
facebook

facebook

Protecting the Graph

Tao Stein, Facebook
W2SP, May 26, 2011

What does Web 2.0 Security mean?

- What was Web 1.0?
  - A document graph
  - Primarily a library with a document index
- What is special about Web 2.0?
  - A people graph
  - A new way for humans to organize
- What does this mean for Security?
  - Need to think about people and their psychology more

Agenda

1. Threats
2. Adversarial Learning
3. Countermeasures and Systems
4. Challenges
5. Final Words

Threats to users and the graph

Protect the Graph

From what?

- Fake accounts/objects
- Compromise
  - Phishing
  - Malware
- Creepers and Spammers

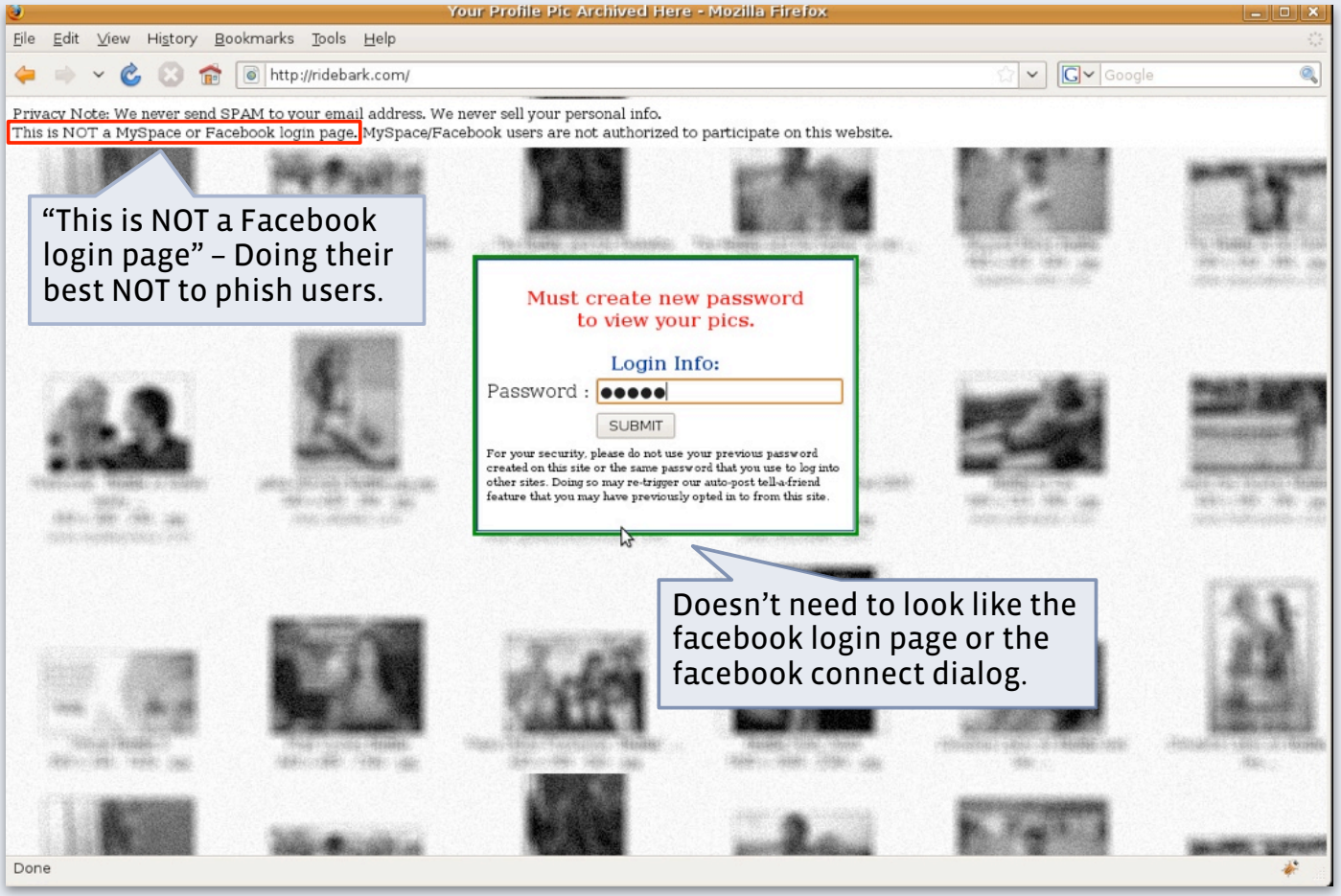
Fake accounts

Image	Id	Name	Email
	**100001705257960** Log in as Go to cs/info.php Go to cs/fit.php Go to si/activity.php	Raegan Mcfadden	akncestorannie@yahoo.com
	**100001736819193** Log in as Go to cs/info.php Go to cs/fit.php Go to si/activity.php	Fiona Allen	sexwy56@yahoo.com
	**100000135058489** Log in as Go to cs/info.php Go to cs/fit.php Go to si/activity.php	Felicity Fuller	guiltyboo5398@yahoo.com

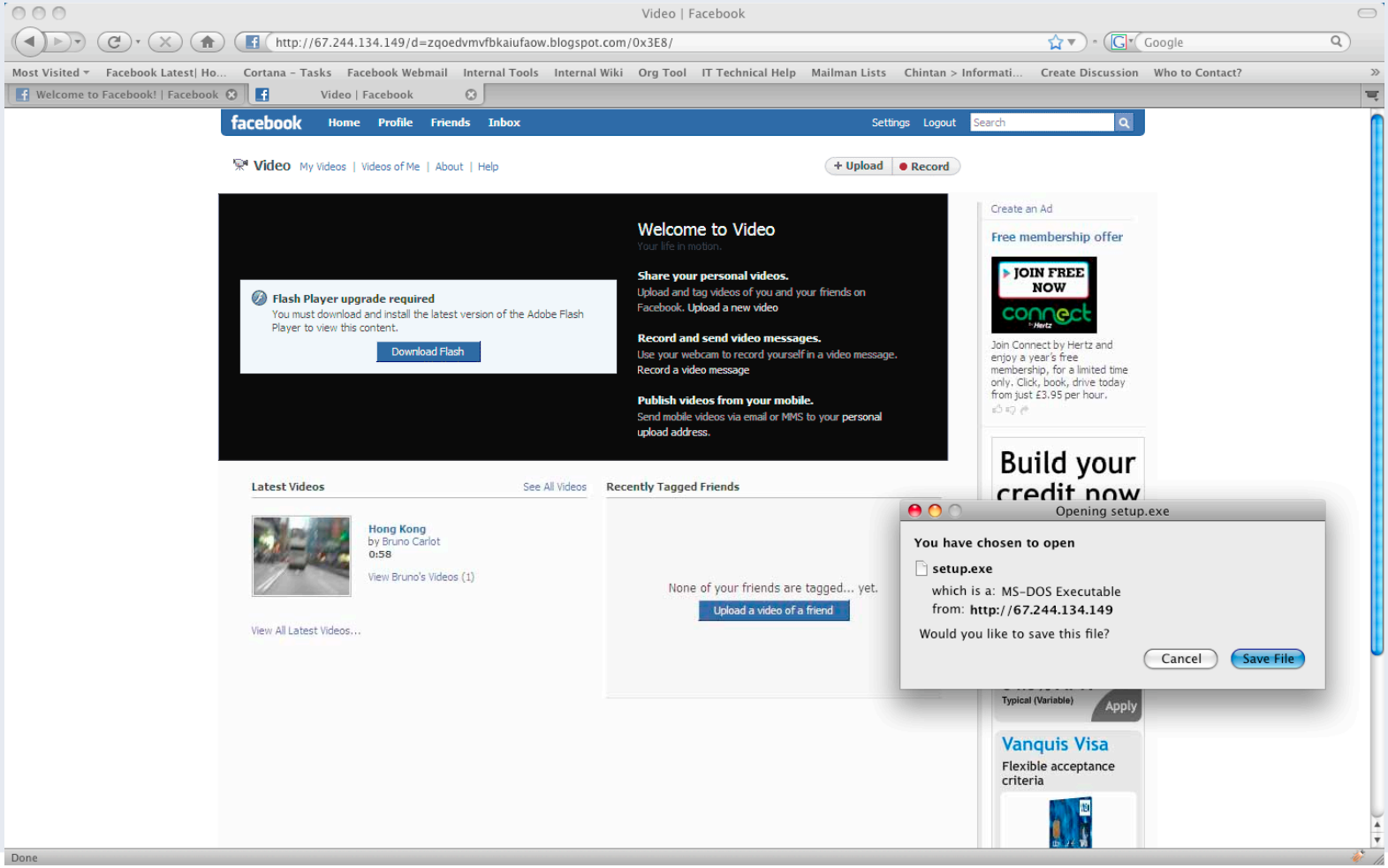
Compromised nodes - Phishing

Domains & URLs are cheap. Text is cheaper!

Compromised nodes - Phishing

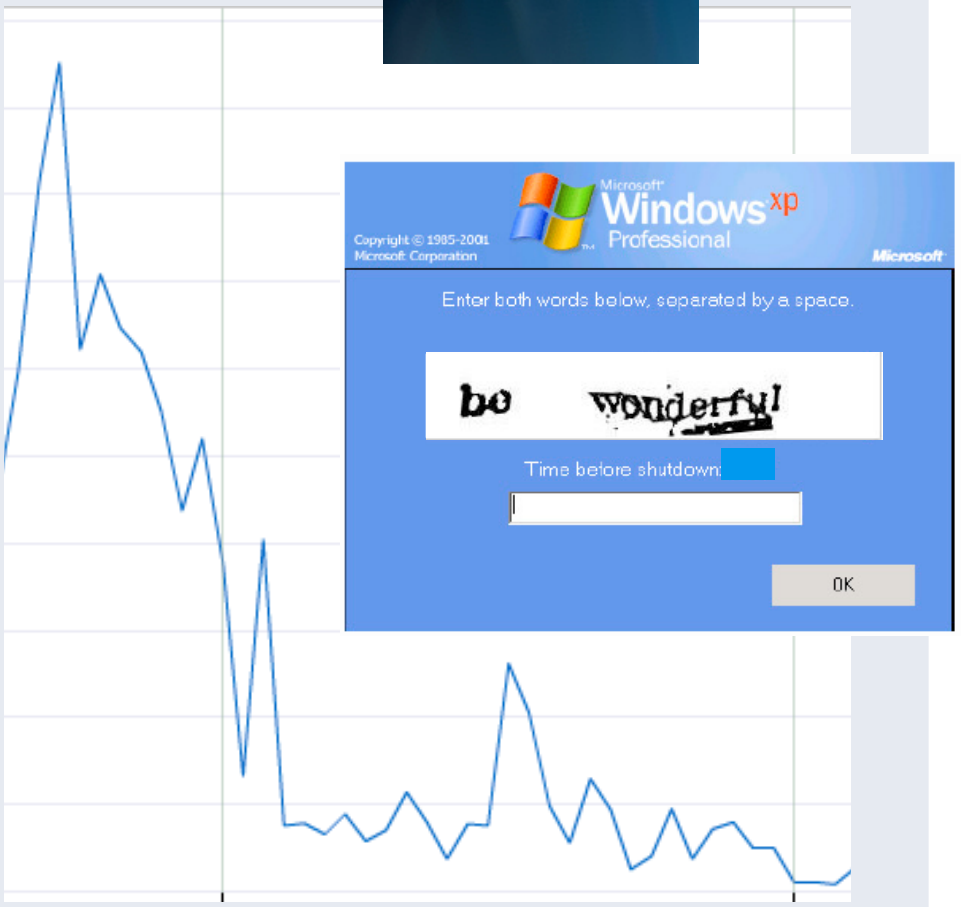


Compromised nodes - Malware



Compromised nodes – Malware

- Active September 2008 – March 2011
- Victims solving CAPTCHAs
- Extremely adaptive
- Monetized via spam, harvesting phone numbers and credit cards
- Not isolated to Facebook, also active on other social networks



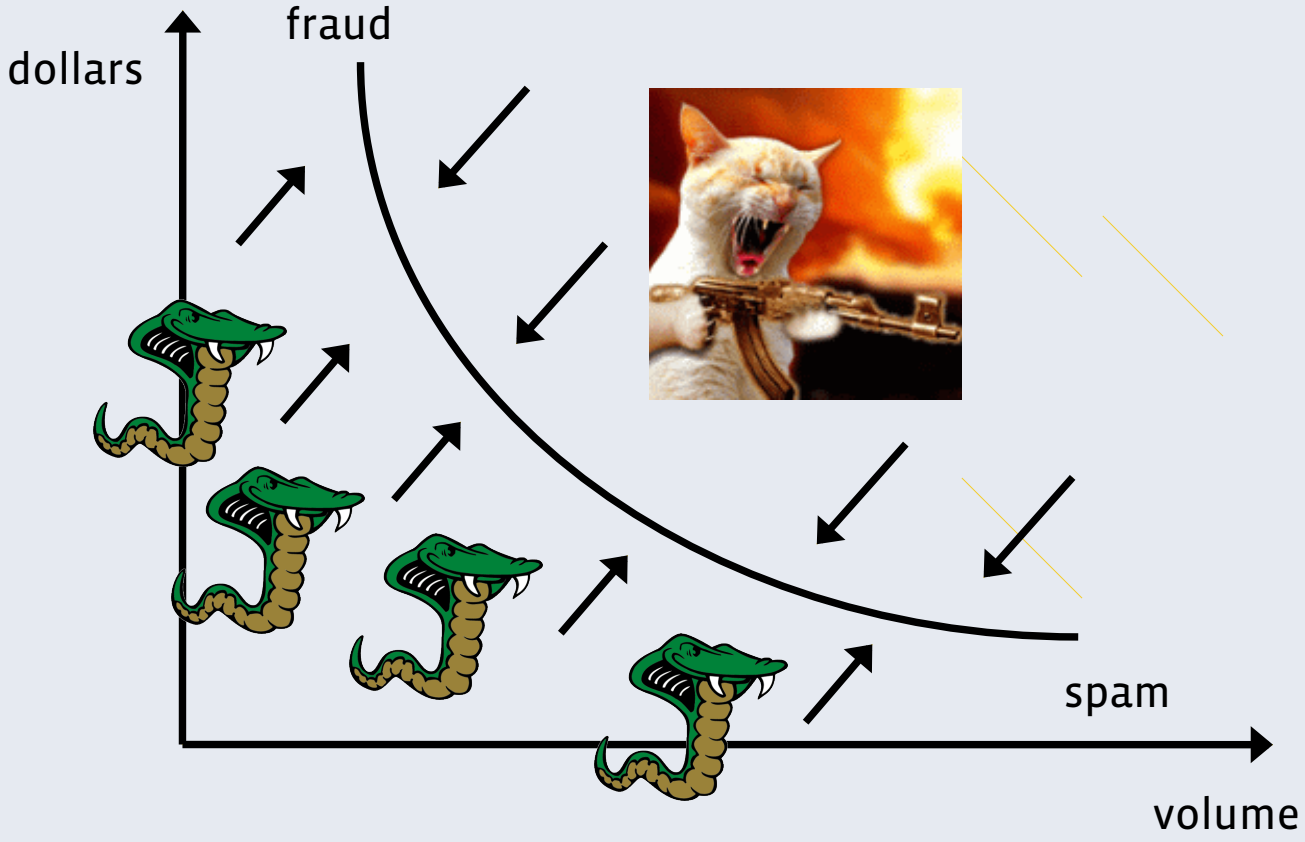
Creepers and Trolls

- Unwanted friend requests
- Chain letters
- Comments and wall posts on pages and social plugin

Agenda

1. Threats
2. Protecting the Graph
3. Countermeasures and Systems
4. Challenges
5. Final Words

The Efficient Abuse Frontier (the Front)



Balance of Power

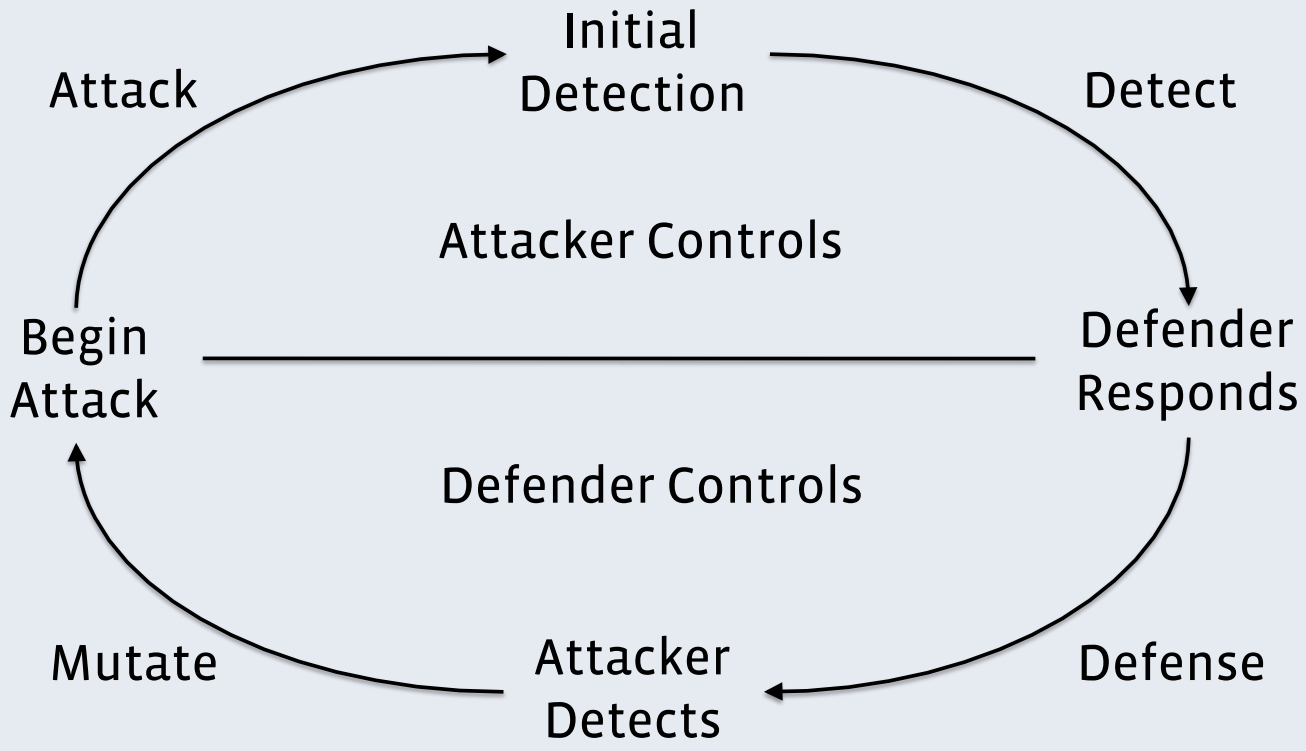
- **What the attackers have:**

- Labor. They have much more
- Distributed botnets, compromised webhosts, infected zombies
- Fake and compromised objects (events, apps, pages, groups, users, ...)

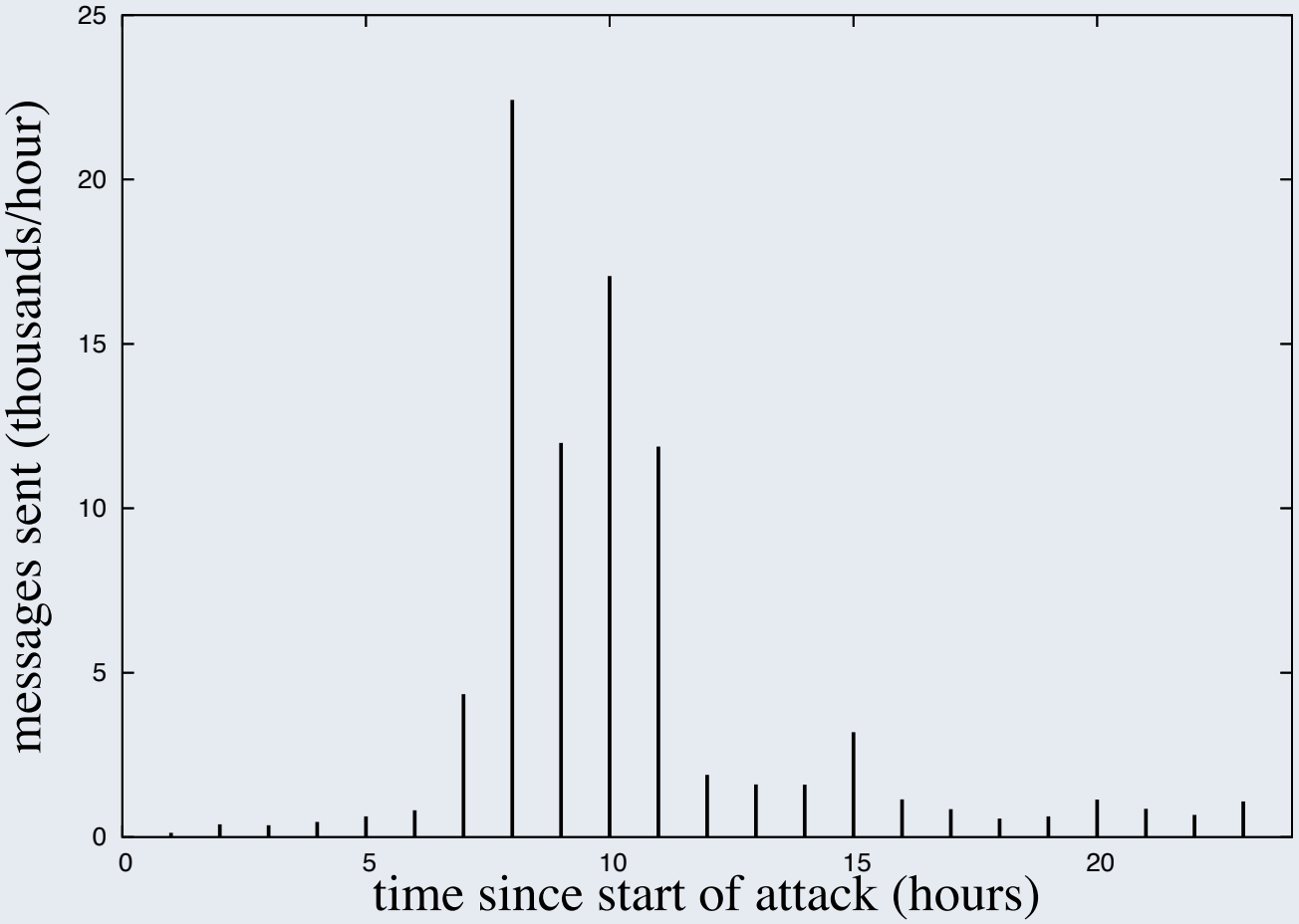
- **What we have:**

- Data centers, content distribution networks, client-side javascript
- User feedback -- spam reports, feed hides, friend rejects
- Knowledge of patterns, anomalies, and global graph structure
- Shared secrets with users

Adversarial Cycle (how the Front shifts)



Phishing Example



User Feedback

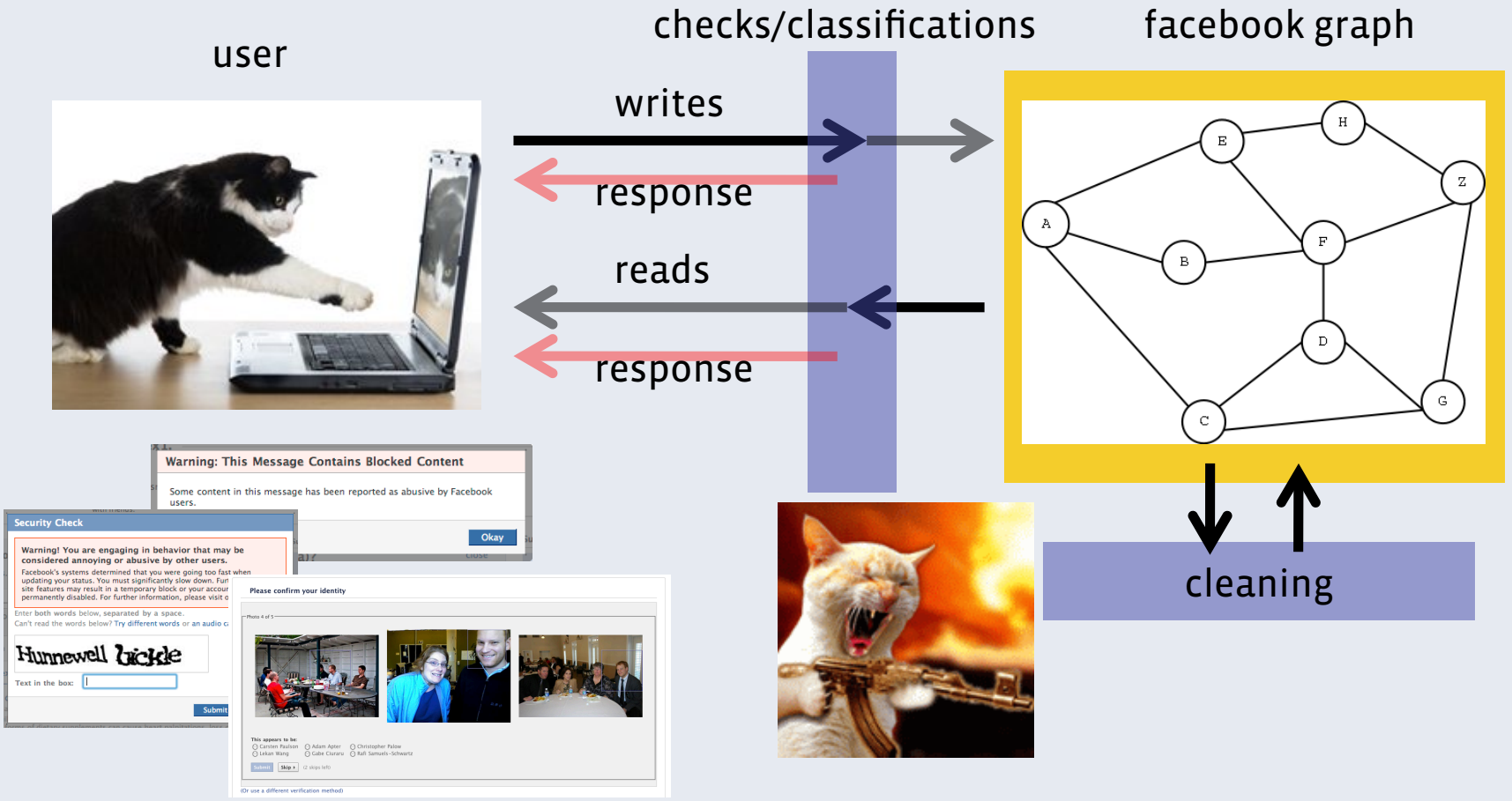
How does this differ from traditional learning?

- Attackers mutate their own patterns
- Latency is really important
- Feature selection depends on economic and system externalities
- Many simultaneous and evolving channels
- Real-time response

Agenda

1. Threats
2. Protecting
3. Countermeasures and Systems
4. Challenges
5. Final Words

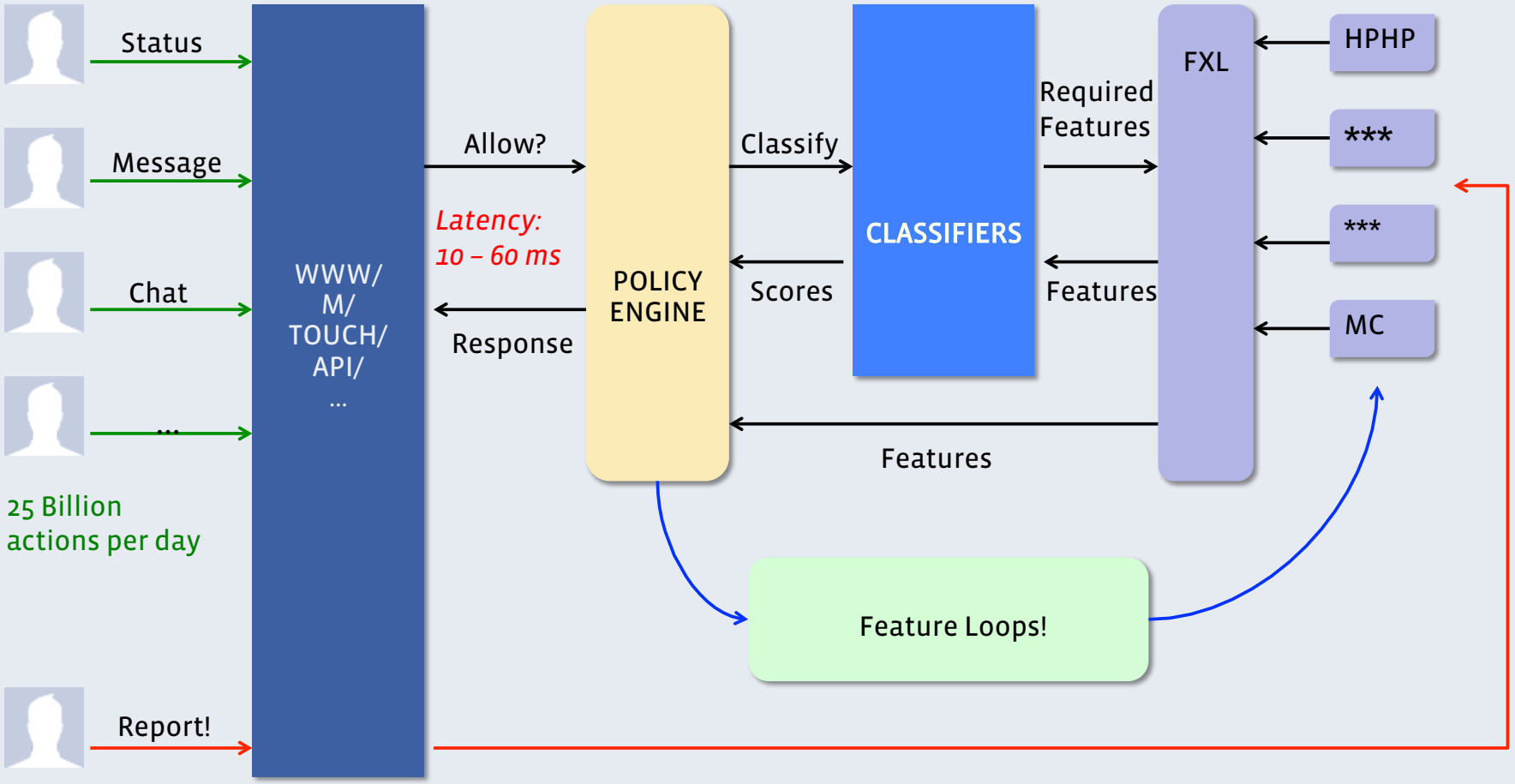
Graph and User Protection



Design Principles

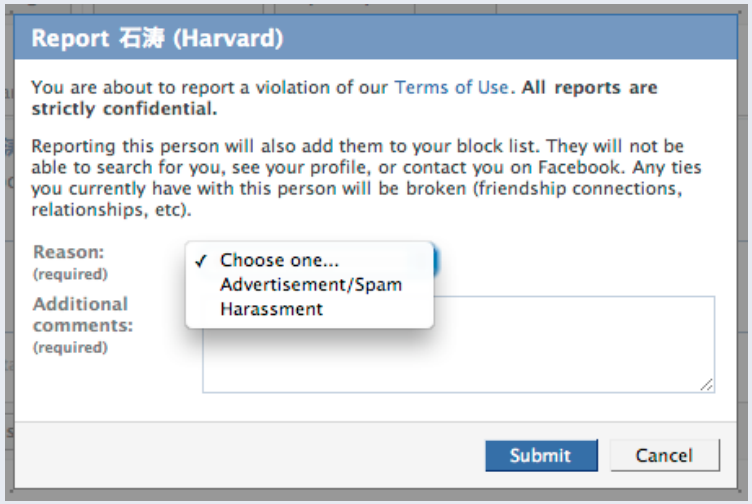
- Many classification/ranking algorithms
- Features are important
  - Make creation easy
  - Many sources
  - Share across channels
  - Feedback
- Policy rules above machine learning
  - Holdouts
  - Business logic
  - Response flexibility (dynamic graphs)

Real time classification architecture



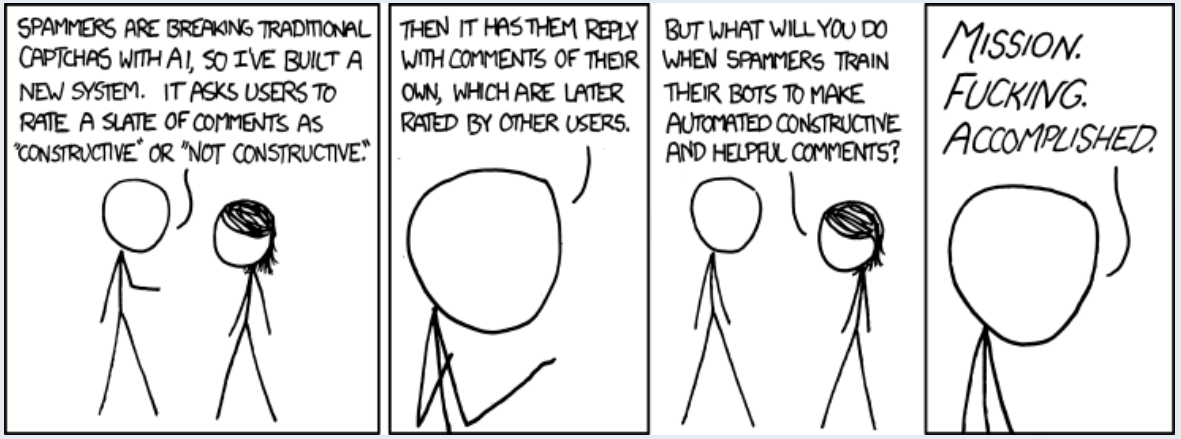
Features

- Leverage the graph to protect the graph
- Anomalous behavior (IP, User, Geo, etc.)
- Entity reputation (IP, User, Cookie, URL, etc.)
- Entity state (# friends, # of likes, etc.)
- Blacklists
- Classifiers
- User feedback
  - Explicit reports
  - Actions



Responses

- CAPTCHAs
- Blocks



- Put through educational flow (feature blocks)
- Asynchronous actions
- Authentication:
  - Social challenge
  - Password reset
  - SMS verification

Policies

- Are Graphs
- Combine the pieces:
  - User Feedback (Seeds)
  - Classification (Deciders)
  - Associations (Expanders)
  - Responses

Expressing Features and Policies

- Two different example features:

```
Max(Map(DomainSpamScore, ExtractDomains(Text)))
```

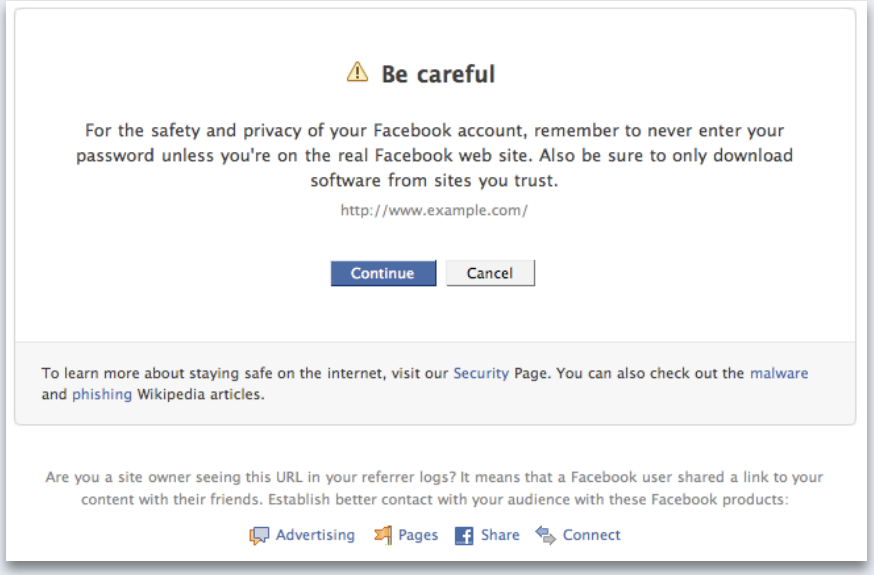
```
Count(Intersect(LikedPages(Sender),
                LikedPages(Receiver)))
```

- An example Policy:

```
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
```

Bad URLs got through! Now what?

- Delay in classifying a URL as bad
- Linkshim
  - Layer of indirection
  - Real-time click stream information
  - Can control access to known malicious sites and warn users, slowing distribution of the attack
- Display time checks



Agenda

1. Threats
2. Protecting
3. Systems
4. Challenges
5. Final Words

UEX challenges – Friction is bad

- Actionable user feedback & user education
  - Self and social remediation

- Best fit response
  - Couple the response form to the attack
  - Contextual to aid education and understanding
  - Make hard for attackers to spoof

Systems challenges

- Classification
  - Pipelines for training, validating & deploying need to be fast
  - Modular design allowing for isolation of classifiers
  - High throughput & uptime

- Feature extraction (feature data layer)
  - Rich features – Offline mining
  - Integration – Multiple data sources
  - Performance – Function of data sources and complexity
  - Integrity and Availability – Failures, defaults.

Systems challenges

- Detection
  - Detect breaches in the ‘front’ & respond
  - Dynamically adapt response graphs on quality feedback

- Code quality
  - Find potential security flaws early
  - Make it impossible for certain security flaws to exist – abstractions
  - Static & Dynamic analysis

Agenda

1. Threats
2. Protecting
3. Protecting the Graph
4. Challenges
5. Final Words

Summary

- Need to understand and think about people:
  - fear and greed attacks
- Adversarial learning problem:
  - Response. Response form and latency are really important
  - Features. Make it easy to try out new features and models
  - Graph-structured responses
- Rich and wide product:
  - Share signals across channels
  - One set of hooks for complete coverage

Questions?

stein@fb.com