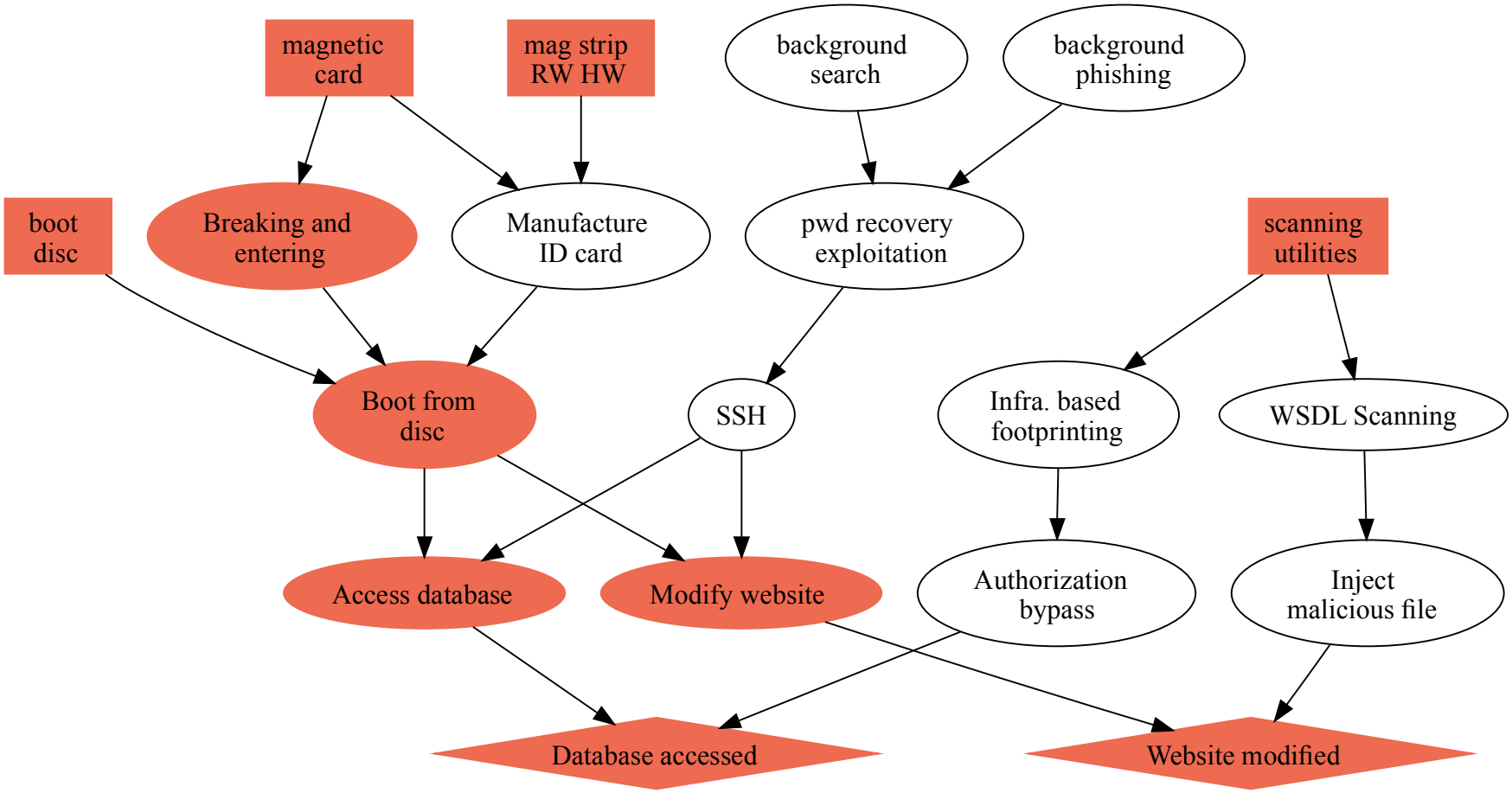
- A common way attacker decision process is modeled in practice (e.g., red teaming) is by using *attack graphs* and/or *scenarios*
- Two main types of attack graphs:
  - "State" Attack Graph
    - Nodes = states of the world (e.g., assignments of values to all state variables)
    - Edge = possible state transition (i.e., there exists an attack action that transitions from one state to another)
  - "Action" Attack Graph
    - Nodes = attack actions
    - Edge = action dependencies (i.e., if an action satisfies a precondition of another action; does not naturally represent *negative dependencies*, that is, *when an action deletes a precondition of another*)

… and Scenarios

Sandia National Laboratories

- When attack graphs are too unwieldy, experts attempt to generate a set of plausible attack *scenarios*
- A scenario is basically an attack plan:
  - A sequence of actions that impacts a particular target
  - For example: starting with some set of capabilities, how an attacker can exfiltrate Data X
- *Automated attack planning can be viewed as automated scenario generation*
  - *Note: must be able to generate multiple plans/scenarios*

("Action") Attack Graph, and an Optimal Plan for an Example Scenario

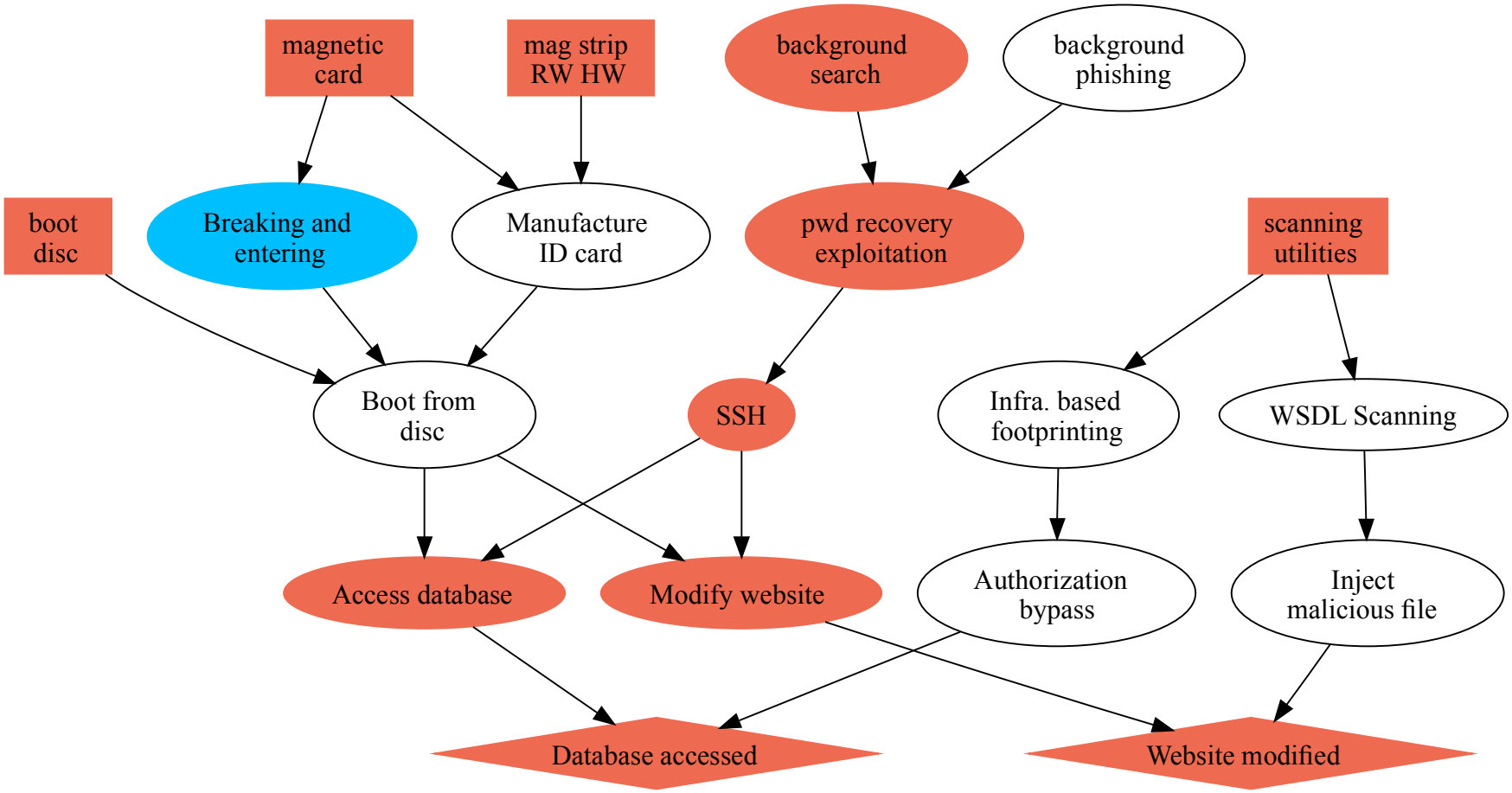


Attack Graphs and Scenarios

Sandia National Laboratories

- Could conceivably identify all possible paths to goals
- Does not scale (particularly "state"-graphs, which are most widely used; 400 variables = 2^{400} states; age of universe, in seconds < 2^{30})
- "Action" graphs lose information
- In realistic settings, often deal with a small number of hand crafted scenarios
  - Mitigations aim to prevent the more likely/consequential of these scenarios
  - *do not account for attacker circumventing the mitigations*

Scenario interdiction and circumvention



Deterministic Plan Interdiction Problem

Sandia National Laboratories

DPIP: defender chooses an optimal subset of mitigations, accounting for attacker's best response plans

(optimal: maximizing defender's utility)

(utility = Value of goals – cost of mitigations)

Deterministic Plan Interdiction (DPIP)

Sandia National Laboratories

- DPIP_DP (deterministic plan interdiction decision problem): can the defender achieve target utility?
- Theorem: DPIP_DP is PSPACE-Complete
- Proof: by reduction from partial satisfaction planning
- Formulation: based on an Integer Program for computing an optimal partial satisfaction plan (Vossen et al., 1999; Briel et al., 2004)
  - Maximize defender utility
  - subject to:
    - a plan is feasible
    - a plan is the best plan for the attacker (*by comparison to all feasible plans*)

DPIP formulation

$$\max_{D_a, D_m, y_{a,t}, \delta_p} \quad \sum_{l \in L} V_l^D s_l - \sum_{m \in M} D_m C_m^D$$

s.t. :

$$\forall_a \quad D_a \leq \sum_m D_m A_{m,a}$$

$$\forall_{m,a} \quad D_a \geq D_m A_{m,a}$$

compute which actions are interdicted

$$\forall_{a,t} \quad y_{a,t} \leq (1 - D_a)$$

attacker cannot choose interdicted actions

$$\forall_{p,a} \quad \delta_p \geq D_a$$

$$\forall_p \quad \delta_p \leq \sum_{a \in p} D_a$$

identify non-interdicted plans

$$\forall_p \quad \sum_{l \in L} V_l^A s_l - \sum_{a,t} C_a^A y_{a,t} \geq U^A(p) - Z\delta_p$$

choose the best non-interdicted attack plan

plan feasibility constraints

DPIP (cont'd)

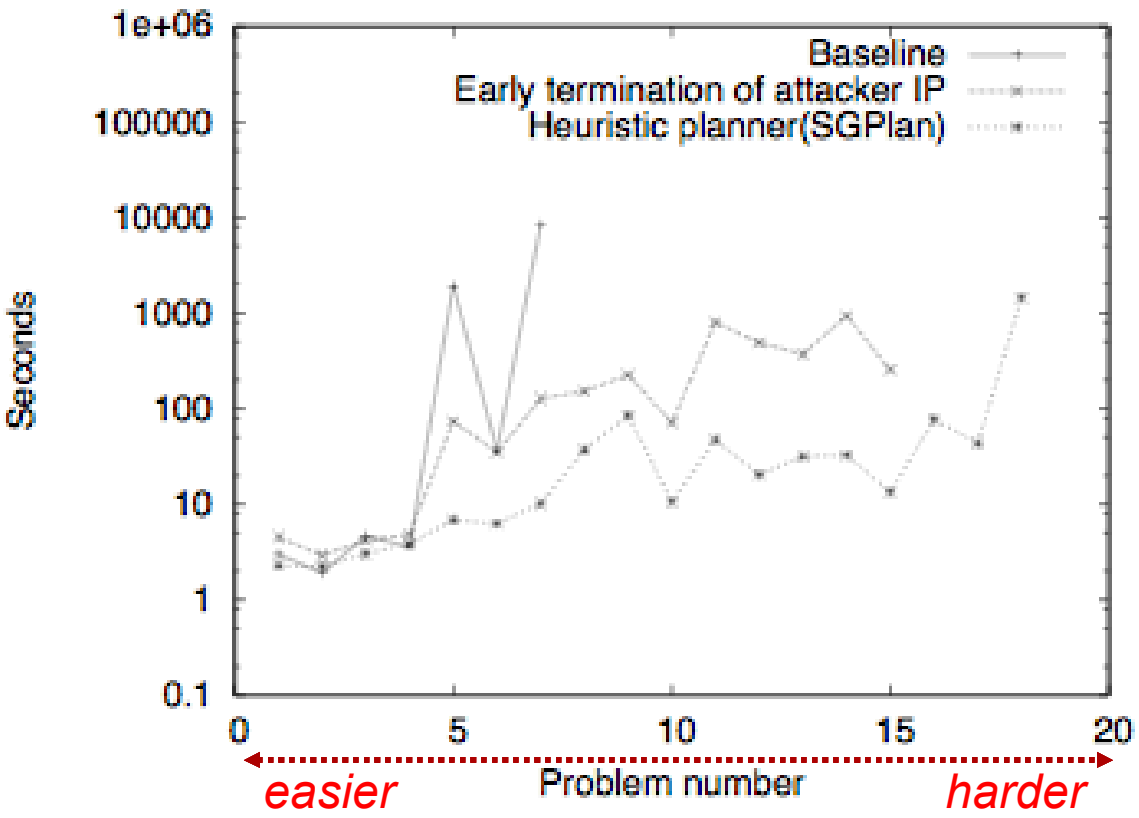
- IP: obviously too large (lots of feasible plans)
- Solution approach: constraint generation
  - We "know" how to compute an optimal plan
  - Iterate:
    - Compute optimal mitigations, given a subset of feasible plans
    - Compute a best response (plan) given mitigations
    - Repeat (until best response is not better than all plans already in the constraints)

DPIP: speeding up

Sandia National Laboratories

- Computing an *optimal* attack plan is quite hard
- Solution: use good heuristic planners in all but the last iteration
  - Only verify optimality at the very end
- Guaranteed to be optimal, but each iteration is much faster
- Heuristic planner candidates:
  - IP, with a time limit
  - SGPLAN5: state-of-the-art heuristic partial satisfaction planner; winner of IPC

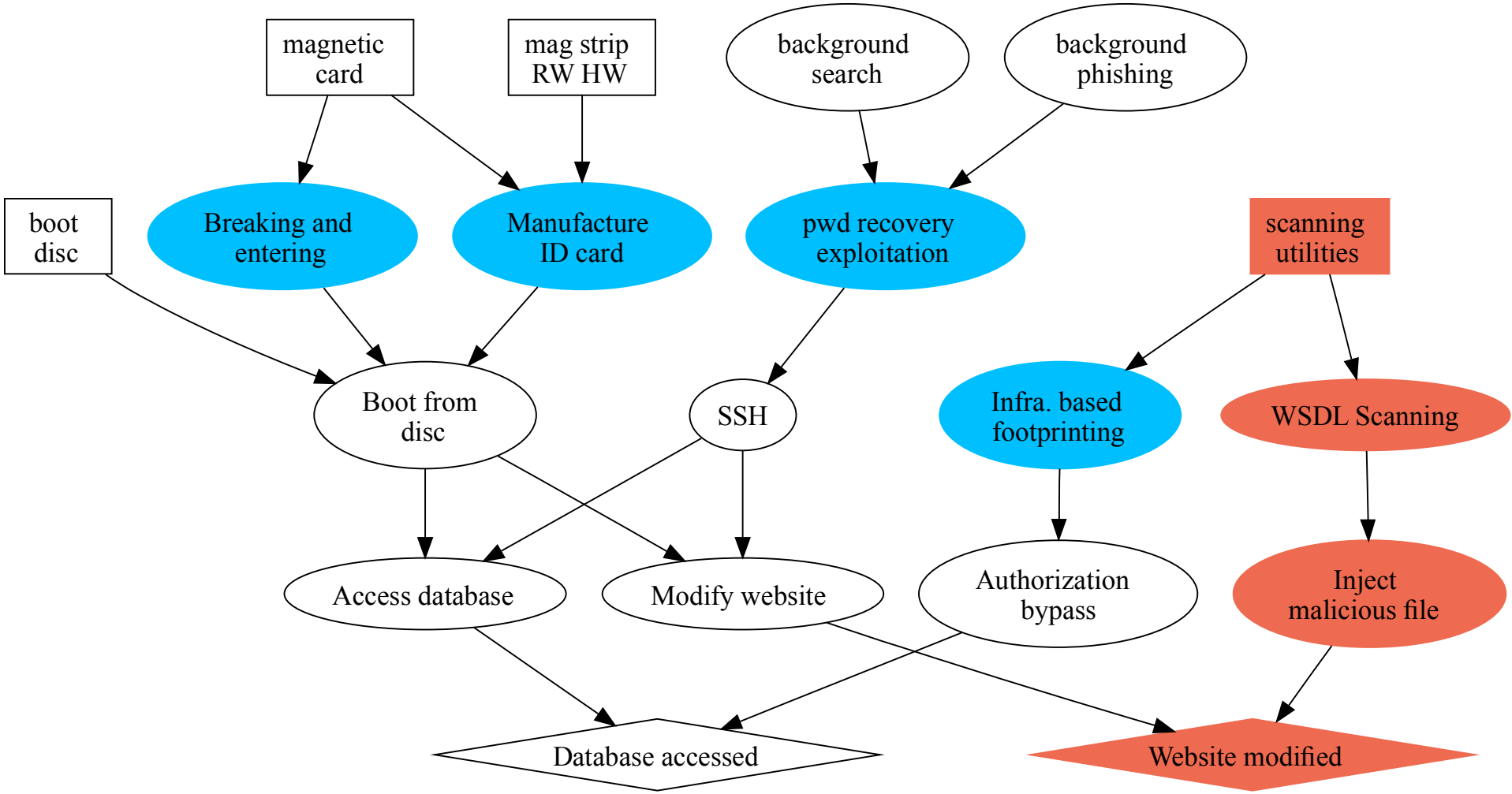
Experiments: DPIP Runtime



Problems from IPC 2005

- *Baseline*: Bender's with IP for optimal planning
- *Early termination*: time limit on the IP
- *SGPlan*: use heuristic planner to generate constraints
  - (check optimality at the end)
- *Orders of magnitude improvement in runtime over baseline*

Optimal Plan Interdiction Example



Sandia National Laboratories

UNCERTAINTY ABOUT ATTACKERS

What about uncertainty?

Sandia National Laboratories

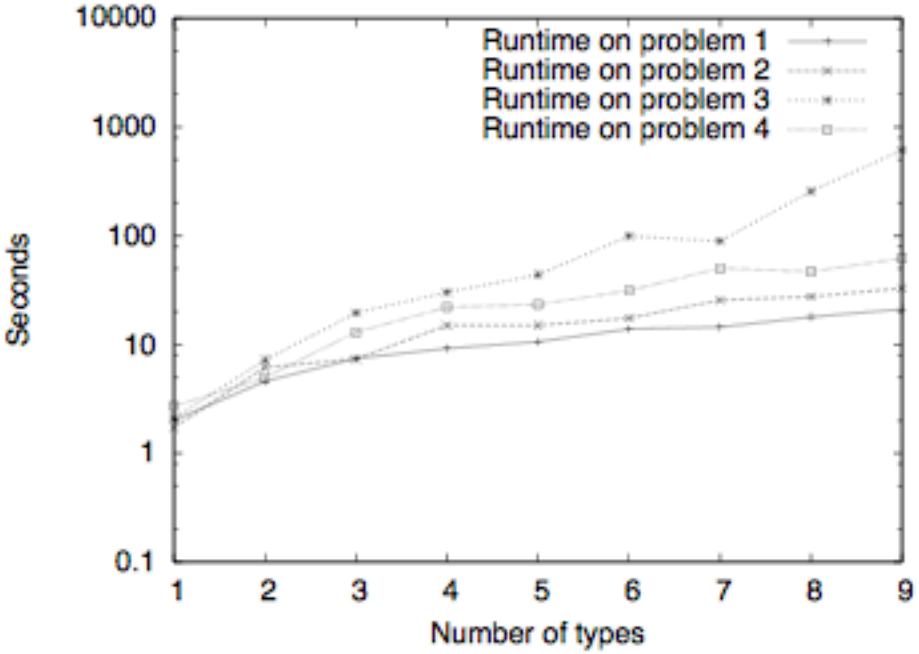
- DPIP assumes everything is certain and deterministic
- Usually uncertain about:
  - Attacker capabilities
  - Existence of vulnerabilities (of specific classes)
  - Attacker goals
  - Execution and effects of actions
  - Attacker's observation of the current state of the system
- Questions:
  - Can we extend DPIP to accommodate all/some of this uncertainty?
  - If not, how should we model it?

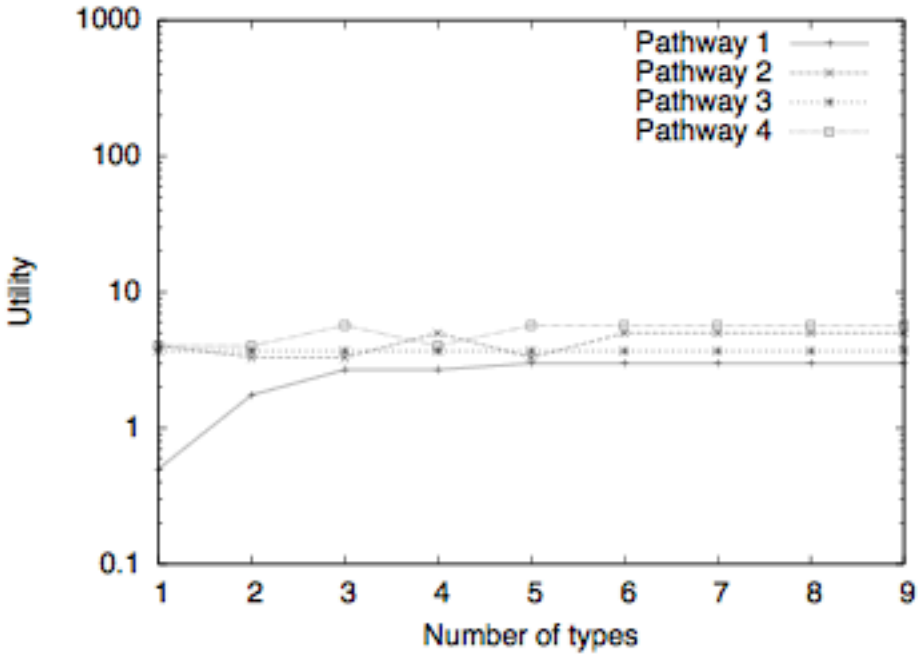
Uncertainty about capabilities/goals

- Can naturally model within the DPIP framework
  - Attacker has multiple types, each type corresponding to a set of capabilities/goals (*Bayesian plan interdiction problem; BPIP*)
  - Each type computes an optimal plan in response to mitigations
  - Objective now to maximize expected defender utility wrt distribution over attacker types

$$\sum_{\theta} \sum_{l \in L} p_{\theta} V^{D}_{\theta,l} s_{\theta,l} - \sum_{m \in M} D_m C^{D}_{m}$$

Experiments: BPIP



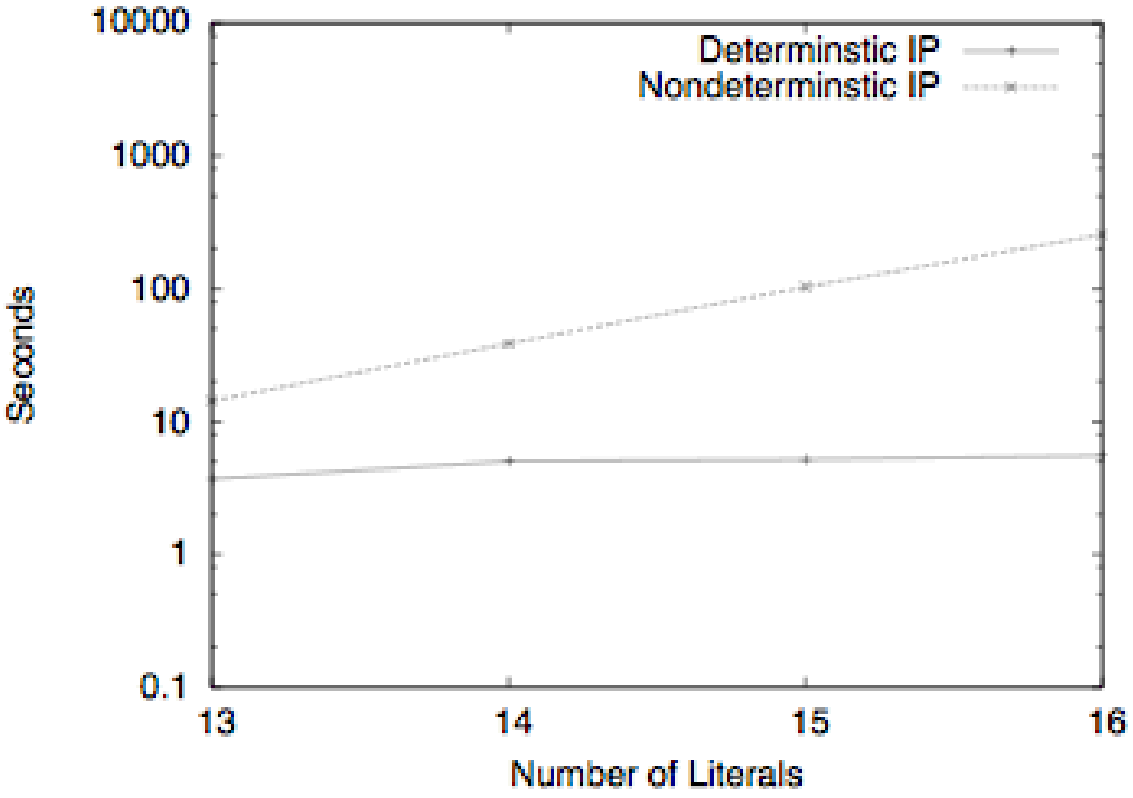


- Created an instance with 9 types, and aggregated (abstracted) types into sets to get fewer types
- Scales reasonably well, and only need relatively few types to get near-optimal solutions

Execution Uncertainty

- *Special case*: only uncertain about whether an action succeeds or not, success is observable, and action can be repeated
  - Can be done using DPIP framework by computing expected number of tries for each action
- *Generalizing*: state is observable, but action may have non-deterministic effects
  - Example: “port scan” action (info you get is uncertain)
  - MPD interdiction (MDPIP)
  - Leverage (dual) LP for computing optimal value in an MDP
  - State space explosion: must now explicitly represent all states in the system

Experiments with MDPIP



Can't scale beyond 15-20 literals

Much slower / poorer scalability than DPIP

But: far more general

Sandia National Laboratories

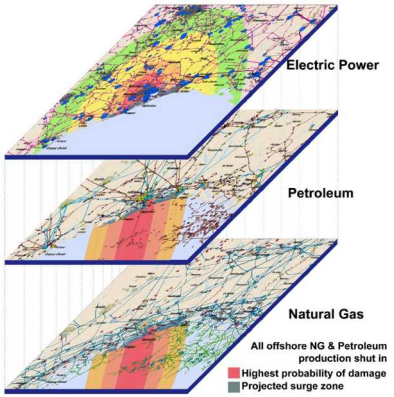
INTERDEPENDENT ASSETS

UAI, 2012; working paper

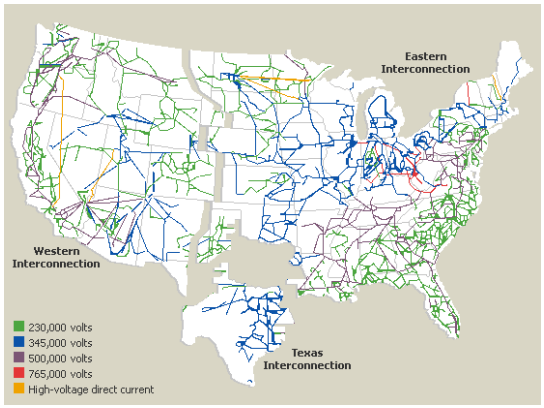
Complex interdependencies are all around us

Sandia National Laboratories

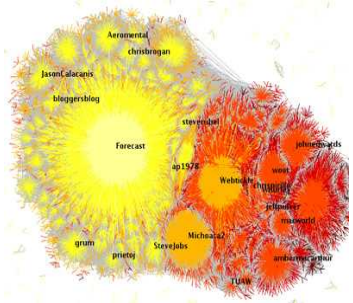
Critical Infrastructure



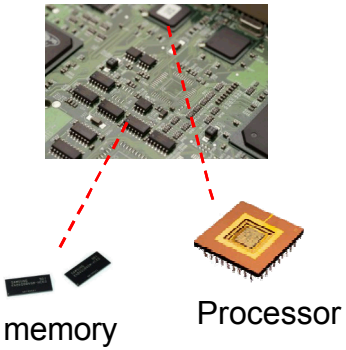
U.S. Power Grid

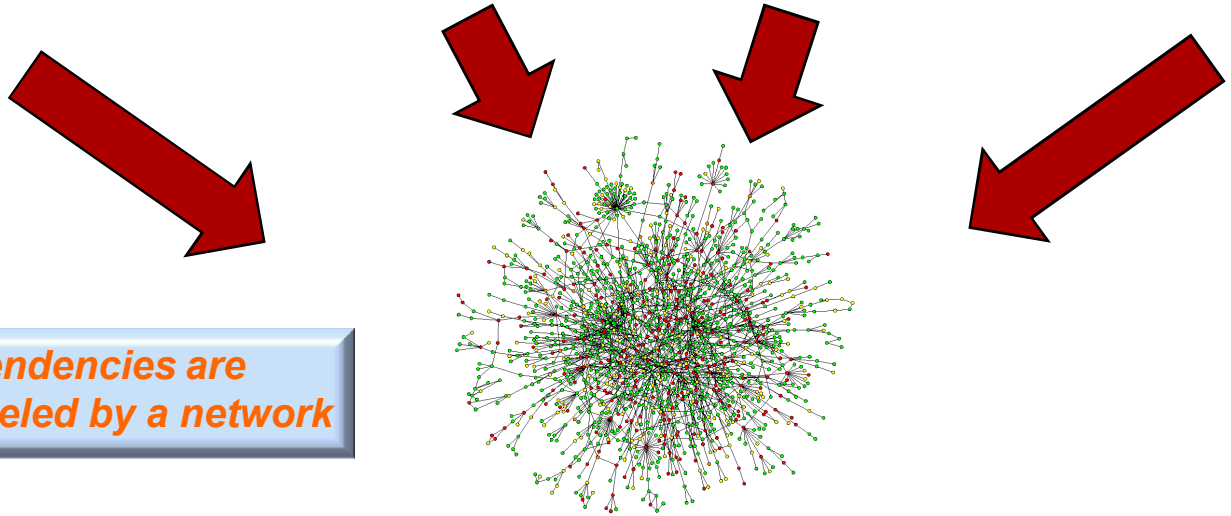


Social Networks



Supply chains

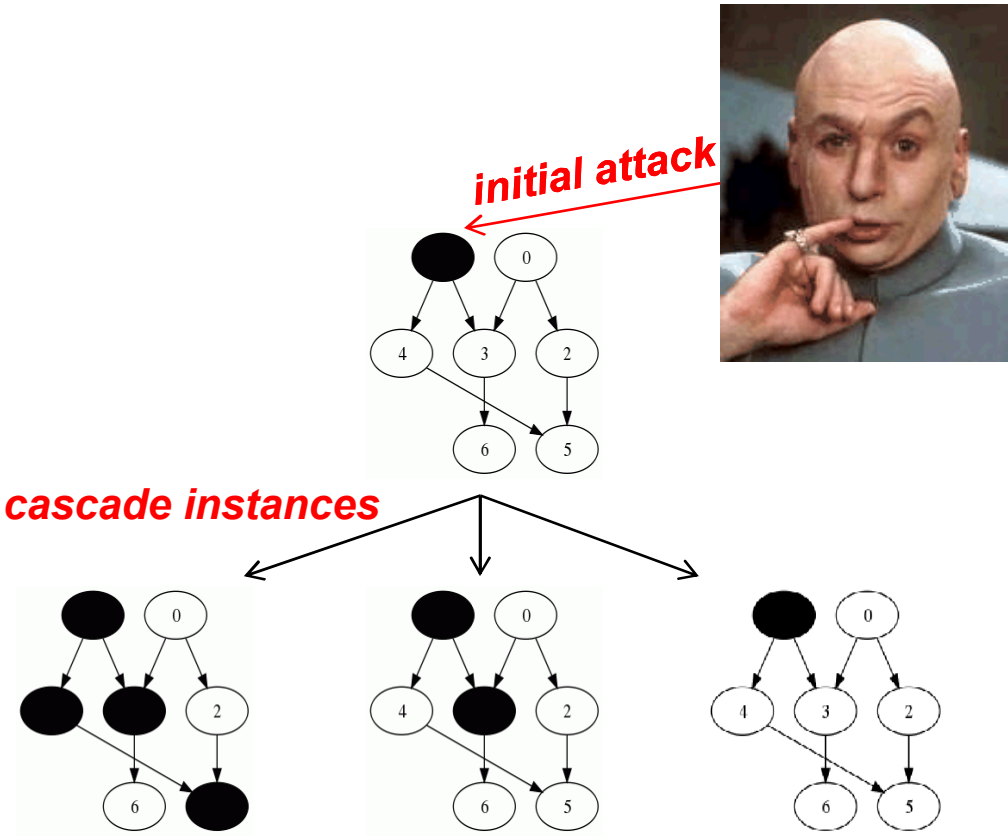




Interdependencies are naturally modeled by a network

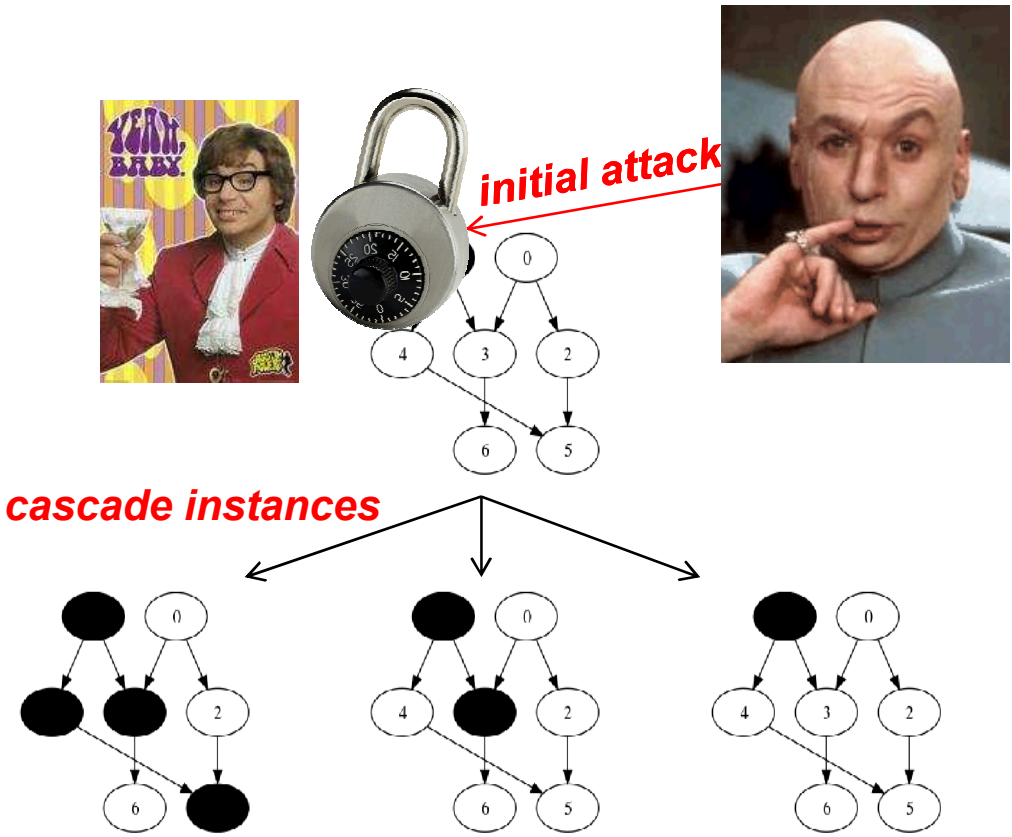
Graphs, Cascades, and "Attackers"

- *Interdependencies represented by a graph (causal, i.e., if A fails, then B can fail)*
  - *Domain experts*
  - *Static analysis: function calls, hardware signals, etc*
- *Consequences: probabilistic failure cascades*
- *Examples:*
  - *electric grid failure cascades (massive blackouts)*
  - *Blaster worm: failures of routers cause lots of routing table update messages, which cause other routers to fail, etc*
  - *Epidemics/bioterrorism*



“The Game”

- *Defender chooses security configuration (possibly randomized) at each node*
- *Attacker “observes” defense choices (probing, surveillance), chooses nodes to attack to maximize its expected utility*
- *Defender: optimal configuration accounts for attacker’s response*



Impact of defense

Sandia National Laboratories

- Defense (at a node) can: prevent attacks, alter cascade probabilities
  - In general, very complex model with lots of parameters
- Restriction #1: defense of a node only affects cascades to that node
  - Example: epidemics and inoculation
  - Effect still global (as long as i is on some path)
- Restriction #2 (stronger): defense only prevents direct attacks
  - Example: transfer luggage, separating insider and outsider

Restriction #2: Optimal Defense via. Simulations + LP

- *Cascade probabilities independent of decisions*
  - *Simulate cascades from each node (possible attack target)*
  - *Use resulting utilities to construct a compact game matrix (utility depends only on whether a target is attacked, and how it is defended)*
  - *Solve LPs for optimal security configuration*

for each target:

$$\max_{q} \quad \sum_{o} U_{o,\hat{t}} q_{o,\hat{t}} - \sum_{t} \sum_{o} c_{o,t} q_{o,t}$$

Subject to:

$$q_{o,t} \geq 0$$

$$\sum_{o} q_{o,t} = 1$$

$$\sum_{o} V_{o,\hat{t}} q_{o,\hat{t}} \geq \sum_{o} V_{o,t} q_{o,t}$$

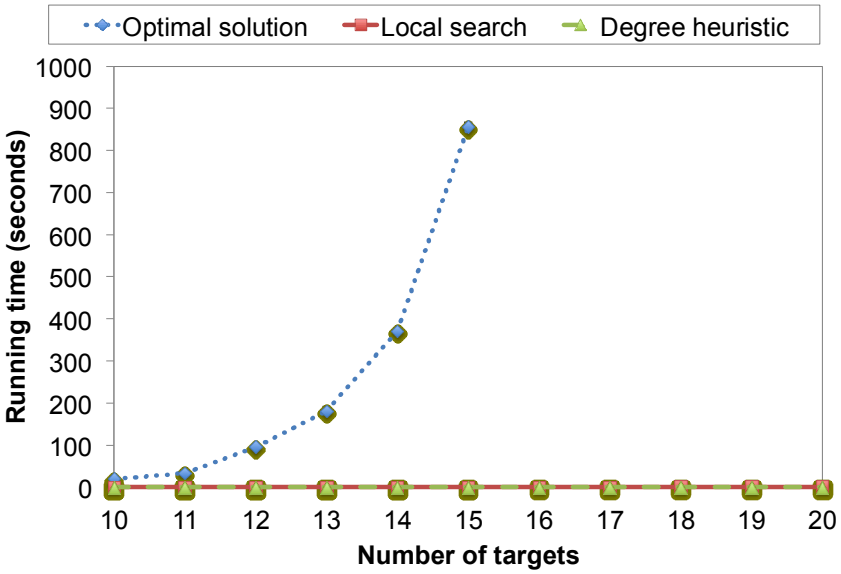
Restriction #1: Simple Greedy Heuristic

- *For N iterations:*
  - *choose a random target*
  - *choose an "optimal" security for that target, fixing decisions for others*

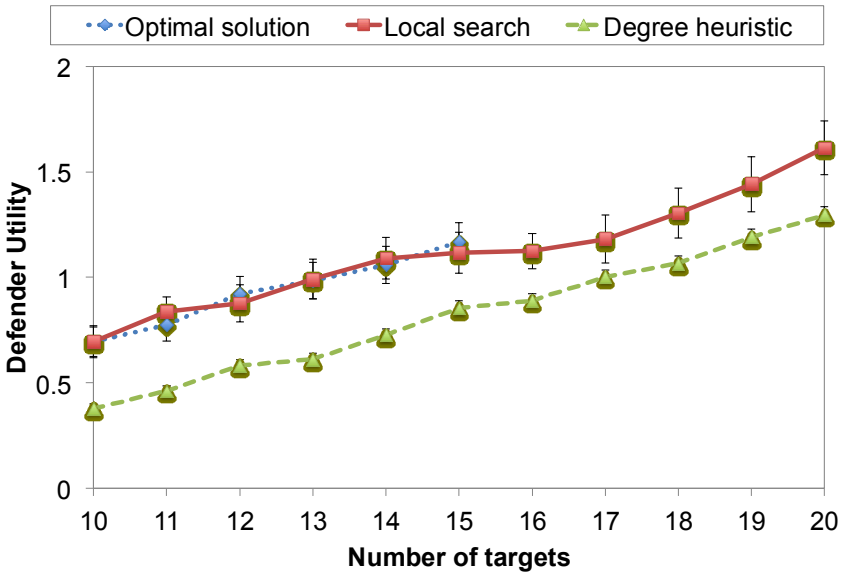
- A relaxation: attacker may choose up to K nodes to attack simultaneously
  - similar to max-influence;
  - attacker's objective is submodular; greedy is near-optimal
  - Using this for defense is near-optimal if the game is "close" to zero-sum

Evaluation: How good is the heuristic

runtime



utility



Heuristic is fast, and near-optimal

(much better than "choose high-degree target" heuristic")

Network resilience: new answers to an old question

Sandia National Laboratories

We can use our tools to compare resilience properties of complex networks

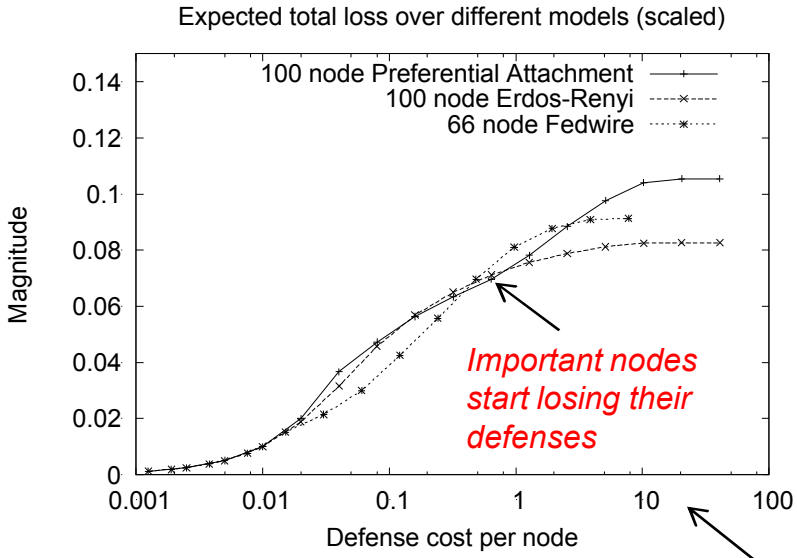
State-of-the-art

- *Barabasi et al., Nature '00: skewed degree distribution => less resilience to targeted failures (attacks); looked at node removal, not contagion; no defense*
- *Pastor-Satorras and Vespignani, Phys Rev E, '02: inoculate (defend) highly connected nodes in an epidemic; no attacks (random failures); "defense" removes a node (no partial protection)*

Our analysis

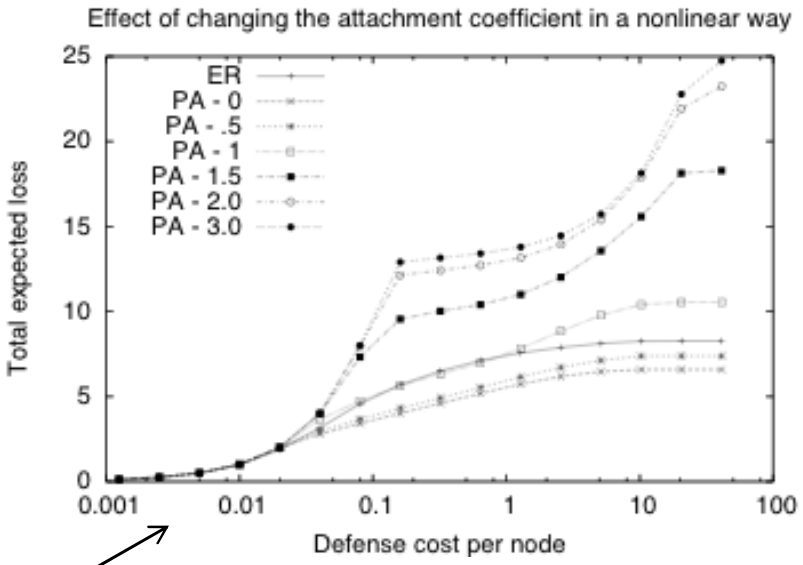
- *Consider both targeted attacks and endogenous security decisions, which are optimized (rather than heuristic)*
- *Skew of degree distribution no longer determines resilience; in some cases, no difference between scale-free and Erdos-Renyi; in general,* ***scale-free can be better or worse***

Degree distribution and resilience



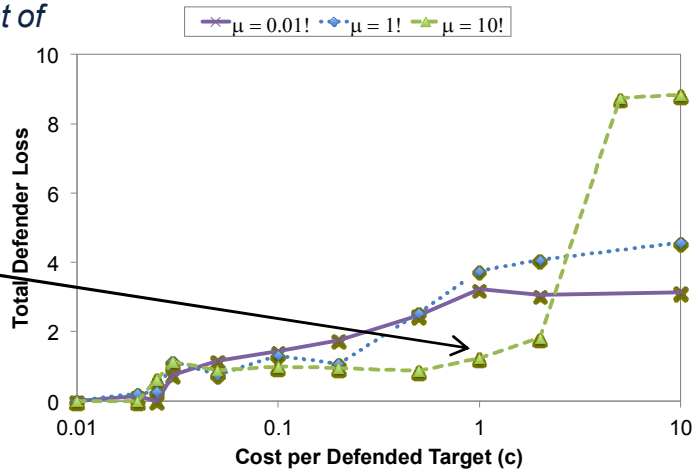
Erdos-Reny ~ Scale free

cascades independent of defense



*More inhomogeneity => **lower** resilience*

*More inhomogeneity => **higher** resilience*



full inoculation

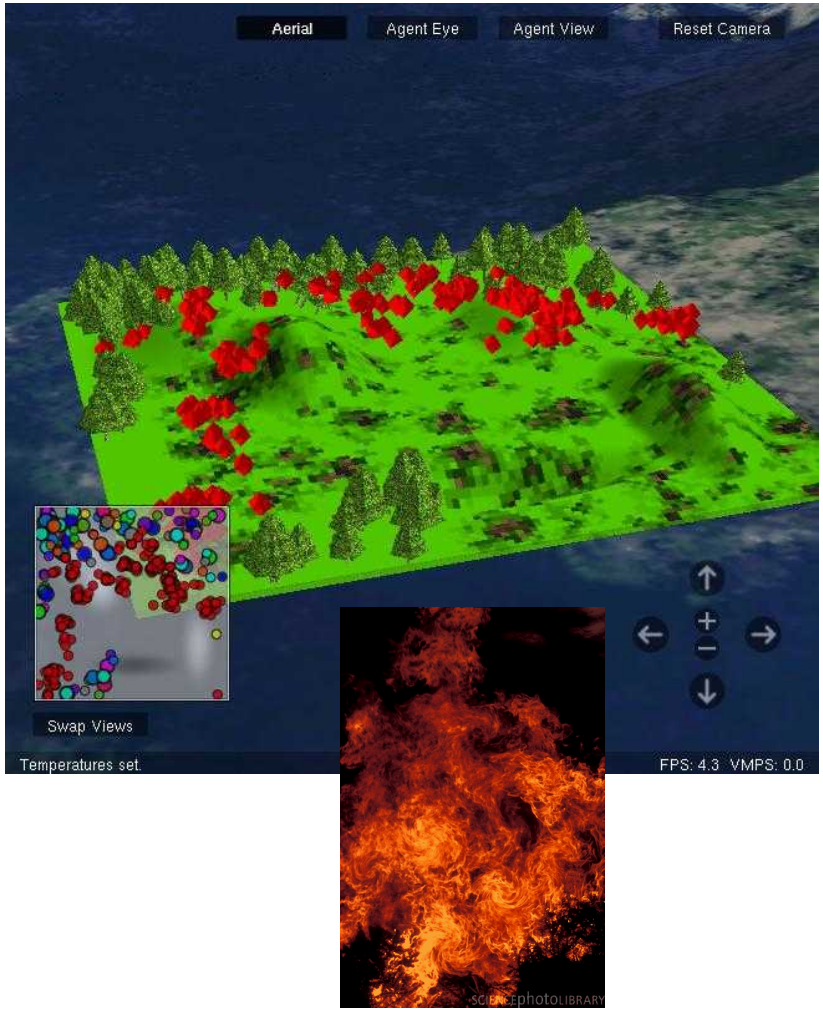
Sandia National Laboratories

EXTERNALITIES IN SECURITY, AND THE IMPACT OF DECENTRALIZATION

Phys. Rev. Letters, 2011

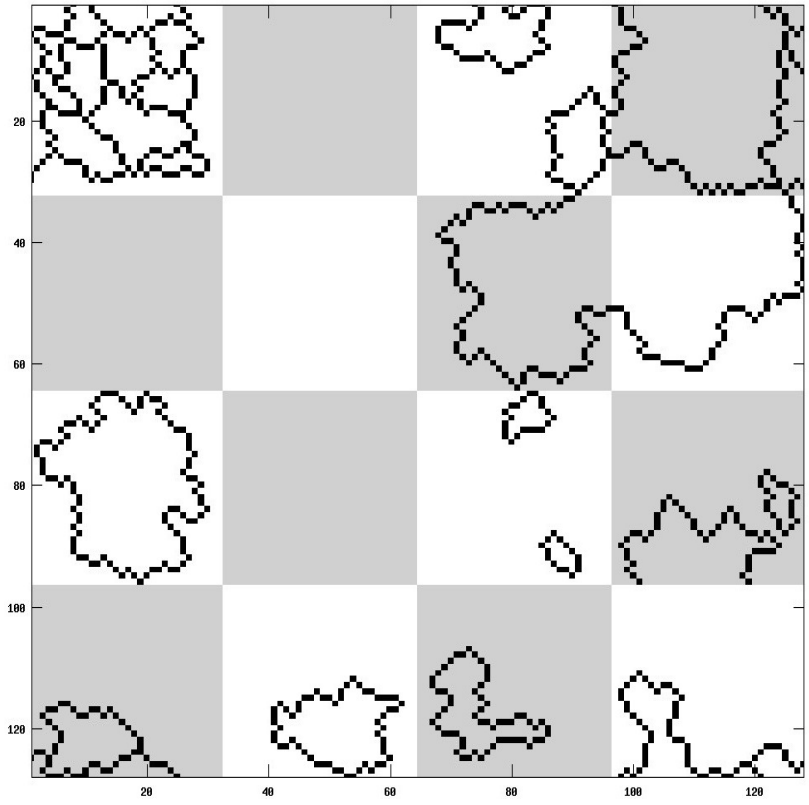
Setting: Forest Fires on a Grid

- Land = square grid of cells
- Each cell can have a tree
- Lightning strikes according to probability distribution over the entire grid
  - if lightning strikes an empty cell, nothing happens
  - if lightning strikes a tree, entire connected component burns down
- Who cares? (except foresters)
  - Models (implicitly) consequences of common operating environments:
  - *tree = preferred, OS (e.g., Windows), email client, etc*
  - *exploit/virus for one COE machine impacts the entire connected component*
- *Previous analyses:*
  - *comes from physics*
  - *SOC (agent-based model)*
  - *HOT (optimization)*



The model of decentralization

- (N x N) grid divided into non-overlapping subgrids
- Each subgrid "owned" by a player
- Planting a tree costs c ∈[0,1]
- Player utility = Yield – Cost
  - *Yield over my subgrid only!*
- Global utility = Global Yield - Cost
- ***Seek***: ***Nash equilibrium grid configurations***
- *Observe: negative externalities of planting trees (failure cascades can pass through my land to others)*



Result #1: PoA & PoS are infinite when each player owns a single cell & c = 0

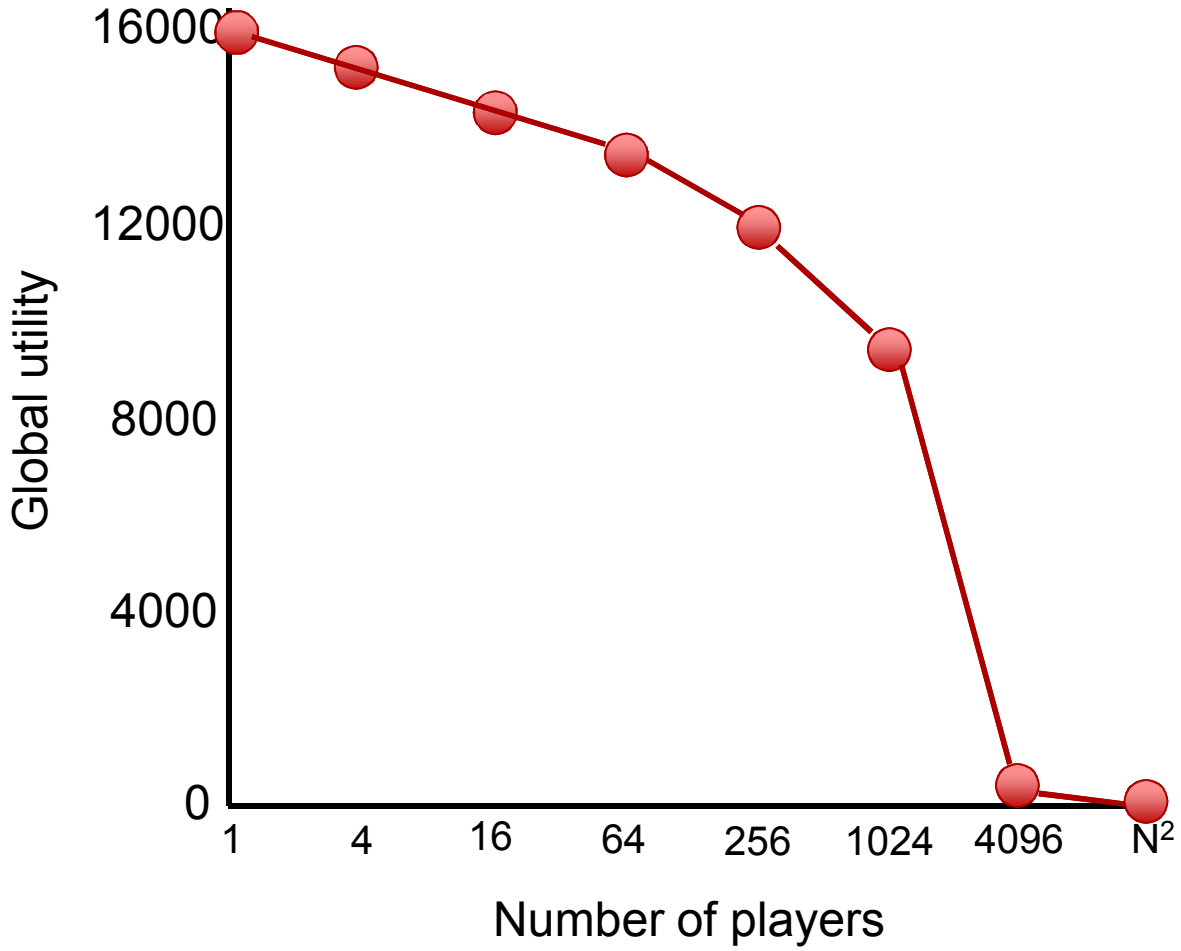
- 1 player: Nash equilibrium = optimal solution (lone player's utility coincides with welfare)
  - Optimal utility is $O(N^2)$
- N x N players (one per grid cell)
  - *Nash equilibrium*: every player plants a tree (actually, weakly dominant strategy)
  - Global utility in Nash equilibrium = 0
  - ***Reason***: costs nothing to plant, so no reason not to!
- *This analysis is only about the two extreme cases*
- *What about the general case (arbitrary # of players)?*

Sandia National Laboratories

General Case: simulation-based game theoretic analysis

- Mathematically challenging
- Algorithmically intractable: even a globally optimal solution very difficult to compute
  - For a 128 x 128 grid, joint pure strategy profile space of the game is 2^{16384}
- *Our approach*: approximate Nash equilibria using best response dynamics
  - Bonus: a principled means to model dynamic behavior + equilibrium selection

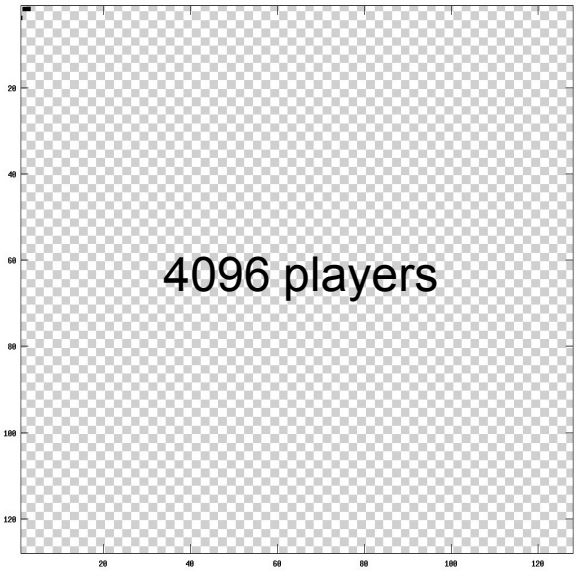
Results: Global utility in NE (c = 0)

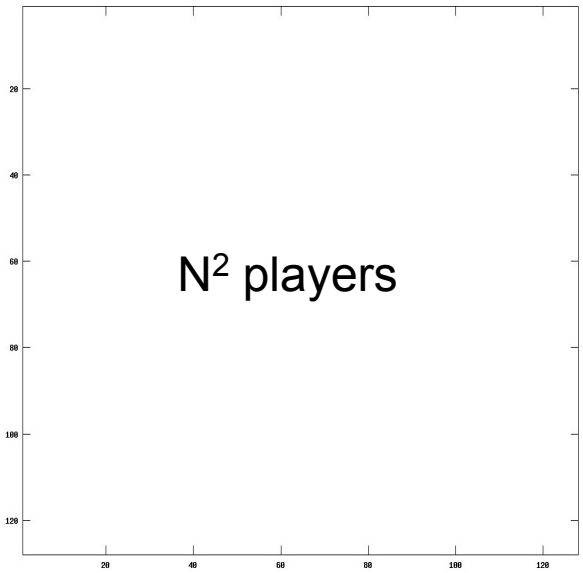


Equilibrium Landscape ($c = 0$)

Players build fenses around their territory to protect against "poor" decisions by neighbors:
maintain robustness despite externalities

But: if too decentralized…

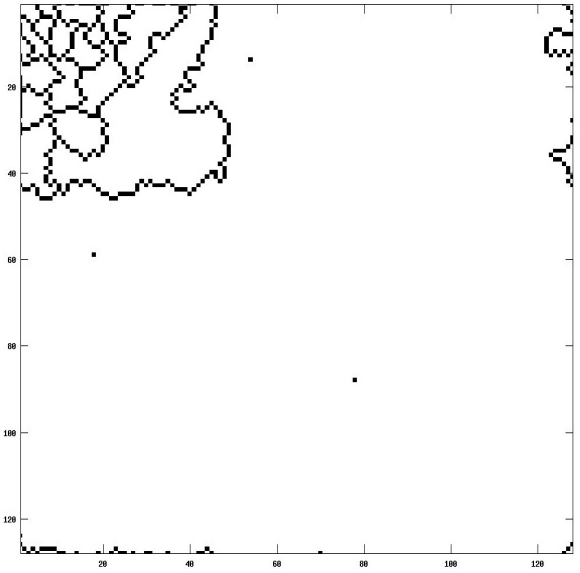


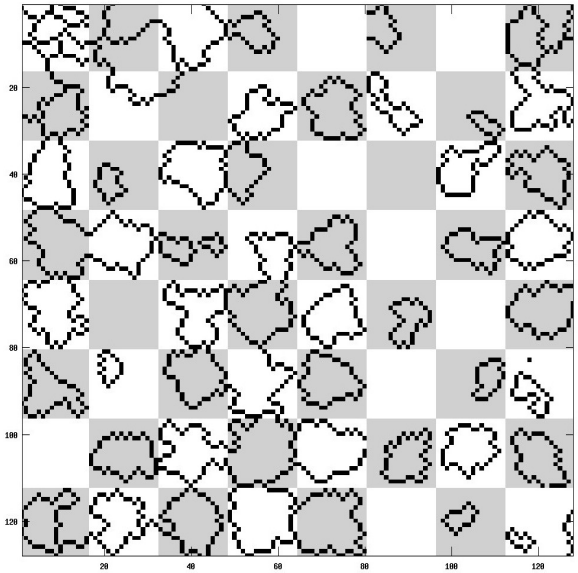


Robustness and Fragility

Sandia National Laboratories

HOT: well adapted to lightning distribution, but fragile to changes

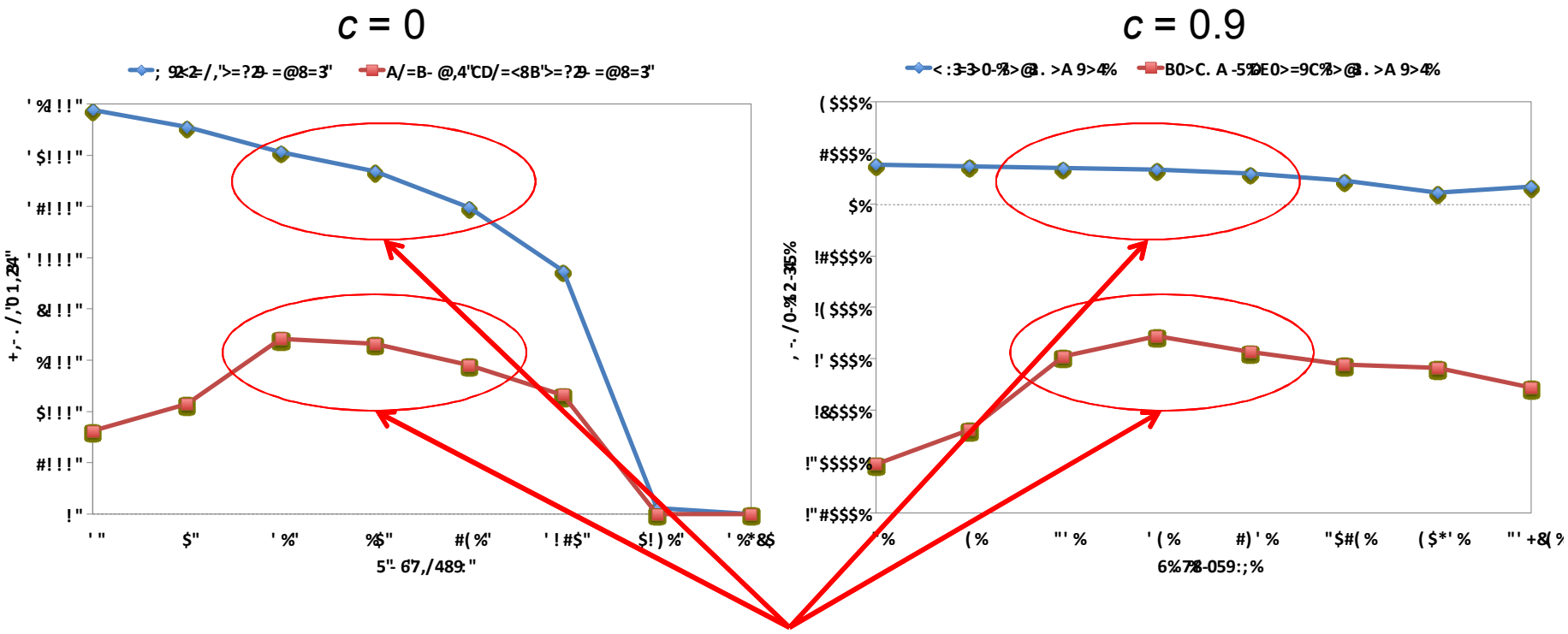




As the number of players increases:

1. ***Landscape is increasingly homogeneous***
2. ***Lower correlation between lightning peak and location of fire breaks***

Decentralization leads to greater resilience to changes



Intermediate number of players: **near-optimal; resilient to changes**

Positive Value of Negative Externalities

Sandia National Laboratories

- *Negative externalities can serve the common good!*
  - Since players protect themselves against inferior decisions by others, equilibrium outcomes are near-optimal AND the system is increasingly resilient
  - However: cannot have too much decentralization, when players become too myopic
- Lessons for complex system design, where decentralization is a necessity
  - Must take care not to divide the system too finely among designers
  - *Rewards to individual design teams should be based on performance of their designs as a part of the entire system*
    - *Creates incentives for individual designers to protect themselves against design/implementation flaws originating elsewhere in the system*

Sandia National Laboratories

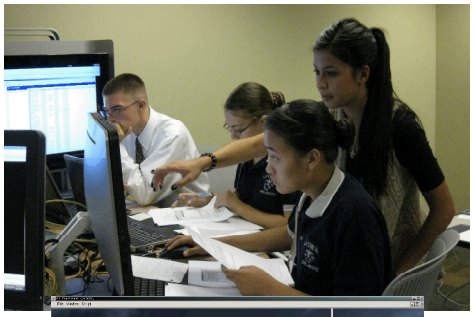
CYBER GAMES WITH INTERNS:
A SECURE DESIGN COMPETITION

CSIIRW, 2013; working paper

The Competition Setting: Sandia's Center for Cyber Defenders (CCD)

Sandia National Laboratories

- CCD is a highly selective, applied research internship institute at Sandia's New Mexico and California sites
- Two student teams (three students each)
  - CA team (in Livermore, CA)
  - NM team (in ABQ, NM)



Competition Structure

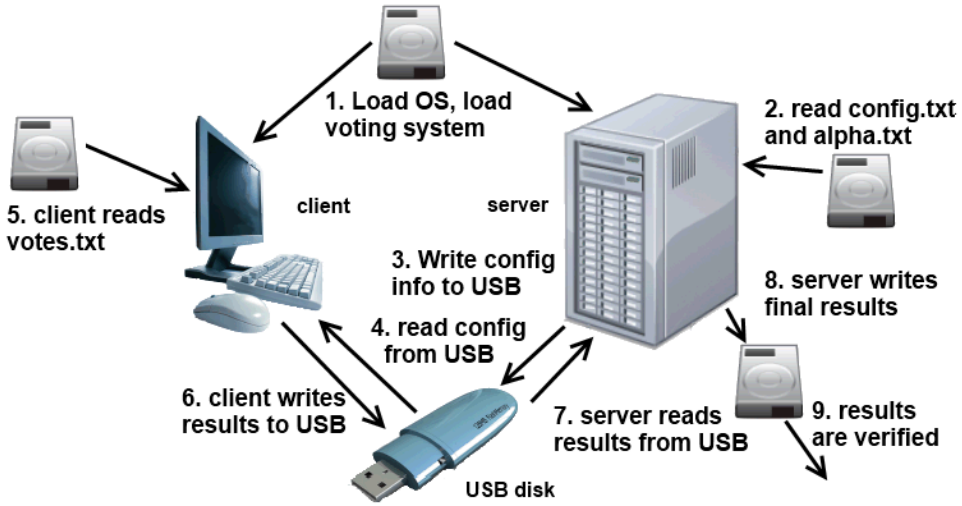
Sandia National Laboratories

- Two rounds. In each round, two phases:
  - *Design phase*: design a software system based on a specification
    - at the end of this phase, software systems are exchanged between the two teams
  - *Attack phase*: design exploits/attacks on the opponent's system

The Competition Specification: A *Stylized* Electronic Voting System

Sandia National Laboratories

- Designed to help students identify and internalize security principles
- Specification: requirements, rules, and evaluation and scoring procedure



EVS consists of a server (election management system) and client (voting station) with sneakernet USB for data transfer

Evaluation

Sandia National Laboratories

- Goal: evaluation should be as objective as possible
- Our solution: objective scoring system, and five scenarios which introduce attacks
  - *Baseline scenario*: no attacks (just checking compliance to spec)
  - *Client/server breach scenarios*: attacks introduced as "EVS user" onto a client/server
  - *Client "subversion"*: attacks introduced as "root" onto the client
  - *Malicious USB scenario*: malicious files placed on a USB drive
- A design team must compute the correct voting tally *in spite of threat* in each scenario
- Successful design (i.e., resisting attack) gets twice as many points as a successful attack

Competition Results

Sandia National Laboratories

- Students produced two substantially different designs
  - NM team focused on customizing the kernel and produced very small, highly restricted OS, while CA team implemented limited user shell and “red pill”
  - Teams chose different development platforms, tools, and crypto libraries
- Common design patterns:
  - Use of crypto libraries to ensure message integrity
  - Customize/restrict the kernel and userspace
    - Remove unnecessary functionality
    - NM team: even from the root user
  - Basic access control (password, CAPTCHA)

Observations

Sandia National Laboratories

- Students improved their understanding of and ability to articulate secure design principles
  - Reduce attack surface
  - Use existing tools
  - Enforce policies at lowest level
  - Defense in depth
  - Prevent easy access
- Specification of an effective competition is nontrivial: despite extensive pre-work, numerous ambiguities surfaced and unanticipated issues arose
- Competitions are great motivator… this was billed as research project but students were quick to forget

Security Lessons Learned

Sandia National Laboratories

- Lesson 1: security is hard/expensive
  - A system that complies with the spec took ~4 hours
  - A “secure” system took ~2 months
- Lesson 2: interleaving design and red teaming is a good idea
  - Second round designs were much better than first round (mostly DoS attacks)
- Lesson 3: defender can succeed
  - round 2 attacks were unable to reliably affect the voting outcomes
  - severely limiting system functionality “works”, even when you start with a general-purpose system!
- Lesson 4: it is possible to protect against root-level attacks
  - severely limiting what one can do as root (or eliminating the root user almost entirely) is highly effective

Takeaways

Sandia National Laboratories

- Can't (shouldn't) do defense without consider the attacker
- *Game theory is a formal way to model how an attacker will respond to defense decisions*
- *Computational* game theory is key:
  - security, particularly cybersecurity, is much too strategically complex to be analyzed "on paper"; stripping away strategic complexity strips away the most important aspect of the problem
  - can provide broad theoretical insights about problems
  - can offer operational guidance (optimal mitigations, or cost-benefit analysis, and a prediction of attacker response)

Other work I have done

Sandia National Laboratories

- Optimal epidemic control (working paper)
  - Mixed-integer program formulations for optimal inoculation and facility closure
  - Consider both pre-emptive (stage 1) models and multi-stage models in which we can do inoculations as the disease manifests in the population
  - The first model of epidemic control on networks that can be explicitly optimized
  - Epidemics are transmitted by contact at a particular location, not on a "social" network
- Theoretical and algorithmic advances in simulation-based games (AAMAS, TOMACS)
  - A game with payoffs/outcomes specified procedurally
  - Convergence of equilibrium estimates from noisy samples
  - Derived confidence bounds on equilibrium approximation quality
  - Algorithms for approximating equilibria in very large games
- Computational Mechanism design (EC, UAI, DSS, JAAMAS)
  - Optimal incentive design in complex systems
  - Applied to ad auctions, combinatorial auctions
- Behavioral experiments on networks (EC [***best paper nomination***], WINE, PNAS)
- Physical security/adversarial patrolling/limited surveillance (AAAI, AAMAS)
- Noise sensitivity of Boolean networks (PRL)