Threats to Voice over IP Communications Systems

MIROSLAV VOZNAK, FILIP REZAC
Department of Telecommunications
VSB – Technical university University of Ostrava
17. listopadu 15/2172, 708 00 Ostrava
CZECH REPUBLIC
miroslav.voznak@vsb.cz, filip.rezac@vsb.cz
http://homel.vsb.cz/~voz29

Abstract: - This article deals with various techniques of VoIP attacks and VoIP communication security. These threats are divided in several categories according to their specific behaviour and their impact on the affected system. We describe effective methods to prevent or mitigate these attacks. Our work was especially focused on Spam over Internet Telephony (SPIT) as a real threat for the future and we consider spam in IP telephony to be very serious due to situation we face in e-mail communication. We have developed both an AntiSPIT tool defending communication systems against SPIT attacks and a tool generating SPIT attacks. AntiSPIT represents an effective protection based on statistical blacklist and works without participation of the called party which is significant advantage, AntiSPIT was implemented into Asterisk open-source platform and became one way of protection against the voice spam expected in near future.

Key-Words: - IP telephony, DoS, Security, Attack, AntiSPIT, Asterisk

1 Introduction

Voice over Internet Protocol is without any doubts a step ahead in communication technologies. But similarly to other technologies, there should be certain specific rules and security measures otherwise the technology does not work properly. Nowadays, SIP protocol is the most frequently used signaling protocol in IP telephony, however it is prone to attacks as described below. We can divide these attacks into several categories based on the threat behavior [1], [2]:

- Scanning and Enumerating a VoIP Network.
- Exploiting the VoIP Network.
- VoIP Session and Application Hacking.
- Social Threats.

The second chapter of this article provides a brief and concise description of the most frequently used attacks. We tried to focus on finding out which of the attacks offers the biggest potential as a future threat and if there is an effective defence against it. The third chapter deals with two tools, SPITFILE and AntiSPIT. While we developed the first one to demonstrate a way to implement and launch SPIT, the latter illustrates the implementation of protection against SPIT into Asterisk (open-source PBX) which is based on ideas of authors originating in the statistical blacklist method. The final section provides our conclusions and summarizes the acquired knowledge.

2 Possible Attacks and Protection

All telecommunication channels, including VoIP, are subject to attacks. These may include interception, modification, or even a data loss during transport. SIP protocol is in particular very susceptible to redirection, modification, or termination of the calls. Moreover in IP telephony, it is not possible to line up individual packets to front and analyze them by return because the communication happens in real time. This is the main difference between the proposed security mechanisms for classical Internet services and VoIP [3], [4].

2.1 Scanning and Enumerating

This is not a case of a true VoIP attack. However, if someone wants to attack the individual components of VoIP infrastructure, he has to locate them in the network and try to get as much information about these VoIP elements and infrastructure as possible. This is how scanning and enumerating tools work. If he is able to obtain information about individual devices then he can plan the next steps of attack.

VoIP infrastructure also comprises elements other than just VoIP phones and servers. Because the availability and security of VoIP networks relies so heavily on supporting infrastructure, an attacker could focus only on devices running VoIP services. It behoves him to identify and map out other core

network devices, including routers and VPN gateways, web servers, TFTP servers, DNS servers, DHCP servers, RADIUS servers, firewalls, intrusion prevention systems, and session border controllers to name a few. For instance, if an attacker was able to locate and disqualify TFTP server, several models of phones trying to download configuration files on bootup might crash or stall [5].

If we should mention specific tools which can be used for realization this type of attacks we can choose Nmap for network scan and SIPSCAN to get the list of SIP accounts from SIP server [6].

2.1.1 Protection Against Scanning and Enumerating Attacks

A well configured firewall or LAN divided into several virtual subnets (VLAN), with separate servers from the end devices, will guarantee at least a partial defence against automatic bots.

2.2 Exploiting the VoIP Network

This category includes attacks which enable obtaining detail information about the VoIP component, restrict its function, or eavesdrop calls/data that was made between the components. These types of attacks are the most widespread and include Denial of Service (DoS), or Man-in-the-Middle (MITM).

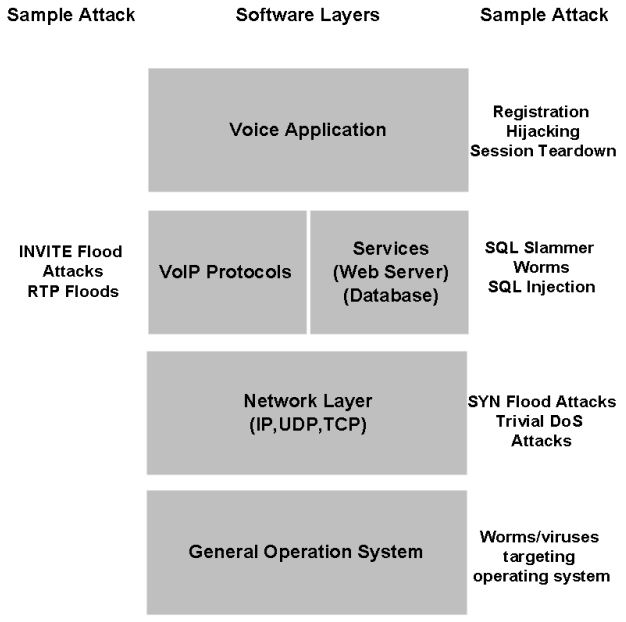

Figure 1. DoS attacks on different software layers.

2.2.1 Denial of Service (DoS)

DoS affects various types of IP networks services. Subscribers are not able to use their services properly or the function of services is limited and marked as unusable. First, we will describe the implementation of the DoS flaw. In this case, the attacker send modified packets, or sequence of packets, through a "flaw" in VoIP component implementation [7], [8].

These packets may be very long and syntactically incorrect which causes a VoIP component to fail as it is not implemented to handle such packets. Another example of an attack is Flood DoS. The primary feature of this attack is that the target component is flooded by many flood packets and then this component is not able to work with legitimate VoIP communication packets. Legitimate packets are either ignored or processed very slowly and the VoIP service becomes unusable.

One variant of this type of attack is that the flood packets cause the target system to allocate resources or consume processing power while waiting for a response that will never be sent. Another variant is a Distributed DoS. Multiple systems are used to generate a massive volume of packets so that the target systems are flooded. Another type of DoS is Application-level DoS which consist of manipulation and changes to VoIP service properties. For example, hijacking the registration data for an IP phone can cause a loss of any inbound call.

Based on the type of protocol (Signal or Transport) we can also determine which DoS attack will be applied. With the signal protocol, DoS attack can assault any part of the VoIP system because every VoIP component must process signalling. In case of a stateful SIP server the probability of successful DoS attack is higher than with a stateless SIP server because the stateful one requires a large volume of computation to be carried out by the server. As VoIP components very rarely use strong authentication, the probability that the attack will be successful is rather high. Besides, most firewalls do not prevent possible DoS at the level of signal protocols.

Real-time Transport Protocol (RTP) is the most frequently used transport protocol. In comparison with signal protocols, RTP is very simple and its implementation is almost without errors. That is why the DoS implementation flaw is not a good choice for possible attack. However, the RTP protocol is very vulnerable to flood attacks as the data are transmitted in real time and the attacker can flood any of VoIP components (IP phone, WAN link etc.). Udpflood or SiVus are for example the good choice for flood based DoS [9].

2.2.2 Man-in-the-Middle (MITM)

In this case, the attacker is located between two communicating parties and is trying to intercept and alter the data stream direction between the subscribers. This attack has a general designation – Man-in-the-middle, or MITM. MITM attack can capture data between participants, and also eavesdrop intercepted calls. Nowadays the most used MITM method is called ARP Poisoning. The method works with the fact, that some systems receive and save an ARP entry in its ARP cache, regardless on the previously sent or not sent ARP request. This means that an attacker can fool one or both subscribers to think that the attacker MAC address is the address of the other computer or SIP server. At this moment the attacker becomes a mediator between the subscribers and may eavesdrop or modify any communication between them. In practice, we can realize ARP Poisoning through five general points:

- Determine both of subscribers' MAC addresses.
- Sending an unsolicited ARP reply to subscriber A, so as to indicate that the MAC address of B was changed to the MAC address of the attacker.
- Sending an unsolicited ARP reply to subscriber B so as to indicate that the MAC address of A was changed to the MAC address of the attacker.
- Attacker allows IP forwarding on his computer, so data will pass freely between subscriber A and subscriber B.
- The attacker will capture and monitor data, or eavesdrop the calls.

The best tool to implement the ARP Poisoning is Cain and Abel [10], [11].

2.2.3 Protection Against DoS

System hardening: hardening is based on removing unnecessary or unusable network services, locking down the operating system and using an internal detection system.

- Strong authentication: VoIP components need to be sure that they are communicating with a legitimate subscriber. This ensures that all unauthorised attempts are recognized and aborted.
- Traditional firewall: FW can provide a very good protection against DoS. It mitigates the attack at individual levels. Firewalls provide a good security mainly in corporate networks where the VoIP is carried out over unsecured networks.

Firewall can provide the following protection against DoS attacks.

A. DoS implementation flaw

- Monitor for known VoIP attack patterns and discard packets accordingly.
- Monitor for new forms of attacks and discard packets accordingly.

B. Flood DoS

- Perform signalling and media rate limiting.
- When a DoS flood is detected, quickly discard obviously malicious packets.
- Allow packets from authenticated sources.
- Maintain adequate bandwidth for known, good calls.
- Monitor for and drop rogue RTP streams.
- Monitor for and drop rogue RTP packets targeting active RTP streams.
- Monitor for and drop illegitimate packets which carry high QoS markings.

C. Application-level DoS

- Monitor for and counter external attacks such as registration hijacking and illegal teardowns.

2.2.4 Protection Against MITM

Strict rules on the switch ports in the local network: manually entered MAC addresses and their authorization for each switch port guarantee a strong protection against possible attackers.

- LAN divided into VLAN subnets: provides an additional level of security against simple MITM attacks.
- Encryption: On the third layer of the OSI model, we can use an IPsec and VPN tunnels, on the fourth transport layer SRTP and ZRTP on the application layer. ZRTP has been designed mainly as a good protection against Man-in-the-Middle. In the case of SIP, we can use TLS encryption. Applications for ARP Poisoning detection – For Linux an arpwatch. This tool watches mapping MAC to IP addresses on the whole ARP protocol route. The equivalent for MS Windows is Xarp.

2.3 VoIP Session and Application Hacking

In the previous chapters, we covered several forms of attacks in which malformed packets or packet floods disrupted service for SIP proxies and SIP phones. In this chapter, we cover other attacks in

which an attacker manipulates SIP signaling or media to hijack or otherwise manipulate calls. As with other attacks we have covered, these attacks are simple to execute and rather lethal [12], [13].

2.3.1 Build-up Call Aborting

If the attacker wants to abort a built-up call, the SIP packet with SIP BYE method needs to be created. The client will receive this packet only if the call identification alias Call-ID is the same.

```
ACK sip:7204@158.196.146.12;transport=udp
SIP/2.0
Via: SIP/2.0/UDP
158.196.192.32:47998;branch=z9hG4bK-d8754z-
429c98f2694bf547-1---d8754z-;rport
Max-Forwards: 70
To: <sip:7204@158.196.146.12>;tag=as06fb1164
From:
"7002"<sip:7002@158.196.146.12>;tag=9197c599
Call-ID:
NzhkNmM2YzZhZDk3NjhmMDUwYTJjZWY5Z
WVkMzY4MWM.
CSeq: 1 ACK
Content-Length: 0
```

Figure 2. Original ACK packet

The best packet for this modification is an ACK packet. This packet confirms a build-up call from the caller to the called subscriber. We only need to change one element in the packet: the ACK method for BYE. If the modified packet is sent to the called subscriber's address, the phone will react as if the caller has ended the call. The transaction between subscribers is controlled by a branch parameter. A tag parameter which is randomly generated to identify a target can be the same as the original [14], [15].

```
BYE sip:7204@158.196.146.12 SIP/2.0
Via: SIP/2.0/UDP
158.196.192.32:47998;branch=z9hG4bK-d8754z-
f773bbf8da513d54-1---d8754z-;rport
Max-Forwards: 70
To: <sip:7204@158.196.146.12>;tag=as640fa018
From:
"7002"<sip:7002@158.196.146.12>;tag=9197c599
Call-ID:
NzhkNmM2YzZhZDk3NjhmMDUwYTJjZWY5Z
WVkMzY4MWM.
CSeq: 2 BYE
Content-Length: 0
```

Figure 3. Modified BYE packet

2.3.2 RTP Data Redirecting

We have to modify SDP packet content for RTP packet redirecting. This protocol defines the IP address and the port which is used during the call. If the attacker has modified these data, subscriber's VoIP client will send an audio stream to a different address and port which is a legitimate subscriber's address.

```
v=0
o=- 0 2 IN IP4 158.196.192.32
s=CounterPath Bria
c=IN IP4 158.196.192.32
t=0 0
m=audio 58940 RTP/AVP 0 8 3 101
a=sendrecv
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

Figure 4. Packet with INVITE from the client to the proxy server

```
v=0
o=- 0 2 IN IP4 195.113.113.147
s=CounterPath Bria
c=IN IP4 195.113.113.147
t=0 0
m=audio 58940 RTP/AVP 0 8 3 101
a=sendrecv
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

Figure 5. Changed packet from the proxy server

Changed IP addresses are in o and c parameter, port value is changed in m parameter. All attacks shown in this part have been realized using Scapy.

2.3.3 Protection Against Session and Application Hacking

The following security measures are used as a protection against this threat.

- SIP over TCP: If you use the TCP protocol instead of the standard UDP, SIP UAs to establish a permanent connection between themselves or with the SIP server. For the attacker in this case is difficult to handle registration because TCP uses sequence

numbers for each connection. Another advantage is, that we can use the TLS protocol which provides strong encryption and authentication.

- Enable authentification: Set up authentication for REGISTER and INVITE requests. For REGISTER messages we can use strong passwords, because these messages are not exchanged between the SIP server and SIP phone as often as INVITE messages.
- Reduction of the registration interval: The second proposed defence is to reduce the interval between registrations. By default, this interval is set to 3600 seconds. If we set the time for example on 60 seconds even if the registration will be removed or hijacked, after minute the phone will be active again.
- Ports change: change the standard port for SIP signaling protocol. It is default set to port 5060. This is a very weak but popular protection.

2.4 Social Threats

Under this term we can imagine the attacks that are not focused to obtain informations about the users accounts, eavesdrop the sensitive data or attack VoIP components in order to disrupt its function. The attacks described in this chapter are designed to make a calls with advertising content and from theperspective of the victims are very annoying. Such an attack is for example SPIT (Spam over Internet Telephony), which is similar to e-mail Spam.

The first type of SPIT is applied by the call centres through which advertisement calls are managed by agents. Subscribers who answer the phone are systematically or randomly searched by computer and if they answer the call, they are redirected to agents. Most advertising messages are recited by agents. The volume of such advertisement calls is in direct proportion to the number of call agents.

Another type of SPIT, so called ‘call bots’, uses the same principle as the call centres except that there are no call agents. The advertisement message is replayed automatically from a computer. There is no need to employ agents. Everything is administered by the computer. Certain VoIP phones may receive an “alert-info” SIP header. This header contains a internet link under which the advertising message is recorded. The message is played when the phone rings. When this type of attack occurs, the calls cannot be answered. The issue can be solved by turning off the internet ringing download on the telephone set. This attack is known as Ringtone SPIT. Most often, a combination of the above described attacks is used as it increases the chance that the SPIT attack will be successful.

In the following subsections we describe theoretical methods of protection against SPIT, how far are effective is a topic of wide discussion.

3.1 Buddylist/Whitelist

Every subscriber has a list of subscribers. Those who are not on the list cannot initiate a call. The problem arises when the subscribers that are not on the list are regular callers and we would like to speak to them. To allow the calls from subscribers who are not on the whitelist is helpful to access a ’web of trust’. Each subscriber grants his trust to several other subscribers.

3.2 Blacklist

This is a reversed whitelist. In this case, the subscriber has a list of possible SPIT attackers and these contacts do not have access to subscriber’s phone.

3.3 Statistical Blacklist

Telephone providers carry out different analyses and apply different statistical methods to create a spammer list. If a caller makes hundreds of phone calls during a short time span, s/he is very likely to be a spammer.

3.4 Voice menu interaction

Before the caller is actually put through to the called subscriber, the caller is directed to the voice menu where he is asked to enter a numeric code (for example 123*) to be able to get through to the caller. This protection is effective against caller bots (relatively until the bots take up using a speech recognition).

3.5 Greylist

This is a modified blacklist (whitelist) under which the phone returns the engaged line tone to the caller who is making the call for the first time. This time, the phone does not actually ring on the called subscriber. If the caller attempts to make the connection again the call is connected. It increases a likelihood that the caller is a human person and not a SPIT bot.

3.6 Law aspects

As telecommunication is protected by several laws, instances filtering mails or calls face several legal

consequences (for example imprisonment). Therefore, it is mandatory not only to construct technical filtering mechanisms, but also to consider implications from telecommunication or privacy protection laws and regulation.

4 Implementation of SPIT

If we want to deal with a protection against SPIT we need to develope SPIT and show how works, how danger is and how easily can be implemented.

4.1 Motivation

Attacks on Internet services become a very frequent issue in the global IP network. Another threat is coming with IP telephony, SPIT. If we take the fact that Spam takes up 80-90% of the total number of attacks on Internet services, the threat of SPIT in the future is very real and dangerous. Many security experts also share this view. We can say that the only reason why the SPIT has no global impact yet is that it, as opposed to Spam, poses greater requirements on computing power, hardware and network connectivity. But this barrier will soon be removed by increasing technological development. SPIT also has a much greater annoying effect and impact on the victim than Spam.

Just imagine, unlike an incoming emails with Spam, which you can easily ignore and delete, the new attack containing SPIT advertising message will ring again and again several times a day. That was the reason why we started developing and implementing our own tool. This tool is called SPITFILE and it is based on well-known SIP packet generator Sipp. Next subchapter deals with SPITFILE application which is programmed in Python and it should demonstrate how easy it is to generate this type of attack.

4.2 SPITFILE

During the development process, we put much emphasis on the simplicity of using and generating SPIT attacks. This decision was made on the ground of a study. We got acquainted with a few implementations of VoIP generators which were suitable for SPIT attack but they were mostly based on rather complex algorithms and demanded good knowledge of Linux-based systems. Therefore our aim was to design and then to implement a SPIT attack into an application which would be user-friendly and with an intuitive menu. We opted for Python to develop our SPIT application. Python is a high-level programming language similar to Perl or Tcl. The objective of designed application is to generate the phone calls and to replay a pre-recorded voice message.

We had to find a suitable SIP generator which we could modify. As an acceptable candidate was adopted an application SIPp which focuses on testing and simulating SIP calls in VoIP infrastructure. SIPp is a open-source test tool or traffic generator for the SIP protocol. SIPp can read custom XML scenario files describing from very simple to complex call flows and also send media traffic through RTP.

Two methods were used for dynamic cooperation with parameters inserted into SIPp and XML. The variables corresponds to appropriate parameters inserted by user and they are sent to SIPp application to initialize a voice call. For the values which have to be dynamically inserted into XML file, a new function was created, enabling to search and to change a particular value. We use a library xml.dom.minidom for handling XML files. Our application called SPITFILE implements a graphic interface for SIPp and works with ready-made .xml diagrams. Thus, the simulation of a SPIT attack is much simpler.

SPITFILE was programmed in Python using wxPython GUI. Its control is very intuitive – the requested values are submitted into relevant fields and the SPIT attack is launched by clicking the SEND button. For a proper operation of the SPITFILE application it is first necessary to install the following packages: Python ≥ v2.6 ,Sipp ≥ v2.1, Python-wxgtk ≥ v2.6. Our application can generate attacks in two modes: Direct and Proxy.

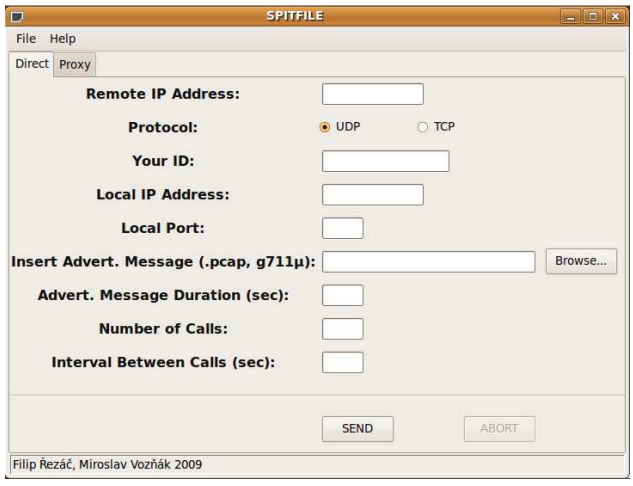

Figure 6. SPITFILE in Direct Mode

4.2.1 Direct Mode

It generates SPIT on IP phone directly in the local network without using the VoIP PBX (some IP phones can refuse a direct calls that avoid SIP Proxy, the Proxy mode is more suitable for such cases).

SPITFILE in Direct mode is depicted in Fig. 6. It is necessary to fill in mandatory information which is used to launch the prepared attack.

4.2.2 Proxy Mode

It generates SPIT via VoIP PBX (SIP Proxy) and the attack thereupon can run against anything that is available behind the Proxy, theoretically involving not only IP phones but also ordinary phones and the whole telephone world. As well as in Direct mode in Proxy mode on Fig. 7, it is also necessary to fill in compulsory information needed to create the SPIT attack.

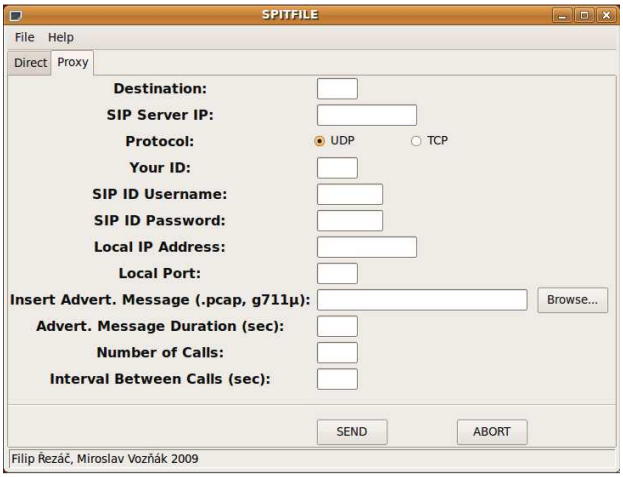

Figure 7. SPITFILE in Proxy Mode

In addition, it is necessary to obtain a user account because a successful registration at SIP Registrar is required before the calls via SIP Proxy can be performed. It is the main difference between Direct and Proxy modes. The attacker should obtain a valid account at SIP Proxy. There exist many ways how to obtain the username and password, for example applications Cain and Abel or SIPcrack. Both applications are a tool for sniffing and cracking the digest authentication which is used in the SIP protocol.

4.2.1 Concept and technology

A concept of SPIT attack, depicted in Fig. 8, can be performed in a few steps:

- The attacker fills in the SPITFILE application form.
- The inserted parameters are used in the chosen scenario generated in XML file for SIPp.
- SPITFILE creates a command consisting of defined parameters.
- SIPp is launched and the phone call is initialized by XML file.
- SIPp is automatically terminated after call disconnection but SPITFILE is running and is ready to launch next scheduled attack..

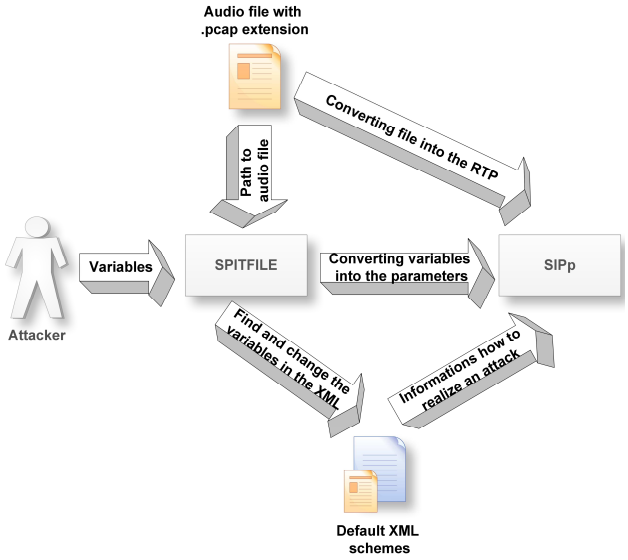

Figure 8. SPITFILE Draft

Before SPITFILE can be opened, preconfigured .xml diagrams (direct.xml and proxy.xml) should be imported into /etc/ directory. Afterwards we can launch SPITFILE and choose one of the two above mentioned attacks that we want to carry out.

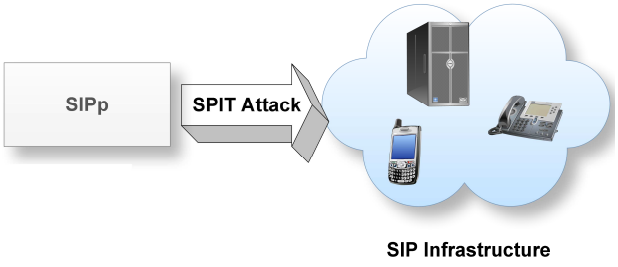

Figure 9. SPIT attack launched by SIPp

To run SPITFILE, just type the following command to the terminal python <location of the SPITFILE.py file>.

The called phone rings after the attack has been sent and a pre-recorded voice message will be played after the incoming call is answered.

SIPp provides information about passed/failed verdict, the situation is depicted in Fig. 10.

```
------------------------------ SCENARIO SCREEN -------------------------------

 CALL-RATE(LENGTH)      PORT    TOTAL-TIME  TOTAL-CALLS  REMOTE-HOST
  1.0(0 MS)/1.000S   5060       1.37 S            1  158.196.146.12:5060(UDP)

 CALL LIMIT REACHED (-M 1), 0.373 S PERIOD  7 MS SCHEDULER RESOLUTION
 0 CALLS (LIMIT 27)                    PEAK WAS 2 CALLS, AFTER 1 S
 0 RUNNING, 1 PAUSED, 0 WOKEN UP
 0 OUT-OF-CALL MSG (DISCARDED)
 1 OPEN SOCKETS
 0 TOTAL RTP PCKTS SENT                0.000 LAST PERIOD RTP RATE (KB/S)

                              MESSAGES  RETRANS   TIMEOUT   UNEXPECTED-MSG
   REGISTER ---------->          1         0
        100 <----------          1         0                   0
        401 <----------          1         0                   0
   REGISTER ---------->          1         0
        100 <----------          1         0                   0
        200 <----------  E-RTD1 1          0                   0

     INVITE ---------->          1         0         0
        180 <----------          0         0                   0
        100 <----------          0         0                   0
        407 <----------          1         0                   0
        ACK ---------->          1         0
     INVITE ---------->          1         0
        100 <----------          1         0                   1
        180 <----------          0         0                   0
        200 <----------          0         0                   0
        ACK ---------->          0         0
              [ NOP ]
      PAUSE [   9000MS]          0                             0
        BYE ---------->          0         0
        200 <----------          0         0                   0

------------------------------ TEST TERMINATED -------------------------------
```

Figure 10. SIPp providing information on action.

A part of XML code is depicted in picture 11 , Sipp application sends INVITE request in the prepared form. The parsing function in SPITFILE replaces 2717 value with a variable from destination field which was submitted in SPITFILE application form. Thereafter Sipp waits for a response. If 100 Trying or 180 Ringing is received than the pointer in schema jumps to label 1 or label 2, it is responding to to a special case, where INVITE is answered without the need of authentication.

If authentization is requested than INVITE is answered with 407 and Sipp confirms this response with ACK. Consequetnly the authenticated INVITE is sent but the situation is not depicted in this part of XML code.

```
- <send retrans="500">
- <![CDATA[
INVITE sip:2717@[remote_ip] SIP/2.0
    Via: SIP/2.0/[transport] [local_ip]:
[local_port];branch=[branch]
    From: [service]
<sip:[service]@[remote_ip]>;tag=[call_number]
    To: 2717 <sip:2717@[remote_ip]>
    Call-ID: d///[call_id]
    CSeq: 3 INVITE
    User-Agent: Grandstream GXP2000 1.1.6.37
    Contact:
<sip:[service]@[local_ip]:[local_port];transport=[tra
nsport]>
    Max-Forwards: 70
    Allow:
INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OP
TIONS,INFO,SUBSCRIBE,UPDATE,PRACK,ME
SSAGE
    Supported: replaces, timer, path
    Subject: Performance Test
    Content-Type: application/sdp
    Content-Length: [len]

    v=0
    o=[service] 8000 8001 IN IP[local_ip_type]
[local_ip]
    s=SIP Call
    c=IN IP[media_ip_type] [media_ip]
    t=0 0
    m=audio [auto_media_port] RTP/AVP 0
    a=rtpmap:0 PCMU/8000
    a=ptime:20
    a=sendrecv

 ]]>
 </send>
 <recv next="2" optional="true" response="180" />
 <recv next="1" optional="true" response="100" />
 <recv auth="true" response="407" />
- <send>
- <![CDATA[

    ACK sip:2717@[remote_ip] SIP/2.0
    Via: SIP/2.0/[transport]
[local_ip]:[local_port];branch=[branch]
    From: [service]
<sip:[service]@[remote_ip]>;tag=[call_number]
    To: 2717
<sip:2717@[remote_ip]:[remote_port]>[peer_tag_p
aram]
    Call-ID: d///[call_id]
    CSeq: 3 ACK
    Contact: sip:[service]@[local_ip]:[local_port]
    Max-Forwards: 70
    Subject: Performance Test
    Content-Length: 0

 ]]>
 </send>
```

Figure 11. A part of XML code

SPITFILE has been tested with HW Grandstream IP phones (GXP 2000, 3000) and with SW IP phones (Sjphone and X-Lite). The Proxy mode has additional fields such as the required account which is consequently used for registration, such as SIP number, username and password. The other fields are the same as in the case of previous Direct type.

We have tested Asterisk PBX and Siemens hipath4000 PBX.

5 AntiSPIT

In chapter 3 we described several theoretical methods how to defend against SPIT, everyone can assess how useful they are and whether they are suitable for a practical implementation.

Irrespective of the types of defence mentioned in the chapter 3,we tried to design and create our own model of security application based on Blacklist which would provide an efficient defence against SPIT, the name AntiSPIT [4] has been given to the new application. AntiSPIT is able to analyse and process input data from Call Detail Records (CDR's) and consequently determine whether the used source will be inserted into blacklist. CDR's are an integral part of every PBX and we decided to implement AntiSPIT into Asterisk PBX. The application gives an output which is inserted as a command which can control the blacklist. Asterisk provides CLI interface enabling us to create or delete the particular records in the blacklist database.

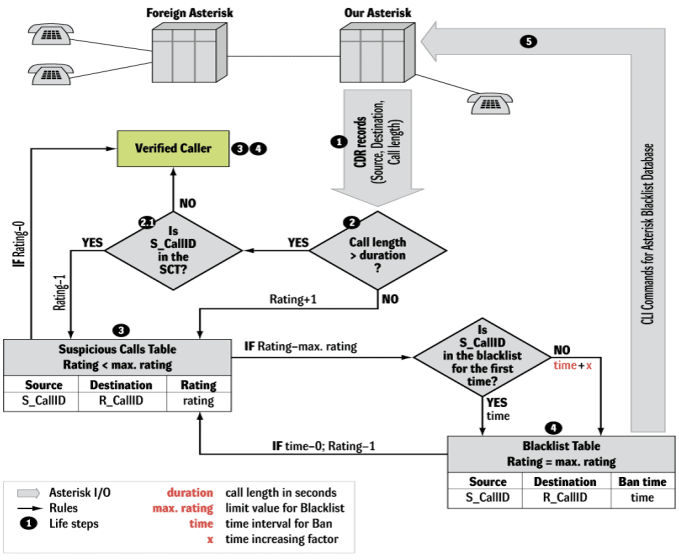

Figure 12. AntiSPIT Concept

The whole concept acts as an effective tool and provides a SPIT recognition and consequently a protection against SPIT attacks. In principle, the main idea of this proposed method is based on the human behaviour in case of an unwanted call. The called party terminates the unwanted call early after the SPIT call is answered and the reaction gets quicker when it reoccurs.

The called party terminates such repeating SPIT call in a substantially shorter time interval than in the previous case. This concept is shown in Fig. 12.

In this case, we can monitor the call duration from CDR's where every call has stored a record and if the call duration is less than a certain interval (duration), the source of the calls will receive the status of suspicious caller and a record with rating is created. In the case of repeated suspicious behaviour the rating will be increased. The record includes sender's called ID (S_CallID), receivers called ID (R_CallID), rating and will be listed in the table named Suspicious Calls Table (hereafter SCT). The maximum achieved rating factor represents a threshold limit value that makes a decision about whether the record from SCT is put onto Blacklist table (BLT). If this record is transferred into BLT then a caller stated in such a log can not carry out inbound calls for a specific time interval (time). The BLT record contains sender's called ID (S_CallID), receivers called ID (R_CallID), and time interval determining the ban (time). After expiration of the ban time the rating of the record is reduced by one and is transferred back to SCT.

However, if the inbound call is repeatedly marked as a suspicious and the threshold rating factor value is exceeded it will be put back onto the Blacklist table, this time for a longer period of the ban time (time + x).

At the same time, as the record is inserted into the Blacklist table database put blacklist <number> command is generated for PBX Asterisk CLI. After the ban time expiration the record is returned to SCT and database del blacklist <number> command is sent Asterisk CLI, the Blacklist table can be modified manually if necessary, depicted in fig. 13.

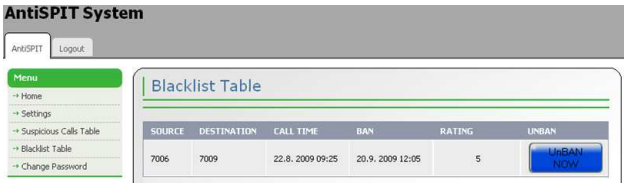

Figure 13. Blacklist Table

Callers who have a record in the SCT can also reduce their rating value and it is also possibly to fully remove them from SCT. Once the caller carries out a call with a longer duration than the set threshold limit (duration) and his S_CallID is a part of the record in the SCT, then the suspicion mark is consequently reduced by 1. If the caller does not have a record in the SCT then he is a certified subscriber and no action is required. The process is denoted in Fig. 12.

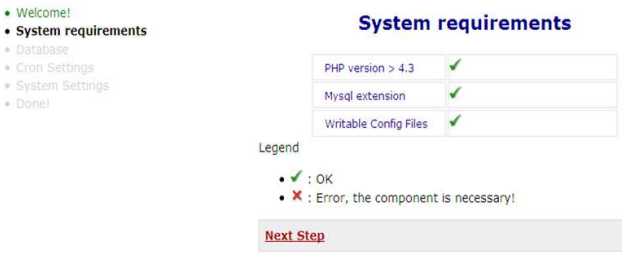

Figure 14. Installation process

The installation process is user friendly and eventual dependecies are solved during step of system requirements checking. Afterwards, the system parameters should be set up, if the prearranged values do not fit to particular situation, fig. 15.

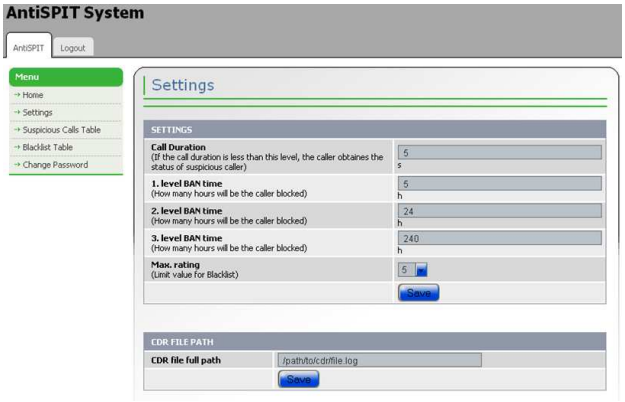

Figure 15. Settings Menu

AntiSPIT has been created in LAMP environment – meaning Linux, Apache web server, MySQL databases and PHP. AntiSPIT offers user-friendly administration through a web front-end enabling a user to set the key parameters such as length of call interval (duration), maximum achieved rating factor (max rating), ban time (time).

The web front-end also enables monitoring and the management of both SCT table and BLT table.The AntiSPIT can be downloaded and freely distributed under the GPL.

6 Conclusion

This article dealt with VoIP attacks and especially SPIT. We divided threats into several categories and mentioned the specific techniques on how to apply these attacks in real-case. Our goal was to determine which of the attacks are currently the biggest threat for VoIP networks and analyse potential risks and defences against them. For this purpose, we developed a SPITFILE application and AntiSPIT. Some of the described security measures in chapter three or combination thereof should be implemented in every IP ready PBX. This should enhance the protection against SPIT attacks. However, as effective SPIT attack methods are being developed very fast, further intensive research is needed to guarantee the security of VoIP systems. Fortunately, most telephone calls are charged which functions as a brake but we cannot not rely on it. SPIT is a threat hanging over the telephony world like the sword of Damocles.

We also made a model of an effective defence against SPIT which we implemented into software IP PBX Asterisk and it has been given a name AntiSPIT. AntiSPIT is based on a call rating factor and blacklist table, together it provides a good protection against SPIT. We hope that applications such as AntiSPIT will help us to define and develop the idea how to defend against the new SPIT attacks and how to break this imaginary sword.

Acknowledgement

This research has been supported by the “Optical Network of National Research and Its New Applications” (MSM 6383917201) research intent of the Ministry of Education of the Czech Republic.

References:

[1] D. Endler, M. Collier, M., *Hacking Exposed VoIP*, McGraw-Hill Osborne Media, 539 p., 1st edition, November 2006, ISBN 978-0-07-226364-0

[2] M. Collier, *VoIP Denial of Service (DoS)*, White paper, SecureLogix Corp., May 2005.

[3] F. Rezac, M. Voznak, J. Ruzicka, Security Risks in IP Telephony, *CESNET Conference 2008 Security, Middleware and Virtualization*, September 2008 , Prague, ISBN 978-80-904173-0-4.

[4] M. Voznak, F. Rezac, The implementation of SPAM over Internet telephony and a defence against this attack, *Telecommunications and Signal Processing (TSP) 2009*, Dunakiliti, Hungary, August 2009, ISBN 978-963-06-7716-5.

[5] V. Novotny, D. Komosny, Large- Scale RTCP Feedback Optimization, *Journal of Networks*, 2008, Volume 3, pp. 1-10, ISSN 1796- 2056.

[6] A. Sisalem, J. Floroiu, *SIP Security*, JWS, Inc., 350p., 2009, ISBN 978-0-470-51636-2.

[7] M. Kumar, M. Hemalatha, P. Nagaraj, S. Karthikeyan, A New Way Towards Security in TCP/IP Protocol, *14th WSEAS International Conference on COMPUTERS*, p. 46-50, Corfu Island, Greece, July 23-25, 2010, ISBN 978-960-474-201-1.

[8] J. Asim, U. Shafique, Network Risk Management, *4th International Conference on Communications and Information*

Technology, p. 141-145, Corfu Island, Greece, July 22-25, 2010, ISBN 978-960-474-207-3.
[9] A. Konheim, Computer Security and cryptography, JWS, Inc. 521p, 2007, ISBN 978-0-471-94783-7.
[10] S. Vincent, J. Raja, A Survey of IP Traceback Mechanisms to overcome Denial-of-Service Attacks, *Proceedings of the 12th International Conference on NETWORKING,VLSI and SIGNAL PROCESSING*, p 93-98, University of Cambridge, UK, February 20-22, 2010 ISBN 978-960-474-162-5.
[11] V. Patriciu, A. Furtuna, Guide for Designing Cyber Security Exercises, *Proceedings of the 8th WSEAS International Conference on INFORMATION SECURITY and PRIVACY*, p. 172-177, Puerto De La Cruz, Tenerife, Canary Islands, Spain, December 14-16, 2009, ISBN 978-960-474-143-4.
[12] M. Voznak, J. Rozhon, SIP Infrastructure Performance Testing, *In Proceedings of the 9th WSEAS International Conference on TELECOMMUNICATIONS and INFORMATICS* , Catania, Italy, May 29-31, 2010, ISBN 978-954-92600-2-1.
[13] M. Voznak, Speech Bandwith Requirements in IPsec and TLS Environment, *13th WSEAS International Conference on Computers*, p.217-220. Rodos, July, 2009, ISBN 978-960-474-099-4.
[12] J. Ransome, J. Rittinghouse, *VoIP Security*, Elsevier, 402p., 2005, ISBN 1-55558-332-6.
[14] T. Porter, *Practical VoIP Security*, Syngress Publishing, Inc., 563p, 2006, ISBN 1-59749-060-1
T. Wallinford, VoIP Hacks, OReilly Media,Inc., 306p, 2006, ISBN 0-596-10133-3.