available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Survey of network security systems to counter SIP-based denial-of-service attacks

Sven Ehlert[a], *Dimitris Geneiatakis*[b], *Thomas Magedanz*[a,*]

[a] *Fraunhofer FOKUS, Berlin, Germany*
[b] *University of the Aegean, Greece*

ARTICLE INFO

Article history:
Received 11 April 2009
Received in revised form
27 September 2009
Accepted 28 September 2009

Keywords:
Anomaly
DDoS
Detection
DoS
Flooding
IMS
Mitigation
RTP
Signature
VoIP

ABSTRACT

Session Initiation Protocol is a core protocol for coming real time communication networks, including VoIP, IMS and IPTV networks. Based on the open IP stack, it is similarly susceptible to Denial-of-Service Attacks launched against SIP servers. More than 20 different research works have been published to address SIP-related DoS problems. In this survey we explain three different types of DoS attacks on SIP networks, called SIP message payload tampering, SIP message flow tampering and SIP message flooding. We survey different approaches to counter these three types of attacks. We show that there are possible solutions for both payload and flow tampering attacks, and partial solutions for message flooding attacks. We conclude by giving hints how open flooding attacks issues could be addressed.

1. Introduction

Since the invention of the telephone, real time communication networks have mostly been built using closed circuit-switched network infrastructures, e.g. the Public Switched Telephone Network (PSTN). With the advent and the increasing popularity of the packet-switched Internet data network, providers are seeking ways to combine both communication and data networks on an all-IP network basis.

Voice-over-IP (VoIP) is the technology used to establish telephone calls and other multimedia streams over the IP protocol. Furthermore, international standards organisations have coined the term Next Generation Networks (NGN) to define a standardised way to encapsulate telecommunication services in an IP network. The 3rd Generation Partnership Project (3GPP) has standardised a NGN called IP Multimedia Subsystem (IMS) (2007), which is a common architectural component of further NGNs, e.g. the European Telecommunications Standards Institute (ETSI) uses IMS in their NGN reference definition Telecoms & Internet converged Services & Protocols for Advanced Networks (TISPAN (2008a)). Current IP Television (IPTV) networks are also mostly based on NGN concepts.

The key protocol for regular VoIP services or NGN architectures is the *Session Initiation Protocol* (SIP) (Rosenberg et al.,

[*] *Corresponding author.*
E-mail addresses: svenehlert@gmx.de (S. Ehlert), thomas.magedanz@fokus.fraunhofer.de (T. Magedanz).

0167-4048/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2009.09.004

2002) as the basic signalling protocol. Unlike the closed PSTN architecture, SIP networks are deployed on the open IP stack and thus vulnerable to many of the same security threats including Denial-of-Service (DoS) attacks. DoS attacks aim to render a service or application inoperable by sending specially crafted messages or sending a huge amount of useless traffic.

In 2005, the US National Institute of Standards and Technology determined DoS flooding to be a serious threat for SIP VoIP infrastructures (Kuhn et al., 2005). In a threat analysis for ETSI TISPAN networks, DoS attacks on publicly available interfaces were *considered a critical risk*. The authors rate the attack potential to be highly likely with a high impact on the attacked network. Furthermore, DoS attacks on non-publicly addressable interfaces are considered a minor risk (TISPAN, 2008b).

As SIP deployment is likely to increase in the future, protection against DoS attacks is becoming a necessity. Sprint, a US communication provider claims that general DoS detection methods should be enhanced to handle SIP VoIP attacks (Larson et al., 2004). Arcor is a major German communication service provider. They are currently deploying large-scale SIP-based communication NGNs. Arcor also claims that DoS protection should be a *requirement for service providers*. They propose the use of Session Border Controllers as the first line of defence with DoS protection features (Tzvetkov and Zuleger, 2007). For ETSI TISPAN architectures, DoS mitigation mechanisms are also a requirement (TISPAN, 2006).

There are several types of DoS attacks on SIP network. First, common IP network and transport layer DoS attacks are also valid for SIP networks. These attacks have been known about for years and have been excessively studied in literature (Chang, 2002; Peng et al., 2007). Furthermore, there have been new attacks that have directly targeted the SIP application layer itself. These attacks include SIP message flooding, SIP message payload or SIP flow tampering attacks. This article focuses on SIP-related DoS attacks. We will present a survey of the methods that have been proposed to counter these attacks.

1.1. *Target audience*

The aim of this survey is to give an extensive overview of the current status of SIP DoS protection research. This article should therefore be especially helpful for new researcher in the field to get a comprehensive overview of the different methods that have been applied for DoS protection and their effectiveness in reaching this goal. To help the reader, we also provide an extensive list of pointers to relevant works in this field for further reference. With the final discussion of the surveyed methods, the interested reader will find topics that might need further research in regard to SIP DoS protection.

1.2. *Article structure*

We will begin our survey by presenting the necessary background information. We will then give an overview of the operation of the Session Initiation Protocol and summarise common IP-based DoS attacks.

We will highlight SIP's vulnerabilities with regard to Denial-of-Service and classify the three main categories of SIP DoS attacks. Based on this we will develop a list of attributes to classify the countermeasure systems used to handle SIP DoS attacks. These attributes fall into two main categories: algorithm-related evaluation criteria and framework-related evaluation criteria.

The main part of this work consists of the actual survey of different protection mechanisms. We will summarise over 20 different works, which have been presented during the last five years. These works all have targeted different types of DoS attacks. Following, we will present a discussion of these methods, where we will indicate positive and negative aspects of the different approaches. To conclude we will explore open issues in the field of SIP attack mitigation and prevention. Readers of this work could thus find tips for possible directions in further research.

2. Background information

2.1. *Real time communication with Session Initiation Protocol*

SIP (Rosenberg et al., 2002) is a text-based protocol that has been standardised by the Internet Engineering Task Force (IETF). It has been designed to establish or terminate a session among two or more partners. The message format is similar to the HTTP protocol with message headers and corresponding values, e.g. From: user@sip.org to denote the sender of a message. Several message types are defined (e.g. REGISTER, INVITE, ACK, BYE …) and encoded in the first line of each message (Request-URI). Several message headers are dedicated to routing purposes in the SIP network:

To: Denotes the receiver of this SIP messages. This is generally the publicly available address of the user (*Address of Record*).
From: Denotes the sender of the message.
Contact: The actual location where a user can be reached. This location can be different from the From URI.
Record-Route: Indicates that an intermediate proxy wants to receive further signalling traffic.
Route: Indicates a route that a new request is going to take.
Via: A list of all intermediate SIP entities that these messages have passed so far.

A SIP-based network consists of several entities (see Fig. 1). SIP entities include *User Agents* that generate or terminate SIP requests, *Registrars* where users log in and announce their availability in the SIP network and *Proxies* that forward requests within the appropriate SIP network. Several proxies can be deployed in a SIP infrastructure, e.g. outbound proxies regulate outgoing routing from one network to a foreign network and incoming proxies handle all incoming SIP requests. They possibly perform additional security checks. Additionally, NGNs introduce different terms (e.g. the inbound proxy is called Call Session Control Function in IMS) and add further entities.

SIP is a complex protocol, with its basic specification alone being the third largest work developed by IETF. The document defines SIP's multi-layered session control mechanism i.e. it distinguishes between *transactions*, *dialogs* and *sessions* between participants. SIP also defines its own reliable transport mode in case it is operated over UDP connections. The UDP transport mode is based on four full state-machine specifications (see Fig. 2).

Together with over 100 available extensions (Rosenberg, 2008), SIP is the most complex protocol suite designed by IETF to date. As the complexity increases, the possibility of potential design and implementation errors also increases. This complexity can be exploited by malicious users.

SIP is a signalling-only protocol. The actual communication data is transported using different protocols, commonly with the Real time Transport Protocol (RTP) (Schulzrinne et al., 2003).

2.2. *Denial-of-Service attacks in the Internet*

SIP is an application-layer protocol on top of the open TCP/IP stack, thus the general IP architecture design has a direct impact on SIP's security features.

IP and the Internet were initially built for openness and scalability, and these features played a key role in the success of today's Internet. However, this design decision has come at a price, as basic IP security mechanisms were not considered in its design phase, and even today are only available through extensions. For instances while it is easy to integrate new hosts into an existing network, there is no packet authentication in place. A host, once connected, can thus immediately start sending packets to all possible destinations.

Also, in the Internet packets are sent end-to-end i.e. intermediate routers forward packets towards the destination while refraining from examining the packet's content. Combined with the missing packet authentication mechanism, this leaves the decision of whether to serve a packet or not to the receiving host. The packet is delivered to the destination in any case, and the destination host has to discriminate between valid and malicious packets.

This situation has led to a certain type of attack known as *flooding Denial-of-Service (DoS)*. A host can be rendered inoperable whenever it is easier for a malicious user to generate requests than for the destination host to validate them (see Fig. 3).

There are many ways to launch such attacks against servers in the Internet. Mostly, an attack is conducted using the transport layer of TCP/IP. A well known attack is TCP SYN flooding (Eddy, 2007), where the attacker launches multiple TCP session initiation requests (TCP SYN), but does not finalise the TCP handshake after the server responds to the request. Thus open sessions that consume memory are created at the target. The server cannot free this memory immediately as it has to assume that the missing TCP handshake messages have been lost and will eventually be re-sent by the sender. With too many concurrently open sessions the server will run out of memory resources and will not be able to respond to further requests.

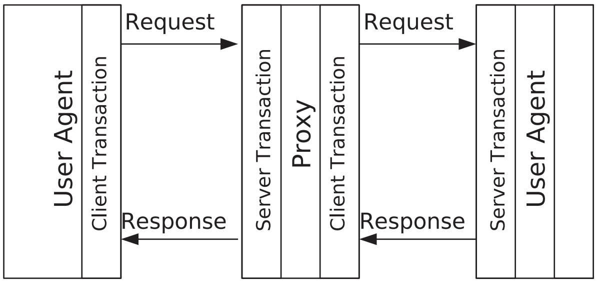

Fig. 2 – SIP Transaction Relationship. Each interacting entity implements up to two sender and two receiver transaction state machines. The SIP specification distinguishes between INVITE transaction state machines and non-INVITE state machines.

These attacks have already been extensively classified and categorised in literature and prevention mechanisms have been proposed (Chang, 2002; Peng et al., 2007; Mirkovic and Reiher, 2004).

Another possibility to launch a DoS attack is to exploit a well known vulnerability at the target host like a buffer overflow. However, once the vulnerability has become known it can be easily prevented with a software upgrade.

3. Classification of SIP-based DoS attacks

Common IP layer-based DoS attacks as described in the previous section are also relevant for SIP networks. Additional dedicated application-layer attacks also directly target SIP-based networks.

In our previous paper we explained the vulnerabilities that allow dedicated SIP attacks to occur (Sisalem et al., 2006).

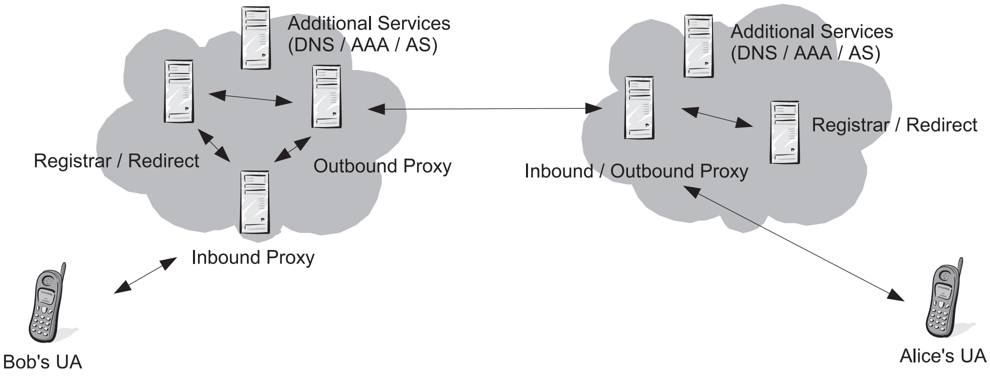

Fig. 1 – SIP Architecture Schematic Overview.

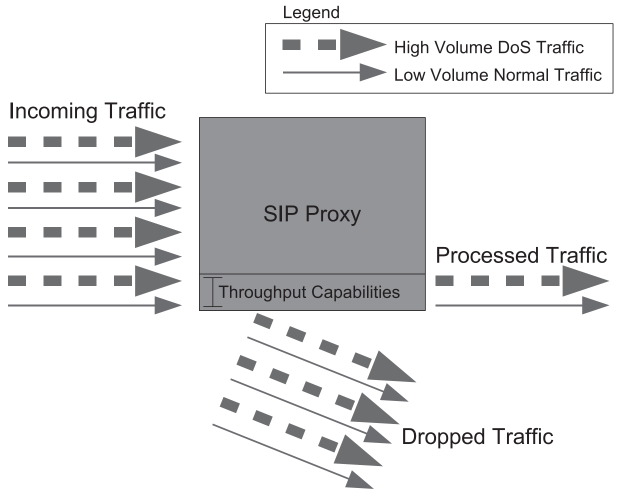

Fig. 3 – Schematic overview of a DoS flooding attack. Due to the server's limited processing capabilities a lot of regular requests cannot be processed if a high load of malicious messages are targeted towards the server.

Basically, these are missing sender authentications for packets as previously described, software errors in SIP implementations, poor design in implementations that allow resource depletion to occur or missing or wrongly applied authentications on the SIP layer.

The goal of a Denial-of-Service attack is to render the service or system inoperable. Hence an attack can be directed toward different entities in the network, depending on the attacker's intent. If the aim is to render the service as a whole inoperable, the main target will be the core servers in the SIP infrastructure. These can be SIP proxies but also other servers which are necessary in a SIP infrastructure: DNS, RTP proxies, gateways to other networks, etc. Direct attacks on the user agent are also possible, however they will have a lesser impact i.e. the attack's effect will only be noticed by the user agent itself.

We will distinguish between three different types of SIP DoS attacks. They are *SIP Message Payload Tampering*, *SIP Message Flow Tampering* and *SIP Message Flooding* (see Fig. 5).

3.1. *SIP message payload tampering*

The first class of attacks is based on tampering with the actual SIP message or more specifically, the SIP payload. SIP is a text-based protocol and messages are transported usually in clear text. Attackers can try to inject harmful content into a message, e.g. by entering meaningless or wrong information with the goal of exploiting a buffer overflow at the target. Also, such messages can be used to probe for vulnerabilities in the target. Harmful code that will be executed in an unforeseen context can be introduced into the payload. An example is SQL code injection, which allows the attacker to execute SQL code within a database.

Such attacks can target both the proxies and UAs. Unintentional attacks are possible due to poor SIP implementations. Especially with probing requests it is likely that these messages will have been launched from different sources.

3.2. *SIP message flow tampering*

A special case of DoS attacks in real time communication networks are attacks that disturb the ongoing communication between users. Common internet services like web browsing or email communication have an asynchronous time model i.e. a requested web page is directly delivered to a user. The user will read it without further communication to the web server. The same applies to email – a user downloads the email and studies it independently of a server connection. In contrast, in SIP real time communication networks two communicating users establish a constant connection with each other whereby content is transmitted continuously between both parties.

An attacker can now target this connection by introducing fake signalling messages into the communication channel. Several different SIP signalling messages can be misused for this task. A BYE message with the right credentials can prematurely terminate a session (see Fig. 4). An injected CANCEL request can prohibit even the establishment of the request. Using an INVITE message, an attacker can renegotiate session parameters and redirect ongoing sessions. More details are available in published papers, e.g. in Geneiatakis et al. (2006).

The attacker needs to know the session parameters for these attacks to succeed. He can sniff them from the network. Tests have shown that multiple implementations do not follow the SIP specification correctly, thus proving the feasibility of such attacks (Seedorf et al., 2008).

These attacks are targeted at SIP UAs only.

3.3. *SIP message flooding*

When talking about a DoS attack, one generally means flooding attacks that overwhelm a victim's resources. There are three main resources that can be targeted in a SIP flooding attack: bandwidth, CPU, or memory (Sisalem et al., 2006).

3.3.1. *Bandwidth*

The target is flooded with more messages than the network can handle, e.g. the attacker manages to generate an attack rate of 10 GB/s while the target is connected to the Internet via

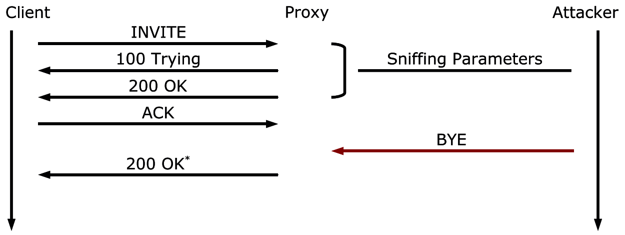

Fig. 4 – Exemplary message flow tampering with an attack using an injected BYE message. The attacker sniffs network traffic to gain the necessary session parameters and injects a fake BYE message, thus preliminarily tearing down the communication channel. Depending on the proxy implementation and the injected BYE parameters, the last 200 OK* message may or may not be visible.

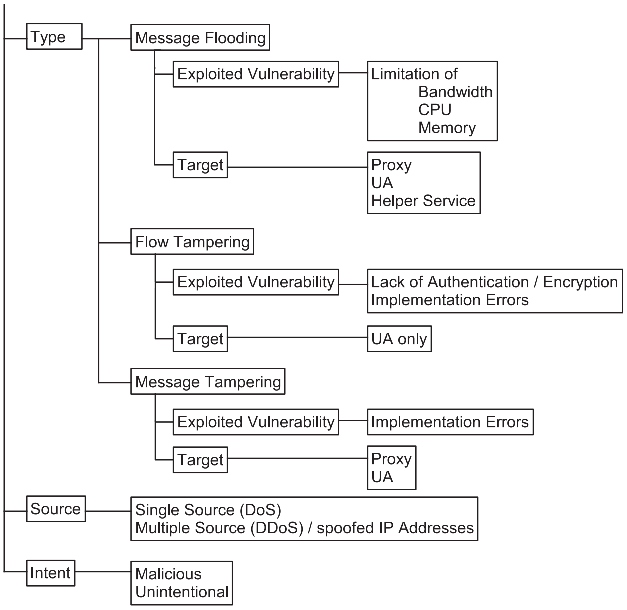

Fig. 5 – Classification of SIP DoS Attacks.

1 GB/s lines. This is a general DoS flooding problem and not specific to SIP networks.

3.3.2. *CPU*

The target is flooded with more messages than it can process at a given time. As SIP is a text-based protocol, it has to parse each incoming message. Furthermore, if SIP authentication info is supplied in the flooding message, it has to calculate if the user is authorised to access the service (digest authentication). A special case is when the target CPU cannot continue its operation because it is waiting for input from other entities, like a database or the DNS service.

3.3.3. *Memory*

Several requests create session state at the target. An INVITE message sent to a proxy will be forwarded and the proxy will wait up to three minutes for a reply. During this time state memory is consumed at the proxy. If too many such messages are encountered, the proxy will run out of memory.

Flooding can be achieved with different SIP messages (INVITE, REGISTER, OPTIONS, etc.), and can be directed either at the proxy or different UAs.

This attack, like the other described attacks, can be launched from a single-source or from multiple-sources. The latter is called a Distributed Denial-of-Service (DDoS) attack, a concept already well known from IP-based flooding attacks. In the case of a distributed flooding attack, the attacker employs a large number of (usually unaware) computers with different IP addresses. These machines are controlled to generate a higher bandwidth stream of messages than it would possible from one single machine. Furthermore, from the defence point of view attacks where source IP addresses in packets are spoofed to escape detection can be considered to be a kind of distributed attack.

Flooding attacks can have different causes: first, there is the planned attack that intends to create damage to the network. Furthermore, there are unintended attacks which can stem from misconfigured devices or wrong SIP implementations. For example, consider a REGISTER flooding after a power outage, when all devices are trying to REGISTER again at the same time. Also, wrongly configured re-REGISTER intervals might be configured at multiple UAs after an automatic software upgrade. These devices can cause an unintentional packet flood.

Detection and mitigation is necessary for both intentional and unintentional attacks, as the consequences will be similar. It is generally easier to detect unintentional attacks, as malicious users will probably implement obfuscating features to evade detections.

3.4. *Further SIP vulnerabilities and attacks*

Several further possibilities for attack exist in SIP networks, with the most common ones being toll fraud or sending unsolicited messages (Spam over IP Telephone, SPIT). They might require different detection and prevention methods and are not the topic of this work. Several studies give a wider overview of multiple SIP vulnerabilities and attacks (Kuhn et al., 2005; Geneiatakis et al., 2006; VoIP Security, 2005; Zhang et al., 2007b; Rosenberg and Jennings, 2008).

4. Evaluation criteria for SIP DoS defence systems

Multiple countermeasure schemes have been proposed to target the new SIP-related DoS attacks. They are necessary as general IP-based DoS protection systems do not address dedicated SIP DoS attacks. For example, SIP messages can use different transport protocols to deliver messages to the destination (such as UDP, TCP or even SCTP). General DoS flooding protection mechanisms operating solely at the IP layer would not be able to detect an attack flow using different transport protocols, as they cannot take the actual message payload into account. SIP message flow tampering detection systems are also only possible if they have SIP knowledge. SIP DoS countermeasure systems can thus be seen as an additional layer of security to be deployed in conjunction with general IP DoS protection mechanisms.

For comparative purposes we will evaluate these methods using different criteria as they target different attacks, use different countermeasure algorithms or have performance differences. The criteria have been chosen to be as general as possible, so that they can be applied to most of the presented algorithms. We have omitted criteria that could have only been applied to a sub-group of the algorithm set. We define two main criteria groups: *algorithm-related* and *framework-related* criteria. The former is related to the theoretical idea of the countermeasure solution. It covers the mathematical principle the method is based on. The latter covers the actual implementation of the theoretical algorithm including setup, architecture and performance.

4.1. *Algorithm-related evaluation criteria*

4.1.1. *Algorithm principle*

This is the core of the defence method, the basic mathematical principle of how the author would handle the envisaged attacks. This could be a statistical model, a data mining approach, etc. As SIP is based on a complex state-machine specification, many authors use this specification as a basis for their defence solution.

4.1.2. *Attack classes*

We indicate what kind of attack class is addressed. Here we consider the three main attack groups as identified in Section 3: payload tampering, flow tampering, and flooding. Especially for flooding attacks we evaluate if the method considers single-source flooding or multiple-source flooding. If other, non DoS related attack types are addressed (e.g. SPIT detection), they will be mentioned. However, these attacks are not the focus of this article.

4.1.3. *Victim*

While most proposals are concerned with the protection of the main servers (which would be the SIP proxy or the P-CSCF in IMS networks), some are targeted at SIP client (UA) protection.

4.1.4. *Protocol*

Our focus in this work lies on specific SIP-based attacks. However, RTP also plays an important role in SIP-based networks. We mention if the discussed method also considers RTP-relevant attacks. This would be mostly the case for RTP flooding attacks.

4.1.5. *Reaction*

This defines what reaction is achieved by the defence method. All algorithms are targeted for attack detection, however not all algorithms can later classify the attack traffic, which is needed for attack prevention. Other ones might not be able to prevent the attack, but could propose a method for attack mitigation i.e. a method to sustain the attack.

4.1.6. *Detection strategy*

To detect an attack, the algorithm has to have some knowledge of the attack. Basically, there are two possible principles. Either the attack is described (e.g. using signatures), which is called pattern-based detection or by defining some type of ‘‘normal’’ network traffic. Attacks are then detected by deviation from this norm, which is called anomaly-based detection.

4.2. *Framework-related evaluation criteria*

4.2.1. *Setup*

The security mechanism has to be employed in a security framework. There are different possibilities that are analogue to setups in Intrusion Detection Systems (IDS). The common case would be a Network-based IDS (NIDS). This is a dedicated entity in the network that can monitor traffic and analyse it. For SIP, this would consist of a traffic monitoring tap, likely a SIP message parser, and the implementation of the proposed algorithm. A host-based IDS (HIDS) is deployed directly at the same host as the target SIP proxy and evaluates information from this host, e.g. application log files. Another option would be to implement the defence solution as an extension module of the target proxy. These possibilities are depicted in Fig. 6.

4.2.2. *Implementation*

Details about the actual implementation, if any. This includes the programming language and the framework used.

4.2.3. *Placement*

The placement of the framework within the network has an effect on what information the framework will have for evaluation, especially in NIDS-setups. A common placement would be at the ingress point of the network, so that all incoming passing traffic can be seen. Some setups are based on distributed systems with multiple different monitoring points.

4.2.4. *Reactive measure*

This means the reactive measure against a classified attack, if a method provides one. This could be firewall control to block certain requests, for example.

4.2.5. *Performance test results*

If an implementation exists we will present performance results as described by the authors. Noteworthy features include:

Detection latency – the time after attack launch needed before it can be accurately detected,
Processing latency – the delay normal users encounter because this method is deployed in the network, and
Processing capabilities – how many messages the framework can process in one second.

Note that these results cannot directly be compared between different implementations – there are too many variations between test setups, test configurations and testing hardware.

4.2.6. *Scalability*

Especially for high-volume message flooding attacks it is important that the framework can scale for higher

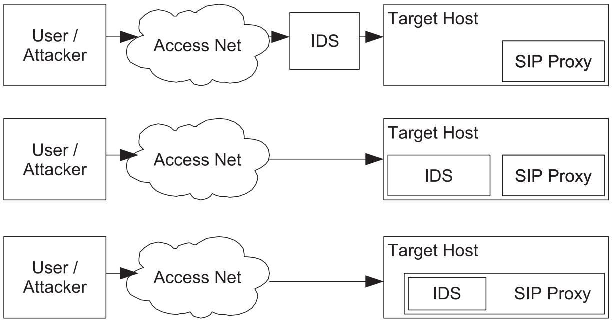

Fig. 6 – Difference between IDS systems. Top: Network IDS (NIDS). Middle: Host IDS (HIDS). Down: Extension Module.

bandwidths. Here we will provide information if the authors have considered this, e.g. by limiting the memory requirements for their solution or by introducing the possibility of using multiple detection machines.

5. Survey of proposed SIP DoS countermeasure solutions

Here we present the actual countermeasure solutions that have been proposed by various researchers. The countermeasures are presented in order of publication. Each proposal is summarised according to the aspects of the evaluation criteria introduced in the previous section and separated into an algorithm-section and a framework-section.

We base this summary on available facts from the publication and do not judge any of the proposed ideas here. This is especially important with regard to measurements because of different testing conditions. Thus neither of the given performance measurements can be used directly for performance comparisons between different ideas.

A comparison and discussion of the presented ideas will follow in a later section.

With the exception of Inacu03, the survey covers proposals that have been described in scientific publications. There are already several commercial SIP security solutions (e.g. from Borderware or AcmePacket) targeting the same threats. However, information on and specifications of these systems are restricted, and thus cannot be evaluated here.

5.1. *Iancu03*

5.1.1. *Algorithm*

Iancu (2003) developed a DoS flooding mitigation mechanism dubbed ‘‘Pike’’ that rate-limits incoming traffic on a per-host basis.

This method is listed as an example for the various rate-limiting software mechanisms available as add on for SIP servers or in commercial security solutions. The algorithm counts all incoming requests per IP address in a defined time frame. Whenever a fixed upper limit is reached, further messages from the offending IP address are not processed for a limited time.

5.1.2. *Framework*

Pike is distributed with the open-source SIP Express Router (ser) (Rebahi et al., 2003). It runs as an extension module within the proxy and operates only on that proxy. This is a common DoS mitigation algorithm that is also deployed in similar forms in other security frameworks, including in commercial setups such as Borderware.

5.2. *Reynolds03*

5.2.1. *Algorithm*

Reynolds and Ghosal (2003) propose a DoS flooding countermeasure mechanism by detecting open SIP sessions using the cumulative sum method (CUSUM) (Page, 1954).

Attack detection is based on the observation that the ratio of connection setup messages (INVITE) and positive replies (200 OK) should be roughly equal at any given time. Hence, when this ratio suddenly changes, this is likely to be an indication of a DoS flooding attack, as such an attack would yield a lot of open connections that are not closed immediately. This principle was first proposed to detect general TCP SYN flooding attacks (Wang et al., 2002). To determine the actual moment when the flooding begins (and stops), the non-parametric cumulative sum is applied. CUSUM is a sequential analysis technique which is used for monitoring change detection. This method is an anomaly-based mechanism, where threshold values have to be set to raise correct alarms.

This method is targeted to protect the end-user terminals (UAs) only; the authors do not consider the main SIP proxy to be the target of an attack. Under this assumption their method can also be used for attack prevention. In case of a detected attack against a UA, traffic to the attacked UA can be throttled down or temporarily disabled by the SIP proxy.

5.2.2. *Framework*

The authors suggest implementing their mechanism within a NIDS (dubbed ‘‘Application-Layer Attack Sensor’’) placed in front of the SIP proxy of the network to be protected. For attack mitigation, the sensor has a connection to the SIP proxy, to instruct it when it should throttle down or temporarily block requests. The framework has not been implemented, however its operation has been simulated with the author’s ‘‘emulation toolkit’’ which is not further specified.

The authors simulate different flooding attacks to multiple UAs in the SIP network. As only UAs are considered to be victims, the attack rate was set rather low, ranging from 1 to 200 msg/min within the simulation. All attacks could be detected with the simulation, however due to the low attack rates it took up to 8 min to detect the attack with less than one attack message per minute.

5.3. *Wu04*

5.3.1. *Algorithm*

Wu et al. (2004) present a stateful data mining IDS dubbed ‘‘SCIDIVE’’ to detect SIP message flow tampering and DoS flooding attacks by correlating SIP and RTP network traffic events.

Both SIP and RTP traffic is monitored and individual events are generated from the monitored packets. Events are pre-defined characteristics that can be extracted from received messages, e.g. a session tear down event (when a BYE message is intercepted), or an RTP jitter event (when two out of order packets are observed). Detection signatures can be applied for these events. For message flow tampering this would be, for example, a tear down event *preceding* the corresponding RTP stream stop event. Normally, it would be the other way around. For flooding detection this would be to detect a large number of unchallenged 401 reply events from the proxy. This is thus a stateful approach: to determine if the target is currently being attacked, the previous state of the system is considered alongside the currently encountered messages.

While the authors provide some hints for proxy protection, their focus lies on UA attack detection. This mechanism does not provide mitigation features.

5.3.2. *Framework*

For cross-protocol correlation to work properly, SCIDIVE has to be deployed at a place where it can monitor both SIP and RTP traffic. As RTP is generally routed end-to-end, the IDS cannot be deployed in the SIP provider network, except if the provider forces RTP traffic to pass its network (e.g. by enforcing the usage of an RTP proxy). However, the authors do not assume this scenario. Instead, they propose to place the IDS directly at all relevant UAs. It is unclear if it is their intent to place the IDS in front of users that are likely to be the target of an attack or in front of malicious users. In the test setup the IDS is placed in front of all users i.e. also in front of malicious users. They hint at the possibility of extending the framework that would allow multiple IDS instances to communicate with each other.

The framework has been implemented as a non-specified prototype NIDS. The need for efficient state handling is mentioned for scalability reasons but not evaluated any further. A theoretical performance projection is given.

5.4. *Geneiatakis05*

5.4.1. *Algorithm*

Geneiatakis et al. (2005) present a signature-based solution to protect SIP network elements from message payload tampering attacks.

They suggest the employment of signature patterns (based on the SIP grammar) to distinguish well formed messages from malformed ones, similar to computer virus signature descriptions. Specifically, any SIP message which does not correctly conform to the SIP grammar is identified as malformed. Two types of signatures are defined. The first type describes a general signature structure, to be applied to any SIP message, whereas the second type defines signatures that are applied only to specific SIP messages. The signatures are created using Perl Compatible Regular Expressions. Any message classified as malformed is dropped and thus will not be processed by the target entity.

5.4.2. *Framework*

The authors have outlined an implementation either in any SIP network element as a pre-filtering mechanism before incoming messages are passed to the actual SIP parser or the solution can be incorporated into a General NIDS setup. For testing purposes the authors have implemented their solution in the core of the SER SIP proxy. They performed measurements for the introduced processing overhead in various testing scenarios and measured false alarm ratios. The results show that the delay introduced on the server side is about 120 μs, while no false alarms have been raised.

5.5. *Ehlert05*

5.5.1. *Algorithm*

Ehlert et al. (Markl et al., 2005; Zhang et al., 2007a) present an enhanced DNS cache solution dubbed ‘‘DNS Attack Detection and Prevention’’ (DADP) to mitigate special SIP-DNS DoS flooding attacks.

A SIP proxy can be rendered inoperable if it has to wait for responses from helper services before it can continue its operation. SIP messages can contain multiple fully qualified domain names, which need to be resolved by a DNS server. When the DNS server is queried with unresolvable domain names, it can take several seconds before a final answer can be delivered from the DNS server. Thus, a low-rate flood with specially crafted SIP messages to the SIP proxy can block it if these messages contain unresolvable domain names in SIP routing header fields.

The authors detect such attacks by monitoring DNS resolving requests issued from the SIP proxy. An enhanced DNS cache is applied to store resolvable DNS names related to the SIP proxy. The cache uses parallel operating queues to resolve external domain names. If the attack is detected, further external domain name queries are only performed by a subset of the parallel operating queues, while the remaining queues return results from the cache only. Thus it is guaranteed that the SIP proxy will always receive an instant reply to DNS requests, which mitigates the effects of the attack. For scalability reasons, cache entries are limited and a cache-entry replacement strategy is applied to the cache.

5.5.2. *Framework*

This solution is implemented as an external DNS cache for the ser proxy. The cache implementation is a modification of the Dnsmasq caching proxy written in the C programming language. The cache control module runs within ser, while the DNS cache can operate on any other host. The authors give figures of proxy operation with and without the cache. A comparison of different cache-entry replacement strategies show that a Least-Frequently-Used replacement strategy yields optimal mitigation. Thus, the limiting factor here would only be the performance of external DNS servers.

5.6. *Markl05*

5.6.1. *Algorithm*

Markl et al. (2005) propose a signature-based message integrity checker and DoS flooding preventing mechanisms based on the Snort IDS (Roesch, 1999).

Passing SIP messages are checked for known malicious content, e.g. SQL code injection. Similarly to Iancu03, single-source message flooding is detected by a threshold message counter, and further messages above this threshold are dropped. Additionally, signatures are applied for general IP-based attacks not related to SIP.

5.6.2. *Framework*

This method is supposed to be a lightweight complimentary prevention system to be deployed together with the two previous prevention systems (Geneiatakis05, Ehlert05). The attack signatures are fed into the Snort IDS. Snort works as a network bridge and captures all passing traffic and is thus best placed at the network ingress point. It controls the network firewall through the use of the SnortSam firewall controller. Different firewalls can be used to block offending senders. Prelude is used to gather intrusion alerts from multiple-sources and present alerts to the operator. Tests have been conducted with flooding rates of 3000 msg/s. As no modification to Snort has been done, performance and scalability are dependent on Snort's abilities.

5.7. *Chen06*

5.7.1. *Algorithm*

Chen (2006) proposes a SIP state-machine specification to detect multiple-source message flooding attacks.

The author models the four defined transaction state machines specified in the SIP RFC (INVITE and non-INVITE transaction state machine, both for the client and server part). For each transaction the according state machine is updated whenever a new SIP message is encountered. This idea was first introduced for TCP/IP intrusion detection (Sekar et al., 2002). For flooding detection Chen adds an error state to each state machine and defines how this error state can be reached. An attack is indicated if the number of error states in one sampling interval surpasses a threshold. The threshold for attack detection is network dependent. This method is for detection only.

5.7.2. *Framework*

This is a theoretical concept. The author proposes to place his mechanism in an external IDS at the network ingress point.

5.8. *Niccolini06*

Niccolini et al. (2006) present a multi-layered IDS to counter different types of attacks.

5.8.1. *Algorithm 1*

The first counter-measure mechanism is a SIP message integrity checker to prevent SIP message payload tampering, similar to Geneiatakis05.

The mechanism checks that all incoming SIP messages are well-formatted by ensuring that all header field sizes are correct, for example. Non-conforming messages can be discarded.

5.8.2. *Algorithm 2*

The second counter-measure mechanism is a basic SIP dialog state machine to detect out-of band message flow tampering and DoS message flooding.

A basic SIP Dialog state machine guarantees that messages within one dialog have the correct order, e.g. that a BYE message follows after the appropriate INVITE message. Out-of-order messages can be discarded. A rate-limiting counter is applied to throttle the number of transactions one user can initiate during one sampling interval.

5.8.3. *Framework*

The countermeasure modules have been implemented as C extension modules to the Snort IDS (Roesch, 1999), which is supposed to be placed at the network ingress point. The authors see the implementation as a prototype and suggest the deployment of these counter-measure mechanisms in higher performing systems for real life setups. The prototype can process up to 860 malformed requests/s and introduces minimal delay. However, it crashes at higher flooding speeds in the test bed.

5.9. *Sengar06-1*

5.9.1. *Algorithm*

Sengar et al. (2006a, 2008) propose a statistical flooding detection method dubbed "vFDS" based on Hellinger (1909) Distance calculation.

This work extends the detection principle of Reynolds03 by not only correlating the amount of INVITE and 200 OK messages in one sampling interval, but also extending this to ACK and BYE/CANCEL messages. The distribution of these four message types in normal traffic is compared to a distribution under attack conditions. For the comparison and computation of similarity, the Hellinger distance is calculated. It is an intrinsic way to estimate the distance between probability measures and closely related to total variation distance. The attack threshold i.e. which calculated Hellinger distance value indicates an attack, is dynamically adapted based on previous monitored network parameters.

The authors also apply this method for general TCP SYN and RTP message flooding detection. This work does not provide mitigation features.

5.9.2. *Framework*

For attack detection, vFDS should be placed at the ingress network point. The authors have implemented vFDS as an add on to the Linux netfilter kernel component, which is used for testing. The implementation works well to detect flooding rates of 500 INVITE messages/s or 2500 RTP messages/s. Generally, vFDS can detect attacks within several seconds; during tests with a sampling interval of 10 s, the average flooding detection time was 18 s. The addition of vFDS in the Linux router adds only a marginal delay to call setup.

5.10. *Sengar06-2*

5.10.1. *Algorithm*

Sengar et al. (2006b) propose an IDS dubbed "vIDS" based on interacting protocol state machines to detect message flow tampering and DoS flooding attacks.

It is based on the same cross-protocol detection approach as Wu04 and therefore targets the same attacks. While Wu04 monitors pre-defined events for correlation, vIDS is using a full state-machine specification similar to Chen06. Compared to pre-defined events a state-machine specification allows more flexibility in attack detection, however the actual attack detection methods presented by the authors are the same: INVITE message flooding is detected by checking if the number of INVITES exceeds a defined threshold. The BYE message flow tampering attack is detected by synchronising the SIP and RTP state machines. The SIP state machine informs the RTP state machine when its *call tear down* state is reached. The RTP state machine continues then to the *RTP close* state. Hence, later arriving RTP packets are an indication of the attack.

To detect these attacks, attack signatures need to be defined that describe the state flow in the state machines. Theoretical, anomaly-based detection is possible if deviation in the state flow is indicated by the state machines. Also like Wu04, they target only attack detection on UAs and not on the SIP proxy. The authors have not considered attack mitigation.

To avoid the problem that UA end-to-end RTP streams might not be visible, they propose vIDS use only for enterprise networks i.e. it is assumed that all SIP UAs to be protected are in the same network as the main SIP proxy.

5.10.2. *Framework*

VIDS is placed at the ingress point of the protected network where it monitors all passing traffic and evaluates it within its state machines. The authors claim an implementation of vIDS which is not further specified. VIDS has been simulated in a VoIP network using the OPNET network simulator. Simulation with 20 different communicating UAs with an unspecified amount of messages shows only marginal traffic latency overheads of about 100 ms. Also, CPU processing overheads for running vIDS are marginal for the simulation scenario. Thus, the authors extrapolate that vIDS might be able to monitor ‘‘thousands of calls at the same time’’. For performance optimisation, vIDS conserves memory by deleting state information as soon as a call is finished.

5.11. *Nassar06*

5.11.1. *Algorithm*

Nassar et al. (2006) present an IDS concept based on a Bayes inference model (Stigler, 1982) for detecting multiple types of attacks.

The IDS considers SIP signalling attack classes, including multiple-source DoS flooding, SPIT, password cracking and vulnerability scans. Bayesian inference, as used in this work, is a statistical inference in which posterior observations are used to update the probability that a prior hypothesis may be true. Here the authors have developed a Bayes network tree where monitored network events relate to posterior observations. The prior hypothesis states that the traffic belongs to one of the introduced attack classes.

The authors define multiple monitoring parameters, like the number of ACK messages in waiting state, request and response distribution in one sampling interval, etc. Each defined parameter is given a probabilistic value for each attack class, e.g. for the DoS attack class they set P (number of ACK messages in Waiting state $> 10) = 0.9$.

Using the Bayes network tree, the actual monitored traffic is evaluated according to these parameters and the attack class is estimated. The defined probabilities are given as reasonable defaults, however the authors propose to define them from previous SIP traffic observations.

5.11.2. *Framework*

This is a theoretical concept only.

5.12. *Rebahi07*

5.12.1. *Algorithm*

Rebahi (2007), Rebahi et al. (2008) present a method to detect DDoS flooding attacks on VoIP and IMS Systems based on Change-Point Detection with the CUSUM algorithm.

Similar to Reynolds03 the authors use the CUSUM algorithm for DoS flooding detection. While Reynolds correlates the number of INVITE and 200 OK messages, here only the number of INVITE messages are used and analysed whenever a sudden rise of INVITE messages are encountered at the proxy (i.e. the determination of the change-point, where the INVITE rate suddenly increases). Contrary to threshold-based counters, this method takes the current network condition into account. The authors correlate the parametric with the non-parametric application of the CUSUM algorithm.

5.12.2. *Framework*

The authors have verified their method with an off-line-analysis of SIP traffic captured from a SIP VoIP provider. They show that constant-rate flooding attacks are clearly marked as attacks and that, depending on the configuration, attacks with an increasing flooding rate are mostly discernible from regular traffic.

5.13. *Ding07*

5.13.1. *Algorithm*

Ding and Su (2007) present a timed Hierarchical Coloured Petri Net IDS that is built from the work of Wu04 and Sengar06-2.

The authors incorporate both works into their IDS without any modifications. Besides the referenced work the authors only present two ‘‘methods’’ to handle CANCEL message flow tampering attacks. The first proposal is that the callee should simply callback the caller, while in the second one INVITEs should be re-sent after a timeout.[1]

5.13.2. *Framework*

The authors propose a NIDS-based setup at the ingress point. It is unclear if in case of the mentioned CANCEL attack the NIDS is supposed to inject packets on behalf of the attacked UA or the UA should re-send packets itself. The authors claim to have conducted simulations, however the setup is not described.

5.14. *Fiedler07*

5.14.1. *Algorithm*

Fiedler et al. (2007) present a SIP monitoring and security framework dubbed VoIP Defender which can be used for the implementation of security algorithms. Here, the authors do not specify one actual security algorithm.

[1] This ‘‘method’’ however, is based on limited assumptions. The authors assume that after an injected CANCEL by a third party the original sender would not not be notified. However, as the injected CANCEL has to share the same VIA header as the original request, the 200 OK acknowledgement will be sent back to the original sender and not to the injecting party. Besides, the original sender will also receive a 487 Request Terminated response code in regard to its initial INVITE request. So, the ‘‘method’’ would only be applicable if nodes that are not fully RFC3261-compliant are involved. These possibilities have not been analysed by the authors.

5.14.2. *Framework*

The framework is a building block for a dedicated SIP NIDS to be placed at the ingress point of the network. It features a multi-layered approach that can be deployed on multiple machines for scalability reasons. This is to cope especially with high messaging floods.

A prototype implementation exists for Linux operating systems. Network monitoring and control is implemented as a network bridge kernel-module with firewall control, while network analysis is performed in dedicated modules in user-space implemented in C++.

Within tests, the monitoring components can process SIP traffic at a rate of 170 Mbit/s. Performance may degrade depending on the number of applied dynamic firewall rules.

5.15. *Nassar07*

5.15.1. *Algorithm*

Nassar et al. (2007) present a holistic multilayer IDS system to detect multiple different attacks based on a honeypot setup and network event correlation.

A honeypot SIP setup is deployed to lure malicious users with the intention of conducting SPIT or phishing attempts to use this setup. Once in the honeypot, senders are classified and cannot access the real SIP network later on.

An event correlator is used for DoS message flooding and message flow tampering detection. The event correlator is the same pattern-based setup as proposed by Wu04. Likewise, attacks are detected with similar signatures. Additionally, the authors propose anomaly-based detection by generating individual SIP user profiles and detecting deviation from this profile. Flooding attacks are detected by monitoring for short inter-arrival times of requests or by detecting open sessions by monitoring for missing ACK messages.

This method does not provide prevention features, however the authors recommend blocking identified users in the honeypot or flooding requests detected by the event correlator.

5.15.2. *Framework*

This approach proposes a distributed protection approach by deploying a fully operational but fake honeypot SIP in conjunction with the real SIP network protected by the event correlator. The authors propose a distributed IDS i.e. security events should be monitored at multiple places of interest like proxies, gateways or UAs. Instead of monitoring network traffic, events are generated directly by each call agent (e.g. by parsing log messages). Events are correlated at one central controlling instance.

The authors have implemented a prototype using the Simple Event Correlator SEC as a correlator control instance. Attack patterns are fed as SEC signatures written in Perl. One event monitor has been developed for the OpenSer SIP proxy, a fork of SER. The prototype implementation only operates as a local IDS.

The performance has not been tested, however due to SEC's compact signature format the authors are expecting good scalability.

5.16. *Barry07*

5.16.1. *Algorithm*

Barry and Chan (2007) present a combination of the work of Geneiatakis05 and Sengar06-2.

They use a layered approach with two layers. The first countermeasure layer consists of a message checker as proposed by Geneiatakis05. The second layer consists of a cross-protocol state-machine specification as proposed by Sengar06-2. The authors use this system to target the same threats with the same methods.

5.16.2. *Framework*

Contrary to Sengar06-2 the authors propose a host-based intrusion detection system like Wu04 to successfully detect BYE attacks. The authors have tested their framework with a Java implementation using 5 different attacks, consisting of three flow tampering attacks, one INVITE flooding attack and a buffer overflow attack. The implementation was able to detect all five attacks. Performance measurements are not presented.

5.17. *Bouzida08*

5.17.1. *Algorithm*

Bouzida and Mangin (2008) present a data mining NIDS to detect multiple SIP intrusion attacks.

This work can be seen as an extension of Wu04. The authors monitor network traffic statefully and gather several attributes from it. Here attributes are finer-grained events than those introduced by Wu04. Gathered attributes can be message header fields and their values (To, From, Nonce ...), message reply codes or gathered statistical data such as the number of INVITE requests per sampling interval, for example. These attributes are correlated into profile classes (normal, known attack, new condition). In the learning phase profiles are generated using decision trees which can then be applied to actual monitored traffic. If no profile matches (new condition), this can be an indication of a potential new attack. Hence, this work contains both signature-based and anomaly-based detection.

The authors have mostly concentrated on information-gathering attacks (user enumeration) and the usage of this information for fraud attacks (password cracking). DoS flooding attacks against individual users are detected at the proxy by threshold-based counters of different flows to each UA.

5.17.2. *Framework*

The algorithm should be implemented in a standard NIDS. The only requirement for the placement is that is sees all relevant network traffic. Testing has been done off-line with an unspecified testing tool with a 2 h traffic trace file from a real VoIP operator. The detection rate of the reviewed attacks was 99%.

5.18. *Rieck08*

5.18.1. *Algorithm*

Rieck et al. (2008) present an anomaly-based self-learning system to protect against message payload tampering attacks and other potential network intrusion attacks.

Contrary to the work of Geneiatakis05 and Niccolini06-1, the goal of this method is to protect against novel attacks and so-called zero-day exploits. The authors use an anomaly-based system and train it with normal traffic to detect deviations from the normal traffic model. The feature set used for anomaly detection is made up from text strings extracted from each monitored SIP message. All text fragments are concatenated to form a new string over which a sliding window of length *n* (a so-called *n*-gram) is moved. At each position in the string the *n*-gram formed there is saved. The occurrence of each *n*-gram in a message defines the feature vector. The authors calculate the Euclidean distance of the feature vector from a ''normal'' feature vector. With a higher distance the probability of a message payload tampering attack or another potential intrusion increases.

The authors have taken heterogeneous SIP network setups into consideration. In such a setup a comparison to one normal vector might not be sufficient, and thus they propose different normal vectors for comparison. To protect their system from (re-)training set poisoning, they propose the combination of their system with other attack detection tools. This is especially important in the case of DoS flooding attacks as accidental re-training during a DoS attack will yield an inaccurate normal vector.

5.18.2. *Framework*

The authors have not introduced a framework for the deployment of their algorithm. Instead, they have provided performance results with an off-line-analysis of the traces captured in their test beds and from providers. Attacks are generated with a SIP traffic generator. Their unspecified off-line implementation showed good detection of the generated attacks. The false-positive rate was up to 1%. The off-line tool was able to process 70 Mbit/s of SIP messages on ''AMD Opteron Servers''.

5.19. *Nagpal08*

Nagpal et al. (2008) present a framework dubbed ''Secure SIP'' to protect against multiple DoS attacks on the proxy.

5.19.1. *Algorithm 1*

Their first line of defence is a return-routability check to detect proxy flooding from sources with spoofed addresses. The authors use a feature of the SIP specification i.e. each request can be challenged before being served. An initial request will only be served if the challenge is correctly handled by the sender. Thus simple flooding bots that do not implement the SIP specification correctly or use spoofed IP addresses cannot pass this test.

5.19.2. *Algorithm 2*

Their second line of defence is a state-machine specification to trace individual requests. It is aimed at detecting BYE flow tampering attacks caused by spurious BYE requests launched with invalid contact header fields. By following the state-machine specification it can also detect and suppress redundant messages flows, e.g. from misconfigured devices. Finally, in a similar manner to Chen06, it is also able to detect messages that do not follow the SIP state specification.

Simple DoS message flooding is detected through a standard threshold-based counter as described by Iancu03. If the sending rate of one sender exceeds an upper limit, the rate is limited for following requests from that source.

5.19.3. *Framework*

The framework consists of a SIP proxy that has been enhanced to control a firewall at the ingress point through a firewall control protocol. The authors have implemented their framework with the sipd SIP proxy and the hardware firewall Cloudshield CS-2000. The hardware firewall in particular was chosen for scalability reasons, as the handling of dynamic firewall rules can limit the usability of any protection mechanism. The test bed has been extensively tested with an array of 17 SUN servers serving as simultaneous attack generators. While the used SIP proxy could only handle up to 700 requests/s, the hardware firewall could handle more than 17.000 flooding requests/s.

5.20. *Ehlert08*

5.20.1. *Algorithm*

Ehlert et al. (2008) present a flooding detection and prevention mechanism based on the SIP state-machine specification.

The method is an extension to the state-machine specification first presented by Chen06. The model is enhanced by adding multiple statistical measurement points in the state-machine, including measuring the time one transaction stays in one state or how many re-transmissions are encountered in one state. The model needs only 2 different state machines contrary to the 4 machines proposed by Chen06. The statistical measurements are different for flooding and regular SIP traffic, thus flooding and other potentially malicious traffic can be detected. The specification allows the detection of single and multiple-source floods on the SIP proxy. Additionally, redundant message flooding from misconfigured devices can be detected and prevented similarly to Nagpal08-2.

5.20.2. *Framework*

The authors propose a NIDS-based setup at the ingress point of the network. The method is implemented using the VoIP Defender framework (Fiedler07). This allows attack detection, while attack mitigation is also possible by controlling a firewall. The framework has been tested with virtual machines with flooding rates of up to 2000 msg/s. Attack detection latency was generally under 1 s. The limiting factor here is the CPU overhead. Keeping state for a message flow of more than 2500 will bring the test system to its limits.

6. Discussion

Since the original specification of SIP was published in 2002 research on DoS-related attacks on SIP networks has been conducted until the present time. In this work we have described over 20 different DoS related counter-measure mechanisms which we have aligned in Tables 1 and 2 for comparison.[2] We will discuss progress in this research area by

[2] Acronyms used in the tables: SS - Single-Source; MS - Multiple-Source.

considering the different DoS threats from Section 3 individually.

6.1. *Payload attacks*

SIP is a text-based protocol, thus messages are human readable. A sophisticated parser is necessary to translate the human readable message payload into a machine-readable representation. As experience has shown, flaws in such an implementation like buffer overflows or missing integrity checking can result into serious security breaches. It is therefore highly important to protect against payload attacks.

SIP is now a mature standard and the techniques used to prevent payload attacks are well known. Additionally, there are now different tools to check SIP implementations for correct operations (Wieser et al., 2003; Abdelnur et al., 2007), thus any well-established SIP agent should generally be hardened against payload attacks. However, many different parser implementations exist and with SIP's popularity new implementations are constantly becoming available. As it is difficult to check each implementation for correct operation, a viable option for the network operator would be to add another payload attack prevention system in the form of a well-specified and tested message integrity checker as proposed by Geneiatakis05 and Niccolini06-1. This setup is also necessary if network operators are aware of implementation flaws in their devices, but there is no software or firmware update available to fix these flaws. The overhead of an additional message check is generally low as no state information needs to be maintained.

The previously mentioned proposals are signature-based and will detect attacks on known security flaws, like buffer overflows or SQL injection attacks. However, this cannot protect against new security holes. Rieck08 targets this with an anomaly-based payload checker. It is a promising approach and it would be very interesting to evaluate its efficiency in a real SIP provider network.

6.2. *Flow tampering attacks*

Several message types can be used to disrupt individual SIP sessions and these attacks are known as Re-INVITE, CANCEL or BYE attacks, depending on the utilised message type. As shown in this survey, multiple researchers have addressed these attacks, but in the end they are all based on the same cross-protocol stateful correlation work that was first presented by Wu04 and later used by Sengar06-2 and Nassar07. We see several problems with this method.

6.2.1. *Monitoring requirements*

This method relies on cross-protocol SIP/RTP event correlation i.e. an IDS needs to monitor both SIP and RTP traffic. In a general SIP network, RTP traffic is routed end-to-end, hence it would not be visible to a NIDS at the ingress point. This is either addressed by placing multiple monitoring points at all relevant network entities (Wu04, Nassar07) or by limiting protection to devices in the same domain (Sengar06-2). A third option would be to use RTP proxies, especially for NGN infrastructures. However, these options increase either administrative or network overheads or limit the protection to one domain.

6.2.2. *Reactive measures*

The cross-protocol correlation method can only detect an attack, but until now there have been no proposals for preventing these types of attacks. The benefit of a complex detection only system that has to be placed at every end devices is not well motivated, considering that its sole purpose would be to state information (an attack was detected) that is immediately visible in the end device itself (the session is terminated).

6.2.3. *Alternatives*

Flow tampering attacks are only possible if an attacker can sniff necessary network parameters. If the signalling flow is encrypted it is nearly impossible to launch this type of attack. SIP already defines mature and established encryption methods, like Transport Layer Security (TLS) (Dierks and Rescorla, 2006) or IPSec (Kent and Seo, 2005), and support for these methods is increasing in end devices. Instead of countering the effects of an attack, encryption would actually prevent the attack itself. Note that while encryption is an advisable option against flow tampering attacks, it does not help against payload attacks or flooding attacks.

Nagpal08-2 proposes a similar flow integrity checking method to detect such attacks. While it is less accurate than the cross-protocol detection schemes, it only relies on SIP monitoring and thus does not have the RTP monitoring problem. As long as SIP traffic continues to be sent unencrypted, then this seems to be a more viable option. It addresses the UA devices flawed transaction matching algorithms as described by Seedorf et al. (2008).

While a cross-protocol correlator thus has only limited benefits for DoS flow tampering attacks, it is nonetheless a viable option for other flow tampering attacks, especially enumeration and fraud attacks. In fact, Bouzida08 considers these attacks in particular instead of DoS flow tampering attacks in his correlation solution.

6.3. *Flooding attacks*

Flooding attacks are the predominant form of DoS attacks. This is reflected by the research papers listed, where the majority of researchers present methods for handling flooding attacks. In this discussion we will consider only attacks directed at the main proxy and not at user agents, as the protection of the latter can easily be controlled by the proxy if the proxy itself is not attacked.

The established method remains the threshold-based rate-limiting method, introduced by Iancu03 and used in variations in Wu04, Markl05, Niccolini06-2, Nassar06, Bouzida08 and Nagpal08-2. In its simplest form it can be used for flooding detection by counting all incoming messages, regardless of its source. For reactive measures, this mechanism needs to separate counters by considering each source individually. This causes a larger processing overhead and is still only effective against single-source flooding attacks.

Table 1 – Comparison of evaluated approaches, Part I.

	Iancu03	Reynold03	Wu04	Markl05	Geneiatakis05	Ehlert05	Chen06	Niccol.06-1	Niccol.06-2	Sengar06-1	Sengar06-2
Attack	Flooding (SS)	Flooding (SS)	Flow tampering, flooding (SS)	Flooding (SS), Payload tampering	Payload tampering	Flooding (special DNS URIs)	Flooding (MS)	Payload tampering	Flow tampering, flooding (SS)	Flooding (MS)	Flooding (SS), flow tampering
Victim	Proxy	UA	UA, possibly Proxy	Proxy	Proxy, UA	Proxy/DNS resolver	Proxy	Proxy	Proxy	Proxy	UA
Protocol	SIP	SIP	SIP, RTP	SIP	SIP	SIP	SIP	SIP	SIP	SIP, RTP	SIP, RTP
Reaction	Detection, prevention	Detection, prevention	Detection	Detection, prevention	Detection, prevention	Detection, mitigation	Detection	Detection, Prevention	Detection, Prevention	Detection	Detection
Strategy	Patter-based	Anomaly-based	Pattern-based	Pattern-based	Pattern-based	Anomaly-based	Anomaly-based	Pattern-based	Pattern-based	Anomaly-based	Pattern-based
Setup	Extension module	NIDS	NIDS, possibly distributed	NIDS	Extension module	Extension module, HIDS	NIDS	HIDS	see left	NIDS	NIDS
Implem.	C SER extension	Simulation only	Unspecified prototype	SIP signatures for Snort	C Ser extension	C Ser extension; C enhanced Dnsmasq DNS Cache	None	C Snort Extension		C Linux Netfilter Extension	Unspecified Implementation, OPNET simulation
Placem.	Proxy	Ingress point	UA	Ingress point	Proxy	Proxy; arbitrary (DNS Cache)	Ingress point	Ingress point		Ingress point	Ingress point
Measure	Proxy control (ser module)	Proxy control (NIDS)		Firewall control (SnortSam)	Proxy control (Ser module)	DNS Cache		Snort Network Bridge Firewall			
Perform.	Limited at high message floods		Theoretical projection	Same as Snort	Same as Ser	Same as Ser, guaranteed operation on the cost of reachability		Same as Snort		No information, low memory overhead	
Scalability	Same as Ser			Same as Snort	Same as Ser			Same as Snort			

Table 2 – Comparison of evaluated approaches, Part II.

	Nassar06	Rebahi07	Ding07	Fiedler07	Nassar07	Barry07	Bouzida08	Riek08	Nagpal08-1	Nagpal08-2	Ehlert08
Principle	Bayes inference model	Change-point detection (CUSUM)	Same as Sengar06-2	None (just a framework)	Distributed event correlator, honeypot	Same as geneiatakis05 and Sengar06-2	Cross-protocol state model	N-Gram distance calculation	Return-routability check	State machine model	Stateful detection; statistical analysis
Attack	Flooding (MS), other	Flooding (MS)			Flow tampering, flooding (MS)		Flooding (SS), other	Payload tampering, other	Flooding (MS)	Flooding (SS), flow tampering	Flooding (MS)
Victim	Proxy	Proxy			Proxy, UA		Proxy, UA	Proxy, UA	Proxy	Proxy, UA	Proxy
Protocol	SIP	SIP			SIP. RTP, other		SIP, possibly RTP	SIP	SIP	SIP	SIP
Reaction	Detection	Detection			Detection, possibly prevention		Detection	Detection	Detection, prevention	Detection, mitigation	Detection, mitigation
Strategy	Anomaly-based	Anomaly-based			Anomaly-based, pattern-based		Anomaly-based, pattern-based	Anomaly-based	Pattern-based	Pattern-based	Anomaly-based, pattern-based
Setup	NIDS	NIDS	NIDS	NIDS	Distributed HIDS	NIDS	NIDS	Not evaluated	NIDS	see left	NIDS
Implem.	None	Off-line-analysis calculation, provider trace file	Unspecified simulation	C Network Bridge (Linux Kernel), C++ Security Framework	C OpenSer Extension, Perl SEC Signatures	Java prototype	Unspecified Off-line tool	Unspecified off-line tool	HW-firewall CS-2000, enhanced sipd Proxy		Using Fiedler07
Placem.			Ingress point	Ingress point	All SIP entities	All SIP entities	Proxy, UA		Ingress point (proxy inside NIDS)		Ingress Point
Measure				Iptables FW control, transaction level					CS-2000 FW control, transaction level		
Perform.				High message throughput, performance penalty with FW usage	Same as SEC, OpenSer				Very high throughput, high fw processing		
Scalability				Scalable multi-host architecture	Same as SEC			Multiple CPU scaling	Scalable HW design		

This method also depends on the correct setting of the flooding threshold, as there are variations in the traffic load, especially in real time communication scenarios. Firstly, traffic patterns change at different times of the day and on different days of the week as communication during the night is less likely, for example. Secondly, sudden increases in traffic can occur (‘‘flash crowds’’) that are not necessarily caused by a DoS attack. For instance, breaking news can cause a sudden increase in communication. These conditions should be taken into consideration when the threshold is set. Currently, most works consider only a static threshold. Some authors hint at the necessity of dynamic updates of these thresholds. But care has to be taken with traffic poisoning attacks: an attacker can slowly increase its traffic generation load to update a dynamic threshold without raising an alarm. These remain unaddressed questions.

The methods that use change-point detection (Reynolds03, Sengar06-1, Rebahi07) already take dynamic network conditions into account. Both Hellinger-distance calculation and CUSUM computation seem to be viable and resource-friendly ways to detect malicious flooding conditions. This principle has been also used in general IP-based flooding detection, but has some limitation in that case because of the diversity of the different protocols used (Peng et al., 2007). This is, however, not the case in homogeneous SIP environments. The biggest drawback of this method is that it can handle only attack detection.

Another alternative to the threshold-based counters is the evaluation of state machine operations (Chen06, Ehlert08). Through the analysis of a state machine it should be possible to detect attacks more accurately. However, the resource overheads increase considerably, as a lot of state information has to be maintained. Attack mitigation features are limited in the same way as the initial threshold method.

So while attack detection is working sufficiently, attack mitigation work is still limited if multiple-source flooding attacks are considered.

There are currently only two works that address multiple-source flooding mitigation. Ehlert05 is able to mitigate flooding attacks with a non-blocking DNS cache solution. This method is successful because is takes only one special type of multiple-source flooding attack into consideration. It cannot be applied to a general multiple-source flooding scenario.

Nagpal08-2 has introduced a first step towards SIP multiple-source flooding attack mitigation by eliminating floods from spoofed sources. They introduced a return-routability checker i.e. they actively use a dedicated SIP feature for attack mitigation. It has been reported (Handley, 2005) that general IP DDoS attacks with spoofed IP addresses are declining in favour of distributed attacks using zombie bot nets. It remains to be seen if this will also be the case in SIP environments.

So the challenge will be to devise better mitigation schemes against SIP DDoS flooding attacks. This is a daunting task, and much research has already been conducted to handle general IP-based DDoS flooding attacks. However, chances are that mitigation might be more easily handled for SIP networks, as there is much more information available if the SIP payload is also considered by security solutions. This increases the chance to correctly classify flooding SIP traffic.

For example, in IP protection Kulkarni and Bush (2006) argue that legitimate traffic tends to have different properties, while malicious flooding attacks seem to be highly correlated because traffic generators can generate the same packets to the same destination. They propose a Kolomogorov complexity-based algorithm to detect correlated traffic i.e. DoS attack traffic. However one cannot depend on correlated DoS attacks any more if one takes the introduction of bot nets into account as bot nets may consist of different types of bot net members, each of them operating a different attack generator.

Contrarily, in SIP networks, even if bot nets are involved, SIP clients to be captured and controlled are not (yet) common in the infected host. Thus the attack still has to be generated by common attack generators and thus the attack traffic is likely to be highly similar. As SIP is a text-based protocol with multiple header fields it allows easy classification of different message classes (so-called fingerprinting of SIP message generators). Rieck08 demonstrates such a payload attack detection method by extracting all text information from a SIP message for correlation. Another method was proposed by Yan et al. (2006) for SPIT prevention. They combine all SIP message header fields to form a unique fingerprint of the sender. Such methods could also be easily adopted for SIP flooding protection, by only allowing hosts with a known fingerprint class to access the service. Such methods would be a feasible option, especially for SIP providers that also provide a standard SIP client with a known fingerprint class with their service.

Ultimately, protection should only be enforced if an attack condition is be detected. For example, under low traffic conditions the return-routability check would unnecessarily increase latency for all users, and a fingerprinting sensor could falsely deny access to regular users even if no flooding attack is under way. Thus, it is advisable to install a lightweight detection algorithm like a change-point detection algorithm and only activate the mitigation feature if server load increases considerably due to detected flooding traffic.

6.4. *Frameworks*

A protection mechanism has only limited applicability if it does not scale with the amount of traffic encountered in real life attacks. DoS attacks, especially distributed flooding attacks, can generate a high load of traffic at the server which a protection framework should be able to process. Currently only Fiedler07 and Nagpal08 have considered a dedicated protection infrastructure for protection. Most of the remaining ideas have been tested using prototype implementations that have only limited scalability support. Some are also considering protection mechanisms deployed directly at the to be secured host. However, this can easily lead to a self-inflicted DoS, as shown by Luo et al. (2008). A NIDS-based-setup has better scalability.

Fiedler07 proposes a multi-layered architecture. Each security algorithm which uses this framework is split into a scalable part and a non-scalable part. The framework can be deployed on multiple different hosts. It is a software-based solution with its protection based on the Linux iptables firewall component. However, this component has performance limitations with dynamic firewall rule updates. Contrarily, Nagpal08 proposes a hardware-based solution. Thus, the firewall is easily able to dynamically update multiple thousand firewall rules per second. However, the detection intelligence is provided by one SIP proxy instance installed on the firewall. There is no way to scale the algorithm controller.

An interesting idea would thus be to combine the scalable algorithm framework from Fiedler07 with the high-performance firewall from Nagpal08.

7. Conclusions and outlook

As SIP plays an integral part in current and future real time communication networks, protection of SIP networks from different types of attacks is essential. Denial-of-service attacks can impose a serious threat to such networks, hence it comes as no surprise that during the last five years more than 20 proposals that aim to address this problem have been published. There are generally three different classes of DoS attacks against a SIP infrastructure: SIP message payload attacks, SIP message flow tampering attacks and SIP message flooding attacks.

SIP payload attacks can be easily handled by a correct and fail-resistant parser implementation. Together with signatures for known intrusions (like in virus protection) this proves to be an effective protection mechanism against known attacks. Anomaly-based detection is currently also evaluated, with the goal of also protecting against unknown attacks, and its efficiency should be evaluated in real life networks.

Flow tampering attacks can be prevented using message encryption. This seems to be more effective than the current cross-protocol evaluation methods that have several shortcomings. If encryption is not an option, a simple SIP flow sanity checker helps to prevent attacks which target implementation flaws in end devices. Unfortunately, many end devices still suffer from poor SIP implementations.

The biggest challenge remains the protection from SIP DoS flooding attacks. There have been some promising approaches to detect the malicious attacks using change-point detection algorithms, and protection against single-source flooding attacks is more or less possible and viable. The interesting part will be how the research community will address the mitigation problem. Currently, some initial steps have been made towards mitigation against a limited subset of flooding messages (targeting the DNS subsystem) and filtering out fake senders with spoofed IP addresses. We believe there will be many possibilities to define mitigation features in the future based on the wealth of information that every SIP message exposes. This is not the case for general IP-based flooding protection.

Now is also the time to test these methods in real operator networks. Not many researchers have considered scalability issues in their work and this will be a problem, especially with DDoS flooding attacks. A joint test bed project with researchers and SIP providers would show which setup would fare adequately under real life conditions.

Acknowledgement

We would like to thank Lucinda Amelia Dayhew for her excellent work in improving the wording of this article.

REFERENCES

Abdelnur H, Festor O, State R. KiF: a stateful SIP fuzzer. In: Principles, systems and applications of IP telecommunications (IPTComm 2007). New York, USA; July 2007.

AcmePacket Net Net SBC, http://www.acmepacket.com.

Barry BIA, Chan HA. A hybrid, stateful and cross-protocol intrusion detection system for converged applications. In: International conference on grid computing, high-performance and distributed applications (GADA'07). Vilamoura, Portugal; November 2007.

Borderware SIPassure VoIP/SIP Security Solution, http://www.borderware.com/products/sipassure.

Bouzida Y, Mangin C. A framework for detecting anomalies in VoIP networks. In: Third international conference on availability, reliability and security (ARES 08). Barcelona, Spain; March 2008.

Chang RKC. Defending against flooding-based distributed denial-of-service attacks: a tutorial. IEEE Communication Magzine October 2002;40(10):42–51.

Chen EY. Detecting DoS attacks on SIP systems. In: 1st IEEE workshop on VoIP management and security. Vancouver, Canada; April 2006.

Cloudshield CS-2000 Content Processing Platform, http://www.cloudshield.com/Products/cs2000.asp.

Dierks T, Rescorla E. The Transport Layer Security (TLS) protocol version 1.1; April 2006. RFC 4346.

Ding Y, Su G. Intrusion detection system for signal based SIP attacks through timed HCPN. In: 2nd international conference on availability, reliability and security (ARES 2007). Vienna, Austria; April 2007.

Dnsmasq DNS caching Proxy, http://www.thekelleys.org.uk/dnsmasq/doc.html.

Eddy W. TCP SYN flooding attacks and common mitigations; August 2007. RFC 4987.

Ehlert S, Wang C, Magedanz T, Sisalem D. Specification-based denial-of-service detection for SIP Voice-over-IP networks. In: Third international conference on internet monitoring and protection (ICIMP2008). Bucharest, Romania; July 2008.

Fiedler J, Kupka T, Ehlert S, Magedanz T, Sisalem D. VoIP Defender: highly scalable SIP-based security architecture. In: Principles, systems and applications of IP telecommunications (IPTComm 2007). New York, USA; July 2007.

Geneiatakis D, Kambourakis G, Dagiuklas T, Lambrinoudakis C, Gritzalis S. A framework for detecting malformed messages in SIP networks. In: 14th IEEE workshop on local and metropolitan area networks (LANMAN). Chania, Greece; September 2005.

Geneiatakis D, Dagiuklas T, Kambourakis G, Lambrinoudakis C, Gritzalis S, Ehlert S, et al. Survey of security vulnerabilities in Session Initiation Protocol. IEEE Communications Surveys and Tutorials July 2006;8(3):68–81.

Handley M. DoS-resistant Internet – subgroup report. Technical report. Internet Architecture Working Groub: DoS-resistant Internet Working Subgroup, www.communicationsresearch.net/object/download/1543/doc/mjh-dos-summary.pdf; 2005.

Hellinger E. Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen. Journal für die reine und angewandte Mathematik 1909;136(3/4).

Iancu B. SER PIKE excessive traffic monitoring module, http://www.iptel.org/ser/doc/modules/pike; 2003.

Kent S, Seo K. Security architecture for the Internet protocol; December 2005. RFC 4301.

Kuhn R, Walsh TJ, Fries S. Security considerations for Voice over IP systems – recommendations of the National Institute of Standards and Technology. Technical Report SP 800-58. USA: National Institute of Standards and Technology; January 2005.

Kulkarni AB, Bush SF. Detecting distributed denial-of-service attacks using Kolmogorov complexity metrics. Journal of Network and Systems Management March 2006;14(1):69–80.

Larson J, Dawson T, Evans M, Straley JC. Defending VoIP networks from distributed DoS (DDoS) attacks. In: VoIP security – challenges and solutions workshop. Dallas, Texas, USA; December 2004.

Luo M, Peng T, Leckie C. CPU-based DoS attacks against SIP servers. In: IEEE/IFIP network operations and management symposium (NOMS 2008). Salvador, Bahia, Brazil; April 2008.

Markl J, Sisalem D, Ehlert S, Geneiatakis D, Kambourakis G, Dagiuklas T, et al. General reliability and security framework for VoIP infrastructures. Technical report. Deliverable SNOCER-D2.2, http://www.snocer.org; September 2005.

Mirkovic J, Reiher P. A taxonomy of DDoS attacks and defense mechanisms. ACM SIGCOMM Computer Communications Review April 2004;34(2):39–54.

IP Multimedia Subsystem (IMS); Stage 2. Technical Report TS 23.238 (Release 8), 3GPP; 2007.

Nagpal S, Yardeni E, Schulzrinne H, Ormazabal G. Secure SIP: a scalable prevention mechanism for DoS attacks on SIP-based VoIP systems. In: Principles, systems and applications of IP telecommunications (IPTComm 2008). Heidelberg, Germany; July 2008.

Nassar M, State R, Festor O. Intrusion detection mechanisms for VoIP applications. In: Third annual VoIP security workshop. Berlin, Germany; 2006.

Nassar M, Niccolini S, State R, Ewald T. Holistic VoIP intrusion detection and prevention system. In: Principles, systems and applications of IP telecommunications (IPTComm 2007). New York, USA; July 2007.

Niccolini S, Garroppo RS, Giordano S, Risi G, Ventura S. SIP intrusion detection and prevention: recommendations and prototype implementation. In: First IEEE workshop on VoIP management and security. Vancouver, Canada; April 2006.

Opnet – Optimum Network Performance Modeler, http://www.opnet.com.

Page ES. Continuous inspection schemes. Biometrika June 1954; 41(1/2):100–15.

Peng T, Leckie C, Ramamohnarao K. Survey of network-based defense mechanisms countering the DoS and DDoS problems. ACM Computing Surveys April 2007;29(1).

Perl Compatible Regular Expressions, http://www.pcre.org.

Prelude IDS Communication System, http://www.prelude-ids.org.

Rebahi Y, Sisalem D, Kuthan J, Pelinescu-Oncicul A, Iancu B, Janak J, et al. The SIP express router – an open source SIP platform. Guildford, UK: Evolute Workshop, http://www.iptel.org/ser; 2003.

Rebahi Y, Sher M, Magedanz T. Detecting flooding attacks against IP multimedia subsystem (IMS) networks. In: The sixth ACS/IEEE international conference on computer systems and applications (AICCSA-08). Doha, Qatar; March 2008.

Rebahi Y. Change-point detection for Voice over IP denial of service attacks. In: 15. ITG/GI - Fachtagung Kommunikation in verteilten Systemen (KiVS 2007). Bern, Switzerland; February 2007.

Reynolds B, Ghosal D. Secure IP telephony using multi-layered protection. In 10th annual network and distributed system security symposium. San Diego, USA; February 2003.

Rieck K, Wahl S, Laskov P, Domschitz P, Müller KR. A self-learning system for detection of anomalous SIP messages. In: Principles, systems and applications of IP telecommunications (IPTComm 2008). Heidelberg, Germany; July 2008.

Roesch M. Snort – lightweight intrusion detection for networks. In: 13th USENIX large installation system administration conference (LISA'99). Seattle, USA; November 1999.

Rosenberg J. A Hitchhiker's guide to the Session Initiation Protocol (SIP). IETF; February 2008. Interet Draft draft-ietf-sip-hitchhikers-guide-05, Work in Progress.

Rosenberg J, Jennings C. The Session Initiation Protocol (SIP) and spam; January 2008. RFC 5039.

Rosenberg J, Schulzrinne H, Camarillo G, Johnston A, Peterson J, Spark R, et al. Session Initiation Protocol; 2002. RFC 3261.

Schulzrinne H, Casner S, Frederick R, Jacobson V. RTP: a transport protocol for real-time applications; July 2003. RFC 3550.

Sec – Simple Event correlator, http://www.estpak.ee/risto/sec.

Seedorf J, Beckers K, Huici F. Testing dialog-verification of SIP phones with single-message denial-of-service attacks. In: Fourth international conference on global E-security. London, UK; June 2008.

Sekar R, Gupta A, Frullo J, Shanbhag T, Tiwari A, Yang H, Zhou S. Specification-based anomaly detection: a new approach for detecting network intrusions. In: Ninth ACM computer and communication security conference (CCS 2002). Washington DC, USA; November 2002.

Sengar H, Wijesekera D, Wang H, Jajodia S. Fast detection of denial of service attacks on IP telephone. In: 14th IEEE international workshop on quality of service. New Haven, CT, USA; June 2006.

Sengar H, Wijesekera D, Wang H, Jajodia S. VoIP intrusion detection through interacting protocol state machines. In: International conference on dependable systems and networks (DSN-2006). Philadelphia, USA; June 2006.

Sengar H, Wang H, Wijesekera D, Jajodia S. Detecting VoIP floods using the hellinger distance. IEEE Transactions on Parallel and Distributed Systems June 2008;19(6):794–805.

sipd – Columbia InterNet Extensible Multimedia Architecture SIP proxy, December, http://www.cs.columbia.edu/IRT/cinema; 2002.

Sisalem D, Kuthan J, Ehlert S. Denial of service attacks targeting a SIP VoIP infrastructure – attack scenarios and prevention mechanisms. IEEE Network – Special Issue on Securing VoIP September 2006;20(5):26–31.

SnortSam. Firewall Control Plugin for the Snort IDS system, http://www.snortsam.org.

Stigler SM. Thomas Bayes' Bayesian inference. Journal of the Royal Statistical Society, Series A (General) 1982;145(2).

Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN). NGN Security (SEC). Requirements. Technical Report ETSI TS 187 001 V1.1.1. ETSI TISPAN; March 2006.

Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN). NGN functional architecture. Technical Report ETSI ES 282 001 V2.0.0. ETSI TISPAN; March 2008.

Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN). NGN Security (SEC). Threat, vulnerability and risk analysis. Technical Report ETSI TS 187 002 V1.2.1. ETSI TISPAN; March 2008.

Tzvetkov V, Zuleger H. Service provider implementation of SIP regarding security. In: 21st international conference on advanced information networking and applications workshops (AINAW 2007). Niagara Falls, Canada; May 2007.

VoIP security and privacy threat taxonomy. Technical Report Public Release 1.0. VOIPSA, http://www.voipsa.org; October 2005.

Wang H, Zhang D, Shin K. Detecting SYN flooding attacks. 2002. New York, USA: IEEE INFOCOM; June 2002.

Wieser C, Laakso M, Schulzrinne H. Security testing of SIP implementations. Technical Report CUCS-024-03. New York, USA: Columbia University, Department of Computer Science, http://www.ee.oulu.fi/research/ouspg/protos/testing/c07/sip; 2003.

Wu YS, Bagchi S, Garg S, Singh N, Tsai T. SCIDIVE: a stateful and cross protocol intrusion detection architecture for Voice-over-IP environments. In: International conference on dependable systems and networks (DSN-2004). Firenze, Italy; July 2004.

Yan H, Sripanidkulchai K, Zhang H, Shae Z. Incorporating active fingerprinting into SPIT prevention systems. In: Third international VoIP security workshop. Berlin, Germany; June 2006.

Zhang G, Ehlert S, Magedanz T, Sisalem D. Denial of service attack and prevention on SIP VoIP infrastructures using DNS flooding. In: Principles, systems and applications of IP telecommunications (IPTComm 2007). New York, USA; July 2007.

Zhang R, Wang X, Yang X, Jiang X. Billing attacks on SIP-based VoIP systems. In: First USENIX workshop on offensive technology (WOOT '07). Boston, USA; August 2007.

Sven Ehlert is the head of the security reserach staff of the "Next Generation Network Integration" divison of the Fraunhofer Institute FOKUS in Berlin, Germany. He has lead two international research projects in the field of SIP security and has published several refereed scientific papers in the security field. Sven Ehlert received his M.Sc in Computer Science from the Technische Universität Berlin.

Dimitris Geneiatakis (PhD) was born in Athens, Greece, in 1981. He received the five-year Diploma in Information and Communication Systems Engineering in 2003, and the M.Sc. in Security of Information and Communication Systems in 2005, and a Ph. D. in the field of Information and Communication Systems Security from the department of Information and Communications Systems Engineering of the University of Aegean, Greece. He has participated in various national and international projects in the area of Information Systems Security. His current research interests are in the areas of security mechanisms in Internet Telephony, Smart Cards, Intrusion Detection Systems and Network Security. He is an author of several refereed papers in international scientific journals and conference proceedings. He has served on program and organizing committees of international conferences on Informatics and is a reviewer for several scientific journals. Dimitris Geneiatakis is a member of the Technical Chamber of Greece.

Thomas Magedanz (PhD) is full professor in the electrical engineering and computer sciences faculty at the Technische Universität Berlin, Germany, leading the chair for next generation networks. In addition, he is director of the "Next Generation Network Integration" division of the Fraunhofer Institute FOKUS, which also provides a national Next Generation Network test bed in Germany. Since more than 18 years he is working in the convergence field of fixed and mobile telecommunications, the internet and information technologies, which resulted in many industry driven R&D projects centred around Next Generation Service Delivery platforms. In the course of his research activities he published more than 200 technical papers/articles. In addition, Prof Magedanz is senior member of the IEEE, and editorial board member of several journals.