FaceTrust: Collaborative Unwanted Traffic Mitigation Using Social Networks

Michael Sirivianos Xiaowei Yang
Department of Computer Science
University of California, Irvine
{msirivia,xwy}@ics.uci.edu

ABSTRACT

Current unwanted traffic mitigation techniques are heavily reliant on centralized infrastructures and place trust on a small number of security authorities. As a result, they offer limited threat coverage and slow response times. To address this problem, we propose FaceTrust: a large scale collaborative system for the rapid propagation of reports concerning the behavior of Internet entities (hosts, email signatures etc). FaceTrust uses the reports and the social network of it's nodes' administrators to enable an application to obtain a quantitative measure of an entity's trustworthiness: the likelihood that the entity is associated with a specified malicious action (e.g. spam). Applications can in turn use this measure to make informed decisions on how to handle traffic associated with the entity in question.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Security, Management

Keywords

Peer-to-peer, reputation management, sybilproof, worm containment, spam filtering

1. INTRODUCTION

The vast majority of existing unwanted traffic mitigation techniques rely on centralized infrastructures and place trust on a small number of security authorities. Email systems depend heavily on a few centralized IP blacklisting services [1, 2]. Yet, the amount of spam is a testament to the inadequacy of centralized blacklisting. End-users rely on software vendors to update their anti-virus/malware tools, or to release updates that patch their systems' vulnerabilities. However, practice has shown that software vendors do not respond to these vulnerabilities in a timely manner [8].

Driven by the inadequacy of existing solutions, we embrace a collaborative approach. End-users participate in a distributed early warning system for unwanted Internet Traffic. We propose FaceTrust: a large scale peer-to-peer system for the rapid propagation of reports concerning the behavior of Internet entities (hosts, email signatures etc). Our system aims at ensuring that the behavioral reports reach the FaceTrust nodes faster than the threat itself, and that these reports are sufficiently trustworthy to warrant action by their receivers. FaceTrust enables an application to obtain a quantitative measure of an entity's trustworthiness using the API `GetTrust(entityID, action)`. The obtained trust metric corresponds to the likelihood that the entity will perform a specified malicious action (e.g. serve malware).

SIGCOMM'08, August 17–22, 2008, Seattle, Washington, USA.
ACM 978-1-60558-175-0/08/08.

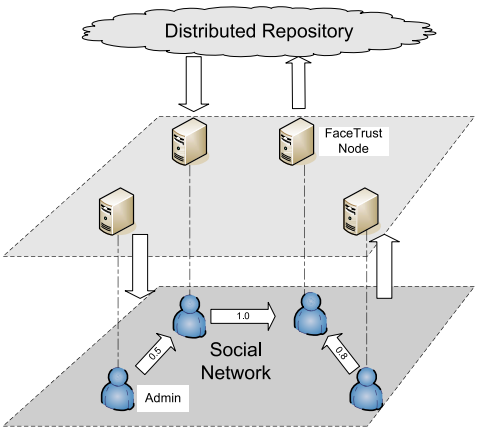

Figure 1: FaceTrust architecture.

A metric for the trustworthiness of entities is most useful during first encounters. For instance, it is very important for an end-system to detect whether a packet is carrying malicious code, even if it has never encountered its packet signature before. Unfortunately, without prior encounters, a FaceTrust node can not speculate a trust metric solely from its own observations. This problem is commonly addressed by using a network-wide reputation system to obtain an assesment of trust [4, 6]. FaceTrust employs a reputation management system that uses the social network of its nodes' administrators and is sybil-resistant [3] Each FaceTrust node reports to the system its own direct experiences with Internet entities and other nodes. These aggregated experiences enable a node to compute a trust metric of an entity on a first encounter. Applications can in turn query the node and obtain this metric and make informed decisions on how to handle traffic associated with the entity in question.

2. DESIGN

Figure 1 depicts the design of FaceTrust, which aggregates the experiences of multiple nodes to compute trust metrics. At a high-level, the FaceTrust system comprises the following components: 1) human users that administer networked devices/networks (*FaceTrust admins*) and join a social network; 2) end systems that are administered by specific admins and participate in monitoring and reporting other entities' behaviors, (*FaceTrust nodes*); 3) behavioral reports submitted by FaceTrust nodes; and 4) a report repository that receives and stores FaceTrust nodes' reports, and aggregates their reports to compute trust metrics. The report repository is an abstract component and we later describe our initial approach to its implementation

A FaceTrust node uses the `Report(entityID, action, confidence)` interface to feedback its observed behavior of an entity to the repository. Each report includes a timestamp and is signed by the reporting FaceTrust node's public key for authentication and integrity. FaceTrust nodes compute the trust metric `GetTrust(entityID, action)` by aggregating relevant reports in the repository. An end system that does not participate in FaceTrust may be configured to send `GetTrust` calls to FaceTrust nodes it trusts, or that reside in their local administrative domains, similar to the DNS system.

2.1 Using Social Networks to Build Confidence in Reports

Malicious nodes may issue *false reports* to manipulate trust metrics. FaceTrust can mitigate these attacks by giving higher weights to reports obtained from more trustworthy FaceTrust nodes when reports are aggregated. Conceptually, each FaceTrust node i maintains an overall trust score t_{ij} to every other FaceTrust node j. This trust score corresponds to node's i estimation of the probability that node's j reports are accurate. It is obtained from three sources: trust attainable from online social networks, direct report verification, and transitive trust.

First, FaceTrust relies on the fact that FaceTrust nodes are administered by human users. Competent and benign human users are likely to maintain their nodes well, and provide honest and truthful reports. The trust on the competency and honesty of human users could be obtained via social networks. In the FaceTrust design, admins maintain accounts in online social networks. An admin i tags her acquaintance admin j with a trust score s_{ij} in $[0,\ 1.0]$ based on her belief on j's ability to manage her FaceTrust node(s). This value is used to initialize a direct trust score between two FaceTrust nodes administered by i and j: $d_{ij} = s_{ij}$. If two nodes do not have a tagged social trust s_{ij}, d_{ij} is initialized to zero.

Second, a FaceTrust node i dynamically updates the direct trust d_{ij} by comparing entity behavioral reports submitted by the node j with its own reports. A node i may verify a report from a node j for an entity e, if i has also generated a recent report with respect to the same threat. It may also probabilistically choose to observe e solely for verification purposes. Intuitively, if i and j share similar opinions on e, i should have a high trust in j's reports. Let v_{ij} be a measure of similarity in $[0,\ 1.0]$ between i and j's reports. A node i may updates its direct trust to j using an exponential moving average: $d_{ij}^{k+1} = \alpha * d_{ij}^{k} + (1-\alpha) * v_{ij}$. As i verifies a large number of reports from j, the direct trust metric d_{ij}^{k} gradually converges to the similarity of reports from i and j.

Third, a FaceTrust node i incorporates direct trust and transitive trust to obtain i's overall trust to j: t_{ij}. Due to the large number of FaceTrust nodes, the FaceTrust admin of a node i may not tag a social trust s_{ij} to the admin of a node j. Moreover, due to the variety and large number of observed entities, nodes i and j may not have encountered the same entities and are therefore unable to directly verify each other's reports. Furthermore, i can further improve the accuracy of its trust metric for j by learning the opinions of other FaceTrust nodes about j. The overall trust t_{ij} can be obtained as the maximum trust path in the FaceTrust's trust graph, in which each edge $u \rightarrow v$ is annotated by the direct trust d_{uv}. That is, for any path p from i to j:

$$t_{ij} = max_p(\Pi_{u \rightarrow v \in p} d_{uv})$$

2.2 Mitigating Sybil Attacks

An adversary may attempt to deploy multiple FaceTrust nodes and create multiple social network admin identities to increase its ability to subvert the system using false reports (a *Sybil attack*). Similar to SybilLimit [9], FaceTrust leverages social networks to mitigate Sybil attacks. Since there is typically a strong tie between a user's social network identity and her real-life identity, a social network provider may use a user's connectivity in the network to determine the authenticity of a user's identity. A FaceTrust node i may obtain an identity trust score I_j from social network providers. A node i updates its trust to j by incorporating the identity trust I_j: $t_{ij} \leftarrow t_{ij} * I_j$.

Social network trust rests on the assumption that a user's social network identity is authentic. Unfortunately, a malicious user may create multiple fake identities (a *Sybil attack*) or create a social network identity to impersonate someone else. We propose to develop mechanisms to defend against these attacks on social network identities. A social network provider may use those mechanisms to provide an identity trust metric $I_j \in [0, 1.0]$ of a social network identity j to applications or users. I_j specifies the likelihood that j is an authentic identity.

Intuitively, malicious users can establish only a limited number of trust relationships with real humans. Thus, groups of Sybil attackers are likely connected to the rest of the social graph with a disproportionally small number of social edges. A social network provider may detect Sybil attackers using a user's connectivity to the densely connected core of high degree nodes in the social network. The existence of this large strongly connected core (or giant) component has been experimentally observed in [7]. Connectivity is measured in terms of the number of disjoint paths in the social graph between a user and the boundaries of the social network's core.

A social network provider may set an upper threshold in the number of disjoint paths. If the threshold is exceeded, a user j has an identity trust $I_j = 1.0$. Otherwise, she has a smaller identity trust, because a user that has few disjoint paths to the strongly connected core component is likely to be part of a Sybil group.

2.3 Disseminating Trust Information

A FaceTrust node i needs to obtain reports submitted by other nodes, and the updated direct trust scores d_{uv} to compute a trust metric `GetTrust(Entity, Action)`. Recent advances in Distributed Hash Tables and gossip protocols [5] suggest that it is possible to implement a scalable distributed FaceTrust repository that disseminates and stores reports and the direct trust updates.

For reasons of scalability and efficiency, nodes receive behavioral reports only from a (possibly random) subset V of all the nodes in the FaceTrust overlay. Similarly, nodes receive direct trust updates only from and regarding nodes in V. Each node maintains a table of all nodes in its view V, and associates each node with its user's trust values and its user's public key. The nodes in V can be obtained either by gossiping between nodes, from a DHT or from the social network provider.

3. REFERENCES

[1] Secure Computing, TrustedSource. `http://www.securecomputing.com/index.cfm?skey=1671`.

[2] Spamhaus. `http://www.spamhaus.org/`.

[3] J. R. Douceur. The Sybil Attack. In *IPTPS*, 2002.

[4] K. Hoffman, D. Zage, and C. Nita-Rotaru. A Survey of Attack and Defense Techniques for Reputation Systems. In *ACM Computing Surveys*, 2008.

[5] A.-M. Kermarrec and M. van Steen. Gossiping in Distributed Systems. *SIGOPS Oper. Syst. Rev.*, 2007.

[6] S. Marti and H. Garcia-Molina. Taxonomy of Trust: Categorizing P2P reputation systems. 2006.

[7] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and S. Bhattacharjee. Measurement and Analysis of Online Social Networks. In *IMC*, 2007.

[8] D. Moore, C. Shannon, G. Voelker, and S. Savage. Internet Quarantine: Requirements for Containing Self-propagating Code. 2003.

[9] H. Yu, P. Gibbons, M. Kaminsky, and F. Xiao. A Near-optimal Social Network Defense Against Sybil Attacks. In *IEEE S&P*, 2008.