A Behavioural Model for Client Reputation

Anirban Basu, Ian Wakeman, Dan Chalmers and Jon Robinson

Abstract In client-server interaction scenarios over a network the problem of unsolicited network transactions is often encountered. In this paper, we propose a reputation model based on the behavioural history of long-lived network client identities as a solution to this problem. The reputations of clients are shared between trusted servers anonymously through global reputation analysers. Shared global reputations and local reputations help servers to infer local opinions of clients and control service levels in attempts to reduce unsolicited network transactions.

1 Introduction

In client-server interaction the problem of unsolicited network transactions is often encountered. We propose a reputation model based on behavioural history of long-lived network client identities as a solution to this problem.

Allman, et al [4] presents an architectural overview of ‘*a distributed system that provides a lightweight actor-based history database*’. Their proposed system accumulates reports of unwanted traffic across the entire network. The information provided by the history database is used by consumers to determine the validity and trustworthiness of the information. The system can be used to build audit trails of network actors, which is useful in enforcing policies by consumers of this shared behavioural history. However, the architecture described provides a very broad overview of behavioural history. Unfortunately, the architecture is susceptible to some forms of attack due to its openness. For instance, the usefulness of a report can be artificially increased in an attack mounted by malicious identities.

Wei and Mirkovic [21] take the ideas from Allman’s work and extend it by identifying some of the problems with the architecture, whereby they propose building reputations for Internet clients. The authors claim that they are the first to provide

Department of Informatics, University of Sussex, e-mail: a.basu, ianw, d.chalmers, j.r.robinson@sussex.ac.uk

a systematic overview of the differences between client and provider reputations. Assuming a realistic adversary model and an open participation reputation system, the authors identify challenges that are unique to client reputation systems. Their system describes a combination of two different ways in building reputation – one way is for servers to rank their experiences with clients whilst the other way is to allow independent observers to rank client behaviour using observations of traffic patterns. However, they define behaviour as either ‘good’ or ‘bad’ when viewing the low-level network traffic. This, unfortunately is not observed in real-world conditions, where behaviour is not absolutely good or bad and can be interpreted against expectations.

Our work addresses the problems of openness and reliance on low-level traffic patterns by postulating a general attack model and by abstracting the definition of good and bad behaviour. We propose a system, which uses behavioural history as a mechanism to augment other measures against unwanted network traffic. We predict that reputations built on behavioural history would be useful in efforts to inform policy decisions for future client-server interactions. In this paper, we explore the research question: *can a local and a global reputation scheme based on behavioural history of long-lived network identities be used to enforce policies for future network interactions?* We describe a system that uses a notion of long-lived network identities that are associated with a corresponding behavioural history. We define an open-ended behaviour analyser, which helps develop the concept of good and bad behaviour of clients conforming to offline agreements and policies. Given locally recorded histories of behaviour of clients, servers can interpret their local reputations, which they periodically report to global reputation aggregators. These global reputations can be queried by other servers, which can infer their views on the clients before providing service. Both the local and the global reputations help servers decide service levels to clients. Also, at times of bottleneck network conditions, servers can choose to deliver service to clients with higher reputations only.

2 Related work

In this section, we discuss related work from the areas of anti-spam systems; trust and reputation systems; and behavioural history.

Anti-spam systems

Email spam is a well established area of traffic management in which reputation is applied. In October 2007, it was observed that over 74.5% of all emails sent over the Internet comprised of spam[14]. There has been substantial academic research as well as industry initiatives towards solving these problems [25, 2, 3, 5, 9] using content filtering, quota management and social networks amongst other techniques.

In particular, SpamAssassin [3] is a well-known open source product that uses Bayesian content filtering [18] of email messages. SpamCop [17] is another example of text-based spam classificiation. Sender Policy Framework [2] is a commercial means of spam identification using a form of sender identification. Other research have used social network relations against spam [9]. Use of P2P overlays in email spam protection is discussed in [5].

Research has also been done towards combatting web spam (e.g., [16, 22]). Google's PageRank [16] is a well-known way for ranking web pages on a logarithmic scale based on their importance.

Trust and reputation systems

There has been a large number of academic research interests in the area of trust and reputation. We present a brief survey of research that places our work in to context.

Eigentrust [12] is a secure and distributed mechanism of computing global trust values, based on eigenvector calculation for nodes in a peer-to-peer network. PowerTrust [24] proposes a fully decentralised trust model based on power-law distribution in user feedbacks. Pinocchio [8] describes a framework for providing incentives for honest participation in large distributed trust management infrastructures. Guha, et al [10] discusses different mechanisms of propagation of trust and distrust in the context of e-commerce and recommendation systems. Jøsang and Pope [11] describe the principles for expression and analysis, and requirements for validity of transitive trust networks. [23] talks about detection of deception in testimony propagation and aggregation in distributed reputation management, with particular reference to an earlier work of theirs on management of reputation. Damiani, et al in [6] describes a reputation sharing system in an attempt to stop the spread of Trojan Horses, viruses and spam over P2P networks.

Behavioural history

The problem of unsolicited messages has gone beyond the domain of email spam. There is an increasing concern about spam over Internet telephony [20] where there is active research being conducted on the prevention of voice spam (e.g., [13]). We envisage that the problem of unsolicited messages can be generalised to any unsolicited network transaction between a client and a server. In most cases, client-server interactions over the Internet are largely anonymous. A server providing a particular service to a client often has no idea about the client's prior activity history. This leaves the server with the option of making one-off decisions about the interaction through content filtering or any other form of analysis based on the network traffic. Allowing a server to have an advanced knowledge of both local and global behaviour of a client's reputation, based on its prior behavioural history, aids the former in its decision making process.

There are different mechanisms in which servers can take advantage of behavioural history. For example, an intrusion detection system may maintain a local cache to track behaviour exhibited by remote hosts. Alternatively, information collected about a particular client could be shared by remote entities [19]. Websites, such as DShield [1] aggregate activity reports about clients to forecast threats on the Internet. Similar means are undertaken in client blacklisting and whitelisting procedures.

3 Proposed solution

To record behavioural history the notion of long-lived identities is necessary. We assume the presence of a record of long-lived identities of all network entities. A namespace based identity architecture is described in [15]. Unless otherwise mentioned, we assume a Public Key Infrastructure (PKI) as the identity mechanism throughout the rest of the paper.

We identify three types of network entities – *clients*, *servers*, and the *Global Reputation Analyser (GRA)*. Servers provide services, which clients consume. Servers maintain local observations of behavioural history as well as reputation of clients, that they provide service to. Servers also report the local reputations of their clients to the GRA, which can be queried by other servers to obtain interpretations of global reputations of clients. Global and local (if available) reputations are useful to servers for determining levels of service provided to clients. Such variations in service levels help control the proliferation of unsolicited network transactions.

While the GRA appears to be a single entity to servers, it is likely to be implemented, in a real world scenario, as a cluster or a P2P overlay of closely administered and trusted nodes. For example, an overlay could be formed out of geographically dispersed ISP backbones.

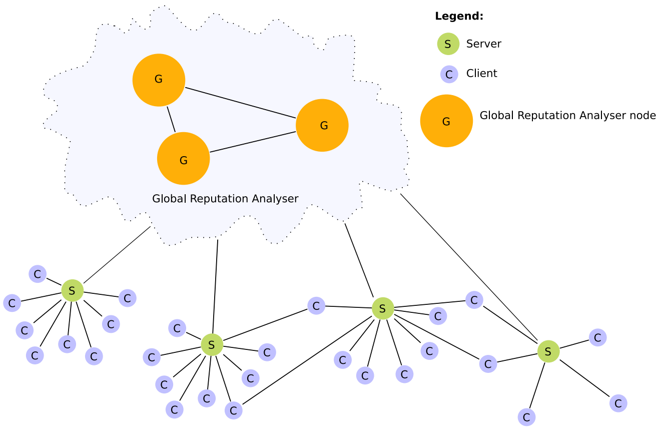

Fig. 1 Conceptual overview of a system of four servers with some common clients and a Global Network Analyser

A conceptual diagram of a system with four servers, some common clients and a distributed GRA is illustrated in figure 1.

Behaviour analysis

The reputation of a client is formed by analysing its behaviour. However, the interpretation of “good” or “bad” behaviour is often relative; hence dependent on policies implemented by specific servers. Behaviour interpretation can be achieved through a variety of monitoring mechanisms. For example, a Bayesian spam filter could scan text in a message to determine its rank as spam. On the other hand, a router traffic monitor could detect how many simultaneous connections are opened by a particular client at any particular period in time. More than one such monitoring system may be used at a time to gather information about a client. Generic data is provided by these monitoring systems and is expressed as a tuple $\tau = \{client\_id, observed\_values, observed\_type, timestamp\}$. We define $client\_id$ as the long-lived identity of the client; $observed\_values$ as the set of output values of the monitor (e.g., $\{0.9\}$ on a scale of 0 to 1 from a spam filter); $observed\_type$ as the type of the monitor (e.g., Bayesian spam filter) and $timestamp$ as the time at which the monitor observed a client behaviour. This $timestamp$ is used to detect multiple occurrences of similar observations. The behaviour analyser maintains a history of previously observed behaviour. The implementation and specific policies determine how large this recorded history can be.

Using this as an input, a policy-specific behaviour analyser can be implemented for specific servers. Such an analyser keeps a history of such τ-tuple inputs and outputs the interpretation of behaviour in discrete integral units (both positive and negative), which is fed into the reputation system. This design keeps the behaviour analyser open-ended and it can augment existing network activity monitoring systems.

Local reputation

In this section, we will be outlining how local reputation is formed from behaviour and also how it is affected by the lack of activity over time.

Reputation response to behaviour A local reputation response is based on the discrete behaviour input corresponding to the locally observed client behaviour. We chose a reputation response with the following intuitive characteristics:

- Good reputation gets better with good behaviour until it reaches a positive saturation. The gradient of improvement slows as reputation rises.
- Good reputation will decrease more rapidly with bad behaviour than it will improve with good behaviour.
- Bad reputation gets worse with bad behaviour until it reaches a negative saturation. The gradient of worsening reputation becomes less steep as the reputation falls.
- Bad reputation increases with good behaviour at a slower rate than it worsens with bad behaviour.

An alternative reputation response with different characteristics could be used depending on the security policy requirements. We use the reputation response described above as a model in this paper.

The following mathematical model has been found to fit the chosen reputation response. Let us assume that client reputation is denoted with r; behaviour variable with b; positive saturation with r_{psat}; negative saturation with r_{nsat}; and two adjustable response parameters λ and μ. Here, $r_{psat} = 1$ and $r_{nsat} = -1$. Other values of positive and negative saturation may be considered in future. Also, for any event (v) for which a change of behaviour is reported, the corresponding cumulative behaviour is b_v and the corresponding reputation is r_v. In addition, p and n, respectively signify positive and negative values.

The equations are presented below. The equation for good reputation getting better with good behaviour is:

$$r = r_{psat}\left(1 - e^{-\lambda b}\right) \quad \text{for} \quad \Delta b > 0, b > 0, r_{v-1} \geq 0 \tag{1}$$

and the equation for bad reputation getting worse with bad behaviour is:

$$r = r_{nsat}\left(1 - e^{\lambda b}\right) \quad \text{for} \quad \Delta b < 0, b < 0, r_{v-1} \leq 0 \tag{2}$$

and the equation for good reputation getting worse with bad behaviour is:

$$r = \frac{r_{v_p}}{b_{v_p}} b \quad \text{for} \quad \Delta b < 0, b > 0, r_{v-1} > r_v \geq 0 \quad \text{and} \quad r_{v_p} = r_{psat}\left(1 - e^{-\lambda b_{v_p}}\right) \tag{3}$$

and the equation for bad reputation getting better with good behaviour is:

$$r = \frac{r_{v_n}}{\left(1 - e^{\mu b_{v_n}}\right)}\left(1 - e^{\mu b}\right) \quad \text{for} \quad \Delta b > 0, b < 0, r_{v-1} < r_v \leq 0 \quad \text{and} \quad r_{v_n} = r_{nsat}\left(1 - e^{\lambda b_{v_n}}\right) \tag{4}$$

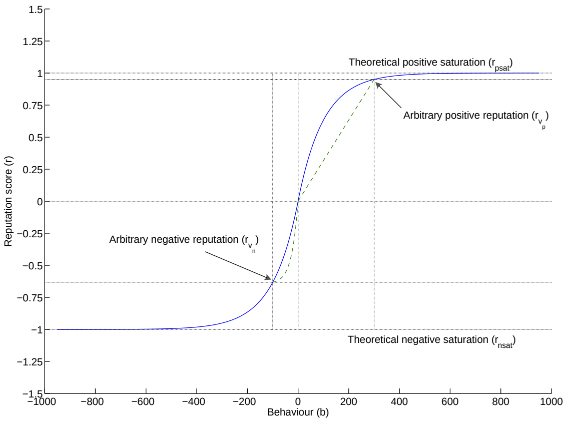

Fig. 2 Graph of reputation versus behaviour

Figure 2 combines equations 1, 2, 3 and 4 to illustrate the nature of reputation response to behaviour. Calculation of reputation is stopped when the reputation value is close enough to either positive or negative saturation (e.g., within 0.1%). It is evident from the graph that an identity with a high reputation (i.e., near positive saturation) will not be able to exploit it because poor behaviour will result in its reputation being reduced along the linear curve in the aforementioned figure (first quadrant). Similarly, if an identity has a bad reputation, it would require a demonstrable amount of good behaviour to improve its standing (third quadrant).

Time decay of reputation

Saturated reputation denotes “too good” or “too bad”, which often needs a decay with no activity over time. This helps a saturated bad reputation to recover slowly with time. It also questions a saturated good reputation if there has been no activity over time. A neutral zone (default values) $[r_{ndef} \quad r_{pdef}]$ such that $r_{nsat} < r_{ndef} < 0$ and $0 < r_{pdef} < r_{psat}$ is defined for this purpose. Positive reputation higher than r_{pdef} decays to the positive default, while negative reputation lower than r_{ndef} ‘decays’ (in essence, increases) to negative default. An adjustable decay rate parameter ε is introduced in this context. The equation for positive reputation decaying over time is given as:

$$r = \begin{cases} r_{v_p}\left(1-\varepsilon t^2\right) & \text{for } r \geq r_{pdef} \\ r_{pdef} & \text{for } r < r_{pdef} \end{cases} \tag{5}$$

and the equation for negative reputation increasing over time is given as:

$$r = \begin{cases} r_{v_n}\left(1-\varepsilon t^2\right) & \text{for } r \leq r_{ndef} \\ r_{ndef} & \text{for } r > r_{ndef} \end{cases} \tag{6}$$

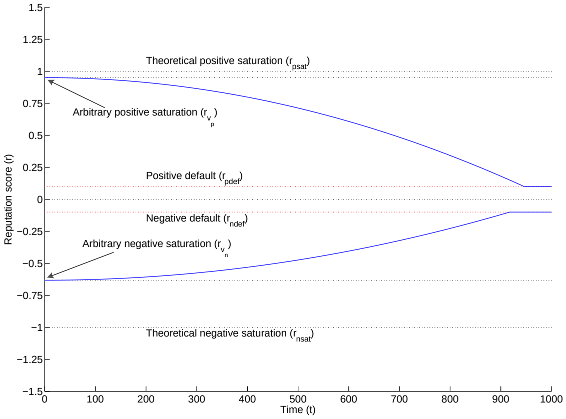

Fig. 3 Graph of reputation versus time

Figure 3 illustrates equations 5 and 6. Once the decayed reputation reaches the boundaries of the neutral zone, the decay stops.

Reputation reporting mechanism

Observed local reputations are submitted by servers to the GRA. The process of submission can either happen at the end or during an on-going service. Submission can be made several times as long as the evidence of service interaction between a server and a client is valid. This evidence is the authorisation token, which the client gives to the server. The different steps of the reputation reporting mechanism are illustrated below. It is essential that prior to this reporting mechanism, the identities of the client and the server are known to the GRA. Any communication between a client and the GRA or between a server and a GRA happen over Secure Sockets Layer (SSL).

1. When the client (C) requests a service from server (S), it provides the server with a signed authorisation token (AT). This token may contain information relevant to the context of implementation but will at least contain a time-stamp signifying the expiry of the token and the server identifier of the requesting server. C provides S with a token signed with its private key – $(AT)_{C_{priv}}$.
2. C also sends $(AT)_{C_{priv}}$ to the GRA to keep track of the active token.
3. S re-signs the token $((AT)_{C_{priv}})_{S_{priv}}$ and sends this to the GRA to query the global reputation of C.
4. The GRA performs a token authenticity check confirming that: the two copies of AT are equal; the copy from S comes from the S identified in the AT by C; and that the time-stamp is valid.
5. On a successful authenticity check, the GRA calculates the global reputation of C with respect to S's confidence in other servers that may have submitted reputations of C. This is accomplished through a social network of servers.
6. S makes inferences from the global reputation of C.
7. S provides service to C and uses the reputation-to-behaviour response mechanism to form its local reputation of C which, if required, is used to change service levels.
8. Either at the end or during an on-going service, S sends the $(AT)_{C_{priv}}$ and its local reputation of C along with other necessary parameters to the GRA – $(\gamma, (AT)_{C_{priv}})_{Spriv}$. Please refer to the discussion on global repuation for the definition of γ.
9. The GRA performs a token authenticity check. If successful, the GRA records the reported reputation and other associated parameters, scavenges out-of-date reputation reports and invalidates the authorisation token. If S has reported a reputation for C in the past then the former is overwritten with the latter if both reports are in the same $context$.
10. If S wants to report more than once during an on-going service, it has to request a new authorisation token from the client every time.

Social network of servers and confidence matrix

In steps 5 and 6 of the reputation reporting algorithm, we noted that the GRA provides a reputation of the client tailored for the querying server. In order to achieve this we use a "social network" model the servers relationships to form their confidence on each other. We now describe how these are formed.

Social network with confidence ratings Servers form a social network, which defines the confidence that each server in the network has on one another. Servers can choose to connect or disconnect from other servers in the social network at any time. This social network is an asymmetric weighted digraph that has nodes (servers), which are connected through directed edges with different *confidence ratings* (denoted by w with range $[0 \quad 1]$) on each edge. Thus, two servers v_i and v_j can be connected through edges $e_{i,j}$ and $e_{j,i}$, which have different confidence ratings ($w_{i,j}$ and $w_{j,i}$). These confidence ratings are used by querying servers to determine their confidence in the global reputations of clients. The confidence ratings can be altered by the servers at any time based on a variety of reasons which may be implementation or policy specific and hence fall outside the remit of this paper. Therefore, if the server v_j has submitted a reputation r_j for a particular client then as a response to a query from server v_i about the client, the confidence on this particular report will be $w_{i,j}$. The degree of separation in the network is no more than one. Therefore, if servers v_i and v_j are connected; and servers v_j and v_k are connected then there is no transitive confidence from v_i to v_k unless they are themselves connected. This method of relative interpretation of reputation makes it more immune to attempts to fix reputation.

Confidence matrix A confidence matrix is essentially the adjacency matrix of the social network graph with weights on connected edges, which will be re-computed only when servers alter their confidence ratings about other servers.

Global reputation

Global reputations of clients act as opinions shared between servers. The interpretation of the global reputation depends on the perspective of a querying server. Through the process of reputation reporting, a set of submitted reputations are attached to a particular client and maintained by the GRA. The size of this set grows in proportion (e.g., logarithmically) to the total number of servers the client has interacted and is interacting with. There is an age-based method of scavenging that prunes off older reputation values to make space when newer values are submitted.

For a particular client C_i, the tuple of reputation and associated parameters submitted by a server S_j can be defined as $\gamma_{i_j} = \{context_j, r_{i_j}, \lambda_j, \mu_j, t_{report_j}, v_j\}$ where v_j is the vertex for S_j in the social network graph; r_{i_j} is the reputation of the client C_i submitted by server S_j for a particular application context $context_j$; t_{report_j} is the timestamp of reporting this tuple; and the λ_j and μ_j correspond to those defined in equations 1, 2, 3, and 4. In the γ-tuple, *context* identifies the application context over which the behaviour analysis was done to generate the corresponding reputation. This could be, for example, email spam analysis, or low-level TCP packet analysis. We leave the semantics of this *context* for future work. Over time when several servers submit such reputations for the client, a set of recorded reputations is defined as $\Gamma = \{\gamma :$ where r component in γ is not too old$\}$. To determine whether r in γ is too old or not, we use the following age-based scavenging method.

In a particular γ-tuple at any time t:

- if $0 < r \leq 1$ and $\lambda(t - t_{report})^2 = 1$ then the tuple is scavenged;

- if $-1 \leq r < 0$ and $\frac{\lambda}{\mu}(t - t_{report})^2 = 1$ then the tuple is discarded;
- if, however, $r = 0$ the tuple is discarded when both conditions are met.

This scavenging method ensures that reputations that have been generated by servers with tougher reputation to behaviour response conditions (i.e., lower λ and μ) are decayed slower than the ones with more lenient conditions. The characteristic of the decay is similar to that defined in equations 5 and 6.

If a server S_i (corresponding to vertex v_i) queries the global score of the client C_i for a particular context ($context_j$) or for all contexts, the query is answered as follows:

- If v_i has no egress edges then the query returns nothing.
- Otherwise, the query returns a set of client reputations (rs) from the j-th γ-tuples for each of which the egress path from v_i to v_j exists. The returned reputations are chosen from the ninth and the first deciles of their corresponding confidence ratings $w_{i,j}$.
- If the Γ set contains γ-tuples that have been submitted by servers which are not directly connected to the querying server then the set of client reputations that is returned contains those with the ninth and the first decile of γ-tuples arranged according to their reputation values.

What the querying server does with the reputation and the confidence is implementation and policy specific. It may, for example, choose to seed its local reputation for the client from the product of the global reputation and the confidence only if the confidence is above a certain level. Another possibility is that it may wish to alter the values of its λ and μ parameters depending on the value of the confidence and then seed its local reputation directly from the global reputation.

4 Adversary model

In this section, we discuss a number of attacks on our model and the possible defences against such attacks.

Manipulation of global reputation

Attack: A group of servers conspire collaboratively to either increase or to decrease the global reputation of a client. This may be done to deliberately confuse a genuine server such that a particular client with a bad reputation appears good on a global scale. In this attempt, the malicious servers and client collaborate and the servers submit unusually high reputations about the client. The other possibility is that some malicious servers can try to make a good client look bad in its global reputation. The servers can do this by submitting unusually low reputations about the client.

Defence: The genuine querying server will need to have high confidence factors on the malicious servers to be tricked into reputation fixing. It can have high confidence in the malicious servers if it has asserted, in the past, its direct confidence in such servers, which is unlikely. Also, the characteristics of clustering in the social network suggest that the querying server is unlikely to be very close to any of the malicious servers in terms of path length. If the path length is long then the confidence factor diminishes and it becomes zero if the path length is larger than the separation threshold. There is an open issue on the feedback of global reputation that still needs to be addressed.

Exploitation of reputation

Attack: A client which has a high reputation uses this to behave badly so that it could possibly take advantage of any time delay before its reputation decreases. The client can exhibit good followed by bad behaviour repeatedly.

Defence: A near saturated high reputation is calculated until it is close enough to saturation (e.g., within 0.1%). Any further good behaviour from that point forward does not increase the reputation any more. This means that bad behaviour reduces the reputation along the linear curve in the first quadrant of figure 2. In addition, the behaviour analyser can detect small repetitions of behaviour change and penalise the client according to some policy.

A corollary of this attack is that a client with a negative reputation could expect to improve its reputation by performing some good activity and then exhibiting bad behaviour and so on repeatedly. The local reputation response model is resistant to this attack because the client has to behave substantially well to be able to improve its negative reputation (see third quadrant in figure 2). Also, if the negative reputation is such that the client has been denied all service levels then it has to wait for a time decay to improve its reputation by doing nothing for a considerable period of time. In addition, short bursts of good behaviour followed by short bursts of bad behaviour, which constitute a repetitive nature can be detected by the behaviour analyser and the client can be penalised accordingly.

DoS or DDoS attack on the GRA

Attack: The GRA is under a Denial of Service or a Distributed Denial of Service attack, which prevents any server from submitting their observations about clients or querying global reputations about clients. Malicious clients can use this attack to stop servers from reporting their reputations to the GRA.

Defence: There is no built-in defence against a DoS or DDoS attack in our model but the GRA is free to implement innovative ways of DoS or DDoS protection, such as the overlay approach in [7]. Even if the GRA was successfully knocked off the network, each individual server can still use their local reputation response mechanisms to continue developing inferences on client reputation and accordingly adjust service levels to clients. Therefore, temporary unavailability of the GRA does not affect the service provided by a server to a client if the provision of the service has already started. Once the GRA becomes available, the servers are able to resynchronise the reputations of clients. During the period of unavailability of the GRA, a server will start off any unknown client identity with zero local reputation. Also, if the GRA is designed, for example, as an overlay of several nodes on a distributed hashtable then the impact of a DDoS attack is lessened in the absence of a central point of failure.

DoS or DDoS attack on the server

Attack: Several clients with the same malicious intentions register as new identities and ask servers for service. New identities have no past behavioural history and hence the malicious intention of one client cannot be inferred from the behavioural history of any other client.

Defence: This attack is a precursor to a Denial of Service attack on a server. Every new client identity so far unknown to any other server and to the GRA will start off with zero reputation. A careful implementation of service levels could mean that at zero reputation, the level of service provided is very basic. The client will need to prove its ability before its reputation improves and is promoted to a higher service level.

Man-in-the-middle attack

Attack: A malicious network entity tries to intercept the communication between clients and the GRA or servers and the GRA, in order to manipulate reputation reports, authorisation tokens, or results of reputation queries.

Defence: Our model is resistant to this attack because all communications between a client and the GRA or a server and the GRA happen over SSL. In addition, reputation reports from servers and authorisation tokens are digitally signed. This makes it impossible for a man-in-the-middle attacker to change any data communicated between both parties.

5 Conclusion and future work

In this paper, in an attempt to resolve some of the open issues with previous work on behavioural history, we have proposed a system for developing local and global client reputation from their behavioural histories in client-server interactions. This paper presents a work-in-progress and considerable amount of work needs to be done to make this a feasible system in practical network environments.

As future work we plan to:

- Use mathematical analysis on or simulation of our proposed model.
- Evaluate the performance of the proposed model with behavioural data. We will be using both data directly from web spam traces and synthetically generated from models of good and bad identities derived from network traces.
- Address an open issue where a feedback from the querying server on the global reputation of a particular client may be used to alter the querying server's confidence in some of the other servers, which have been used to build the response to the reputation query;

References

1. Dshield. URL http://www.dshield.org/
2. Sender Policy Framework. URL http://www.openspf.org/
3. The Apache SpamAssassin Project. URL http://spamassassin.apache.org/
4. Allman, M., Blanton, E., Paxson, V.: An Architecture for Developing Behavioral History. In: Proc. Workshop on Steps to Reducing Unwanted Traffic on the Internet (2005)
5. Chung, A., Tarashansky, I., Vajapeyam, M., Wagner, R.: SpamStrangler: A Chord-Based Distributed Spam Detection Tool
6. Damiani, E., De Capitani Di Vimercati, S., Paraboschi, S., Samarati, P.: Managing and sharing servants' reputations in P2P systems. Knowledge and Data Engineering, IEEE Transactions on **15**(4), 840–854 (2003)
7. Ellis, D., Wakeman, I.: Lessons for Autonomic Services from the Design of an Anonymous DoS Protection Overlay. Lecture Notes in Computer Science (Springer) **4195**, 86 (2006)
8. Fernandes, A., Kotsovinos, E., Ostring, S., Dragovic, B.: Pinocchio: Incentives for Honest Participation in Distributed Trust Management. In: Trust Management: Second Intl. Conf., ITrust 2004, Oxford, UK, March 29-April 1, 2004: Proc. Springer (2004)
9. Garriss, S., Kaminsky, M., Freedman, M., Karp, B., Mazières, D., Yu, H.: Re: Reliable Email. In: Proc. of the 3rd Symposium of Networked Systems Design and Implementation (NSDI '06) (2006)
10. Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: Proc. of the 13th Conf. on World Wide Web, pp. 403–412. ACM Press (2004)
11. Jøsang, A., Pope, S.: Semantic constraints for trust transitivity. Proc. of the 2nd Asia-Pacific Conf. on Conceptual modelling-Volume 43 pp. 59–68 (2005)
12. Kamvar, S., Schlosser, M., Garcia-Molina, H.: The Eigentrust algorithm for reputation management in P2P networks. In: Proc. of the 12th Intl. Conf. on World Wide Web, pp. 640–651. ACM Press (2003)
13. Kolan, P., Dantu, R.: Socio-technical defense against voice spamming. ACM Transactions on Autonomous and Adaptative Systems **2**(1), 2 (2007). DOI http://doi.acm.org/10.1145/1216895.1216897
14. MessageLabs: Messagelabs intelligence: October 2007 (2007). URL http://www.messagelabs.com/intelligence.aspx
15. Moskowitz, R., Nikander, P.: RFC4423: Host Identity Protocol (HIP) Architecture (2006). URL http://www.ietf.org/rfc/rfc4423.txt
16. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998 (1998)
17. Pantel, P., Lin, D.: SpamCop: A Spam Classification & Organization Program. In: Proc. of AAAI-98 Workshop on Learning for Text Categorization (1998)
18. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A Bayesian approach to filtering junk e-mail. In: Learning for Text Categorization: Papers from the 1998 Workshop, vol. 62. Madison, Wisconsin: AAAI Technical Report WS-98-05 (1998)
19. Sommer, R., Paxson, V., Munchen, T.: Exploiting Independent State For Network Intrusion Detection. Computer Security Applications Conf., 21st Annual pp. 59–71 (2005)
20. Technology Review (MIT): Kill Voice Spam Before It Grows (2004). URL http://www.technologyreview.com/Infotech/13823/?a=f
21. Wei, S., Mirkovic, J.: Building Reputations for Internet Clients. Electronic Notes Theoretical Computer Science **179**, 17–30 (2007). DOI http://dx.doi.org/10.1016/j.entcs.2006.11.033
22. Wu, B., Goel, V., Davison, B.: Propagating Trust and Distrust to Demote Web Spam. Proc. Models of Trust for the Web Workshop (MTW), Intl. World Wide Web Conf. (2006)
23. Yu, B., Singh, M.: Detecting deception in reputation management. Proc. of the second international joint Conf. on Autonomous agents and multiagent systems pp. 73–80 (2003)
24. Zhou, R., Hwang, K.: PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing. IEEE Transactions on Parallel and Distributed Systems **18**(4), 460–473 (2007)
25. Zimmermann, P.: The official PGP user's guide. MIT Press Cambridge, MA, USA (1995)