University of Pittsburgh

University of Pittsburgh
School of Computing and Information

Welcome to the

SAC-PA Workshop

The CI Cybersecurity Workshop for Education and Research

This workshop is part of the project supported by the National Science Foundation under Grant No. 1642117, entitled *CICI Regional: SAC-PA: Towards Security Assured Cyberinfrastructure in Pennsylvania.*

Supported by

University of Pittsburgh

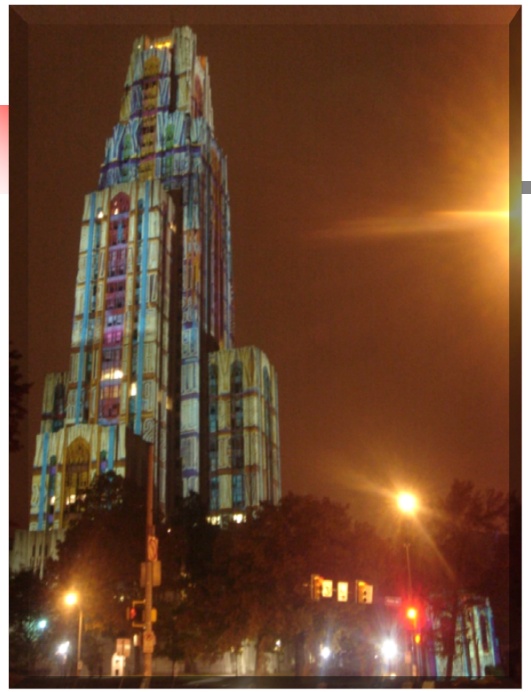
KINBER, A Commonwealth of Collaboration

Basic Info

- Breakfast, coffee breaks
- Meals
  - Lunch provided both days
    - Supported by University of Pittsburgh
      - Provost's Office, SCI
  - Dinner – on your own
- WiFi – Wyndham Pittsburgh <v93j3q>
- Need help?
  - Kelly Shaffer, Program Director at SCI
  - Runhua Xu, LERSAIS PhD student
  - Project team

Insider Threat Mitigation

Access Control Approach



James Joshi
Professor, Director of LERSAIS

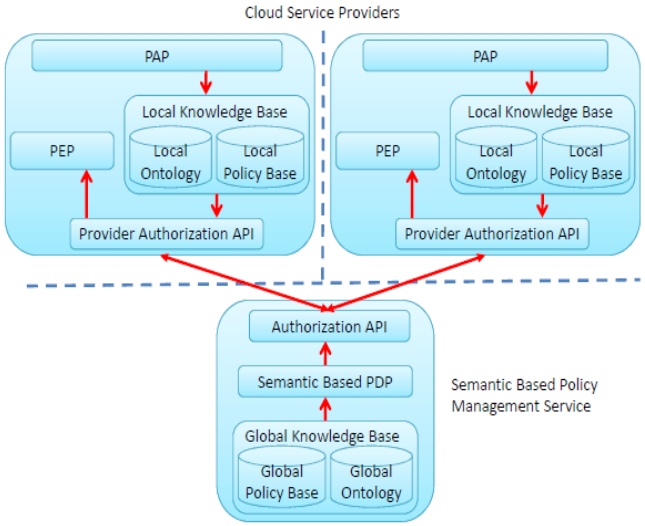
SAC-PA Workshop
June 22-23, 2017

But first …
Research Activities

- Advanced Access Control/ Trust Management Models/Approaches
  - Context based, Geo-social RBAC, Privacy/Trust aware RBAC
  - Secure Interoperation
    - RBAC, Trust based approaches
  - RBAC & Insider Threat Mitigation
  - Attribute based access (e.g., in Cloud)
- Insider Attack Mitigation
  - Cloud computing, Critical Infrastructure
  - Risk, Trust aware Access management
- Network Security
  - DDoS Attack, Some prior work in IPv6

Research Activities

- Security & Privacy in
  - Cloud computing & Social Network
    - Policy as a service; Access control in Cloud
    - **Privacy conscious execution in Cloud**
    - **Anonymization techniques**
    - **Privacy threat analysis (e.g., Identity Clone & Mutual Friend based attacks)**
    - Insider threats (NSA grant)
  - HealthCare IT
    - **Privacy aware Social Networks for Intimate Partner Violence**; Access control in Healthcare Systems
  - Location based services
    - Access/privacy control in LBSN
    - **Anonymization techniques**



Insider threat

Edward Snowden

"The year 2013 may be the year of the insider threat. … These incidents highlight the need to improve the ability of organizations to detect, deter, and respond to insider threats".

- Computer Emergency Response Team (CERT), **January 2014**.

Insider Attacks' Impact

- Accounted for around **30% of total incidents** reported from 2004 to 2014
- Monetary losses up to **$10 million**
- 75% of organizations had a negative impact on their operations
- 28% on their reputations
- 60% of respondents reported monetary losses caused by non-malicious insiders

Sources: Computer Crime and Security Survey 2010/2011 and The US Cyber Crime Survey 2014

More Recent …

- Insider attack frequency
  - Credential thief (imposter risk): 09.7%
  - Criminal & malicious insider: 21.8%
  - Employee or Contractor negligence: 68.4%
- Average annualized cost
  - Credential thief (imposter risk): $ 776,165
  - Criminal & malicious insider: $1,227,812
  - Employee or Contractor negligence: $2,291,591

"2016 Cost of Insider Threats" Ponemon Report

Current Approaches

- Access control systems are highly static
  - Only credentials are required
  - What about their behavior?
- Anomaly detection systems require manual verification and/or input
  - Unreliable and slow
- Risk methodologies are performed sporadically (e.g., NIST, Octave, etc.)
  - Do not minimize risk exposure continuously and automatically

So, what can we do about it?

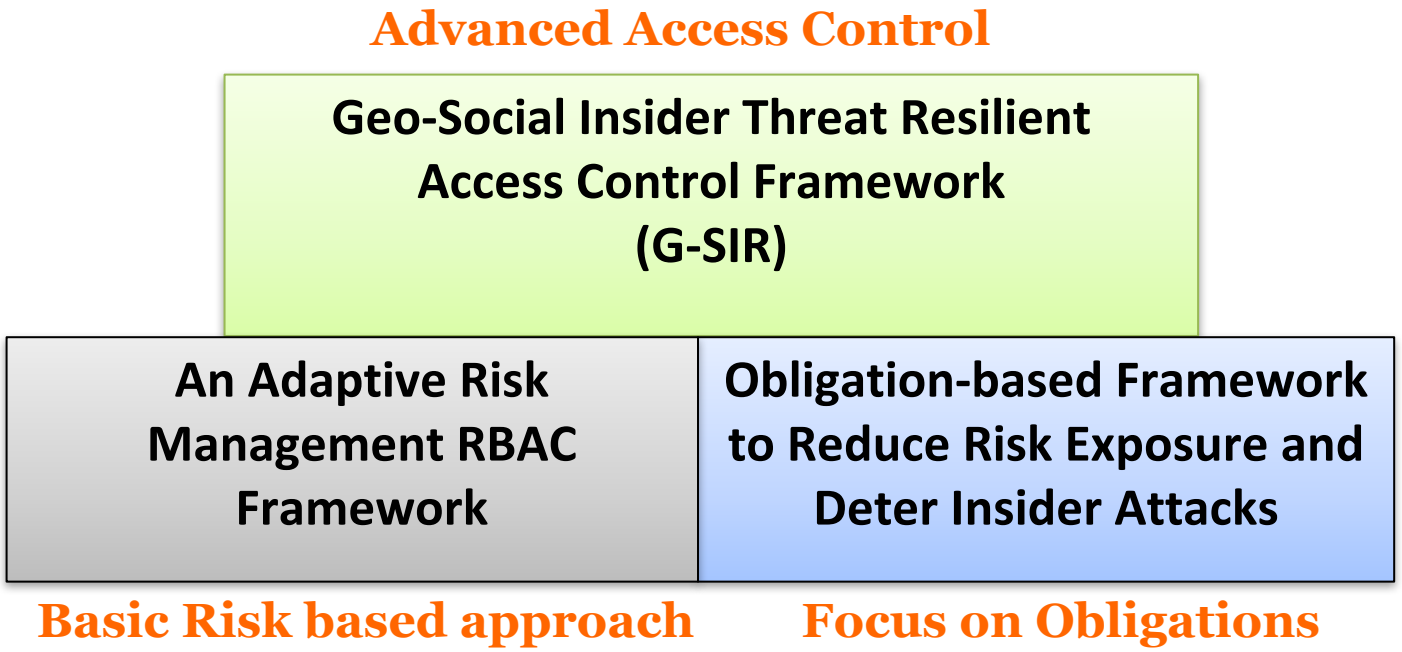
- Statistics show that insider attacks are *typically preceded* by
  - ***technical precursors*** and
  - *psychological precursors*

Our Research

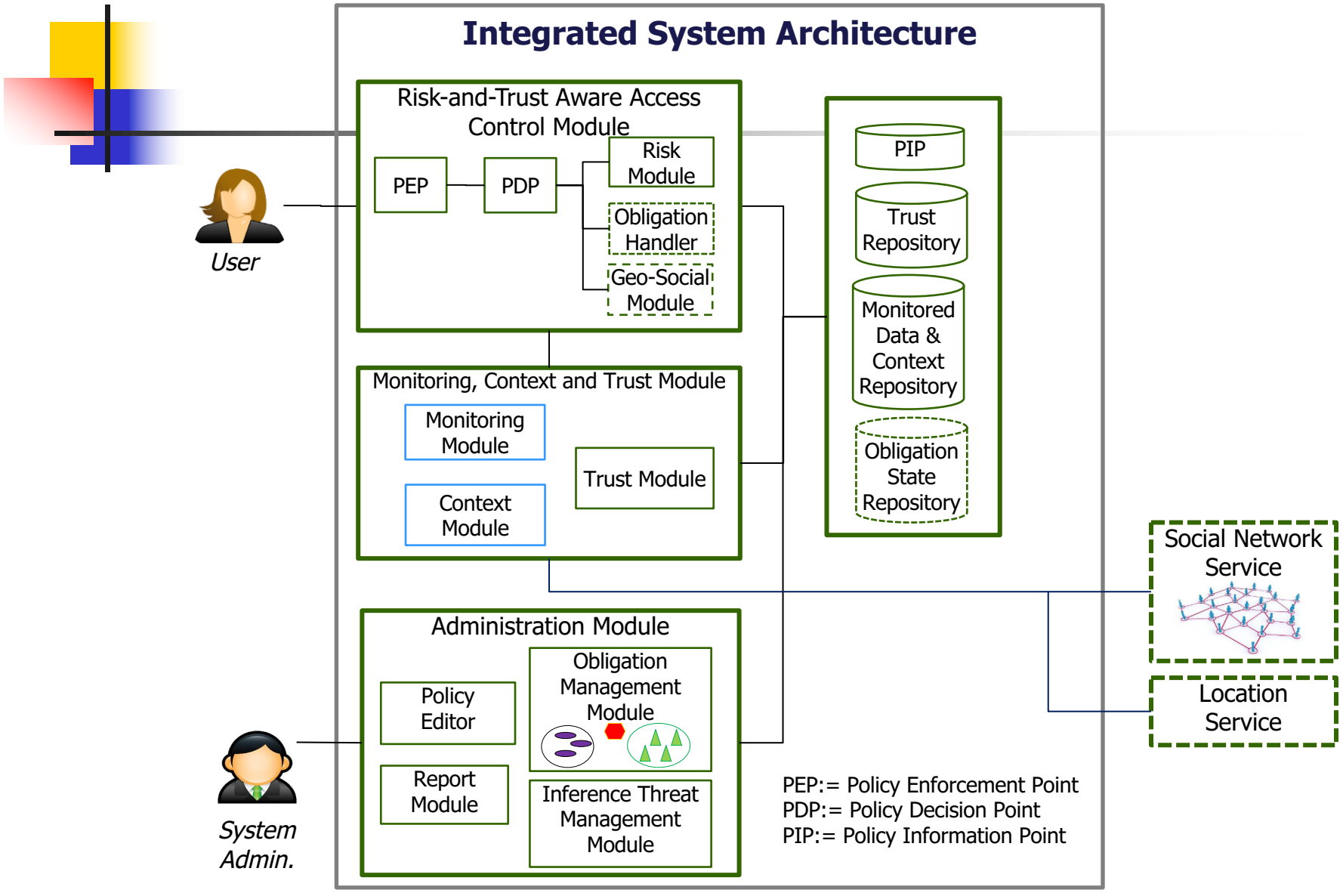
Control risk for each access request automatically ☺

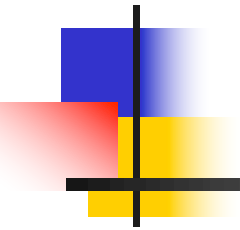
- Utilize wo concepts:
  - **Trust:** expectation of future behavior based on the history
  - **Risk:** likelihood of a hazardous situation and its consequences if it occurs
- We include **risk** and **trust** in access control systems to adapt to *anomalous* and *suspicious* changes in users' behavior

Access Control for Insider Threat Mitigation



Joint work with Dr. Nathalie Baracaldo, IBM Almaden Research (PhD Thesis) & Prof. Balaji Palaniamy


Integrated System Architecture
Risk-and-Trust Aware Access Control Module
PEP
PDP
Risk Module
Obligation Handler
Geo-Social Module
User
PIP
Trust Repository
Monitored Data & Context Repository
Obligation State Repository
Monitoring, Context and Trust Module
Monitoring Module
Context Module
Trust Module
Social Network Service
Location Service
Administration Module
Policy Editor
Obligation Management Module
Report Module
Inference Threat Management Module
System Admin.
PEP:= Policy Enforcement Point
PDP:= Policy Decision Point
PIP:= Policy Information Point

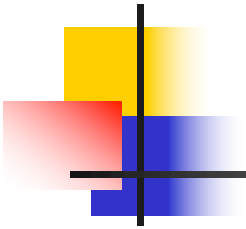


Framework I

An Adaptive Risk Management RBAC Framework

Nathalie Baracaldo, James Joshi "An Adaptive Risk Management and Access Control Framework to Mitigate Insider Threats" *Computers & Security*. 2013.(Journal)

Nathalie Baracaldo, James Joshi "A Trust-and-Risk Aware RBAC Framework: Tackling Suspicious Changes in User's Behavior" *ACM Symposium on Access Control Models and Technologies (SACMAT),* Newark, USA. 2012.



Requirements

1. Enforce *separation of duties* (SoD) and cardinality constraints
2. Detect *suspicious activities*, and establish a trust level for each user
   - Different trust values for users depending on the *context*
3. Different *permissions* may have different risks associated with them
   - Adapt to suspicious changes in behavior of users by restricting permissions depending on *risk* values
4. Risk exposure should be *automatically* reduced, minimizing the impact of possible attacks

In a nutshell…

role

permission

trust_threshold(role)

authorized(*u*,role) & trust(u,c)≥trust_threshold(role)

Trust value of users

- Each user *u* is assigned a trust value:
  - $0 \leq trust(u,c) \leq 1$ → reflects his **behavior**
  - Where *c* is the context, and *u* is the user
- Prior work exists to calculate this value

Assigning risk to permissions

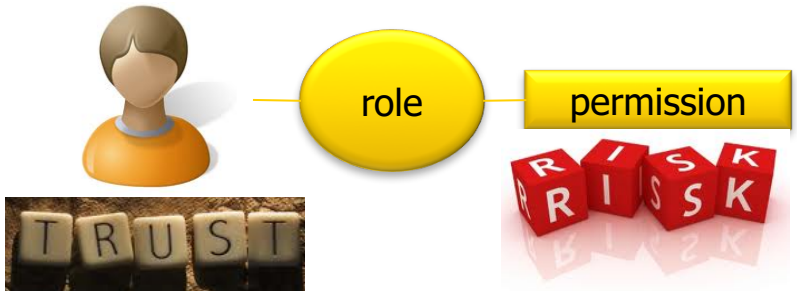
- Each permission is assigned a risk value according to:
  - The *context*
  - The *likelihood* of misuse
  - The *cost* of misuse

DEFINITION 1. *The risk of permission* $p = \langle obj, act \rangle \in P$ *in context* $c \in C$, *written as* $rs(p, c)$, *is defined as follows:*

$$rs(p, c) = \sum_{x_p \in MaliciousUsage} Pr[x_p |\ c\ ] * \mathcal{C}(x_p)$$

Risk of roles

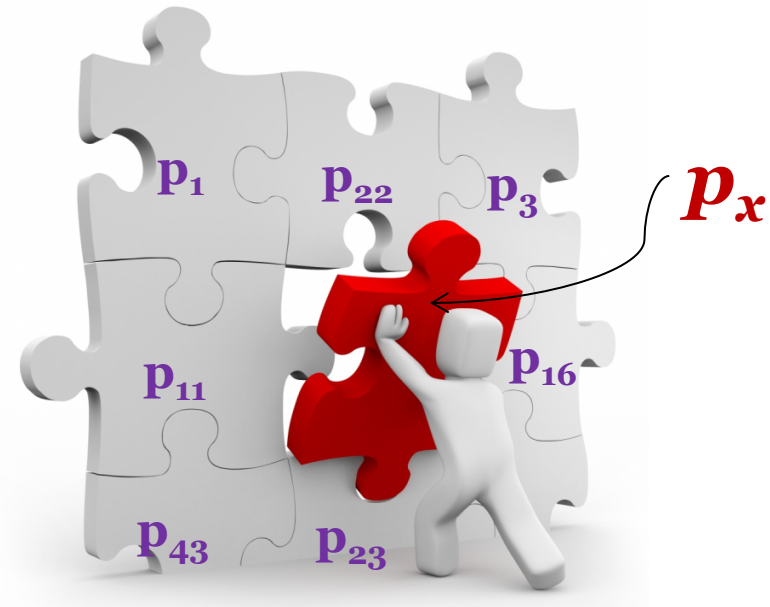
- The risk of *activating* a set of roles depends on:
  - *Context*
  - The *user* that is going to activate the roles
  - Authorized *permissions* & their *risk*
  - *Inference risk*



Inference risk

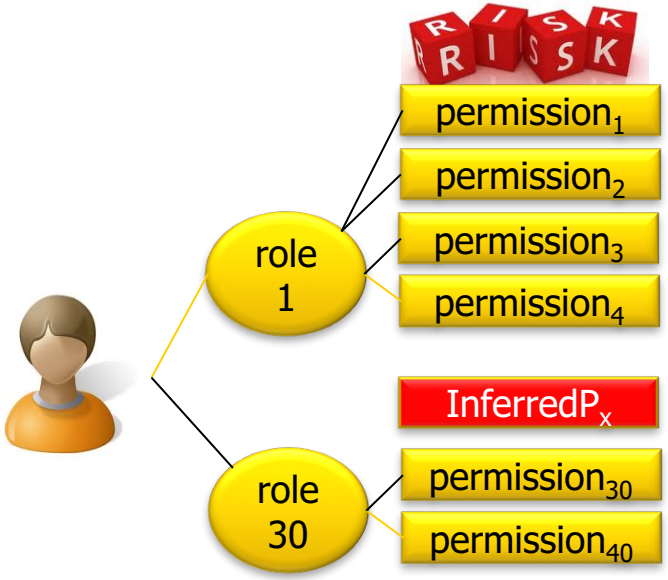
- *Inference Threat:* exists when a user is able to infer unauthorized sensitive information through what seems to be innocuous data he is authorized for
- *Inference tuple:*
  <*PS*, p_x>
  Shows the minimum information needed (*PS*) to infer p_x

  Colored Petri-net for analysis



Risk of roles

- Risk exposure of activating a set of roles



- For a set of roles *RS*, the **trust threshold** is the normalized version of their risk

Reduction of risk exposure

- **Select roles** with **minimum risk** that also respect the policy constraints & provide the requested permissions
  - Role activation algorithm based on this

DEFINITION 3. *The Trust-and-Risk Aware Role Activation Optimization Problem for a query* $q = \langle u, PS, c \rangle$, *consists of finding a solution,* R_q, *such that:*

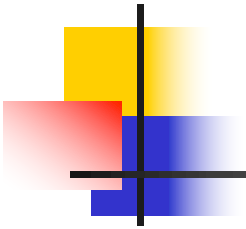
$$\min_{R_q \subseteq authorized(u)} rs(R_q, c, u)$$

$$\textbf{s.t.} \ \forall\ dsod(RS_i, k_i) \in DSoD \ : |R_q \cap RS_i| < k_i$$

$$\forall\ card(r_c, k) \in CARD \wedge r_c \in R_q : activated(r_c) + 1 \leq k - 1$$

$$trust(u, c) \geq \tau(R_q, c, u)$$

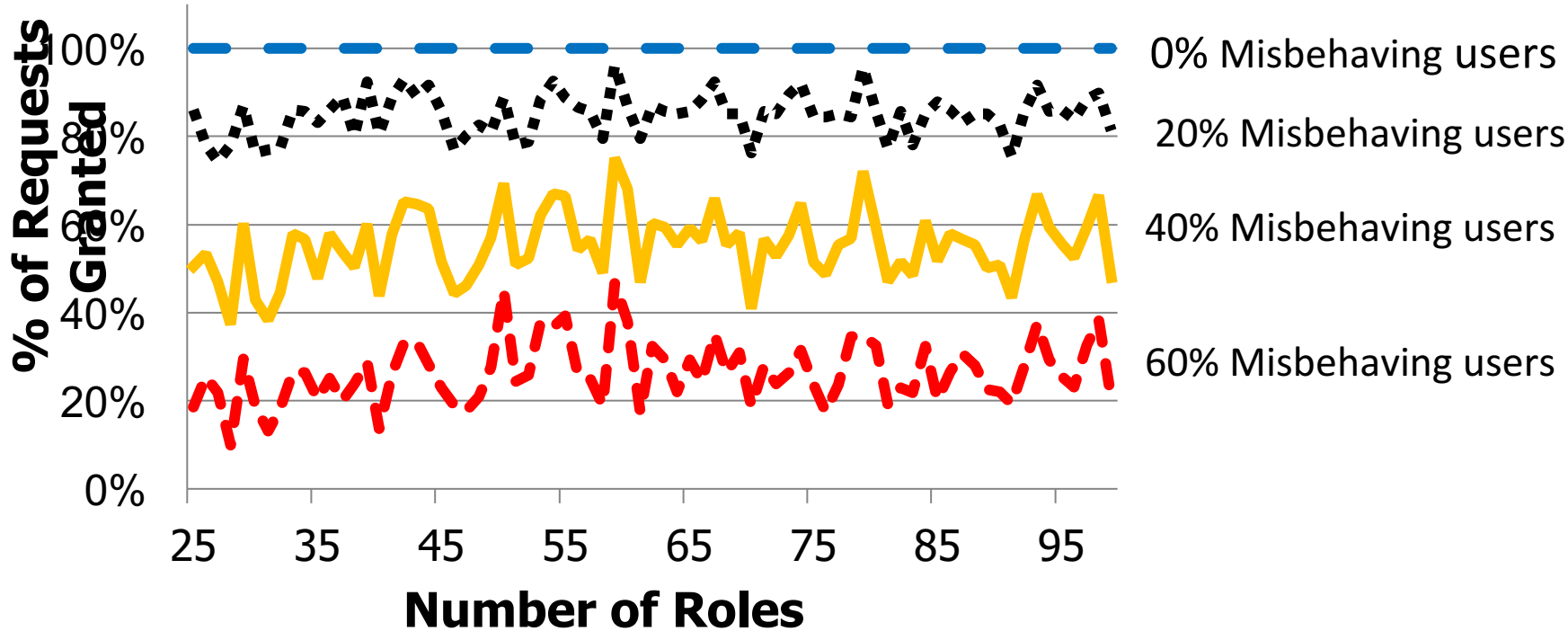
$$P_{au}(R_q) \supseteq PS$$



Experimental Setup

- Generate synthetic *well-formed policies*
- Each point represents the average time of running the algorithm for 30 different policies
- Evaluated the proposed algorithm under two different heuristics for several types of policies

Granted requests for different percentage of misbehaving users



Critical accesses are denied preventing possible attacks

Framework II

Obligation-based Framework To Reduce Risk Exposure And Deter Insider Attacks

Nathalie Baracaldo, James Joshi "Beyond Accountability: Using Obligations to Reduce Risk Exposure and Deter Insider Attacks" *ACM Symposium on Access Control Models and Technologies (SACMAT),* Amsterdam, The Netherlands. 2013.

Motivation

- Many application domains require the inclusion of **obligations** as part of their access control policies

...

A *posteriori* obligations

- Assigned to **users** when they are granted access, and need to be completed before a deadline
  - In a healthcare environment e.g., after 30 days of accessing a patient's sensitive information, a report needs to be filed
  - The obligation is ***fulfilled*** if it is performed before its deadline (30 days), otherwise it is ***violated***

Managing *a posteriori* obligations is challenging

- Once you grant access to a user, there is **no guarantee** that he will fulfill the associated obligation

Ideally

But this may happen

- *Statistics show that it is not wise to trust users blindly*!

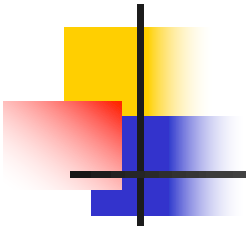
Obligation violation

- Every time an *a* *posteriori* obligation is assigned to a user, there is some risk of non-fulfillment
- The ***risk exposure*** depends on the impact of not fulfilling the obligation
  - Delays on the operation
  - Fines
  - Loss of good will
  - Lawsuits

Current Approaches…

- Accountability
- Provision resources necessary to fulfill obligations

- But they ignore that users may *misbehave* and can't blindly be trusted to fulfill *a posteriori* obligations!



Requirements

- Reduce the **risk** exposure caused by *a posteriori obligations*
- Identify the **trust** value of a user based on the pattern of fulfillment of *a posteriori* obligations
- Identify **policy misconfigurations**
- Identify when a user is **likely** to become **an insider** attacker, without invading users' privacy

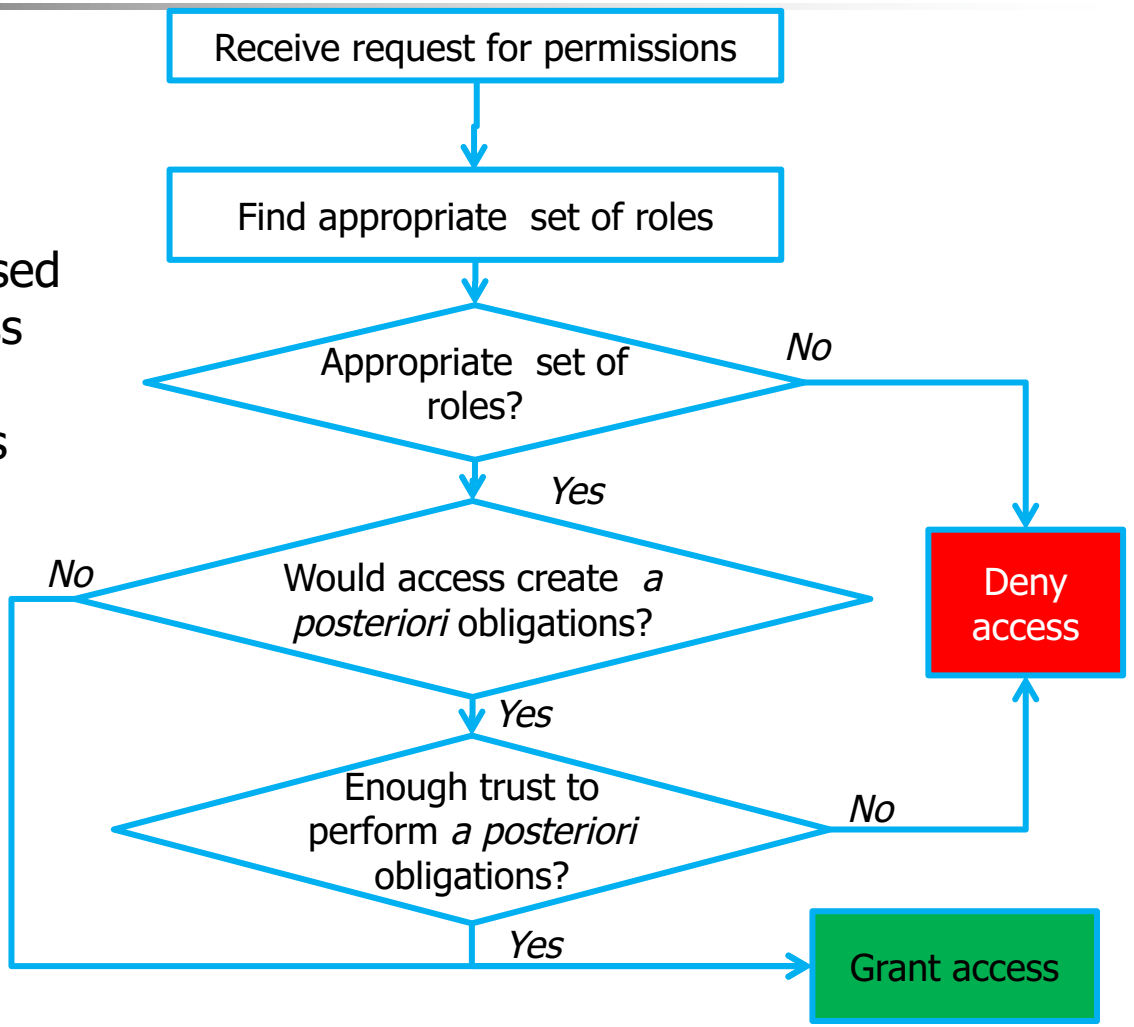
Criticality of Obligations

- Criticality represents the **severity** of not fulfilling an obligation for the organization

- We use the criticality as a **threshold** to determine how much a user needs to be **trusted** in order to be assigned the obligation

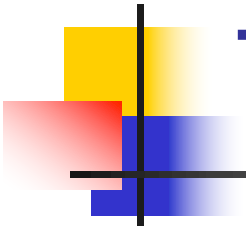
System Overview

- We use standard RBAC
- However, ourtrust approach can be used for any other access control model that includes obligations



Why can we identify suspicious insiders through obligations?

- Psychological precursors: **disregard of authority** and **lack of dependability**
  - Decrease in productivity and rate of fulfilled tasks (obligations)
- The **lack of fulfillment of obligations** is used as an **indicator**



Threat model

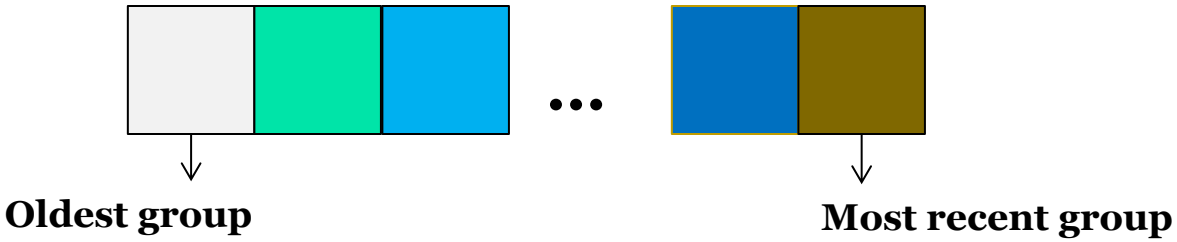
- We consider two types of users:
  - *Naïve users:* don't know how the system works
  - *Strategic users:* know about the system's mechanisms to compute trust values. May try to maintain their trust levels within the expected thresholds
- Both types of users know they are being monitored

How trusted is a user?

- Current behavior
- Historic behavior
- Sudden changes in behavior
- His behavior with respect to his ***peers***

Trust computation

- An observation of a user's behavior is:
  <obligation, (fulfilled |violated)>
- We group observations based on when they are generated



- The most recent group reflects the current behavior

Raw trust of an *observation group*

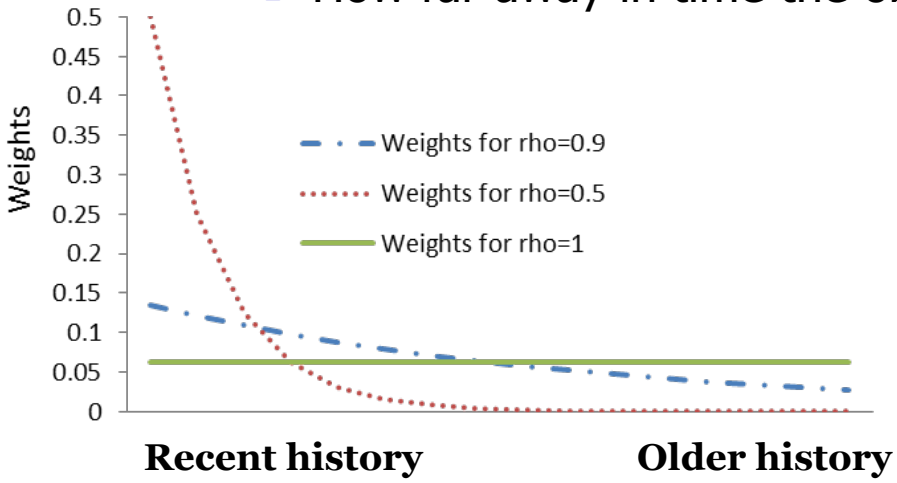
- Weighted average of:
  - The number of obligations fulfilled
  - *over* the total number of obligations assumed by the user
- The *weight* is provided by the *criticality* of each obligation
  - To avoid attacks from strategic users

DEFINITION 3. *The* raw trust $RT_u[T]$ *of user* u *in observation group* T *is calculated using the following expression:*

$$RT_u[T] = \frac{\sum_{b \in \mathcal{B}} b.\varphi * m(GB_u^T, b)}{\sum_{b \in \mathcal{B}} b.\varphi * m(GB_u^T, b) + \sum_{b \in \mathcal{B}} b.\varphi * m(BB_u^T, b)}$$

Historical trust

- Based on previous observation groups
- Weighted average of the *raw trust* of each group
  - The weight of each raw trust of a group depends on:
    - How **critical** the obligations in each group are
    - How far away in time the observation group occurred



DEFINITION 4. *The* historical trust *of user u for observation group T_n, $H_u[T_n]$, is computed as follows:*

$$H_u[T_n] = \sum_{k=1}^{n-1} RT_u[T_{n-k}] * w_k$$

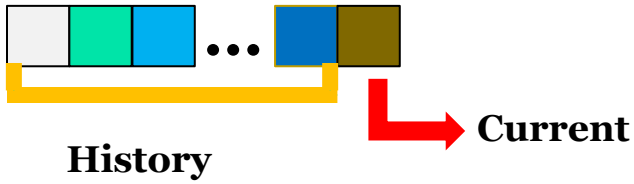
where w_k is the weight of observation group T_{n-k} which is calculated as follows:

$$w_k = \frac{\rho^{k-1} + totalRisk(u, T_{n-k})}{\sum_{i=1}^{n-1}(\rho^{i-1} + totalRisk(u, T_{n-i}))}$$

where $0 \leq \rho \leq 1$.

Trust fluctuation

- Difference between current raw trust and historical trust



- Positive difference: user improved his behavior ☺
- Negative: his behavior worsened ☹

DEFINITION 5. *The* trust fluctuation $D_u[T]$ *of user* u *in observation group* T *is defined as follows:*

$$D_u[T] = RT_u[T] - H_u[T]$$

which represents the variation of the current trust with respect to the historical trust.

Group Drift and Penalty

- When a user is the only one to violate an obligation his group drift is 1
- That deviation should be penalized!

Identify a black sheep!

DEFINITION 6. *The* group drift, $G_u^b[T]$, *of obligation* $b \in \mathcal{B}$ *for user* u *in observation group* T *is defined as follows: If* $m(BB_u^T, b) = 0$, *then* $G_u^b[T] = 0$. *Otherwise:*

$$G_u^b[T] = \frac{m(BB_u^T, b)}{m(TBB^T, b)} - \frac{m(BB_u^T, b) + m(GB_u^T, b)}{m(TBB^T, b) + m(TGB^T, b)}$$

Obligation-based trust

- Finally, we combine the components to find the trust of a user:
  - ✓ Current behavior (raw trust of last group)
  - ✓ Historic behavior
  - ✓ Sudden changes in behavior
  - ✓ His behavior with respect to his peers

DEFINITION 8. *The* individual obligation-based trust $trust(u, T_n)$ *of user* u *in observation group* T_n *is calculated as follows:*

$$trust(u, T_n) = \begin{cases} trust(u, T_{n-1}) & if\ \gamma(D_u[T_n]) = 0 \\ 0 & if\ \mathcal{T} \leq 0 \\ \mathcal{T} & otherwise \end{cases}$$

where
$\mathcal{T} = \alpha \times RT_u[T_n] + \beta \times H_u[T_n] + \gamma \times (D_u[T_n]) - PG_u[T_n]$
and $\alpha + \beta + \gamma = 1$.

Administration Module

- Identify policy misconfigurations/ users colluding
  - Several people not fulfilling the same obligations
- Outliers: users that may require further monitoring (higher risk)

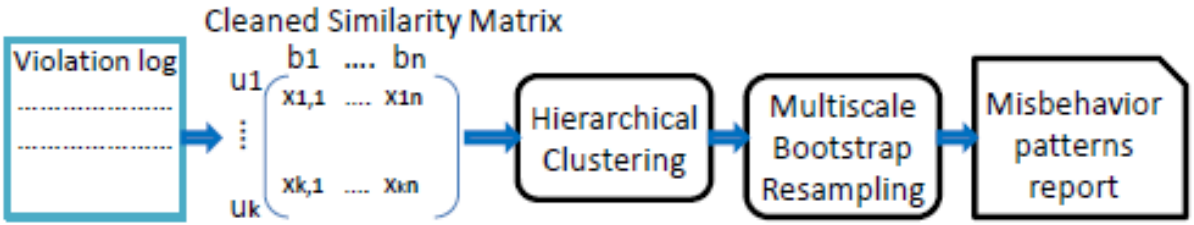
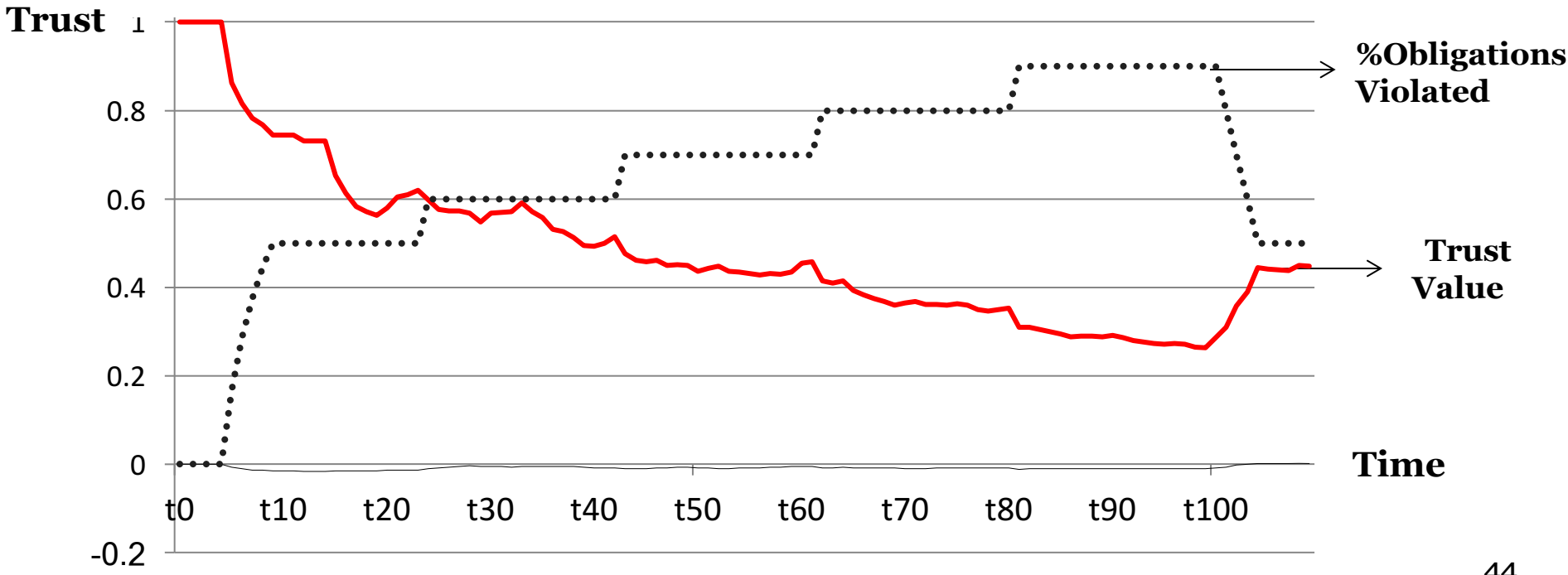


Figure 4: Procedure to find the patterns of misbehavior.

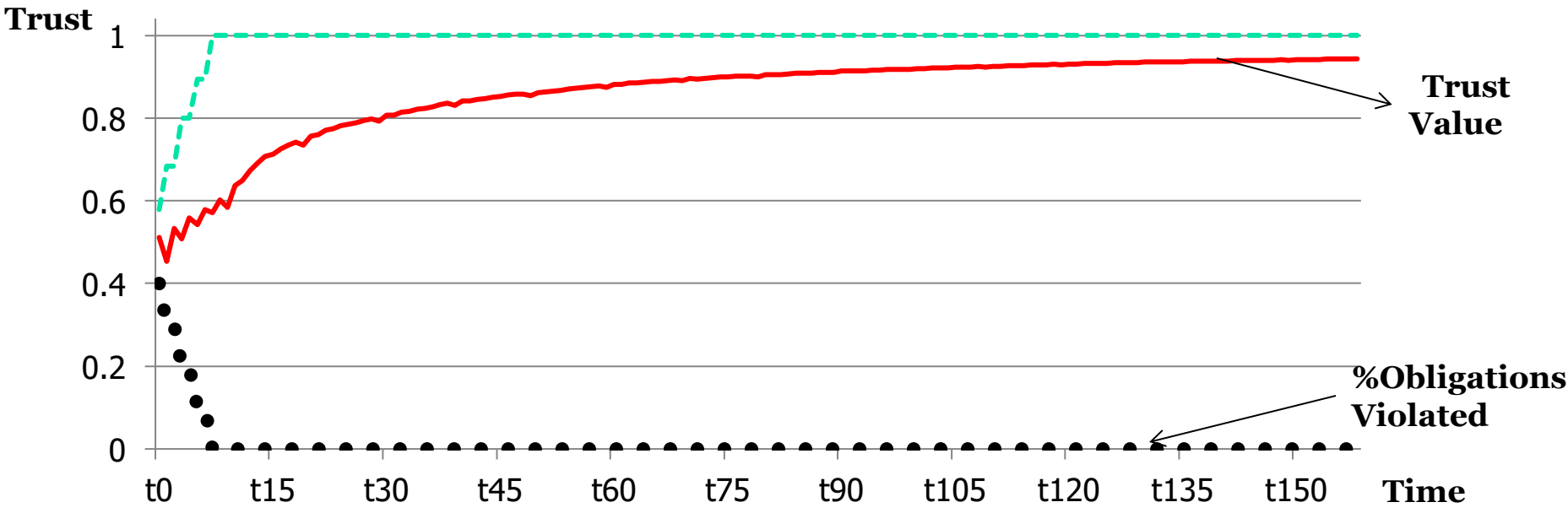
Some Evaluation Results (Framework II)

- Quick to stop damage



Some Evaluation Results (*cont.*)

- Slow to recover trust ☺



Framework III

G-SIR -An Insider Attack Resilient Geo-Social Access Control System

Nathalie Baracaldo, Balaji Palanisamy, James Joshi, G-SIR: An Insider Attack Resilient Geo-Social Access Control Framework, IEEE Transactions on Dependable and Secure Computing (**Accepted**)

Motivation

- Access to users' whereabouts and social interactions
- Location and social relations information can be used as *context* to determine how users may **access** information or resources in a secure way
- For the most part, *social contracts* are currently used to regulate geo-social behavior

G-SIR Policy

role

Associated with a constraint vector

(1)  **Spatial scope**

(2) 

Geo-social contracts

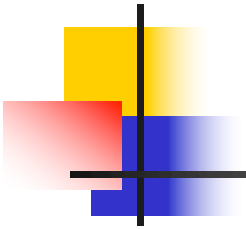
(3)  **Enabling constraints**

(4)  **Inhibiting constraints**

(5)  **Geo-social traces**

(6) Geo-social Obligations

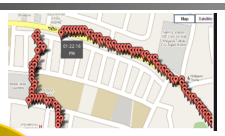
A role can be **activated** iff:
1) all its constraints are fulfilled
2) the risk management procedure allows it!



Key issues

- Geo-Social Contracts
  - Indicate where users should not visit or people user should not interact with/visit
- Enabling and inhibiting constraints
  - Collusion free enabling
- Geo-social obligations
- Trace-based constraints

Overview of G-SIR



Monitoring
- Technical indicators
- G-SIR compliance logs

Analytics

Access Request

G-SIR policy evaluation
- Geo-social context evaluation
- Risk Management

Access control decision

So far we have considered a single actor: the requester

- Privilege misuse threats:
  The requester becomes rogue
- Who are the users in the vicinity?
  - New actors: *enablers* and *inhibitors*

Requester

Classifying Users in the Vicinity: Social Predicate

- Defines a set of users based on
  - A social graph and labels of social relations
  - We can even use more than one graph (e.g., graphs formed using tweets and retweets)

Is a user part of community *X*?
Are two users friends?
What is their relationship?
Are they connected?

Inhibitors

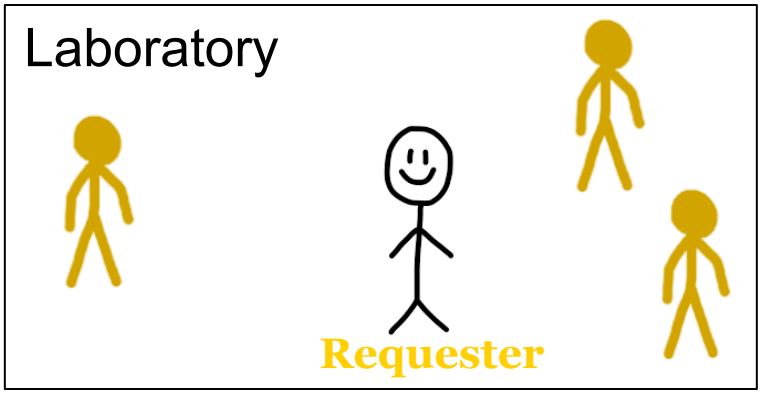
- An ***inhibitor*** is an undesirable user for an access
  - E.g., conflicting project, undesirable community, etc.

- Proximity threats: Insider adversaries who may gain access to information by placing themselves (strategically or opportunistically) *close* to the requester

Enablers

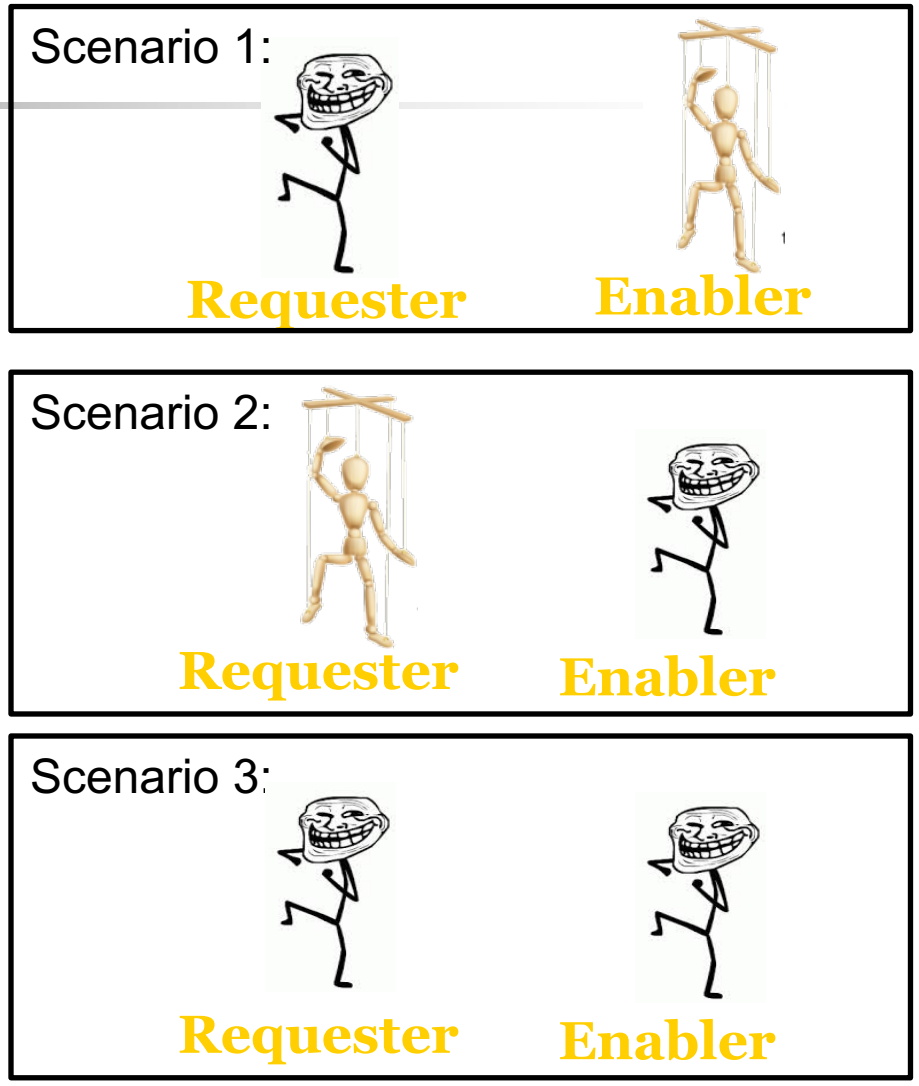
- ***K*** users in the vicinity who validate an access request:
  - bootstrap the trust of a requester ☺

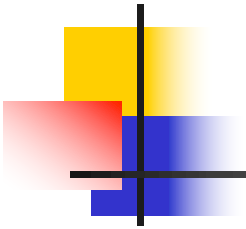
Place:= laboratory
Relationship:= superior
K:=3



Some caveats…

- **Social engineering** Trick the enabler or the requester to enter into a targeted place
- **Collusion threats** Requester and enablers may collude to gain access

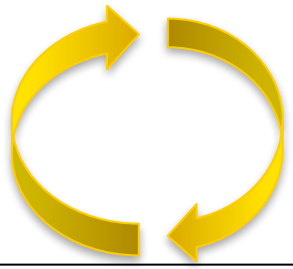




Requirements

- Classify users in the vicinity
- Design policy constraints to capture and prevent undesirable geo-social behavior: geo-social contracts, geo-social obligations and trace-based constraints
- Mitigate the risk of colluding users
- Adapt access control decisions to negative changes in behavior of users

Overview of G-SIR



Monitoring
- Technical indicators
- G-SIR compliance logs

Analytics

Access Request

G-SIR policy evaluation
- Geo-social context evaluation
- Risk Management

Access control decision

G-SIR Policy

role

Associated with a constraint vector

(1)  **Spatial scope**

(2) 

Geo-social contracts

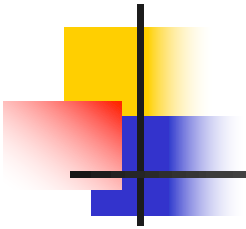
(3)  **Enabling constraints**

(4)  **Inhibiting constraints**

(5)  **Geo-social traces**

(6) Geo-social Obligations

A role can be **activated** iff:
1) all its constraints are fulfilled
2) the risk management procedure allows it!



Key issues

- Geo-Social Contracts
  - Indicate where users should not visit or people user should not interact with/visit
- Enabling and inhibiting constraints
  - Collusion free enabling
- Geo-social obligations
- Trace-based constraints

Geo-Social Contracts

- Places and people that users assigned to the role should not visit

  <<place, *Social_Predicate*>,φ >

  - Is the geo-social contract violated? φ indicates how bad the violation is

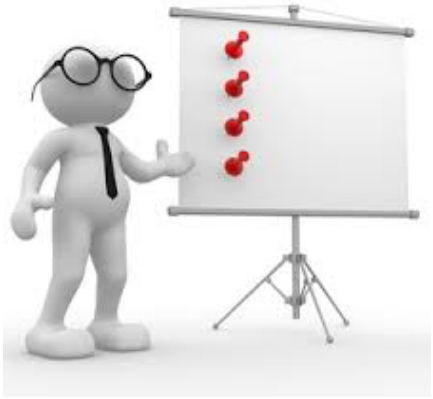
Users assigned to role receptionist should not enter the server rooms:
<<serverRooms, $\perp$ >, 0.8>

Inhibiting Constraints

<context, place, *social_predicate*, α>

- α := minimum confidence level required to classify a user as inhibitor

<laptop, 2FeetRadiusAroundRequester, belongToCommunity(u?,BadGuys),0.95>



<projector, sameRoom, belongToCommunity(u?,BadGuys),0.95>

Collusion-free Enabling Constraints

<*Place, k, social_predicate,* τ_c >

- τ_c := maximum tolerance to collusion
- Collusion-free enforcement:
  - If *PrCollusion*(Enablers ∪ requester) > τ_c, the candidate enablers are rendered untrustworthy

Geo-social obligation

- Geo-social actions that users need to fulfill after they have been granted an access

< dir, duration, φ>

where $dir \in$ {<+visit, place>, <-visit, place>, <+meet, social_predicate>, <-meet, social_predicate>}

φ := criticality of violations

<-meet, belongsToCommunity(u?, Y),1year, 0.5>

Trace-based constraints

- Constraints recent whereabouts

<*lst*, Duration, > where
lst =<<place$_1$, social_predicate$_1$>, ...<place$_k$,social_predicate$_k$>$_n$>
And is the criticality of a violation

  - If a doctor was in a contagious unit, he cannot enter the new born unit in a week
    - Unless you go to a sanitizing facility

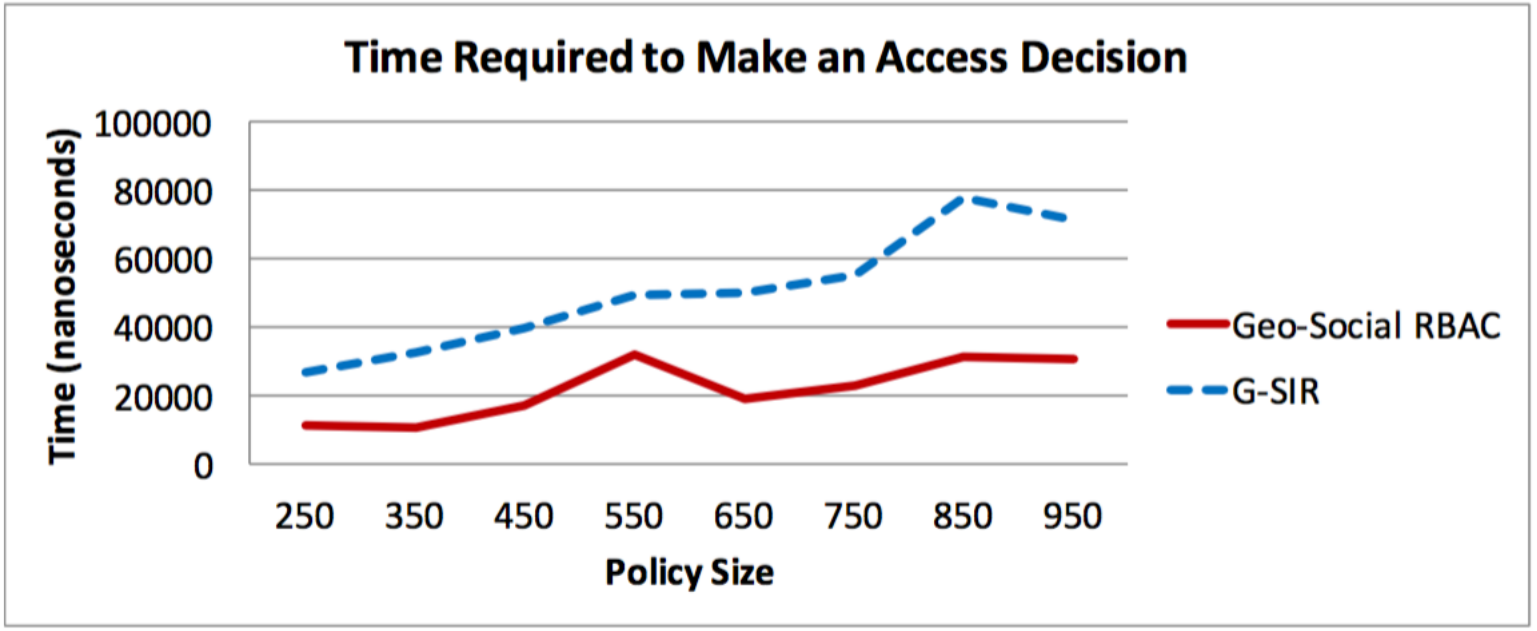
$$w_1 = (\langle \langle Sanitizing\ Facility, in \rangle, \perp \rangle, 15minutes, 0.8),$$

Risk Management: Formulated using utility theory

- Utilities depend on the context and the permissions authorized by an access
- We find a threshold which is compared to the probability of attack

	Grant Access	Deny Access
Attack	☹ Utility depends on the cost of the attack	☺ Thwarted the attack
No attack	☺ Utility depends on the gain from transaction	Based on cost of annoyance

Average time as the policy size increases



Some additional runtime overhead due to the extra verifications performed. However, the overhead is acceptable in comparison to Geo-Social RBAC

Conclusions

- We proposed **three adaptive access control** frameworks that can **reduce the risk of insider threats**

Framework I: Contributions

- Presented a model that includes **risk** and **trust** in **RBAC** to adapt to anomalous and suspicious changes in users' behavior
- Proposed a comprehensive way to **calculate risks** of permissions and roles
  - We introduce the notion of **inference of unauthorized permissions** & formulated a **Colored Petri-net**

Framework I: Contributions (*cont.*)

- We define an **optimization problem** to enforce the policy, **reduce the risk exposure** of the organization, and ensure that all constraints are respected
- We present a **role activation algorithm** to solve the optimization problem and **evaluate** its performance using well-formed policies
- Provide a **simulation** methodology to help identify policies with unacceptable inference risk

Framework II: Contributions

- Proposed a framework that **reduces** the **risk exposure** caused by *a posteriori* obligations
- Presented an **obligation-based trust methodology** that is resistant to naïve and strategic users
  - It can be integrated into any access control model with *a posteriori* obligations (e.g., UCON)

Framework II: Contributions (*cont.*)

- Showed that based on previous work on **psychological precursors** *a posteriori* obligations can be used to identify suspicious users
- Presented an **administration module** to identify patterns of misbehavior, suspicious users and non-updated policies

Framework III: Contributions

- First research effort to analyze geo-social access control systems to thwart insider attacks: We **uncover some novel insider threats**
- We provide an **access control model** to mitigate those threats with novel constraints: *geo-social contracts, geo-social obligations, inhibiting, collusion-free enabling constraints* and *trace-based constraints*
- Show that G-SIR can prevent some insider threats

Limitation and Future Work

- We only deter insider threats that are regulated by the *Policy Enforcement Point*
- We assumed that monitored information was available, but there may be privacy concerns
- As future work a *policy specification* framework needs to be provided
  - Graphical interface
  - User studies

Associated publication

- **Nathalie Baracaldo,** James Joshi "An Adaptive Risk Management and Access Control Framework to Mitigate Insider Threats" *Computers & Security*. 2013.
- **Nathalie Baracaldo,** James Joshi "A Trust-and-Risk Aware RBAC Framework: Tackling Suspicious Changes in User's Behavior" *ACM Symposium on Access Control Models and Technologies (SACMAT),* Newark, USA. 2012.

RBAC

- **Nathalie Baracaldo,** James Joshi "Beyond Accountability: Using Obligations to Reduce Risk Exposure and Deter Insider Attacks" *ACM Symposium on Access Control Models and Technologies (SACMAT),* Amsterdam, The Netherlands. 2013.

Obligations

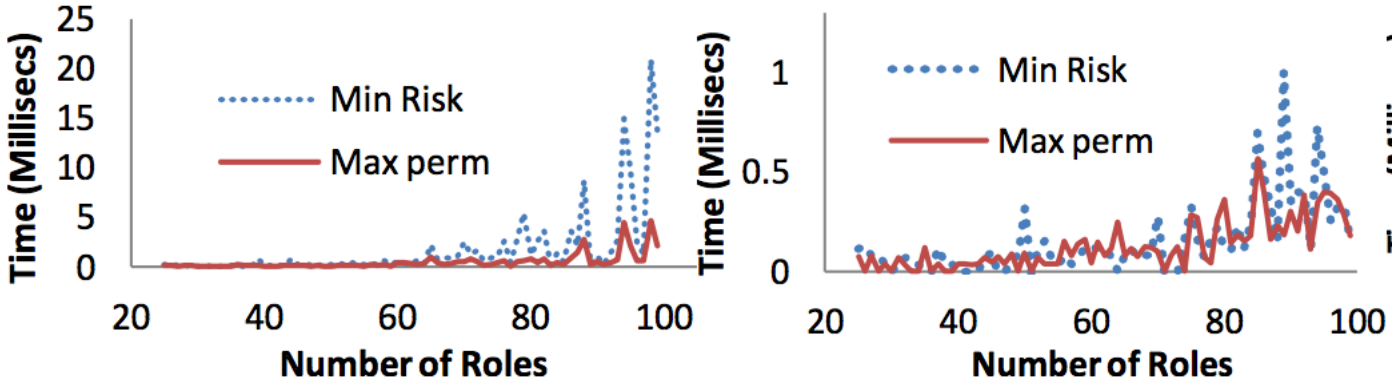
- **Nathalie Baracaldo,** Balaji Palanisamy, James Joshi "Geo-Social-RBAC: A Location-based Socially Aware Access Control Framework" *The 8th International Conference on Network and System Security (NSS 2014)*. 2014.

Geo-social

Questions? Comments? Ideas?

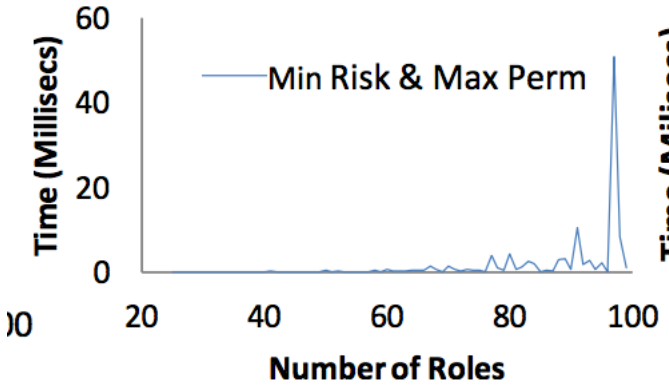
Thanks!

Comparison of two heuristics: *Min risk* & ***Max perm*** for different policy configurations

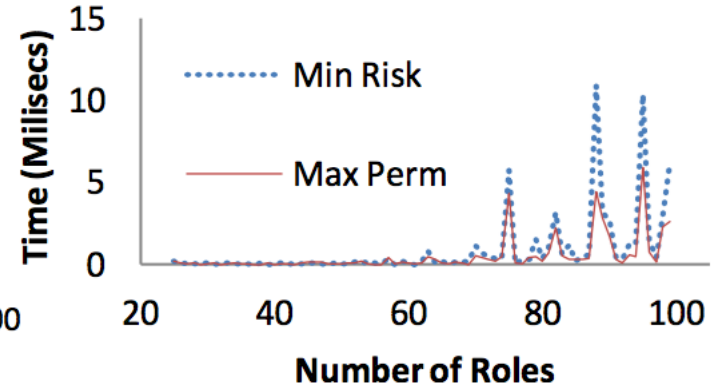


(a) $I{:}A{:}IA = 0{:}0{:}1$

(b) $I{:}A{:}IA = 1{:}0{:}0$



(c) $I{:}A{:}IA = 0{:}1{:}0$



(d) $I{:}A{:}IA = 1{:}1{:}1$

Max perm **heuristic outperforms the** ***Min risk*** **heuristic consistently**

Overview of G-SIR

Monitoring

Monitored Information

- Technical indicators

G-SIR policy compliance logs

Analytics

Colluding Communities

Trust

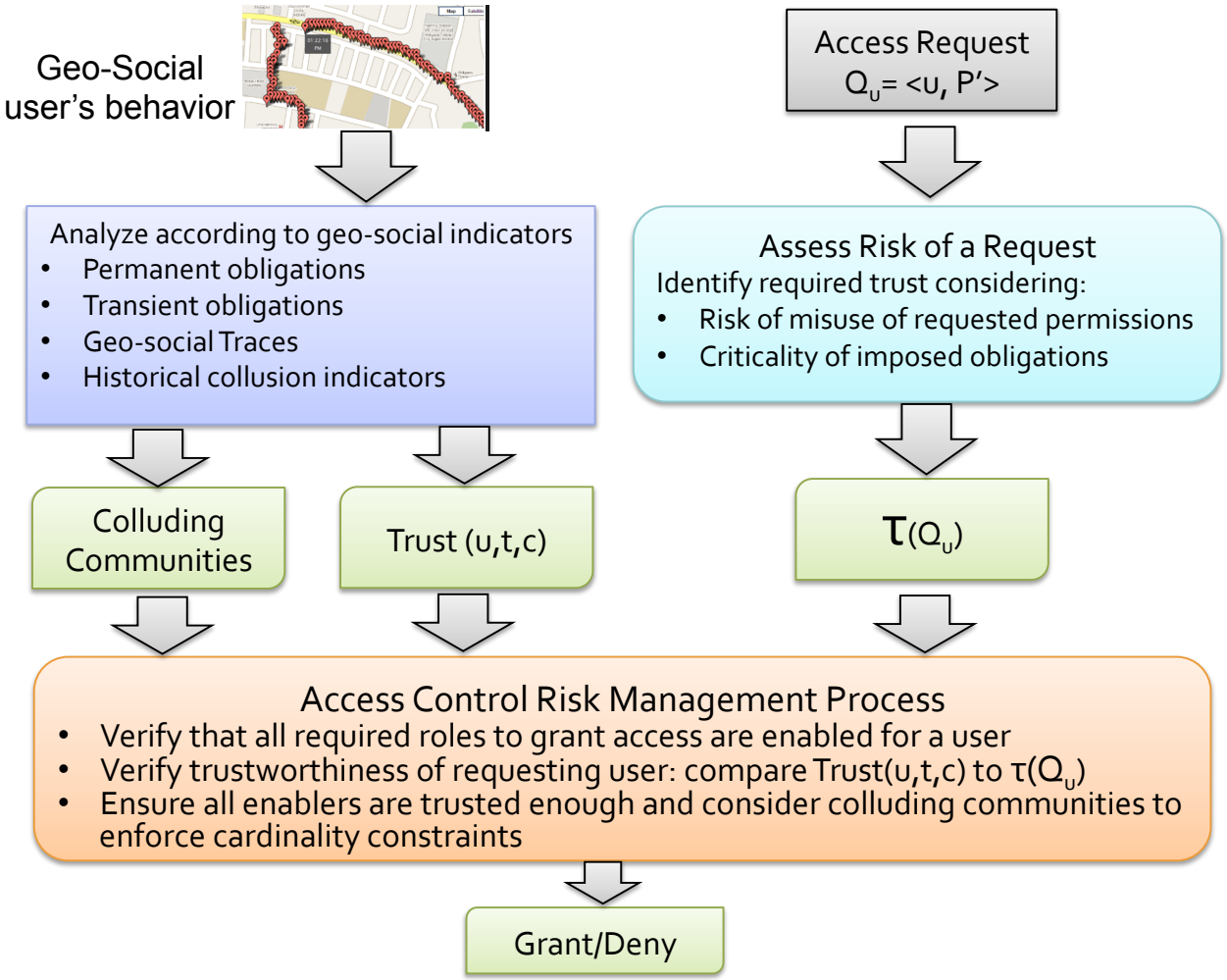
Probability of Attack

Risk management

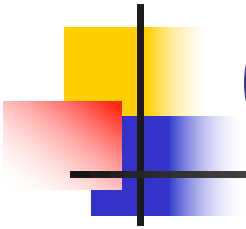
G-SIR policy

Access control decision

Another piece of my *insider threat* research:
An Adaptive Geo-Social Access Control System

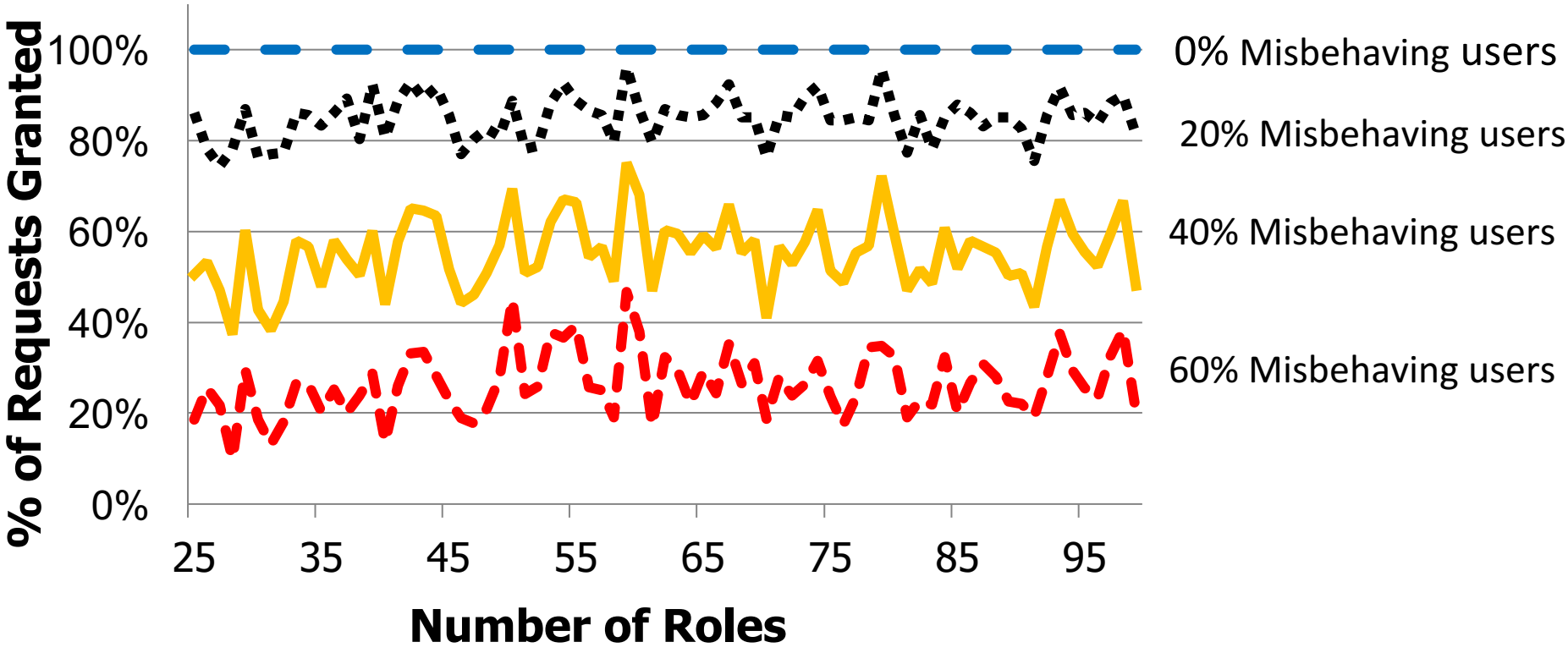


Experimental Setup (Framework 1)



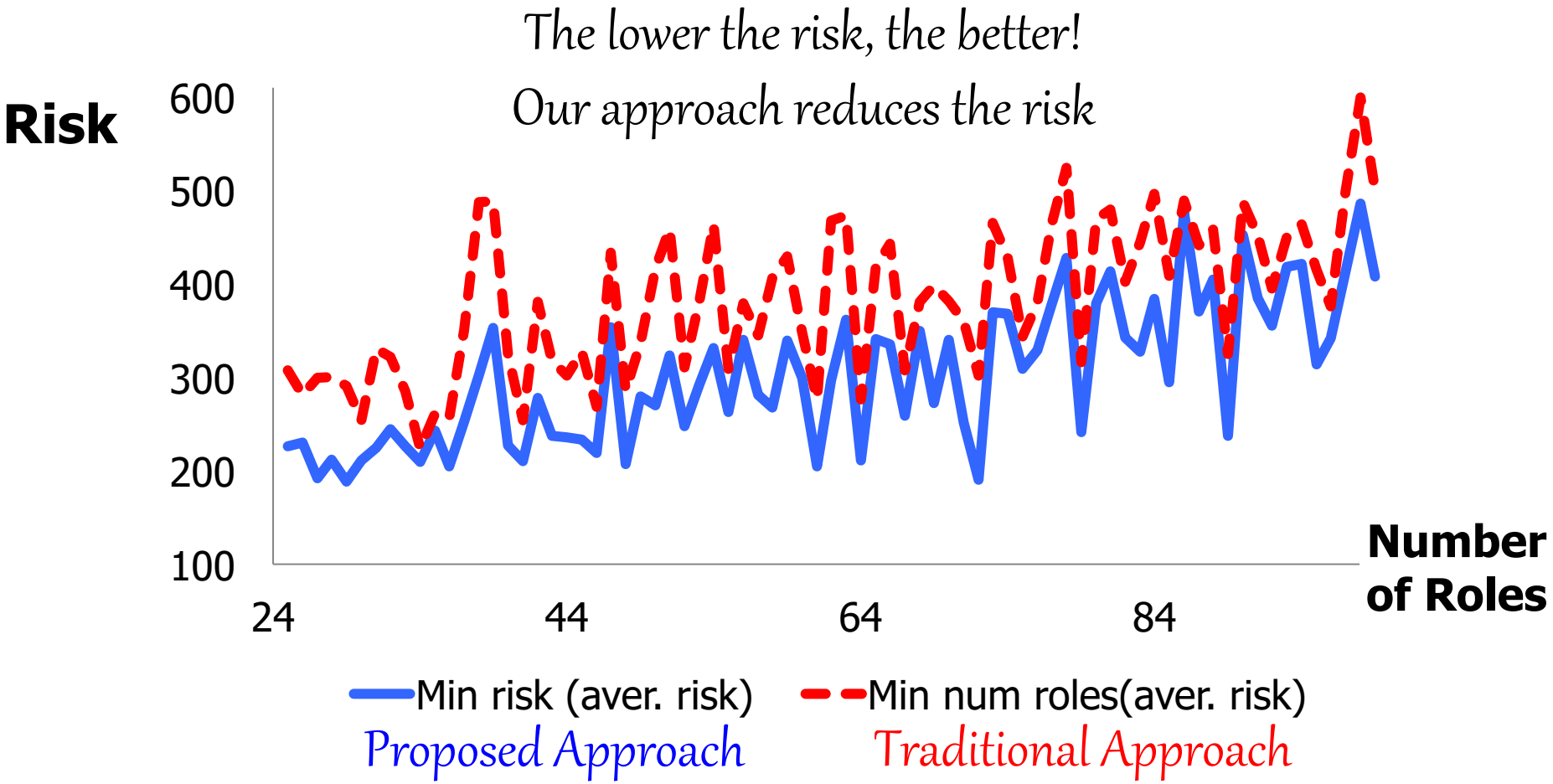
- We generated synthetic *well-formed policies*
- Each point represents the average time of running the algorithm for 30 different policies
- We evaluated the proposed algorithm under two different heuristics for several types of policies

Granted requests for different percentage of misbehaving users



Critical accesses are denied preventing possible attacks

Risk exposure of the proposed system (min. risk) vs. traditional role activation



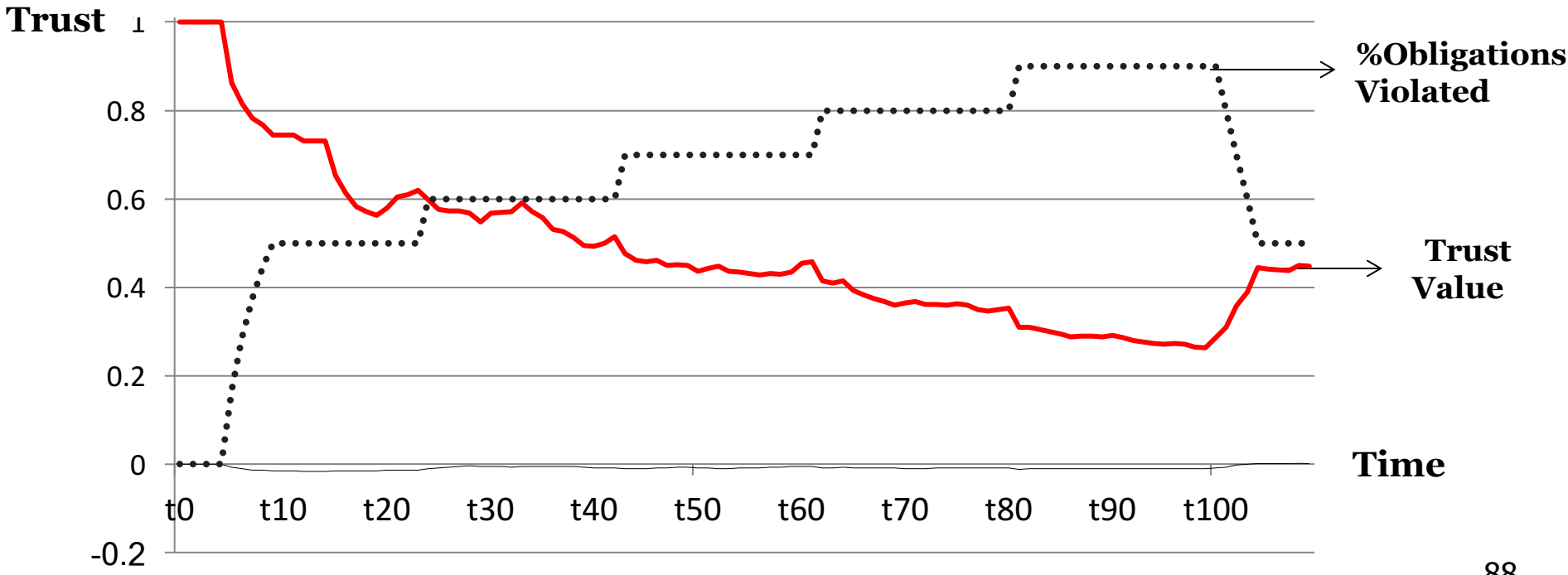
Administration Module: Example Simulation Results

- Managing active inference threats
  - Simulate users' behavior to identify active inference threats and prioritize threat mitigation

Users	Active Inference Threat	Risk Associated	Responsible Inference Tuple	Roles responsible for inference	Average trust of user
u1	p1	Low	<{p2,p4,p8},p1>	r1,r5	0.9
u1	**p3**	**High**	**<{p4,p5,p9},p3>**	**r4,r5,r15**	**0.9**
u1	p69	Low	<{p5,p6},p69>	r5,r20	0.9
u3	**p3**	**High**	**<{p4,p5,p9},p3>**	**r5,r20**	**0.5**
u55	p22	Low	<{p3,p4,p5},p22>	r2,r5	0.3
u122	p50	Medium	<{p1,p8},p50>	r25	0.5

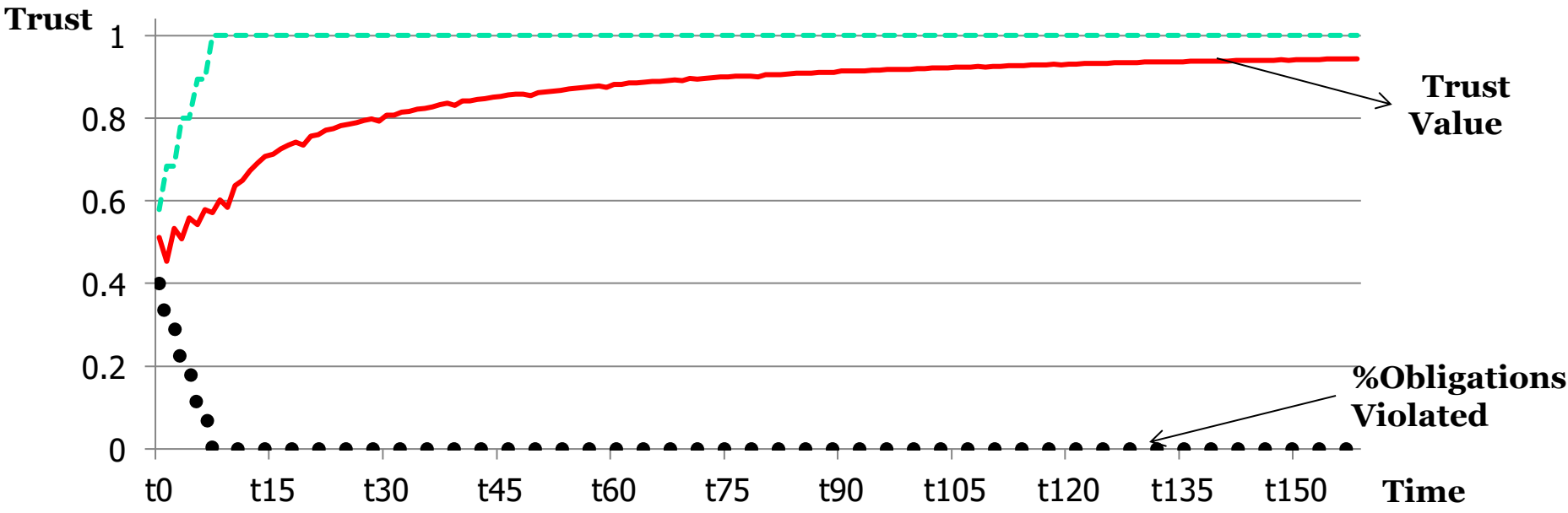
Some Evaluation Results (Framework II)

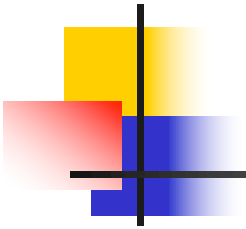
- Quick to stop damage



Some Evaluation Results (*cont.*)

- Slow to recover trust ☺

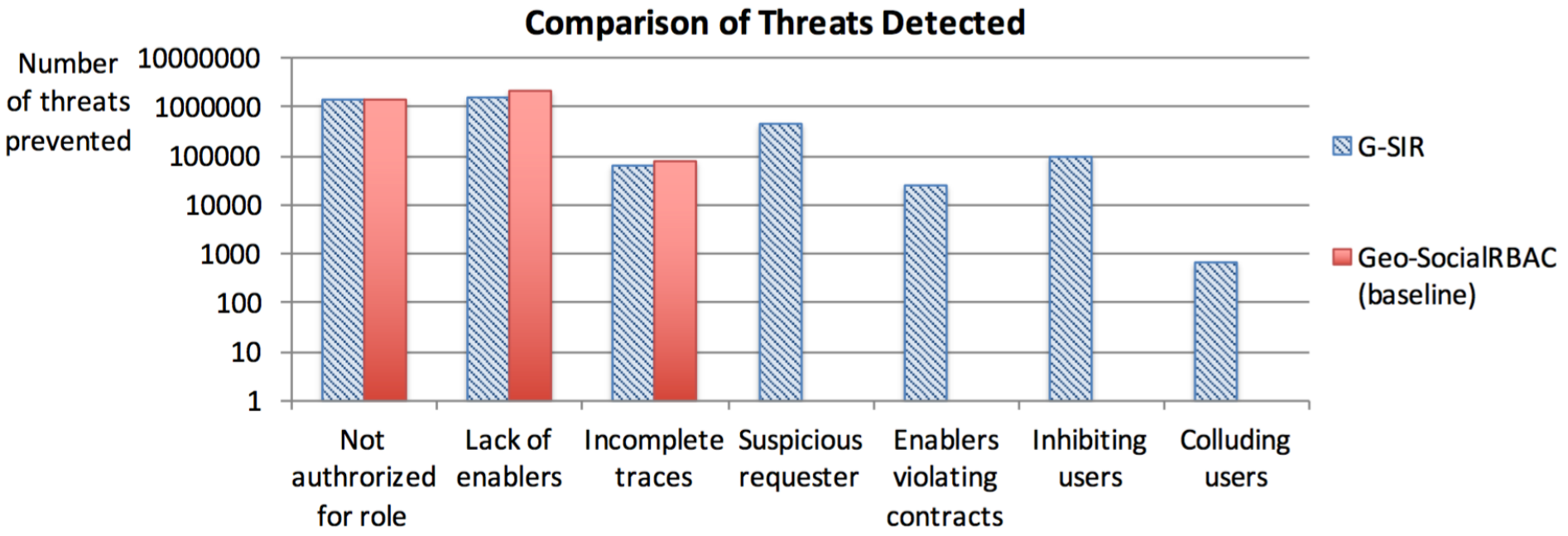




Experimental Setup

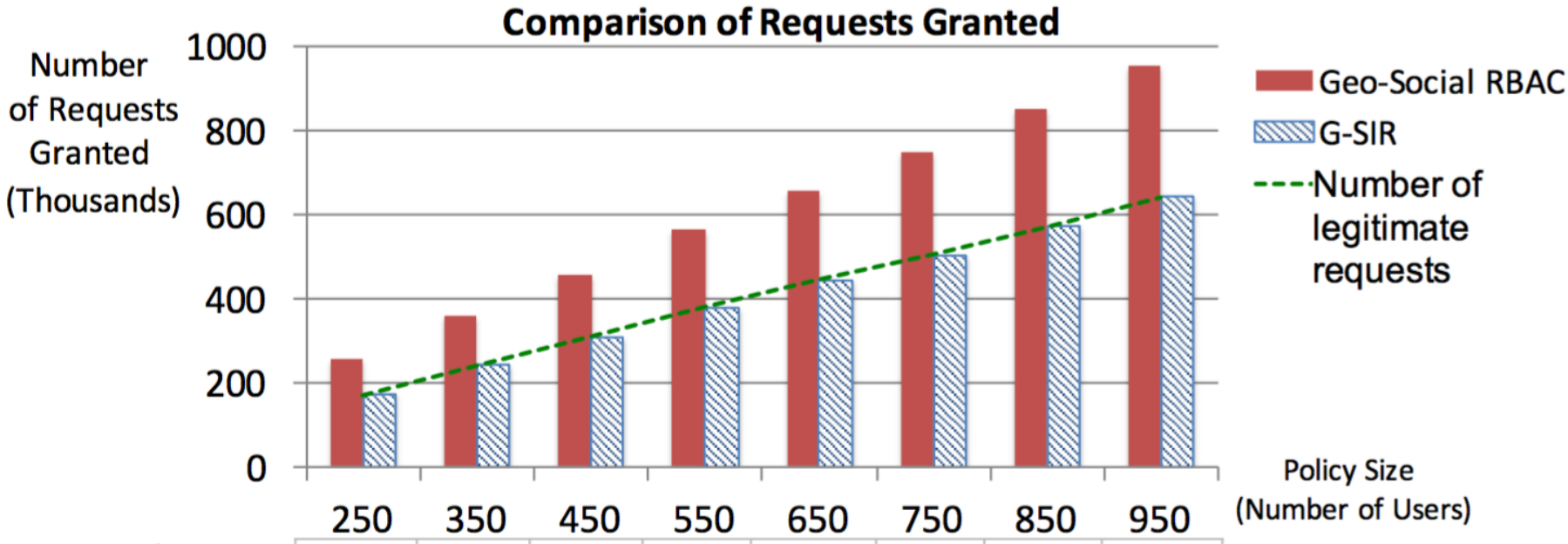
- Mobile simulator written in java
- Users move randomly at every time instant and are related through a random social network
- Random well-formed policy was generated
- If a user stepped into a protected place, an access request for the particular role was generated on his behalf
- Each point represents the average time of running the simulation for 30 different policies

Baseline comparison



G-SIR captures more threats than the baseline

Requests granted



Percentage of requests granted by Geo-Social RBAC successfully denied by G-SIR	250	350	450	550	650	750	850	950
	33%	32%	32%	33%	33%	33%	33%	33%

Inhibiting constraints