FaceTrust: Collaborative Unwanted Traffic Mitigation Using Social Networks

Michael Sirivianos, Xiaowei Yang, Kyungbaek Kim

Motivation

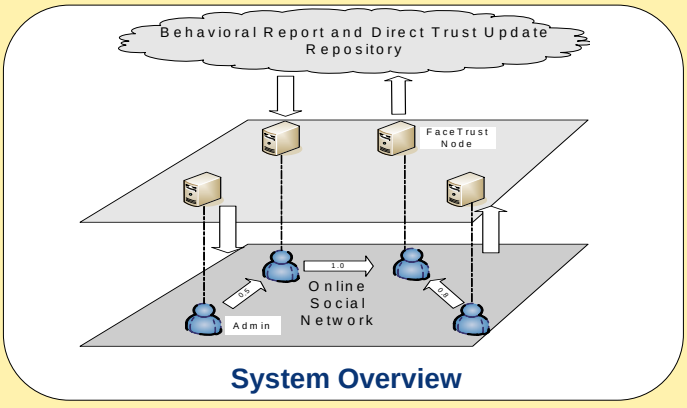
Rapid and Reliable Suppression of Unwanted Traffic

Current unwanted traffic mitigation techniques rely heavily on centralized, non-collaborative infrastructures with limited coverage and slow response times e.g.:

- **SpamHaus, Dshield, TrustedSource, SiteAdvisor:** IP or site reputations for spam and malicious code
- **CounterMalice, EarlyBird**: network-layer worm detection/ containment
- **Software Vendors:** Security patches for vulnerable systems

However, threats spread too fast and are hard to identify!

- A TCP flash worm could infect 1 million nodes under 4 secs [Staniford WORM 04]
- Many spam bots appear low volume if observed at any single domain [Ramachandran SIGCOMM 06]



System Overview

Prior Approaches and Our Goal

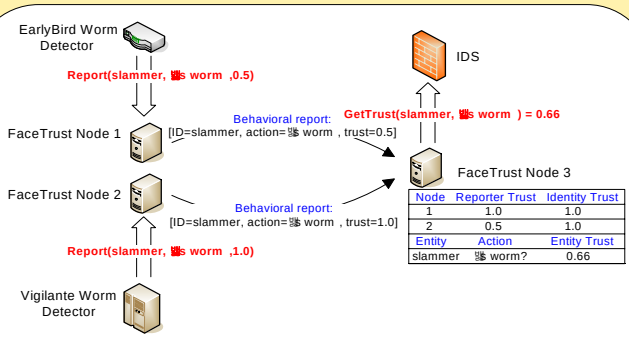
- **ALPACAS:** Collaborative spam filtering. Assumes correctness of spam reports
- **Vigilante, Sweeper, NetShield, [Weaver Security 04]**: Collaborative worm detection, early warning and containment. Assume all nodes can verify reports, or that all nodes are trustworthy

Our goal :

- A *Collaborative Trust Management System for Internet Entities*, e.g. IP address, packet signature, email signature etc
- Applications determine with a quantifiable certainty whether an entity is performing a specific malicious *action*, e.g. ``is a spam bot''

Challenges:

- Fake *reports* regarding the *behavior* of entities
- Fake *updates* regarding the *trustworthiness* of *reporters*
- Sybil attack



FaceTrust Operation Example

Trustworthy Behavioral Reports

Reporter Trust:

- Any two socially acquainted FaceTrust nodes i,j initialize the *direct trust* d_{ij} between their devices to a social trust $2\ [0.0,1.0]$.
- Any two nodes i,j able to verify each other's *behavioral reports*, update d_{ij} based on similarity $s_{ij}\ 2\ [0.0,1.0]$ between their reports

$$d^{k+1}_{ij} = \alpha * d^{k}_{ij} + (1-\alpha) * s_{ij}$$

- Node k retrieves d_{ij} to build the *reporter trust graph* $T(V_k,E_k)$.
- For each node $j\ 2\ V_k$ of which node k considers the behavioral reports, k analyzes T to compute the transitive *reporter trust* t_{kj}.

For every path p from k to j: $t_{kj} = max_p\ (\Pi_{u\ !\ w\ 2\ p}\ d_{uw})$

Identity Trust. OSN Providers as Certification Authorities:
OSN providers analyze the social graph using a SybilLimit-like algorithm to derive the probability I_j that a node j is a Sybil

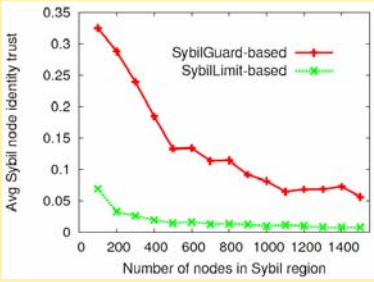
$$t_{kj}\ \tilde{A}\ t_{kj}\ £\ I_j$$

Entity Trust:

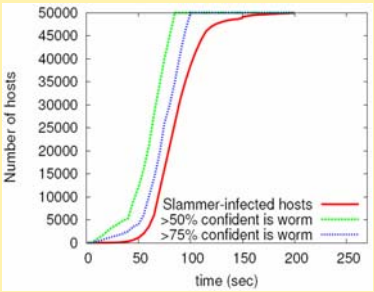
- For the same entity e and the same action a, node k may receive multiple behavioral reports from nodes $j\ 2\ V_k$, with trust c_j.

$$GetTrust_k(e,a) = \frac{\Sigma_{j\in V_k}\ t_{kj}\ c_j(e,a)}{\Sigma_{j\in V_k}\ t_{kj}}$$

Evaluation



- Facebook 50K node sample.
- Sybil region forms a random graph with avg degree 14. A single attack edge to the honest region.
- Average **honest** node identity trust is **~0.9**
- **Identity trust of Sybil nodes decreases substantially with their number**



- 2000 honest worm reporters
- 500 dishonest worm reporters
- $|V| = 1000$
- Social trust random in [0.0, 1.0].
- SQL Slammer Worm dynamics from [Moore S&P 03].
- SimPy-based discrete-event simulator.
- **Nodes conclude that slammer is worm faster than worm spreads**

University of California Irvine / Duke University

Network and Distributed Systems Group