A VOIP ANTI-SPAM SYSTEM BASED ON MODULAR MECHANISM DESIGN

Alexander J. Johansen and Woraphon Lilakiatsakun

Faculty of Information Science and Technology, Mahanakorn University of Technology, Bangkok, Thailand
Emails: ajohansen@ieee.org, woraphon@mut.ac.th

ABSTRACT

Internet telephony has gained much popularity as an easy way to communicate and make calls to distant places through the internet. It provides a feature of free telephone calls to be made with their software anywhere in the world. These internet telephony products are sometimes called VOIP products. Like e-mail spam, voice spam also called SPIT (Spam over Internet Telephony) is a common misuse of VOIP products and services that transfer bulk messages to phones through internet and broadcasted through VOIP. This project aims at developing a modular framework for SPIT detection and prevention. This framework is designed to be modular and extensible. The method applied for this purpose is to utilize test server for the recognition of spam voice with the help of multiple modules to detect and prevent SPIT. It fights against spam before the problem of unsolicited voice calls escalates to the same level of email spam. Experiments conducted during the project clearly manifest that designed modular framework is capable of differentiating between human calls and system generated calls. These experiments also validated that this modular framework can filter calls based on mathematical calculations performed by the caller in the Turing module.

Index Terms—*Internet Telephony; VoIP; SPIT; voice spam; Turing Test*

1. INTRODUCTION

Voice spam also called SPIT (Spam over Internet Telephony) refers to unsolicited bulk messages transferred to phones through the internet and are broadcasted through VoIP [5]. The contents delivered could be voice messages, images or videos. Voice spam includes various kinds of advertisements, telephonic polls and other forms of telemarketing. One factor that has augmented use of voice spam for advertisement and commercial purposes is the availability of VoIP services at cheaper rates. Thus Telemarketers find it easier and more economical to send their messages out through VoIP services over the internet [8]. It should be noted that SPIT (Spam over Internet Telephony) is more obtrusive compared to e-mail SPAM, as one only reads e-mail spam with one's consent according to availability of time whereas voice spam is uncalled-for and this can cause disturbance at odd times without prior approval.

To identify SPIT calls we have introduced here a framework with multiple modules, which will identify these calls with the help of a reverse Turing test in the second stage of the test. We have placed an anti-SPIT server in between the SIP (Session Initiation Protocol) proxy and the VoIP gateway so that every call passes through it and only a call passing the tests can reach VoIP server and thus the call receiver. The reverse Turing test we used here was initially developed by Alan Turing in 1950 [7]. A CAPTCHA was initially introduced by Luis von Ahn et-al in 2000. CAPTCHA is a computer program that can generate and grade tests that (a) most human can pass, but (b) current computer programs cannot pass [2][3]. We have employed this reverse Turing test in our modular framework. A detailed design of modular framework has been demonstrated in the module description section.

2. PROBLEM STATEMENT

As the use of VoIP continues to grow at a tremendous pace, the problem of Spam over Internet Telephony is likely to increase in the future. More and more people and companies are switching to VoIP from the traditional telephone networks. A detailed study by Pras and Sinderen shows that 25% of the Western European households switched to VoIP from PSTN (Public Switched Telephone Network) [1]. Thus with the growth of VoIP communication its misuse will also grow and advertisers will take advantage of this facility and will send numerous voice spams to VoIP users. In addition to inconvenience caused to customers, VoIP spam will put more strain on networks as compared to email spam as the message size of VoIP is 10 times larger than email messages.

As SPIT calls are increasing in number with more advertising companies and call centers are using this internet telephony frequently to send a bulk of unwanted voice messages to the people connected over the Internet, it necessitate a solution to stop or avoid such unsolicited calls as they perturb call receivers. For this purpose, we have developed a solution in the form of a modular design that can be implemented in the form of an application server in between the telephone device and the VoIP network and are capable to identify the system generated calls and human calls. Thus this server will help to pass

only authenticated calls to the user. We will see its design and implementation in the further sections.

3. RELATED WORK

VoIP systems, like other email and text based applications are susceptible to abuse by malicious parties. SPIT is somewhat common to SPAM of emails as they both involve sending a bulk of malicious messages and disturbance to a network. As the problem of email SPAM is not a new one, so many models are available in the market to prevent these. One such model is Bayesian model which can filter the SPAM messages based on some predefined words called content filtering and thus can prevent SPAM mails. The same method cannot be applied to VoIP applications because VoIP spam is synchronous in nature, while email spam is asynchronous. Thus the need for research on the creation of a Bayesian model for voice SPIT as well.

Various techniques have been developed for the recognition of SPIT and solve the problems related to it. Some of these methods use a Blacklist method which solves SPIT by blocking the call from the people who are blacklisted in their list. Problem with this methodology is that requirement of white lists limits the call only from the listed users. Additionally, in these techniques it is difficult to identify the authenticated person at the first call. In order to remove these deficiencies, reverse Turing test is applied in these approaches as well and the users who pass this test are added to white lists, and those who fail are added to the blacklist. Now next time when the call is made the system will recognize that person from their lists.

Also a voice detection system based on feedback from users is made where the black and the white lists are prepared based on user's feedback on the call. This algorithm calculates the reputation value through Bayesian learning. Another spam protection algorithm is Progressive Multi Gray-Leveling (PMG) method [9] which calculates two gray levels to form black and white list. However black and white lists are not sufficient for spam detection, and a reverse Turing machine like CAPTCHA can be efficiently used to detect SPAM in VoIP.

An approach of statistical filtering which is similar to Bayesian filtering is also used, however to find SPIT in real time environment is difficult using this technique. Thus we found that a lot of work related to SPIT prevention is in progress and our modular framework is one of these new inventions.

4. SYSTEM DETAILS

This paper aims at developing an anti-spit solution for preventing system generated automated-calls that cause spit into VoIP networks, and an open source software has been used to implement this system. The system has been designed in a way to be easily integrated with productive VoIP networks using SIP protocol.

4.1. System Overview

The modular framework to detect and prevent SPIT is a system which is capable of identifying difference between automated computer based call and a regular human phone call made by open source applications or VOIP applications. This is a system which identifies the SPIT call using the multiple modules in order to tackle the SPIT over VOIP network. We propose this method which will put some form of tests to every incoming call and allow only the valid human call to reach the recipient, thus filtering all the SPIT messages and giving a reliable method of voice data protection over VOIP networks.

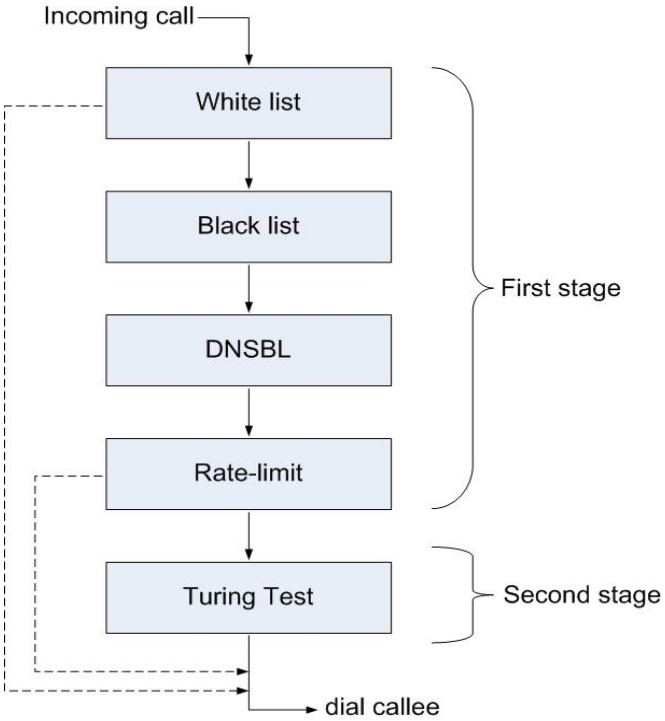

Figure 1. Overview of modular Anti-SPIT framework.

The system is a stand-alone server acting as an application server for a VoIP system. The incoming call is sent to the application server to be tested by multiple modules written as Perl script for the Asterisk PBX. The IP address of the call is extracted from the SIP VIA header to be processed by different modules. For our second stage of the test (Turing test module), if the caller does solve the test, the call is preceded to the next stage i.e. the call is forwarded to the VoIP network to the reach the receiver's end (which is registered on PBX/SIP-SERVER) and if the caller fails the test the call is stopped. This process is demonstrated in the figure 1.

SPIT call that is created as pre-recorded message will be terminated (hang up as no human is to be interacted with the system in the second stage) at the application server and a SIP BYE message is sent to the originator (caller).As mentioned ahead, the media stream could also be connected directly between each end-points by-passing the application server after the caller has solved the Turing test module. This can be done by using SIP RE-invite method. Since not all end-points support RE-invite, we have chosen not to do it. We can see the use

of SIP server and the navigation of call to the call receiver through figure 2.

The media stream (audio) will be bridged as packet2packet bridging after the recipient has answered the call. We don't use re-invite to native bridge RTP stream for compatibility issues. With Packet2Packet bridging, the audio will not go through the Asterisk core (base system used for application server) and it comes directly into the RTP stack and goes directly out. This decreases the amount of memory allocation needed, and enables a reduction in processing requirements.

4.2. Module Design And Development

The system has been developed on a Linux operating system with Asterisk PBX installed. Asterisk was chosen because we don't want to be reinventing the wheel. The modules are a set of code written with Perl working in conjunction with Asterisk via Asterisk Gateway Interface. The modules are executed for each incoming call that is entering the system. The test starts by running the whitelist module first to indicate if the call should skip other modules and send the call directly to the call receiver. The second module is the blacklist module which determinate if the call should be stopped at this stage. Third, a DNSBL module is executed. The last module at the first stage is the rate-limit module which checks if the caller form a specific IP have been calling into the VoIP network more than the threshold that have been set. If the rate limit is over threshold, we can choose if the call should be further tested by sending it to the Turing test module to determinate if there is a human to interact with the test or not.

5. MODULE DESCRIPTION

5.1. White List

A White-list is a database containing IP addresses that system allows to access the SIP proxy without any further tests. The module work by simply extract the SIP header 'VIA' for the IP address the call is coming from. If the IP is found in the database, this means that the call is not a SPIT and should skip all further tests. The module has been implemented as an AGI (Asterisk Gateway Interface) script using the PERL programming language. The script is first executed in the system to determine if the call should be passed on without any further tests. The idea of using a white-list is being currently widely used as Email Anti-spam mechanism. This module requires little processing power from the CPU and it is easy to implement. To add a record to the white-list database, an administrator could be using a MySQL front-end, for example, phpMyadmin. phpMyAdmin is a free software tool written in PHP intended to handle the administration of MySQL over the World Wide Web.

5.2. Black List

The Black-list module works the same way as white-list but in a reverse manner. This module blocks calls whose IP address is in the database. The black-list module is implemented in the same way as the white-list module. The script works in the following steps; The IP address is first extracted from the VIA header and then compared to the records in the database. If it is a match, the call is terminated. A black-list is a list where we can certainly know that the source of the call is sending SPIT. The IP could be from an end-point (softphone or hardphone) or IP addresses of SIP proxies. Once we know for certain that the source is sending SPIT, we can insert the IP address to the database. One drawback of the black-list is that in a SIP environment, the SIP source IP can be spoofed. This is why we have designed a modular Anti-SPIT architecture by using several modules to tackle SPIT.

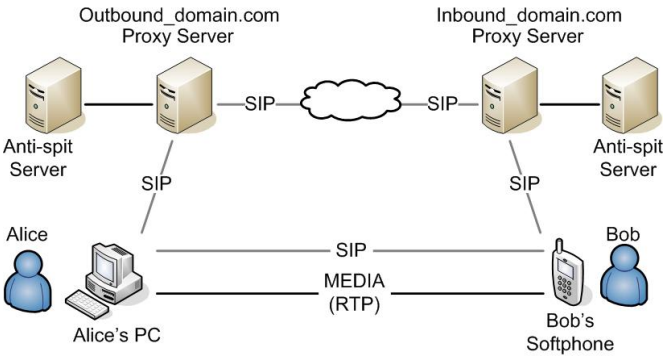

Figure 2. System Design.

5.3. DNS-based Black List (DNSBL)

It is believed that today's spammers will move to the VoIP world to create SPIT on the Internet. If we already know who the spammer is, we could use this information to protect the VoIP network by using a DNSBL lookup. The DNSBL module works as the following. It first extracts the IP address from the VIA header and reserves the order of octets (say, 80.202.87.85, yielding 85.87.202.80). The next step is where the query begins. If the return is positive, which mean that the IP address is returned, then the IP address is a SPAM source. If the return is negative, then the call is not SPIT and will be forward to the network. This module is quite effective where it requires very little of time to determinate if the call is from the spam source or not. The script execution's time is as little as 0.0735 second. This module is also implemented in PERL scripting language as an AGI script. A major advantage of this module is that we don't have to maintain the spam source list. This is taken care of by the provider. The Spamhaus Block List is an example of a DNSBL provider.

5.4. Rate-limit

The rate-limit module is a novel method of fighting Spam over Internet Telephony by limiting calls to be made during a given period of time. This is configurable in the script to suit the environment. For a legitimate caller, he/she probably wouldn't call into system more than a few calls within 1 hour. A person sending unsolicited messages may try to call as much as possible during a particular period. This module works by getting the IP address from the 'VIA' header and does a SQL query to find how many

calls have been made from that IP address. If the amount exceeds the rate-limit, then such a call is considered to be Spam over Internet Telephony and is sent to the second stage (the Turing Test). This will in effect enable the definition of specific rate-limiting, which will ensure that the traffic from such a user is limited.

5.5. Turing Test

In the second stage, the Turing test module is an intrusive test that is trying to determinate if the call is an automated call or a regular human call. While other modules don't need to interact with the caller, this module needs to interact with user. The Turing test module is an enhanced version of Turing test described by Johansen and Lilakiatsakun in their valuable work "A VOIP anti-Spam System based on Audio Turing Test Server" [10]. This version of Turing test has a background noise inserted into the prompt so a transcriber (for example sphinx) can't transcribe the announcement and try to guess the numbers that are going to be calculated. This version of Turing test also has a new algorithm to randomize 2 numbers. Turing test works by asking the caller to make a summation of 2 numbers by using the keypad. If there is a human to interact with the system, the call is granted access to the VoIP network and the call is passed through to the call receiver.

6. CONCLUSIONS AND FUTURE WORK

It is important to notice that to handle spam efficiently, multiple techniques are needed to be applied. We have demonstrated that our framework is capable to recognize between human call and system generated call by using multiple modules to tackle Spam over Internet Telephony (SPIT). The Turing test module is capable of distinguishing between auto- generated and human calls and can thus prevent the SPIT calls to be passed to the person being called. However if has also been found that the reverse Turing test that the modular system performs for authentication is not very user-friendly and requires the caller to interact with the system. This problem can be solved by using a communication pattern detection scheme to test if the call is an automated call by monitoring the voice channel for unexpected voice pattern.

Moreover, our modular framework is an initiative in this field and many other projects with similar aims are in progress. Some of these works include detection of spam in videos, push to-talk and instant message sessions. Also further enhancements to the Turing test can be made to make it more effective for SPAM detection. Future studies should aim at implementing an anti-spam system including more modules that could be added on the fly to respond to new Spam over Internet Telephony behaviors.

As for the claim that audio CAPTCHA can be broken, I would like to propose the use of ideas similar to those proposed in this paper to test the security of audio CAPTCHAs. Audio CAPTCHAs enable the distinguishing of humans from computer robots with high probability, and are still used in various security applications. If audio CAPTCHA is successfully broken these robots and computer programs will be able to impersonate humans and gain access to activities and services that they usually would not. It is therefore of great importance for audio CAPTCHAs to be made more secure.

REFERENCES

[1] Pras Aiko, and Marten J. Sinderen. Dependable and Adaptable Networks and Services: 13th Open European Summer School and Ifip Tc6.6 Workshop, Eunice 2007, Enschede, the Netherlands, July 18-20, 2007: Proceedings. Lecture notes in computer science, 4606. Berlin: Springer, 2007.

[2] Luis von Ahn, Manuel Blum, and John Langford, "Telling Humans and Computer Apart (Automatically) or How Lazy Cryptographers do AI", to appear in Communications of the ACM

[3] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford, "CAPTCHA: Using Hard AI Problems For Security", in Proceedings of Eurocrypt'03 International Conference on the Theory and Applications of Cryptographic Techniques, LNCS 2656,pp. 294-311, Springer-Verlag, Berlin Heidelberg, 2003.

[4] Mobih.com. "VoIP Voice Spam". 13 February 2010 http://www.mobih.com/voice_spam.php

[5] Searchunifiedcommunications.com Definitions. SPIT. 7 Mar 2008. <http://searchunifiedcommunications.techtarget.com/sDefinition/0,,sid186_gci1024458,00.html>.

[6] S Felipe & Deshpande M. "SPAM over IP Telephony (SPIT), Identification and prevention techniques". Georgia Institute of Technology

[7] Turing AM (1950), "Computing Machinery and Intelligence," Mind, 59:236, pp.433-460

[8] Vincent M. Quinten, Remco van de Meent and Aiko Pras. Lecture Notes in Computer Science. ISBN: 978-3-540-73529-8. Vol. 4606/2007, pg 70-77

[9] S. Dongwook, J.Ahn & C. Shim., "Progressive multi gray-leveling: a voice spam protection algorithm," Network, IEEE, vol. 20, pp. 18-24, 2006.

[10] Johansen, A., Lilakiatsakun, W., "A VOIP anti-Spam System based on Audio Turing Test Server ", Proceedings of the ECTI-CARD 2nd Conference on Application Research and Development, 10-12 May 2010, Thailand, 2010.