Collaborative Reputation-based Voice Spam Filtering

Ruishan Zhang, Andrei Gurtov
Helsinki Institute for Information Technology
Helsinki University of Technology and University of Helsinki
zhangruishan@gmail.com, gurtov@hiit.fi

***Abstract*—We propose a collaborative reputation-based voice spam filtering framework. Our approach uses the cumulative online duration of a VoIP user to derive his reputation value. And we leverage user feedback to mark unsolicited calls. For each unwanted call, our voice spam filter charges the caller a reputation point, and transfers this reputation point to the callee. To avoid VoIP users to manually label nuisance calls, our voice spam filter automatically marks all VoIP calls with short call durations as unsolicited. The preliminary simulation results show that our approach is effective to counter voice spam.**

I. Introduction

Recent years have witnessed the phenomenal growth of VoIP. It has been predicted that VoIP will surge Public Switched Telephony Network (PSTN) and capture the future telecommunication market [1]. Due to the convenience and low cost of placing VoIP calls, VoIP has become an increasingly attractive channel for voice spam whereby spammers launch unsolicited calls to massive recipients.

Though content analysis has been widely used to filter email spam, such an approach is ineffective to thwart voice spam. The reason is that in the case of VoIP, the recipient is somewhat disturbed even if an unsolicited call just makes the phone ring. To prevent voice spam, we need to utilize some identity-based solution.

In this paper, we aim to propose an effective voice spam filtering framework to identify and combat VoIP spam. Our voice spam filtering approach should implement three goals.

Prevent unwanted bulk calls. Our goal is NOT to stop all nuisance calls. Instead, we focus on blocking unsolicited calls in large quantities.

Minimize the possibility of blocking normal calls. In some scenarios, VoIP users need to receive calls from people without prior contact, e.g., a travel agency may accept many calls from strangers.

Limit the inconvenience caused to VoIP users. VoIP users are not required to perform much manual intervention to stop spam, e.g., pass a voice CAPTCHA or mark a call as unwanted.

Most Internet services, e.g., email, blogging and instant messaging services, are free, and use a visual CAPTCHA to distinguish a human from a computer to prevent malicious bots from opening massive accounts. However, it was reported that CAPTCHA implementations of Windows Live Hotmail, Yahoo and Gmail, have been cracked [2], and a spammer can achieve a success rate of 20% to 60%, using a bot. This implies that spammers are capable of acquiring numerous accounts.

In this paper, we propose a collaborative reputation-based spam defense framework for VoIP systems. Key points are summarized below.

First, we use the cumulative online duration of a VoIP user to derive his reputation value. A VoIP user can utilize his own reputation, or leverage the reputation of his buddy to make legitimate calls. And we leverage user feedback to mark unsolicited calls and charge the caller a reputation point for each unwanted call. The reputation value determines the amount of spam calls that an individual spammer can launch. Therefore, a single spammer can just launch a few spam calls before exhausting his reputation.

Second, our voice spam filter automatically marks all VoIP calls with short call durations as unsolicited, thus freeing VoIP users from manually labeling unwanted calls. Our voice spam filter is easy to be deployed, requiring no modifications to the existing VoIP clients or VoIP servers.

Finally, we extend the range of the whitelist from a user's one-hop buddies to his two-hop buddies. Consequently, all calls from the recipient's two-hop friends will be accepted.

Using data of 1659 users from a leading social networking website, LiveJournal, we performed several simulation tests. The simulation results demonstrate that our approach is effective to stop spam. Even if the number of user accounts that spammers could control accounts for 10% of all the benign user accounts, each user just receives about 0.5 spam calls weekly. Additionally, only 0.3% normal calls are blocked.

The rest of this paper is organized as follows. Section II surveys related work. Section III describes our methodology. Section IV presents our simulation design and results. Finally, Section V concludes the paper and introduces our future work.

II. Related Work

In this Section, we briefly overview some relevant spam detection and filtering techniques.

Blacklisting and whitelisting are two basic techniques to fight spam. All calls from a user in the blacklist are rejected, whereas calls initiated by a whitelisted friend are accepted. In Skype, a user can customize his privacy setting: either accept calls from anyone or user accounts in his buddy list. Koskela et al [3] developed a peer to peer VoIP prototype with spam prevention, based on Host Identity Protocol (HIP) [4]. When

whitelisting is enabled, important calls from people without prior contact would be rejected.

Dantu et al [5] proposed a multi-stage voice spam detection filter: VSD. VSD employs Bayesian inference technique to compute the spam probability of an incoming call. During the learning period, human intervention is required to mark unsolicited calls. In this paper, our approach does not require VoIP recipients to manually label unwanted calls.

Balasubramaniyan et al [6] proposed CallRank, a voice spam detection scheme, to filter voice spam. CallRank uses call duration to build social network linkages and global reputations for users. A long call duration can serve as a *call credential* from the caller *Alice* to the call recipient *Bob*. Then *Bob* could leverage this call credential to call a user *Charlie* who has called *Alice* recently. CallRank assumes that each VoIP user has a public/private key pair and can generate digital signatures. However, VoIP clients in most of the existing VoIP services do not own a private key [7]. So upgrading existing VoIP clients is required to use CallRank.

Shin et al [8] proposed to use graylisting to combat voice spam. The grey level of a caller determines whether the call is accepted. If a caller launch numerous calls in a certain time span, his gray level will increase. Once the gray level exceeds a threshold, all later calls from the caller within a given period will be blocked. After the caller stops making calls, the grey level will decrease and eventually become below the threshold. The problem of the graylisting mechanism is that whether it is still effective when spammers exploit numerous VoIP accounts to launch spam.

Collaborative spam detection approaches have been applied to stop email spam in Razor [9] and Distributed Checksum Clearinghouse (DCC) [10]. In Razor, trustworthy users can mark a received email as spam, and send the signature of the spam email to the Razor server, so that other users can share the human effort. DCC employs a slightly variant idea: if a message has been seen many times elsewhere on the Internet before reaches you, then it is likely that this email is spam. Unlike Razor and DCC, which rely on content analysis, our collaborative spam detection framework is based on identity.

Some recent work [11] has shown that in some deployed VoIP systems remote attackers can circumvent "Do Not Disturb" and place annoying calls to VoIP phones directly rather than via VoIP servers. In this paper, we just focus on voice spam sent through VoIP servers.

III. Collaborative Reputation-based Spam Defense

In the following subsections, we first discuss our assumptions, and elaborate several techniques to fight spam. Then, we describe our collaborative voice spam filter architecture and how to apply our approach to protect VoIP calls.

A. Assumptions

In this paper, we assume VoIP user identities are not spoofable. This means that a spammer is unable to forge a benign user's identity. Moreover, we assume spammers might be able to achieve a significant proportion of user accounts, e.g., 20%. However, we exclude the extreme case that spammers can obtain unlimited user accounts. Although some visual CAPTCHA schemes have been partially broken, and attackers can exploit bots to create numerous new email accounts. We hold it's impossible to create unlimited user accounts. The reasons are as follows. Firstly, new and secure CAPTCHA schemes will be invented and applied to avoid attacks from automatic bots. Additionally, even if the CAPTCHA mechanism is subverted by malicious attackers, we can employ strong entity check, e.g., prove the ownership of a bank account, to stop attackers from creating unlimited user accounts. This will incur a little convenience to VoIP users. However, given that most Internet users has been accustomed to this when making online payments, this method should be acceptable to VoIP users when they realize various advantages of free VoIP services. Currently, a lot of users of VoIP services, e.g., Skype, Vonage and AT&T, are even willing to purchase some phone credit to call regular home phones or mobile phones.

B. Collaborative User Feedback

Since the conversion rate of spam is very low, e.g., the conversion rate of botnet-generated email spam is 1 out of 200,000, spammers must launch massive spam messages or calls to reach potential clients. If each single user account just can transmit a few spam calls, even if spammers can own a relatively high proportion of user accounts, they still can hardly launch bulk unsolicited calls. So the key point is to limit the amount of spam that one single user account can launch. In this paper, we propose to use collaborative user feedback to achieve this goal. Once an unwanted call is labeled, one reputation point is transferred from the caller to the callee. In our framework, VoIP recipients determine whether incoming calls are unwanted. However, no extra operation from VoIP clients is needed to label unsolicited calls. Instead, we use the duration of an incoming call to represent the recipient's feedback. If the call duration is less than 20 seconds, then the voice spam filter on the VoIP server labels the call as unwanted.

Compared to blacklisting whereby the recipient puts the caller of an annoying call in the local blacklist, this technique utilizes user feedback in a collaborative way. All users can benefit from each other's feedback to the voice spam filter. After nuisance calls from the same user account have been reported a few times, subsequent calls from this account will be filtered, thus limiting the number of spam calls from a single user account.

C. Reputation

To keep reachable, a VoIP client needs to continually send the VoIP server some keep-alive messages, e.g., registration request. For example, a Vonage phone sends a REGISTER request to its REGISTRAR server every 18 seconds. We use Cumulative Online Duration (COD) in hours to calculate the reputation value. The reasons are as follows. First, COD is reliable and trustworthy. In a VoIP network, the VoIP server can keep track of the online status of a VoIP user, and

calculate his COD. Second, this approach incurs little burden to legitimate users. A normal user's COD will gradually grow over time. Finally, a spammer has to spend some network bandwidth and computer resources to obtain a high COD.

The reputation value determines the number of spam calls that a user account can launch. According to the caller's reputation value, the recipient determines whether to accept the incoming call. Suppose the reputation value of the caller is x. If $x \geq 1$ holds, then the recipient accepts the call. Once the recipient marks the incoming call as spam, the reputation value of the caller will decrease by one. Therefore, the reputation value determines the number of spam calls that a user account can launch.

D. Extension of the Whitelist

Whitelisting is one basic mechanism to defend against voice spam. It has some advantages. First, the recipient would not miss important calls from his two-hop friends, even if his friends lack enough reputation. Additionally, the recipient avoids labeling calls from his friends as spam accidently. Whitelisting is an important complementary technique to the collaborative user feedback approach.

Conventionally, a user manually adds some user accounts to his buddy list, and this buddy list serves as the whitelist. That is, whitelist is limited to the one-hop friends, and the user know all whitelisted user accounts. Although whitelisting is very simple, it's a very effective means. A study on email spam defense [12] showed that over 55% of received emails are from the recipients' one-hop buddies.

Considering friends of a friend are likely to be trustworthy, we extend the coverage of the whitelist from one-hop friends to two-hop friends of the user. Garriss et al estimated [12] that two-hop whitelisting can accept approximately 10% emails more than one-hop whitelisting.

E. Voice Spam Filtering Architecture

In this subsection, we introduce how to integrate our collaborative user feedback framework into a centralized VoIP system. Figure 1 depicts the architecture of our collaborative voice spam filtering framework. There exist three types of VoIP entities: the VoIP server, the voice spam filter and VoIP phones. The voice spam filter is responsible for calculating the cumulative online durations of VoIP clients and filtering spam calls. Here the voice spam filter acts like a firewall. All incoming calls need to be inspected by the voice spam filter first. Then the voice spam filter will deny unsolicited calls and let normal ones pass through. Numerous VoIP phones are located behind NAT routers, and connects to the Internet via NAT. VoIP phones need to regularly update their location information to the VoIP server to keep reachable.

F. Message Flow

In theory, our collaborative voice spam detection framework could be used in any kind of VoIP networks. Since the Session Initiation Protocol (SIP) [13] is the dominant signaling protocol for VoIP systems, we describe how to incorporate our framework into a SIP-based VoIP system.

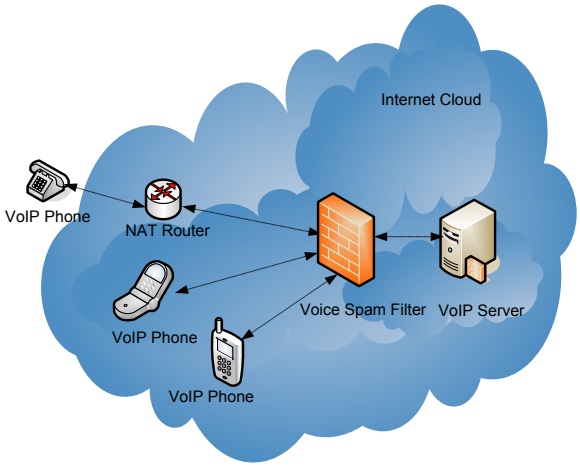

Fig. 1. Voice Spam Detection Architecture

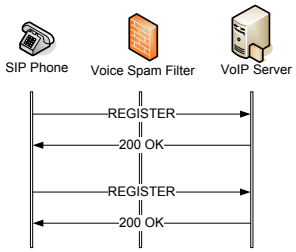

Fig. 2. Transmitting SIP REGISTER Messages Through the Voice Spam Filter

Figure 2 depicts the message flow of transmitting SIP REGISTER messages through the voice spam filter. By observing SIP REGISTER messages from a VoIP client, the voice spam filter could calculate the cumulative online duration of the VoIP client.

Figure 3 shows the message flow of a SIP call passing through the voice spam filter. When receiving a call request to the VoIP server, the voice spam filter first intercepts the call, and checks whether the caller is qualified to place a call. If the caller is NOT qualified, the call is rejected as spam. Otherwise, the call is allowed in and delivered to the VoIP server. The check process consists of three steps.

(1) Check whether the caller *Alice* is on the whitelist of the recipient *Bob*. If the caller is whitelisted, the call is allowed in. Otherwise, go to step (2).

(2) Check whether *Alice* could employ her own reputation to make the call. Suppose the reputation value of *Alice* is x. If $x \geq 1$ holds, the call is accepted. Otherwise, the call is blocked.

The voice spam filter can obtain the call duration by observing the time interval between the `BYE` message and the `ACK` message.

After the call is terminated, if the call duration is shorter than 20 seconds, the voice spam filter labels it as unsolicited. Here we suppose that 20 seconds is adequate for a person to determine whether an incoming call is annoying. Then the voice spam filter decreases *Alice*'s reputation value by

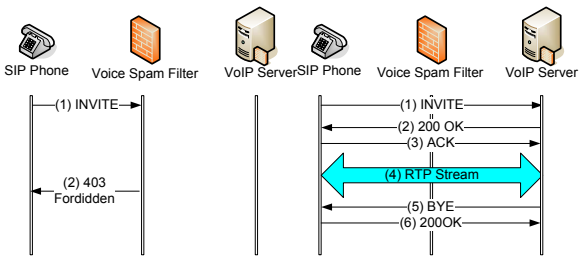

Fig. 3. Message Flow of a SIP Call Passing through the Voice Spam Filter

one. Meanwhile, the voice spam filter increases the callee's reputation value by one. That is, the voice spam filter transfers one reputation point from the caller to the callee.

Note that VoIP users are not required to manually label unsolicited calls. Instead, we let the voice spam filter tag unwanted calls. The benefits of such a choice includes:

(1) No manual intervention is needed to tag nuisance calls.

(2) The existing SIP protocol does not specify how to send a spam report. If VoIP clients are required to report spam, we have to modify SIP clients, and make them NOT SIP compliant.

(3) Most importantly, since no modifications to VoIP clients or servers are required, we can readily deploy our voice spam filter as a pluggable security module on the VoIP server side.

The disadvantage is that this choice might cause a high mislabeling rate. So in our simulation, we use a high mislabeling rate, 20%. Considering all calls from a two-hop buddy are whitelisted, and most calls from strangers should have a long call duration, 20% is reasonable.

IV. Simulation

To evaluate the effectiveness of our framework, we performed some simulation tests. The better way to simulate our approach is to collect some data from instant messaging systems, e.g., cumulative online duration, one-hop and two-hop buddies of Skype or MSN Messenger, and use such data to conduct simualtion. Unfortunately, instant messaging systems do not publicize this kind of data. However, we found that we can obtain similar data by crawling LiveJournal (LJ) [14], a popular social networking site with over 18 million accounts.

Our simulation comprises of two parts. One part is about the extended whitelist, measuring the buddy statistics of LJ users. The other part is to simulate the interactions between callers and callees, and evaluate the effectiveness of our voice spam defense approach. Specifically, our simulator simulates that even in the presence of a high mislabeling rate, our framework has little effect on normal users, and they could still place many calls to strangers over two-hop away.

A. The Extended Whitelist

A LJ user has two lists of friends: **Friends** and **Friend of**. When a LJ user *Alice* adds *Bob* as his friend, *Bob* appears at *Alice*'s **Friends** list and *Alice* appears at *Bob*'s **Friend of** list. In this paper, we refer to friends in **Friend of** as a user's forward one-hop buddies, and those in **Friends** as reverse one-hop buddies. A user's forward one-hop buddies trust this user, whereas a user trusts his reverse one-hop buddies.

TABLE I
The Average Numbers of Buddies

FOHB	ROHB	FTHB	RTHB
97.91	73.84	8987.37	7275.27

We randomly selected 957 LJ accounts, and collected their forward and reverse buddies. Table I shows that the average numbers of forward one-hop buddies (FOHB), reverse one-hop buddies (ROHB), forward two-hop buddies (FTHB) and reverse two-hop buddies (RTHB) are 66.73, 62.72, 537.50 and 496.38 respectively. This indicates that, on average, the number of two-hop buddies is approximately two orders of magnitude more larger than that of one-hop buddies, and thus the coverage of the whitelist is greatly extended. Additionally, in our framework, once the call from an unknown user is identified as normal, the caller is automatically added to the whitelist of the recipient. This will further expand the whitelist.

B. Collaborative User Feedback

The aim is to simulate that even with a relatively high mislabeling rate, normal users are still able to make a lot of calls to strangers over two-hop away.

Each LJ user has a user No. n, which represents he is the nth created user. Additionally, we can compute a user's LJ age according to the creating time in his user profile.

1) Simulation Design: A user's application age, e.g., Skype age, could be served as the indicator of his cumulative online duration. A survey by China Internet Network Information Center (CNNIC) [15] showed that weekly online durations of 99.9% of Chinese Internet users exceed one hour. This indicates even if we lack a direct approach to obtaining cumulative online durations of users, we can derive some useful information of online durations from their application ages. If VoIP users always keep online when connected to the Internet, cumulative online durations of VoIP users will grow. For example, if VoIP user *Alice*'s application age is over 1 month, it's most likely that her cumulative online duration is above 4 hours. In our simulation, we use a user's LJ age to infer his cumulative online duration.

Our simulation proceeds as follows.

(1) Determine the simulated user set. First we set the starting and ending point of our simulation as 2820 and 2659 days respectively, before the date when we ran the simulation tests, January 13th, 2009. At the starting point, the number of LJ users reached about 100,000. Next we randomly selected 100,000 LJ users, and obtained their user profiles. Then from this user set, we chose those early users who were created before the ending point when the number of LJ users reached about 360,000. The chosen users form our simulated user set. The simulated user set contains 1659 users.

(2) Run our simulator. From the starting point, we ran our simulation tests for 24 weeks (simulated time). Note that new

TABLE II
THREE TYPES OF SIMULATED USERS

Type	Description
A	Created more than 10 weeks before the starting point
B	Created less than 10 weeks before the starting point
C	Created after the starting point

TABLE III
SIMULATION PARAMETERS

Parameter	Value
Initial reputation result for a type *C* user	7
Mislabeling rate	20%
Average number of calls of per user weekly	10

users join the system over time.

As shown in Table II, our simulated users consist of three types. Both type *A* and *B* users were created before the starting point, whereas type *C* users were created after the starting point.

A newly joined type *C* user is assigned an initial reputation value: 7. At the starting point, the reputation value of a type *A* user is set to 50, whereas that of a type *B* user is set to the product of 5 and the difference between his age and the starting point in weeks. The reputation value of each user weekly increases by 5.

Suppose the number of active user accounts is t. Then $10 \times t$ calls are made weekly, which means that on average each user makes 10 calls daily. That is, a user can place over 3600 calls to strangers two-hop away per year. Since all newly encountered benign user accounts will be put in the extended whitelist, 3600 is sufficient for ordinary users. For each call, both the caller and the callee are randomly determined. 20% of all the placed calls are mislabeled as spam. Table III lists simulation parameters.

2) Simulation Results: The simulation results are shown in Table IV. 99.7% of all placed calls are accepted. And 0.3% calls are blocked.

Suppose the number of user accounts that spammers can control accounts for 10% of all the benign user accounts, which means that the ratio of the number of malicious accounts and that of benign accounts is 1:10. The amount of spam calls that spammers could launch weekly is $10\% \times$ *number of all benign user accounts* $\times\ 5$. On average, each benign user receives 0.5 spam calls weekly. And the ratio of spam calls to normal ones is 0.7%, which means 1 of 140 received calls is spam.

In summary, our simulation results show that even spammers can control a high proportion of user accounts, our collaborative user feedback framework still can provide good defense against voice spam: each user receives about 0.5 spam calls weekly. Additionally, only 0.3% normal calls are blocked.

TABLE IV
SIMULATION RESULTS

Metric	Value
Percentage of accepted calls	99.7%
Percentage of blocked calls	0.3%
Spam calls received by each user weekly	0.5
Ratio of spam calls to normal ones	0.7%

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a collaborative reputation-based spam detection framework for VoIP environments. We employ the cumulative online duration of a VoIP user to infer his reputation value. Then we leverage collaborative user feedback to enable the voice spam filter to automatically mark unwanted calls and limit the amount of spam calls from a single user account. Finally, we employ the extended whitelist to complement the collaborative user feedback mechanism. Our preliminary simulation results on LiveJournal users demonstrate that our approach is effective to fight voice spam.

In real VoIP systems, incoming calls may be answered by voicemail or forwarded to other phone numbers. Our future work will enhance the current approach to deal with voicemail and call forwarding. Furthermore, we'll also explore the feasibilities of applying our collaborative user feedback framework to peer-to-peer VoIP environments.

REFERENCES

[1] Enterprise voip adoption in north america will more than double in 2010. [Online]. Available: http://www.voip-news.com/press-releases/enterprise-adoption-america-forecast-projection-021407/

[2] Captcha. [Online]. Available: http://en.wikipedia.org/wiki/Captcha

[3] J. Koskela, J. Heikkila, and A. Gurtov, "A secure p2psip system with spam prevention," in *Proc. The 14th Annual International Conference on Mobile Computing and Networking(MobiCom'2008), poster session*, San Francisco, CA, USA, Sep. 2008.

[4] A. Gurtov, *Host Identity Protocol (HIP): Towards the Secure Mobile Internet*. Wiley, 2008.

[5] R. Dantu and P. Kolan, "Detecting spam in voip networks," in *Proc. The Steps to Reducing Unwanted Traffic on the Internet Workshop(SRUTI'2005)*, Cambridge, MA,USA, Jul. 2005.

[6] V. A. Balasubramaniyan, M. Ahamad, and H. Park., "Callrank: Combating spit using call duration, social networks and global reputation," in *Proc. The 4th Conference on Email and Anti-Spam(CEAS'2007)*, Mountain View, CA, USA, Aug. 2007.

[7] R. Zhang, X. Wang, X. Yang, and X. Jiang., "Billing attacks on sip-based voip systems," in *Proc. The lst USENIX Workshop on Offensive Technologies (WOOT'2007)*, Boston, MA, USA, Aug. 2007.

[8] D. Shin, J. Ahn, and C. Shim, "Progressive multi gray-leveling: A voice spam protection algorithm," *IEEE Netw.*, vol. 20, no. 5, pp. 18–24, Sep. 2006.

[9] Vipul's razor. [Online]. Available: http://razor.sourceforge.net/

[10] Distributed checksum clearinghouses. [Online]. Available: http://www.rhyolite.com/dcc/

[11] R. Zhang, X. Wang, X. Yang, R. Farley, and X. Jiang., "An empirical investigation into the security of phone features in sip-based voip systems," in *Proc. The 5th Information Security Practice and Experience Conference(ISPEC'2009)*, Xi'an, China, Apr. 2009.

[12] S. Garriss, M. Kaminsky, M. J. Freedman, B. Karp, D. Mazieres, and H. Yu., "Re: Reliable email," in *Proc. the 3rd Symposium on Networked Systems Design and Implementation(NSDI'2006)*, San Jose, CA, USA, May 2006.

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

[14] Livejournal. [Online]. Available: http://www.livejournal.com/

[15] CNNIC. Statistical survey report on the internet development in china abridged edition. [Online]. Available: http://www.cnnic.cn/download/2008/CNNIC22threport-en.pdf