Detecting SPIT Calls by Checking Human Communication Patterns

J. Quittek, S. Niccolini, S. Tartarelli, M. Stiemerling, M. Brunner, T. Ewald
NEC Europe Ltd., Kurfürsten-Anlage 36, 69115 Heidelberg, Germany; Tel +49.6221.4342.0
e-mail: {quittek, niccolini, tartarelli, stiemerling, brunner, ewald}@netlab.nec.de

Abstract — SPam over Internet Telephony (SPIT) is expected to become a very serious issue in the next years. The threat is going to spin out from the well known email spam problem by bot nets being re-programmed to initiate not just spam emails but also Voice over IP (VoIP) calls. Preventing SPIT is a new problem, because many well-established methods for blocking email spam cannot be applied. Currently, several SPIT prevention methods are being proposed but SPIT prevention research is still at a very early stage. In this paper, we propose an innovative way to detect SPIT calls by comparing applying hidden Turing tests that compare them with typical human communication patterns. For passing these tests, significant resource consumptions at the SPIT generating bot nets would be required which contradicts the spammer's objective of placing as many SPIT calls as possible. The proposed method has several advantages compared to other methods that also interact with the caller. We validated its feasibility with a prototype implementation that we integrated into our modular VoIP security system called VoIP SEAL.

Keywords—SPIT, VoIP, Spam, Turing Test, Communication Patterns

1. INTRODUCTION

Spam is a phenomenon of the networked world. Spam messages are typically related to marketing, fraud, or phishing, and they come in form of traditional mail, email, and phone calls over the Public Switched Telephony Network (PSTN). Now, spam is starting to use a new medium. This new kind of spam is commonly referred to as SPam over Internet Telephony (SPIT).

1.1. Spam over Email

Spam over email has reached a worrying dimension. Today, more spam emails than regular emails are transmitted over the public Internet. Many organizations could not operate anymore without effective email spam filters and spam email has become a significant social and economical factor in the networked society. Spam email is created by a highly efficient infrastructure making use of the so-called bot nets. A typical bot net consists of a huge number of computers at which a spammer, the owner of the bot net, has illegally installed software for producing spam email. This software has been installed by a virus, a Trojan horse, or another program that exploits security leaks of networked computers and that silently installs the spam software without the regular operator of the computer noticing it. For distributing a certain spam message, a spammer triggers reachable computers in a bot net to generate huge numbers of emails. Typically, the generated average load per computer in a bot net is relatively low in order to avoid the spam software to be detected by the computer's administrator. Once a bot net has been built, the cost for producing a single spam email is very low.

1.2. Spam over PSTN

Spam over PSTN is by far not as relevant in the PSTN as email spam in the Internet. Only a very small fraction of all PSTN calls are spam. This is mainly due to the fact that producing PSTN spam is relatively expensive. Typically, PSTN spam is generated at call centers that are served by a PSTN operator that charges the spammer for each single call. Consequently, the cost for a PSTN spam call is several orders of magnitude higher than the cost for a spam email.

1.3. Spam over Internet Telephony (SPIT)

With the migration from PSTN to Internet telephony, the cost for spam over phone calls will drop dramatically. The bot nets serving as core of the email spam infrastructure can easily be modified to serve also for generating SPam over Internet Telephony (SPIT). Therefore, we expect SPIT prevention systems to become an essential feature of the future Internet telephony infrastructure as spam filters are in today's email infrastructure.

The first reported incident of SPIT distribution occurred in the Skype network where, in June 2006, spammers called Skype network participants for delivering telemarketing messages. There is little doubt that this first incident will have many successors. As soon as the number of Internet phones has reached a critical number that makes SPIT more attractive, spammers will start generating SPIT more extensively.

1.4. SPIT Prevention Methods

Extrapolating the strong grows of Internet telephony, this critical number will be reached in the next few years. Then effective countermeasures will be needed. Unfortunately, many means very useful for blocking email spam cannot be applied to SPIT. This is caused in the different nature of the medium. Email is transmitted asynchronously, while SPIT uses real-time communication. Email can be stored and its entire content can be checked before delivering it to its final destination. This is different for SPIT. A SPIT call immediately disturbs the callee by a ringing phone. Content checking can only be performed after the SPIT calls have already caused damage, such as reducing productivity at work by interruptions or reducing quality of life by disturbing callees in their private environments.
Hence, new methods for SPIT prevention need to be developed. Rosenberg and Jennings [1] give an overview of the current state of the art. Basically, we can distinguish three kinds of methods:

1. Non-intrusive methods are just based on the exchange and analysis of signaling messages. They do not create inconveniences for the caller and they do not disturb the callee, if they block SPIT calls successfully. However, the effectiveness of such methods is limited and not expected to be sufficient. This kind covers blacklisting, detections of call rates, call patterns, reverse lookup of caller DNS entries, etc.

2. Caller interaction methods that create inconveniences for the caller by requesting him/her to pass a checking procedure for the call which is typically based on question and answer or more general on action and reaction checking.

3. Callee interaction methods that exchange information with the callee on a per-call basis. An example is asking the callee before accepting a certain call. These methods are problematic, because they might miss the general goal of protecting the callee from disruptions. An example for a useful callee interaction method is receiving feedback from the callee on false negatives. With this feedback, an adaptive SPIT prevention system can avoid letting a SPIT message pass the system twice.

1.5. Turing Tests

In this paper we focus on caller interaction methods. We assume that smart spammers will be able to fake signaling messages and might be able to pass non-intrusive checks. Then more protection can be provided by caller interaction methods. One general method already suggested in [1] is running a Turing test on the caller. A Turing test [5] is a test that tries to find out whether a communication partner is a human being or a machine. In our case, the Turing test tries to find out whether the caller is a human being calling or a computer of a bot net. Certainly, SPIT software could be equipped with means to make it hard to distinguish its behavior from human behavior, but since required technologies, such as speech recognition and artificial intelligence are still in their infancies, we expected, that the high effort required to pass such a test will not be spent by spammers.

Classical Turing tests [5] consist of a sequence of questions to be answered in order to pass the test. Applied to SPIT prevention, such a sequence might be perceived as annoying or even impolite by the caller. Therefore, we were looking at Turing tests that are not explicitly performed, but hidden in an acceptable procedure that does not give the impression of being a test.

This paper introduces a group of Turing tests that are based on checking human conversation patterns. Section 2 explains the general communication patterns that are the checked by the tests. A variety of Turing tests based on these patterns that we developed for SPIT prevention is introduced in Section 3. Section 4 describes our implementation of these tests and their integration into our VoIP Secure Application Level firewall (VoIP SEAL) that provides protection from Denial of service and SPIT attacks.

2. HUMAN TELEPHONY COMMUNICATION PATTERNS

Human communication pattern in phone conversations have been studied extensively and some basic patterns are assumed to be well understood. The ITU-T [3] and Hammer et al. [4] use a simple conversation model with 4 states. In state M (mutual silence) both participants are silent, in states A and B exactly one of them speaks, and in state D (double talk) both are speaking at the same time, see Figure 1.

![Figure 1 – Conversational model with four states according to [4]](image)

2.1. Double Talk

For obvious reasons it turned out that typical conversations do not remain for long periods of time in state D. How much time during a conversation is typically spent in time D varies among different cultures. In Japanese conversations a change of the speaker from state A to B or vice versa typically includes a short period of mutual silence (state M), while in Italian conversations much more often there is a period of double talk (state D) included. As an average for English, Italian and Japanese, the ITU reports 6.59% of the time of a conversation is spent in state D with an average length of 0.23 seconds that the conversation remains in this state before switching to another state. We see that double talk is rare over time in a conversation and if entered, it does last only for a very short period of time.
2.2. Call Start Pattern

There is a typical common state sequence at the beginning of a telephone call. The callee starts the call by accepting it, traditionally by lifting the handle of the ringing phone. Then typically the callee is the first to speak stating his name and/or a short greeting. Then it is the callers turn to explain who is calling before the core conversation can start. Figure 2 illustrates this pattern. The transition from callee speaking to caller speaking may occur without any intermediate state, with a short period of mutual silence, or with a short period of double talk. A long period of double talk at this stage of the conversation is considered as unusual.

3. Hidden Turing Tests

We tried to realize a Turing test without letting the caller know that he or she is being tested. We hide our Turing test by just checking the communication pattern of the caller and not asking explicit questions for performing the test. The basic approach we followed is assuming that a violation of the call start pattern described in the previous section is a strong indication of a calling machine. Based on this assumption we perform one or two tests. The first one, silence checking is performed always and the second one, answer length checking, is performed conditionally.

3.1. Silence Checking

For silence checking the voice energy of the caller during the initial greeting at call start is evaluated. If the signal energy exceeds a certain value, then it is assumed that the caller is violating the call start pattern and hence is classified as machine.

Some constraints apply to this check. First, as already mentioned earlier, a short period of double talk at the end of the greeting is acceptable for a fraction of a second. The acceptable length of a double talk period varies among different cultures.

Second, the caller might call from a noisy environment. Then the received voice data stream might not carry complete silence, but a continuous background noise. For still detecting that the caller is silent, a threshold is required that distinguishes speech from background noise.

Third, there might be short peaks of noise on the voice channel caused by transmission or codec problems or caused by sources of noise in the caller's environment. Such short peaks of noise should not be used as indicator for a calling machine. Rather it is necessary to check whether the caller exceeds the threshold for some given minimum period of time.

Considering all these constraints, the hidden silence checking can be performed by receiving the call at an answering machine that will generate the greeting for the caller, which could be something like "Welcome to the X company. Please wait while your call is forwarded".

A nice trick that can be played on the caller is sending instead of a greeting the sound of a ringing phone. A human caller listening to this sound would identify it as the phone that he or she is calling to be still ringing. A machine for generating SPIT would probably just concentrate on sending the voice signal with SPIT content and not have the capability and resources to analyze the received voice signal such that it could identify the sound of a ringing phone.

3.2. Answer Length Checking

Answer length checking can be applied if silence checking does not produce a clear result. Here the communication pattern after the initial greeting is checked. The check is based on a question the caller is asked at the end of the greeting, for example, the caller can be asked for the name of the person that is to be called. Anyway, the question must be selected such that any valid human reply would only last a very short period of time. If a caller gives a long answer, such as the streaming of a SPIT message, then the caller is assumed to be a machine that should be blocked.

3.3. Challenges for SPIT Generators

The tests described were invented in order to constitute a big challenge for SPIT generators to overcome. The "business model" of SPIT is based on the fact that SPIT generators have to deliver as many messages as possible in a short period of time in order to increase the number of "hits" per time unit. Moreover, it is also likely that the same bot nets that are used today for sending email spam will tomorrow be re-programmed to send SPIT calls.

In order to pass the tests described (silence checking, answer length checking), the software for sending SPIT calls would need to check the voice stream sent by the callee in order to understand when the other party is talking and even more challenging what the other party is saying. In order to achieve this, a SPIT generation software would need to consume resources for receiving the voice stream from the callee and for performing computationally expensive speech recognition in

---

Figure 2 – Call start pattern.
order to understand which question the Turing tests is asking.

Both actions would consume resources and reduce the achievable SPIT call placing rate while their effectiveness would still be questionable for the spammer. With all means available today, speech recognition from an unknown speaker is still unreliable. But even if some tests could be passed by really intelligent SPIT generators (anyway it is not expected that SPIT bot net will be equipped with such intelligence in the near future), consuming resources in tasks not functional to the SPIT objective of placing as many calls as possible is contrary to the “business model” of SPIT.

Thus, the hidden Turing tests constitute a significant challenge for SPIT generators. They have potential to reduce the SPIT problem either by improving the detection of SPIT calls from generators that cannot deal with the test, or by reducing the SPIT generation rate by consuming resources at generators that have means to pass the tests.

4. Implementation and Integration

In this section we describe our implementation of the hidden Turing tests how the tests were integrated into an existing SPIT prevention system. The application scenarios for the resulting system are also presented and resulting advantages of the solution are discussed.

4.1. Turing Tests Implementation

We implemented the hidden Turing tests described in Section 3 as a separate client application. The Turing Test (TT) client was built using the C/C++ eXtended osip (eXosip2) library and it is therefore a SIP client conformant to core SIP standards [2].

The TT client listens on a specified port and waits for incoming calls; as soon as an INVITE is received by the client the call is initiated following standard SIP procedures and a configurable audio file is streamed over RTP back to the caller. While doing this the TT client checks for the RTP stream coming from the caller, decodes it using the negotiated codec and computes the instantaneous signal energy of the stream and the window-averaged one. In order to decide if the caller passed the first test (silence checking) the TT client computes the percentage of time where the window-averaged signal energy of the caller exceeded the threshold configured. If this percentage is above a configurable parameter, set after a learning period in order to avoid false positives due to background noise, the call is torn down because it is thought to be a SPIT one initiated by a machine. An analogous check on the signal energy is performed for detecting the answer length in the period after the greeting message.

If both tests are successfully passed the TT client forwards the call to the intended recipient of the call using standard SIP REFER message. Moreover it adds a randomized cookie that is appended to the REFER message certifying that the call passed the tests avoiding malicious request being forged to pretend having passed the tests.

Figure 3 – Turing Tests client monitoring interface

For monitoring TT client and fine tuning SPIT detection we added a graphical interface shown in Figure 3. The graph in this screenshot shows the signal energy of an incoming voice call (green/light grey line) that is first flattened (blue/dark line line) and then compared to a threshold separately in phase 1 (silence checking) and in phase 2 (answer length checking). The depicted instance shows the signal energy of an incoming SPIT call. The call is identified as SPIT because the energy level clearly exceeds the threshold during silence checking in phase 1.

4.2. VoIP SEAL Overview

VoIP SEAL (SEcure Application Level firewall) is a prototype of a modular intrusion detection and prevention system against VoIP DoS (Denial of Service) and SPIT attacks. When talking about SPIT, VoIP SEAL can be referred to as a SPIT prevention system. The SEAL framework is described in detail in [6].

VoIP SEAL relies on the cooperation of multiple modules identifying the Maliciousness Level (ML) of a SIP message (in the case of SPIT prevention, a SIP INVITE is used to initiate a call and it is object of the inspection) as a result of the sum of multiple scores generated by the single non-intrusive modules.

The task of VoIP SEAL in its SPIT prevention mode is to generate alarms and/or block malicious calls initiations when the ML exceeds a certain threshold. The VoIP SEAL architecture was kept modular in order to easily integrate new detection and prevention modules as the research and development in the VoIP security area progress.

4.3. Turing Tests Integration into VoIP SEAL

VoIP SEAL was originally implemented using only non-intrusive methods and one threshold to block or pass calls initiations. The decision on whether to block or not the INVITE
was taken as the result of the comparison of the ML calculated by non-intrusive methods with the threshold. If the ML was higher of the threshold the message was blocked. For the purpose of SPIT prevention, VoIP SEAL was extended in order to make use of caller interaction methods. To achieve this, the number of thresholds used was increased to two.

4.4. Application scenarios

The hidden Turing tests described above classifies an incoming call as SPIT if a calling machine immediately starts its SPIT message, but also if a human caller interrupts the greeting being impolite by not following the common communication pattern. In both cases, the SPIT prevention system would terminate the call because silence checking failed (see section 3.1), optionally after sending a pre-recorded voice message explaining this.

The intrusiveness of these Turing tests can be minimized by different configurations and it is customizable depending on the application scenarios. For example, in order to keep the intrusiveness low (e.g. in the case of a company willing not to scare its customers) instead of a greeting the TT client could send a ring tone. A human caller would then assume that the call has not yet been established, while a SPIT engine that does not analyze the greeting message would assume that it can start sending the SPIT message, because an established connection was signaled.

More intrusiveness could be used depending on scenarios requirements; what is still considered to be well acceptable is a greeting message that tells the caller that his call is being forwarded and will be established soon.

A stronger check can be achieved with the answer length checking method described in Section 3.2, according to which the greeting can be followed by a quick simple question, for example the name of the called person or the solution of a simple mathematical puzzle. Such a question should be made such that a short answer can be expected with high probability. Then the method could check if first the caller starts speaking briefly after the question was made and stops talking and remains silent for at least a short time after answering the questions. For both checks, no speech recognition is necessary. Detection of the voice energy level switching from 'low' to 'high' and back to 'low' is sufficient.

4.5. Advantages of the Turing tests

The advantages that the implemented Turing tests have are listed here in terms of functionality that they provide to the system:

1. are polite enough not to offend the caller,
2. are quick enough not to require too much patience from the caller,
3. work well with callers that have different kinds of background knowledge,
4. work well with caller using different kinds of pronunciation,
5. work well with callers speaking different dialects or languages,
6. are simple enough to be implemented on relatively cheap devices at the callee's side,
7. create a resource-intensive and complex task for a
machine imitating a caller.

So far extensive functional testing of the software implementing the Turing tests was performed only in a laboratory environment to show correctness of implemented functionality against both SPIT generators built for testing purposes and humans. Scalability tests of the TT client showed that it is able to handle about 30 calls on parallel on a standard desktop PC with a 2.08 GHz AMD Athlon CPU equipped with 192MB of memory.

It is difficult to assess the effectiveness that such methods have in the real world because of the lack of publicly available traces documenting SPIT calls but, as said in section 3.3 it is sure that they will help in reducing the number of SPIT calls that a SPIT generator can initiate.

5. CONCLUSIONS

In this paper we described how the email spam problem, could evolve in the next years towards a problem affecting the next generation telephone system based on IP. The name of this threat is commonly referred to as SPam over Internet Telephony (SPIT). In order not to underestimate the problem as happened in the email case, SPIT prevention methods should be investigated already at the early stage when only few occurrences of the threat are reported.

We classified the SPIT prevention methods in three categories (non-intrusive methods, caller interaction methods and callee interaction ones). After discussing non-intrusive methods in [6], this paper focuses on caller interaction methods. We proposed two Turing tests for distinguishing human callers from automatic SPIT generators running on bot nets.

They are based on checking human communication patterns and have the advantage that they are hidden from the caller and therefore will not be perceived as impolite call interference.

An implementation and integration of the hidden Turing tests into our modular VoIP SEAL system for DoS and SPIT prevention showed the feasibility and scalability of the proposed methods. Objective of further work will be the performance evaluation of the entire SPIT prevention system using both non-intrusive and caller interaction methods. A problem for full evaluation of the system is the small number of available SPIT call records that can be used for testing the failure rate. Therefore the first evaluation will use spam calls recorded in the PSTN.

REFERENCES