SIP Proxy Robustness against DoS Attacks

Miroslav Voznak, Jakub Safarik
CESNET
Zikova 4, 160 00 Prague
Czech Republic
voznak@cesnet.cz, kuba.safarik@gmail.com

Abstract: - This paper deals with one of key issues of IP telephony regarding SIP Proxy robustness against attacks and compares security risks inherent to various Denial–of–Service (DoS) attacks and addresses effective protection against them. The proposed solution is based on Snort and SnortSam and has been implemented and evaluated in testbed. Denial of Service – is one of most frequent attacks nowadays due to its simplicity and great impact. This paper describes DoS attack types, and the knowledge is used to test the robustness of the SIP proxy server. Attacks are described in detail, and a security precaution is made to prevent each of them. The solution is an IPS system, composed as a combination of Snort, SnortSam and Iptables applications. The presented solutions were tested in experiments.

Key-words: Protection against DoS, DoS attacks, IPS, Security, SIP.

1 Introduction

As a result of the ever more widespread implementation of VoIP solutions, PSTN networks are likely to be completely replaced one day. A frequently implemented solution is Asterisk – an open–source SIP server. Security was not the main goal in developing the application; it was actually rather on a sideline. Yet, security has become more and more important with its increasing popularity.

This situation is simplified by similarity of SIP protocol to HTTP and SMTP protocols, so potential hacker can use existing weakness of these protocols against SIP. One of the most used attacks is DoS (denying service completely or just particularly). On top of used attacks is because of its high efficiency and relatively simple feasibility.

The paper describes the vulnerability of Asterisk SIP proxy servers to DoS attacks and methods for server protection. For each attack, this paper describes their impact on a SIP server, evaluation of the threat and the way in which they are executed.

2 Classification of DoS Attacks

Denial of service can be achieved in several ways – flooding a server with malformed, damaged or useless packets as a result of which the server runs out of its resource capacity. The affected server is then unable to communicate with its regular users or process regular requests.

Security threats such as DoS almost do not affect the previous generation PSTN networks. This is due to their closed network topology originally designed to transfer voice information [1]. With the rising numbers of VoIP implementation, the situation is changing. And the users expect the same behaviour from the new technology. We can divide DoS attacks into three general classes [2], [4]:

- Flooding attacks – targeting on server resources (CPU, memory or link capacity).
- Misuse attacks – the hacker uses 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.

The impact of a DoS attack depends on the target. Targeting a particular client can lead to denying the service to this user only but when a SIP server is the target, no user can use VoIP. When a SIP server is attacked, the provider’s reputation also suffers. As a result, the provider may lose some of his existing and potential customers [5].

In recent years, due to their increasing frequency, impact and complexity, DoS attacks became a major issue. But we need to distinguish between intentional and unintentional attacks. As VoIP solutions have been developing fast, many server break–downs are caused by software bugs or bad configuration. Unintentional attacks, on the other hand, are for example instances of crowd frenzy when a high number of users is trying to communicate and the server cannot withstand such a high load (natural disasters, holidays, etc.). This state obviously passes very quickly, as more and more users are served by the server.

2.1 Memory depletion attacks

When a server accepts a SIP message, it has to store small chunks of information. The time for which such information chunks are stored depends on the server mode – stateful or stateless. While the transaction is being performed, the information is kept in memory, and is deleted only once the transaction is closed or timed out. The most frequent attack is a TCP SYN flood attack. The server is flooded with packets with the SYN flag set. The server allocates the necessary resources and responds with packets with SYN+ACK flags. The attacker keeps on sending new packets with SYN flags, and does not respond to packets sent by the server. The server then quickly depletes all memory resources for a new TCP connection and starts refusing regular requests.

Another example of this type of attack is to send highly fragmented packets with certain parts intentionally omitted. The server attempts to request the missing parts and stores the received packets in memory. The useless information is then stored on the server until it is timed out.

2.2 CPU depletion

Another way to effectively limit the server's ability to process regular requests is CPU depletion. A higher load may be caused by a higher number of requests received or by receiving requests requiring additional complex calculations. The server can become flooded with ICMP packets but sending malformed REGISTER messages creates the same effect with a significantly smaller number of messages.

This is due to the fact that messages are analysed after the server receives them. Even if the server is capable to process hundreds of regular messages, it can be easily forced to perform different calculations using malformed messages with bogus or invalid data or sent from nonexistent user accounts.

Paradoxically, enabled authentication on a server can trigger off more challenging operations, which makes it easier to deplete CPU resources. When a server uses certificates, the attacker may send a message with an invalid certificate. In the end, the server discovers that this certificate is invalid but the processing of the message had already consumed much of server resources.

2.3 Bandwidth depletion attacks

This type of attack does not consume resources of the physical server but rather the capacity of the link connecting the server to the network. When the link is not able to transfer regular packets, these are discarded before they can reach the SIP server. This is why it is not possible to distinguish between regular and malicious packets. Using the UDP protocol to transmit SIP messages makes the situation even worst. For obvious reasons, attackers use the stateless UDP protocol with the maximum packet size.

2.4 Misuse attacks and attacks on SIP features

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

This attack attempts to deny the users the access to VoIP, i.e. users are the victims of these attacks. This attack does not necessarily affect all users. But from the provider's point of view, this attack is much more dangerous than the above described attacks. In order to be able to carry out this type of attack, the hacker has to be able to capture the network traffic, modify SIP messages or to disguise himself as a different user.

In the case of BYE attacks, one of the parties is convinced that the call was terminated. The attacker uses data captured from the SIP headers to create malicious BYE messages. CANCEL attacks are similar, except that they affect the calls before they are connected. The attacker sends a malicious CANCEL message, using the same sequential number as the INVITE message. Using the same sequential number causes that the malicious message does not have to be authenticated provided it arrives before the final answer from the legitimate user.

The only protection against these attacks is to ensure an encrypted transfer of SIP messages.

2.5 Amplification attack

This is an instance of a distributed denial–of–service (DDoS) attack. The attacker sends packets to broadcast addresses in a specific sub–network with a spoofed source address (victim's IP address). The packet is delivered to all hosts in the sub–network and they respond back to the spoofed address.

The attacker does not need to infiltrate other hosts, he only uses them. Smurf attack and Fraggle attack are examples of this type of DDoS attack.

3 Technology Used

There are many possible ways to secure against DoS attacks. Due to features, performance and abilities of embedded systems; we choose to run the SIP server on this device. The protection mechanism should be a part of this solution. Attacks against the embedded systems are more dangerous due to their relatively lower performance which makes the attacks more efficient. We chose an IPS system, consisting of three applications.

3.1 Snort

The core of the entire IPS solution is IDS system Snort which detects malicious activity in the network [3]. The detection is based on signatures or detection of anomalies. The whole IDS system is modular, consisting of the following components:

- Packet decoder – Captures packets from network interfaces, prepare them to pre–processing.
- Pre–processor – Prepares or modifies packets before
- the processing (packet defragmentation, URI decoding, reassembling TCP streams, . . .).
- Detection engine – Responsible for attack detection.
- Logging and alerting system – Depending on detection engine, the packet may be used to log activity or generate an alert.
- Output modules – Or plugins, for adding another features.

3.2 SnortSam

This application operates on the client–server model. It allows Snort to dynamically intervene into IPtables rules. To ensure its proper operation, we need to first upgrade our Snort installation with a SnortSam plugin.

The user communicates with the Snort's sensor, sends commands to the server (where incident has been detected). The server listens on port 898, applying information from clients to IPtables rules. SnortSam messages are transferred as encrypted. A whitelist of non–blockable IP addresses is also available.

The detected traffic is then blocked for some time. Once the attack is over and timed out, the blocked IP is allowed to communicate again. Thus, only malicious traffic that poses a threat to our server is blocked.

3.3 IPtables

An open–source firewall for Linux–based operation systems. It is used to block malicious traffic on a server.

4 Results

We created a testing topology to measure DoS effectiveness and for further testing. It consists of SIP proxy server, hacker's PC and some endpoint devices. SIP proxy runs the Asterisk application, and the operation system implemented is Linux for servers – Ubuntu 10.04 LTS.

The malicious tools applied by hackers with the same OS as the SIP server are as follows [6]-[9]:

- Sipp (in repository named as sip–tester)
- inviteflood
- udpflood
- flood2
- juno

4.1 Attacks on server CPU using sipp

The Sipp programme is primarily used to simulate calls and to carry out SIP proxy stress tests. But with a simple upgrade of the call scenarios, it can make malicious calls on SIP proxy. These calls are intended to overload server's CPU. Figure 1 shows the impact of these attacks on the server. The attack scenario applied was the same for each attack. Sending malicious packets started in 10 s and continued for 60 s. Another 30 s shows the time for which the server is still inhibited by the attack.

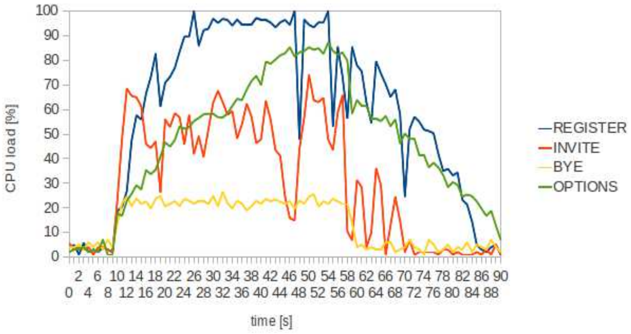

Figure 1: The impact of different attack message types on a server's CPU load.

To enable the comparison of the efficiency of individual malicious SIP messages, the messages had been sent to the SIP server with the same rate (250 messages per second). Clearly, the most effective SIP messages to attack a SIP server are REGISTER and OPTIONS. In the first case, the endpoint could not register or make calls, though running calls was not affected (the RTP stream only between endpoints). OPTIONS flood caused merely a delay in request processing, yet the situation deteriorated as the attack continued. In the end, not a single endpoint was able to register or make calls. The relatively long time necessary for the server to recover (in both cases) was rather surprising.

The delay in connection was evident in the attack performed by means of INVITE messages. Some calls failed to be connected at all. The attack was performed by a non–existing source user.

Attacks performed by means of BYE, CANCEL and ACK messages returned almost the same results (the figure illustrates only the attack by means of the BYE message). During the attack, no call or registration was affected. BYE and CANCEL were not sent to end a particular call.

Security precautions against all these attacks include Snort rules tracking the number of messages sent to the SIP server from a particular source address. Where the limit for messages was exceeded, the blocking rule was activated on the firewall. The CPU load with the activated IPS system was about 9% during these attacks (fig. 2).

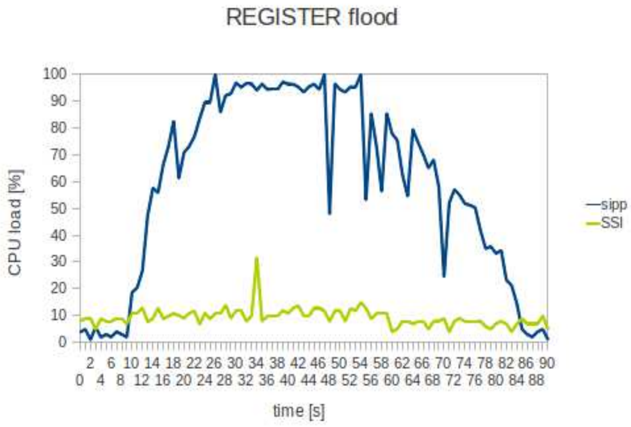

Figure 2: The impact of an attack with (SSI) and without the protection.

The attacker could be sending all the above mentioned malicious messages at a higher rate. In this way each malicious message can consume up to 100% of the server's CPU. Just to compare, the INVITE messages need 10 times higher rate than the REGISTER messages to consume a similar load of the affected machines CPU. The INVITE messages can also send the inviteflood application and create a situation very similar to the flooding with UDP packets (the same is true for any attack with a high rate of packets sent).

4.2 Link flooding attacks

Unlike the above mentioned attacks, udpflood only floods the target destination with useless UDP packets. These packets contain a sequence from 1 to 9, followed by zeros. The packet size is 1400 bytes, and the tool can spoof the source address. The CPU load is very low during the attack but all communication with the server is blocked due to a high volume of traffic. Blocking the traffic on server's interface is useless as the link would still be flooded. There is no efficient protection to be applied on the server, it is only possible to eliminate the impact of such an attack.

4.3 TCP SYN flood attack

The last type of attack against SIP proxy tested was to flood it with TCP SYN flag set packets. We used flood2 and juno applications. The Juno tool is especially dangerous as it can be easily upgraded to spoof the source address and ports.

When the attack was launched, the connection with the server was lost almost instantly. Detecting this attack is simple but surprisingly useless. Even with an active firewall rule, Snort still analyzes the malicious traffic and the server's CPU load approaches 100%.

4.4 Assessment of results

The performed tests clearly indicate that SIP proxy is rather vulnerable to DoS attacks. As the server runs on a limited physical machine, only very basic protection mechanisms against certain DoS attacks can be implemented. This system consists of the following applications: Snort, SnortSam and Iptables. The tests proved that the analysis of the server's traffic does not significantly affect server's performance (except for TCP SYN flood attack).

The most dangerous attacks include flooding with REGISTER, INVITE and OPTIONS messages, link bandwidth depletion using udpflood and TCP SYN flood attack. The attacks using malicious ACK, BYE or CANCEL messages are harmless at lower rates, with the same impact as udpflood at higher rates. No effective protection to be applied directly on the server exists against certain attacks. In this case, a more secure network topology is the only solution (Figure 3).

The main change in this topology is the inclusion of a demilitarized zone (DMZ). It is located between two firewalls (inner and outer). The purpose of this zone is to separate the safe inner part from the rather dangerous outer part of the network. Both firewalls run SnortSam agents so rules can be dynamically applied on both machines.

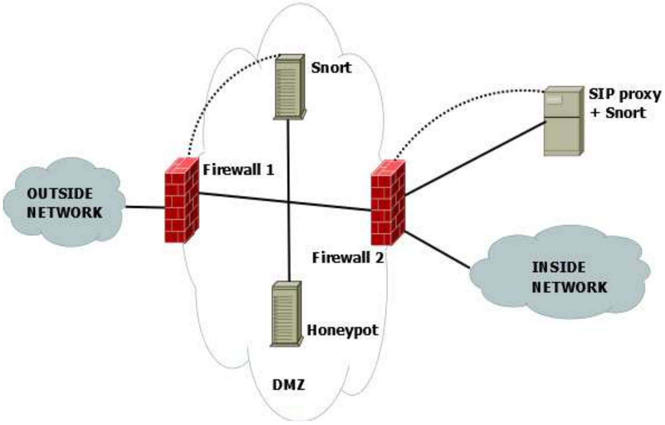

Figure 3: The proposal of a safer topology.

The inner firewall (marked as Firewall 2) serves to protect the SIP server against the attacks from inside of the network. All traffic to the SIP server has to pass through at least one firewall. The safe inner network should be implemented as a matter of course. The potential attack from inside of the network would affect many users. Using encryption, VoIP VLANs and methods such as ARP inspection and DHCP snooping should provide an adequate response to possible security breaches. The implementation of a QoS mechanism should further reinforce the protection.

A honeypot located in the DMZ is an inspiration for further security precaution to be implemented.

5 Conclusion

DoS attacks can be carried out in many different ways. We tested their efficiency in practice and documented the results. This article maps the most frequently used attacks of today and evaluates the risk inherent to each of them.The resulting solution is an IPS system based on the Snort application. This application is combined with two other – SnortSam and Iptables. The disadvantages of this solution include the delay between the detection and response (typically where the firewall is not on the same physical machine). If the attacker eliminates the IDS system, the whole protection system is useless. The impact of certain attacks can only be reduced by implementing changes to the topology. In this paper, we propose to reinforce the topology's security by introducing a demilitarised zone. The argument here is the impact of udpflood and a TCP SYN flood attacks. The paper also mentions other security precautions which help to enhance the server endurance against attacks in general.

As a result of attacks, the high computing capacity can be significantly reduced. This can be prevented by using parallel computing and a link with high capacity (Etherchannel, optical cables, . . .). However, such measures can increase the cost of the proposed solution slightly. The solution proposed in this article should ensure a basic level of protection suitable for small and middle–size offices or detached workplaces requiring their own VoIP solution.

The contribution of this paper includes the performed comparison of the DoS attacks' efficiency. It was tested both without any protection and then with implemented Snort and SnortSam applications as proposed in our solution.

Acknowledgement

This work has been supported by the Ministry of Education of the Czech Republic within the project LM2010005.

References:

[1] B. Johnston, *SIP: Understanding the Session Initiation Protocol*.London: Artech house, 2009.

[2] A. Steffen, D. Kaufman, A. Stricker, *SIP Security*. Winterthur: Gesellschaft fur Informatics, 2004.

[3] R. Rehman, *Intrusion Detection Systems with Snort*. New Jersey: Prentice Hall PTR, 2003.

[4] D. Sisalem, J. Kuthan, T. Elhert, Denial od Service Attacks Targeting SIP VoIP Infrastructure: Attack Scenarios and Prevention Mechanisms. *IEEE Network*, 2006.

[5] D. Sisalem, J. Floroiu, J. Kuthan, U. Abend, H. Schulzrinne, *SIP SECURITY.* Wiley, 2009.

[6] D. Endler, M. Collier, *Hacking Exposed VoIP*. McGraw–Hill Osborne Media, 2009.

[7] S. Elhert, G. Zhang, D. Geneiatakis and et. al., *Two layer Denial of Service preventionon SIP VoIP infrastructures.* Computer Comm., Elsevier B.V., 2008.

[8] M. Voznak, F. Rezac, K. Tomala, SIP Penetration Test System, *In Conference Proceedings TSP 2010* , August 2010, Baden near Vienna, Austria, pp. 504-508.

[9] M. Voznak, F. Rezac, The implementation of SPAM over Internet telephony and a defence against this attack. Publisher: Asszisztencia Szervező Kft. Budapest, *32nd International Conference TSP* , August 2009, Dunakiliti, Hungary.