Heat-ray: Combating *Identity Snowball* Attacks Using Machine Learning, Combinatorial Optimization and Attack Graphs

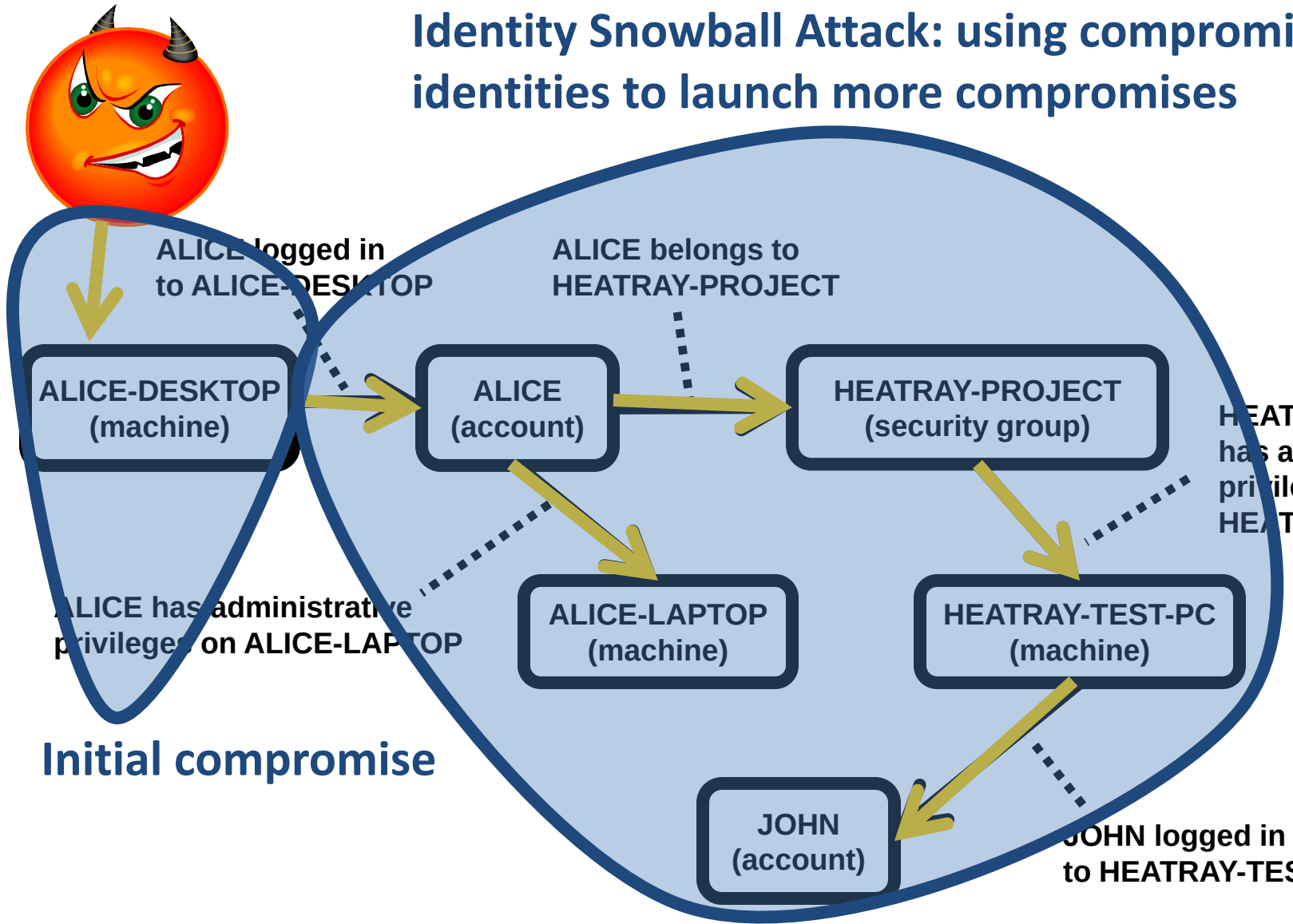
John Dunagan, Alice Zheng
Microsoft Research
Dan Simon
Microsoft

Outline

- Problem
  - Define identity snowball attack
  - Measure attack potential in large organization
- Heat-ray Solution
  - How Heat-ray **scales** to the amount of configuration in a large organization
- Evaluation
- Related Work
- Conclusion

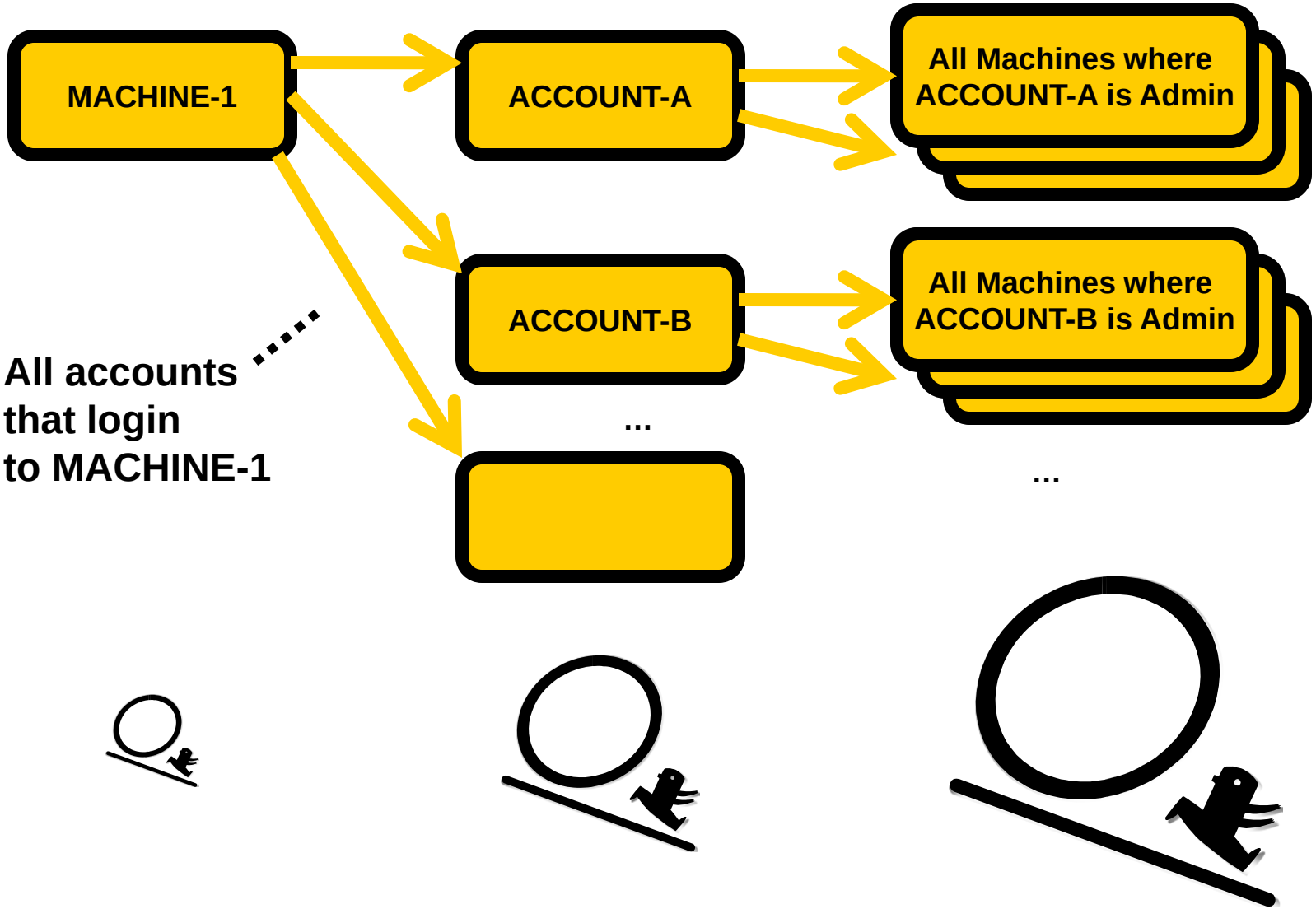
How An Initial Compromise Can Lead To Additional Compromises

Identity Snowball Attack: using compromised identities to launch more compromises



Initial compromise

“Snowball Effect” of Additional Compromises

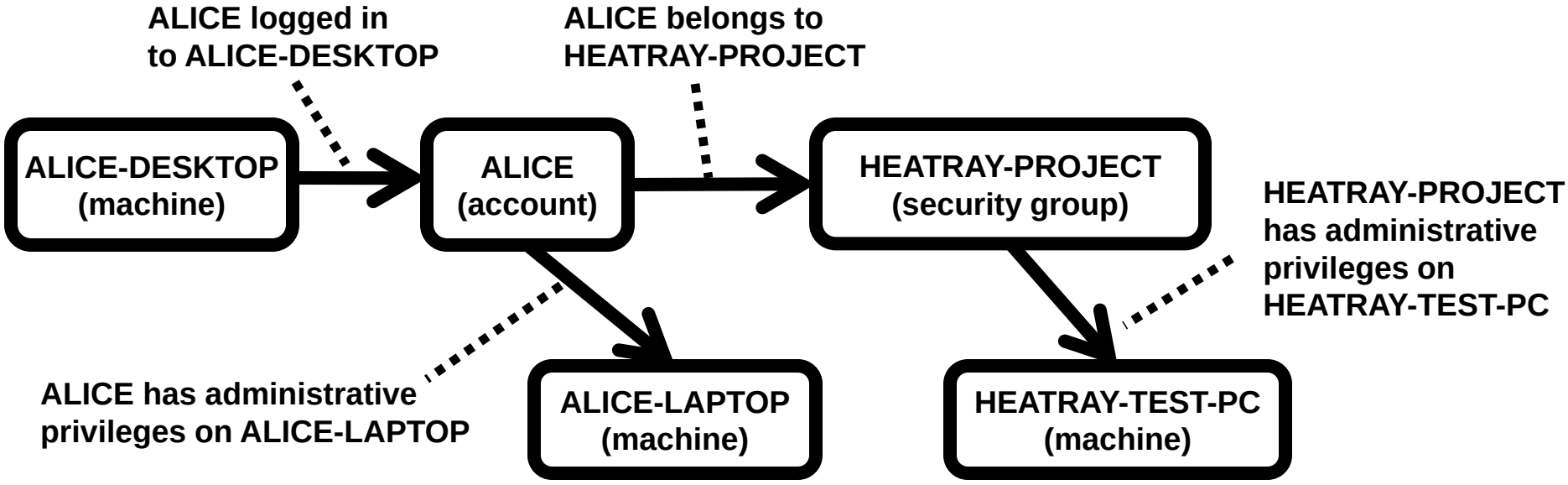


Threat In Practice?

- Some public attacks have iteratively used compromised identities
  - Morris worm (1988)
    - Back when the Internet was tiny
  - Attack reported by Singer (2004)
    - Cross-organization attack on academic and government sites
- No previous analysis on the threat of such attacks **within** a single large organization
  - Lots of computing done in large organization context
  - A large organization can have **millions** of locally reasonable security configuration choices
  - Are these choices globally reasonable?

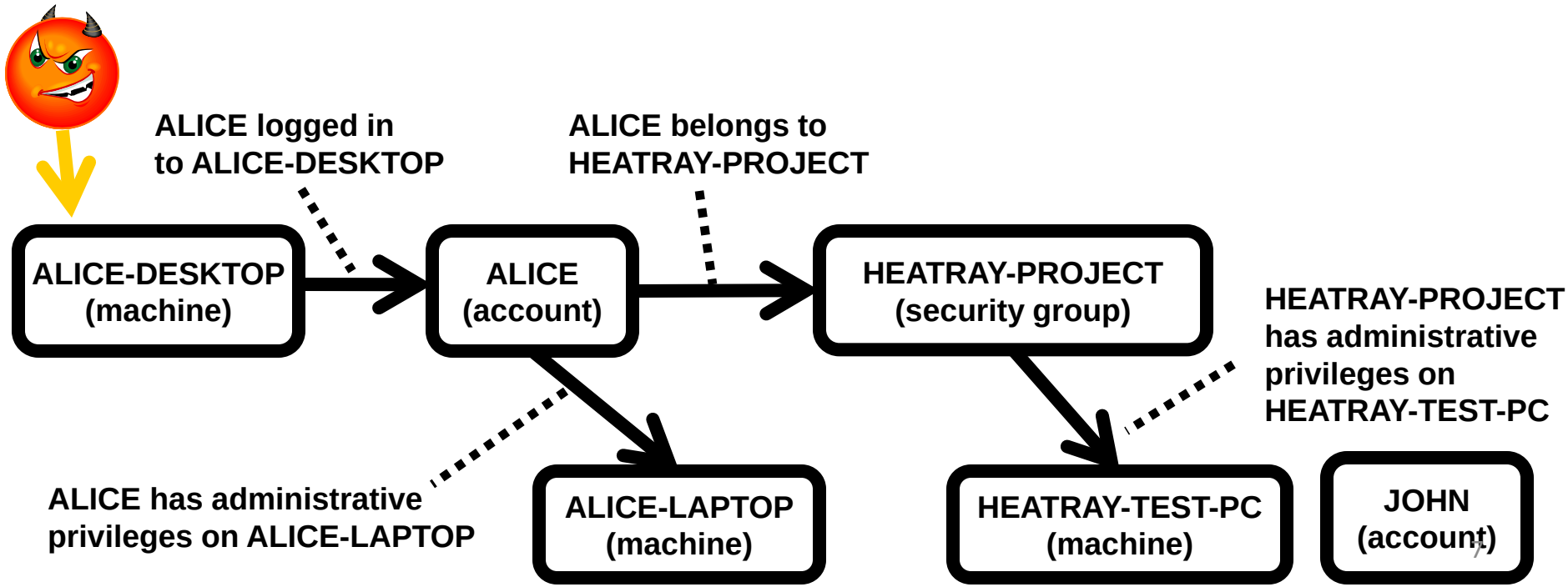
Let's Measure!

- 1 organization with ~100K accounts and ~200K machines
- Over 1 week, measure all the arrows shown below
  - Where accounts and groups have administrative privileges
  - What accounts belong to what group
  - Who logs in where

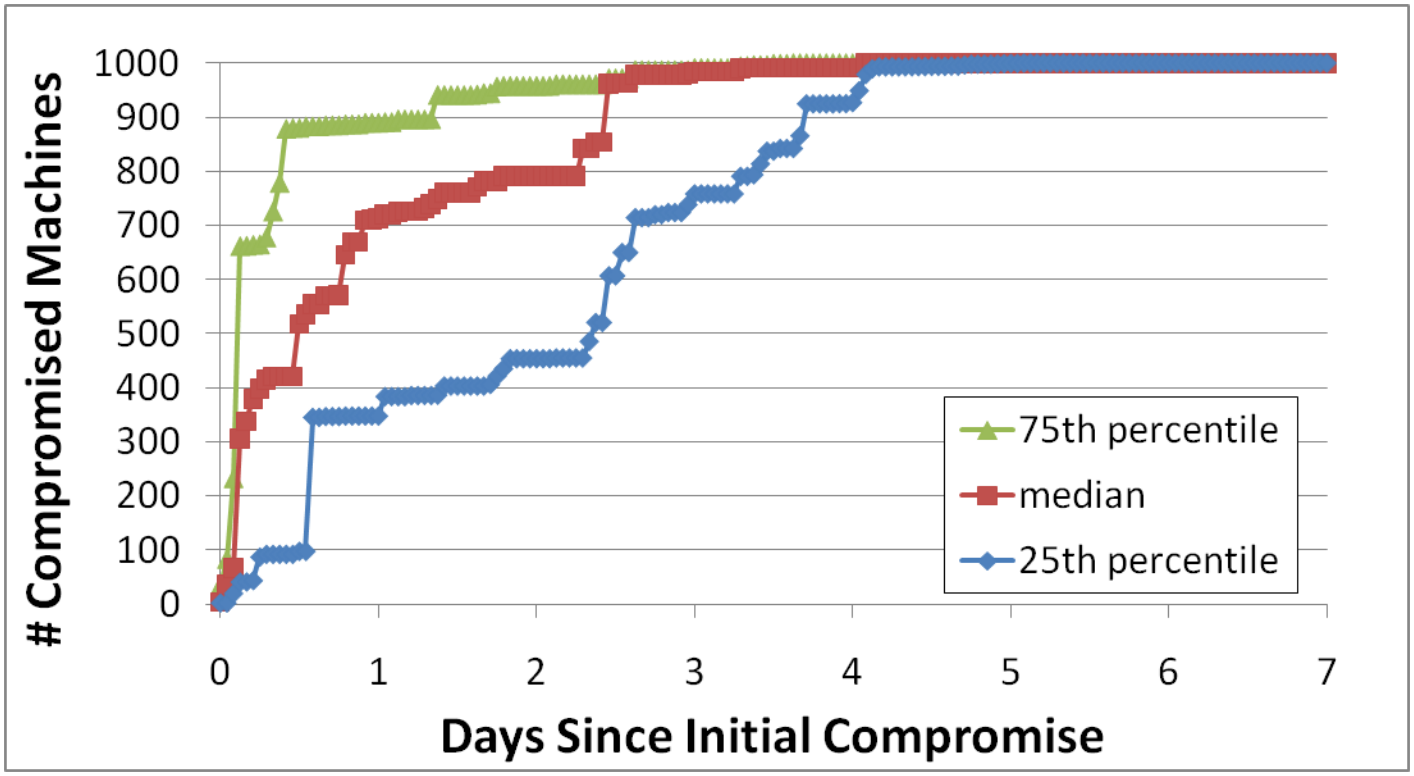


Modeling Issues

- What is attacker's "window of opportunity" after login?
  - Model accounts as immediately logging out (optimistic for defender)
- How fast does attacker compromise nodes where attacker now has administrative privileges?
  - Assume instant (pessimistic for defender, but rootkit install is quick compared to duration of login)



Reason For Concern



Cutoff at 1,000 for confidentiality reasons

- 100 trials, each with a single random initial compromise
  - Model progression of an identity snowball attack under assumption of immediate logout.

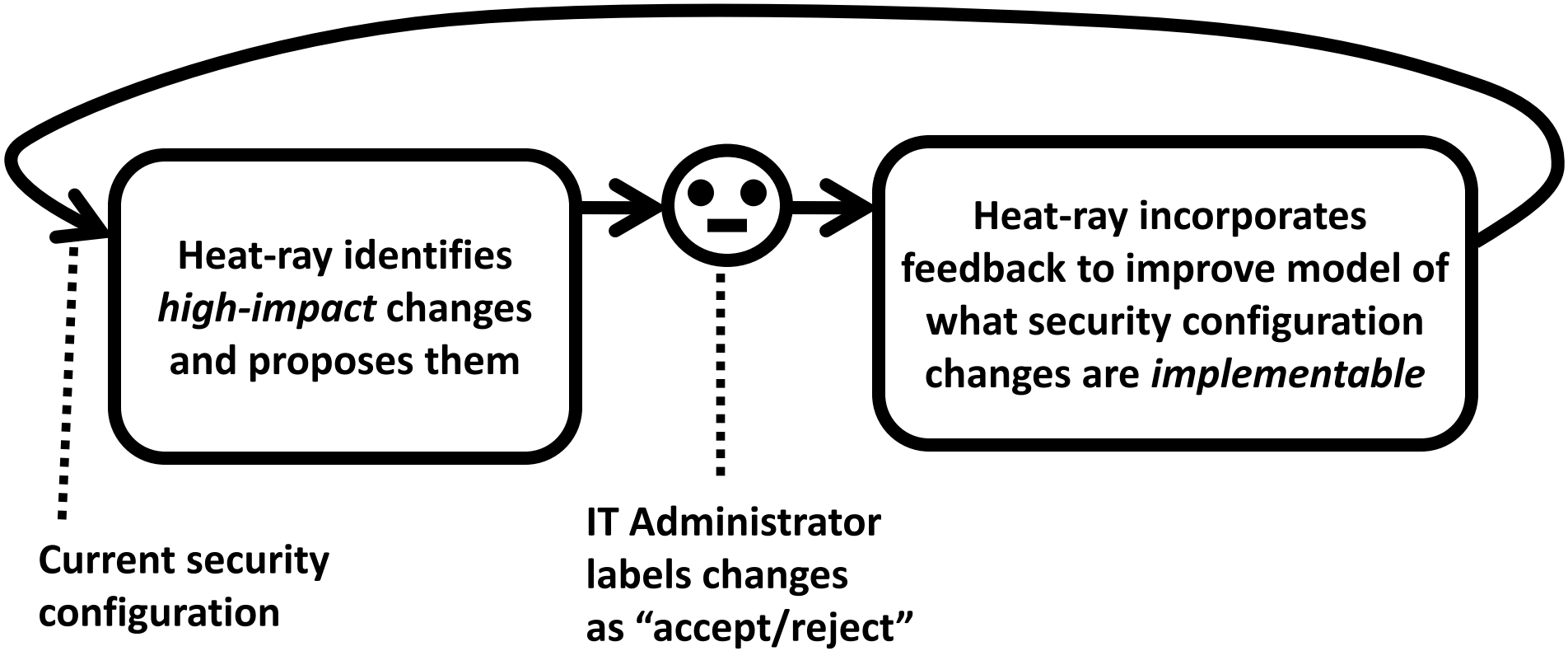
Problem Summary

- Identity snowball attacks…
  - magnify the impact of an initial compromise
    - With 200K machines, not realistic to assume zero initial compromises
  - have been used in the past in other contexts
  - could cause significant harm in the context of large organizations

Motivation for Heat-ray Approach

- Understanding the cumulative impact of individual trust relationships requires an algorithmic approach
  - Also the motivation for prior work on attack graphs. This prior work…
    - focused on defending a small set of high-value machines
    - relied on manual examination of many possible changes
- Securing large organizations requires **scaling** to the amount of security configuration in the organization
  - **millions** of possible configuration changes
  - some changes are **low impact**
    - i.e., little reduction in spread of an identity snowball attack
  - some changes are **not implementable**
    - e.g., person who patches the software needs those privileges

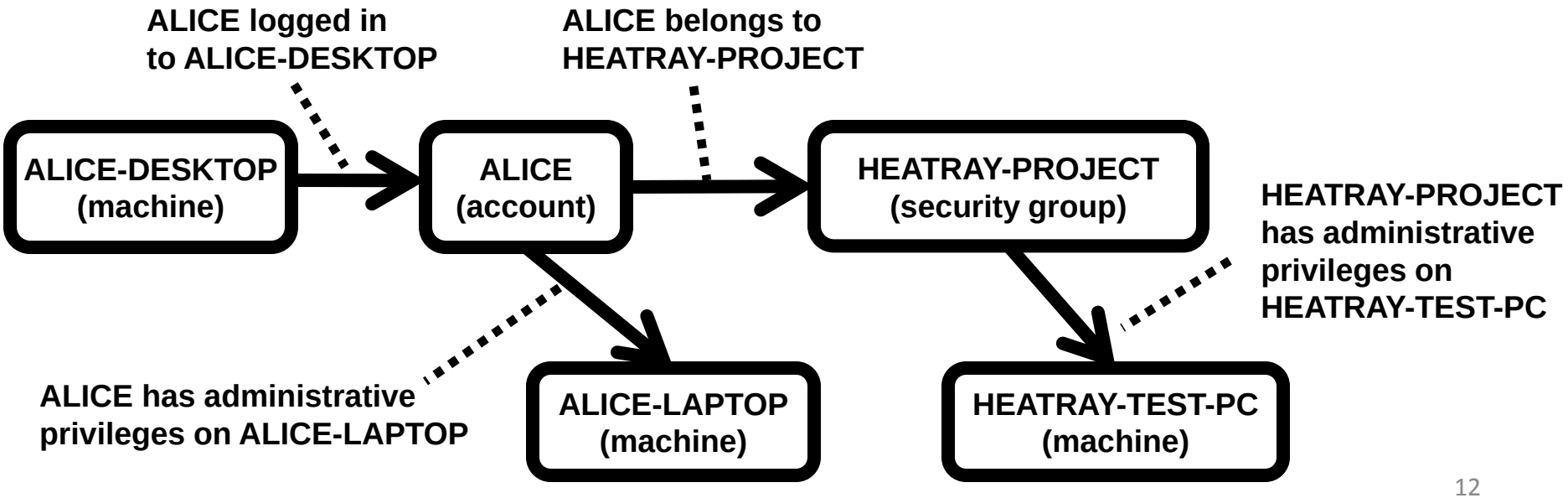
Heat-ray Solution



- Repeat loop until secure.

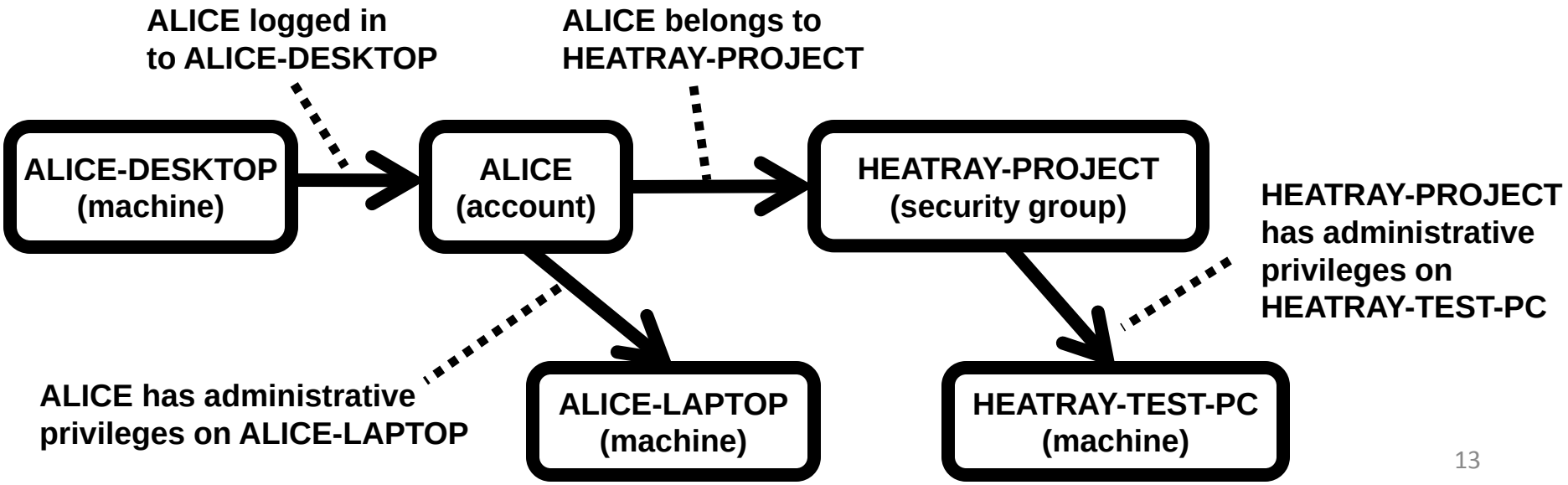
Proposing High-Impact Changes (1/4)

- Make problem suitable for algorithmic analysis using the formalism of an attack graph
- Node in graph = Asset to protect
- Edge in graph = Admin privilege, login, group membership
- Security configuration change = remove edge in graph
  - E.g., remove ALICE's administrative privileges on ALICE-LAPTOP



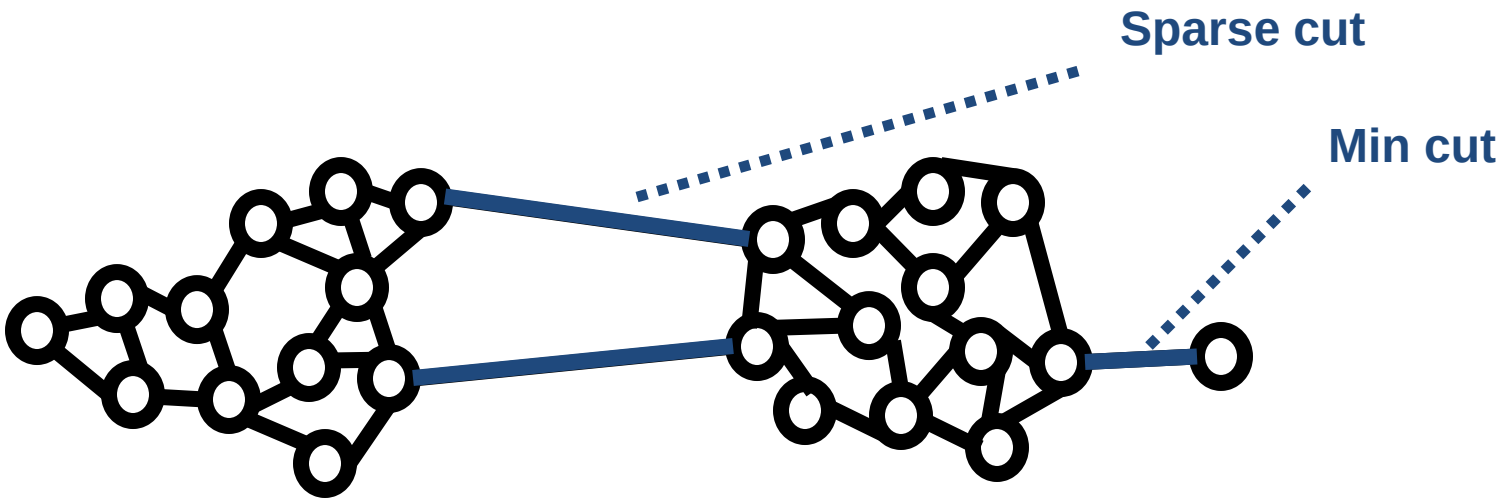
Proposing High-Impact Changes (2/4)

- Intuitively, a set of changes is **high-impact** if
  - It's a small # of changes and it prevents many compromised nodes from threatening many other nodes
- In graph terms, this becomes
  - A small set of edges that separates a large set of nodes from another large set of nodes



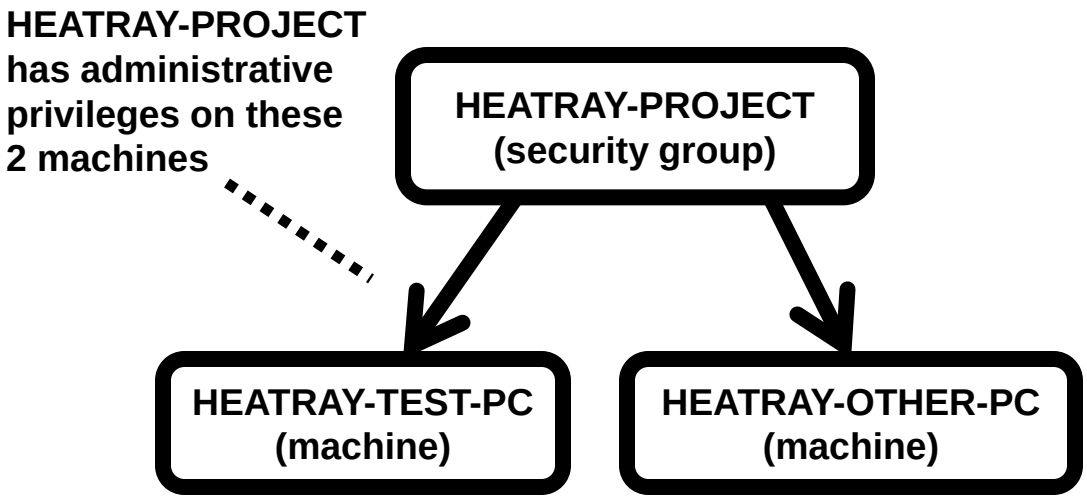
Proposing High-Impact Changes (3/4)

- This mathematical problem is exactly *sparsest cut*.
- Similar to min-cut, but balances
  - small number of edges in cut with
  - large number of separated nodes
- We modify an existing sparsest cut algorithm to run faster by relaxing its approximation guarantee

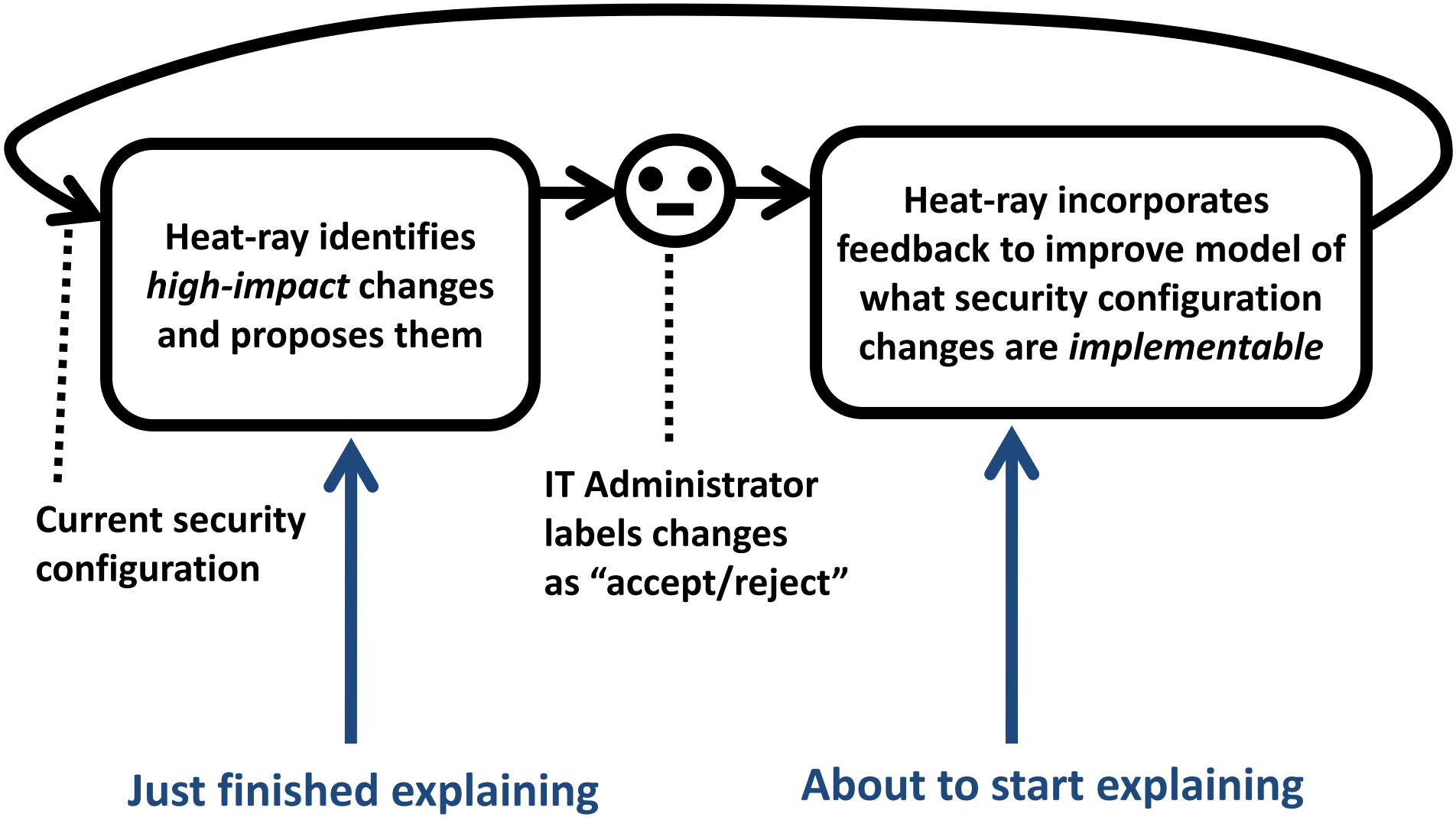


Proposing High-Impact Changes (4/4)

- **Group** edges to further reduce burden on IT Administrator
  - common start or destination node → "edge group change"
  - E.g., "Remove HEATRAY-PROJECT security group from having administrative privileges on every machine" refers to a group of 2 edges
- Use **impact** to rank groups and individual edges and **present**

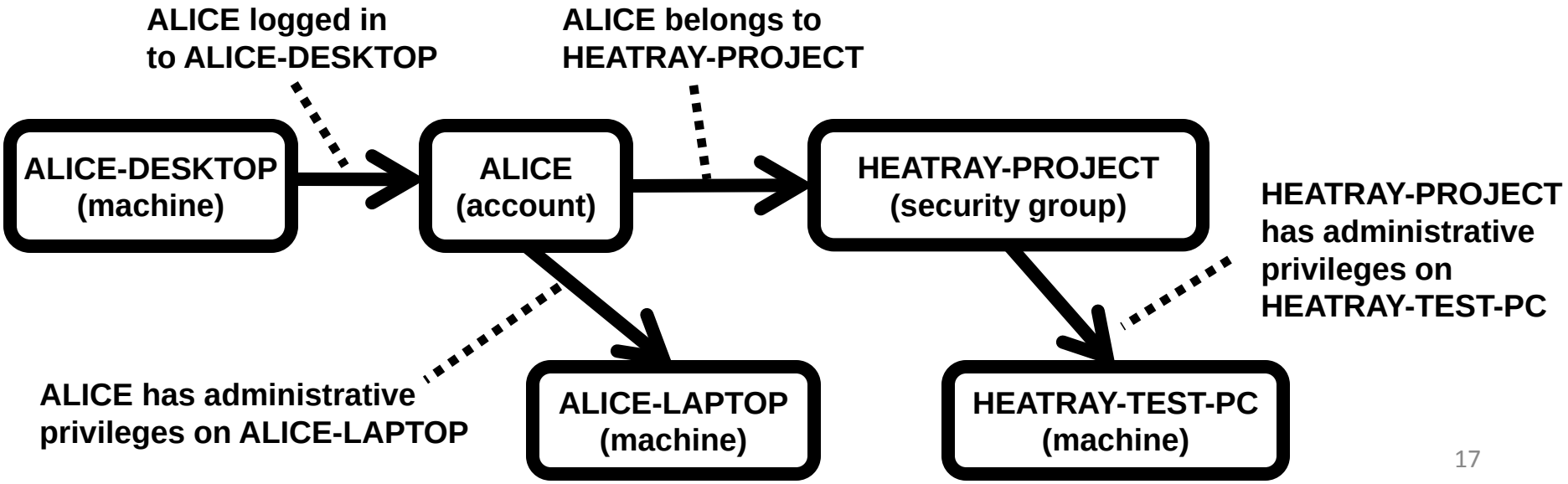


Quick Recap


Heat-ray identifies *high-impact* changes and proposes them
Heat-ray incorporates feedback to improve model of what security configuration changes are *implementable*
Current security configuration
IT Administrator labels changes as "accept/reject"
Just finished explaining
About to start explaining

Identify Implementable Changes (1/2)

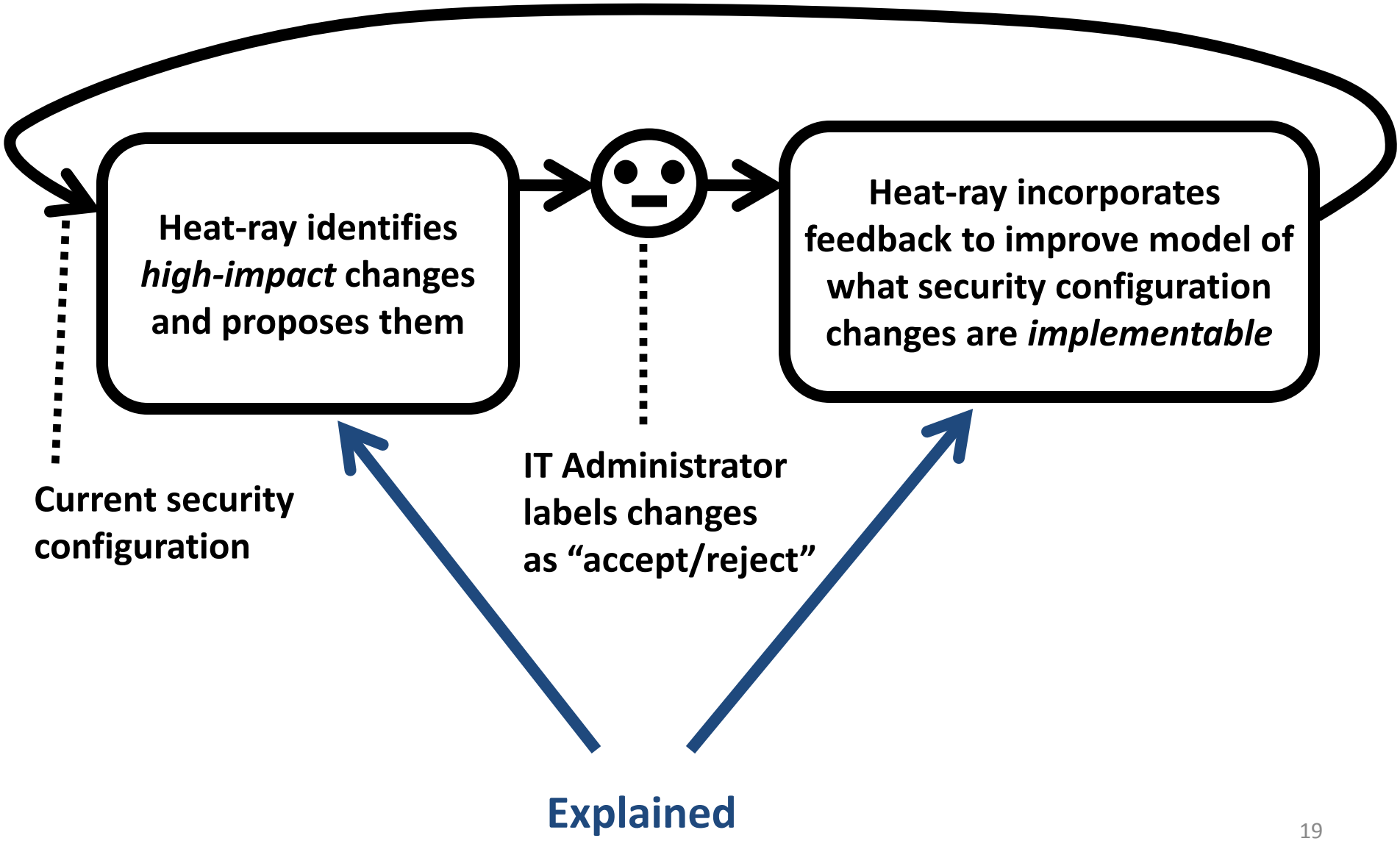
- There are too many edges to label them all manually
- Instead, use machine learning to generalize from the small number of labels already provided by the IT Administrator
  - Changes that IT Administrator accepted = cheap edges to cut
  - Changes that IT Administrator rejected = expensive edges to cut



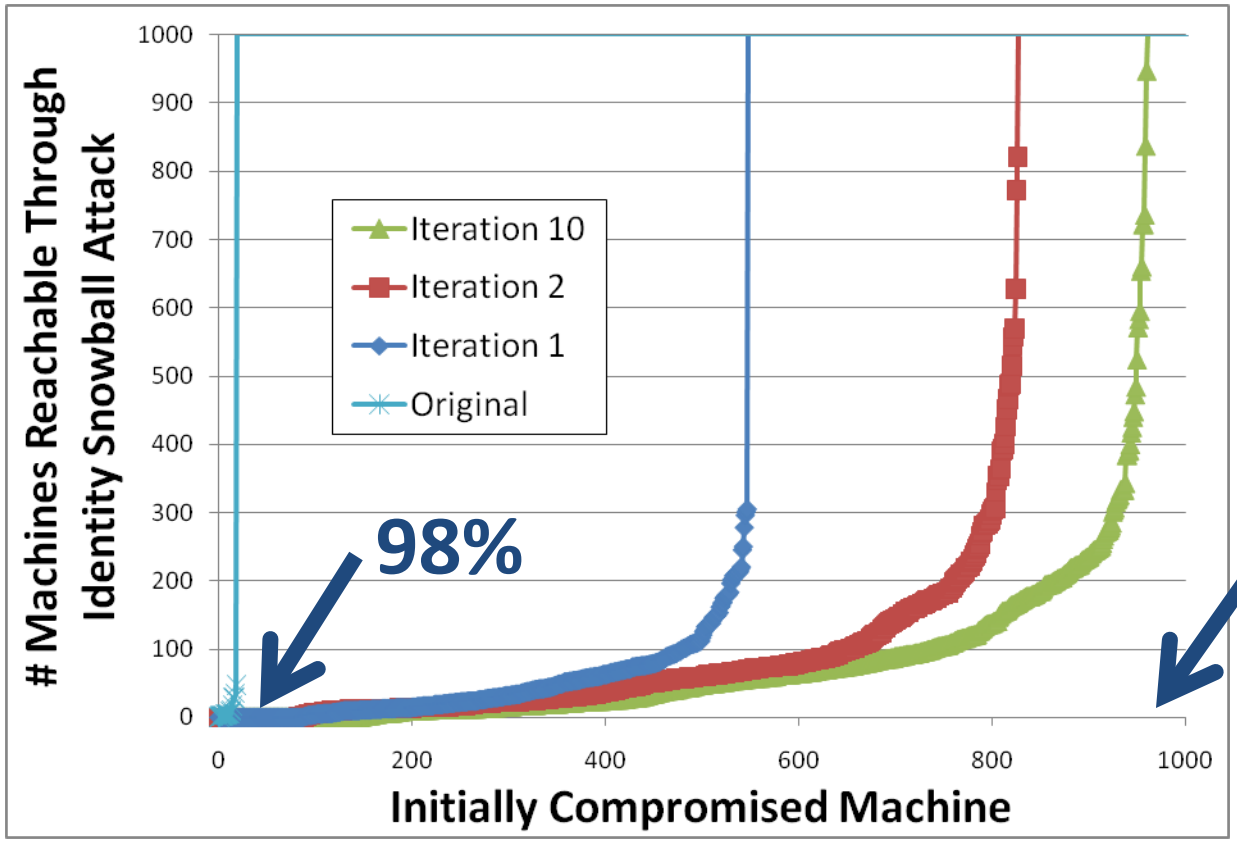
Identify Implementable Changes (2/2)

- How to determine if an unknown edge is more like the “known cheap” or “known expensive” edges?
  - Model unknown edge cost as function of other attributes (linear function over in/out degrees of edge’s start/destination nodes)
  - Sparse cut algorithm already yields edge benefit as intermediate result
  - **Accept** configuration change → constraint that edge benefit **greater than** edge cost
  - **Reject** configuration change → constraint that edge benefit **less than** edge cost
- Use Support Vector Machine (SVM) approach from machine learning to find cost model that best fits constraints
- Use learned cost model to estimate cost (= implementability) of all unknown edges
  - sparsest cut will now automatically balance **impact** with **implementability**

On To Evaluation



Evaluation of Effectiveness



10 iterations through Heat-Ray loop

Examine 900 changes on each iteration

- After each iteration, do 1K trials, each with a single random initial compromise
  - Model progression of identity snowball attack assuming logins don't go away
    - I.e., switch to using defender-pessimistic model of logins
  - Sort trials by # machines reached, generate 1 curve from these 1K trials

Evaluation and Responsible Disclosure

- This work was done in coordination with the IT group in the studied organization
- Model for accept/reject that we used in our evaluation was developed in collaboration with this IT group
- We helped the IT group identify (and implement) security configuration changes that reduce the identity snowball threat

Additional Evaluation in the Paper

- Comparison of Heat-ray to alternatives
  - E.g., simple heuristics for identifying configuration changes
- Analysis of SVM
  - Misclassification rate
  - How learned model captures IT administrator's preferences
- Analysis of the changes identified by Heat-ray

Related Work

- Already compared to prior research using attack graphs
  - Heat-ray addresses new **scalability** challenges
- Prior research on authorization often focused on new mechanisms
  - Decentralized mechanisms (SFS, SDSI/SPKI, …)
  - Fine-grained Delegation (Singularity, …)
  - Information Flow Control (SIF, HiStar, …)
  - Heat-ray focuses on identifying the right **policy** for an existing mechanism

Take Away Principle

Managing security configuration requires system support.