SPAM over Internet Telephony and how to deal with it

Dr. Andreas U. Schmidt[1] · Nicolai Kuntze[1] · Rachid El Khayari[2]

[1]Fraunhofer Institute
SIT
{andreas.schmidt|nicolai.kuntze}@sit.fraunhofer.de

[2]Technical University Darmstadt
{rachid.el.khayari}@googlemail.com

Abstract

Nowadays telephony has developed to an omnipresent service. Furthermore the Internet has emerged to an important communication medium. These facts and the raising availability of broadband internet access have led to the fusion of these two services. VoIP is the keyword that describes this combination.

Furthermore it is undeniable that one of the most annoying facets of the Internet nowadays is email spam, which is considered to be 80 to 90 percent of the email traffic produced.

The threat of so called voice spam or Spam over Internet Telephony is even more fatal than the threat that arose with email spam, for the annoyance and disturbance factor is much higher. From the providers point of view both email spam and voice spam produce unwanted traffic and loss of trust of customers into the service.

In this paper we discuss how SPIT attacks can be put into practice, than we point out advantages and disadvantages of state of the art anti voice spam solutions. With the knowledge provided in this paper and with our SPIT producing attack tool, it is possible for an administrator, to find out weak points of VoIP systems and for developers to rethink SPIT blocking techniques.

1 What is SPAM over Internet Telephony?

In order to know how to deal with SPIT, we must at first know what SPIT is and we will find that SPIT is described very similar in different publications and the descriptions can be summarized as ‘unwanted’, ‘bulk’ or ‘unsolicited’ calls. In [2] e.g. SPIT is defined as ‘unsolicited advertising calls’, which is of course already a special form of SPIT (namely advertising calls). In [3] SPIT is defined as ‘transmission of bulk unsolicited messages and calls’ which is a more general definition than the first one, as it doesn’t characterize the content and includes also messages. Nevertheless the most precise definition is found in [1] where ‘Call SPAM’ (as the authors call it) is defined as ‘a bulk unsolicited set of session initiation attempts (e.g., INVITE requests), attempting to establish a voice, video, instant messaging, or other type of communications session’. The authors of [1] go even one step further and classify that ‘if the user should answer, the spammer proceeds to relay their message over the real-time media.’ and state that this ‘is the classic telemarketer spam, applied to SIP[1]’. We can easily see that the presented definitions so far are very similar, but differ in their deepness.

[1] The whole discussion is based on the Session Initiation Protocol (SIP, RFC 3261)

1.1 SPIT is not SPAM!

Although SPIT contains the phrase ‘SPAM’ and has some parallels with email spam, it also has major differences. The similarity of email spam and SPIT is that in both cases senders (or callers) use the Internet to target recipients (or callees) or a group of users, in order to place bulk unsolicited calls [3]. The main difference between email spam and SPIT is that an email arrives at the email server before it is accessed by the user. This means that structure and content of an email can be analyzed at the server before it arrives at the recipient and so SPAM can be detected before it disturbs the recipient. As in VoIP scenarios delays of call establishment are not wished, session establishment messages are forwarded immediately to the recipients. Besides this fact the content of a VoIP call is exchanged not until the session is already established. In other words if the phone rings it is too late for SPIT prevention and the phone rings immediately after session initiation, while an email can be delayed and even, if it is not delayed, the recipient can decide if he wants to read the email immediately or not. In addition to these aspects another main difference between email spam and SPIT is the fact, that the single email itself contains information that can be used for spam detection. The header fields contain information about sender, subject and content of the message. A single SPIT call in contradiction is technically indistinguishable from a call in general. A SPIT call is initiated and answered with the same set of SIP messages as any other call.

1.2 How does a SPIT producing tool work?

The next questions that have to be considered are, how attackers behave and what techniques are used in order to generate SPIT. We can split the SPIT process into three steps:

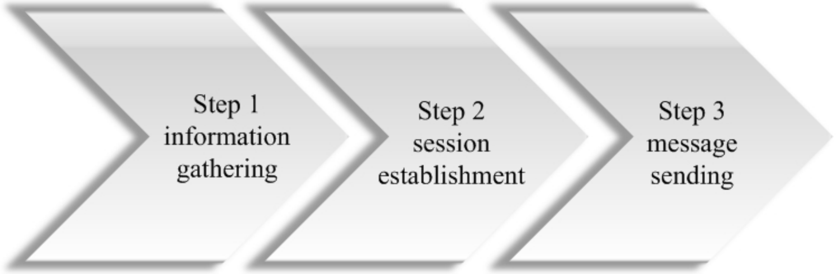

Figure 1: Three steps of SPIT

These three steps can be viewed as fundamental and are fulfilled by any attacker in a systematic manner. The first step the ‘Information gathering’ is used in order to find out targets for possible future attacks. The second step the ‘session establishment’ leads to the establishment of a communication session wit the victims. In the last step of SPIT the ‘message sending’ media is exchanged between attacker and victim.

1.2.1 SIP XML Scenario Maker

Now what we need is a tool that implements the presented SPIT process. Therefore we developed SIP XML Scenario Maker (SXSM), a tool with which it is possible to scan systematically for targets, establish sessions to these targets and exchange pre recorded media. SXSM is based on SIPp [12] developed by HP and expands SIPp with a graphical user interface that allows us to fulfill the requirements stated above. SXSM can be used in order to create any kind of SIP messages, put them into a sequence as XML scenarios, execute created scenarios and evaluate the result of the execution.

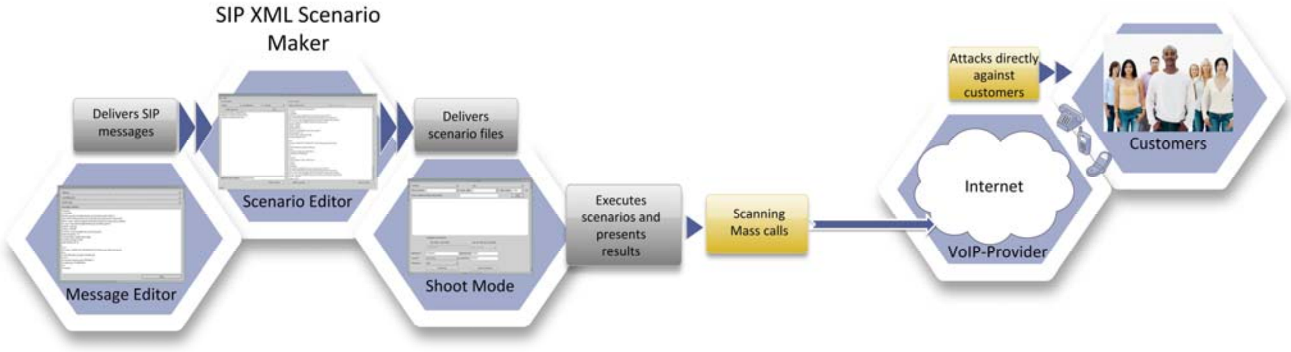

Figure 2: SXSM Workflow

We will now take a look on how SXSM works. First SXSM consists of the following three modes.

1.2.1.1 Message Editor

The message editor can be used to organize and create custom SIP messages that can later be used in the scenario editor. The power of this mode lies within the possibility to create any kind of SIP message and manipulate SIP header fields in any way. SXSM is pre configured with a complete set of standard compliant SIP messages.

1.2.1.2 Scenario Editor

The scenario editor is the core element of SXSM. In this mode the user can create SIP scenarios based on the message bricks created in the message editor mode. Additionally the created scenarios can be organized in different sets.

In order to create a new scenario the user simply needs to select the messages that should be contained in the scenario in the preferred order and then let SXSM create an XML scenario file automatically. The XML file can then be viewed in detail and tweaked manually (if wished).

1.2.1.3 Shoot Mode

Within the shoot mode the user puts the previously generated scenarios into a batch and execute them one after the other. The results of the execution are presented within the process output window. Before execution the user can specify scenario specific settings, such as e.g. how often and in which time intervals the scenario should be executed. Additionally he can adjust and set global parameters such as information about target (targeted username, remote IP, remote port) and about himself (local IP, local Port).

The scenarios are then played with the specified settings and the result is presented as a success rate. If e.g. 5 out of 10 selected scenarios were finished successfully the success rate would be fifty percent. Additionally the user can consult log files that are presented in case of unsuccessful execution.

1.2.2 How can we use SXSM as attack tool?

The goal that we wanted to reach was the creation of a tool with which it is possible to implement a SPIT attack in its fundamental three steps.

With SXSM we can now create scenarios for each of the three steps and execute them against any target. At first we can create a scenario for information gathering which we can call a scan attack scenario. The scan attack scenario can be implemented as follows. As SXSM contains the possibility of injecting values from CSV (Comma Separated Values) files, we would create a CSV file containing all usernames that we want to scan for. Then we would create a scenario with standard SIP messages that works e.g. as follows:

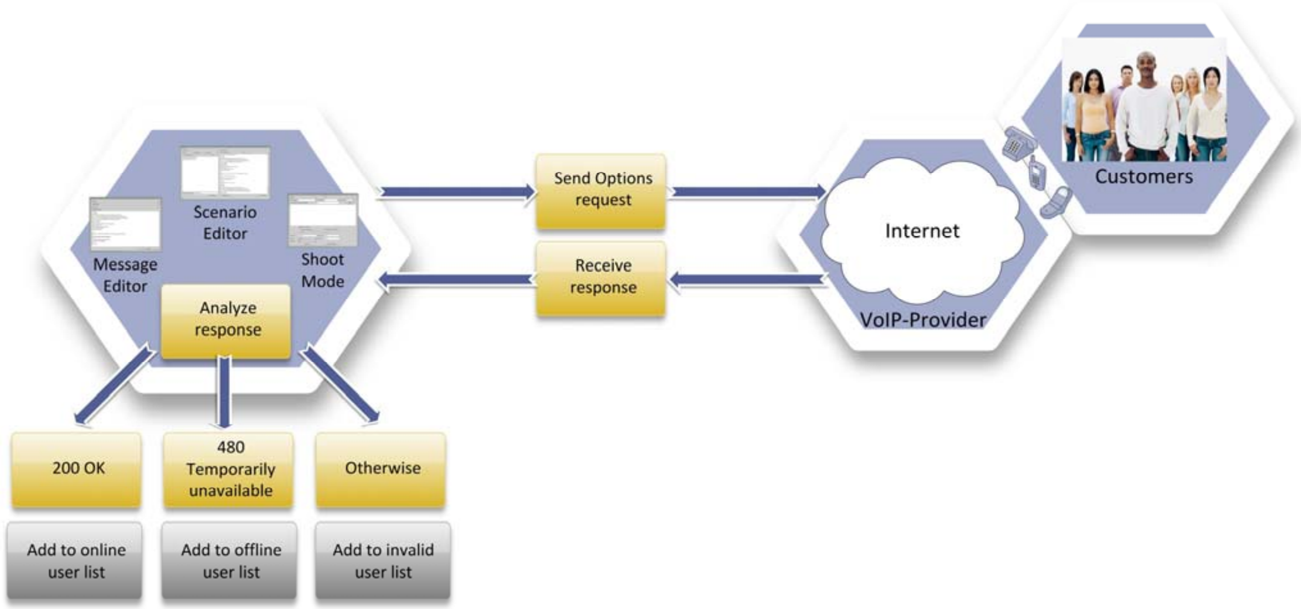

Figure 3: SXSM Scan Attack

With a scenario file that corresponds to the presented sequence of messages and logging methods, an attacker can populate lists of targets for future attacks.

The next steps would be session establishment and media exchange. With SXSM we could create a second scenario that establishes sessions to the targets collected in the first step.

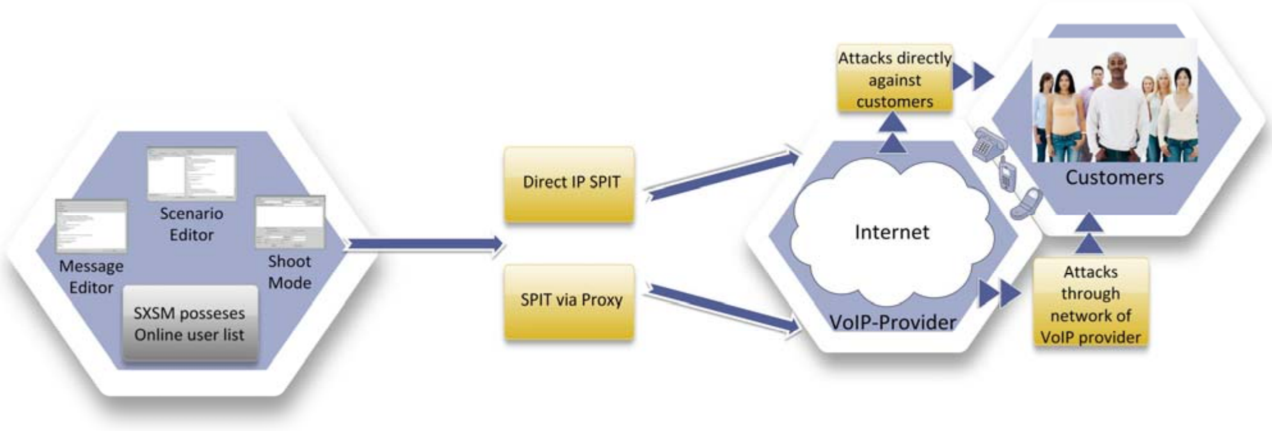
Scenario Editor
Message Editor
Shoot Mode
SXSM posseses Online user list
Direct IP SPIT
SPIT via Proxy
Attacks directly against customers
Internet
VoIP-Provider
Customers
Attacks through network of VoIP provider

Figure 4: SXSM Session establishment

In Figure 4 we can see, that session establishment can be reached via two ways. SPIT via Proxy uses a valid account from targeted VoIP provider and the provider’s network (Proxy, Registrar) in order to establish a session. In Direct IP SPIT context the targeted endpoint (telephone) is contacted directly via his IP and Port.

With these two simple scenarios based on standard SIP messages we implemented the whole SPIT process and are able to execute the attack against any target.

1.3 How can we stop these SPIT attacks

The question that we will answer now is what has been done so far in order to mitigate the threat of SPIT. Therefore we will present a short overview of countermeasures that have crystallized in research.

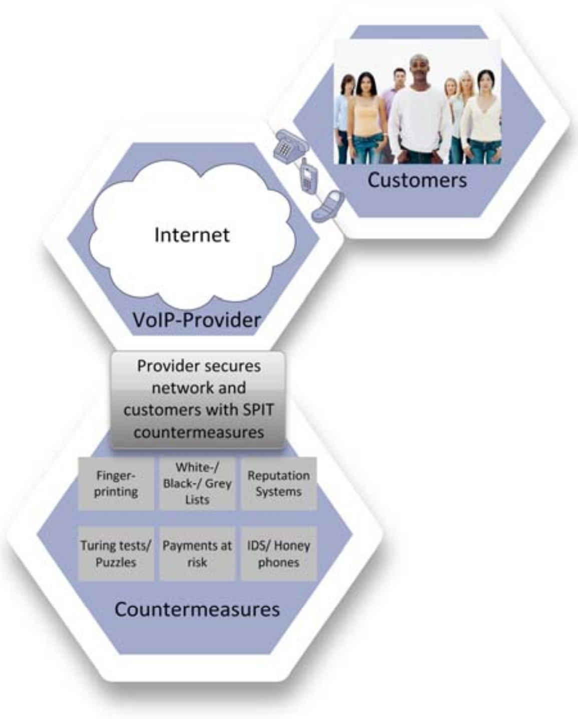

Figure 5: SPIT Countermeasures

1.3.1 Device Fingerprinting

The technique of active and passive device fingerprinting is presented in [4] and is based on the following assumption: Having knowledge about the type of User Agent that initiates a call, helps finding out whether a session initiation attempt can be classified as SPIT or not. So if we can compare the header layout and order or the response behavior of a SIP User Agent with a typical User Agent, we can determine if the initiated session establishment is an attack or a normal call. The authors describe two types of techniques that can be used for that purpose Passive and Active Device Fingerprinting.

1.3.1.1 Passive Fingerprinting

The e.g. INVITE message of a session initiation is compared with the INVITE message of a set of ‘standard’ SIP clients. If the order or appearance of the header fields does not match any of the standard clients, the call is classified as SPIT. The fingerprint in this case is the appearance and the order of the SIP header fields. The authors of [4] present a list of collected fingerprints of standard hard and soft phones.

1.3.1.2 Active Fingerprinting

User Agents are probed with special SIP messages and the responses are analyzed and compared with the response behavior of standard clients. The fingerprint in this case is the returned response code and the value of certain header fields. If the fingerprint doesn’t match any of the standard clients, the call is classified as SPIT. The authors recommend the sending of specially crafted standard compliant and non compliant OPTIONS requests, in order to analyze the response behavior of a client.

1.3.2 White Lists, Black Lists, Grey Lists

The White List technique is presented e.g. in [2] [1] and works as follows: Each user has a list of users that he accepts calls from and any caller who is not present in the list will be blocked. In addition the private White Lists can be distributed to other users. If e.g. a caller is not present in the White List of the callee, White Lists of other trusted users can be consulted and their trusted users (up to a certain level). Black Lists are the contradiction of White Lists and contain only identities that are already known as spammers. Any call from a caller whose identity is present in the callee's Black List is blocked. Even Black Lists can be implemented as distributed Black Lists, where a callee can consult the Black Lists of other users. Grey listing works as follows: On initial request of an unknown user (not in White List) the call is rejected and the identity is put on the Grey List. As stated in [2] in case the caller tries calling back within a short time period, the call will be accepted. An adaption of this technique is described in [1] as Consent Based Communication. In case of Consent Based Communication the call of an unknown caller is initially blocked and put on the Grey List. The callee can consult the Grey List and decide, if he will accept future calls from this identity or block it permanently (e.g. put it on the Black List).

1.3.3 Reputation Systems

Reputation based mechanisms are described in [5] or in [1] and can be summarized as follows: After receiving a call, the callee can set a reputation value for the caller, that marks this caller as spitter or not. This reputation value must be assigned to the identity of the caller and can be used for future session establishment requests. This technique can be used e.g. as attachment to Grey listing [1] in order to provide a better decision basis. The authors of [5] explain that the user feedback can be used additionally for calls that were not detected by other SPIT preventing components. The way the reputation value is generated can differ. The SPIT value can be e.g. an additional SIP header, or included in a special error response code or distributed via SIP event notification mechanism. An adaption of this method can be found in [6] where user feedback is combined with statistical values in order to calculate a reputation value. The reputation value is e.g. composed of a value representing the number of times an identity occurs in other users' Black Lists, call density, call length or similar statistic values. The assumption behind this approach is that the calculated value will differ much between 'normal' users and spitters.

1.3.4 Turing tests, Computational Puzzles

Turing test are tests where the caller is given a challenge that a human can solve easily and that is hard to solve for a machine. Therefore Turing tests or CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) are tests, that countermeasure Calling Bot attacks in VoIP scenarios and work as follows: On initial call establishment attempt, the caller is transferred to an interactive System where he is challenged with a task e.g. dialing 5 digits that he is hearing (so called Audio CAPTCHA). While the numbers are read out background music or any other kind of noise is played, so that speech recognition systems can't be used to solve the task. A human caller in contradiction will solve the task without difficulties and only if the task is solved, the call will be forwarded to its destination. Turing tests can be used in combination with white lists, solving the introduction problem as described in [8].

Computational Puzzles seem at first sight very similar to the Turing tests concept. As described in [7] a SIP Proxy or User Agent Server can request from a User Agent Client (caller) to compute the solution to a puzzle. The goal of this method is to raise CPU costs of a call and so reduce the number of undesirable messages that can be sent. Turing test in contradiction have the goal to block non-human callers, as described above.

1.3.5 Payments at risk

Payments at risk mechanisms can be used in order to demand payment from an unknown caller. In [1] this technique is described as follows: If user A wants to call user B, he must first send a small amount of money to user B. When User B accepts the call and confirms that the call is not a SPIT call, the amount will be charged back to user A. With this technique it is possible to raise costs for SPIT callers while keeping ‘normal’ calls cheap. In [1] it is described as an auxiliary technique that solves the introduction problem of White lists, this means, that payment is only required for callers who are not on the White list of callee. In general the payment could be demanded for every call, but this would make the telephony service more expensive. An adaption of this method is described in [9]. Here the Payment technique is used in combination with a SPIT prediction value that is computed at server side. If the SPIT likelihood is high the call is rejected, if the SPIT likelihood is small the call is forwarded to the callee and if the SPIT likelihood value is in between payment is demanded automatically. Only if the payment is fulfilled the call will be forwarded to its target. The difference between the two approaches is that in the first case the paid amount is only charged back for non SPIT calls and in the second case, callers who reject payment are treated as spitters.

1.3.6 Intrusion Detection Mechanisms, Honey phones

Intrusion Detection Systems are (generally described) systems, that can be used for detection of any kind of abnormal behavior within an e.g. network and so reveal attacks. An implementation of this technique is presented in [10] based on the Bayes inference approach combined with network monitoring of VoIP specific traffic. The Intrusion Detection System is designed as a defense mechanism against different VoIP specific attacks including scan attacks and SPIT attacks. For every attack a conditional probability table (CPT) is defined for variables such as request intensity, error response intensity, parsing error intensity, number of different destinations, maximum number of dialogs in waiting state, number of opened RTP ports, request distribution and response distribution. The concept behind this technique is that the different attacks affect these variables in different ways, e.g. a SPIT attack usually has a higher probability of a higher number of destinations than normal traffic. So a belief of a network trace can be calculated with the aid of likelihood vectors that were defined in the CPT. In the end the trace can be categorized as an attack or normal trace (refer to [10] for detailed description).

Honey phones can be used as part of an Intrusion Detection Systems as described in [2] [11] and can be viewed as VoIP specific Honey pots. A Honey pot represents a part of a network that is not accessible by ‘normal’ users and therefore any access to the Honey pot can be viewed as an attack. VoIP specific Honey pots can be used in order to detect Scan attacks or SPIT attacks. As described in [11] the Honey pot is implemented as a complete parallel VoIP infrastructure that is logically and physically separated from the normal network and so simulates a whole VoIP network. Let us assume a Scan attack as described earlier. When the attacker sends e requests to valid assigned contact addresses they are forwarded through the normal SIP network (Proxy, UAC), but when the attacker tries to send requests to an unassigned or invalid contact the request will be forwarded to the Honey pot, where the requests can be monitored and treated adequately.

1.4 Are we now secured against SPIT?

The question must be answered wit ‘no’, because the presented countermeasures inherit weaknesses that can be exploited by attackers. In fact we can expand SXSM with special scenarios that implement exploits of weaknesses of countermeasures, as we can see in the figure below.

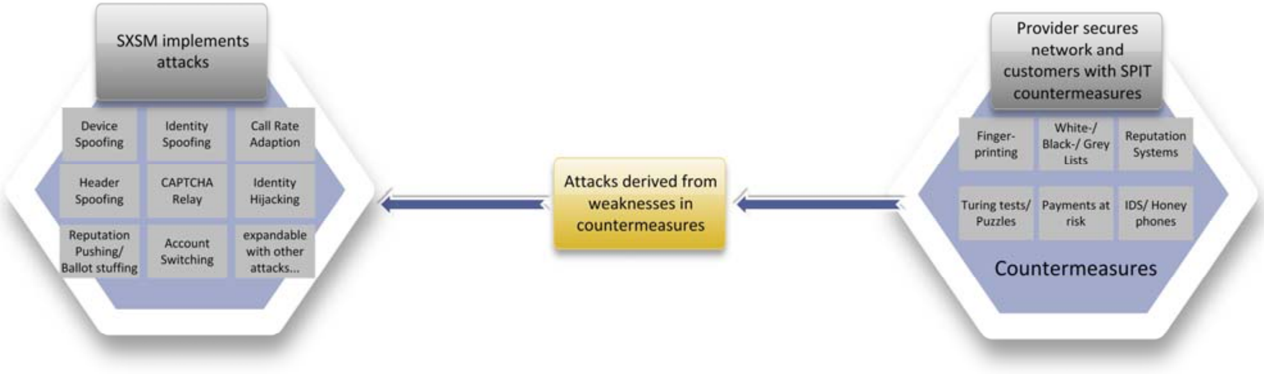

Figure 6: SXSM Attacks

1.4.1 Weakness of Device Fingerprinting

The weakness of passive fingerprinting is described by the authors of [4] themselves. As passive fingerprinting only analyses the order and existence of the header fields of an INVITE message, an attacker simply needs to order the header fields in the same way as one standard client. In that case the passive fingerprinting mechanism can’t detect the attack.

We can state nearly the same for active fingerprinting as an attacker only needs to behave like one standard client when receiving unexpected or non standard compliant SIP messages. It is very simple for an attacker to develop an attacking SIP client that behaves exactly like a standard client, because he can imitate the behavior of a standard client e.g. with SXSM. We can call this attack **Device Spoofing**. As Device Fingerprinting is discussed as a server side anti SPIT mechanism, it is useless against Direct IP Spitting as the clients don’t have any chance to verify the fingerprint of the attacking client.

In the end we will take a look on practical issues of Device Fingerprinting. When we take a look at today’s VoIP universe, we will find out that there exist a vast variety of hard- and soft phones. Each of these phones has its own SIP header layout and behavior and even within a product family header layouts and behavior differ even between two versions of the same device. The result is, that an administrator who uses Device Fingerprinting in order to protect his system, must always keep the list of fingerprints up to date. Otherwise it can lead to blocking of calls although the calls are not SPIT. Let us e.g. assume that a caller uses a standard client and that the manufacturer sends out a firmware upgrade, that makes major changes to the SIP Stack. Any calls of this user are blocked or marked as SPIT, until the administrator of the VoIP network updates the fingerprint list and this procedure will repeat any time a new firmware version is rolled out or new clients are released. Taking it even one step further, we can see, that as more and more clients and versions are released, the fingerprint list will become wider and wider and in the end nearly any combination of e.g. header fields will be present in the list. The main problem of device fingerprinting is that it is derived from a HTTP security technique. In that scenario only few clients (web browsers) from few developers exist, in contradiction to the VoIP world.

1.4.2 Weaknesses of White Lists, Black Lists, Grey Lists

Black Lists can not really be viewed as a SPIT countermeasure, because additional methods are needed to classify a caller a spitter. A Black List on server side would require e.g. statistical methods for classifying a caller as spitter. In case of a client side Black List, the user must mark a caller as a Spitter, e.g. after receiving an initial SPIT call from this caller. Both server side and client side Black List are very useless against Direct IP Spitting for different reasons. Server sided Black Lists are bypassed by Direct IP Spitting, because the SIP messages are sent directly to the client. Client sided Black Lists are circumvented by Direct IP Spitting, because the caller can take on any identity in order to place calls. So if one identity is blocked he can simply switch the Identity. We can call this attack **SIP Identity Spoofing** and any attacker, who can spoof SIP identities, can easily bypass Black Lists.

White Lists are at first sight harder to circumvent than Black Lists, because the attacker has no knowledge about the entries of the White List of the victim. So even if he wants to spoof an identity, the attacker doesn't know which identity he must take on, in order to place a successful call. In case of Direct IP Spitting the attacker could simply try out all existing accounts with a brute force attack until he finds out which identities are not blocked. A less exhausting procedure can be performed in case of distributed or imported white lists [2]. In that scenario the attacker needs one valid account. After adding the victim to the attacker's white list, he can now select that he wants to import the white list of the victim. So he can get access to all entries of the victim's white list and can spoof these identities e.g. in a Direct IP Spitting attack.

The Grey List mechanism can be bypassed the same way as White List mechanisms, as it just represents a mechanism that allows first time contact. All in all we can say, that any attacker who is able to perform SIP Identity Spoofing, can bypass Black Lists, White Lists and Grey Lists.

In the end we will take again a look at the practical side of the presented mechanisms. The concepts of Black, White and Grey Listing are derived from the Instant Messaging world, where it is a matter of course, that users first ask for permission, before they are added to another user's buddy list and only buddies can communicate with each other. When a user receives a communication request, he receives the profile of the other user containing e.g. nick name, email address, full name or even profile photo. On basis of this information, the user can decide and is able to decide, if he wants to accept messages in future from that party or not. Taken to the VoIP scenario this mechanism seems very impractical as the introduction problem has to be solved. Let us assume e.g. an employee of a bank wants to call one of his customers. In case of white listing the call can not be successfully routed to its target, as customers usually don't have the phone numbers of employees of their home bank listed in the White List. The decision basis for accepting or rejecting a call is simply the phone number that is sent by the caller. If the call is rejected at first (Grey listing) the callee must decide if he wants to accept future calls and he must base this decision on the phone number. We can easily see that this fact is very impractical.

1.4.3 Weakness of Reputation Systems

Reputation systems that are based on negative reputation can be bypassed in same way as Black Lists [1]. A user with a negative reputation can be viewed as globally blacklisted as his calls are blocked e.g. for any user (this depends on the policy that is used). Nevertheless an attacker that is black listed simply needs to gain access to a new ‘clean’ account. In case of a SPIT value as SIP header, the SPIT value can be spoofed by the attacker (e.g. with Direct IP Spitting) and we can call this attack **SIP Header Spoofing**. The attacker can simply set or change values of header fields, when he uses Direct IP Spitting. In addition an attacker can create several accounts with the aim of pushing the SPIT value of one account up or down (depending on implementation). This attack can be called Reputation Pushing or Pulling and is also referred as Ballot Stuffing [14].

Again we will also take a closer look at practical issues of the anti SPIT mechanism. At first we must admit, that Reputation systems are more auxiliary features than SPIT blocking mechanisms. The reason for this argumentation is that the user must classify a call as SPIT via a button or by entering a value. This value is used for future decisions on that SIP identity. So initially SPIT is not prevented by this technique. Then the SPIT value of an identity has to be shown to callees, so that they can decide about accepting or rejecting the call. Let us assume a Spitter has achieved a SPIT value or SPIT probability of e.g. thirty percent and then calls a victim. What should happen now? When the call is forwarded to the user and the value is e.g. shown in the display of the callee’s phone, he can decide to accept or reject the call on a better decision basis. The problem is that anyhow his phone rings and that is what should be prevented. He could have just picked up the call and listened the first 5 seconds to know that it is SPIT. So the SPIT value didn’t just add one percent of benefit. On top of this fact attackers could misuse the scoring system and create enough accounts in order to threaten ‘normal’ users with collectively giving them negative reputation [1].

1.4.4 Weakness of Turing tests and Computational Puzzles

Turing tests seem at first sight very effective for SPIT prevention in combination with white lists, but nonetheless have weak points. The first approach of bypassing Audio CAPTCHA is relaying the CAPTCHA to human solvers. An attacker could pay cheap workers, who are only hired to solve Audio CAPTCHA. In countries with cheap labor this would raise the costs per call only marginally [1]. In order to reduce the costs, an attacker could even e.g. set up an adult hotline and could dispatch Audio CAPTCHA to the customers of this service. This technique is known from visual CAPTCHA where the images from CAPTCHA protected sites are copied and relayed to a high traffic site owned by the attacker. All in all we can state, that an attacker who can detect CAPTCHA and relay it to human solvers is able to bypass Turing tests and we can call this attack **CAPTCHA Relay Attack**.

Computational Puzzles can not really be viewed as SPIT prevention mechanisms. It is obvious, that attackers usually possess high computational power. So circumventing a system protected by Computational Puzzles, doesn’t even demand a special attack. The attacker just needs sufficient CPU power.

In the end again we will take a look at some practical issues of the described techniques. As far as Turing tests are concerned, we can see that this method is very intrusive. User Interaction is forced every time a caller is not present in the White List of a callee. The difficulty with Computational Puzzles is that different VoIP endpoints have different abilities in computational power. So if the task is to hard to solve (consumes too much CPU power), session establishment will be delayed very much for e.g. a low-end cell phone, while attackers with high CPU power PCs won’t be concerned much. With this fact Computational Puzzles are very ineffective and contra productive, because they only bother ‘normal’ users.

1.4.5 Weakness of Payment at risk

In which way Payment at risk can be bypassed depends mainly on the way it is implemented. Demanding payment for each call won't be very realistic, because this would require a high administrative overhead and more costs for service providers. Let us assume Payment at risk combined with White listing, so that payment is only required for callers that are not present in the callee's White List. In this case a caller could simply spoof identity as described in the section about White List. In the second scenario, where Payment at risk is combined with a Reputation system, the attacker just needs to achieve an adequate reputation value, as described in the corresponding section. Let us even assume that Payment at Risk is used for every call. Even In that case an attacker could circumvent it, by impersonating as another user, so that he can establish calls and shift the costs on to 'normal' customers. In which way this kind of **SIP Identity Hijacking** [13] attack is fulfilled is another question and out of scope for now.

Besides the technical aspects, practical issues of Payment at Risk are numerous. At first the relative high costs, that are required for micropayment will must be viewed, the inequities in the value of currency between sender and recipient [1] and the additional interactions that a user must take (e.g. confirming a call from an unknown party as non SPIT).

1.4.6 Weakness of IDS, Honey phones

Intrusion Detection Systems base on the assumption, that the characteristics of attacks differ much from characteristics of normal calls. At first sight this assumption seems logic, as e.g. within a SPIT attack, the attacker calls hundreds or thousands of victims within an hour, while a normal user wouldn't even send out one percent of this amount of calls. Nevertheless the attacker has two possibilities in order to bypass detection by an Intrusion Detection System. The first is to align his behavior with the behavior of normal users, e.g. adjust the call rate to 5 calls per hour. Obviously this technique is hard to fulfill, because this would make an attack very inefficient as it would consume too much time, but on the other hand the goal of a spitter is not to reach as much users as possible within the shortest time period. Reaching e.g. thousand users with a call rate of 5 calls per hour would take approximately 8 days. We can call this technique **Call Rate Adaption**. This means that an attacker is able to adjust his call rate (e.g. number of calls per time slot, number of simultaneous calls). As the call rate is not the only variable that is used in order to detect abnormal behavior an attacker can use a second technique in order to not be detected by Intrusion Detection Systems. The attacker can use different accounts for his attacks, so that statistic values are spread over several accounts. Let us assume that an attacker has one hundred valid user accounts. With this amount of accounts he can partition the targeted user accounts into one hundred groups and use only one account per group. The users from group one are only called with account one and so on. It is harder for a monitoring system to detect attacks that are originated from different sources, as there must be a technique to correlate partial attacks to one complete attack. This attacking technique can be called **Account Switching**, as the attacker switches the used account while he is performing an attack.

Honey pots are very effective against scan attacks as anyone who tries to reach invalid or unassigned identities, will be trapped and so Honey pots are very effective against SPIT. When the Spitter can't scan the network for assigned and unassigned numbers, he is forced to view all numbers as assigned. When he views all numbers as assigned, he will sooner or later step into the trap, because he will establish calls to endpoints that are part of the Honey pot. Nevertheless attackers can trick the Honey pot mechanism with SIP Identity Hijacking. When an attacker impersonates the accounts of normal users and then performs SPIT attacks with these normal accounts, he will access end points in the Honey pot system with normal accounts. So the assumption that accesses to the Honey pot are only established by attackers is lapsed.

In the end we will take again a look at the practical issues of the presented solutions. The practical problem with intrusion detection systems in general is that they base on statistical assumptions that are not verified. The question that has to be solved is: Where is the borderline between normal usage and abnormal usage? The publishers state that statistical values are assumed or derived from attack characteristics, but in order to reduce the rate of false negative and false positive classifications, the knowledge basis must be precise. So we can say that what we lack is knowledge of SPIT characteristics as we nowadays can't really distinguish SPIT from normal traffic unless the SPIT attacks are excessive. Honey pots have the disadvantage, that they only detect access to invalid or unassigned accounts, this means that an attacker who only accesses valid accounts won't be handled by a Honey pot.

2 Conclusion

As a summary we can now say that we learned how SPIT is put into practice, why it is a threat, what mechanisms have already been developed against it and how these mechanisms can be bypassed. The following figure puts the puzzle together and shows how SXSM can be used in order to evaluate SPIT countermeasures:

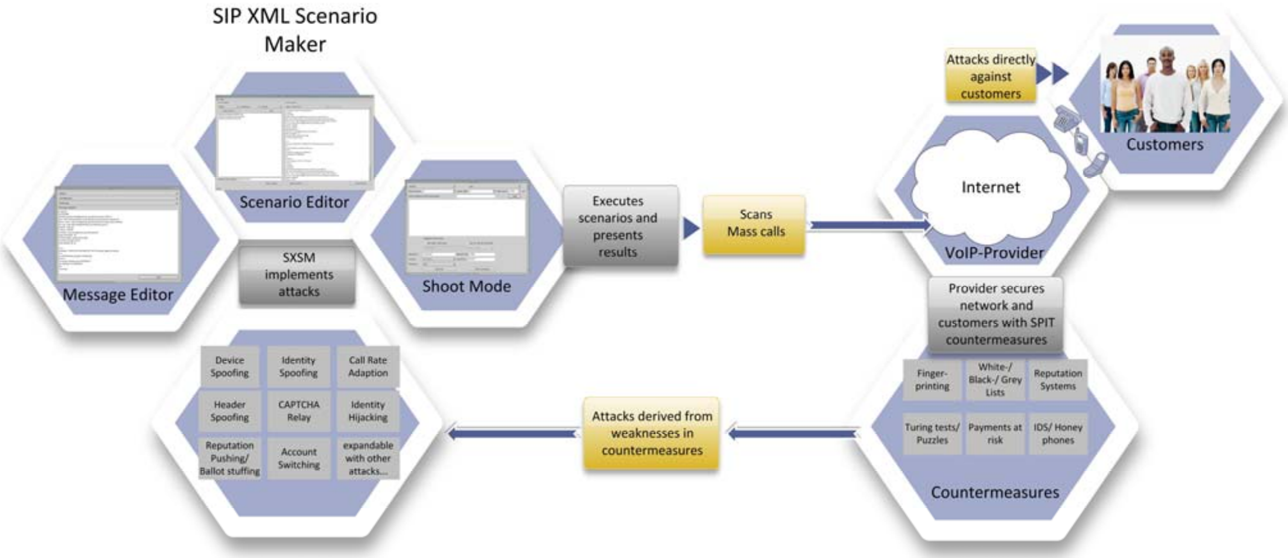

Figure 7: Holistic view on SXSM in SPIT context

The image shows that SXSM can be used by a VoIP network administrator in order to reveal weak points of the system or in order to test the robustness of used countermeasures, by figuring out weak points deriving attacks from these weak points and putting the attacks into practice.

References

[1] J. Rosenberg, C. Jennings, RFC 5039 - The Session Initiation Protocol (SIP) and Spam, IETF, 2008.

[2] M. Hansen, M. Hansen, J. Müller, T. Rohwer, C. Tolkmit and H. Waack, Developing a Legally Compliant Reachability Management System as a Countermeasure against SPIT, 2007.

[3] S. Dritsas, J. Mallios, M. Theoharidou, G.F. Marias and D. Gritzalis, Threat Analysis of the Session Initiation Protocol Regarding Spam, IEEE, 2007.

[4] H. Yany, K. Sripanidkulchaiz, H. Zhangy, Z. Shaez and D. Saha, Incorporating Active Fingerprinting into SPIT Prevention Systems, 2007.

[5] M. Stiemerling S. Niccolini, S. Tartarelli, Requirements and methods for SPIT identification using feedbacks in SIP. Internet-draft, 2008.

[6] F. Wang, Y. Mo, B. Huang, P2P-AVS: P2P Based Cooperative VoIP Spam Filtering, 2007.

[7] C. Jennings, Computational Puzzles for SPAM Reduction in SIP. Internetdraft, 2008.

[8] H. Tschofenig, E. Leppanen, S. Niccolini, M. Arumaithurai, Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) based Robot Challenges for SIP. Internet-draft, 2008.

[9] S. Liske, K. Rebensburg, B. Schnor, SPIT-Erkennung, -Bekanntgabe und -Abwehr in SIP-Netzwerken, 2007.

[10] M. Nassar, R. State, O. Festor, Intrusion detection mechanisms for VoIP applications, 2007.

[11] M. Nassar, S. Niccolini, R. State, T. Ewald, Holistic VoIP Intrusion Detection and Prevention System, 2008.

[12] SIPp at Sourceforge, http://sipp.sourceforge.net/index.html.

[13] Two attacks against VoIP, http://www.securityfocus.com/infocus/1862.

[14] Dellarocas, C., Immunizing Online Reputation Reporting Systems Against Unfair Ratings and Discriminatory Behavior, 2000

Keywords

SPIT, SPAM, VoIP Security, SPIT Countermeasures, SIPp, SIP XML Scenario Maker