3RD QUARTER 2006, VOLUME 8, NO. 3

IEEE COMMUNICATIONS SURVEYS
The Electronic Magazine of Original Peer-Reviewed Survey Articles

www.comsoc.org/pubs/surveys

SURVEY OF SECURITY VULNERABILITIES IN SESSION INITIATION PROTOCOL

DIMITRIS GENEIATAKIS, TASOS DAGIUKLAS, GEORGIOS KAMBOURAKIS, COSTAS LAMBRINOUDAKIS, AND STEFANOS GRITZALIS, UNIVERSITY OF THE AEGEAN, KARLOVASSI
SVEN EHLERT AND DORGHAM SISALEM, FRAUNHOFER FOKUS INSTITUTE

ABSTRACT

The open architecture of the Internet and the use of open standards like Session Initiation Protocol (SIP) constitute the provisioning of services (e.g., Internet telephony, instant messaging, presence, etc.) vulnerable to known Internet attacks, while at the same time introducing new security problems based on these standards that cannot been tackled with current security mechanisms. This article identifies and describes security problems in the SIP protocol that may lead to denial of service. Such security problems include flooding attacks, security vulnerabilities in parser implementations, and attacks exploiting vulnerabilities at the signaling-application level. A qualitative analysis of these security flaws and their impacts on SIP systems is presented.

One of the main challenges that telecommunication providers are facing is the convergence of data and voice networks. The idea of utilizing data networks for transmitting voice was originally proposed in 1970 [1], while the Internet evolution has pressed telecommunication providers and Internet Service Providers (ISPs) to transmit Voice over Internet Protocol (VoIP). VoIP utilizes one common network for signaling and voice, thus enjoying several advantages [2] and offering modern telephony services like *instant messaging, Internet conferencing rooms, personalized call transfer, and so forth.*

In PSTN, security, reliability, and availability rely on a closed networking environment dedicated to a single service (namely, voice). On the other hand, VoIP is based on an open environment such as the Internet, which simplifies mounting an attack (e.g., on a VoIP server). This is due to the fact that VoIP services are based on open standardized protocols for signaling (i.e., SIP, H.323, MGCP) and transport protocol technologies (i.e., RTP), using servers reachable through the Internet and often provided over general purpose computing hardware. As a result, these technologies are not designed mainly with security features/functionalities in mind. Therefore, a malicious user can exploit any possible misconfiguration in the aforementioned signaling or voice protocols, attempting to disturb or disrupt VoIP services. Additionally, such services inherit numerous vulnerabilities from the utilization of the underlying transport protocols like TCP, IP, and UDP. For example, instead of generating thousands of costly voice calls as required in PSTN, the attacker can easily and in a similar manner generate and send thousands of VoIP signaling messages to attack VoIP servers.

Session Initiation Protocol (SIP) [3] has been adopted as the signaling protocol to handle multimedia sessions at both the Internet and the 3G realms [4]. This article aims to identify and describe existing and potential new categories of security threats that a SIP-based application service provider will have to face and deal with. Despite the diverse security mechanisms that have been proposed for SIP-based infrastructures [3], there are still vulnerabilities that affect this architecture. Such vulnerabilities aim to exhaust available resources, create false responses upon to the reception of malicious requests, and discover possible security vulnerabilities in the applications.

The rest of the article is organized as follows: we briefly present background information regarding SIP-based infrastructures. We discuss and analyze current SIP signaling security mechanisms while we describe potential new security concerns and DoS attacks in SIP that cannot be resolved by the mechanisms described in the previous section. We give a qualitative analysis for attacks presented. The last section concludes the article and gives pointers to some future work.

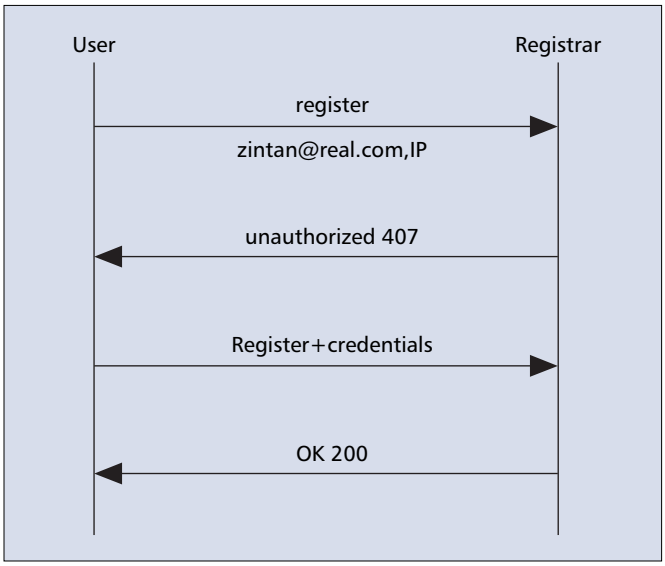

Figure 1. *Registration procedure.*

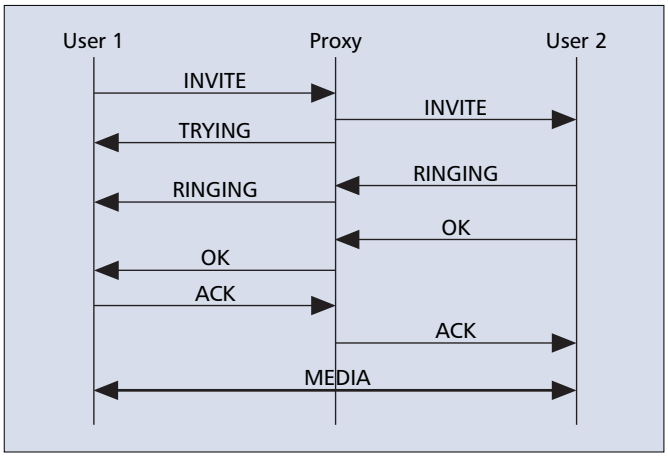

Figure 2. *Calling a user in SIP.*

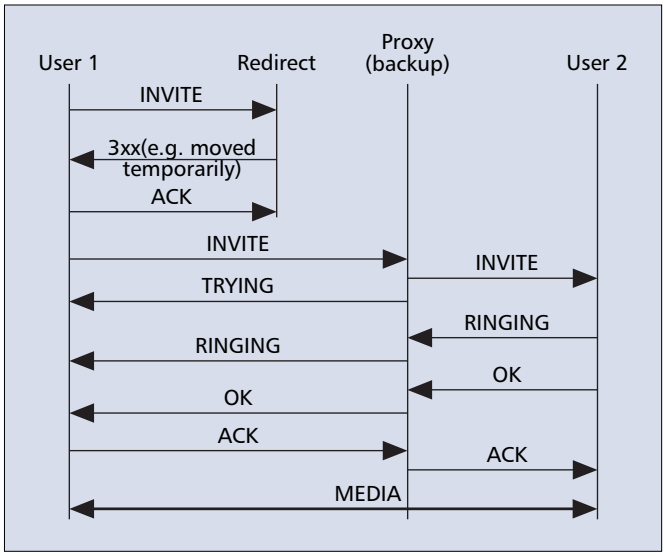

Figure 3. *SIP Call utilizing a Redirect proxy.*

The SIP Architecture

SIP is an application-layer signaling protocol [3] for handling multimedia sessions over the Internet. In a typical SIP-based network infrastructure, the following network elements are involved:

- **User Agents**: user agents (UAs) act on behalf of an end user terminal. A user agent client (UAC) is responsible to create requests and a user agent server (UAS) processes and responds to each request generated by a UAC.
- **Registrar**: UAs contact registrar servers to announce their presence in the network. The SIP registrar server is a database containing locations as well as user preferences as indicated by the UAs.
- **Proxy**: A proxy server receives a request and forwards it towards the current location of the callee — either directly to the callee or to another server that might be better informed about the actual location of the callee.
- **Redirect**: A redirect server receives a request and informs the caller's UA about the next hop server. The caller's UA then contacts the next hop server directly.

Various types of text based messages have been introduced in SIP following the HTTP message structure [5]. SIP messages must also identify the requested resource, which corresponds to a unique address. The SIP address (SIP-URI) is aligned with the general form of the HTTP addressing scheme, which is: "address_scheme:resource." As a result, a user is identified through a SIP URI in the form of sip:user@domain. As an example, the URI sip:zintan@real.com is a valid SIP address. This address can be resolved by a SIP proxy that is responsible for the user's domain. The first step for a user to use a SIP-based service is to identify his/her actual location in terms of an IP address. Consequently, the user needs to register the combination of his/her SIP address and current IP address at the SIP registrar responsible for his domain. This registration procedure is depicted in Fig. 1.

When inviting a user to participate to a call (callee), the calling party (caller) sends a SIP INVITE to the corresponding SIP proxy, which checks in the registrar's database or in the Domain Name System (DNS), the location of the callee and forwards the invitation to the callee. The latter can either accept or reject the invitation. During this message exchange, both the caller and the callee exchange the addresses/ports at which they would like to receive the media as well as the type of media (i.e., video, voice) they can accept. After finalizing the session establishment, the end systems can exchange media data directly without the involvement of any SIP proxy. This procedure is depicted in Fig. 2.

However, under certain circumstances the aforementioned procedure is not feasible because the corresponding proxy may be temporarily unavailable (e.g., through overload, or because of a software update). Under such situations the mediation of a Redirect server is required in order to inform the caller (user 1) on possible alternative locations to reach the requested URI. As soon as the caller receives this information, he/she generates a new request towards one of the alternative locations. This procedure is depicted in Fig. 3.

SIP Signaling & Media Security

The development of new services for the establishment of multimedia sessions over the Internet requires security mechanisms to protect the transmitted data against modification, eavesdropping, session disruption, imitation, and so forth. These kinds of attacks can take place either during the signaling phase or during the transmission of the media packets (e.g., voice). Thus, both signaling and media data demand the utilization of certain security mechanisms. This section provides information about SIP-based signaling services security as well as for the corresponding transmitted media data, after the session has been established.

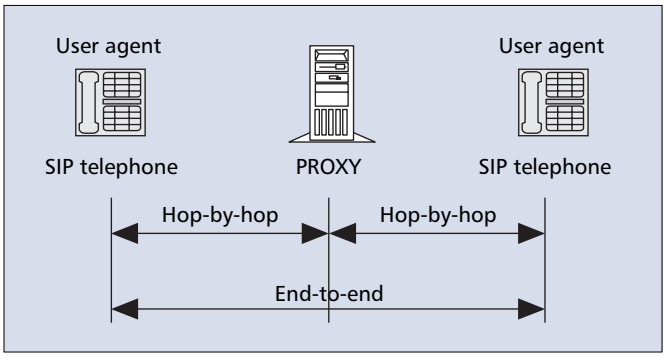

Figure 4. *SIP security mechanisms.*

Available Security Mechanisms for SIP

The SIP specification [3] does not include any specific security mechanisms. On the contrary, the utilization of other well-known Internet security mechanisms is suggested. As illustrated in Fig. 4, SIP security can be provided either in a hop-by-hop or end-to-end fashion.

More specifically, the following security methods are described in [3]:

- **SIP Authentication**: The digest authentication algorithm specified in RFC 2617 [6] is a challenge response based protocol and until now the most frequently deployed security mechanism with SIP for verifying the identity of users and performing message authentication. Based on the preconfigured settings of SIP servers, a server might want to authenticate the sender of a SIP request before forwarding his/her request. This authentication mechanism can be applied to certain requests only, certain users or requests coming from certain proxies or redirect servers. The following message sequence of messages is required to perform HTTP Digest authentication in SIP:
  - –The client sends a SIP message (i.e., a SIP REGISTER), to a server requiring authentication, which in turn responds either with a *proxy authentication required* (407, proxy server response) or *unauthorized* (401, registrar or redirect server response) reply.
  - –This reply contains a WWW-Authenticate header, including a challenge that will be used by the client to compute the credentials.
  - –Upon reception, the client creates a new SIP message including an authorization header with the computed credentials. A detailed description of the credential computation procedures can be found in [6].
- **IPsec and SIP**: The use of IP to transport SIP messages is vulnerable to attacks like spoofing, session hijacking, traffic analysis, and so on [7]. The IP Security (IPsec) protocol [8] acts on IP layer independently and provides a set of services to protect IP packets from such kind of attacks. IPsec can offer confidentiality, integrity, data-origin authentication services, as well as (optionally) anti-replay and traffic analysis protection by utilizing the Encapsulating Security Payload (ESP) and Authentication Header (AH) protocols. Introducing IPsec in SIP can safeguard signaling and data from various network vulnerabilities, provided that some sort of trust (e.g., pre-shared keys, certificates) has been established beforehand between the communicating parties. For instance, 3GPP in UMTS release 5 employs IPsec (ESP) to protect SIP signaling between the UE and the proxy server (P-CSCF) which resides in the serving network's IP multimedia subsystem (IMS)
- **Transport Secure Layer (TLS)**: Another solution to protect SIP communications is the use of the Transport Layer Security (TLS) protocol [9]. Authentication for the corresponding network elements during the handshaking procedure can be mutual and is performed by exchanging their certificates. TLS has many of the advantages of IPsec and the successful introduction of the protocol in the wired Internet has proved its usability and effectiveness. Likewise, TLS can be part of SIP environment, as it runs above TCP/IP and higher-level protocols such as HTTP or FTP; consequently, the TCP header is not encrypted. However, TLS cannot be combined with UDP. In addition, keeping up many TCP connections open simultaneously may be too heavy for SIP proxy servers. SIP provides a notation to request a secure connection with the SIP Secure (SIPS) URI, (e.g., sips:dgen@aegean.gr). It must be noted that, in every case, TLS support is not yet fully implemented in current SIP UAs. Among some of the currently working solutions are Kphone [10], Minisip [11], and hardware phones from Snom [12].
- **Authentication, Authorization, Accounting Services in SIP**: In order to authenticate and/or authorize users and utilize the corresponding accounting services, it is typically more convenient for SIP entities to communicate with an authentication, authorization, and accounting (AAA) sever than attempting to store users' credentials and profiles locally as required by the HTTP Digest. Moreover, AAA gives the ability to administrators to dynamically configure the type of authentication and authorization required (e.g., per user or per service). Besides that, the interconnection of heterogeneous networks usually mandates access in the SIP-based multimedia services independently of the utilized access network. This fact gives mobile users the ability to gain access to multimedia services when roaming to different administrative domains. For instance, consider the case in which a mobile user may use a SIP proxy located in the visited (serving) network, but his/her SIP messages may be finally proxied back to a SIP server in the home network that implements call control features. This situation is typical for 3G subscribers. More details about mobility in SIP and its associated security issues can be found in [13]. Moreover, user mobility necessitates the employment of additional AAA mechanisms for mobile users. In order to provide the corresponding AAA services in such hybrid environments, the utilization of protocols like Radius [14] or Diameter [15] is suggested. Both of these protocols have been proposed for employment in the SIP core architecture. Radius or Diameter SIP-oriented applications can be used in a SIP environment where the corresponding interface to the AAA infrastructure is required to authenticate and authorize the usage of SIP resources. The description of such applications can also be found in the literature [16, 17]. Furthermore, in order to take advantage of AAA services in SIP the utilization of the appropriate security requirements as described in RFC 3702 [18] is mandatory
- **S/MIME and SIP**: SIP messages are capable of carrying MIME bodies [3]. The provisioning of security services can be accomplished by utilizing Secure MIME (S/MIME) [19]. S/MIME provides a set of functionalities of which SIP utilizes two [3]: integrity and authentication tunneling and tunneling encryption. However, this solution mandates the deployment of a global S/MIME Public Key Infrastructure (PKI). Otherwise, the exchanged public keys would be self-signed, which makes the initial key exchange susceptible to man-in-the-middle (MITM) attacks.

Header	Request URI	From	To	Via	Record route	Record	Call Id	Cseq
Modification allowed	Yes	No	No	Yes	Yes	Yes	No	No

Table 1. *Legal modifications in SIP messages.*

Analysis of the SIP Security Mechanisms

As stated above, the HTTP digest authentication algorithm is currently the most frequently deployed security mechanism with SIP. This authentication scheme can offer one-way message authentication and replay protection but cannot support message integrity and confidentiality. According to RFC 3261 [3], it is possible for a malicious user to place spam calls. Moreover, this method is vulnerable due to the use of plaintext, which enables MITM attacks, as both the plaintext (challenge) and the ciphertext can be easily captured by a potential aggressor simply by sniffing the network traffic. Digest authentication also requires some sort of prearranged trusted environment for password distribution. Passwords may be stored either in plaintext or ciphertext form at the server side. However, ciphertext cannot offer an advanced security level, since it is feasible to compute the message credentials by launching a brute force attack on the encrypted password. Besides, due to the absence of any correlation between the user name and the SIP URIs, a malicious user may masquerade itself as a legitimate user. Recently, various solutions have been suggested [20, 21] to recover of such limitations found in the HTTP Digest mechanism. Nevertheless, it is stressed that such solutions require modifications in the SIP user agent, which of course is not always easy to implement.

Furthermore, considering that there is no authorization model, it is possible for an attacker to gain access to services that are normally available to legitimate users only. Another important issue is that the intermediate SIP proxies cannot be certain that the SIP UA has been authenticated. It has already been suggested in [22] that SIP messages must include a cryptographic token to confirm that the originating user's identity has been verified by the corresponding network. Performance issues are also reported for authentication procedures. Simulations showed that they highly strain SIP servers' performance [23].

In relation to authentication issues, it is of equal importance to protect the user's personal information and his real identity providing anonymity, privacy, and location privacy. SIP UAs can support anonymity by obscuring the *From*: header contained in SIP requests. However, not all headers can be obscured. For instance, the *Contact*: header is required for request routing and cannot be protected. Consequently, a satisfying level of privacy is not possible without adequate support from the SIP proxy infrastructure. As suggested in [24], the privacy service can be implemented in a proxy server that can also act as a back-to-back UA and proxy media streams.

As mentioned above, the protection offered by IPsec assumes preestablished trust among the communicating parties and it can only be utilized in a *hop-by-hop* fashion. Since IPsec is implemented at the operating-system level, most SIP clients do not implement this protocol yet. For this reason, IPsec can only protect the traffic between the corresponding network servers. Moreover, SIP specifications do not suggest any framework for key administration, which is required by the Internet Key Exchange (IKE) part of the IPsec protocol. However, recently has been suggested a draft describing the corresponding requirements for IPSec negotiation in SIP [13].

In contrast to IPsec, TLS does not assume any trust relation among communicating parties. TLS can be utilized either for one-way or mutual authentication schemes and maybe it is more suitable for inter domain authentication. Of course, there is always the risk that the message can be intercepted inside the recipient's network assuming that the last hop is not encrypted. Additionally, TLS is used by the SIPS scheme to offer an *end-to-end* security. However, TLS fails to deliver end-to-end security as, at least until now, no mechanism exist to ensure that along the whole path from the source to the destination in a hop-by-hop fashion TLS is utilized by all the involved parties. Recently, some security requirements and directions for providing such a mechanism have been suggested [25]. Moreover, TLS protects only connection-oriented protocols. To put it simply, the lack of PKI in VoIP does not offer the appropriate environment for the utilization of TLS.

S/MIME is used to support either integrity or confidentiality in an end-to-end fashion. It should be noted that S/MIME adds considerable overhead in SIP messages. More importantly, the integrity and confidentiality of the entire SIP message cannot be protecteed due to the existing restriction of header modification (Table 1), as the intermediate nodes must have access to the SIP header to process and route the SIP message to the appropriate destination. Finally, as in the TLS case, the absence of PKI is an additional restriction for the operation of S/MIME in SIP.

Apart from the abovementioned restrictions, in some cases, security services may require the combination of TLS and S/MIME. This includes the usage of TLS to support integrity and authentication, while S/MIME is used to provide mainly privacy for some parts of the transmitted data. However, some SIP intermediaries (e.g., servers) may require reading these data. This situation requires a security mechanism to secure message bodies and/or headers between the UA and the proxy servers, while at the same time revealing information to those that actually need it. This is called an "end-to-middle security." Security requirements for "end-to-middle" security" can be found in [26].

Media Security

While the first level of defense is to protect the signaling data to establish a multimedia connection, the second one is the protection of the transmitted multimedia. As a result, the issue of media security is tightly bound with signaling security. In fact, all media information is passed within signaling messages by making use of Session Description Protocol (SDP) [27]. After the session has been established, the real-time protocol (RTP) [28] is employed in order to transport multimedia-data (real-time traffic) between the communicating parties. Real traffic needs to be sent and received in a very short time period. Two examples of such traffic are: audio conversations between two users, and playing individual video frames at the receiver as they are received from the transmitter. Moreover, such traffic has specific requirements for end-to-end delivery. Thus, RTP provides the appropriate services (such as time reconstruction, loss detection, etc.) for data with real-time characteristics. However, RTP specification does not provide any specific mechanisms for protecting the transmitted data against eavesdropping or other active attacks, but rather suggests the utilization of the underlying network security mechanisms. As a result, Secure RTP (SRTP) [29] is proposed as it is designed specifically for media packets to provide the corresponding security services like confidentiali-

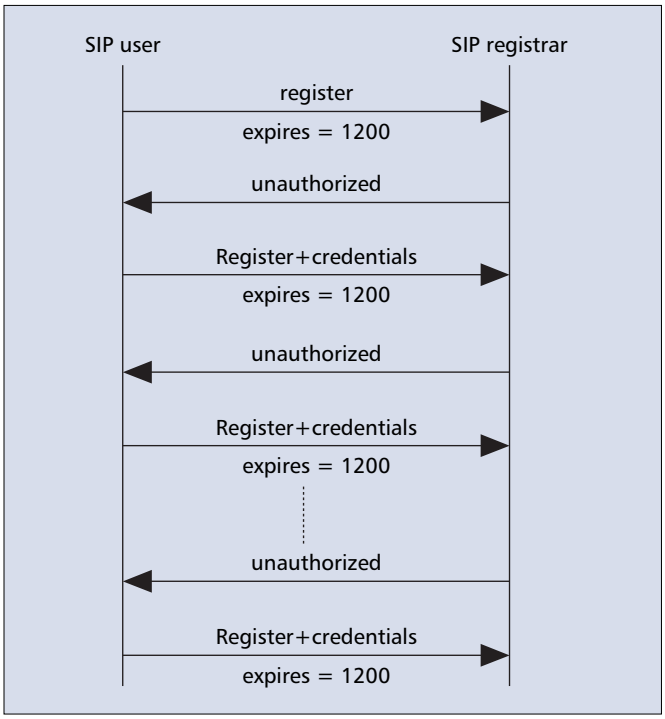

Figure 5. *Attack against SIP registrar server.*

ty, message authentication, and replay protection to the RTP traffic. SRTP defines a strict format for security services, specifies encryption algorithms to use, and finally supplies a key derivation mechanism. Note, that SRTP encrypts only the payload of a voice packet without adding additional encryption headers.

As it is designed specifically for streaming real-time data, secure RTP is more efficient than IPSec in terms of bandwidth [29]. Moreover, SRTP is suitable for voice privacy and confidentiality in LAN environments to protect against internal threats. For example, voice data will be protected against eavesdropping, if a given user initiates a secure call towards his/her interlocutor. The need for such services seems to be urgent for VoIP services, in contrary to PSTN, because the possibility to eavesdrop over an IP call is much greater. For example, the aggressor may easily use well-known open source tools like ethereal to capture RTP packets.

Security Vulnerabilities and Denial of Service in SIP Based Networks

Despite the use of security mechanisms, SIP Services are subject to certain vulnerabilities. The aim of any kind of attack is either the interruption/destruction of service provisioning known as denial of service (DoS) or to gain some sort of unauthorized access in computation resources. While SIP utilizes well-known Internet technologies, it inherits all known threats and vulnerabilities that exist in the Internet realm.

Moreover, there are also specialized attacks on the SIP protocol itself. All the elements of the SIP architecture (i.e., proxy-registrar servers and end-user devices) are vulnerable to these kinds of attacks. It is possible that such attacks may have a different form whether they take place either in a network element (e.g., proxy server) or in an end-user device.

One of the most well-known methods to create problems in the availability of the provided service is the consumption of existing resources by creating a large number of requests against the providing VoIP service. Another possible vulnerability is the exploitation of developing errors in SIP servers. Finally, as Internet telephony is considered a service, the attacker will try to discover possible security flaws on the applications level or take advantage of existing protocol misconfigurations similar to attacks in Internet applications and services.

Flooding Attacks in SIP

A flooding attack against an Internet application or service can be launched either from a single or multiple sources. The latter exploits 'innocent' Internet hosts, known as attack reflectors, which create a large number of requests (e.g., TCP connection requests) against the victim. This kind of attack is called a reflection distributed DoS (RDDoS) attack [30]. Such an attack as well as other generic-transport layer attacks (see a CERT report [31]) can be used to paralyze SIP infrastructure, too. Below we describe how this type of attack can be launched towards the SIP infrastructure.

Flooding Registrar Server — One of the cardinal network elements in SIP telephony service is the registrar. When an attacker manages to paralyze the registrar (e.g., by sending numerous bogus registration requests), it can easily cause a DoS. This situation can be avoided only if the SIP server blocks all messages coming from unknown origins.

As discussed eariler, a REGISTER request can add a new binding between a user's SIP address and one or more contact addresses (currently IP addresses) so that the user can utilize the provided telephony service. Consequently, when the attacker launches an attack against a REGISTRAR by employing a large number of registration requests, he/she aims to accomplish one of the following goals:

- To guess legitimate users' passwords
- To cause a DoS in the SIP registrar

Figure 5 depicts such an attack scenario against a SIP registrar server. This attack can be possibly launched by employing different ways to send a SIP message, each time changing

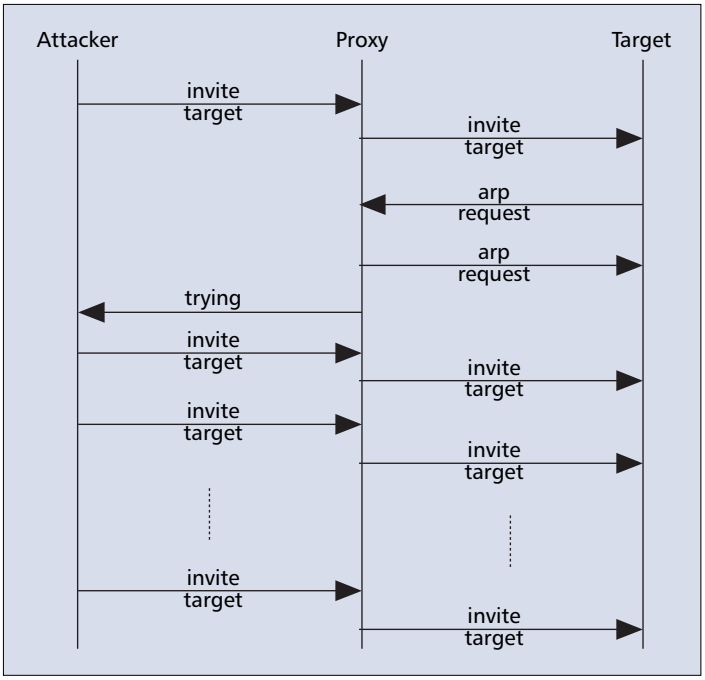

Figure 6. *Flood with INVITE messages.*

just a few parameters. For example, the attacker can try to de-register the legitimate user to cause DoS. The only difference between registration and deregistration (terminate the session) procedure is the value of the EXPIRES header. This header is set to zero when the UA wants to terminate the session (deregister). In this way, an attacker will try to evade any existing countermeasure. In both procedures, the attacker needs to "guess" the legitimate user's password.

This type of attack can also be launched in a distributed manner. As an example, multiple attackers can either undertake the task to find out a legitimate user's password or disrupt the provided service by sending simultaneous REGISTER messages to the registrar, as the authentication procedure is considered computationally expensive.

Flooding Proxy Server and End-User Terminal — One of the most utilized messages that must be processed by SIP proxy servers is the INVITE message. The INVITE message, as described in the above section "The SIP Architecture," is used to establish a connection among two or more participants in a SIP session. Until this connection is established, the SIP proxy must keep the connection state. This fact makes the proxy most vulnerable to flooding attacks.

According to RFC 3261 [3], upon forwarding an INVITE by the SIP proxy, a timer of minimally three minutes is set. After the timer expires, the callee is considered to be incapable of providing a final response (i.e., a response between 200 and 699). Also, after forwarding a final non-200 response (i.e., a response between 300 and 699), the server needs to wait for the ACK message and retransmit the response for a period of up to 64 × T1 seconds, where T1 is usually set to 500 msec. In case the server has forked a request to different destinations, the server must maintain a copy of the incoming request as well as a copy of all the forked requests. In case the server receives a response indicating a redirect situation, the server might initiate the redirect transaction by himself. In this case the server must maintain the state until the redirect transaction has been replied as well. The corresponding server has to retransmit a 2xx response periodically, as it cannot guarantee an all-reliable connection.

An attacker can possibly launch a flooding attack by utilizing INVITE messages not only against a proxy, but also against an end-user's terminal. For example, end-user devices have been designed mainly to respond under normal conditions. This means that they are able to process few incoming messages simultaneously. Considering the situation where an attacker impersonates himself as a legitimate user, he will possibly generate numerous INVITEs as illustrated in Fig. 6. In this situation the attacker builds up INVITEs only, without waiting for any respond message trying to paralyze the victim. Additionally, in this scenario the SIP proxy is utilized by the attacker to amplify the generating INVITE messages.

Another scenario that an attacker could possibly exploit is illustrated in Fig. 7. In this scenario, the attacker tries to behave as a legitimate UA. The attacker will try different INVITE scenarios so as to cause a DoS either in the proxy or in the end-terminal device by attempting to evade any countermeasure or identification mechanism in place.

Moreover, a legitimate user can launch this attack even unintentionally if a "poor" implementation exists, which contains developing errors.

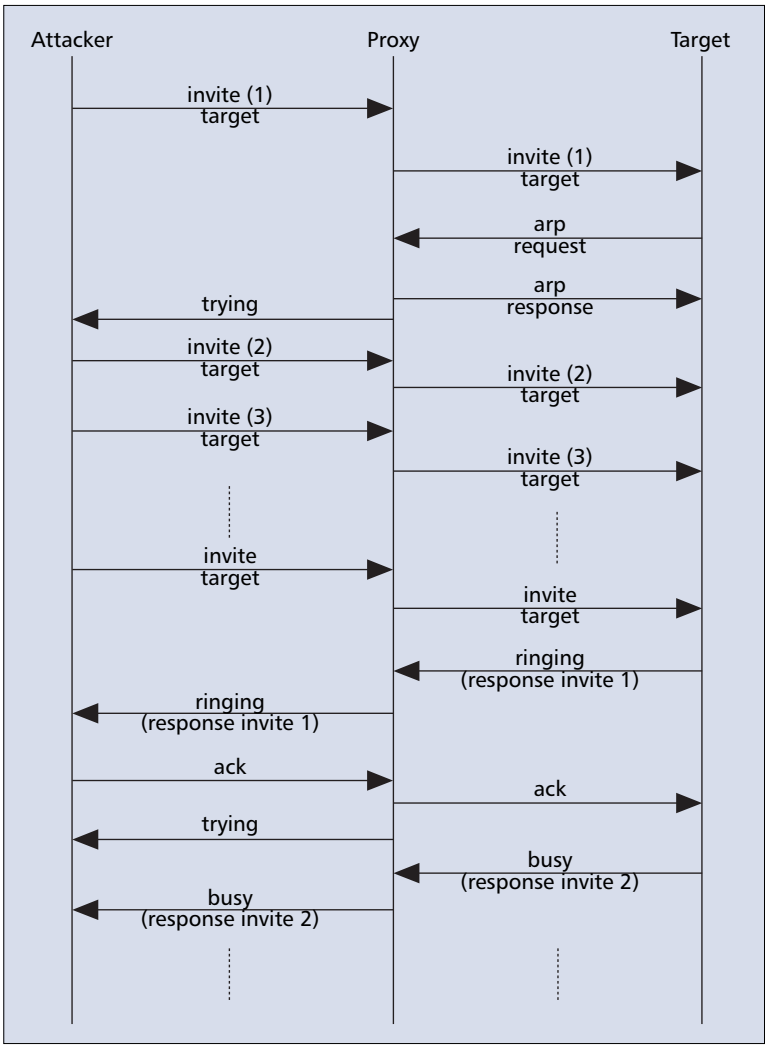

Figure 7. *Alternative flooding INVITE scenario.*

SIP Parser Attacks

As SIP is a text-based protocol with a highly degree of freedom, an efficient parser is needed which only parses messages up to the point the information is required. However, even a perfectly valid SIP message can be constructed in a way to hamper proper parsing. Here we give a list of possible scenarios that complicate message parsing.

An attacker can create unnecessarily long messages in a simple way by adding additional headers (such as informative header fields, e.g., Supported) in conjunction with a large message-body. Many SIP messages may include bodies, even when they are not needed in every message. Istead of only depleting processor power, longer message also increase network utilization and memory usage. For an attacker to be effective with this method, he has to utilize only well-formed header fields, as other header fields should be ignored by a well implemented parser. Server implementations should thus check messages for a certain size limit and reject messages exceeding this limit with a 413 (Request Entity Too Large) message.

Under certain conditions clients have to send messages using a congestion controlled protocol, which generally results in the usage of TCP. To avoid fragmentation, the condition is met if a request is within 200 bytes of the path MTU, or if it is larger than 1300 bytes and the path MTU is unknown. By forcing a server to accept TCP connections, it becomes vulnerable to general TCP DoS attacks, as additional state is created, even in a stateless proxy. As a countermeasure, SIP

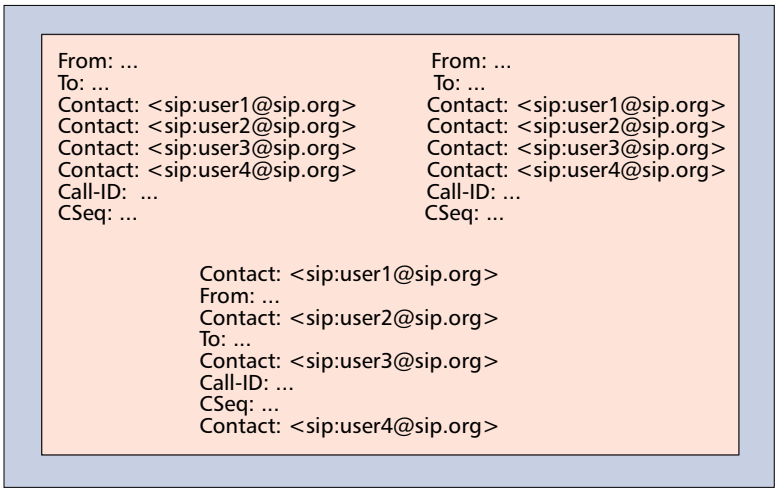

Figure 8. *Multiple header possibilities.*

entities could be configured to not support TCP messages. This behavior is conform to the current SIP RFC, but is expected to be deprecated in a further revision of the SIP specification.

Poor parser implementations can be rendered inoperable by including message bodies of a size that does not match the one indicated in the Content-Length header.

Additionally, the SIP standard mandates that headers that have multiple values can be separated into individual header fields so that each only contains one value. If multiple message headers of the same field are included in a message where theses headers are spread all over the message, this will further complicate the parsing. Figure 8 illustrates three possibilities to compose a message with multiple Contact fields. Especially, the following header fields can be distributed in such a way in a SIP message: Accept-Encoding, Accept-Language, Alert-Info, Allow, Authentication-Info, Call-Info, Contact, Content-Encoding, Content-Language, Error-Info, In-Reply-To, Proxy-Require, Record-Route, Require, Route, Supported, Unsupported, User-Agent, Via, and Warning.

Some message headers are more vital for processing than others. Vital header fields are all routing-specific fields (such as To, Via, Route, etc.), so messages with these fields placed towards the end of the message are more complicated to parse. One way to accomplish this is by inserting multiple informative header fields before the routing fields, for example, Allow or Supported.

Parsing attacks can be countered by an efficient implementation, for example, by parsing only those parts that are needed for its correct functioning. In general, a server that is overloaded with message parsing is an indication of a bad implementation of the server or under dimensioned hardware. Additionally, monitoring incoming messages for suspicious content will further mitigate parser attacks.

SIP Application-Level Attacks

In addition the aforementioned attack categories, and likewise with regard to various vulnerabilities found in Internet application protocols and services, an attacker might try to exploit other SIP protocol's weaknesses, due to the security vulnerabilities found in the protocol itself or in the provided service.

Attacks Based on SIP Signaling — The SIP protocol specification [3] describes methods to end/terminate a session, cancel an invitation, redirect a call, and update session

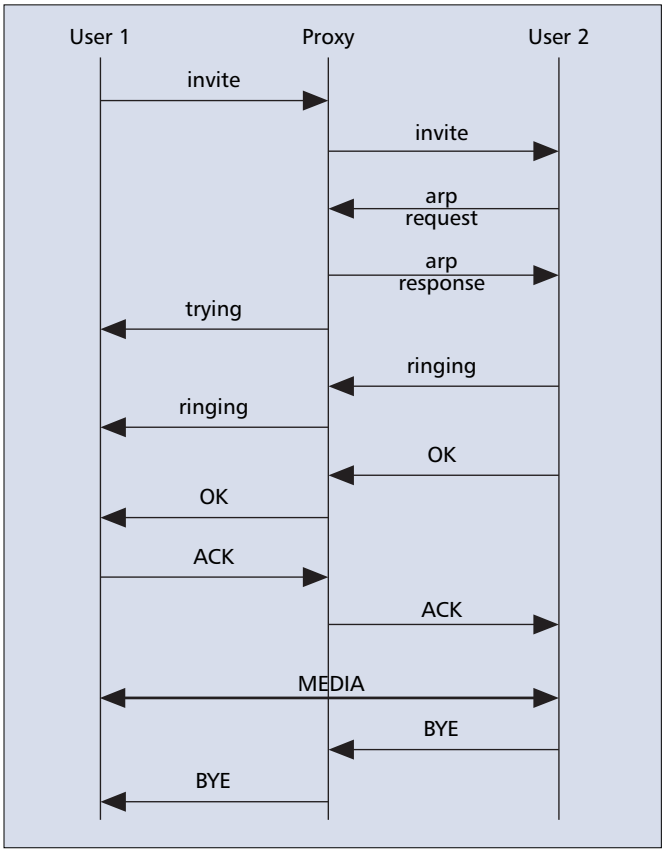

Figure 9. *Normal session termination.*

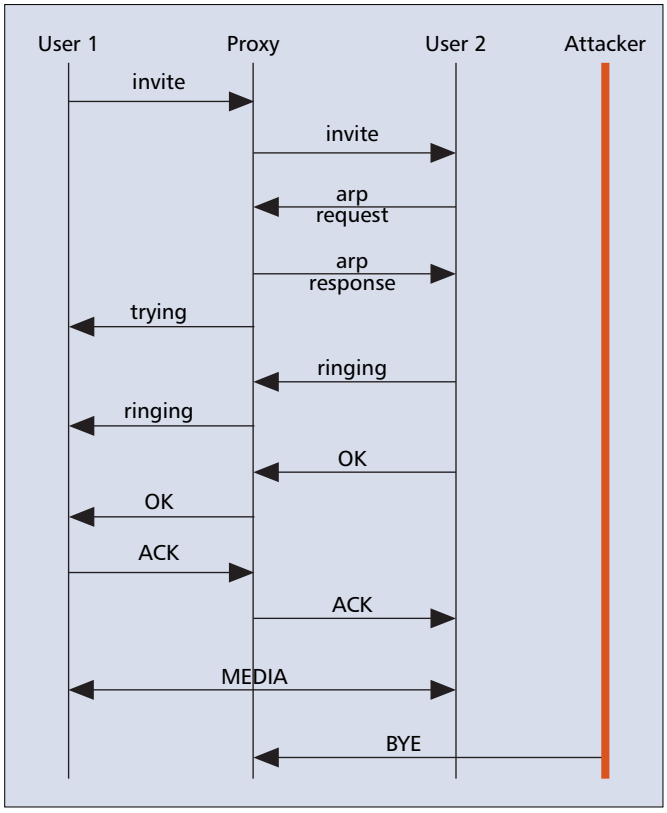

Figure 10. *BYE attack.*

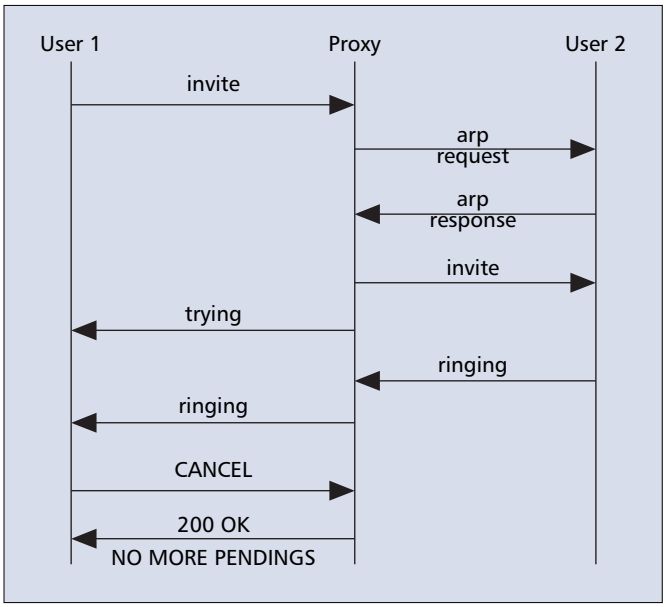

Figure 11. *CANCEL request.*

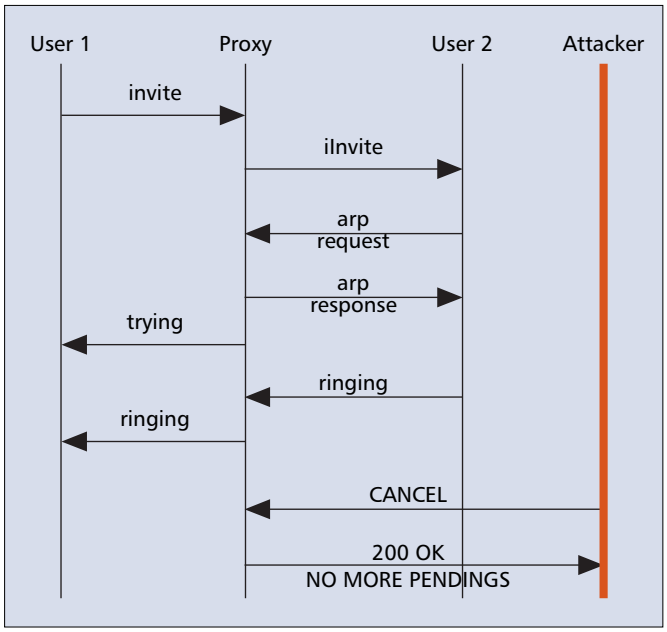

Figure 12. *CANCEL attack.*

parameters. It is very likely that the attacker will try to exploit any security vulnerability in the aforementioned methods and cause DoS to the provided service. The main reason that an attacker can launch attacks by employing these messages is the utilization of improper authentication mechanism. At the perils, current SIP specifications do not mandate authentication for all of the aforementioned methods. More specifically, for each of the previous procedures the following SIP attacks could be launched:

- **BYE ATTACK**: The BYE request is used to terminate an established session, as shown in Fig. 9. An attacker possibly can utilize the BYE request to tear down a session, as depicted in Fig. 10. To launch this attack, the attacker needs to learn all necessary session parameters (e.g., Session-ID, RTP Port, etc.). This can be accomplished either by sniffing the network traffic or performing a MITM attack to insert a BYE request into the session. The BYE method as mentioned above is used to terminate an established media session. However, this attack can be launched successfully only in the case when no authentication mechanism is in place, considering of course the attacker's ability to discover the current session parameters.

Thus, the protection of the session's critical parameters regarding confidentiality must be considered mandatory. As discussed previously, either TLS or IPSec can be employed to provide such kind of security services. Moreover, the authenticity of a BYE message must be ensured by utilizing either HTTP Digest or TLS.

- **The "CANCEL" ATTACK**: The CANCEL request, as its name implies, is used to cancel a previous request sent by a client. More specifically, it asks the corresponding server to cease processing the request and generate an error response designating that request. This procedure is shown in Fig. 11. The attacker may utilize the CANCEL method to cancel an INVITE request generated by a legitimate user, as illustrated in Fig. 12. A CANCEL request must only be sent to cancel an INVITE request [3]. Thus, when a SIP-proxy receives a CANCEL request for any other message type (than INVITE), it must not process this message, but rather produce an appropriate error response. Moreover, incoming CANCEL requests must not be processed if the original request has already generated a final response. This is because CANCEL has no effect on requests that have already generated a final response.

It must be mentioned that CANCEL requests are generated in a hop-by-hop fashion and cannot be resubmitted. As a result, they cannot be challenged by the server in order to get proper credentials in an Authorization header field. Thus, the utilization of any applicable, underlying security mechanism, such as IPSec or TLS, is considered mandatory. However, the processing of an incoming CANCEL message from a different administrative SIP domain is still an open and unresolved issue. Additionally, the monitoring of INVITE messages that have not already generated a final response could possibly help to identify any illegitimate CANCEL requests.

- **The "REFER" ATTACK**: The REFER extension [32] provides a mechanism where one party (the referrer) provides a second party (the referee) with an arbitrary URI to reference. Assuming that this URI is a SIP URI, the referee will send a SIP request (usually a SIP INVITE), to that URI (the refer target). As a result, REFER can be used to enable many applications, including call transfer. RFC 3892 [33] extends this method by allowing the referrer to provide information about the REFER request to the refer target using the referee as an intermediary. The refer target can use this information to decide whether to accept the reverenced request from the referee or not. This scheme enables the referee to act as an eavesdropper, giving him the ability to launch MITM attacks. For example, the referee can forge the Referred-By header or/and eavesdrop on the referred-by information. The referee may also copy all the related information into future unrelated requests. Although the specification uses an S/MIME-based mechanism to enable the refer target to detect possible manipulation of the Referred-By header data, this protection is completely optional.
- **The "Re-INVITE" ATTACK**: Once a dialog-session has been established by initial messaging, subsequent requests can be sent that attempt to modify the parameters of the dialog-session (e.g., address or port modification). Thus, any unauthorized modification with a forged re-INVITE (Fig. 13) of a dialog-session by a potential attacker may cause a DoS.
- **The "UPDATE" ATTACK**: The SIP UPDATE method [34] gives end users various capabilities, such as muting

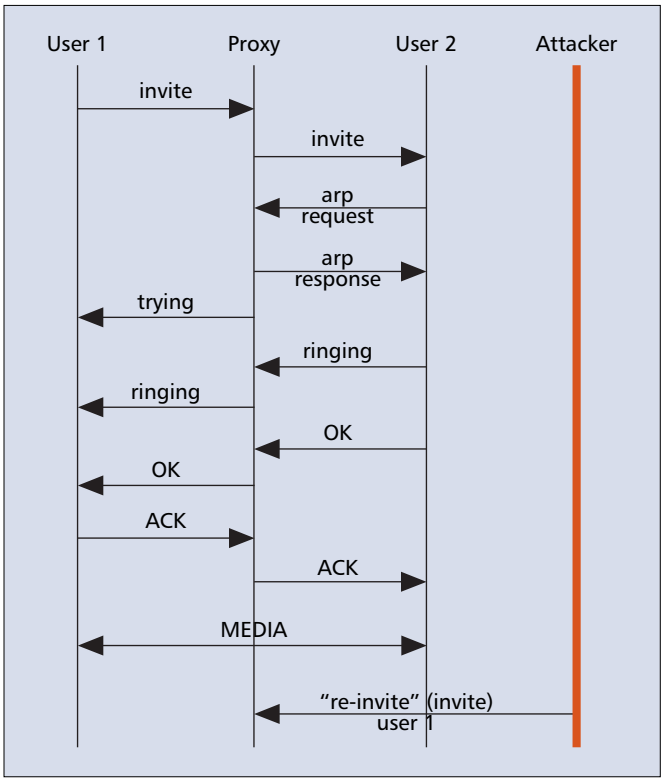

Figure 13. *"Re-INVITE" attack.*

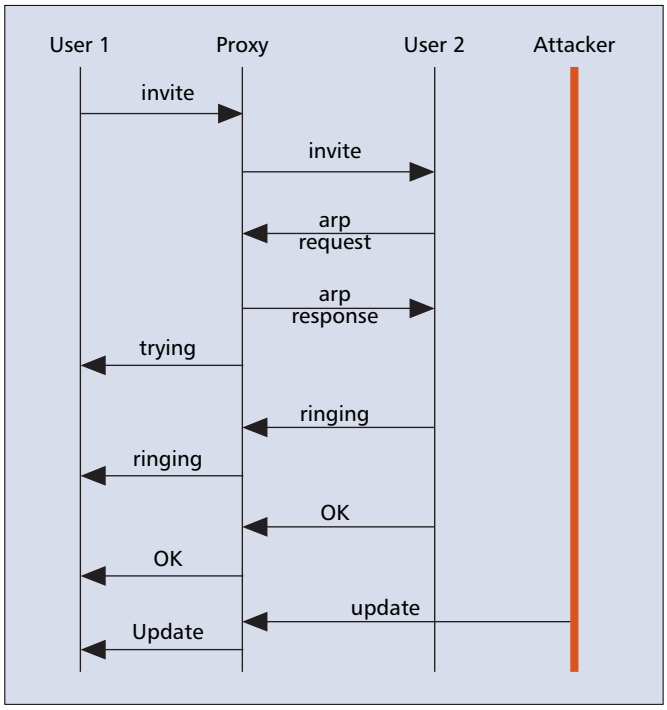

Figure 14. *UPDATE attack.*

or placing on hold incoming calls, identification of QoS service, and negotiation for other session attributes like "RE-INVITE." The only difference is that "RE-INVITE" can be utilized only after a session has been established, while UPDATE is utilized to modify session parameters before the final response to the initial invitation. So, similar to the "RE-INVITE" attack, an attacker may send a forged UPDATE message, as depicted in Fig. 14, in order to modify the initial session parameters to cause a DoS change of parameters like QoS or initial addresses and ports.

- **The "INFO" ATTACK**: In many cases, SIP networks can be used as a mediator to interconnect the PSTN carrier. The reasoning for this case involves SIP for telephones (SIP-T) [35] being used in order to convey PSTN signaling from one PSTN carrier to another and vice versa. Figure 15 depicts an architecture in which a SIP network is the bridge between two different PSTN networks.

The INFO method is described as a general mechanism to carry application-level information along the SIP signaling path so as to allow tunneling mechanisms [36]. It has been proposed (and, in fact, used) for a wide variety of functions, including:

- Carrying mid-call PSTN signaling messages between PSTN gateways
- Carrying DTMF digits generated during a SIP session
- Carrying account balance information

The message body of an INFO message can be encrypted for privacy reasons. However, there is no suggestion for any security mechanism to provide integrity and authenticity of INFO method. Thus, malicious modification of the INFO method is possible and it can cause serious problems for the communication parties like unauthorized access to a call, DoS for the initial invitation, billing errors, and so forth.

SQL Injection Attack in SIP — In order to store and administer user credentials and appropriate data for providing value-added services to end-users, SIP relies on databases such as MySQL [37], Postgress [38], or Oracle [39]. This fact makes SIP-based services and specifically any authentication procedure vulnerable to attacks similar to a known Internet attack known as SQL Injection.

Open-source SIP implementations (e.g., SER [40], VoVida [41]) provide build-in modules in order to support MySQL and Postgress databases for administration purposes. This database schema is composed of various data tables. Among them, Subscriber and Location tables are of major importance, as they store critical data required for smooth VoIP operation. More specifically, the "Subscriber" table stores the appropriate data (such as user name, domain, password, etc.) for SIP authorized users, while the "Location" table stores all the data representing the current available contact addresses for the legitimate subscribers.

In case an SQL injection attack is triggered against a SIP installation, any corruption in the integrity of database and especially in the "Subscriber" and "Location" tables drives the provisioning of services to fail. Furthermore, the utilization of WEB interfaces for the provision of SIP services makes this attack more attractive to the potential perpetrators.

The concept of SQL injection in SIP is similar to the SQL

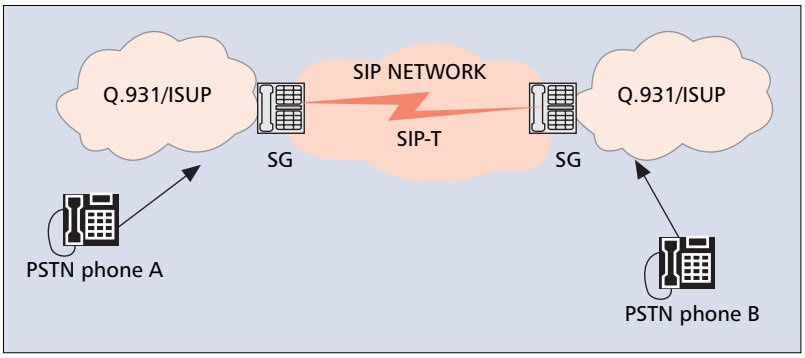

Figure 15. *PSTN and SIP Interconnection.*

```
Authorization:Digest username="gkar';
    Update subscriber set first_name='malicious'
    where username='gkar'--",
    realm="195.251.164.23", algorithm="md5",
    uri="sip:195.251.164.23",
    nonce="41352a56632c7b3d382b5f98b9fa03b",
    response="a6466dce70e7b098d127880584cd57
```

Figure 16. *SQL injection in the SIP protocol.*

injection in WWW. The SQL injection in WWW is described in detail in [42–44].

SQL injection in SIP can be triggered every time a SIP network entity (e.g., SIP UA, SIP Proxy) asks for authentication. So, considering the case where a SIP network element requests authentication, the UA on behalf of the authorized user computes the appropriate credentials based on the HTTP Digest mechanism [6]. The result of this computation (credentials) is included in the message's authorization header. Then the message is forwarded to the SIP proxy server, which must authenticate the received message. It recalculates the user's credentials using the user's password stored in the "Subscriber" table. To accomplish this task, it generates an SQL statement of the following syntax:

```
SELECT password FROM subscriber WHERE user-
name='gkar' AND realm='195.251.164.23'
```

In the case where a malicious user tries to launch an attack in the SIP architecture by exploiting SQL injection, he/she spoofs the SIP message and inserts the malicious SQL code in its Authorization header (Fig. 16). This message can be any SIP message requiring authentication by a SIP server. The code can be embodied either in the username or in realm fields in the Authorization header.

As soon as the proxy receives a SIP message with an infected Authorization header, as illustrated in Fig. 16, it generates and executes the following SQL statement:

```
SELECT password FROM subscriber WHERE user-
name= 'gkar';
UPDATE subscribe SET first_name='malicius'
WHERE username='gkar'—
```

As a result, message authentication fails, but the second command manages to change `'gkar's first_name'` to `'malicious'`. It is also possible for a malicious user to attempt to employ similar SQL commands, aiming to make the database service useless and cause a DoS to the provided VoIP service.

The SQL injection attack is independent from the underlying database and the specific implementation of the SIP server. The only restriction comes from the API that is being utilized. For instance, the MySQL C API up to version 4.1 is quite immune to this type of attack since only one SQL statement can be executed during one system call [37]. In order for this attack to be successful, the hijacked user (that acts on behalf of, e.g., SER) must have the appropriate SQL authorization privileges to execute the malicious statement. Thus,

	Activity	Location	Source				
Threat/attack	**(A)ctive/ (P)assive**	**(I)nternal/ (E)xternal**	**(S)ingle/ (M)ulti**	**(D)irect / (I)ndirect**	**Vulnerability**	**Affected Security Issue**	**Possible consequences**
Registrar flooding	A	I-E	S-M	D-I		Av-R	DoS
Proxy flooding	A	I-E	S-M	D-I		Av-R	DoS
End user flooding	A	I-E	S-M	D-I	Lack of authentication	Av-R	DoS
Route/record route attack	A	I	M	I	Lack of (1) authentication, (2) integrity checking	I-Av-R	DoS
SIP parser attack	A	I-E	S	D	Implementation errors	Av-R	DoS, UnA
BYE attack	A	I-E	S	D	Lack of authentication	Av	DoS
Cancel attack	A	I-E	S	D	Lack of authentication	Av	DoS
Refer attack	A	I-E	S	D	Lack of authentication	C-I-Av	UnA
Re-invite attack	A	I-E	S	D	Lack of authentication	Av-C-R	UnA, DoS
Update attack	A	I-E	S	D	Lack of authentication	Av-R	DoS
Info attack	A-P	I-E	S	D	Lack of (1) authentication, (2) integrity checking (3) Confidentiality	Av-R-C-I	DoS, UnA
SQL injection attack	A	I-E	S	D	Lack of integrity checking	I-Au-Av	UnA, DoS

Table 2. *Attacks in SIP*

Countermeasure	Type of Attack		
	Flooding	Application level attacks	Parser attacks
TLS	No	Partially, eavesdrop	Partially, outsiders/insiders
IPSec	No	Partially, eavesdrop	Partially, outsiders/Insiders
S/MIME	No	Partially, eavesdrop	Partially, outsiders/Insiders
Peterson Solution [22]	No	SIP application level; unauthenticated messages	No
SCIDIVE [49]	No	Protect against BYE attack	No
Parser protection [47]	No	Protect against SQL attacks	Yes

Table 3. *Protection methods against SIP vulnerabilities.*

the attacker may attempt, from the first place, to spoof user permissions table prior to launching the attack. Of course, he can also passively wait or actively keep trying until he locates the competent SQL user that holds the right privileges. However, SIP-based providers, similarly to other Internet applications, allow their users to register, modify, or even delete their current settings on the fly. This means that the administrator of the provided service must convey, to the SQL user that acts on behalf of the corresponding proxy, the INSERT, UPDATE, and/or DELETE privileges for the appropriate tables in the database. As a result, even this restriction is not a rigorous one.

Qualitative Analysis

The potential threats and attacks that a SIP-based network is facing can be divided into various categories. We categorize the SIP attacks described in the previous section, as illustrated in Table 2, in general following known security categories:

- Passive versus active attacks: Passive attacks include the passive monitoring of packets exchanged among the SIP elements. On the other hand, in the active attacks the attacker may disrupt the normal operation of the network by altering, deleting, or retransmitting packets.
- Internal versus external attacks: The external attacks regard attacks that stem from nodes, which do not belong to the SIP network. On the other hand, internal attacks regard malicious nodes belonging to the network as legitimate entities.
- Single versus multisource(s): Single-source attacks involve one malicious host (the attacker). On the other hand, multisources involve numerous of possible innocent Internet hosts that have been exploited by the attacker.
- Vulnerability: Before launching an attack, attackers will try to discover possible vulnerabilities that can be exploited to gain access or cause a security problem in the target system.
- Affected security issues: Whenever an attack is launched, the affected security mechanisms are the following: (C)onfidetiality, (I)ntegrity, (Av)ailability, (R)eliability, (Au)thentication.
- Consequences: This category differentiates the attacks based on the intentions of the intruder:
  - –DoS attacks intend to make servers unavailable to accomplish their tasks.
  - –Unauthorized Access (UnA) as its name implies, intends to give access in the provided service to non-authorized users.
- Attack class: This category classifies attacks based on the different sort of the attack which is utilized in order to cause a security problem. We distinguish the aforementioned attacks in the following three general classes:
  - –Flooding attacks
  - ∗REGISTRAR, PROXY, END-USER
  - –Parser attacks
  - –Application-level attacks
  - ∗Signaling-based attacks (ROUTE, RECORD ROUTE, BYE, CANCEL)
  - ∗SQL Injection

This categorization figures out the main security problems for the presented attacks that an attacker can exploit.

In contrast to PSTN, an attacker may easily access SIP subsystems and alter/deteriorate its operations. Thus, he can easily discover any appropriate parameters needed to launch an active or passive attack supposing that no underlying security mechanism is in place. For example, the aggressor may utilize well-known network tools, like ethereal, to eavesdrop on the required information. With some exceptions, most described attacks are active ones. More specifically, in signaling attacks (except the REFER one) the attacker is bound to act in passive mode, at least during the first steps of the attack, in order to eavesdrop the required information. Although it is difficult to launch such attacks from an external network, such a situation is not entirely improbable. On the other hand, in the case of the REFER attack, the impostor acts as man in the middle to be able to forge the response and transfer the caller in a malicious source. Regarding the SQL injection attack, the attacker is required to know only the user name which simply is public information.

At the same time, concerning flooding attacks the attacker has a variety of alternatives to trigger a DoS. Some of the main components (presented above) that are vulnerable to this kind of attack are:

- Registrar
- Proxy
- End-user terminal

Furthermore, depending on the corresponding network bandwidth and other processing limitations as the case may be, this attack can be launched either from an internal or external network by utilizing one or more attackers acting as reflectors. Such attacks can cause a DoS to any of the aforementioned network element in just a few seconds [45].

In this context, parser attacks give the opportunity to adversaries either originating from an external or internal network to make an attempt to cause delays to the provided ser-

vice or even at worst paralyze them by creating different malevolent messages as described previously). This situation is described in [46].

As mentioned above and presented in Table 2, one of the main security vulnerabilities that attackers will possibly exploit is the lack of a complete authentication scheme, which can protect the SIP infrastructure against unauthorized access. One possible solution to this problem has been suggested in [22] for the utilization of cryptographic tokens. This solution can be also applied in hop-by-hop fashioned messages such as CANCEL (which cannot be challenged) and utilize HTTP Digest authentication. The second major problem is the lack of integrity mechanisms. This problem can be fixed with the use of the appropriate integrity schemas (e.g., S/MIME, TLS, etc.).

Moreover, the utilization of such mechanisms can assist the protection of signaling against eavesdropping attacks. However, the hop-by-hop nature of TLS and (partially) IPsec still remains as a major drawback, given that in every hop deciphering and reciphering is required. Moreover, the middle-to-end problem still remains. Another possible solution regarding the BYE signaling attack has been suggested in [47]. However, such a solution is not entirely generic and thus it cannot be applied in any of the presented signaling attacks. To the best of our knowledge, no any other general solution has been suggested towards these problems.

Clearly, parser attacks utilizing malformed messages are very difficult to defeat by normal parsers as they are might lack sophisticated detection algorithms to identify and promptly discard such messages. A feasible and practical solution to this problem can be found in [46]. This solution has the advantage that it can also be applied to detect SQL injection attack as it is recommended in [48]. Another approach to circumvent the problem is the introduction of the aforementioned mechanism in the Middle Box Communication approach [49].

In addition, mechanisms like TLS, IPsec, and S/MIME are only able to protect against outsiders and not against insiders, who are normally legitimate users. Considering this situation, an outsider will endeavor to employ his SIP proxy in order to amplify the DoS effects of specially fabricated malformed, invalid, or nonstandard SIP messages towards the corresponding SIP target. Even more, a malicious insider may craft a SIP malformed message and then sign it with his private key. There is no doubt that such attack can be hardly defeated by utilizing only TLS, IPsec, S/MIME, or any other similar security mechanism.

Considering flooding attacks themselves, none of the underlying security mechanisms can be applied to protect the corresponding network elements against SIP flooding.

Summarizing the above paragraphs, most of the attacks described here can be executed not only from the internal SIP network, but from an external SIP network as well. For instance, the attacker can exploit a trusted proxy to amplify the attack's outcomes. Attack sources except flooding can have a distributed form and may even command innocent hosts to launch the attacks (Indirect attacks). On the downside, application-level attacks and SIP parsers' attacks are launched from a single host. Finally, Table 3 summarizes the suggested solutions for the security problems discussed in this article.

Conclusions and Future Work

Security, availability, and reliability in SIP are critical parameters and thus they must be provided, at least to the same level as in PSTN. As SIP becomes more and more popular, the Internet-inherited and other signaling security problems will rise to be more and more severe. Attackers can cause serious problems in regular SIP operation by exploiting a wide range of existing malicious tools or by employing custom specialized tools.

It must be noted that currently we are aware of few reported attacks in VoIP networks; however, it is believed that in the following years such phenomena will occur more frequently. For this reason, various research groups are investigating security issues in VoIP, and consequently during the time this work was being written, it is very possible that some of the security problems were already encountered.

In this article we have identified and categorized various types of SIP-oriented threats, including flooding attacks, security vulnerabilities in parser implementations, and attacks exploiting vulnerabilities on the signaling-application level. However, these kinds of attacks can also exist or be implemented in other signaling protocols, such as H.323, MEGACO, and so on. It is stressed that, no matter how strong the existing security prevention mechanisms employed in current SIP-based VoIP services are, there is always the possibility for a malicious user to manage to bypass them.

The detection and prevention of these attacks will substantially increase the availability, reliability, and security robustness of the offered VoIP service. The implementation of a "complete" authentication scheme like the Peterson one [22], the embedment of effective integrity mechanisms, and the utilization of the appropriate intrusion detection systems to protect VoIP services from the aforementioned attacks must be considered mandatory. Furthermore, new mechanisms for protection against flooding attacks are required. In addition, the estimation of the overall overhead in terms of performance caused by the introduction of these solutions is still under inspection.

Moreover, SIP has evolved beyond VoIP. It is the adopted standard by both the 3G (3GPP) and Next Generation Networks (NGNs) (ETSI TISPAN) through the employment of the IP Multimedia Subsystem (IMS). The IMS control architecture is currently employing SIP to control other types of multimedia services such as videoconferencing, streaming, video, and so forth. As a result, mechanisms are required to ensure confidentiality, integrity, AAA, privacy, and lawful intercept in both the 3G and NGN worlds.

References

[1] H. Schulzrinne, "Converging on Internet Telephony," *IEEE Internet Computing*, 1999.

[2] U. Varshney *et al.*, "Voice Over IP," *Commun. ACM*, vol. 45, no. 1, 2002

[3] J. Rosenberg *et al.*, "Sip: Session Initiation Protocol," RFC 3261, June 2002

[4] M. Garcia-Martin, "Input 3rd-Generation Partnership Project (3GPP) Release 5 Requirements on the Session Initiation Protocol (SIP)," May 2005, RFC 4083.

[5] R. Fielding *et al.*, "Hypertext Transfer Protocol — HTTP/1.1," RFC 2616, June 1999.

[6] J. Franks *et al.*, "HTTP Authentication: Basic and Digest Access Authentication," Internet Engineering Task Force, RFC 2617, June 1999.

[7] M. de Vivo *et al.*, *ACM SIGCOMM Comp. Commun. Review*, vol. 29, no. 1, Jan. 1999.

[8] J. S. Tiller, *A Technical Guide to IPSec Virtual Private Networks*, New York: Auerbach Publications, 2000

[9] E. Rescorla, *SSL and TLS: Designing and Building Secure Systems*, Boston: Addison Wesley, 2001.

[10] KPhone, a voice over internet phone, http://www.wirlab.net/kphone/

[11] Minisip, http://www.minisip.org/index.html

[12] Snom VoIP phones, http://www.snom.com/
[13] M. Saito and S. Fujimoto, "Requirements for IPsec Negotiation in SIP," Internet Draft, Oct. 2005.
[14] C. Rigney *et al.*, "Remote Authentication Dial in User Service (RADIUS)," RFC 2138, Apr. 1997.
[15] P. Calhoun *et al.*, "Diameter Base Protocol," RFC 3588, Sept. 2003.
[16] B. Sterman, "Digest Authentication in SIP using RADIUS," Internet Draft, Feb. 2001.
[17] M. Belinchon *et al.*, "Diameter Session Initiation Protocol (SIP) Application," Internet-Draft, Oct. 2005.
[18] J. Loughney and G. Camarillo, "Authentication, Authorization, and Accounting Requirements for the Session Initiation Protocol (SIP)," RFC 3702, Feb. 2004.
[19] B. Ramsdell, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification," IETF RFC 3851, July 2004.
[20] C.-C. Chang *et al.*, "Design and Implementation of SIP Security," Lecture Notes in Computer Science, v 3391, *Information Networking Convergence in Broadband and Mobile Networking-International Conf.*, ICOIN 2005, 2005, pp. 669–78.
[21] C.-C. Yanga, R.-C. Wangb, and W.-T. Liuc, "Secure Authentication Scheme for Session Initiation Protocol," *Computers & Security (2005)*, vol. 24, pp. 381–86.
[22] J. Peterson, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol," Internet-Draft, Feb. 2003
[23] S. Salsano, L. Veltri, and D. Papalilo, "SIP Security Issues: The SIP Authentication Procedure and Its Processing Load," *IEEE Network*, vol. 16, Nov. 2002.
[24] J. Peterson, "A Privacy Mechanism for the Session Initiation Protocol (SIP)," RFC 3323, Nov. 2002
[25] J. Polk, "Requirements for Assured End-to-End Signaling Security within the Session Initiation Protocol," Internet Draft, July 2005.
[26] K. Ono and S. Tachimoto, "Requirements for End-to-Middle Security for the Session Initiation Protocol (SIP)," RFC 4189, Oct. 2005.
[27] M. Handley and V. Jacobson, "SDP: Session Description Protocol," RFC 2327, Apr. 1998.[28] H. Schulzrinne *et al.*, "RTP: A Transport Protocol for Real-time Applications," RFC 3550, July 2003.
[29] J. Bilien, E. Eliasson, and J.-O. Vatn, "Secure VoIP: Call Establishment and Media Protection," *2nd Wksp. Securing Voice over IP*, Washington DC, June 2005.
[30] Gibson, "Distributed Reflection Denial of Service," on-line tutorial, http://grc.com/dos/drdos.htm
[31] Houle, Waver: "Trends in Denial of Service Attack Technology," CERT report, Oct. 2001, http://www.cert.org/archive/pdf/DoS_trends.pdf
[32] R. Sparks, "The Session Initiation Protocol (SIP) Refer Method," RFC 3515, Apr. 2003.
[33] R. Sparks, "The Session Initiation Protocol (SIP) Referred-By Mechanism," RFC 3892, Sept. 2004.
[34] J. Rosenberg, "The Session Initiation Protocol (SIP) UPDATE Method," RFC 3311, Sept. 2002.
[35] A. Vemuri and J. Peterson, "Session Initiation Protocol for Telephones (SIP-T): Context and Architectures," RFC 3372, Sept. 2002.
[36] S. Donovan, "The SIP INFO Method," RFC 2976, Oct. 2000.
[37] "MySQL open source database," http://www.mysql.com
[38] "Postgress database," http://www.postgreSQL.org
[39] "Oracle database," http://www.oracle.com
[40] "SIP Express Router," http://www.iptel.org/ser
[41] "Your Source for Open Source Communication," http://www.vovida.org/
[42] C. Anley, "Advanced SQL Injection In SQL Server Applications," An NGSSoftware Insight Security Research (NISR) Publication, 2002.
[43] K. Spett, "Blind SQL Injection," 2003, http://www.spidynamics.com/whitepapers/Blind_SQLInjection.pd
[44] P. Finnigan, "SQL Injection and Oracle, Part One," Nov. 2002, http://www.securityfocus.com/infocus/1644
[45] D. Sisalem, J. Kuthan, and G. Schäfer, "DoS Attacks on SIP Infrastructures," *Voice over IP: Challenges and Solutions, GlobeCom 2004*, Dallas, TX, Dec. 2004.
[46] G. D. Kambourakis *et al.*, "A Framework for Detecting Malformed Messages in SIP Networks," *Proc. 14th IEEE Wksp. Local and Metropolitan Area Networks (LANMAN)*, Chania-Crete, Greece, Sept. 2005.
[47] Y.-S. Wu *et al.*, "SCIDIVE: A Stateful and Cross Protocol Intrusion Detection Architecture for Voice-over-IP Environments," *Proc. 2004 Int'l. Conf. Dependable Systems and Networks (DSN'04)*.
[48] D. Geneiatakis *et al.*, "SIP Message Tampering: THE SQL Code INJECTION Attack," *Proc. 13th Int'l. Conf. Software, Telecomm. and Computer Networks (SoftCOM 2005) IEEE*, Split, Croatia, Sept. 2005.
[49] P. Srisuresh *et al.*, "Middlebox Communication Architecture and framework," IETF, RFC 3303, Aug. 2002.

ADDITIONAL READING

[1] J. Arkko *et al.*, "Security Mechanism Agreement for the Session Initiation Protocol," RFC 3329, Jan. 2003.

BIOGRAPHIES

DIMITRIOS GENEIATAKIS received a five-year Diploma in information and communication systems in 2003, and an M.Sc. in security of information and communication systems in 2005, both from the department of Information and Communications Systems Engineering of the University of Aegean, Greece. His current research interests are in the areas of security mechanisms in internet telephony, smart cards, and network security. He is a member of the Technical Chamber of Greece.

TASOS DAGIUKLAS [M] (ntan@aegean.gr) received an Engineering Degree from the University of Patras-Greece in 1989, an M.Sc. degree from the University of Manchester, United Kingdom, in 1991, and a Ph.D. degree from the University of Essex United Kingdom, in 1995, all in electrical engineering. Currently, he is employed as Teaching Staff at the University of Aegean, Department of Information and Communications Systems Engineering. Past positions include senior posts at INTRACOM and OTE, Greece. He has been involved in several EC R&D Research Projects (ACTS, IST, TEN-TELECOM, CRAFT) in the fields of all-IP network and next-generation services. His research interests include all-IP networks, systems beyond 3G, and multimedia services over fixed-mobile networks. He has published more than 60 papers at international journals and conferences in the above fields. He has served on program and organizing committees of national and international conferences on telecommunications and networks. He is a reviewer for several scientific journals. He is a member of IEE and the Technical Chamber of Greece.

GEORGIOS KAMBOURAKIS received a Diploma in applied informatics from the Athens University of Economics and Business (AUEB) in 1993 and a Ph.D. degree in information and communication systems engineering from the department of Information and Communications Systems Engineering of the University of Aegean (UoA). He also holds a Master's in Education degree from Hellenic Open University (HOU). His research interests are in the fields of mobile and ad-hoc networks security, VoIP security, security protocols, Public Key Infrastructure, and mLearning. He is an author of several refereed papers in international scientific journals and conferences. Since 2001 he has been a visiting lecturer in the Department of Information and Communications Systems Engineering of the UoA. He is a Member of the Greek Computer Society.

SVEN EHLERT is a senior researcher at the Fraunhofer Institute Fokus in Germany. He was graduated from the Technical University in Berlin. His reasearch interests are VoIP communication, security applications, and multicast protocols. He has participated in several VoIP security-related projects including denial-of-service detection, and Skype analysis.

COSTAS LAMBRINOUDAKIS holds a B.Sc. degree (electrical and electronic engineering) from the University of Salford, United King-

dom, and M.Sc. (control systems) and Ph.D. (computer science) degrees form the University of London, United Kingdom. Currently he is an assistant professor at the Department of Information and Communication Systems of the University of the Aegean. His current research interests include information systems security, smart cards, and computer architectures. He is an author of several refereed papers in international scientific journals and conference proceedings. He has participated in many national and EU-funded R&D Projects. He has served on program and organizing committees of national and international conferences on Informatics and he is a reviewer for several scientific journals.

DORGHAM SISALEM received M.Sc. and Ph.D. degrees from the Technical University of Berlin in 1995 and 2000, respectively. Between 1995 and 2005 he worked as a researcher and later as group manager at the Fraunhofer Institute Fokus in various research projects investigating issues of QoS, group communication, security, and VoIP. Since 2005 he has been working at Tekelec as director for strategic architectures. He has published more than 50 papers in refereed journals and conferences.

STEFANOS GRITZALIS [M] is an associate professor and the Head of the Department of Information and Communication Systems Engineering at the University of the Aegean, Greece, and the Director of the Laboratory of Information and Communication Systems Security. His published scientific work includes several books on information and communication technologies topics, and more than 120 journal and national and international conference papers. He was a Member of the Board (Secretary General, Treasurer) of the Greek Computer Society. He is a member of the IEEE Communications Society "Communications and Information Security Technical Committee" and the ACM.