SECURITY EVALUATION OF MULTIMEDIA SYSTEMS
Jakub SAFARIK, Miroslav VOZNAK, Filip REZAC
safarik@cesnet.cz, voznak@cesnet.cz, filip@cesnet.cz

Research paper

Keywords: penetration tests, VoIP attacks, SIP server defense, IPS, countermeasures

Abstract:
High popularity of VoIP in last few years leads to higher concern of hackers. Many solutions of VoIP servers are created, but security was not always the top feature of these solutions. IP telephony infrastructure provides also other services, which are not related only to VoIP traffic. All these factors lead to a situation, when a VoIP server easily becomes a target of attacks.
A wide spectrum of attacks can be done against VoIP, starting from misuse attacks, congestion of existing infrastructure to a completely (or partially) denial of service. Very popular are also SPIT attacks – an analogy to spam e-mail messages. Typical SPIT attacks are unwanted, automatically dialed prerecorded calls with commercial content.
This paper deals with the need of application for automatic testing VoIP server robustness against different types of attacks. This application was developed on VSB-TUO Ostrava as a testing tool to verify if the target VoIP PBX is adequately secured and protected against any real threats. The system tests the SIP element for several usually occurring attacks and compiles evaluation of its overall security based on successful or unsuccessful penetrations. The article describes the application and algorithms that are used by system. 
Another way to improving VoIP server security is an automatic defense mechanism, which blocks incoming attacks – typically various Denial-of-Service (DoS) attacks. This proposed solution is based on Snort and SnortSam and has been implemented and evaluated in a test bed. Different DoS attack types are described in detail and knowledge is used to test the robustness of the VoIP server. These two applications give us an effective way for complex testing of VoIP infrastructure key elements and to effective blocking of unwanted, malicious traffic.

1 Introduction
This paper deals with an increasing need of securing a VoIP infrastructure. There are systems designed to test and monitor networks like Nessus, Retina, Snort, OpenVAS and so on. These applications allow testing whole network infrastructures or various end-point devices. None of these solutions is designed to complex testing of a VoIP infrastructure with SIP servers, which are the most vulnerable parts of the network.
So we developed an application for automatic testing of a VoIP server under a working title SPT (SIP Penetration Testing). The person who initiated the testing (the tester) can easily test his VoIP infrastructure and receives a form of test results as feedback. The feedback also contains a recommendation on how to mitigate discovered potential threats.

![Figure 1: SPT system schemes](image-url)
The one of advantages of this solution is that the system simulates a real attack from the external network. It is not necessary to run a test in a same network as a target component DUT (Device under test). This behavior leads to a situation very similar as a real attack. A typical testing tool needs to be placed in the target network. The SPT system is...
a web application accessible through a web browser and, therefore, independent on the tester’s operation system platform. For prevent the system of being used for other than testing purposes, authentication using SSO (Single Sign-On) service – Shibboleth was implemented [1]. Whole SPT concept illustrates a Figure 1.

On the other hand, the knowledge about security gaps is not the only way to improve VoIP server security. We are also running an automatic defense mechanism – IPS (Intrusion Prevention System). Using the IPS we can block one of the most used attack nowadays – DoS (Denial of Service), independently on a used VoIP PBX. On top of used attacks is DoS because of its high efficiency, relatively easy feasibility, and VoIP server fragility against it. We decided to perform all experiments in a real used SIP server platform based on a solution Asterisk. The protection mechanism consists of three applications – Snort, SnortSam and IPtables. The following chapter describes DoS attacks in detail.

2 Classification of DoS attacks

DoS attack aim to run the target server out of its resource capacity. It means flooding a server with malformed, damaged or useless packets. The affected server is then unable to serve regular users and their requests. Security threats such as DoS almost do not affect a previous generation PSTN (Public switched telephone network) networks. This is due to their closed network topology originally designed only to transfer voice information [2]. This situation is changing with rising numbers of VoIP implementation and users expect the same behavior from the new technology. DoS attacks can be divided into three general classes [3, 4].

- Flooding attacks – targeting on server resources (CPU, memory or link capacity).
- Misuse attacks – the hacker use a modified SIP message to cancel or redirect calls or misuses the service. These attacks typically affect a small group of users only.
- Unintentional attacks – the attacker targets the supporting services (DNS, call billing, etc.) in order to distort or restrict the service.

In recent years became DoS attacks a crucial issue because of their increasing frequency, impact and complexity. Another significant fact is to distinguish between intentional and unintentional attacks [5]. Unintentional attack is crowd frenzy, for example, when a high number of users are trying to communicate, and servers cannot withstand such a high load. This state obviously passes quickly. Intentional attack can be classified as following.

2.1 Depletion attacks

Each server must store small chunks of information after receiving a SIP message. The server store this information for some time, depending on the server mode, which can be stateful or stateless. While performing a transaction, information is kept in memory. Typical attack from this class is TCP SYN flood attack. The server is flooded with TCP SYN packets and server response back to an attacker with SYN+ACK message. The attacker keeps on sending new SYN packets and ignores packets incoming from server. On the side of the server is quickly depleted all memory resources for a new TCP connection, and regular request are being refused.

Another kind of these attacks is sending highly fragmented packets to server with some parts intentionally omitted. The server attempts to request the missing parts and store the received packets in memory. This useless information is the stored on the server until it is timed out.

More efficient way to limit server’s ability effectively serve regular request is CPU depletion. It is achieved because messages are analyzed after the server receives them. Even when server is capable to process hundreds of messages we can easily force him to perform another calculation using malformed messages. Sending malformed REGISTER messages with bogus, invalid data or sending them from non-existing account has the same effect as has flooding server with a much higher number of ICMP packets.

Enabled authentication on the server is paradoxically a trigger off more challenging operations that only makes easier to deplete server’s CPU resources.

Last type of depletion attacks is called bandwidth depletion. It consumes the capacity of a link connecting the server to a network. When link is not able to transfer such a high load, regular packets are discarded before reaching the SIP server. Attackers use the stateless UDP protocol with the maximum packet size for this kind of attack.

2.2 Misuse attacks

An attack of this type needs only a small number of packets to achieve DoS. They use weaknesses of the target to their benefit. We can divide them into three subgroups: attacks against operating system, TCP/IP stack implementation and against SIP protocol. Below we describe attacks using the SIP protocol.

In case of affecting SIP communication must be hacker able to capture network traffic, modify SIP messages or disguise himself as another user. Misuse attacks also do not affect whole network, but only small part of it. From a provider’s point of view is this attack much more dangerous, because of reason mentioned above. Representative of this attack is BYE and CANCEL attack. In both cases is one of the parties convinced that the call was terminated. The attacker uses data from captured SIP headers to create malicious BYE (CANCEL) message. The attack with CANCEL message only termite the calls before they are connected. The only protection against these kinds of attacks is to ensure an encrypted transfer of SIP messages.
3 SPT Modules, platforms, algorithms and their time evaluation

The SPT system was designed as a LAMP (Linux, Apache, MySQL and PHP) server [6], and its complete administration including the installation is carried out via a web interface. The system itself is primarily designed for penetration test on SIP servers, but it can perform the full-scale attack on a particular component. Running SPT as a web application ensures that a third party cannot abuse the developed system. Only authorized persons can use the SPT once they pass through an authentication. The tester may use anyone of offered modules, which are described below.

3.1 Scanning and Monitoring Module

Almost every potential attacker starts with scanning a target network to gain as much information as possible. For testing security of SIP server against these types of attacks was developed a Scanning and Monitoring module. Common tools for scanning are, for example, Nmap [7] and, in case of SIP scanning, a popular tool SIPVicious [8]. The SPT system uses both these applications.

As a result, the tester gain from this module a list of listening ports and a list of user accounts created on SIP server. If the server’s security is not configured properly, it is even possible to get passwords for individual accounts. The tester can run this test with default settings or set own values of scanned port and extensions range.

Testing itself start with Nmap application looking for opened ports. The test is by default restricted only to a several most frequently used ports given the time requirements. The maximum time set for testing using Nmap $T_n$ is 1800 seconds. Test looking for user’s accounts list starts after that. SIPVicious uses an OPTION and ACK requests and by default tries the 100-999 range of accounts. It is possible to import a text file with alphanumeric strings $E_{dp}$ or numbers $E_{wp}$. Time required to check and create a list of accounts can be expressed by equation (1) where the constant $c = 0.02603$ (obtained by repetitive measurements) [3].

$$T_n = (E_{wp} + E_{dp}) \cdot c.$$  

(1)

Once the system has tested existence of defined accounts, possibility to detect passwords of individual accounts is also tested. The passwords are guested using a pre-defined range of password $(P_{nw})$ or from a given text file $(P_{dw})$. Required time of this test is expressed by the following Eq. (2). $E_{valid}$ is the number of valid SIP accounts.

$$T_p = \left[E_{valid} \cdot (P_{nw} + P_{dw})\right] c.$$  

(2)

The estimated time needed for all these tests is a sum of each one. Server’s security against scanning and monitoring can be now simply checked using this module.

3.2 Denial of Service Module

As was mentioned before, DoS attack is one of the most frequently used ones. In reality, there are several types of DoS. These types were described above. The DoS module only floods the target device with large volume of INVITE SIP messages or simply with UDP packets. To generate these floods SPT uses udpflood and inviteflood applications. In case of udpflood is used 1400 bytes UDP packet, which is sent at default SIP port 5060 on the target server. Using udpflood, attacker flood a link with useless packets, causing dropping of regular packets. The tester defines the number of packets send to server. An estimated time $T_{udp}$ for this attack we can get from generated packet count $P_n$, an Ethernet frame size $F_{udp}$ and link bandwidth $B_w$ (3).

$$T_{udp} = \left(F_{udp} \cdot P_n\right) B_w.$$  

(3)

When an inviteflood is used for testing, system generates INVITE SIP request, which are directed on an existing account. The SIP server will typically need an authentication, so it answers back with 407 Proxy Authentication Required. With a large number of messages incoming on a server the CPU load increases. The tester can set the value of target account or let it be chosen from previously obtained list. The equation for a time needed for attack is similar as (5) with proper values for inviteflood.

3.3 Registration Manipulation Module

When a potential intruder gains access to information about existing account, it can use this information for manipulation. The SPT module has also implemented this functionality. To carry out this test, the system uses reghijacker [9], substituting a legitimate account registration with another. It is possible to expand this attack easily to MITM (Man-in-the-Middle) attack [10].

Therefore, this module substituted a non-existent user with a valid SIP registration. All communication – signaling and media for a legitimate user will be redirected to the substituted account. The tester needs to define values of SIP account and password of the target account. The system can assign these values automatically from list created by scanning and monitoring module.

3.4 SPIT module

Spam is one of unwanted traffic in a current internet. SPIT (Spam over IP telephony) is very similar to spam, but it uses recorded voice messages instead of text. Security experts predict SPIT as a major threat in the future of IP telephony. The level of annoyance is in case of SPIT even greater than with SPAM. We developed an application SPITFILE [11] for such type of attacks. Together with Sipp [11] we can simulate a SPIT attack on target SIP server.

The tester only need to fill a value with valid SIP account (on which will be SPIT directed) and then other values for an
account which will initiate the SPIT call itself. The system can choose these values from the list of SIP accounts that were created while scanning and monitoring the central. In case of a successful attack, a SIP call is initiated, and pre-recorded message is played after the call is answered. The time required for this test \( T_{spit} \) depends mainly on a pre-recorded message length. The tester is informed about the success rate of the SPI module’s test.

![Diagram of SPT and its modules](image)

Figure 2: SPT and its modules.

Figure 2 shows the division of the SPT into its individual models with time requirements for each module. This time requirement varies with a configuration of modules from the tester.

4 IPS technology used

There are many ways of securing SIP server against DoS and other threats. Within an IPS (Intrusion prevention system), we have a solid tool for detecting and blocking malicious activity in a network. We also choose to run SIP server (Asterisk application) on an embedded device due to its features, performance and abilities. This choice made attacks on the server even more dangerous. The server runs a debian-based server’s linux and parameters were as follows:

- Single-core CPU at 2.2 GHz
- 512 MB RAM
- 4 GB HDD

The protection mechanism should be a part of this solution. So we choose an IPS build on three applications. Core of the IPS is an IDS (Intrusion detection system) open-source application Snort that detects malicious activity in the network [9]. The detection is based on signatures or recognition of anomalies. IPS functionality gives us another application SnortSam. It operates on a client-server basis and allows Snort to intervene dynamically into firewall, which is in our case IPTables. SnortSam can run remotely on any place in a network topology, but we try to invent a self-defensing mechanism for the server itself. Although this prerequisite brings us some limitation, it is still an adequate solution for the last line of defense. When we put the server in non-secure network, it would block some of the attacks. In secure network it further improves the existing security policies.

The main limitation of the proposed technology is caused by Snort IDS. Snort can only block traffic corresponding to a signature for detecting. Another limitation is in delays between attack recognition and its suppression. Minimizing this delay is also one of relevant facts. The delay is caused by processing all packets arriving in a given threshold, encryption/decryption a notification by SnortSam, and lastly applying a rule in firewall configuration.

5 Practical testing

Both IPS and SPT system were tested in our test bed. Although the SPT is still in phase of development, each module is used, and test topology is shown in fig. 3.
5.1 SPT modules test

The practical test of SPT was same as running SPT in a real environment. The scanning and monitoring module was set to find all extensions from range 1000-9999. The device found all three registered accounts with open ports on SIP server. SPT also successfully found passwords (Figure 4) from our given text file. Total time needed for first module was 235 seconds.

In DoS module sent the tester 500000 UDP packets to the SIP server. Invite flood was set to 100000 INVITE messages. Both attacks blocks communication with SIP server successfully. SPT recognizes successful attack with ICMP protocol. Before the test start SPT run ICMP to measure an average response time from server $T_{avg}$. During the test is measured time $T_{dosa}$ and if it is 150 times greater than $T_{avg}$, test is considered as successful (Fig. 5).
For testing of the possibility for registration hijacking was used information of valid account 7003 and its password 7003ab. Successfulness of this module is tested by sending an INVITE message to the test machine. The attack aim to de-register account 7003 and tries to direct all communication on non-existing account. Based on given response, system mark test as failed (with response 180 Ringing) or as successful (answer 503 Service Unavailable). A diagram of the test scenario illustrates figure 6.

The last tested is SPIT module. The tester simply enters values of valid account 7002 (victim) and account 7003 (originator) with its password. SPITFILE then tries to register to SIP server as account 7003 and initiate a call to 7002. The system itself detects successfulness by monitoring responses from SIP server. SPT perform five consecutive SPIT calls and all responses from SIP server must be 180 Ringing (see Fig.7). If even one of these answers is other, whole test is considered as not successful.
5.2 IPS results

As in SPT case we also made a testing topology for measure DoS effectiveness against SIP server. The topology consists of SIP server, some end-point devices and an attacker’s computer. Many tools were applied by tester at hacker’s computer. Most important was Sipp (in repository named as sip-tester) for simulating calls on SIP server. With our own call scenario can be also used as attacking tool. We used flooding tools like inviteflood, udpflood, flood2 and juno. These are not accessible from the repository, but source code is available for downloading on the internet. Tested were application for network scanning – hping, fping, nmap, SipVicious [13, 14].

All of these applications mentioned before we use against our testing SIP proxy. All attacks follow the same attacking scenario. Sending malicious packets started after 10s and continued for 60s. Last 30 seconds show recovering of server after attack. Graph on figure 8 illustrates the impact of flooding SIP proxy with different SIP messages.

5.2.1 CPU deletion attacks

Sipp application is typically used to carry out stress tests of SIP proxy [8]. Our scenario allows us to flood the target SIP server with different types of SIP messages. There is also inviteflood application, which sends only INVITE messages. With inviteflood, we can change only count of INVITE messages sent to the server and not the rate of packets per second. So much useful is Sipp application. For acquiring efficiency of different types SIP messages, we use a same sending rate for each attack. It was set to 250 messages per second. Efficiency of SIP messages shows figure 8.

![Figure 8: The impact of different SIP messages on SIP server’s CPU load](image)

In figure 8 we omitted attacks with CANCEL and ACK messages because its impact was very similar as an attack with BYE massage. It is clear that the biggest threat is flooding the server with REGISTER and OPTIONS messages. INVITE message flooding is also critical but only with higher rates. Even time required for a recovery of server after the attack was longer in case of REGISTER and OPTION messages. For reaching equal CPU load on the server as with REGISTER message, we must use a ten times bigger load with INVITE messages. That is because SIP server does not have enough information in a register message, so it tries to find or guess missing data, which consumes more CPU resources. Flooding with OPTIONS messages (which have a structure similar to INVITE message) consumes the more resources the longer an attack last. Performing attacks, with BYE, CANCEL and ACK messages, have a small impact on server. With increasing rate was attack more and more like flooding the server with udp packets.

Against all flooding attack was created an appropriate rule. IPS system successfully suppressed all attacks as shows figure 9 on example with REGISTER flooding.
5.2.2 Link flooding attacks
Udpflood floods the target SIP server with useless UDP packets. It aims to consume more link capacity than CPU resources. With running attack, it is not possible to register on SIP server or to create call. Current calls routing through SIP server are also interrupted. With IPS is possible to detect this attack, but its blocking on the server side is effortless because the attack do not allow correct packets to reach SIP server. Elimination of this attack right on the server is not possible.

5.2.3 TCP SYN flood attack
The last kind of attack against SIP server was to flood it with TCP SYN flag set packets. Applications flood2 and juno were used. Both applications interrupt all communication with server. An active IPS made the situation even worst because it consumes almost all CPU resources for analyzing incoming traffic.

5.2.4 Assessment of results
The performed tests clearly indicate SIP server’s vulnerability against DoS attacks. Even with the server running on a limited physical machine, it is possible to implement effective security mechanism. But some of the attacks cannot be blocked right on the server as was shown in provided tests. The most dangerous attacks include flooding with REGISTER, INVITE and OPTIONS messages. ACK, CANCEL and BYE messages are nearly harmless on lower rates and on higher rates have the same impact as udpflood.

The defense against not block able attacks lies in secure network topology like inclusion of a DMZ (demilitarized zone) located between trusted and untrusted zone (inner and outer). The purpose of the DMZ is to separate the safe inner part of network from the rather dangerous outer part. The potential attack from the inside network should be another source of threat. So security mechanisms like encryption, VoIP VLANs and methods as dynamic ARP inspection, DHCP snooping will provide an adequate response to security breaches. Using a honeypot in DMZ is an inspiration for further security precautions to be implemented. Although encryption is a good way to increasing security, we cannot deploy it, if we do not have its full support on all end-point devices.

6 Conclusion
The goal of authors was developing a tool to carry out penetration test on SIP server and to create an automatic defense mechanism against DoS. The SPT was designed as modular application able to generate various types of tests, which the authors deem the most popular. System itself run specific attacks, drafts assessments containing test results and propose a recommendation to ensure security against the detected threat. The assessment report is sent as a text document to an e-mail. Application itself is accessible through the internet, but only allowed user can use this application. Using of this application is under monitoring by an administrator because of its misuse potential.

The IPS security mechanism for active blocking of attacks on SIP server was based on three open-source application – Snort, SnortSam and IPtables. The main disadvantages of this solution include the delay between detection and response. If the attacker eliminate the IDS Snort whole protection system turns to nothing. There are also certain types of attacks, which can be reduced only by changes in network topology rather than right on SIP server. The paper mentions other security precautions for enhancing the server’s endurance against attacks in general. However, the proposed solution ensure a basic level of protection suitable for needs of small and middle-size offices or detached workplaces requiring their own VoIP solution.

A further research is in developing other modules for SPT like stress testing end-point devices. In case of IPS are next steps in improving of current security mechanisms and developing DDoS blocking feature.
References:


Vitae

Jakub SAFARIK received his M.S. degree in telecommunications from VSB – Technical University of Ostrava, Czech Republic, in 2011 and he continues in studying Ph.D. degree at the same university. His research is focused on IP telephony, computer networks and network security. He is with CESNET as a researcher since 2011.

Miroslav VOZNAK was born in 1971. He holds position as an associate professor with Department of Telecommunications, VSB-Technical University of Ostrava, Czech Republic. He received his M.S. and Ph.D. degrees in telecommunications, dissertation thesis “Voice traffic optimization with regard to speech quality in network with VoIP technology” from the Technical University of Ostrava, in 1995 and 2002, respectively. Topics of his research interests are the Next Generation Network, IP telephony, speech quality and network security. He is with CESNET since 1999, currently in R&D department.

Filip REZAC was born in 1985. He received M.S. degree in telecommunications from VSB – Technical University of Ostrava, Czech Republic, in 2009 and he continues in studying Ph.D. degree at the same university. His research is focused on IP telephony, computer networks and network security. He is with CESNET since 2009 as a researcher.