CallRank: Combating SPIT Using Call Duration, Social Networks and Global Reputation*

Vijay A. Balasubramaniyan, Mustaque Ahamad and Haesun Park
College of Computing, Georgia Institute of Technology
Atlanta, Georgia 30332
{vijayab,mustaq,hpark}@cc.gatech.edu

ABSTRACT

The growing popularity of IP telephony systems has made them attractive targets for spammers. Voice call spam, also known as Spam over Internet Telephony (SPIT), is potentially a more serious problem than email spam because of the real time processing requirements of voice packets. We explore a novel mechanism that uses duration of calls between users to combat SPIT. CallRank, the scheme proposed by us, uses call duration to establish social network linkages and global reputations for callers, based on which call recipients can decide whether the caller is legitimate or not. CallRank has been implemented within a VoIP system simulation and our results show that we are able to achieve a false negative rate of 10% and a false positive rate of 3% even in the presence of a significant fraction of spammers.

1. INTRODUCTION

Voice over Internet Protocol (VoIP) systems rely on an IP network to set up voice calls and transmit voice packets. The growing popularity of VoIP, the relatively low cost of access to IP networks, and the vulnerabilities that exist in systems connected to such networks makes VoIP an attractive tool for spammers. Spammers and telemarketers will use SPIT to make unsolicited calls and to send voice mails for the same purposes for which email spam is currently used. SPIT would not only degrade our confidence in telephony but it would be more difficult to handle because of the real-time processing requirements of voice calls. Examples of large scale SPIT already exist - a company sent out voice mails to all its customers detailing its initial public offering[15]. If we are not able to combat SPIT effectively, we face an unhappy future where picking up a ringing phone would be a frustrating experience and voice mailboxes would become clogged with advertisements for unwanted products.

The first stage of voice communication is call setup, a handshake mechanism between the caller and the call recipient after which the phones start ringing. At this stage the only information available is the identity of the caller and the call recipient. It is only after the call recipient accepts the call, that voice media is exchanged. A spam engine that filters based on the media content, however successful it is, will not be able to prevent the phone from constantly ringing. In addition voice packets, unlike email, must be delivered to the user synchronously. Any delay in delivery due to spam engine processing will result in degraded call quality. Thus, an effective method for dealing with SPIT must rely on the identity of the caller rather than call content. However, determining the exact identity of a user on the internet is a hard problem. It is sufficient if we are able to differentiate between a legitimate caller and a spammer. In this paper, our focus is on developing a scheme that achieves this goal.

This paper proposes CallRank, a novel mechanism built around call duration, to differentiate between a legitimate user and a spammer. Our approach is motivated by the simple observation that a legitimate user typically makes and receives calls and many of the calls last for significant durations. On the other hand a spammer's/telemarketer's goal is to deliver information to as many people as possible by making a large number of relatively brief calls. A spammer will typically receive no calls or a much smaller number of calls. The difference in call patterns is that, for a spammer, the call pattern is largely unidirectional while it is bidirectional for legitimate users. We take advantage of this difference in call patterns and use call duration to create *call credentials* that callers can provide to call recipients as proof of an implicit level of trust.

The following simple scenario shows how our call credential based approach can be used to identify spammers. Assume that *Alice* makes a call to *Bob*. If *Bob* picks up the phone and talks to *Alice* a call credential can be generated, after completion of the call, signifying that *Bob* and *Alice* trust each other enough to have talked for the duration of their call. *The longer the call duration, stronger is the call credential.* Intuition suggests that if a user receives calls of significant duration on a regular basis it is likely that he[1] is a legitimate user and not a spammer. There are several ways in which call credentials can be created when calls are made. For example, when *Alice* calls *Bob* and talks to him for time t, she can create a call credential and provide it to *Bob*. It is also possible that the recipient of the call, *Bob*, generates a call credential for *Alice* or both generate call credentials for each other. In this paper, however, we explore a mechanism where a caller provides a call credential to the call recipient when he makes a call and speaks to the call recipient.

For each user we use these call credentials to determine

*This work was supported in part by funding provided by IBM ISS and AT&T.

CEAS 2007 - Fourth Conference on Email and Anti-Spam, August 2-3, 2007, Mountain View, California USA

[1]we use the pronoun *he* for both users and spammers for the sake of convenience

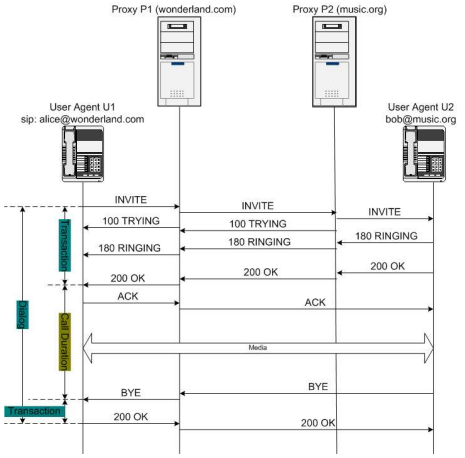


Figure 1: SIP Call Trapezoid

Social Network (SN)[1] linkages, thus enabling us to distinguish between legitimate users and spammers. We also use call duration along with the Eigentrust algorithm[10] to develop a global view of the reputation of all users who either belong to or interact with a domain. For a spammer to be successful in a system that employs CallRank, he must get other legitimate users to call and speak to him for significantly long durations. We believe this will be extremely hard as people rarely call up a spammer. If they inadvertently do make a call to a spammer, the conversation will not last for very long.

The following are the key contributions of this paper:

- We introduce call duration based credentials as the uniform underlying mechanism to support a number of techniques to determine if a caller is a spammer.
- We explore the use of SNs based on call credentials to allow two users to make a call.
- If SN linkages are unavailable between users, we use a variation of the Eigentrust algorithm to assign global reputations based on call durations.
- We perform a detailed evaluation of CallRank and show that we are able to achieve low false negative and low false positive rates even in the presence of a significant fraction of spammers.

The rest of the paper is as follows. Section 2 discusses VoIP basics, SNs, Eigentrust algorithm and other related work that deals with the spam problem. The key components of CallRank are presented in Section 3. An evaluation of CallRank and its results are discussed in Section 4. This is followed by conclusion and future work in Section 5.

2. BACKGROUND AND RELATED WORK

2.1 VoIP Fundamentals

VoIP is the umbrella term given to a set of protocols that allow the routing of voice calls over the internet (IP network). The signaling can be enabled using either the Session Initiation Protocol (SIP)[14], proposed by the IETF or H.323[18] proposed by the ITU. Our solution is applicable to either of these protocols but in our discussions we assume SIP is used for signalling.

For two users to communicate with each other using SIP, they need to know each other's SIP URIs (Uniform Resource Identifier). SIP then uses an application overlay consisting of proxy servers and location services to locate these end points. A typical SIP call trapezoid is shown in Figure 1. When *Alice* identified by SIP URI *sip:alice@wonderland.com*, calls *Bob*, *sip:bob@music.org*, the call request message, known as the $INVITE$ message, is sent to the proxy server responsible for the *wonderland.com* domain, $P1$. $P1$ then determines how to route the call to the proxy responsible for *Bob*'s domain, *music.org*, $P2$. Once $P2$ receives the request it looks up user *Bob* and then routes it to the appropriate endpoint. On receipt of the $INVITE$ message *Bob*'s user agent (UA) starts to ring, shown by the 180 *Ringing* in Figure 1. When *Bob* picks up the phone the UA sends a 200 OK message. This initial message exchange forms the call setup transaction. When *Bob* or *Alice* hang up, the respective UA sends a BYE message and this initiates the call tear-down transaction. Call duration represents the time between the end of call setup (200 OK) to the start of call teardown (BYE) (see Figure 1). Call duration is the basic building block of the CallRank scheme proposed in this paper.

2.2 Social Networks

In CallRank, SNs are used to decide when to accept a call credential. SNs model associations that exist between a set of entities (typically humans). A distinctive feature of these networks is their tendency to cluster, measured by the clustering coefficient[21]. Mathematically an SN can be described as a graph $G = (V, E)$, where V, the set of vertices/nodes represent people and E, the set of edges represents some relationship/association between the people. G is referred to as the community. Consider a three vertex community consisting of nodes A, B and C. If a particular node, A, is connected to the other two nodes, B and C, then for the community to exhibit a high clustering coefficient B and C must also be connected. This tendency to form triangles from wedges is the nature of a highly clustered SN. In a voice communication system if there is a scenario where user A calls user B and user B calls user C, then due to the similar clustering nature in these systems, it is highly likely that user C will at some point call user A. This high likelihood coupled with call credentials is used in CallRank to provide a local mechanism to determine if a caller is a spammer or not.

2.3 Eigentrust

The Eigentrust algorithm[10] is used to determine the reputation of a set of peers based on their interactions. Each peer i decides a normalized local trust value for another peer j, based on the number of satisfactory and unsatisfactory transactions it has had with that peer. This value is represented as c_{ij}. It then uses a transitive notion of trust to aggregate these local trust values to a system wide reputation value for all peers. If $\overrightarrow{t}$ represents a vector containing the system wide reputation values, the Eigentrust algorithm can be used to determine this vector by solving $\overrightarrow{t} = (C^T)^n * \overrightarrow{e}$ for $n = large\ number\ of\ iterations$. C is the matrix containing the normalized local trust values $[c_{ij}]$, $\forall i, j$. $\overrightarrow{e}$ is a vector with unit 1-norm with its ith component $e_i = 1/m$,

where m is the total number of peers in the system. $\overrightarrow{t}$ converges to the left principal eigenvector of C, assuming that $\overrightarrow{e}$ has a component in the direction of the corresponding dominant eigenvector. In general, computing the nth power of the matrix (C^T) should be avoided and instead one should iteratively reassign the vector $\overrightarrow{t}$ as $\overrightarrow{t} = (C^T) * \overrightarrow{t}$ with the initial value of $\overrightarrow{t} = \overrightarrow{e}$. This method (known as the power method) for computing the eigenvector, that corresponds to the largest eigenvalue of the matrix, will converge with any initial vector for $\overrightarrow{t}$, as long as the initial vector has a component in the direction of the dominant eigenvector. In case there exists pre-trusted peers P we need to ensure that these end up with high reputations. Therefore to converge faster we can use $\overrightarrow{p}$, instead of $\overrightarrow{e}$ where $p_i = 1/|P|$ if $i \epsilon P$ and $p_i = 0$ otherwise. The system to solve, in the presence of pre-trusted peers, is $\overrightarrow{t} = (C^T)^n * \overrightarrow{p}$.

2.4 Related Work

Rosenberg and Jennings provide a good reference for the possible solutions that can be explored for VoIP spam which among others, include Content Filtering, Black and White lists, Turing tests and Computational puzzles [13]. We discuss this along with other related work based on the spam detection methodology used in the next few paragraphs.

In [13], [4] and [7], techniques used for combating email spam such as Blacklists, Statistical Blacklists, Greylists and Consent Based Systems are adapted for VoIP spam. The techniques mentioned above are subverted easily by the creation of new identities, a mechanism used in attacks such as the Sybil attack[5]. We show that CallRank, however, is more resistant to these kind of attacks in Section 3.3.1. In [17], [16] and [20], spam detection techniques based on anomalous characteristics of a spam call are described. However, it seems fairly simple for a spammer to subvert the detection of these characteristics and make a spam call. Strong authentication is probably the best counter measure against SPIT, however techniques based on DKIM[8], P-Asserted-Identity[9] and SAML[19] specified in [17] and [13] will only be successful when adopted by a large number of users.

Establishing absolute identity on the Internet is always going to be a hard problem. Reputation based techniques use the Internet's inherent democratic nature to provide a practical and effective alternatives. [4], [12] suggest the use of buddy lists and user ratings for buddies to create dynamic localized Whitelists. However, this restricts the scope of users that can call to strictly the user's SN linkage and it requires explicit user feedback in the form ratings. CallRank on the other hand, uses call duration, which is recorded automatically by the system without requiring explicit user action.

The transitive nature of social networks has been explored in [2], while the use of global reputation scheme to determine reputation values is present in [3]. These systems were designed to prevent email spam and use email relevant metrics to detect SN linkages and calculate reputation. They do not address the cryptographic security of their metrics. In CallRank we use call duration as the metric and cryptographically secure it using local signatures.Finally, the use of Turing tests or Client puzzles as suggested by [13], if used in a stand-alone fashion, leads to longer and more annoying call setup times. However, when combined with CallRank they can be used to reduce false positives of the CallRank system even further.

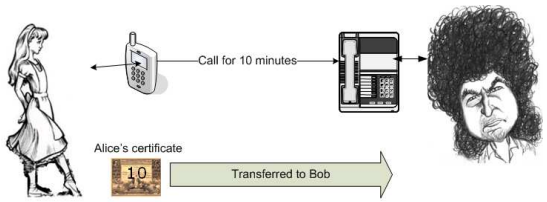


Figure 2: Reputation credential - Call Duration

3. CALLRANK

3.1 Voice Call Duration

Consider a call from Alice to Bob where the call duration is 10 minutes, as shown in Figure 2. This, to us, represents an implicit statement that Alice trusts Bob enough to speak to him for 10 minutes. On termination of the call Alice's user agent (UA) will then automatically hand a secure call credential to Bob stating that "Alice spoke to Bob for 10 minutes", represented by CC_{AB}. We ensure its security through cryptographic primitives discussed in Section 3.5. The next section discusses how we can combine this credential and SN theory to determine what call credentials can be trusted.

3.2 Using SNs to Accept a Call Credential

Consider, once again, the system as described in Figure 2, following which Bob talks to Charlie for 15 minutes. At this point Bob's UA hands a credential capturing this information to Charlie, CC_{BC}. At a later point in time assume Charlie tries to call Alice. If Charlie's UA presents CC_{BC} to Alice's UA at call setup time, then Alice can accept the call since she knows Bob (as she has recorded information of the call from her to Bob). In a general scenario the caller UA will present to the call recipient's UA a set of credentials when initiating a call. The call recipient's UA will see if any of the credentials can be used to establish a SN linkage and then decide either to accept or reject the call. Such a decision may consider several factors to determine how important or useful a particular credential is. For example, when Alice receives call credential CC_{BC} from Charlie the factors that will influence Alice's decision to accept the call are: (1) How strong is CC_{BC}?, and (2) How fresh is CC_{BC}?

The strength of the credential is dependent on the call duration value encapsulated within it. Thus, Bob speaking to Charlie for an hour will generate a stronger credential than Bob speaking to Charlie for a couple of minutes. Alice's UA also checks for the freshness of the credential. For this we assume that the UA's have access to approximately synchronized common clocks and we believe most phones will be time synchronized in a commercial VoIP deployment. Alice's UA can be configured with a policy stating that only call credentials with durations greater than a particular threshold, say T_{CD}, and timestamps within a certain time window shall be considered. We use the average call duration of the user as the value for T_{CD}, that is

$$T_{CD} = \frac{\sum Duration\,of\,Calls\,made\,by\,user}{Total\,number\,of\,calls\,made\,by\,user}.$$

A simpler scenario is when Alice speaks to Bob and Bob later wants to talk to Alice. Bob can use the credential that Alice provided to him. In this case there is a direct

relationship between caller and call recipient and the call can be accepted. In general, calls are accepted only if there exists, between caller and call recipient either a direct relationship, or a transitive single hop SN linkage. We restrict the linkage to a single hop because then callers can only use credentials directly presented to them. This restricts misuse of credentials and keeps the design simple.

In our evaluation of CallRank each UA maintains a record of all the people he called and a list of call credentials from users who made calls. When making a call the user can present these credentials as part of the initial $INVITE$ request until a suitable credential is found. The call recipient's decision to accept or reject a call is at the UA level and no other SIP component needs to get involved. This forms a scalable, load distributed solution as each UA is responsible for the calls it accepts or rejects. In most commercial phones, similar call history information is maintained under *Dialled Calls* and *Received Calls*. We can extend *Received Calls* to also store the call credentials.

We can consider a third factor that can influence Alice's decision of accepting a credential from Bob: (3) How reputed is Bob?. This reputation can either be Bob's reputation with respect to Alice or system wide reputation assigned to Bob. This is an interesting factor to incorporate into the system and we plan to explore it in future work.

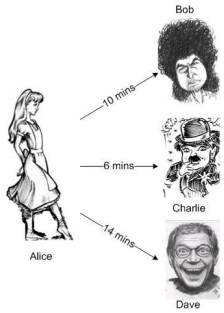
As users start accumulating credentials it might be hard to present all the call credentials in the first $INVITE$ request. In such a case further credentials can be presented in subsequent $INVITE$ messages. Deciding how many credentials are presented in each $INVITE$ message, defining an upper limit on the number of subsequent $INVITE$ messages and determining an optimal way to perform this initial handshake of credentials are all topics for future work. In our present implementation, the case where no credential is found is handled in Section 3.3.

3.2.1 Evaluating Threats to SN Based Scheme

If a spammer needs to defeat our SN based model and make a call to a particular user, he will have to penetrate the immediate SN of the user. Consider the scenario where a spammer wants to call Alice. He will either have to get a call credential directly from Alice or from someone to whom Alice makes calls. Since it is unlikely that a legitimate user, such as Alice, or her immediate SN will call the spammer and talk to him for sufficiently long periods of time, the spammer will find it hard to obtain such a credential.

Assume, however, that the spammer manages to convince a user Bob (who is part of Alice's immediate SN) to talk to him for a sufficient duration. This may happen when Bob inadvertently calls the spammer once. Since the spammer now has a credential from Bob, he is able to spam everyone who makes calls to Bob including Alice. However, the freshness constraint of the credential will only allow the spammer a short time window. If the spammer, on the other hand, is able to get Bob to call him regularly, then he will have a constant supply of fresh credentials. In such a case Alice on receiving a spam call can now decide that she will no longer accept calls which present call credentials from Bob. If the spammer needs to disseminate information to a large number of users, he will need to penetrate all their SNs (possibly disjoint) in a similar fashion.

The down side of our SN scheme is that there will be situations where even legitimate users will not be able to use call credentials because there exists no SN linkage between them or no SN linkage can be agreed upon after the initial handshake. The global reputation scheme discussed in the next section will be used to address this problem.



	A	B	C	D
A	0	.33	.2	.47
B	.5	0	.5	0
C	.08	.35	0	.57
D	0	0	0	0

Figure 3: Global Reputation Scheme

3.3 Global Reputation Using Eigentrust

Over the course of some period of time, assume that Alice talks to Bob, Charlie and Dave and the talk times are as shown in Figure 3. We can use call duration to represent the reputation value that Alice implicitly assigns to people she calls. Formally, the normalized local reputation value that a user i assigns to a user j is can be specified as

$$r_{ij} = \frac{Duration\ of\ all\ calls\ made\ by\ user\ i\ to\ j}{Duration\ of\ all\ calls\ made\ by\ user\ i\ to\ any\ user}. \quad (1)$$

This ensures that r_{ij} is between 0 and 1 and for any row i, $\sum_{\forall j} r_{ij} = 1$. The reputations assigned using this method are similar to the normalized local trust values in the Eigentrust system[10]. The advantage of normalizing is that reputation values are not arbitrarily high or low. This prevents users who form a malicious collective from assigning a high reputation value to other users in the collective and low values to legitimate users.

The first row in Figure 3 represents Alice's reputation values towards Bob, Charlie and Dave based on equation (1). Similarly the reputation values that Bob, Charlie and Dave assign to each other and Alice can be calculated and form the subsequent rows in a reputation matrix, R. For the system comprising only of Alice, Bob, Charlie and Dave, the reputation matrix is shown in Figure 3. If we need a system wide view of reputation values then we have to aggregate these local reputation values. We discussed in Section 2.3 that this is the leading left eigenvector, λ, of the matrix R. λ_i then represents the reputation of user i as perceived by the system as a whole. In calculating the leading eigenvector, we use the power method specified in [6]. The minor difference between our implementation and the power method presented in [6] is that in each iteration, we normalized the computed trust vector using its 1-norm (the method in [6] uses the 2-norm). Using the 1-norm ensures that the final computed eigenvector λ satisfies $\|\lambda\|_1 = 1$, i.e., $\sum_i \lambda_i = 1$. Note that each component λ_i of the vector λ satisfies $0 \leq \lambda_i \leq 1$ due to the characteristics of the matrix R. Thus the system as a whole has a total possible reputation of 1 and each individual has some fraction of this reputation. Using the 1-norm over the 2-norm should not affect the convergence rate in general since the normalization step in the power method can be done using any vector norm.

Proxies that provide billing services maintain call duration information for all users within their domain. The proxy is, therefore, the best place to maintain and update the reputation matrix. Periodically it can calculate and update the leading eigenvector of the matrix. In addition the proxy can also include users (from other domains) who have either made or received calls to or from this domain in its reputation matrix. In CallRank, when a proxy server receives a call request it consults the eigenvector calculated to obtain the reputation value for the caller and appends this information to the request. This reputation information can be sent securely to the call recipient as this only requires a secure path between the call recipient and his proxy (within a domain we can hope to use strong security mechanisms). The call recipient can then decide based on a threshold value if the calls will be accepted. Ensuring that only the call recipient's proxy appends a reputation value thwarts the attack that spammers can employ - creating their own proxy and providing high reputation values for their spam calls.

3.3.1 Evaluating Threats to Global Reputation Scheme

We discussed how it is hard for a spammer to penetrate a legitimate user's SN and thus compromise CallRank's effectiveness. It is equally hard for the spammer to obtain a high global reputation value. This is because the reputation value is based on call interactions with a number of users and takes into account the reputation of these users. If a spammer needs to have a high reputation value, he will need a significant number of moderately reputed users to call him and speak for significantly long durations. This is unlikely to occur. A legitimate user, on the other hand, will have a high reputation value due to call interactions with other legitimate users (a feedback loop). This implies that CallRank can counter attacks where new identities are created each time the old ones are flagged as malicious (as done in a Blacklist). This is because the new identities are only accepted as legitimate when they can provide SN credentials or garner significant reputation values.

In the case of Sybil attacks[5], where a small number of entities counterfeit multiple identities to compromise the system we note that CallRank is fairly resistant. If the entities themselves have weak SN linkages or low reputation values, then creating new identities will not help at all. If the entities, however, have strong SN linkages or high reputation values, then the identities they create can be made reputable or provided with these linkages. In such a scenario the system will soon realize that credentials coming from this set of entities lead to spam calls following which the reputation of the entities and the identities they have created begin to drop, therefore affecting their ability to continue making spam calls.

3.4 The Introduction Problem

When a new legitimate user joins a VoIP system he has no SN linkages in that system and a low reputation value. This will change if other users call him, thereby increasing his reputation value and providing him with call credentials. However, other users are unaware of his entry into the VoIP system. In order to notify other users he will need to make the first call. In CallRank, however, all calls he makes will be flagged as spam calls, which amounts to a false positive. We can fix this by combining CallRank with other schemes proposed for VoIP spam such as an audio Turing test or a computational puzzle. When a user is flagged as a spammer he will then be subject to the Turing test or a computational puzzle or even a personalized question from the call recipient (what is my high school nickname). The call is accepted if the caller is able to successfully answer any of these tests. In our simulation we have not included such a Turing test and this forms part of our future work.

3.5 Call Credentials

The call credential needs to have accurate and secure information about the call durations. A call credential CC consists of A, the identity of the caller, B, the identity of the call recipient, t, the call duration and TS, the time stamp of the call along with a digital signature of the same information. We assume that each user has a public/private key pair which is used to generate the digital signature. If not already available, this pair can be generated by the UA on first use. Associating a public key with a particular user is done with key rings in the manner proposed in [11], thus avoiding the use of an infrastructure such as the PKI.

The accuracy of the information within the credential can be verified by the proxy which also records call duration information. We assume the proxy has an accurate value of call duration as it provides billing services. Therefore, the proxy does not need call credentials for calculating reputation values. In fact, if the proxy is used to determine the SN linkage for a call, we do not need call credentials. However, we believe moving the SN linkage detection to the proxy makes the system unscalable.

To understand the call credential better, we consider what it means from a human perspective. This credential is a record of the user's past observed behavior in the system or his call history. If the user is an active member of a particular VoIP community, making and receiving calls, he will accumulate the community relevant credentials through his interactions, making it easier to identify him accurately within the community. If for some reason there is a sufficiently long break from the community then when he reenters, he will once again have to reestablish himself. This is how it works in the real world. Since credential collection can be done by the user's phone without any input from the user there is minimal impact on usability. Using call duration as a building block has the following advantages. It is (i) implicit, (ii) quantifiable (iii) easily verifiable, and (iv) easily understood.

3.6 Discussion of CallRank Algorithm

To summarize, the CallRank algorithm works as follows. On receipt of a call setup message, the UA first checks to see if any call credentials presented by the caller belong to users to whom the UA has made calls. If such a credential is found and it satisfies the policy duration and freshness constraints, the call is accepted. If no credential satisfies the constraints then the algorithm checks the reputation value of the caller. If this satisfies a particular acceptable reputation threshold, then the call is accepted else it is rejected. Rather than rejecting the call the caller can be made to go through a Turing test or a call recipient specific computational puzzle. However we propose to explore this in future work.

Integrating the CallRank algorithm into SIP will require that the initial $INVITE$ message also carry call credentials as well as proxy appended reputation scores. Clients that do not implement the CallRank algorithm can simply choose to

ignore this information.

CallRank does have some limitations. The first limitation is that legitimate users, who make a large number of outgoing calls but receive very few incoming ones, would not be able to collect call credentials. Typical examples are emergency services and banks. Since these systems are part of critical infrastructure, they can be seeded with high global reputation values. The second concern is one of privacy as the collection of call credentials provides a user with the call history information of their immediate SN. We plan to address this limitation as part of future work.

4. CALLRANK EVALUATION

An evaluation of CallRank in the real world would require call logs from a VoIP system along with actual cases of VoIP spam. Call logs are hard to come by due to privacy concerns and VoIP spam is still not widespread enough. Instead, we simulate CallRank with a synthetic call workload to evaluate its effectiveness, ensuring that the simulations model real world call characteristics as closely as possible. In particular, we measure how CallRank can be used to distinguish between legitimate callers and spammers and the results are discussed in Sections 4.1, 4.2 and 4.3. We study the acceptance of a legitimate caller into the CallRank system in Section 4.4.

Our initial experimental setup consists of DNS, proxy and statistics servers and user agents (phones). Initially, only the DNS and the statistics server are running. Each proxy server registers with the DNS server, and the user agents (UAs) register with the proxy. UAs either behave as reputed users (seeded with high reputation values), legitimate users (users who make legitimate calls but are not seeded with high reputation values), or as spammers. A legitimate or a reputed UA makes calls to other phones with inter call and call duration values that are Poisson distributed. The choice of call recipient is Zipfian distributed. Spamming UAs, however, make calls to as many other UAs as possible.

Call setup goes through the proxies which consult the DNS server and then route the call to the proxy in the call recipient's domain, which in turn forwards the call to the call recipient. During the learning period (which can be set), a call recipient will accept all calls. After the learning period, a call is accepted or rejected based on the working of CallRank. All call interactions are recorded at the statistics server which tracks the number of accepted and rejected calls for both legitimate users and spammers. Our initial setup consists of three domains each served by a proxy server and 200 users initially registered in each domain. 1% of the 600 users are reputed. The number of spammers and regular users is varied based on the experiment. We use a simulated call workload model. To simulate call processing for a sufficient period of time, 100 seconds of machine time models 1 day of simulated time.

4.1 Effect of Spammers

The first set of experiments determines the effect of spammers on CallRank. Three runs are conducted where the spammers present are varied from 1%, 10% and 20% and the fraction of spam calls accepted for each case is measured. The results are as shown in Figure 4 which plots the fraction of spam calls accepted with time. When legitimate users join the system they have a learning period during which they accept ALL calls. This period is essential for the user

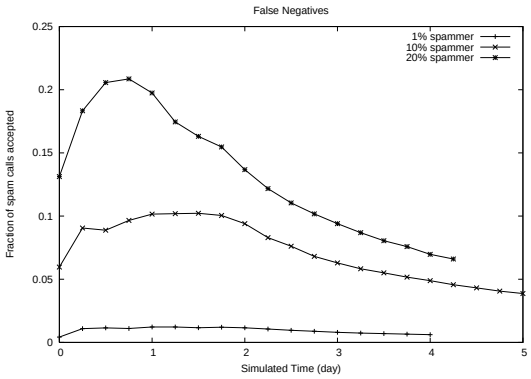


Figure 4: Effect of spammers

to gather credentials and build reputation. However, during this period, they are vulnerable to spam calls. The spammer thus needs to detect a new user within the learning period time window and then send all the spam they can generate. In our simulation the learning period for all UAs is fixed at 1 day. All 3 lines initially show increase as spammers learn about more and more legitimate users and are able to send spam to them successfully. This increase lasts roughly for the learning period and then starts decreasing rapidly. This is because legitimate users, using the CallRank scheme, are able to differentiate between spammers and legitimate users soon after their learning periods. For all 3 lines there are no new spam calls accepted after 4.5 days.

As the percentage of spammers increases from 1% to 10% and then to 20%, the probability of some spammer discovering a legitimate user increases and the ability to send larger amounts of spam increases as well. This is seen in Figure 4 as each of the curves shows higher false negative rates of 1%, 10% and 22% respectively. Thus, the false positive rate increases linearly with the number of spammers. However, these numbers are contingent on the fact that legitimate users are discovered by spammers within their short learning period time window. If the legitimate user is undiscovered then the rates will drop down even further. In fact, once a legitimate user crosses his learning period, he is able to identify spammers (old and new) with ease.

4.2 Addition of New Spammers

We start with an initial population of 600 UAs, 1% of which are reputed UAs, 10% spammer UAs and the rest are legitimate UAs. We wait until the system stabilizes, that is no new spam calls are accepted or no new legitimate calls are rejected. From Figure 5 we see this occurs after 2 days and the number of accepted spam calls has saturated around 1000 calls. We then add spammers, 1%, 10% and then 20% of the current UA population. As seen, the addition of these spammers does not increase the number of accepted spam calls illustrating that CallRank's mechanisms ensure that new spammers do not affect existing legitimate users. The reason behind this is that a new spammer, when introduced, does not have any SN linkage or reputation. Therefore, existing legitimate users will not accept any calls originating from them. Thereafter a spammer, due to his behavior, will not improve either his SN or reputation implying that at no stage will a legitimate user accept a call from him. This is

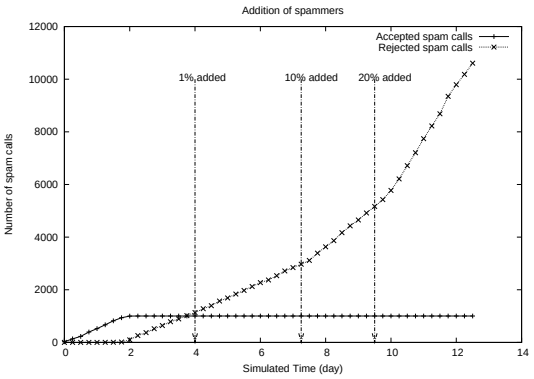


Figure 5: Impact of new spammers

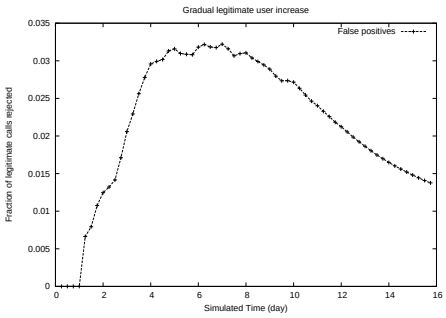


Figure 6: Adding legitimate users

a big advantage of the CallRank scheme where older users by virtue of their good call history become more adept at rejecting spam calls.

The addition of new spammers generates more spam and we can see this in the increase in the number of rejected spam calls in Figure 5. At each stage of the introduction (marked by arrows) we can see an increase in the slope of rejected spam calls thus corroborating CallRank's effectiveness.

4.3 False Positives

Although not shown in the previous experiments, the false positive rates are also extremely low. For example in the simulation run that involved 600 users, 1% of whom are reputed and 10% are spammers, there were only 3 calls that were wrongly rejected to give a false positive rate of .02%. This low rate is because all users are introduced at the same time and their learning periods coincide. Therefore, all users were simultaneously aware of the rest of the users by the end of this period. However, in a realistic scenario users join a system over a period of time. To simulate this we created 600 users, 200 in each domain, over a period of 10 days. Within a domain, new users are added at intervals of 3 hours (simulated time). The false positive rate of such a system is shown in Figure 6. The rate initially increases to a high of 3% and then reduces gradually. This is because when a user joins the system, he has no SN linkage and no reputation which by CallRank's perspective is the characteristics of a spammer. Therefore, most of his calls will be rejected. However, if the user behaves legitimately, this rate drops soon enough showing that CallRank is able to determine that the user is legitimate. False positives do not have the same connotation as in the email world, where it implies a permanent loss of information. In the VoIP world, since interactions are synchronous, a user whose call is rejected can be asked to take an audio Turing test. This will only result in occasional longer call setup times, typically occurring when the user initially joins the system.

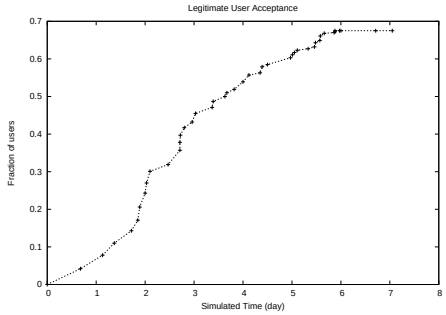


Figure 7: Legitimate user acceptance

4.4 User Acceptance

We studied the acceptance of a legitimate user into a system containing 1000 existing users. This is shown in Figure 7. As we can see a legitimate user is accepted by half the total user base in 3.5 days. However, this factor can be used by a spammer to alternate between being a legitimate user and a spammer, thus, providing him with the ability to spam a large set of users. This threat is not as large as it seems because behaving as a legitimate user entails getting people with significant SN linkages or moderate reputation values to talk to the spammer on a regular basis. When a legitimate user joins a voice communication system, almost immediately there are other legitimate users who talk to him, thus creating his SN and establishing his reputation. This happens naturally for most people who have an established life outside the VoIP system. Their SN linkages or their reputations are just extensions of their real world persona. On the other hand a spammer has no existence outside the VoIP system, and so, legitimate users will never call him when he gets introduced into the system.

We also see that the graph in Figure 7 saturates at 70% (say set S) of the user base. That implies that anytime this user calls any of the users belonging to the remaining 30% (S'), he will be treated as a spammer. This is because there exists no SN linkage between the user and members of S' and the user's reputation value is significantly lower than users in S'. From our logs we see that S' consist of either the initially pre-reputed users or users that have been in the system in a legitimate fashion long enough to have become extremely reputed. This behavior is beneficial as it implies spamming highly reputed users is going to be significantly hard.

5. CONCLUSIONS AND FUTURE WORK

In this paper we proposed CallRank, a system that uses call duration to determine if a caller is a spammer. Our simulation explored the effectiveness of CallRank and showed that it adapts over time, allowing users with legitimate call history to make calls easily while defeating spammers. In

addition, our system is able to accept new legitimate users relatively easily while ensuring that new spammers are not able to affect existing users. In the future we plan to explore mechanisms that maintain privacy of users by creating aggregate call credentials for a group of users.

6. ACKNOWLEDGMENTS

We would like to thank Deepak Manohar, Chris Rouland, Tom Cross, Dr. Nathaniel Borenstein and the GTISC VoIP Security team for valuable discussions. We would also like to thank the anonymous reviewers of CEAS for the valuable review comments provided.

7. REFERENCES

[1] J. A. Barnes. Graph theory and social networks: A technical comment on connectedness and connectivity. *Sociology*, 3(2), 1969.

[2] P. Boykin and V. Roychowdhury. Leveraging social networks to fight spam. *IEEE Computer*, 38(4):61–68, 2005.

[3] P.-A. Chirita, J. Diederich, and W. Nejdl. Mailrank: using ranking for spam detection. In O. Herzog, H.-J. Schek, N. Fuhr, A. Chowdhury, and W. Teiken, editors, *CIKM*, pages 373–380. ACM, 2005.

[4] R. Dantu and P. Kolan. Detecting Spam in VoIP Networks. In *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet Workshop*, pages 31–37, Cambridge, MA, July 2005.

[5] J. R. Douceur. The sybil attack. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 251–260, London, UK, 2002. Springer-Verlag.

[6] G. H. Golub and C. F. Van Loan. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)*. The Johns Hopkins University Press, October 1996.

[7] M. Hansen, M. Hansen, J. Mller, T. Rohwer, C. Tolkmit, and H. Waack. Developing a legally compliant reachability management system as a countermeasure against spit. In *Proceedings of Third Annual VoIP Security Workshop*, Berlin, Germany, Jun 2006.

[8] T. Hansen, D. Crocker, and P. Hallam-Baker. Domainkeys identified mail (dkim) message signing service overview, Mar 2007. IETF-DRAFT draft-ietf-dkim-overview-04.txt.

[9] C. Jennings, J. Peterson, and M. Watson. Private extensions to the session initiation protocol (sip) for asserted identity within trusted networks, 2002.

[10] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proc. 12th International World Wide Web Conference*, Budapest, Hungary, May 2003.

[11] L. Kong, V. A. Balasubramaniyan, and M. Ahamad. A lightweight scheme for securely and reliably locating sip users. In *1st IEEE Workshop on VoIP Management and Security*, Vancouver, Canada, Apr 2006.

[12] Y. Rebahi and D. Sisalem. Sip service providers and the spam problem. In *2nd Workshop on Securing Voice over IP*, Washington DC, USA, Jun 2005.

[13] J. Rosenberg and C. Jennings. The session initiation protocol (sip) and spam, Feb 2007. IETF-DRAFT draft-ietf-sipping-spam-04.txt.

[14] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. Sip: Session initiation protocol, Jun 2002. RFC 3261.

[15] R. Shaw. Four reasons why vonage ipos email and phone pitch is the wrong strategy. *ZDNet*, 2006.

[16] D. Shin and C. Shim. Voice spam control with gray leveling. In *2nd Workshop on Securing Voice over IP*, Washington DC, USA, Jun 2005.

[17] B. Sterman. A security model for spit prevention. In *2nd Workshop on Securing Voice over IP*, Washington DC, USA, Jun 2005.

[18] G. A. Thom. H.323: the multimedia communications standard for local area networks. *Communications Magazine, IEEE*, 34(12):52–56, 1996.

[19] H. Tschofenig, J. Peterson, J. Polk, D. Sicker, and M. Tegnander. Using saml for sip, Jul 2005. IETF-DRAFT draft-tschofenig-sip-saml-04.txt.

[20] VoDaSec. Spitting over the internet. http://www.vodasec.com/.

[21] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, April 1998.