Taxonomy of Email Reputation Systems

(Invited Paper)

Dmitri Alperovitch, Paul Judge, and Sven Krasser
Secure Computing Corporation
4800 North Point Pkwy Suite 400
Alpharetta, GA 30022
678-969-9399
{dalperovitch, pjudge, skrasser}@securecomputing.com

Abstract

Today a common goal in the area of email security is to provide protection from a wide variety of threats by being more predictive instead of reactive and to identify legitimate messages in addition to illegitimate messages. There has been previous work in the area of email reputation systems that can accomplish these broader goals by collecting, analyzing, and distributing email entities' past behavior characteristics. In this paper, we provide taxonomy that examines the required properties of email reputation systems, identifies the range of approaches, and surveys previous work.

1. Introduction

As spam volumes have continued to increase with high rates, comprising 90% of all email by the end of 2006 as determined by Secure Computing Research**,** the need for fast and accurate systems to filter the malicious email traffic and allow the good mail to pass through has provided greater motivation for development of email reputation systems. Traditional content filtering anti-spam systems can provide highly accurate detection rates but are usually prohibitively slow and poorly scalable to deploy in high-throughput enterprise and ISP environments. Reputation systems can provide more dynamic and predictive approaches to not only filter out the unwanted mail but also identify the good messages, thus reducing the overall false positive rate of the system. In addition, reputation systems allow for real-time collaborative sharing of global intelligence about the latest email threats, providing instant protection benefits to the local analysis that can be performed by a filtering system. Finally, the ease of creation and spoofing of identifiers, such as e-mail addresses and domains, creates a very strong incentive for people to act maliciously without paying reputational consequences [1]. While this problem can be solved by disallowing anonymity on the Internet, email reputation systems are able to address this problem in a much more practical fashion. By assigning a reputation to every email entity, reputation systems can influence agents to operate responsibly for fear of getting a bad reputation and being unable to correspond with others [2].

The goal of an email reputation system is to monitor activity and assign a reputation to an entity based on its past behavior. The reputation value should be able to denote different levels of trustworthiness on the spectrum from good to bad. In 2000, Resnick *et al.* described Internet reputation system as having three required properties [3]:

- Entities are long lived,
- feedback about current interactions is captured and distributed, and
- past feedback guides buyer decisions.

We will examine email reputation systems according to these properties to clarify the goals and view the solution space. With these definitions in place, we can define criteria for systems claiming to be email reputation systems as well as identify the open areas of future work. We have devised this taxonomy while designing the TrustedSource email reputation system [4].

The rest of the paper is organized as follows. Section 2 provides some background regarding reputation-based filtering approaches in the email messaging space. Section 3 defines abstract terms that generalize the notion of reputation. In Section 4, we present and categorize the feedback mechanisms that a reputation system can incorporate as input data. Architectural considerations are discussed in Section 5. Section 6 concludes the paper.

2. Background

With respect to email systems, an early approach for effective filtering has been the use of real-time blacklists (RBLs). Whenever a message is sent to an email server, this receiving email server queries an RBL server to lookup the IP address or domain that is connecting to it to deliver messages. The RBL returns a yes/no result indicating whether the sending IP/domain is a known source of spam or is associated with malicious activity. The receiving email server then typically rejects messages from this sender or takes the RBL result into account while performing local analysis to determine the maliciousness of the message. Typically, these RBL-lookups are conducted over the DNS protocol.

RBLs generally receive information about malicious senders from spam messages hitting spamtrap email addresses, manual listings, or user feedback. We will cover these feedback techniques in more detail in Section IV. One drawback of this approach is the slow reaction time to new sources of malicious activity and the narrow coverage of the sender universe. For example, a particular spam run needs to hit a spam trap address or be reported by a user before an IP or domain would be listed on an RBL, by which time the spam activity from that sender may have already been terminated. Since it is common for zombie sources of spam to have only a few hours of sending activity, the effectiveness of an RBL can be severely degraded if the time for listing a sender exceeds that average.

One approach to counter these shortcomings is to take more feedback into account. Real-time queries sent to RBLs by mail servers seeking answers about senders they see contacting them can yield valuable insights. With each query, the RBL gets an information record comprising the queried IP/domain, the source IP of the query, and a timestamp. Ramachandran *et al.* analyze the source IPs querying an RBL [5]. They investigate a dataset of RBL queries to spot exploited machines in botnets that are sending queries to test whether members of the same botnet have been blacklisted. Therefore, additional information can be obtained with regard to the maliciousness of the source IP. In previous work, we showed that based on query patterns it is possible to detect spam senders [6]. This approach allows gaining additional information on the queried IP.

However, such classifiers cannot always make a definite decision. Introducing a continuous reputation value in contrast to a discrete yes/no decision allows the user of such a system to define its own thresholds. More importantly, this continuous value can be used as a feature in a local classification engine.

3. Definition of entity and identifier

Besides the IP-based approaches discussed so far, there are other means to assign a reputation. To do so, we need to step back and look at where a message originates and how we identify this origin. We define an *entity* as the origin (or sender) of a message. An entity could be a specific machine or a group of users. When an entity connects to the Internet to send messages, it exposes certain features inherent to it. If a feature is *sufficiently unique* and strongly correlates with an entity, we call this feature an *identifier*. Examples of identifiers include IP addresses (as discussed before), verified sender domains or addresses, the message itself, or URLs inside the message. Figure 1 shows an overview how identifiers can be categorized.

An entity can have multiple identifiers at the same time or over time, an example being a mail server with changing IP addresses. Furthermore, an identifier can be associated with multiple entities. This is the case for a URL advertised by two distinct groups of people with different intentions. However, it is important to choose

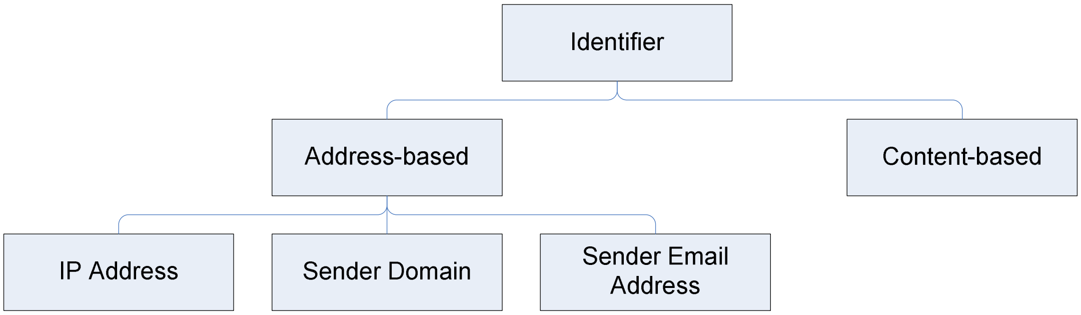

Figure 1: Identifier overview.

a setup in which entities and identifiers strongly correlate.

3.1. Address-based identifiers

3.1.1. IP-based identifiers

The most common identifier that has been used is the IP address of the sender. The entity in this case can be seen as the actual machine behind this IP address or more precisely the group of people using it. There are several advantages to using the IP address as the identifier such as the finite number of possible values and the complexity of spoofing an IP address. However, there are disadvantages. A machine can change its IP address, be compromised and used to send spam, its IP address can be hijacked using attacks on the Border Gateway Protocol (BGP), or legitimate users can share an IP address with others that send spam. In all of these cases, this changes the entity behind the IP address.

3.1.2. Domain-based identifiers

Sender domain name is another identifier type that can be used with the presence of domain-based authentication, which is needed since the from address seen in an email can be easily spoofed. One system to assure the authenticity of the from domain is SenderID [7]. SenderID matches the IP address of the sending mail server against the from domain using a domain to IP mapping stored in the DNS. Another authentication scheme is DKIM [8]. In DKIM, a message is signed with a private key associated to a domain. The domain advertises the public key in the DNS so that the receiver can verify the signature. Using those mechanisms, the domain can serve as the identifier. The advantage over IP-based identifiers is that entities can change the IP address of their mail server as long as they keep the same from domain. The disadvantage is that this approach is more course-grained. For example, a big ISP that uses the same from domain for their corporate mailboxes as for their users' mailboxes will show up as one identifier even if the corporate mail server uses a different IP address.

3.1.3. Email address-based identifiers

A more fine-grained identifier mechanism can utilize the full email address of the user, i.e. a domain-based identifier plus the local part. Since the local part can also be spoofed, validation mechanisms must be in place. Examples of those are LDAP and Active Directory.

3.2. Content-based identifiers

The entity to be identified can also be the actual message. There are several approaches to uniquely and accurately represent a message including fingerprinting, approximate text addressing, or digest-based indexing. Also, the call to action advertised in a spam message such as a URL can serve as an identifier. This type of identifier can identify a particular spammer trying to advertise a specific product.

3.3. Discussion

There are many ways to address the many-to-many mapping issue of entities and identifiers. For example, IP address identifiers can be combined based on

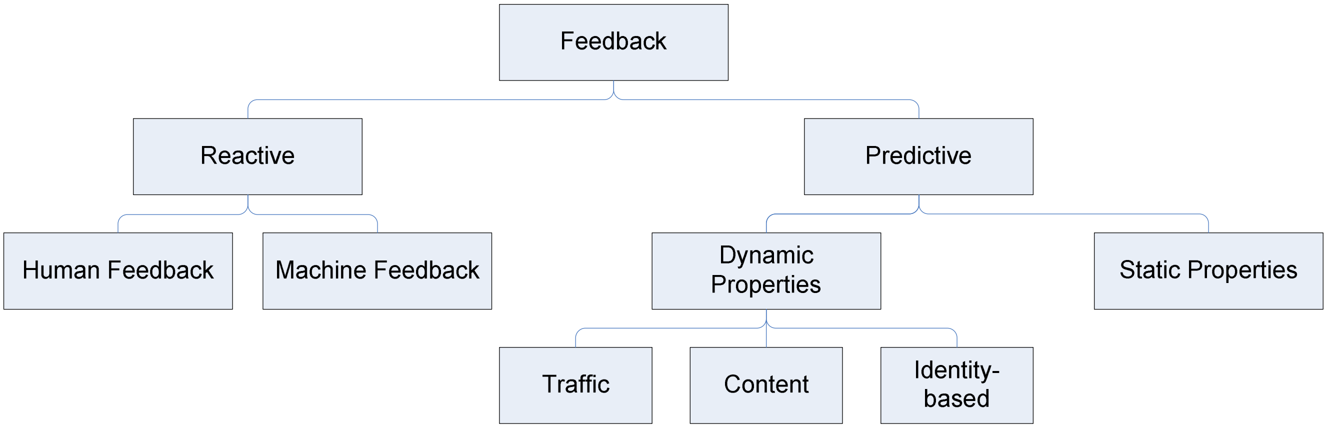

Figure 2: Feedback overview.

WHOIS netblock information. Also, SPF/SenderID can be seen as a way to combine IPs in a single domain into one identifier.

In contrast to traditional Real-time Blacklist (RBL) approaches, reputation systems are generally designed to act quickly on changes in the behavior observed for an identifier. When the entity behind an identifier changes, the reputation system should quickly update the reputation assigned to that identifier.

4. Feedback about interactions

The two categories of feedback about interactions of entities that are used as input into the analysis and assigning of a reputation to that identifier are reactive feedback and predictive feedback, which we will outline in the following sections. Figure 2 gives an overview of these categories and their subcategories.

4.1. Reactive feedback

Reactive feedback is an input into the reputation system that classifies an identifier based on human or machine-provided classifications of that identifier's behavior. Examples include users reporting observed malicious activity associated with an identifier, such as viruses or directory harvesting attacks (DHA) originating from an IP address or domain. It also includes automated feedback from an email classification system that identifies spam messages based on textual analysis or destination address, such as a spamtrap address that is not associated with any legitimate email account, and reports them and any associated identifiers to the reputation system. Blacklist systems, such as RBLs and virus and spam signature-based filtering systems, use reactive feedback as input for classification of malicious identifiers like IPs, domains, and messages. There are two types of reactive feedback: human and machine-generated feedback, which we discuss below. Typically, with both types of reactive feedback precautions must be taken to prevent intentional or unintentional data pollution.

4.1.1. Human feedback and collaborative filtering

Human feedback is classification data about identifiers provided by users examining those. Examples of such examination processes are classifying individual messages, classifying IPs, and classifying organizations from which messages originate. Classic solutions for human feedback include users submitting spam or ham messages, distributed voting systems [9], peer-to-peer email filtering systems [10], and accreditation systems [11,12].

4.1.2. Machine feedback

Machine feedback is classification data about identifiers extracted through automated means. Examples include message feeds collected from spamtraps and honeypots, spam feeds as identified by other filtering systems, such as content classifiers or hybrid spam detection systems that include multiple classification techniques. Identifiers, such as sending IP addresses, domains, URLs, and message fingerprints can be extracted from those feeds and classified using the overall classification of the message.

4.2. Predictive feedback

The differentiating factor between reputation systems and blacklists is the use of predictive feedback to classify a set of identifiers prior to observing any behavior from those identifiers. In that sense, the reputation system attempts to predict other identifiers a known entity is likely to use. Predictive feedback can work on dynamic and static properties of identifiers.

4.2.1. Dynamic properties

Dynamic properties include behavioral feature vectors characterizing sets of identifiers based on all activity detected from them. These can be traffic features, such as volume and time of day; content features, such as subject, EHLO/from domain, and MIME types present in the traffic originating from a sending identifier; and identifier-based features, such as relationships among senders. These techniques typically work well when the analysis is performed on vast amounts of messaging data, such as those that can be collected in high-volume ISP environments or through a large distribution of sensors that in combination see a statistically significant percentage of the world's email traffic. Dynamic properties allow reputation systems to rapidly adapt to changes in the messaging characteristics. They fall in multiple categories.

4.2.1.1. Traffic properties. We call properties that consider the volume, frequency, and distribution of identifiers traffic properties.

An example of a reputation system using traffic properties is DCC [13]. DCC uses message fingerprints as identifiers (a content-based identifier) and detects if these fingerprints indicate bulk transmissions. Another common approach is to use IP-based identifiers and measure the volume and the changes in it over time for each IP. TrustedSource allows access to this part of its data to the public.

Martin *et al.* use features extracted from the message content such as the presence of HTML, hyperlinks, etc. or the number of attachments, words, etc. to detect spam messages [14]. In addition, they also consider properties such as the number of messages sent and the number of unique recipients.

4.2.1.2. Content properties. Properties that are observed in transmissions from an identifier are called content properties. Note that there is a difference in traffic properties of content-based identifiers and content properties of other identifiers. A content property is not sufficiently unique to be tied to an entity but is able to yield statistical insight into transmissions from it. For example, Clayton outlines a spam detection technique through analysis of incoming server logs [15] that identifies malicious sending IPs by applying heuristics, such as the number of EHLO domains seen from each IP. Since the EHLO domain can be arbitrarily chosen by the sender, this is an example for content property-based detection. The reputation is assigned to IPs, so the EHLO domain is a content property of an IP-based identifier.

4.2.1.3. Identifier-based properties. Other dynamic properties are directly linked to identifiers. As an example, the age of an identifier can in many cases yield useful information.

Identifier-based properties also include relationships among identifiers. For example, Boykin and Roychowdhury investigate the social email networks generated by data based on from, to, and cc addresses in a user's email corpus [16]. Based on these interactions, they calculate clustering coefficients, which can be used to identify spam messages. The outlined system runs only on one user's system, but the idea can be scaled to a global social network. Goldbeck and Handler extend this concept to a global scale by using user-assigned reputation score propagated throughout the user's social network [17]. Another approach to propagate trust using graphs is the Advogato system [18,19], which groups good and bad nodes based on edge information.

4.2.2. Static properties

Static properties include feature vectors characterizing identifiers based on external properties. The difference to identifier-based properties is that static properties do not change over long periods of time. Examples include hostname, geolocation, time zone, and administrative information like IP network ownership information or domain WHOIS ownership information. Leiba *et al.* consider ranges of IP addresses for which they keep counts of spam and ham messages [20]. The ranges an IP is part of are a static property of the IP while the feedback of ham and spam counts is reactive.

4.3. Analysis approaches

Above we discussed the different features that can be captured and extracted from each of reactive and predictive feedback sources and fed as inputs into the analysis module of a reputation system. The analysis module assigns a reputation to each of the identifiers out of either a complete universe, such as the entire IPv4 address space, or a smaller subset, such as one containing only identifiers for which certain features are present. It uses the vector of features for each identifier and feeds it into a classifier system, such as Bayesian or Support Vector Machine (SVM), the output of which is transformed into a reputation for that identifier.

The amount of spam currently seen on the Internet tends to highly imbalance the data towards the spam side. Precautions have to be taken to be able to model an effective classifier [6].

Since certain types of feedback, such as human-contributed feedback, is susceptible to malicious or accidental data pollution, trust models such as the EigenTrust method [21] of determining the trustworthiness of a data submitter can be used to make the system more resilient to such attacks.

As mentioned above, the analysis can include features based on feedback gained from queries to the system, which has been investigated in previous work [5,6].

5. Architecture

Email reputation systems are faced with the same architectural challenges as other networked data access systems. In this section we discuss the various options for deploying reputation systems, obtaining feedback, and choices for points of enforcement.

5.1. Centralized versus distributed

Reputation systems can be either centralized or distributed. Centralized systems gather feedback data at a central location, which calculates reputation values, which then can be queried by clients. TrustedSource [4] and SenderBase [22] are systems that work in this fashion. In a distributed system, each client requests information from neighboring clients. The client then calculates a reputation locally based on the knowledge it could gather. The MailTrust system uses such a distributed architecture [23]. MailTrust distributes message digests of spam messages through a peer-to-peer overlay network that is built using email messages between nodes and therefore does not require any additional infrastructure other than a client program running within the user's mail client.

5.2. Trusted versus untrusted feedback

Reputation systems can use feedback from trusted and from untrusted sources. Trusted feedback is known to be accurate while untrusted feedback is subject to intentional and non-intentional pollution. Since participating clients provide feedback to the system, this is also related to the issue of a reputation being open for participation of the public.

Systems open to the public tend to use a greater amount of untrusted feedback. An example of such a system is DCC, which uses message volume information from any participating client. This generally requires a trust model to be established concerning the reporting sensors.

Closed systems use a greater amount of trusted feedback but limit the number of sensors. An example of such a system is one in which the feedback sensors are part of a commercial email security appliance where the code is trusted as well as the input from the device.

Considering the openness of a system is actually a question of the licensing model of the reputation system's data. Some reputation systems provide free public information and a different level of information to paying subscribers. Trustedsource.org and Senderbase.org are examples of freely available limited views of commercial reputation systems.

5.3. Multiple layers of enforcement

Organizations enforce email security at several points within their architectures including at the desktop, the mail server, mail gateway, firewall, and at the service provider. The reputation system output can be leveraged at any or several of these points. For example, some organizations enforce different actions at different points depending on the reputation.

At a gateway, the reputation can be used to reject, throttle, or pass through messages. At the mailserver, an anti-spam solution can use the reputation in conjunction with the results of other detection techniques to infer a combined classification for a message. At the end-user desktop, the reputation can be used to move a malicious message into a separate folder or graphically indicate to the user the positive reputation of the entity behind the message within the mail client.

6. Conclusions

In this work we provided a taxonomy and framework for email reputation systems. Email reputation systems have matured over the years, but there remain open problems that can help fully achieve the broad role that email reputation systems can fulfill in securing messaging systems.

As other messaging paradigms outside of email such as Instant Messaging and Voice over IP have become more pervasive, similar threats have emerged in those systems. Similarly, URL reputation that can be used by Web proxies at the network gateway is an area that recently received more interest. Such a reputation scheme can be used to protect users browsing the Web from malware. Thus, there is a need to provide reputation systems for these paradigms. Ideally a single generic reputation system can function based on a set of identifiers from various types of network systems and correlate feedback from across these systems.

Now that several reputation systems are emerging there is a need for a single framework that allows them to be plugged in and consulted easily. The framework presented here provides an outline towards how the different types of existing reputations systems can be integrated. Future research we focus on is the design of novel analytic methods to efficiently classify dynamic properties across multiple identifiers.

References

[1] E. Friedman and P. Resnick, “The Social Cost of Cheap Pseudonyms,” *Journal of Economics and Management Strategy* 10(2): p173-199, 2001.

[2] A. Abdul-Rahman, S. Hailes, “Supporting Trust in Virtual Communities,” in *Proc. 33rd Hawaii International Conference on System Sciences*, pp. 6007-6015, vol. 6, Hawaii, USA, 2000.

[3] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara, “Reputation Systems,” *Communications of the ACM*, *43*(12), pp. 45-48, December 2000.

[4] Secure Computing TrustedSource Website, http://www.trustedsource.org/.

[5] A. Ramachandran, N. Feamster, and D. Dagon, “Revealing botnet membership using DNSBL counter-intelligence,” Proc. 2nd USENIX Steps to Reducing Unwanted Traffic on the Internet, 2006, pp. 49-54.

[6] Y. Tang, S. Krasser, P. Judge, and Y.-Q. Zhang, "Fast and Effective Spam Sender Detection with Granular SVM on Highly Imbalanced Mail Server Behavior Data," in *Proc. 2nd International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, Atlanta, Georgia, USA, November 2006.

[7] SenderID, http://www.microsoft.com/mscorp/safety/technologies/senderid/default.mspx.

[8] J. Callas, M. Delany, M. Libbey, J. Fenton, M. Thomas, “DomainKeys Identified Mail (DKIM),” Internet Draft draft-allman-dkim-base-01, http://mipassoc.org/dkim/specs/draft-allman-dkim-base-01.txt, October 2005, work in progress.

[9] Vipul’s Razor, http://razor.sourceforge.net/.

[10] F. Zhou, L. Zhuang, B. Zhao, L. Huang, A. Joseph, and J. Kubiatowicz, “Approximate Object Location and Spam Filtering on Peer-to-peer Systems,” in *Proc. ACM Middleware 2003*, Rio de Janeiro, Brazil, June 2003.

[11] IADB, http://www.isipp.com/iadb.php.

[12] Habeas, http://www.habeas.com.

[13] Distributed Checksum Clearinghouse, http://www.rhyolite.com/anti-spam/dcc/.

[14] S. Martin, A. Sewani, B. Nelson, K. Chen, A. Joseph, “Analyzing Behavioral Features for Email Classification,” in *Proc. Second Conference on Email and Anti-Spam*, Mountain View, California, USA, July 2005.

[15] R. Clayton, “Stopping Outgoing Spam by Examining Incoming Server Logs,” in *Proc. Second Conference on Email and Anti-Spam*, Mountain View, California, USA, July 2005.

[16] P. Boykin and V. Roychowdhury, “Leveraging Social Networks to Fight Spam,” *IEEE Computer*, vol. 38, no. 4, pp. 61-68, April 2005.

[17] J. Golbeck and J. Hendler, “Reputation Network Analysis for Email Filtering,” in *Proc. First Conference on Email and Anti-Spam*, Mountain View, California, USA, July 2004.

[18] Advogato, http://www.advogato.org/trust-metric.html.

[19] R. Levien and A. Aiken. “Attack resistant trust metrics for public key certification.” in *Proc. 7th USENIX Security Symposium*, pp. 229-241, Berkeley, California, USA, 1998.

[20] B. Leiba, J. Ossher, V. Rajan, R. Segal, and M. Wegman, “SMTP Path Analysis,” in *Proc. Second Conference on Email and Anti-Spam*, Mountain View, California, USA, July 2005.

[21] S. Kamvar, M. Schlosser, and H. Garcia-Molina, “The EigenTrust Algorithm for Reputation Management in P2P Networks,” in *Proc. Twelfth International World Wide Web Conference*, Budapest, Hungary, May 2003.

[22] SenderBase, http://www.senderbase.org.

[23] J. Kong, P. Boykin, B. Rezaei, N. Sarshar, V. Roychowdhury, “Scalable and Reliable Collaborative Spam Filters: Harnessing the Global Social Email Networks,” in *Proc. Second Conference on Email and Anti-Spam*, Mountain View, California, USA, July 2005.