University of Piraeus

Department of Digital Systems

M.Sc. in «Security of Digital Systems»

Master's Thesis:
"IDPS for IMS and VoIP Services"

Mouratos Georgios

Supervisor: Prof. Konstantinos Lamprinoudakis

Piraeus, December 2012

Acknowledgements

Special thanks to Miltiadis and Dimitris for their invaluable help, and my family for their support all those years. I would, also, like to thank Mr. Lamprinoudakis for his patience and hard work that he did all this time. So, to his assistant, Nikolaos Vrakas, who helped me understand and analyze all the aspects of the topic we dealt with.

This worksheet is dedicated to Spyridoula.

Contents

Contents

Contents

Table of figures

1. Introduction

The multimedia services provided through the internet have become an inseparable fact in people's life. In addition, the development of the technology has given the opportunity for such services through mobile devices and handheld devices. This is achieved with the deployment of the IMS [1]. The high resource demanding services that the IMS provides such as video conferences, audio calls, applications, IP television and many more, must be streamed with high Quality of Service (QoS). Considering QoS, these infrastructures are employing a lightweight signaling protocol; SIP [2]. This text based protocol is flexible enough to easily incorporate and provide different services. It is also a low resource demanding protocol without burdening the infrastructure with further delays during the session establishment handshakes.

These advantages also have an inevitable drawback; there are many security vulnerabilities that can be exploited by malicious internal or external users in order to degrade the QoS causing Denial of Service (DoS), intercept the communication sessions, steal user's identities and credentials, utilizing different techniques. Moreover, the attacker can utilize techniques form the lower layers of the internet protocol stack in order to threaten SIP services. For instance, IP spoofing [3] or Address Resolution Protocol (ARP) poising [4] can be first step for an attacker in order to be able to manipulate a SIP request. Every architecture that utilizes SIP as signalling protocol is susceptible to such behaviours. Many scientific works pinpoint these vulnerabilities [5-7].

In VoIP and IMS environments are deployed different security protocols hardening the defence against the above mentioned behaviours. For instance, in IMS it can be utilized the Authentication and Key Agreement (AKA) with IPSec [8], or the SIP Digest with TLS [8]. These protocols provide authentication, confidentiality and integrity services to the communication. Also the SIP Digest can be utilized for low resource enabled devices [9] but it provides only authentication support to SIP messages. Nevertheless, these mechanisms can prevent the most of the attacks that originated by external users but they cannot effectively discourage malicious subscribers to launch flooding or SIP signaling attacks through their security tunnels.

Many researchers have presented scientific works towards the detection of such security incidents but the most of them cover only a minority of the attacks [10-12], or stay only in detection without being unable to prevent them [12-14] or even they utilize heavy weight protocols such as Public Key Infrastructures (PKIs) with a trade off between security and the introduced delay.

Also, other solutions such as Transport layer Security (TLS) and (Secure Multi-Purpose Internet Mail Extensions) (S/MIME) cannot deter internal flooding and signalling attacks and on top of that, they introduce large amount of overhead [16] while the throughput can be 17 times greater (in the worst case scenario) without utilizing TLS [17].

In this paper we present, to the best of our knowledge the most comprehensive and thorough cross layer mechanism that is able to detect and deter most of the attacks that can be launched against environments which use the SIP. Furthermore, it considered lightweight since it does not utilize any strong security protocols or executes expensive mathematical calculations. Finally, it is also transparent to system's and user's operation while no modifications are required to take place to any of them and can be easily deployed in both IMS and VoIP infrastructures. Specifically, all the messages are first checked for their originality using the data

gathered from layer 2, 3 and 5 of the internet protocol stuck. If the SIP registration message is authenticated then a bind is created which correlates information of the protocols which has been involved in the communication. This bind is stored into a table with the help of a bloom filter. All the incoming messages are checked for originality against the entries of this table. However, the non spoofed messages are not always legitimate since there are flooding attacks which can be launched without forged SIP request. Thus, a second statistical module is utilized that detects deviations among all bindings with respect to their traffic behaviour. When a binding falls into a rule that has been created during the training period of the mechanism, then the corresponding messages are dropped and their sources are blacklisted. It is worth noting that the mechanism is not restricted only in detecting INVITE or REGISTER message floods but also it can effectively deter floods which launched with any of the SIP's available request methods. We also introduce a classification of flooding attacks in order to illustrate and detect all the different cases that the intrusion detection and prevention system may have to confront, in VoIP/IMS environments.

This work is structured as following: In Chapter 2 and 3 the VoIP architecture and the IMS architecture is being reviewed, respectively. In chapter 4, attacks and security vulnerabilities in SIP protocol are demonstrated and, finally, in chapter 5 the IDPS for VoIP-IMS is being structured and overviewed. In Chapter 6 we mention the related work as we come to a conclusion in Chapter 7.

2. VoIP Architecture

Two major VoIP and multimedia suites dominate today: SIP and H.323. Others (like H.248) exist, but these are the two major players. For simplicity, we will define SIP and H.323 as signalling protocols. However, whereas H.323 explicitly defines lower level signalling protocols, SIP is really more of an application-layer control framework. The SIP Request line and header field define the character of the call in terms of services, addresses, and protocol features.
Voice media transport is almost always handled by RTP and RTCP, although SCTP (Stream Control Transmission Protocol) has also been proposed and ratified by the IETF (and is used for the IP version of SS7, known as SIGTRAN).The transport of voice over IP also requires a large number of supporting protocols that are used to ensure quality of service, provide name resolution, allow firmware and software upgrades, synchronize network clocks, efficiently route calls, monitor performance, and allow firewall traversal.

2.1 SIP architecture

SIP is a signalling protocol for Internet conferencing, telephony, presence, events notification, and instant messaging. SIP is an IETF-ratified response-request protocol whose message flow closely resembles that of HTTP. SIP is a framework in that its sole purpose is to establish sessions. It doesn't focus on other call details. SIP messages are ASCII encoded.

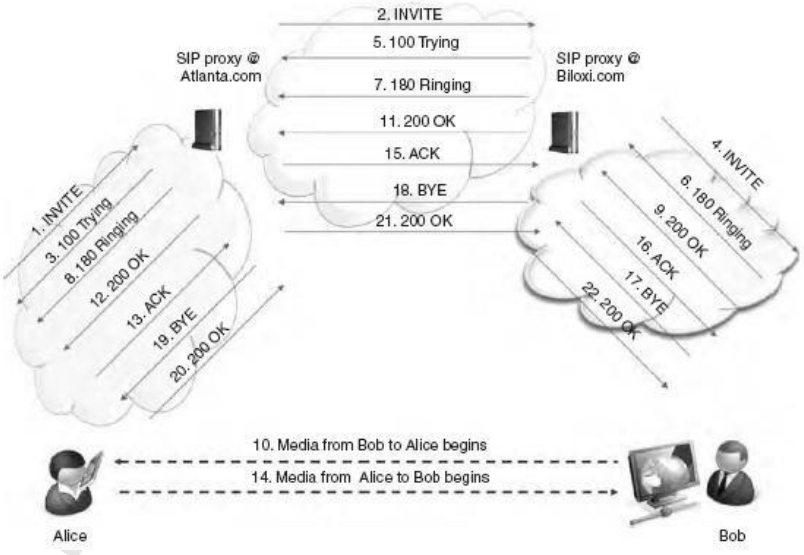


Figure 1. SIP-based VoIP Network

In SIP architecture three main entities are identified:

- **User Agents (UA):** These are the end devices. They are only used to initialize a session. A regular UA can be a soft or hard phone (i.e. software or hardware endpoints) as well as a gateway that connects to other VoIP protocols or the PSTN. UA can be divided in two logical entities: User Agent Client (UAC), which is the one in charge of initiating the request, and the User Agent Server (UAS) which is the one responsible for generating the responses to the received requests.

- **SIP Servers:** These services are not mandatory to establish a session between two SIP UA devices but they provide a vast range of extra functionalities to make it easier. According to their functionality, SIP servers can be subclassified as follows.

– SIP Registrar Server: It is used from a UA in order to register in a SIP domain address. The Server obtains the UA's IP address as well as the associated user and stores them for future use.
– SIP Proxy Server: It is used to forward requests on behalf of other SIP entities. It can not initiate a request by itself, but it can offer additional services like for instance security, authentication and authorization.
– SIP Redirect Server: It is used to redirect the caller to the searched UA. The difference with respect to the proxy is that the Redirect Server tells the entity the contact address (of the UA) rather than forward requests itself. The redirect server is also able to retrieve multiple locations in order to fork the call.

- **SIP Location Server:** It is used to keep a database of the users containing their URLs, IP address, features and other preferences. It is used by the SIP Servers to allow an application level mobility.

2.2. H.323 architecture

H.323, on the other hand, is an ITU protocol suite similar in philosophy to SS7. The H.323 standard provides a foundation for audio, video, and data communications across IP-based networks, including the Internet. Carriers tend to prefer H323 because the methods defined by H.323 make translation from ISDN or SS7 signalling to VoIP more straightforward than for SIP. SIP, on the other hand, is text-based, works better with IM, and typically is implemented on less expensive hardware. H.323 has been the market leader, but SIP rapidly is displacing H.323.
In H.323 there are four main elements that are explained below:

Terminals: H.323 terminals are LAN-based end points for voice transmission. Some common examples of H.323 terminals are a PC running Microsoft NetMeeting software and an Ethernet-enabled phone. All H.323 terminals support real-time, 2-way communications with other H.323 entities.

Gateways: The gateway serves as the interface between the H.323 and non-H.323 network. On one side, it connects to the traditional voice world, and on another side to packet-based devices. As the interface, the gateway needs to translate signalling messages between the two sides as well as compress and decompress the voice. A

prime example of a gateway is the PSTN/IP gateway, connecting an H.323 terminal with the SCN (Switched Circuit Network).

Gatekeeper: The gatekeeper is not a mandatory entity in an H.323 network. However, if a gatekeeper is present, it must perform a set of functions. Gatekeepers manage H.323 zones, logical collection of devices (for example: all H.323 devices within an IP subnet).

Multipoint Control Unit (MCU): MCU's allow for conferencing functions between three or more terminals. Logically, an MCU contains two parts:

- Multipoint controller (MC) that handles the signalling and control messages necessary to setup and manage conferences.
- Multipoint processor (MP) that accepts streams from endpoints, replicates them and forwards them to the correct participating endpoints.

An MCU can implement both MC and MP functions, in which case it is referred to as a centralized MCU. Alternatively, a decentralized MCU handles only the MC functions, leaving the multipoint processor function to the endpoints.

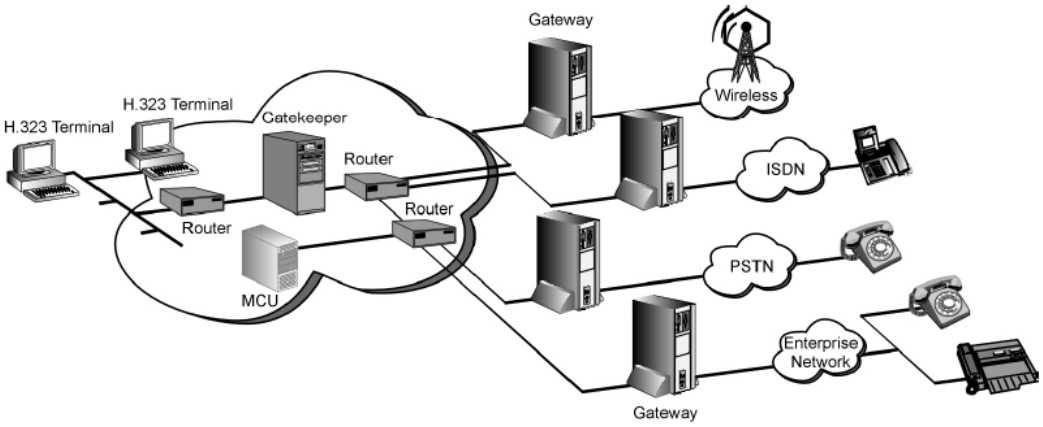


Figure 2. H.323-based Network

2.3. Structure of a SIP Message

The following is the format of INVITE request as sent by user1.

INVITE sip:user2@server2.com SIP/2.0
Via: SIP/2.0/UDP pc33.server1.com;branch=z9hG4bK776asdhds Max-Forwards: 70
To: user2 <sip:user2@server2.com>
From: user1 <sip:user1@server1.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.server1.com
CSeq: 314159 INVITE
Contact: <sip:user1@pc33.server1.com>
Content-Type: application/sdp
Content-Length: 142
---- User1 Message Body Not Shown ----

The first line of the text-encoded message is called Request-Line. It identifies that the message is a request.

Request-Line

Method SP Request-URI SP SIP-Version CRLF
[SP = single-space & CRLF=Carriage Return + Line Feed (i.e. the character inserted when you press the "Enter" or "Return" key of your computer)]
Here method is INVITE, request-uri is "user2@server2.com" and SIP version is 2.
The following lines are a set of header fields.

Via:
It contains the local address of user1 i.e. pc33.server1.com where it is expecting the responses to come.

Max-Forward:
It is used to limit the number of hops that this request may take before reaching the recipient. It is decreased by one at each hop. It is necessary to prevent the request from traveling forever in case it is trapped in a loop.

To:
It contains a display name "user2" and a SIP or SIPS URI <user2@server2.com>

From:
It also contains a display name "user1" and a SIP or SIPS URI <user1@server1.com>.
It also contains a tag which is a pseudo-random sequence inserted by the SIP application. It works as an identifier of the caller in the dialog.

Call-ID:
It is a globally unique identifier of the call generated as the combination of a pseudo-random string and the softphone's IP address.
The Call-ID is unique for a call. A call may contain several dialogs. Each dialog is uniquely identified by a combination of From, To and Call-ID.

CSeq:
It contains an integer and a method name. When a transaction starts, the first message is given a random CSeq. After that it is incremented by one with each new message. It is used to detect non-delivery of a message or out-of-order delivery of messages.

Contact:
It contains a SIP or SIPS URI that is a direct route to user1. It contains a username and a fully qualified domain name (FQDN). It may also have an IP address.
Via field is used to send the response to the request. Contact field is used to send future requests. That is why the 200 OK response from user2 goes to user1 through proxies. But when user2 generates a BYE request (a new request and not a response to INVITE), it goes directly to user1 bypassing the proxies.

Content-Type:
It contains a description of the message body (not shown).

Content-Length:
It is an octet (byte) count of the message body.

The header may contain other header fields also. However those fields are optional. Please note that the body of the message is not shown here. The body is used to convey information about the media session written in Session Description Protocol (SDP). You may continue your journey through SIP without worrying about SDP right now. However it doesn't hurt to take a peep.

Your SIP request is waiting to get a SIP response message.

Request Methods – Response Codes

In the figure below, the list of the request methods in detailed description is presented.

Request name	Description
INVITE	Indicates a client is being invited to participate in a call session.
ACK	Confirms that the client has received a final response to an INVITE request.
BYE	Terminates a call and can be sent by either the caller or the callee.
CANCEL	Cancels any pending request.
OPTIONS	Queries the capabilities of servers.
REGISTER	Registers the address listed in the To header field with a SIP server.
PRACK	Provisional acknowledgement.
SUBSCRIBE	Subscribes for an Event of Notification from the Notifier.
NOTIFY	Notify the subscriber of a new Event.
PUBLISH	Publishes an event to the Server.
INFO	Sends mid-session information that does not modify the session state.
REFER	Asks recipient to issue SIP request (call transfer.)
MESSAGE	Transports instant messages using SIP.
UPDATE	Modifies the state of a session without changing the state of the dialog.

Figure 3. List of the request methods

2.4. List of Response codes

Class	Description	Action
1xx	Informational	Indicates status of call prior to completion. If first informational or provisional response.
2xx	Success	Request has succeeded. If for an INVITE, ACK should be sent; otherwise, stop retransmissions of request.
3xx	Redirection	Server has returned possible locations. The client should retry request at another server.
4xx	Client error	The request has failed due to an error by the client. The client may retry the request if reformulated according to response.
5xx	Server failure	The request has failed due to an error by the server. The request may be retried at another server.
6xx	Global failure	The request has failed. The request should not be tried again at this or other servers.

Figure 4. SIP Response Classes

In this section are demonstrated the most common response codes and what they represent. For further examination and more analytical view of all the response codes, there are in the appex review.

100 Trying
This special case response is only a hop-by-hop request. It is never forwarded and may not contain a message body. A forking proxy must send a 100 Trying response, since the extended search being performed may take a significant amount of time. This response can be generated by either a proxy server or a user agent. It only indicates that some kind of action is being taken to process the call—it does not indicate that the user has been located. A 100 Trying response typically does not contain a To tag.

180 Ringing
This response is used to indicate that the INVITE has been received by the user agent and that alerting is taken place. This response is important in interworking with telephony protocols, and it is typically mapped to messages such as an ISDN Progress or ISUP Address Complete Message (ACM) [2]. When the user agent answers immediately, a 200 OK is sent without a 180 Ringing; this scenario is called the "fast answer" case in telephony. A message body in this response could be used to carry QoS or security information, or to convey ring tone or animations from the UAS to the UAC. A UA normally generates its own ring back tone or remote ringing indication, unless a Alert-Info header field is present.

200 OK
The 200 OK response has two uses in SIP. When used to accept a session invitation, it will contain a message body containing the media properties of the UAS (called party). When used in response to other requests, it indicates successful completion or receipt of the request. The response stops further retransmissions of the request. In response to an OPTIONS, the message body may contain the capabilities of the server. A message body may also be present in a response to a REGISTER request. For 200 OK responses to CANCEL, INFO, MESSAGE, SUBSCRIBE, NOTIFY, and PRACK, a message body is not permitted.

401 Unauthorized
This response indicates that the request requires the user to perform authentication. This response is generally sent by a user agent, since the 407 Proxy Authentication Required (Section 5.4.8) is sent by a proxy that requires authentication. The exception is a registrar server, which sends a 401 Unauthorized response to a REGISTER message that does not contain the proper credentials. An example of this response is:

SIP/2.0 401 Unathorized
Via: SIP/2.0/UDP proxy.globe.org:5060;branch=z9hG4bK2311ff5d.1
;received=192.0.2.1
Via: SIP/2.0/UDP 173.23.43.1:5060;branch=z9hG4bK4545
From: <sip:explorer@geographic.org>;tag=341323
To: <sip:printer@maps-r-us.com>;tag=19424103
From: Copernicus <sip:copernicus@globe.org>;tag=34kdilsp3
Call-ID: 123456787@173.23.43.1
CSeq: 1 INVITE
WWW-Authenticate: Digest realm="globe.org",
nonce="8eff88df84f1cec4341ae6e5a359", qop="auth",
opaque="", stale=FALSE, algorithm=MD5
Content-Length: 0

The presence of the required WWW-Authenticate header field is required to give the calling user agent a chance to respond with the correct credentials. Note that the follow-up INVITE request should use the same Call-ID as the original request as the authentication may fail in some cases if the Call-ID is changed from the initial request to the retried request with the proper credentials.

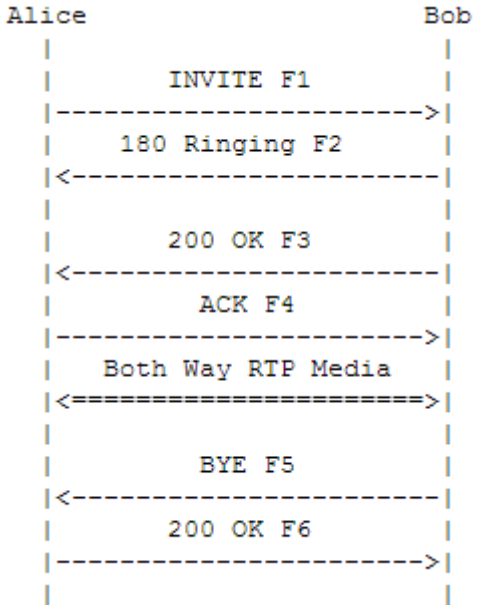
404 Not Found
This response indicates that the user identified by the sip or sips URI in the Request-URI cannot be located by the server, or that the user is not currently signed on with the user agent.

480 Temporarily Unavailable
This response indicates that the request has reached the correct destination, but the called party is not available for some reason. The reason phrase should be modified for this response to give the caller a better understanding of the situation. The response should contain a Retry-After header indicating when the request may be able to be fulfilled. For example, this response could be sent when a telephone has its ringer turned off, or a "do not disturb" button has been pressed. This response can also be sent by a redirect server. [15]

2.5. SIP Session Establishment

This section details session establishment between two SIP User Agents (UAs): Alice and Bob. Alice (sip:alice@atlanta.example.com) and Bob sip:bob@biloxi.example.com) are assumed to be SIP phones or SIP-enabled devices. The successful calls show the initial signalling, the exchange of media information in the form of SDP payloads, the establishment of the media session, then finally the termination of the call.

Successful Session Establishment



In this scenario, Alice completes a call to Bob directly.

Message Details

F1 INVITE Alice -> Bob

INVITE sip:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:alice@client.atlanta.example.com;transport=tcp>
Content-Type: application/sdp
Content-Length: 151

F2 180 Ringing Bob -> Alice

SIP/2.0 180 Ringing
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
;received=192.0.2.101
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>;tag=8321234356
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:bob@client.biloxi.example.com;transport=tcp>
Content-Length: 0

F3 200 OK Bob -> Alice

SIP/2.0 200 OK
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
;received=192.0.2.101
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>;tag=8321234356
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:bob@client.biloxi.example.com;transport=tcp>
Content-Type: application/sdp
Content-Length: 147

F4 ACK Alice -> Bob

ACK sip:bob@client.biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bd5
Max-Forwards: 70
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>;tag=8321234356
Call-ID: 3848276298220188511@atlanta.example.com

CSeq: 1 ACK
Content-Length: 0

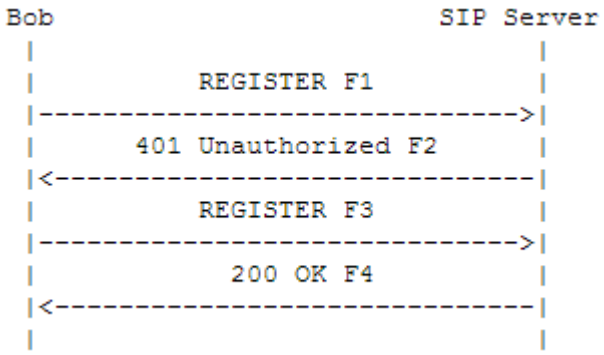
F5 BYE Bob -> Alice

BYE sip:alice@client.atlanta.example.com SIP/2.0
Via: SIP/2.0/TCP client.biloxi.example.com:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
From: Bob <sip:bob@biloxi.example.com>;tag=8321234356
To: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 BYE
Content-Length: 0

F6 200 OK Alice -> Bob

SIP/2.0 200 OK
Via: SIP/2.0/TCP client.biloxi.example.com:5060;branch=z9hG4bKnashds7
;received=192.0.2.201
From: Bob <sip:bob@biloxi.example.com>;tag=8321234356
To: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 BYE
Content-Length: 0

Successful New Registration



Bob sends a SIP REGISTER request to the SIP server. The request includes the user's contact list. This flow shows the use of HTTP Digest for authentication using TLS transport. TLS transport is used due to the lack of integrity protection in HTTP Digest and the danger of registration hijacking without it, as described in RFC 3261. The SIP server provides a challenge to Bob. Bob enters her/his valid user ID and password. Bob's SIP client encrypts the user information according to the challenge issued by the SIP server and sends the response to the SIP server. The SIP server validates the user's credentials. It registers the user in its contact database and returns a response (200 OK) to Bob's SIP client. The response includes the user's current contact list in

Contact headers. The format of the authentication shown is HTTP digest. It is assumed that Bob has not previously registered with this Server.

3. IMS (IP Multimedia Subsystem)

3.1. Introduction

The IP Multimedia Subsystem (IMS) is a key enabler for the convergence of fixed and mobile communications — devices, networks, and services. It is designed to allow the gradual migration of existing core infrastructures to a new IP framework that enables the easy and cost-effective launch of new services and can substantially reduce operating costs, providing benefits to both subscribers and service providers. To enable person-to-person and person-to-content communications, IMS uses a layered architecture in which service enablers and common functions can be reused for multiple applications. This horizontal approach involves a plethora of gateways and media servers. The first layer translates the bearer and signalling channels of traditional networks to packet-based streams and controls. The second provides elementary media functions to the higher-level applications. In addition, IMS uses a higher level of application services and API gateways to allow third parties to take control of call sessions and access subscriber preferences. The horizontal nature of IMS provides an opportunity for system and application enablers, such as Dialogic, to direct their rich web-based development environments and platforms in line with this paradigm. As a result, taking advantage of our deep understanding of media processing, flow management, signalling, and provisioning, Dialogic is making new offerings available to service providers. At the same time, our family of media servers and gateways and our system building blocks allow equipment providers and developers to economically build modular, highly available, scalable solutions for their service provider customers.

3.2. IMS Architecture Overview

The IMS architecture gives service providers the opportunity to deliver new and better services, with reduced operating costs, across wireless, wire line, and broadband networks. IMS is defined by the Third Generation Partnership Project (3GPP) and supported by major Network Equipment Providers (NEPs) and service providers. IMS unifies applications enabled by the Session Initiation Protocol (SIP) to connect traditional telephony services and non-telephony services, such as instant messaging, push-to-talk, video streaming, and multimedia messaging.

The IMS architecture gives service providers the opportunity to deliver new and better services, with reduced operating costs, across wireless, wire line, and broadband networks. IMS is defined by the Third Generation Partnership Project (3GPP) and supported by major Network Equipment Providers (NEPs) and service providers. IMS unifies applications enabled by the Session Initiation Protocol (SIP) to connect traditional telephony services and non-telephony services, such as instant messaging, push-to-talk, video streaming, and multimedia messaging. The IMS architecture involves a clear separation of three layers:

- Transport and Endpoint
- Session and Control
- Application Services

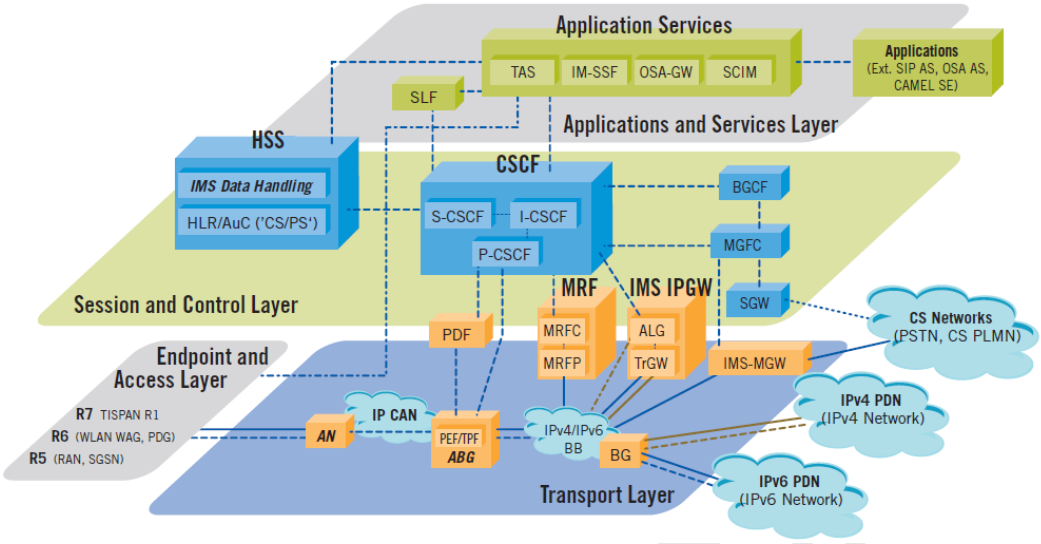


Figure 5. IMS Architecture

3.2.1 Application Layer

Application layer consist of the Application Servers capable of providing verity of services including text, VoIP, multimedia and video services etc. The main elements in this layer are Home Subscriber Server HSS, Subscriber Location Function SLF and Application Server AS. AS include Open Service Access-Service Capability Server OSA-SCS, IP Multimedia Service Switching Function IM-SSF and SIP Application Server SIP-AS.

3.2.1.1. Home Subscriber Server (HSS)

HSS is a data base server that contains the information about the end users. It contains all the user related information or user profiles including location base information, security profiles, user base services information i.e. what services a user is entitled to? HSS controls the user call and session using their profile information stored in its database such as user location information is used for mobility management and security information is used for user authorization. Similarly user privilege profiles are used to allow or deny user for a specific service it requests.
There can be more than one HSS in a single IMS network. If the number of users is quite huge to be handled by a single server or users are having complex and heavily populated profiles or even for redundancy purpose there can be multiple HSS servers. HSS communicates with serving call session control function S-CSCF using Diameter protocol defined by 3GPP.

3.2.1.2. Subscriber Location Function SLF

SLF is a data base containing information about HSS locations and the user addresses who's profile information are stored in that particular HSS.
It is only used when there is a case of multiple HSS servers in a network. SLF keeps track of the HSS servers and the user's profiles it contains. In other words its function resembles to the function of DNS server in internet. The DNS server maps IP

addresses against the domain names similarly in IMS, SLF maps the user address with the HSS server in the network, that is which user‟s information are stored in which HSS server.

3.2.1.3. Application Server AS

AS provide application services including IP telephony, multimedia applications, voice call and video conferencing applications etc. It uses SIP to communicate and provides the service applications that a user requests for (if the user is entitled to use the requested services). There can be a verity of application servers in IMS network for the ease of management, each dedicated to a specific family of services like web servers, VoIP server, multimedia servers etc.

3.2.1.3.1. SIP Application Server SIP-AS

SIP-AS is a SIP protocol enabled application server that offers the multimedia services and multimedia applications. A very fast and rapid development for IMS services is in progress due to the non standardized approach of IMS towards the user services and service applications. All these new services will be implemented on SIP-AS server.

3.2.1.3.2. IP Multimedia Service Switching Function IMS-SSF

As it is clear from the name IMS SSF is an application server that act as switching entity between the GSM supported multimedia applications and IMS services. For this backward compatibility purpose, Customized Applications for Mobile Enhanced Logic CAMEL is implemented on IMS-SSF application server. CAMEL operates with CAP protocol [21]. IMS-SSF has two interfaces, one with Serving Call Session Control Function S-CSCF signalling on SIP and the other is with traditional GSM multimedia service that attempts to access IMS applications using CAP protocol.

3.2.1.3.3. Open Source Access-Service Capability Server OSA-SCS

In some cases the IMS user requests to access some services that are hosted by a third party or an external Application Server. In that case OSA-SCS which itself is also an application server provide connection between IMS S-CSCF and the external application server in a secure manners. In short OSA-SCS is an application server that is responsible to provide a secure link between S-CSCF of an IMS network and any 3rd party application server.

3.2.2. Session and Control Layer

In session control layer there are numerous core elements that control, manage and initiate or dismiss a session. The user authentication, authorization, billing and resource allocation all functions take place in control layer.

3.2.2.1. Proxy Call Session Control Function P-CSCF

P-CSCF is the contact point between User Equipment UE and IM network; in fact it is the first element of IMS network that is exposed to EU. P-CSCF is a SIP enabled proxy server and all user requests, signalling and control information passes through it.

During the IMS registration process P-CSCF address information and its allocation to user is taken care of.

- As it is mentioned earlier that the first IMS element that a UE make contact with is P-CSCF so it is obvious that the security parameters (shared Authentication and Key Agreement AKA) and security associations SAs must be negotiated and agreed on at this point between P-CSCF and UE for IPSec and Encapsulated Security Payload ESP
- Once the SAs are agreed between these two entities, serving call session control function (S-CSCF) is informed by P-CSCF about the security associations established with UE and UE then can forward a message (Register message) to S-CSCF which would not be re-authenticated but in case if UE send an unprotected message then S-CSCF will authenticate it with Authentication and Key Agreement AKA
- P-CSCF transfers the user registration message to Interrogating Call Session Control Function I-CSCF
- P-CSCF identifies the concerned S-CSCF address in user registration process. During SIP based transmission between UE and S-CSCF it acts as post man between user and S-CSCF. P-CSCF also plays a vital role in fast registration process by reducing the packet size with the help of compression algorithms
- P-CSCF includes a policy decision function (PDF) which is responsible for implementing QoS on the media plan for efficient utilization of bandwidth providing QoS enhancement. PDF can be included in same server or it can also be a separate element
- A single IMS network can have more than one P-CSCF for replication and backup purposes.

3.2.2.2. Serving Call Session Control Function S-CSCF

S-CSCF involves in verity of services and work as the most important element of IMS core. Most of its functions are related to user oriented services & applications including registration, session and application services.

- Registration request from end user is received by the S-CSCF and authenticated by contacting HSS for user's security and authentication parameters. The reference point between S-CSCF and HSS is denoted by Cx and Diameter protocol is applied on this interface. During the connection with HSS it also downloads the user profiles to determine the services &

applications they have subscribed to. During the whole registration procedure the concerned user and S-CSCF stay connected.

- S-CSCF inform HSS about the concerned S-CSCF SIP Server allocated to the user for the rest of registration process, and all the end user's signalling traffic passes through S-CSCF.
- S-CSCF accepts requests from end users and after translation the request it decides that whether the request should be process by S-CSCF locally or it should be forwarded to appropriate entity to process.
- After a user gets register with S-CSCF, it controls and configures the registered user's sessions. It also acts as service monitor for operator and can allow user for session creation. Similarly it can deny a specific session for a specific user. S-CSCF takes decisions of allowance or denial of user sessions on the bases of user privileges and subscription for the requested sessions or services.
- S-CSCF also works as user agent and can negotiate with different application servers on behalf of the end User to invoke the services provided by the ASs. It uses IP-Multimedia Subsystem Service Control (ISC) interface to communicate with different application servers.
- During the registration process of registration requesting unit, S-CSCF determine the I-CSCF address and then forward the request to the concerned I-CSCF (internal or external, depending on request).
- S-CSCF has also an interface with Break out gateway control function BGCF. When a request or response is directed to an external circuit switched network then S-CSCF forwards it to BGCF for session routing.
- S-CSCF plays a vital role in UE termination procedure. It forwards the request or response messages to the concerned P-CSCF to terminate the end user from the session. It applies in both cases, if the user equipment is logged to the home network then information is forwarded to home P-CSCF or to visited network P-CSCF in case of roaming.
- One of the most important task that S-CSCF take care of; is modification of SIP request according to the pre-requisites of HSS for session roaming to circuit switched networks. This is an important factor in IMS to Non-IMS (circuit switched communication networks) communication.
- Last but not the least is the contribution of S-CSCF in user billing procedure. It produces Charging Data Records CDR for user billing purpose. CDR includes the service usage details, applications invoked, traffic details etc. This helps the operator to charge the user according to the agreed tariff. Although a single S-CSCF is always a part of the home network but it may be possible to have more than one S-CSCF in a single network to provide redundancy or backup facility. Multiple S-CSCF can also be applied in a single network if the network size is reasonably large and a single S-CSCF server can not tackle the load.

3.2.2.3. Interrogating Call Session Control Function I-CSCF

Like other call session control function servers I-CSCF is also a proxy server which deals with numerous tasks. It is the point of contact during connection establishment of a user. It provides the contact point for user connections and sessions regardless of whether a user belong to the same network or it is a visiting or roaming user from another network currently in coverage area of the specified I-CSCF network. The address of I-CSCF is stated in Domain Name System DNS and made visible to the SIP servers when they follow the protocol for next hope identification.

- During registration procedure of UE, I-CSCF assigns S-CSCF to the user.
- Like S-CSCF, I-CSCF also communicate with HSS and the communication take place with the help of Diameter protocol. It downloads the user profiles from HSS and allocate a S-CSCF proxy server to the user according to the user needs
- I-CSCF also work as router when it receives a SIP signal from another network, it ping HSS for the S-CSCF address and rout the SIP signal to the appropriate S-CSCF
- I-CSCF have the capability to supplement the security mechanism by the hiding the network topology details from external networks. This technique is called Topology Hiding Inter-Network Gateway THIG. I-CSCF encrypt the part of the SIP message which includes information that could be used by hackers to attack the network infrastructure.
- All call session control functions keep track of service usage for users. I-CSCF also play its part in user charging mechanism and forward the generated CDR, to the accounts management element.

3.2.2.4. Breakout Gateway Control Function BGCF

BGCF is actually a SIP server with routing functionality. It can create session based on the user telephone numbers instead of IP addresses. BGCF provide connectivity between the IMS packet switched network and circuit switched network (Public Switching Telephone Network PSTN and Public Land Line Mobile Network PLMN). It transports signalling through Media Gateway Control Function (MGCF), Media Gateway (MGW) and Signaling Gateway (SGW). In fact MGCF, MGW and SGW collectively act as a gateway to circuit switching networks. The main reason of existence of BGCF in IMS core is that when an IMS user initiates a request for a session with a non-IMS circuit switched network user then BGCF is what that has to offer this session establishment.
When SIP signalling is received by BGCF from S-CSCF, it locates the PLMN network and also the breaking point between the packet switched and circuit switched network. If the session belongs to the same network then BGCF forward request to the specific MGCF and in case if the session establishment involve some external network then BGCF forward the request to the concerned network's BGCF. If security is the primary concern for the subjected session then the request can be forwarded through I-CSCF for topology hiding. Like every other core element in session control layer, BGCF also provide the billing information for account management in the form of CDR.

3.2.2.5. Media Gateway Control Function MGCF

MGCF has a link with BGCF on one hand while on the other hand it interfaces with IMS-MGW and it deals with the connection control of media channels in IMS MGW. It finds the next hope for the incoming call on the bases of routing number. The main task of MGCF is inter-conversion of SIP and Integrated Service Digital Network User Part/Transaction Capability Application Part ISUP/TCAP signals.

3.2.2.6. Multimedia Resource Function Controller MRCF

MRCF has two interfaces, one with S-CSCF on Mr Reference point which utilizes SIP protocol for communication and on the other hand it is linked with Media Resource Function Processor MRFP. It controls the media stream resources in MRFP and receives information from application server and severing call session control function S-CSCF. It controls MRFP according to the information received from S-CSCF and MRFP. It also keeps track of billing and produces CDRs.

3.2.3. End Point or Transport Layer

Access or transport layer is the termination point of signalling to the end point entities. In a broad sense it is a gateway to the IMS core from PLMN or PSTN. It transforms the SIP signalling to the end point nodes and control the data traffic and routing.

3.2.3.1. Media Resource Function Processor MRFP

MRFP is controlled by the MRFC and the reference point between the MRFP and MRFC is Mp. The Mp has not been specifically assigned a protocol to operate on, therefore the architecture of Mp is open for further standardization and H.248 protocol is fully supported by MRFP.

3.2.3.2. Signalling Gateway SGW

Signalling Gateway is the gateway between circuit switch networks and packet switched networks, providing signalling conversion from circuit switched networks signalling to IP network signalling and vice versa. It provides the lower level protocols conversion services e.g. it converts Message Transfer Part MTP protocol into Stream Control Transmission Protocol SCTP.

3.2.3.3. Media Gateway MGW

MGW connects the media plan of PSTN/PLMN to IMS media plan [6] and provides interworking between IMS and PLMN/PSTN. Mn is the reference point between Media Gateway Control Function MGCF and IMS-MGW. MGW is completely supported by H.248 protocol and is flexible in support of different media types. It can share physical resources and can be partitioned in virtual separate MGWs. It provides interceding services between IMS and CS domain when there is compatibility barrier between IMS and CS networks.[18]

3.3. IMS Security Mechanisms

In IMS architectures, authentication takes place during the registration procedure. The SIP Digest, IMS Authentication, and Key Agreement (IMS AKA) with the IPSec and TLS with SIP Digest are the main authentication schemes [19] that provide mutual authentication between the UE and CSCF components. Whenever authentication is required, the UE should notify the network side about the cipher suites that it supports. The supported security mechanisms are included in the "Security-Client" header [20] of the very first registration message, which, according to IMS specifications, is unprotected. If this specific header is missing, then the GPRS-IMS-Bundled Authentication (GIBA) [21] or the NASS-IMS-Bundled Authentication (NIBA) [38] will be employed depending on the type of connectivity. It is worth mentioning that the IMS AKA and TLS authentication schemes provide integrity and confidentiality services to the communicating entities through the establishment of security tunnels.

3.3.1. SIP Digest

The SIP digest [22] is a password-based challenge-response authentication protocol without any integrity or confidentiality provisions in communication (Fig. 2). It is utilized in the IMS whenever the UE lacks IMS Subscriber Identity Module (ISIM) or Universal Subscriber Identity Module (USIM). The response of the network side to every non-authenticated registration request is a "401 Unauthorized" message, which indicates that authentication is required. More specifically, whenever a UE sends a registration request to the IMS, the P-CSCF forwards it to the I-CSCF to locate the S-CSCF, which is responsible for handling the specific request. The S-CSCF, in turn, responds with a "401 Unauthorized" message (via I-CSCF), including an Authorization Vector (AV), which is utilized by the UE to compute a valid authentication string for the next registration message. Note that the AV contains all the information (such as nonce, hash algorithm, opaque, etc.) that are required for message authentication, according to RFC 2617 [22]. The nonce is generated by the S-CSCF and is a unique data string that, for instance, could be the result of hashing a timestamp concatenated with the server's private key. The opaque is a data string of the same encoding and is also generated by the S-CSCF. The opaque must be returned to the S-CSCF in the "Authorization" header field of the UE's authentication response.

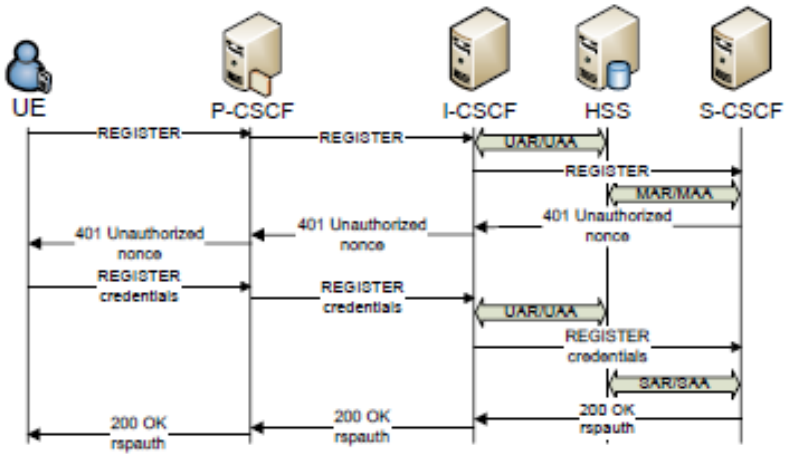


Figure 6. SIP Digest Authentication in IMS

Afterward, the UE extracts the AV from the response message (401 Unauthorized) in order to compute the credentials that will be used in the next UE's request, utilizing the following: H(H(A1), nonce ":" H(A2)), where H is a hash function, A1 = username ":" realm ":" password and A2 = Method ":" digest-uri-value. Whenever an S-CSCF receives an "authenticated" request, the S-CSCF analyzes it: In case that the request is successfully authenticated, the S-CSCF obtains from HSS the user's profile and calculates the rspauth value that enables the user to authenticate on the network. This value is included in the "200 OK" final response generated by the server. However, computed credentials are not included in the other parameters of the request instead of the AV, so the message is not protected against unauthorized modification and the Man-in-the-Middle attacks.

3.3.2. TLS with SIP Digest

The SIP digest can be also deployed in cases where the authentication protocol is a TLS. The main difference in the TLS authentication scheme is that the second REGISTER message (that contains the UE's authentication string) is protected through the integrity and confidentiality mechanisms provided by the specific protocol. After reception of the 401 response, the UE initiates the handshake required for the TLS connection. The authentication of the server side is achieved through the use of valid certificates and then the security tunnels are established. Then the UE sends the response to the second REGISTER message, encrypted, and the authentication procedure continues in exactly the same way as in the SIP digest. All the traffic following the second registration message is encrypted and integrity is protected.

3.3.3. IMS AKA with IPSec

According to the IMS specifications, the IMS AKA [19] is considered as the strongest authentication scheme that can be deployed in cases where a UE embeds an ISIM. Up to a point, this scheme is similar to the SIP digest; however, the IMS AKA establishes a secure tunnel between the UE and P-CSCF (see Fig. 3) that provides integrity and confidentiality services as well as protects communication messages against attacks like the Man-in-the-Middle, eavesdropping, etc. Furthermore, all subsequent messages are protected through the same security tunnel.

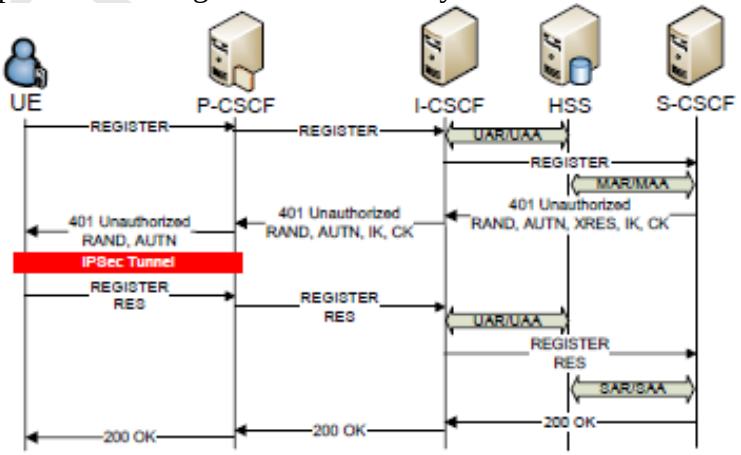


Figure 7. IMS-AKA Authentication

As depicted particularly in Fig. 3, in the IMS AKA, the S-CSCF retrieves from the HSS the AKA AV, instead of the SIP Digest AV, which is utilized to establish a secure tunnel between the UE and P-CSCF. The AKA AV consists of the concatenation of a random number (RAND), the expected response (XRES), the cipher key (CK), the integrity key (IK), and the authentication token (AUTN). When the P-CSCF obtains the AKA AV (without the XRES value) from the I-CSCF, which is included in the "401 response," it strips out the IK and CK and forwards the response to the UE. The UE, in turn, authenticates the server through the validation of the AUTN (note that AUTN includes a Message Authentication Code (MAC)). Afterward, the UE generates the corresponding response message (RES) and computes the IK and CK to establish a secure tunnel (through the IPSec in ESP) between the UE and P-CSCF, which provides integrity and confidentiality services in the subsequent messages. The P-CSCF forwards the new authenticated request to the S-CSCF in order to check its validity. The S-CSCF authenticates the UE by comparing the RES and XRES. In case of a successful authentication (RES=XRES), the S-CSCF continues with the user's profile retrieval and terminates the procedure with a "200 OK" response. In contrast to TLS with SIP Digest, the final 200 OK does not contain the rspauth value because the UE has already authenticated the server during the validation of the AUTN. Further information about IMS AKA authentication and the generation of AVs can be found in [23].

4. Attacks and security vulnerabilities

The SIP protocol is responsible for session establishment and handling in IMS and in the majority of VoIP deployments. The loose syntactic rules and the text based format of the messages comprise a lightweight and flexible protocol. These facts actually facilitate the session establishment and modification, providing high Quality of Service (QoS) with low response times. Nevertheless, these features also render the protocol vulnerable to various attacks and security breaches. There are many scientific works in literature that pinpoint various vulnerabilities of the SIP protocol [24-27]. The employment of a security mechanism such HTTP Digest [9] may deter the attacks originated from external attackers but not from the internal malicious users. The same are applied also in IMS infrastructures. The employment of an authentication mechanism such AKA with IPSec [8] cannot prevent all the threats from IAs. The latter may launch attacks through their legitimately established IPSec tunnels.
Moreover, a successful attack may involve the compromise of different layers of the internet protocol stack, such as the network or the data link layer. For instance, an attacker may launch an ARP poisoning [28] attack in order to gather the Authentication Vectors (AV) from a handshake breaking the authentication mechanism [27] or intercepting the communication [29].

4.1. Forged Message Attacks

As it is already stated, threats in VoIP/IMS environments may involve the manipulation of messages of layers 2, 3 or 5. Concerning the application layer, the attacks can be categorized in four main categories: SIP signalling manipulation, masquerade, Man in the Middle (MitM) and replay attacks.
In signalling attacks the attacker utilizes SIP protocol's requests in order to cause DoS to the server or to specific users. The CANCEL and BYE request are responsible for revoking or terminating multimedia sessions correspondingly. Spoofing the headers "From" and "Call-id" of such requests, an attacker can tear a session down illegally. This method can be launched through the security tunnels by an IA especially in case of a weak parser's implementation. Another DoS attack can be launched utilizing UPDATE or re-INVITE requests. Specifically the malicious user is able to mute a multimedia session or launch a hijacking attack as described in [10]. Authentication and integrity mechanism may deter the EAs but they do not always provide a comprehensive solution against malicious subscribers.
In masquerade attempts an attacker's objective is to impersonate a specific UE or even a user. These attacks are known as SIP spoofing or identity theft correspondingly. The attacker includes a stolen IP Multimedia Public Identity (IMPU) (or private identity – IMPI) to his messages instead of his real one, in order to charge the provided services to victims" identities. Thus, the IA is charged only for the IP connectivity (during the IP allocation from the GGSN) and not for the multimedia services provided by the IMS. The employment of authentication and integrity mechanism in messages or security tunnels cannot guarantee the discouragement of such behaviours. A masquerade attack can also be applied in the third layer where the attacker spoofs the 32bit string of the source IP address header of the packet in order to bypass the SA-SIP check: A correlation between the IP and the given public ID of the messages derived from the SAs which have been established during the AKA in

the registration handshake. This procedure is executed by the S-CSCF (in VoIP architectures such checking procedure is not implemented since is not described in the specifications).
In a MitM attacks, malicious users are placed in the middle of the communication path between user and server. Various incidents based on this technique have been published [30-32]. In this type of attack the attacker bypasses both integrity and authenticity security requirements and consequently is able not only to impersonate users or network elements but also to gain unauthorized access to the provided services, intercept the communication channel or even worst to cause denial of service. These attacks can be launched either by utilizing ARP poisoning [4] (in layer 2) or Domain Name System (DNS) poisoning (in layer 5) [28] techniques. The attacker changes the IP-MAC or the domain-IP associations correspondingly in order to redirect the traffic through him (acting as gateway) and gathers communication channel's data.
In fact, in VoIP/IMS infrastructures, after an ARP or DNS poising attack follows a SIP based attack where the messages are manipulated imposing further damage to the system. For instance, the attacker may spoof the expires header of a registration request to zero causing an immediate deregistration of the victim [33]. Another attack can be launched after a successful MitM by downgrading the security level of the upcoming session. Specifically, during the session establishment handshake, the intermediate manipulates the header (security-client value in authorization) which includes the available security suites removing the stronger ones [28]. Thus the S-CSCF (or the SIP server) will inevitably choose one of the weak security protocols that the attacker has left available in the header during the first offer. Usually, another attack follows the previous one, since the attacker will be able break the employed security mechanism. Also a conference interception could be the result of a MitM attack between the user and the MRFC/AS. An IA spoofs the header Refer-to or Refer-by of the gathered messages, in order to silently invite himself in a conference room [29]. Finally, a MitM can lead to abuse of the authentication mechanism. As described in [27], the attacker acts as intermediate between the proxy (or the P-CSCF in IMS environments) and the user and by masquerading as both of them, is able to steal the AV in order to authenticate his messages. This attack concerns only the SIP Digest authentication mechanism.

4.2. Flooding Attacks

Generally, flooding attacks in SIP based environments such as VoIP and IMS, adopt the methodology of similar attacks occurred in the transport layer and the Transmission Control Protocol (TCP) [36]. More precisely, when a client wants to establish a TCP connection with another host a three-way handshake is required. The client sends a synchronization number (SYN) to the server; the latter allocates memory resources for a specific amount of time and responds with a SYN-ACK containing the received SYN incremented by 1. Finally, the client acknowledges the reception with an ACK including the SYN-ACK incremented by 1 and then connection is established and the server releases the allocated resources. If the client spoofs its source IP address with a non existing one, the server with never get the final ACK and thus it the allocated resources will not be release). It is through that if the client forwards a large amount of such requests the server will soon run out of memory resources causing DoS [37].

This can be applied during a session establishment in SIP. For instance, the attacker may spoof the contact header of an INVITE request. The server (or P-CSCF in IMS) will allocate memory resources for session handling. While the contact header points to another IP the server will not receive the SIP ACK request in order to release the memory resources. A flood of such requests may lead to DoS as described above. A possible solution could be a stateless proxy (P-CSCF) that does not maintain the transaction state.

Another case of flooding attack can be launched when the attack forward an enormous amount of request to a specific network entity in order to cause large delays in active sessions or in the session establishment procedures. The target can be the P-CSCF or the MRFC/AS or even a UE. The latter can be easily flooded due to low amount of CPU and memory resources that they can utilize and the maximum incoming traffic they can handle. The Distributed Denial of Service (DDoS) attacks are executed against the more hardened core network entities. The attacks are originated by multiple sources which forward SIP requests with high rates draining the memory and CPU resources of the target system. The attackers can be innocent UEs/servers or attacking networks. In the case of innocent UEs or servers the attacker may have infect them with malicious software turning them into zombies (called slaves or zombies because they execute orders as dictated by the master namely the attacker). Then, the attacker can deploy all of them at the same time in order to flood a target machine introducing large delay overhead in sessions.

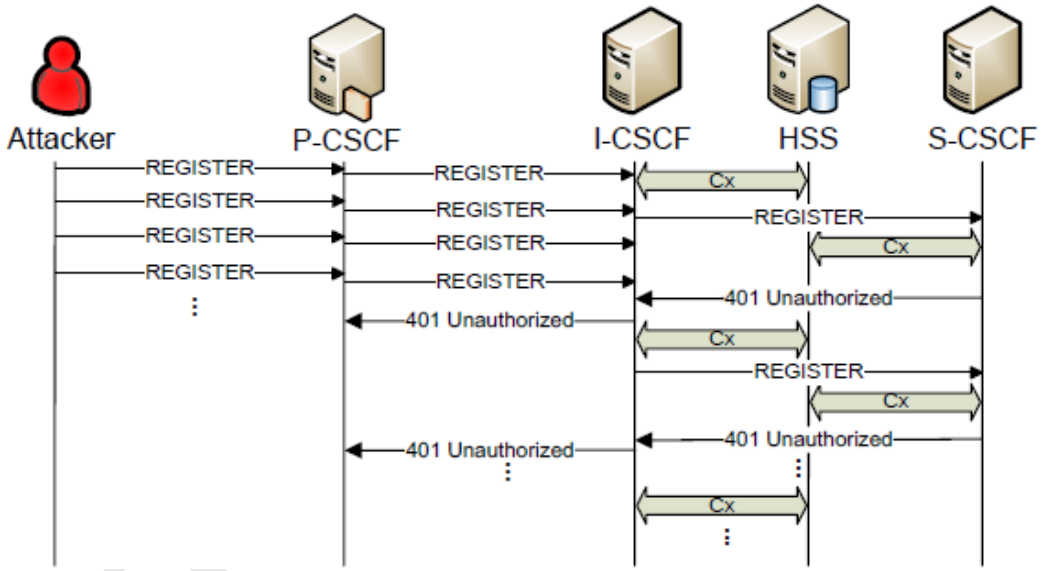


Figure 8. Flooding attack in IMS

Another case of flooding which involves innocent entities is the INVITE reflection syndrome. The attacker sends many INVITE requests to different servers/UEs with spoofed the SIP contact header. Thus, all the involved servers will respond to the given IP of the contact header namely the target machine causing DoS circumstances.

Finally, a CPU resources flooding attack can be launched utilizing registration request. The malicious user can force the IMS core or the registrar server in VoIP infrastructures, to execute cryptographic functions which are considered computationally expensive, in order to validate all these incoming requests (i.e. for every new registration message the IMS core must accomplish at least: the detection

of an appropriate S-CSCF and the computation of new/fresh AVs). The specific attacks, in addition to the quick memory consumption due to numerous half-open connections, are mainly focused on consuming target system's processing resources. Especially in IMS environments, they can introduce delays to network due to the heavy weight employed security mechanisms and the large number of network entities that are involved in every registration request (i.e. opens many diameter connections over Cx, Dx with HSS and Gq interfaces with PDF).

4.2.1. Flooding attack's classification

To consider the flooding attack in SIP environments in conjunction with the proposed mechanism, we have categorized such behaviours in terms of the type of the messages and the access level of the attacker. Therefore, the attacker is divided to Internal (IA) and External (EA). An IA is the entity which has a legitimate subscription to the IMS services and uses its credentials in order to launch flooding attacks. Especially in IMS environments, they can launch attacks through the security tunnels when AKA with IPSec [8] is employed. On the other hand, the EA does not have any legitimate subscription to the server. However, the latter can launch flooding attack since the first registration request is sent without any protection both in VoIP and IMS environment according to specifications: [2] and [1] correspondingly.
Furthermore, we discriminate between registration messages and the rest of SIP requests due to their crucial difference: The registration requests are sent without security protection while all almost the others require authentication such as IMS AKA with IPSec [8], SIP Digest [9] or SIP Digest with TLS [8]. Afterwards every request of these two categories can be spoofed or not. Detecting the type of spoofing, we can deduce whether the attacker uses the same IP address in all its messages or different ones. This slight difference enables the proposed mechanism to determine the type of the attack and its target. These different categories can be further divided relating to whether a message contains a valid authentication string or not and consequently to conclude the target of the attack (e.g. CPU resources exhaustion, end-user flooding, etc.). Finally, there are two main types of DoS attacks the distributed (DS) and single source (SS). This final classification is of major importance in such attacks because the attacker can "silently" flood the servers utilizing lower rate flooding attacks from different machines (zombies) at the same time. Fig. 1 depicts the above mentioned classification tree of flooding attacks. The circles denote the target and the specific type of the attack. The red coloured ones represent the attacks that can only be deterred utilizing the proposed mechanism while the others by its predecessor presented in [35].

4.2.1.1. Internal Attacker

A legitimate subscriber may act maliciously (as an IA) by launching flooding attacks from SS. The utilized messages can be either registration or non-registration requests. If the attacker use the same spoofed IP in all the malicious messages then he probably launches a SYN syndrome attack or a resources exhaustion attack (see section II.B) by forcing the server to open an enormous amount of encrypted connections (over Dx, Cx) with the HSS, and challenge string calculations. This observation based on the fact that the attacker avoids the reception of all responses and also tries to allocate server's memory resources per different IP addresses.

In case where the attacker utilizes spoofed registration messages with same IP address in all messages we can conclude that he launches a brute force attack trying to break user's password or resources exhaustion attack as formerly noted, utilizing a powerful attacking server. This based on the fact that the attacker needs to gather the server's 401 and 200 OK responses.
Another SS attack can be launched by an attacker utilizing other SIP signalling messages such as the INVITE request. Depending on the type of the DoS, the attacker decides to spoof or not the specific INVITE message. When an end-user is the target of the attack, the malicious signalling traffic must be authenticated so the attacker does not spoof the IP addresses. Therefore, an enormous amount of apparent legitimate (the originator is a subscriber and the messages have been authenticated) traffic reaches the user's UE causing DoS. If the messages are not authenticated they can also lead to DoS as a resources exhaustion attack.
When the attacker achieves a replay attack, he forges the messages with (random or fixed) IP addresses and includes the victim's valid authentication string, in order to launch a flooding attack against a specific end-user. More specifically, the attacker can use different IP address for the same authentication string when SIP Digest is employed, because the calculation of responses on this security protocol does not depend on IP addresses. The response is calculated through the following function: H(H(A1), nonce ":" H(A2)), where H is a hash function, A1 = username ":" realm ":" password and A2 = Method ":" digest-uri-value. If the replay attack is not launched or it is not successful, the large volume of unauthenticated traffic can lead to CPU resources exhaustion in the case of fixed IP addresses. Otherwise, the randomly chosen IP addresses add the probability of a SYN syndrome attack through a large amount of half-open connections (server maintains a specific amount of memory per IP for a predefined and fixed period of time).
All the above mentioned cases of attacking possibilities exist and can be launched from multiple sources either as an invite reflection or zombie attacks (see section II.B). Actually, in terms of classification, SS and DS differ only in the effects they can induce to the network/servers due to the massive amount traffic that the latter can utilize. Therefore, when the attacking entities forge registration messages can achieve not only a resources exhaustion attack but also a SYN syndrome attack due to many different IP addresses which reach the servers.
Also, in DS attacks where invite or other non-registration requests are utilized, the core network entities are also threatened due to the large volume of traffic and resources deployed by the attackers. If the requests are deliberately unauthorized can induce resources exhaustion to the core servers as in SS attack, and also a SYN syndrome effect (many IP addresses). Likewise, for not spoofed SIP messages with random IP addresses are applied exactly the same. Moreover, a reflection syndrome can be achieved with not spoofed messages because the innocent servers/UEs are involved without any spoofed headers; note that only the originator (namely the attacker) spoofs the contact header. On the other hand, when the messages are spoofed with fixed IPs are also applied the same except for the unauthenticated ones: the SYN syndrome in this case is not possible because numerous messages reach the servers but with the same IP, hence no more memory resources will be allocated.

4.2.1.2. External attacker

An EA can launch flooding attacks both from SS or DS. Basically, an EA does not have a subscription to the server; so, many of the attacks cannot be launched. As a

matter of fact, when AKA IPSec is employed, the tunnelling is mandatory for every session between the P-CSCF and UE except for the first registration message, as the IMS's specifications [8] define. Therefore, non-registration requests can be forwarded to the proxies only when SIP Digest, GPRS-IMS-Bundled Authentication (GIBA) [38] or the NASS-IMS-Bundled authentication (NIBA) [39] are employed. These cases are represented with dotted arrows in Fig. 1. Moreover, the non-registration requests originated by an EA can only be spoofed and unauthenticated. The authenticated ones fall into the same category as the in the IA cases because this section concerning flooding attacks and their effects. Therefore, either SS or DS flooding attacks can lead to DoS by exhausting CPU's resources or by a SYN syndrome if the IPs have been chosen randomly.
However, registration messages can be forwarded from an EA and processed by the signalling core due to the lack of authentication requirements. Taking into account the aforementioned facts about the registration messages we can deduce that their effects on the architecture are exactly the same in both cases even if the attacker is internal or external.

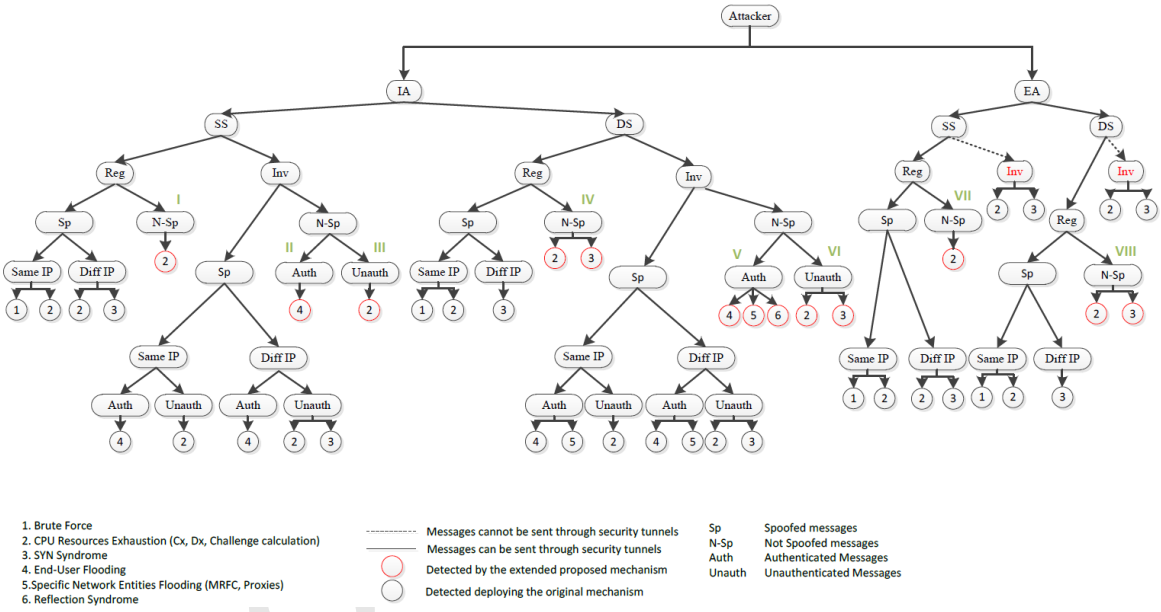


Figure 9. Flooding attack's classification tree.

5. IDPS for VoIP/IMS

Threats in such environments come from different layers, since the attacker in order to compromise architecture's security exploits more than one protocol's vulnerabilities. Thus it is important for the IDPS to be able to detect such behaviours before they pose a threat to the higher layers and affect multimedia services. The proposed IDPS is based on the concept of the cross-correlation table that we have introduced in [29, 35]. A table has been employed in order to keep specific information for every registration request from the layers 2, 3 and 5 of the protocol stack. Every row has been responsible for holding the specific requests‟ values after their successful registration. A decision tree has been also introduced towards the discrimination between fake and legitimate messages. The mechanism consists of two different modules: the first module is responsible for registrations and the other one for all the remaining requests.

This model from its structure is not only able to deter spoofing attacks such as SIP signalling, UE and User ID impersonation, MitM but also the most threatening ones for the architecture's availability, the flooding attacks. However, some cases of flooding attacks cannot be detected, so we extend the mechanism's functionality in order to be able to handle them. It is true that most of the flooding attacks come with spoofed messages, therefore they can be mitigated. However, they can also be launched without utilizing spoofed messages, for instance though a security tunnel. Considering Fig. 1, we can pinpoint seven cases that the conventional firewall/IPS cannot detect/deter:

(I). If a UE and a user id have never been registered before, namely it is the first registration message, this message will be dropped. This happens because there is not any record for this combination. Thus, the UE cannot be registered. On the other hand, if the IDP gives unprotected access only for registration messages then the architecture will be vulnerable to folding attacks.

(II). In case of an IA who authenticates all of his messages without forging them, neither a conventional firewall nor a security mechanism such as IPSec can detect them during a flooding attack.

(III). These requests are not spoofed but also they are unauthenticated. Taking into account that the IE is already registered he can utilize his legitimately established security tunnel to send them. According to specifications [8], after the tunnel establishment, authentication vectors are not included on SIP messages. Hence, flooding attacks in such a case cannot be deterred.

(IV). In this case are applied the same as in I

(V). As in II

(VI). As in II

(VII). and VIII. are applied the same as in I

The above mentioned special cases and the others depicted in Fig. 1, can be detected and mitigated through the deployment of the second module.

5.1. Mechanism Description

As it is already stated, the proposed mechanism consists of two modules. Concerning the first module it handles every incoming message and take decisions about the originality of the message in order to route it to its destination or not.

Specifically, the idea is based on cross layer binding between six values that can be gathered from layers 2, 3 and 5 of the network protocol stack. These values correlate a

specific UE with a session, a set of IP addresses and the identities of the subscribers. For instance, the frames located at the data link layer (layer 2) bears the network or MAC address of the utilized UE. Furthermore, the binding among the IP address of the 3rd and 5th layer and the MAC address must be unique at a specific point of time.
For every incoming message a tuple Ei, $\forall$ I $\in$ {0, …, n} is generated, where n is the number of incoming messages and n $\in$ N.
Every Ei passes through the spoof checking module which decides if the message is legitimate and thus will be forwarded to the second module or it will be dropped setting the appropriate rule to the Policy Enforcer (PE). The latter holds a blacklist of known malicious Ei. An incoming message is firstly checked for existence in the PE‟s list. Only the non-listed messages should be forwarded and handled by the next modules.
Afterwards the legitimate Ei are handled by the second module. The second module consists of two tables: the registration table for holding registration messages‟ data and the request table that holds the data of all the other requests. Every Ei must be inserted to one of these tables depending on the type of request. The proposed mechanism's architecture is presented in Fig. 2. The position on the table where a tuple must be stored is calculated according to the theory of the bloom filters [40].
A bloom filter is a data structure which facilitates the procedure of testing the existence of an element xi in a set X. Every xi is hashed through λ different hash functions. The result of every hash function points to a specific position on vector of m bits. The vector's length is equal in bit as the output of the hash functions. Initially, all the vector's bits are zero. When a hash output point on such a position, it turns the null bit to 1. A xi exists in the set/vector when all λ outputs points on a 1 bit position.
A more advanced bloom structure is the counting filter. The conventional filter does not allow any other calculations (e.g. subtraction or summation) than turning only the zero bits to 1. On the other hand, counting bloom filters are capable of counting how many times a hash output points on specific table position. We use this model in order avoid searching and sorting through the second module's tables: The first column in both tables is a counting bloom filter. The input to hash functions the tuple
Fi = {MACi, IMPIi, IPi}, FI $\subseteq$ Ei. These three values denote a unique combination that is extracted from every Ei.

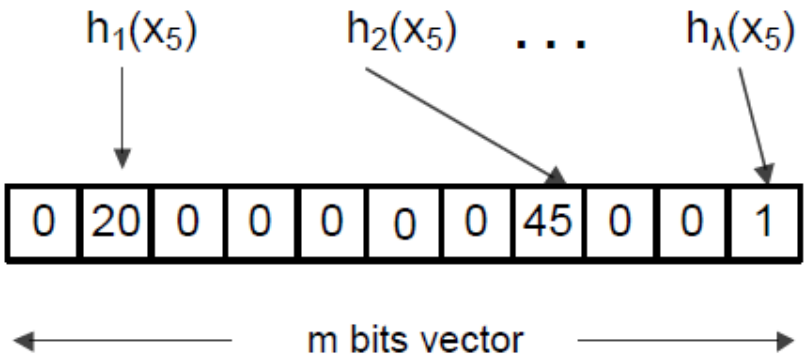


Figure 10. Counting bloom filter

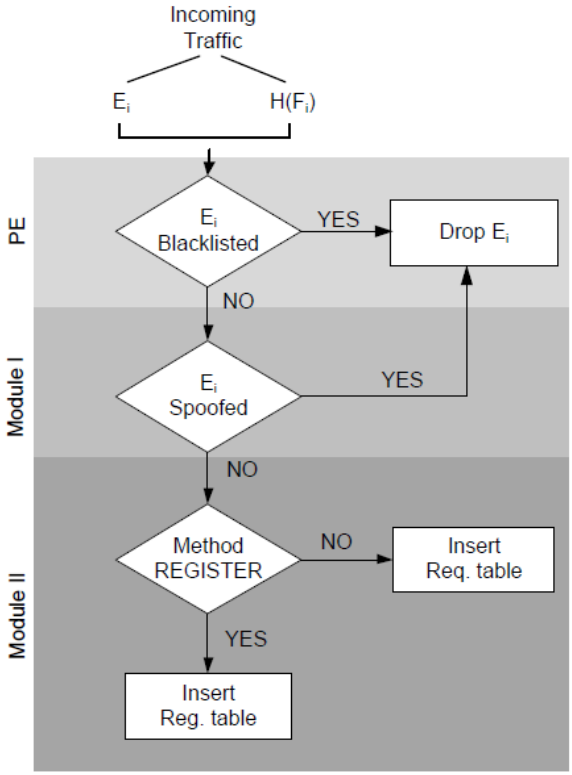
Therefore, the exact position for every user per UE/IP on the tables is the value H(Fi). Both of the tables have eleven different columns as depicted in Table I: C is the counting bloom filter that provides the mechanism with information about the number of messages per subscriber/UE and eliminates the time needed for detecting a specific

tuple's position. The values MAC, IP, SIP-IP, denote the corresponding address of layer 3, 4 and 5 of the network protocol stack that have been involved in a specific request. The IMPI/IMPU holds the private/public id of the corresponding incoming message. The TS holds a timestamp and T.Dist the time distance between the last two timestamps and calculated: T. Dist = TS – TS_{i-1} , where i is the number of messages that stored in a specific row. The values Init.D.Avg and Curr.D_Avg denote the initial T.Dist and the current correspondingly. Finally, the Trs value is a threshold for alarm triggering in single source flooding attacks. The monitoring method is depicted in Fig. 4. The Fig. 5 presents a pseudo-code of the monitoring procedure and how the incoming messages are handled by the proposed mechanism's components.

5.1.1. Spoofing detection method

In order to provide a more accurate and descriptive presentation of the proposed mechanism's spoofing detection procedure, we define the following sets for every network layer involved; M = {the set of MAC addresses}, I = {the set of IPs}, S = {the set of SIP-IPs}, D = {the set of IMPIs and IMPUs}, A = {the set of SIP methods}. Moreover, if R = {REGISTER} then

$R \subseteq A$. Therefore, Ei = {m_i, i_i, s_i, a_i, d_i}, where $m_i \in M$, $i_i \in I$, $s_i \in S$, $a_i \in A$, $d_i \in D$. Also, K_i = {m_i, i_i, s_i, d_i} and $O = \bigcup_{i=1}^{m} K_i$ and finally $W = \bigcup_{i=1}^{m} E_i$.




```
INPUT: The extracted Ei tuple from incoming traffic
OUTPUT: An Ei that shall be blacklisted or a
legitimate Ei that shall be inserted to the second
module's table

1   modulei(Ei);
2
3           if module1 == 1 /* module1 returns 1 when
4           message shall be routed and 0 otherwise*/
5           then
6                   Hash = H(Ei.MAC||Ei.IP||Ei.ID);
7                   if Ei.method = REGISTER
8                           insert_reg_table(Hash, Ei);
9                   else
10                          insert_request_table(Hash, Ei);
11          else
12                  drop_packet(Ei);
13                  upadate_PE(Ei);
```

Figure 11.
Monitoring Method (left) – Mechanism's message Handling Pseudo-Code (right)

For every incoming message the values Ei and H(Fi) are generated. The spoofing module (module I) detects the position H(Fi) on the registrations' table and retrieves the corresponding Ci's data. Let Ec be the corresponding tuple. The mechanism executes the following procedure between the tuples Ei and Ec. This procedure is depicted in Fig. 6. We have introduced this methodology in [35] and include it slightly modified into herein for the sake of completeness.

If $Ei \cap Ec = \varnothing$ and $ai \in R$, then Ei corresponds to a new registration procedure, therefore there is no identical set in W. If the specific message has been authenticated and the ii= si (intra-packet check) then the corresponding tuple will be forwarded to module II in order to be stored in the registration table denoting the first registration and the binding of the specific UE subscriber for this specific period of time.

If $Ei \cap Ec = \varnothing$ and $ai \in R$, then there is no identical set in W, thus the message has been spoofed and shall be dropped, the PE has to be updated as well with the Ei tuple. The UE is actually not yet registered and this is derived from the fact that the two (Ei, Ec) sets do not have common elements.

If $Ei \cap Ec \neq \varnothing$, then at least one of the $m_i, i_i, s_i, a_i, d_i \in Ec$.

i. Let only $mi \in Ec$, then $i_i, s_i, a_i, d_i \notin Ec$. Therefore the corresponding message shall not be processed because this corresponds to an identity theft attempt or the IP addresses have been spoofed. The specific Ei must be forwarded to the PE.
ii. Let di, $mi \in Ec$, given that $i_i, s_i \notin Ec$, then if $a_i \in R$ and $i_i = s_i$, the corresponding tuple (H(Fi)) shall be updated only when the message has been successfully authenticated. This registration message comes from a UE that has changed location and has been allocated a new IP address.
iii. Let ai, $mi \in Ec$ given that $i_i, s_i, d_i \notin Ec$. If $ai \in R$ then corresponding registration message has been initiated from the same UE but the subscriber has changed. After the successful registration the corresponding sc has to be updated with the incoming s_i. For instance, a user swaps the Universal Integrated Circuit Board (UICC) with another one utilizing the same UE and proceeds to a new registration procedure. The case where $ai \notin R$ is covered in (i).
iv. Let only $s_i, m_i \in Ec$, then there is application or network layer spoofing attempt and the corresponding message shall not be processed. The PE must be updated with the specific Ei.
v. Let only the $i_i, m_i \in Ec$, then it is straightforward that $s_i \in Ec$ and thus there is an application layer replay or SIP signalling attack and the message shall be dropped. The PE must be updated. The attacker has reused a previously gathered SIP message from another subscriber. The attacker's objective is to bypass authentication mechanisms.
vi. Let the $i_i, m_i, s_i \in Ec$, the message includes an IMPI/IMPU from another subscriber. This identity theft attempt comes from an IA and the message shall be dropped. The Ei forwarded to the PE. We can assume that the attacker is an insider because a registration for his UE already exists in the table (the only element that does not belongs to Ec is the d_i). This behaviour may enable the attacker to charge the provided service to the actual IMPI/IMPU owner.
vii. Let the $m_i, i_i, s_i, d_i \in Ec$, then also $m_i, i_i, s_i, d_i \in Kc$. Then the sets Ki and Kc are identical (Ki = Kc) and the message is legitimate one and shall be processed. When Ki = Kc, the message is legitimate irrespectively if $ai \in Ec$.
viii. Let only the $i_i \in Ec$ then $m_i, s_i, a_i, d_i \notin Ec$. The message that corresponds to this specific tuple is spoofed and shall be dropped. The PE function must be updated.

ix. Let the i_i, s_i, $d_i \in Ec$. If $a_i \in R$ then the corresponding shall be processed because it can be considered as a legitimate one. The subscriber has initiated a registration procedure utilizing a new UE and the tuple has to be updated (with the new MAC address) after a successful authentication. If $ai \notin R$, then the IPs of both protocols (SIP and IP) are spoofed or there is an ARP poisoning attempt. The correspondence between MAC and IP has changed and the sets $Ei \neq Ec$ and $Ki \neq Kc$.

x. We know that $Kc \subset Ec$ because $a_c \notin Kc$ then of course neither $a_i \notin Kc$. Therefore if $Ki \cap Kc = \varnothing$, given that $Ei \cap Ec \neq \varnothing$, it is derived that only $a_i \in Ec$. Then if the specific $a_i \in R$ and $i_i = s_i$ the message comes from a UE that has initiated a registration procedure for the first time. The tuple that corresponds to the specific Ei has to be updated (ii→ic and mi→mc) and stored to the registration table after a successful authentication. Otherwise, if $ai \notin R$ or $ii \neq si$ the message is spoofed while an unregistered UE tries illegally to forward a message or to be registered with forged IP.

xi. Let only the $si \in Ec$ then m_i, i_i, a_i, $d_i \notin Ec$. The message is spoofed at the application layer. This can be a replay attempt or a SIP signalling attack initiated from an external user. The deduction that the attacker comes from the outside is derived from the fact that none of m_i, i_i belong to E_k, thus the specific UE has not been registered until then. This PE is informed for that action.

If the crosschecking between Ei and Ec concludes that the message is not spoofed, then the Ei is forwarded to module II. The specific message may be not spoofed but we do not know if it is also a legitimate one (it may takes part in a flooding attack).

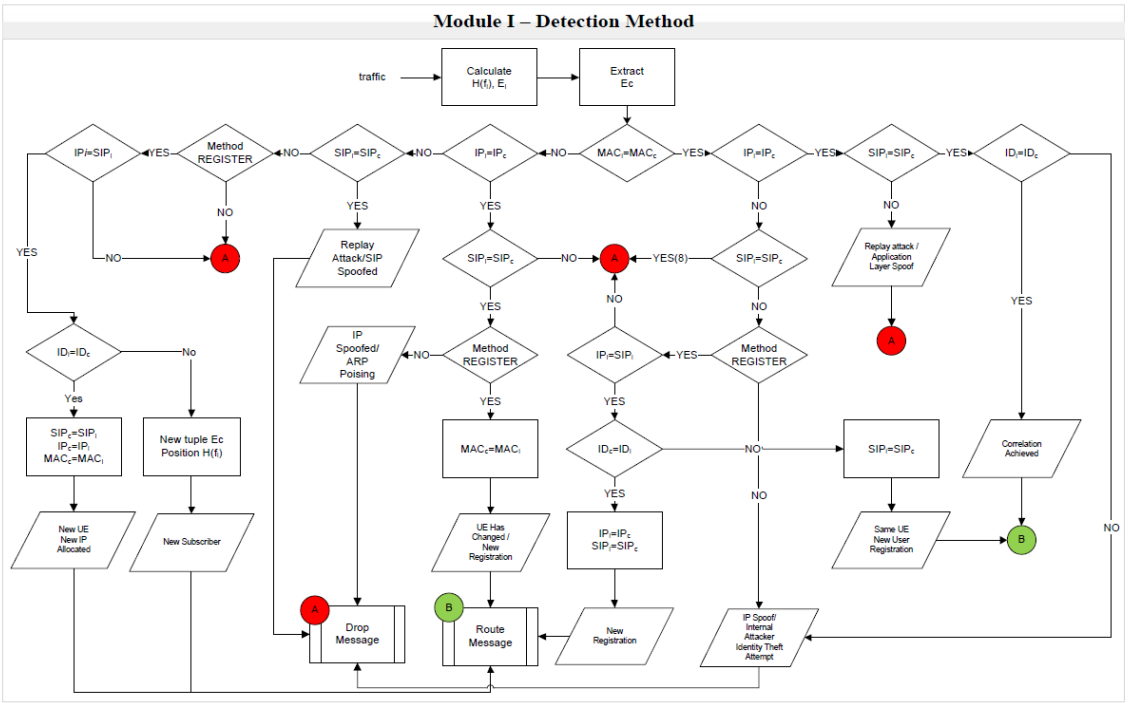


Figure 12. Module 1 – Detection Module

5.1.2. Flooding attack detection method

The second module is fed with the Ei that has successfully passed the cross-correlation checking of module I. As it is already said, every Ei has always its own same position on the table and thus the values are overwritten except for I.D_Avg.

The latter value is calculated during the initial handshakes of a specific UE with the server. If the T.Dist between two consequent messages is smaller than average (T.Dist_Avg), the Trs value is incremented by 1. The T.Dist_Avg is calculated during a raining period of 30 minutes in normal traffic environment:

$$T.Dist\_Avg = \sum_{i=1}^{m} T.Dist_i \Big/ z \quad (1)$$

where z is the number of Ci ≠ 0. An alarm is triggered (SSalarm) when the Trs exceeds a predefined number of messages that are below the T.Dist_Avg. The specific Ei tuple is send to the PE in order to deny access to the corresponding UE.
In the case of distributed flooding attacks, the attacking entities may flood the server with low rate of consequent messages so as not to exceed the T.Dist_Avg value. Such a behavior can be detected utilizing the Init.D_Avg and Curr.D_Avg values in conjunction with the average of the incoming messages (Init.C_Avg). The Init.C_Avg can also be measured during the training period by the function:

$$Init.C\_Avg = \sum_{i=1}^{m} C_i \Big/ z$$

If this value grows with an unusual rate an alarm is triggered (DSalarm1). A tolerance rate (tr) should be estimated according he server's capabilities during high traffic periods. Thus, if the Curr.C_Avg > Init.C_Avg + tr then a DS flooding attack is under way. As it is already said, the attacking entities cannot be detected by calculating only the T.Dist because it may not be exceeded and the alarm not triggered. This can be prevented by detecting variations between the Init.D_Avg and Curr.D_Avg of every row. An alarm is triggered when the value.

$$\lim_{Curr.D\_Avg \to 0} Curr.D\_Avg \,/\, Init.D\_Avg = x \quad (2)$$

Therefore, it is calculated the increment of the average response time of the specific tuple. If that happens, for instance with a rate of 60%, namely x=0.4, during the first DS flooding alarm (DSalarm1) it can be derived that the Ei has been involved to the attack (DSalarm2) and thus must be sent to PE to restrict its access to the server. Another case of attack that can be detect by (2) is the increasing rate flooding attack [41].

	C	MAC	IP	SIP-IP	IMPI/IMPU	Method	TS	T. Dist	Init. D_Avg	Curr. D_Avg	Trs.
H(F_{74}) →	54	MAC_{11}	IP_{11}	$SIP\text{-}IP_{11}$	clam	INVITE	100	15	14	10	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
H(F_{42}) →	20	MAC_4	IP_4	$SIP\text{-}IP_4$	nvra	NOTIFY	10	5	7	5	5
	60	MAC_7	IP_7	$SIP\text{-}IP_7$	unipi	INFO	1500	12	8	4	7

Table I. Request Table

In such situations the attackers try to bypass a traffic rate-based detection mechanism by gradually increasing the attack rate. Therefore the distance average is decreased

gradually until very low values. This can be detected employing the function (2) which calculates slight or enormous deviation in UEs' traffic behaviour. The detection is illustrated in Fig. 13.Table I SSalarm can detect only CR attacks by calculating the function (1), where the attackers send an enormous amount of messages per time unit. On the other hand in IR attacks, the gradual increment of flooding rate slowly decreases the T.Dist_Avg value and thus function (1) tends to be incapable of coping with them.

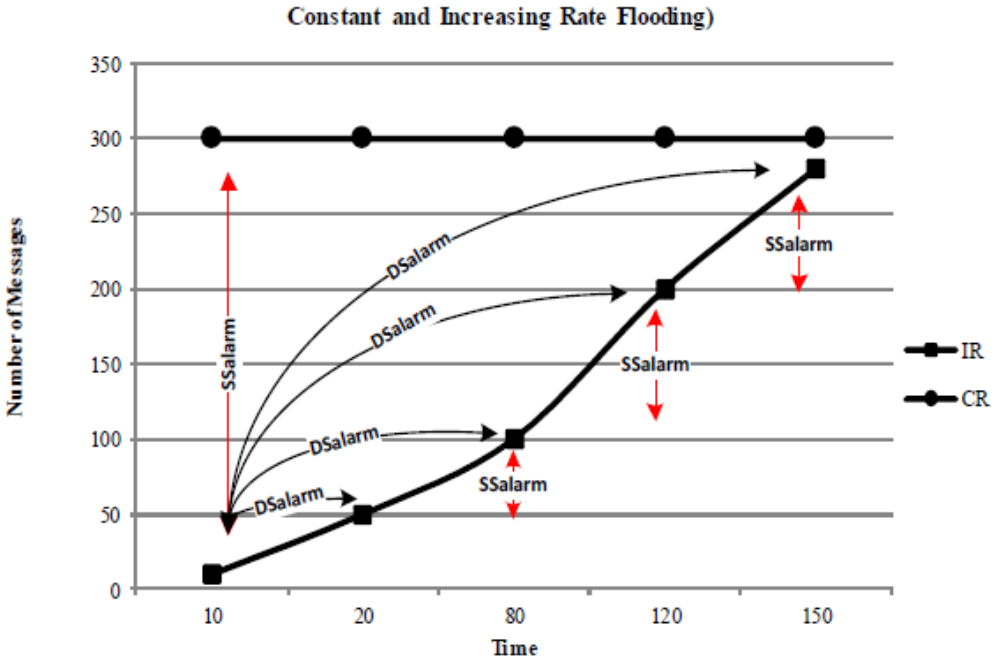


Figure 13. Detection of Increasing and Constant Rate attacks

6. Related Work

In literature, can be found different approaches and proposals of how VoIP and IMS environments can be hardened against such attempts. However, many of them are focused only on the detection without being able to actively deter the attackers while others can only discourage only some cases of these above mentioned attacks. In [14]1 is introduced a flooding detection model based on priority queues. More specifically, it deploys two queues, one of high and one of low priority. All the INVITE messages are inserted in the low priority queue while the responses are inserted in the high priority one. The messages in low-priority queue will be processed only when the high priority queue is empty. The result is that the legitimate requests are the one that handled first while their responses are in higher priority. On the other hand the INVITE messages are processed with an increased delay but the server can still be on-line avoiding DoS consequences.
Moreover, the illegitimate INVITEs are discarded faster since they do not come with responses so the high priority queue remains empty. Nevertheless this model does not actively deter or blocks the attacker but only mitigate the effects of the flooding attack. Furthermore, the attacker can bypass the mechanism by flooding the server and consequently the two queues, both with INVITE requests and responses (e.g. 100, 200, 180 e.t.c.). In [13]2 the authors utilize bloom filters and a SIP specific metric called "session distance", in order to correlate the number of the received requests with the number of 200 OKs and ACKs and thus detect deviations from the normal traffic. Similarly, in [12] a DoS mechanism that extends the detection to the transport layer is proposed. The mechanism detects traffic abnormalities by calculating the Hellinger distance between requests and responses taking into account not only the SIP traffic but also the transport protocol's traffic. Both of the mechanisms do not provide a methodology to detect and deter the attackers but only alarms are triggered when a flooding attack is under way. Moreover, they are focused only on INVITE request flooding attack case.
In [42] is described a mechanism that monitors REGISTER messages. A successful registration takes place when a 200OK is received by the server which includes the REGISTER value in the CSEQ header. Then a tuple is created in a table containing information gathered from the SIP message: the user's identity (UID), the UE's IP address a timestamp and the duration of the specific registration in seconds as it is derived from the expires header. The UID acts as primary key in the table. This table is actually a white list and therefore only the stored UIDs will be processed. This mechanism does not offer any protection against IAs which have a legitimate subscription and can authenticate all of their requests. Moreover it is unable to deter INVITE flooding attacks and also it cannot be deployed in IMS infrastructures in conjunction with security tunnels because the messages do not contain SIP authentication except for the first REGISTER request. Finally, messages with forged UIDs can bypass the mechanism and also the authors do not provide any description for handling the DDoS attacks.

Table II: Security Mechanism's comparison

Layer	Thread Category	Attack	Proposed		[14]1		[10]4		[35]5		[12],[13]2		[34]3		[36]6	
			D	P	D	P	D	P	D	P	D	P	D	P	D	P
Layer 5	SIP Signaling	BYE	✓	✓	X	X	✓	X	✓	✓	X	X	✓	✓	✓	✓
		CANCEL	✓	✓	X	X	✓	X	✓	✓	X	X	✓	✓	✓	✓
		Re-INVITE	✓	✓	X	X	✓	X	X	X	X	X	✓	✓	✓	✓
		UPDATE	✓	✓	X	X	X	X	X	X	X	X	✓	✓	✓	✓
	Masquerade / ID Theft (L5)	UE Impersonation (SIP IPs)	✓	✓	X	X	X	X	X	X	X	X	✓	✓	X	X
		User Impersonation (SIP IDs)	✓	✓	X	X	X	X	X	X	X	X	✓	✓	✓	✓
	MitM	Registration Expiration	✓	✓	X	X	X	X	X	X	X	X	✓	✓	✓	✓
		Bid Down	✓	✓	X	X	X	X	X	X	X	X	✓	✓	✓	✓
		Generic Authentication	✓	✓	X	X	X	X	X	X	X	X	✓	✓	✓	✓
		Conference Interception	✓	✓	X	X	X	X	X	X	X	X	✓	✓	✓	✓
	Replay	SIP Replay	✓	✓	X	X	X	X	X	X	X	X	✓	✓	✓	✓
	Flooding Attacks	Non-INVITE	✓	✓	X	X	X	X	X	X	X	X	X	X	X	X
		Invite	✓	✓	✓	✓	X	X	✓	✓	✓	X	X	X	X	X
		Single Source	✓	✓	✓	✓	X	X	✓	✓	✓	X	X	X	X	X
		Distributed Source	✓	✓	✓	✓	X	X	✓	✓	✓	X	X	X	X	X
Layer 3	Masquerade (L3)	UE IP spoof	✓	✓	X	X	X	X	X	X	X	X	X	X	X	X
Layer 2	MitM (L2)	ARP Poison	✓	✓	X	X	X	X	X	X	X	X	X	X	X	X

With D is denoted that the mechanism provides only detection while with P that it can also prevent from the corresponding attack.

In [10] is presented a mechanism which utilizes rules in order to examine events and acts that developed in VoIP architectures. An alarm is triggered when a collection of events corresponds to predefined rules. This alarm indicates that an attack occurs. The majority of the rules are based on an end to end matching rule which actually detects deviations in signalling flows. This mechanism is able to detect MitM and billing fraud attacks. However, an attacker may bypass the rules by launching a layer 3 impersonation attack. Finally, this approach does not offer protection against many MitM, signalling and flooding attacks.

A flooding attack prevention mechanism is presented in another work [43]. The authors propose a detection mechanism that is able to detect DoS including some SIP signaling attacks. Specifically, Honey Pot architecture is deployed in order to provoke attacker's interest and thus gathering useful data towards the detection of such attacks. Utilizing anomaly and signature based detection techniques; the mechanism creates profiles of "normal behaviour" for users and network entities and signatures of known attacks. Any deviation from the normal behaviour standards can be considered as an attack. An attack is detected by correlating different events through specific rules. For instance, the BYE signaling attack can be detected by spotting orphan RTP flows after a period of time (only from one participant while the other has received the BYE message) utilizing signature-based correlation with attack patterns.

In [44] the authors propose a mechanism for integrity protection in SIP messages. Specifically, all the contents of the SIP messages are hashed including the header and the body of the message with the user's password. Also, this model proposes the embodiment of an extra header (Verify-Body) which contains all the necessary information (e.g. username, hash function, realm etc.) for the server in order to verify the integrity of the message. Although this mechanism can deter signaling attacks, replay and MitM attempts in SIP layer, it cannot prevent flooding attacks and man in the middle in all the remaining layers.

In Table II is depicted a comparison among the above mentioned models with respect to their efficiency in detecting and preventing the attacks.

7. Conclusion

In this worksheet we have introduced an intrusion prevention mechanism that can be deployed both in IMS and VoIP infrastructures that utilize the SIP as signalling protocol. The detection covers the most of the spoofed SIP message attacks and also flooding attacks when originated either from single or distributed sources. Its design poses a lightweight mechanism, free of complex and resource demanding calculations, a very crucial factor in such environments where the high QoS is the top priority. The on-line position in the network, offers real time detection and prevention of the attacks and its cross layer structure facilitates the detection of an incident from the lower layers of the internet protocol stack. It is also covers the largest amount of the security breaches that can be developed in SIP environments in relation to the ones proposed in literature [10, 12-14, 36-38].
Our objective, in the upcoming works is to extend the mechanism in order to cover the rest of the attacks that can be launched against such infrastructures that may threaten their availability. Specifically, malformed message attacks are posing a threat even in the more contemporary implementation and can cause DoS introducing large amount of overhead in the communication [39].

References

[1] 3GPP, "TS 23.228: IP Multimedia Subsystems (IMS)," ed: Third Generation Partnership Project, Technical Specification Group Services and System Aspects, 2008.
[2] J. Rosenberg, et al., "RFC 3261: SIP: Session Initiation Protocol," 2002.
[3] M. Tanase, "IP spoofing: an introduction," Security Focus, vol. 11, 2003.
[4] R. Wagner, "Address resolution protocol spoofing and man-in-the-middle attacks," The SANS Institute, 2001.
[5] D. Geneiatakis, et al., "Survey of security vulnerabilities in Session Initiation Protocol," IEEE Communications Surveys and Tutorials, vol. 8, pp. 68-81, 2006.
[6] Y. Park and T. Park, "A Survey of Security Threats on 4G Networks," in Globecom Workshops, 2007 IEEE, 2007, pp. 1-6.
[7] A. Keromytis, "A Survey of Voice over IP Security Research," in Information Systems Security. vol. 5905, A. Prakash and I. Sen Gupta, Eds., ed: Springer Berlin / Heidelberg, 2009, pp. 1-17.
[8] 3GPP, "TS 33.203: 3G security; Access security for IP-based services (Release 10)," ed: Third Generation Partnership Project, Technical Specification Group Services and System Aspects, 2010.
[9] J. Franks, et al., "RFC 2617: HTTP authentication: basic and digest access authentication," 1999.
[10] Y. Wu, et al., "Intrusion detection in voice over IP environments," International Journal of Information Security, vol. 8, pp. 153-172, 2009.
[11] Y. Wu, et al., "Scidive: A stateful and cross protocol intrusion detection architecture for voice-over-ip environments," in Proceedgins of the 2004 International Conference on Dependable Systems and Networks (DSN 2004), Firenze, Italy, 2004, pp. 433-442.
[12] H. Sengar, et al., "Detecting VoIP floods using the Hellinger distance," IEEE Transactions on Parallel and Distributed Systems, pp. 794-805, 2008.
[13] D. Geneiatakis, et al., "Utilizing Bloom Filters for Detecting Flooding Attacks against SIP Based Services," Computers & Security, vol. 28, pp. 578-591, 2009.

[14] X. Y. Wan, et al., "A SIP DoS flooding attack defense mechanism based on priority class queue," in IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS), Beijing, China, 25-27 June, 2010, pp. 428-431.
[15] A.Johnson, "SIP, Understanding the Session Initiation Protocol: Second Edition" pp. 108-126, 2004.
[16] P. G. Argyroudis, et al., "Performance analysis of cryptographic protocols on handheld devices," in Network Computing and Applications, 2004. (NCA 2004). Proceedings. Third IEEE International Symposium on, 2004, pp. 169-174.
[17] C. Shen, et al., "The impact of TLS on SIP server performance," in IPTComm 2010: 4th Conference on Principles, Systems and Applications of IP Telecommunications Principles, Systems and Applications of IP Telecommunications, Munich, 2010, pp. 59-70.
[18] Aftab Ur Rehman, "Investigation of Interworked IMS Architecture In Terms Of Traffic Security", 2009.
[19] 3GPP, "TS 33.203: 3G security; Access security for IP-based services (Release 10)," ed: Third Generation Partnership Project, Technical Specification Group Services and System Aspects, 2010.
[20] J. Arkko, et al., "RFC 3329: Security mechanism agreement for the session initiation protocol (SIP)," 2003.
[21] 3GPP, "TR 33.978 Security aspects of early IP Multimedia Subsystem (IMS)," ed: Third Generation Partnership Project, Technical Specification Group Services and System Aspects, 2008.
[22] J. Franks, et al., "RFC 2617: HTTP authentication: basic and digest access authentication," 1999.
[23] 3GPP, "TS 33.102 Universal Mobile Telecommunication system (UMTS); 3G Security; Security Architecture," ed: Third Generation Partnership Project, Technical Specification Group Services and System Aspects, 2010.
[24] D. Geneiatakis, et al., "SIP Security Mechanisms: A state-of-the-art review," in Proceedings of Fifth International Network Conference, Samos, Greece, 2005, pp. 147-155.
[25] D. Geneiatakis, et al., "SIP message tampering: The SQL code injection attack," in Proceedings of 13th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2005), Split, Croatia, 2005.
[26] A. Bremler-Barr, et al., "Unregister attacks in SIP," in 2nd Workshop on Secure Network Protocols, NPSec, 2006, pp. 32-37.
[27] H. Abdelnur, et al., "Abusing SIP authentication," Journal of Information Assurance and Security Volume, vol. 4, 2009.
[28] A. Klein. (2007, BIND 9 DNS cache poisoning. Available: http://www.trusteer.com/docs/bind9dns.html
[29] N. Vrakas, et al., "A Call Conference Room Interception Attack and its Detection," presented at the 7th International Conference on Trust, Privacy & Security in Digital Business, Bilbao, Spain, 2010.
[30] N. Asokan, et al., "Man-in-the-middle in tunnelled authentication protocols," Lecture Notes in Computer Science, vol. 3364, p. 28, 2005.
[31] H. Xia and J. Brustoloni, "Hardening web browsers against man-in-the-middle and eavesdropping attacks," 2005, p. 498.
[32] R. Zhang, et al., "On the feasibility of launching the man-in-the-middle attacks on VoIP from remote attackers," presented at the 4th ACM Symposium on

Information, Computer and Communications Security, Sydney, Australia, March 2009.
[33] C. Callegari, et al., "A novel method for detecting attacks towards the SIP protocol," in International Symposium on Performance Evaluation of Computer & Telecommunication Systems, 2009. SPECTS 2009., pp. 268-273.
[34] D. Sisalem, et al., SIP Security: Wiley, 2009.
[35] N. Vrakas and C. Lambrinoudakis, "A Cross Layer Spoofing Detection Mechanism for Multimedia Communication Services," International Journal of Information Technologies and Systems Approach, 2011.
[36] J. Postel, "RFC 793: TCP: Transmission Control Protocol," 1980.
[37] S. Bellovin, "Security problems in the TCP/IP protocol suite," ACM SIGCOMM Computer Communication Review, vol. 19, p. 48, 1989.
[38] 3GPP, "TR 33.978 Security aspects of early IP Multimedia Subsystem (IMS)," ed: Third Generation Partnership Project, Technical Specification Group Services and System Aspects, 2008.
[39] ETSI, "TS 187 003: Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN): Security Architecture," ed, 2008.
[40] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," Commun. ACM, vol. 13, pp. 422-426, 1970.
[41] J. Udhayan and T. Hamsapriya, "Statistical Segregation Method to Minimize the False Detections During DDoS Attacks," International Journal of Network Security, vol. 13, pp. 152–160, 2011.
[42] E. Y. Chen and M. Itoh, "A whitelist approach to protect SIP servers from flooding attacks," in IEEE International Workshop Technical Committee on Vancouver, BC, 8-10 June, 2010, pp. 1-6.
[43] M. Nassar and S. Niccolini, "Holistic VoIP intrusion detection and prevention system," in Principles, Systems andApplications of IP Telecommunications (IPTComm 2007). New York, USA, 2007, pp. 1-9.
[44] H. Takahara and M. Nakamura, "Enhancement of SIP Signaling for Integrity Verification," in Applications and the Internet (SAINT), 2010 10th IEEE/IPSJ International Symposium on, 2010, pp. 289-292.

APPEX

List of Response codes

1xx—Provisional Responses

100 Trying
Extended search being performed may take a significant time so a forking proxy must send a 100 Trying response

180 Ringing
Destination user agent received INVITE, and is alerting user of call.

181 Call is Being Forwarded
Servers can optionally send this response to indicate a call is being forwarded.

182 Queued
Indicates that the destination was temporarily unavailable, so the server has queued the call until the destination is available. A server may send multiple 182 responses to update progress of the queue.

183 Session in Progress
This response may be used to send extra information for a call which is still being set up.

199 Early Dialog Terminated
Can used by User Agent Server to indicate to upstream SIP entities (including the User Agent Client (UAC)) that an early dialog has been terminated.

2xx—Successful Responses

200 OK
Indicates the request was successful.

202 Accepted
Indicates that the request has been accepted for processing, but the processing has not been completed.

204 No Notification
Indicates the request was successful, but the corresponding response will not be received.

3xx—Redirection Responses

300 Multiple Choices

301 Moved Permanently

302 Moved Temporarily

305 Use Proxy

380 Alternative Service

4xx—Client Failure Responses

400 Bad Request
The request could not be understood due to malformed syntax.

401 Unauthorized
The request requires user authentication. This response is issued by UASs and registrars.

402 Payment Required
Reserved for future use.

403 Forbidden
The server understood the request, but is refusing to fulfill it.

404 Not Found (User not found)
The server has definitive information that the user does not exist at the domain specified in the Request-URI. This status is also returned if the domain in the Request-URI does not match any of the domains handled by the recipient of the request.

405 Method Not Allowed
The method specified in the Request-Line is understood, but not allowed for the address identified by the Request-URI.

406 Not Acceptable
The resource identified by the request is only capable of generating response entities that have content characteristics not acceptable according to the Accept header field sent in the request.

407 Proxy Authentication Required
The request requires user authentication. This response is issued by proxies.

408 Request Timeout
Couldn't find the user in time.

409 Conflict
User already registered (RFC 2543)

410 Gone
The user existed once, but is not available here any more.

412 Conditional Request Failed
Conditional Request Failed (RFC 3903)

413 Request Entity Too Large
Request body too large.

414 Request-URI Too Long
The server is refusing to service the request because the Request-URI is longer than the server is willing to interpret.

415 Unsupported Media Type
Request body in a format not supported.

416 Unsupported URI Scheme
Request-URI is unknown to the server.

417 Unknown Resource-Priority
Unknown Resource-Priority (RFC 4412)

420 Bad Extension
Bad SIP Protocol Extension used, not understood by the server.

421 Extension Required
The server needs a specific extension not listed in the Supported header.

422 Session Interval Too Small
It is generated by a UAS or proxy when a request contains a Session-Expires header field with a duration below the minimum timer for the server (RFC 4028)

423 Interval Too Brief
Expiration time of the resource is too short.

424 Bad Location Information
Bad Location Information (RFC 6442)

428 Use Identity Header
Use Identity Header (RFC 4474)

429 Provide Referrer Identity
Provide Referrer Identity (RFC 3892)

433 Anonymity Disallowed
Anonymity Disallowed (RFC 5079)

436 Bad Identity-Info
Bad Identity-Info (RFC 4474)

437 Unsupported Certificate
Unsupported Certificate (RFC 4474)

438 Invalid Identity Header
Invalid Identity Header (RFC 4474)

480 Temporarily Unavailable
Callee currently unavailable.

481 Call/Transaction Does Not Exist
Server received a request that does not match any dialog or transaction.

482 Loop Detected.
Server has detected a loop.

483 Too Many Hops
Max-Forwards header has reach value '0'.

484 Address Incomplete
Request-URI incomplete.

485 Ambiguous
Request-URI is ambiguous.

486 Busy Here
Callee is busy.

487 Request Terminated
Request has terminated by bye or cancel.

488 Not Acceptable Here
Some aspects of the session description of the Request-URI is not acceptable.

489 Bad Event
Bad Event (RFC 3265)

491 Request Pending
Server has some pending request from the same dialog.

493 Undecipherable
Request contains an encrypted MIME body, which recipient can not decrypt.

494 Security Agreement Required
Security Agreement Required (RFC 3329)

5xx—Server Failure Responses

500 Server Internal Error

501 Not Implemented: The SIP request method is not implemented here

502 Bad Gateway

503 Service Unavailable

504 Server Time-out

505 Version Not Supported: The server does not support this version of the SIP protocol

513 Message Too Large

580 Precondition Failure

6xx—Global Failure Responses

600 Busy Everywhere

603 Decline

604 Does Not Exist Anywhere

606 Not Acceptable