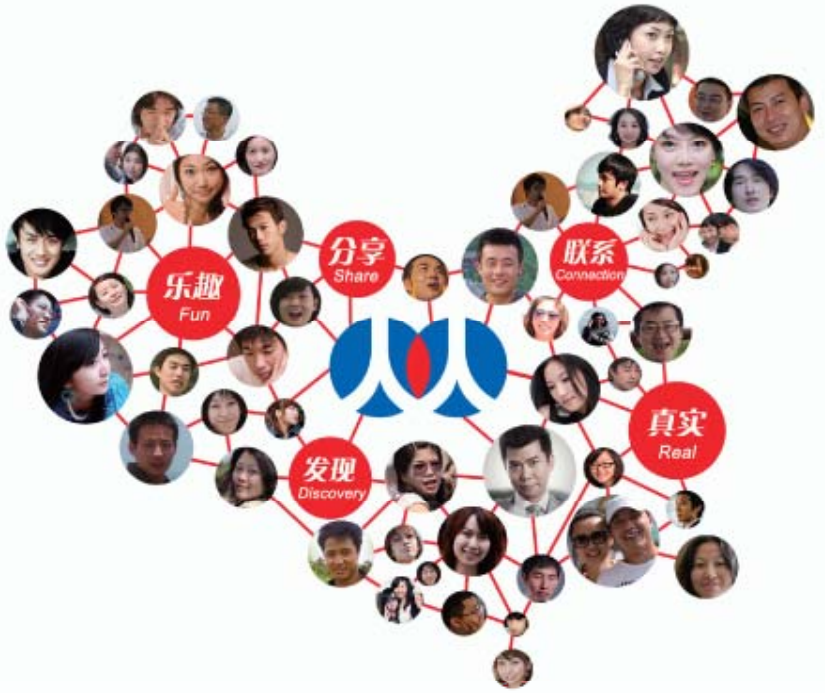
Facebook Immune System

人人安全中心 姚海阔



人人网 中国最大SNS

人人网
renren.com

Immune

- A realtime system to protect our users and the social graph

- Big data, Real time
  - 25B checks per day
  - 650K per second at peak
  - Realtime checks and classifications on every read and write actions

Agenda

- 1 Threats
- 2 Adversarial Learning
- 3 System Design
- 4 Summary

Protect the Graph

- Threats to the social graph can be tracked to three root causes.
  - Compromised accounts
    - return
  - Fake accounts
    - delete
  - Creepers
    - educate

Compromised Accounts

- Phishing
  - The trust is a target for manipulation
  - IP and successive geo-distance
- Malware
  - Target the propagation vector
  - using user feedback
  - require Turing tests*

Fake Accounts

- Created by both scripts and raw labor
  - overcome rate limits
  - boost the reputation or ranking
  - Short lifetime
- Fake accounts have limited virality because they are not central nodes and lack trusted connections
  - Comments and wall posts on pages

Creepers

- Unwanted friend requests
  - Beauty hunter
- Chain letters
  - motivate users to take damaging actions on false pretenses
  - create a global misinformation wall that hides critical or time-sensitive information

Balance of Power

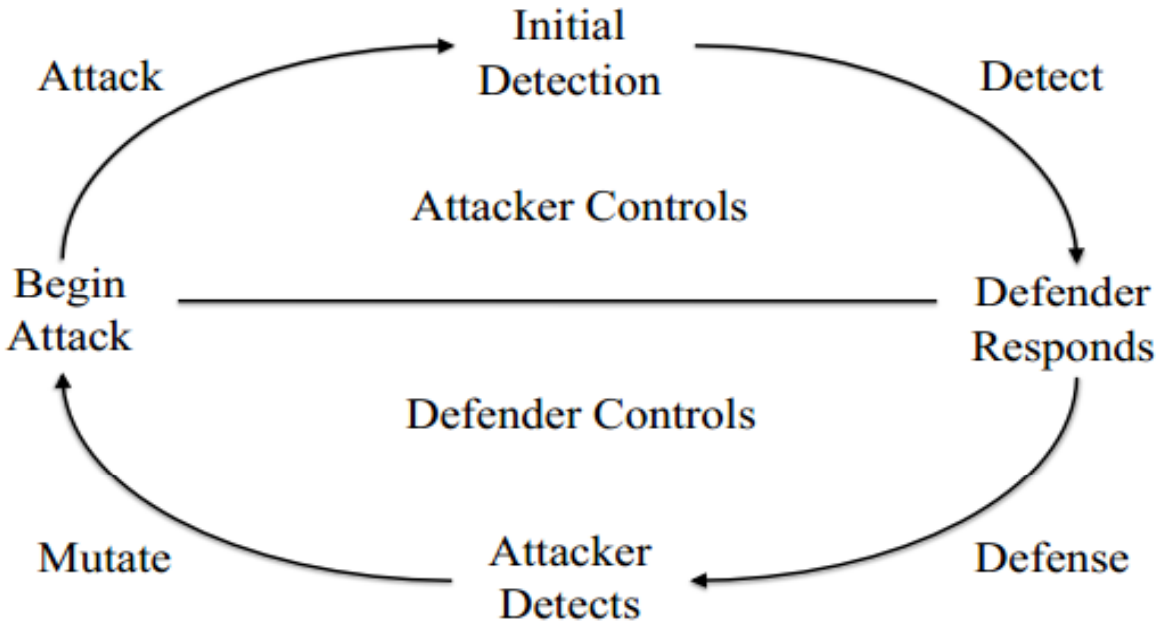
- What the attackers have:
  - Labor. They have much more
  - Distributed botnets, compromised webhosts, infected zombies
  - Fake and compromised objects (events, apps, pages, groups, users, ...)
- What we have:
  - Data centers, content distribution networks
  - User feedback -- spam reports, feed hides, friend rejects
  - Knowledge of patterns and anomalies
  - Shared secrets with users

Adversarial

- Adversarial learning
  - The attacker works to hide patterns and subvert detection
  - The system must respond fast and target the features that are most expensive for the attacker to change
  - not to overfit on the superficial features that are easy for the attacker to change*

Circle

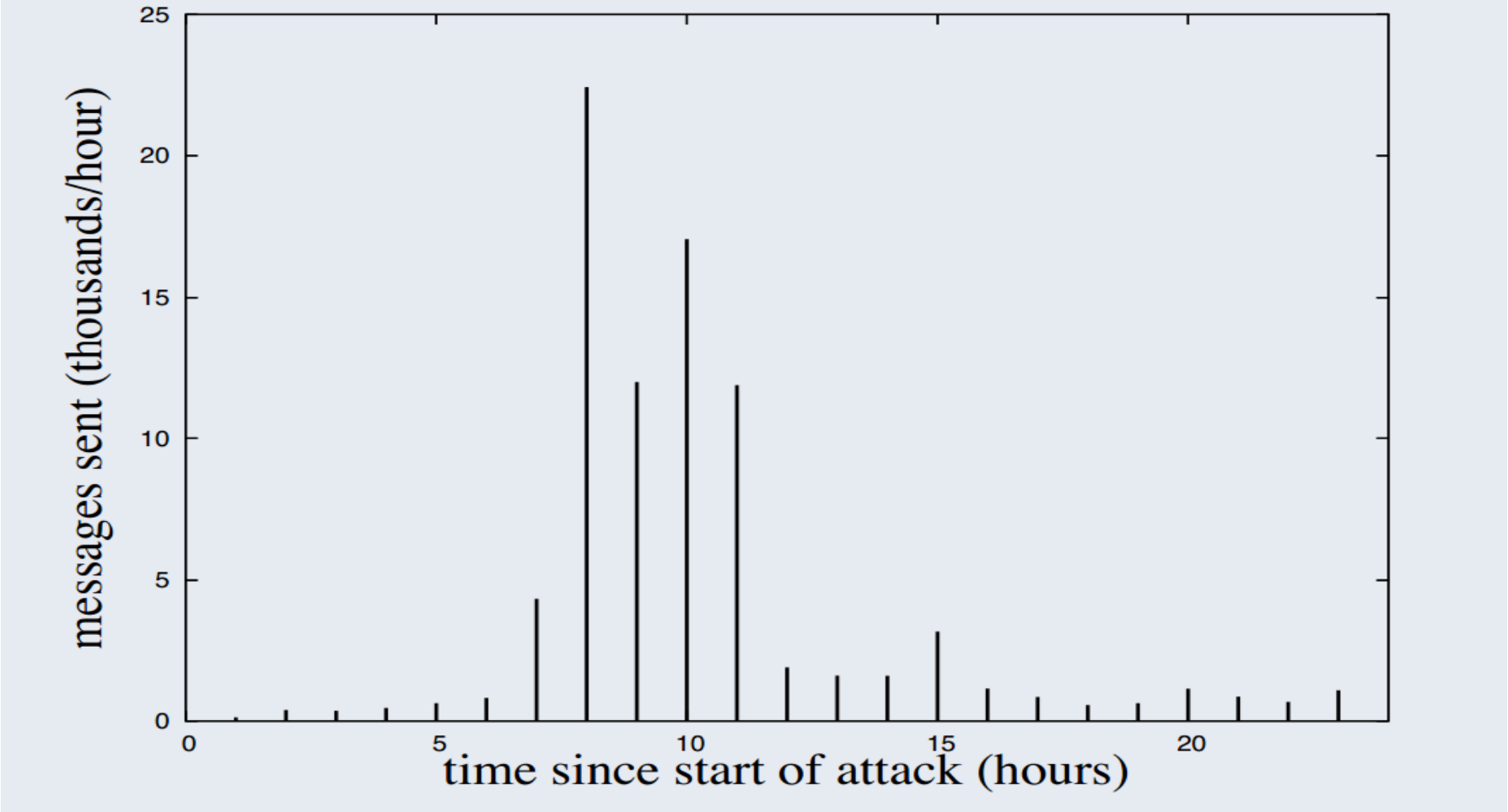
- The defender seeks to shorten Attack and Detect while lengthening Defense and Mutate



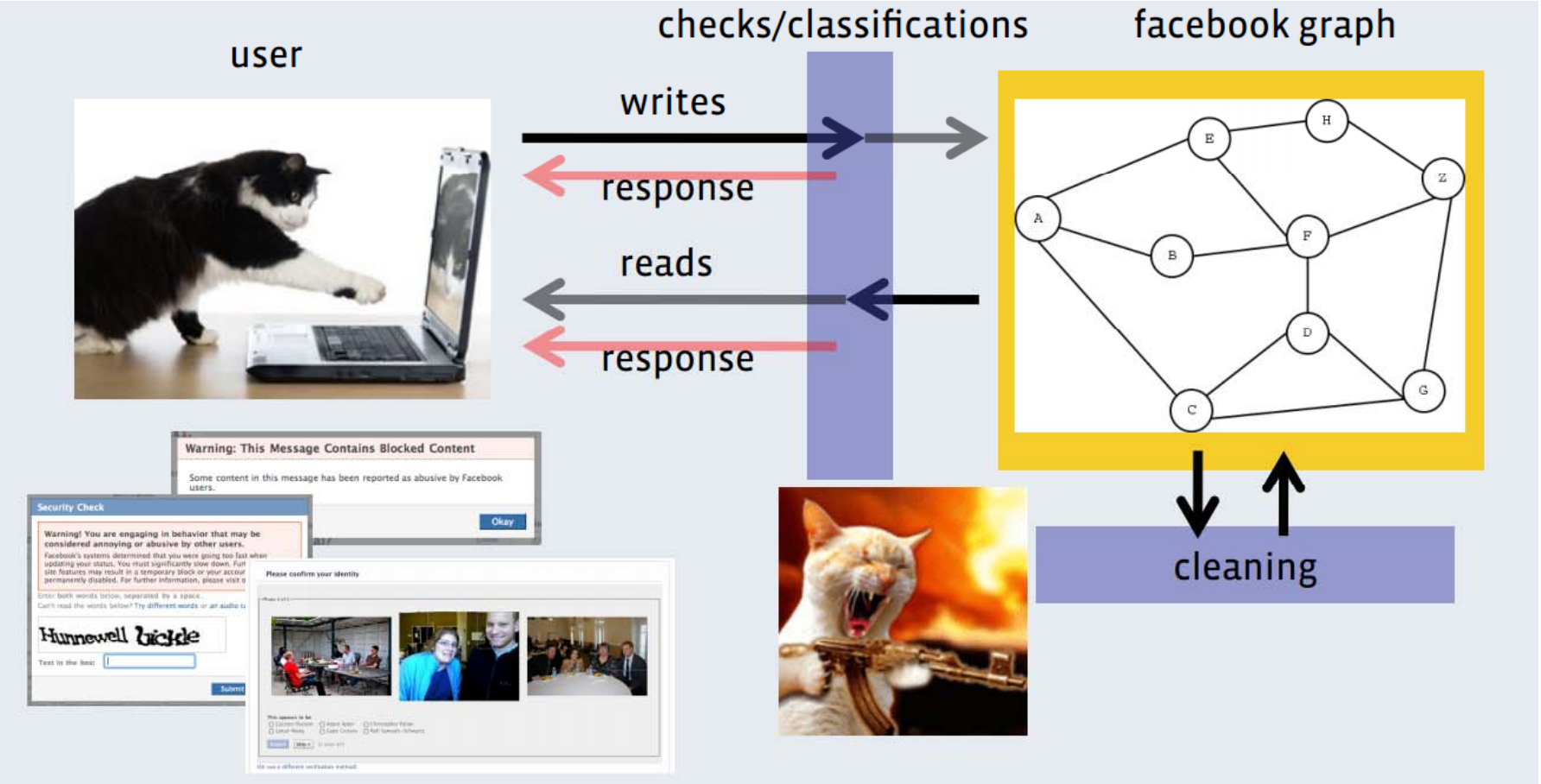
Method

- The Attack phase
  - better user feedback
  - more effective unsupervised learning and anomaly detection.
- The Detect phase
  - quickly building and deploying new features and models.
- The defense and mutate phases
  - making it harder for the attacker to detect and adapt
  - obscuring responses and subverting attack canaries*

Phishing Case



Graph and User Protection



Main components

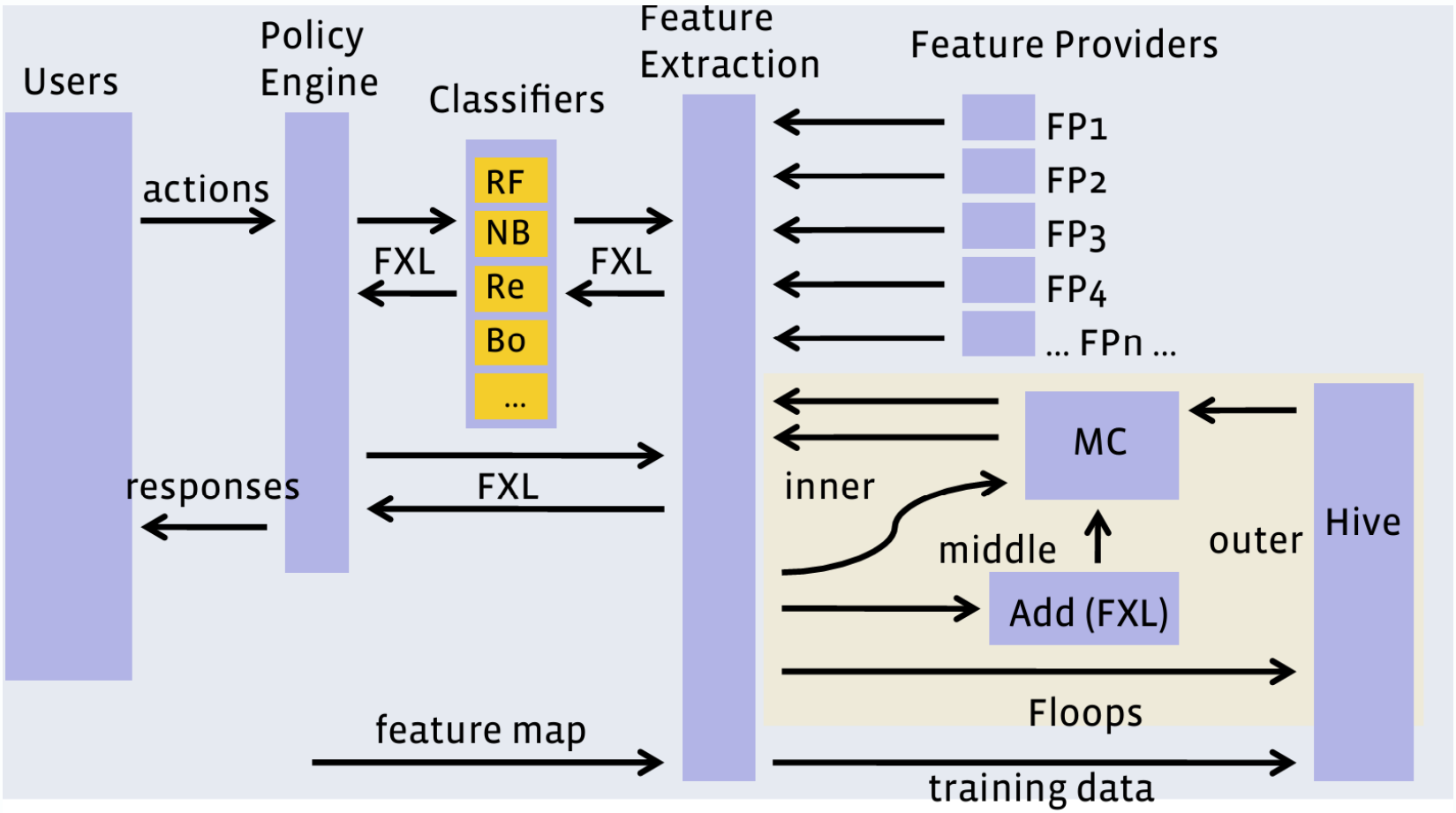
- Classifier services
  - SVM, Random Forest, Logistic Regression, Boosting
- Feature Extraction Language (FXL)
  - Dynamically executed language for expressing features and rules
- Dynamic model loading
- Feature Loops (Hooks)

```
And(IsChannel("messages"),
 And(GreaterThan(Count(ExtractURLs(Text)), 0),
  And(
   GreaterThan(
    ClassifyScore("fakers", "2011-03-15"), 0.41),
   GreaterThan(
    ClassifyScore("bad_urls", "2011-03-14"), 0.74)
 )))
=> SpamFolder
```

start

bling an

Sigma: High-level design



Feature Loops (an FXL feature provider)

- Inner Floop (counters), 10ms latency, Memcache
  - The number of login failures per IP
  - Number of users that cleared User Experience per IP
  - Rate that app has published to Feed
- Middle Floop (tailers), 10s latency, Scribe HDFS
  - The number of times a domain has been sent in a message
  - Trigger recomputation of friend network coherence
- Outer Floop (Hive), 1d latency
  - Unique users disabled from an IP over past 15d
  - Number of users with given name per country and gender

Challenges

- Lack of a clean data layer. Opaque data definition
- Feature reliability
- Many channels, and they evolve
- Actionable detection
- Scaling to deeper classification at display-time

Summary

-
-
-

情 系 人 人

谢 谢