

Enhancements to SIP to prevent abuse of Voice-over-IP services

T. Neumann and M.S. Olivier

Information and Computer Security Architectures (ICSA) Research Group

Department of Computer Science

University of Pretoria

Email: tozzi@intdev.co.za, molivier@cs.up.ac.za

Abstract: This paper reviews the IETF proposed enhancements to the Session Initialisation Protocol (SIP) to prevent the abuse of its implementation in Voice-over-IP (VoIP) communication services. We look at some of the benefits and drawbacks of these proposed enhancements to ensure trustworthy SIP message exchange. Many proxies assist in establishing of a SIP session. These proxies may modify the traversing SIP messages during transit by injecting routing, logging and call control headers. Due to the distributed nature of VoIP, this could have various implications. We highlight these and proposes a new mechanism of certifying these changes, allowing communicating parties to audit the committed changes. Our aim is to compliment the suggested enhancements to further secure SIP communication.

Keywords: sip message exchange, messaging routing, security and authenticity, identity assertion

I. INTRODUCTION

The Internet has grown since its inception in the 1960s to a large scale recreational and commercial packet-switching network. The adoption of the internet was predominantly through its openness and unrestricted use. It has become the largest network in the world, carrying a multitude of data and information between public and private infrastructure.

The continued use of the internet has spawned new, previously unthought-of services. One such development was the addition of multimedia services to the decentralized communication architecture, allowing for the end-to-end transmission of voice and video media over the digital network. A multitude of vendor-specific products emerged innovating within this realm. The primary uptake of industry was the specific implementation of voice services, accordingly named Voice-over-IP (VoIP). The continuous adoption of VoIP is motivated by the long term cost savings and consolidation of communications infrastructure [1].

In order to address the proprietary vendor implementations, the IETF (1998) collaborated to draft a comprehensive communication protocol. They defined protocols that assist in the establishing of VoIP communication, yet clearly separate the function from the data transfer. They created the SIP protocol, and underlying signalling mechanism. It was designed work in a distributed fashion, and simply coordinates the data exchange

between communicating parties. VoIP is therefore achieved through the implementation of SIP to allow remote parties to establish a call.

The current situation is that SIP has proven to be scalable and efficient. However, it is still an emerging technology and certain shortcomings have been identified. These predominantly surround the facades of security, creating weaknesses in ensuring confidentiality, integrity and auditability of SIP messages. Newer protocols designed for use in other web transactions have considered security through their design. SOAP, for example, similarly to SIP, has an extensible structure, yet already caters for the implementation of signatures and encryption [2].

This paper discusses the design of SIP message exchange. It considers SIP as the underlying technology in enabling the wider adoption of VoIP. We detail the protocol and review the motivation for certain . This paper draws on recently drafted enhancements that proposed a new mechanism to certify a caller. This creates a level of trust amongst communicating parties, as the identity of remote devices can be assured. This important enhancement does however not ensure that the intermediaries assisting in the SIP session are trustworthy. We believe this mechanism should be extended to certify the path chosen when establishing a VoIP call. This paper will discuss our approach to asserting the identity of intermediaries that assist in a SIP session. Thus we aim to establish an acceptable level of trust over the chosen path, through certifying the intermediaries and the changes they make to SIP messages in transit.

In Section II, we review the headers used in SIP messages and their function. Various discussions have arisen about the integrity of these headers and we give background to the proposed enhancements in Section III. Section IV extends these concepts and details our implementation of securing SIP messages. The adjusted state of messages in transit and methods of certifying the participating intermediaries are explained in Section V. We conclude on the gained trust and how our contribution assists in securing SIP in Section VI

II. SESSION INITIALISATION PROTOCOL

A. Messages

The Session Initialisation Protocol (SIP) is a generic session management protocol. It was drafted by the IETF and

is an “application-layer control protocol that can establish, modify, and terminate multimedia sessions (conferences) such as Internet telephony calls” [3]. The protocol itself is not comprehensive communications mechanism, but a signalling standard for the discovery and communication of remote parties. SIP, therefore, does not define the characteristics of a session and relies on a suite of complimentary protocols. The draft was further formalised and established itself as RFC 3261 in 2002.

The design of SIP allows two remote parties to *locate* each other, using URI addressing comparable to email addresses. Through conventional DNS lookups, the caller determines a route to the network of the destined VoIP device. Often, however, devices such as softphones or physical VoIP phones require the assistance of SIP proxies to perform these lookups. The complexity of this architecture and the multitude of proxy servers involved in establishing a call, result in numerous devices participating in a SIP session. This is further complicated in the case of three-way or conference calling [4].

The protocol is based on HTTP-like messages, which when sent to either a proxy or device, invoke at least one response. The first line of a SIP messages always contains a *Method* name, the lines to follow are optional *Headers*. The *Method* name is an instruction which is inspected by the recipient, and processed accordingly. This name, comparable to requests in HTTP, assists in managing the session and either creates, modifies or terminates sessions. The *Headers* provide additional information about the session, passing attributes such as caller name, date and time or routing information. Additionally *Headers* convey signalling details such as sequence numbers, call identifiers and message expiry. Since messages are transported over UDP, a stateless protocol, SIP depends on these attributes to correlate messages to a session.

A caller would initiate a session by specifying a SIP URI e.g. sip:alice@atlanta.com. The calling device does not know of the destination, and thus the domain name portion of the URI is used to locate an authoritative proxy. The URI is passed on to this proxy, and the specified username used to communicate with the intended recipient.

```

INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP bigbox3.site3.atlanta.com
    ;branch=z9hG4bK77ef4c2312983.1;received=196.35.68.101
Via: SIP/2.0/UDP pc33.atlanta.com
    ;branch=z9hG4bK776asdhd;received=196.35.68.99
Record-Route: cdr-svr.atlanta.com
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
User-Agent: <Motorola VT1000 mac: 000F9F466CD0>
Contact: <sip:bob@192.168.1.110>
Contact: <tel:+27837890107>
Max-Forwards: 70
CSeq: 314159 INVITE
Content-Type: application/sdp
Content-Length: 131

```

Fig. 1. SIP message detailing headers

Once contacted, the recipient responds and messages are sent back to the originator indicating a resulting state. SIP responses are “...consistent with, and extend, HTTP/1.1 response codes” [3]. They are prefixed by a three-digit code followed by a descriptive phrase. Response codes allow for both machine and human interpretation, and are defined for all

communication during session initiation. Response messages are sent, along the same path that they were sent, back to the caller. Thus, in most scenarios, the messages will be handled by two or more proxy servers. Communicating over a public network further implies that SIP messages will traverse multiple proxies and service provider networks.

B. Routing

There are two influential factors that distinguish SIP from any other technology implementation to date. Firstly, SIP messages are designed to be handled by many intermediaries. Messages are passed on until they reach their destination, incurring numerous DNS lookups and routing decisions. This leads on to the second factor. The design of SIP contravenes the guidelines set out by the ISO model through incorporating routing and network related details in its *Headers*. It thereby abstracts the *Transport Layer* routing intelligence, giving the *Application Layer* the ability to decide on the next hop. Messages can be passed to any chosen proxy (or device) for handling. The implications of *Application Layer* routing are discussed in [5], [6]. The agility of SIP is therefore also its weakness.

The involvement of numerous unknown parties brings with it various implications of privacy and trust. The SIP messages pass between proxies outside of the control of the caller [7]. This arises from SIP having the ability to route messages independent of the governing networking topology. The resulting path which SIP messages take is initially undetermined. Unknown to the caller, the messages could be sent over unreliable or untrusted networks. The destined device responds to the caller, and the messages are directed back over the path dynamically created during session initiation.

The application layer routing logic could choose these paths based on various criteria. Very feasible are rules arising from least cost routing agreements, fault tolerant routes and government or corporation interconnect policies. This criteria influences the created path, yet other scenarios should be considered. Rules could be injected by compromised proxies or manipulated by malicious devices. The consequence of the aforementioned is that SIP messages would be directed through proxies and networks not previously anticipated. This naturally gives rise to certain privacy concerns [8]. Details such as source and destination of call can be logged, and various attributes about the session recorded. Calls, and potentially individuals, can be monitored by intermediaries.

In contrast to the proxies, it can be presumed that the user might want to modify specific *Via* or *Route* instructions. If a user consciously wants to avoid being monitored, the caller or recipient could manipulate the additional routing headers. Thus, a user could remove these headers to by-pass proxies or media controllers. This results in that a user could avoid billing or logging systems. Since the session would then no longer be communicated to all participants, no accurate records about the events could be recorded.

The RFC outlines the many aspects of negotiating and establishing a communications session. Our discussions will

focus on three headers used in establishing a call. These headers are: *Via*, *Route* and *Record-Route*, and have been identified as those vulnerable to abuse. The following section will review efforts of the IETF to secure these headers, and create a level of trust between communicating parties.

III. PROPOSED ENHANCEMENT

In order to ensure that the SIP messages remain unchanged, a new method to securing specific message headers [9] has been proposed. The motivation for securing the headers is to ensure that the content of the SIP message itself has not been modified in transit. This section will explain the proposed enhancements, which leads us onto our research of certifying the intermediaries involved in a SIP session.

A device initiating a call creates a SIP message and passes it to a proxy server. Depending on the destination, the proxy will in turn pass the message on. However, of concern is the integrity of the message once it has been forwarded. The next-hop proxy determined through resolution and routing mechanisms, can not be assured that the content has not been changed. Thus, it and any consecutive proxy (or devices) can not ascertain whether any of the presented attributes are true. This raises concern about the possible manipulation of SIP messages resulting in deception, abuse or denial of service. The implication is that plausible scenarios such as the construction of deceitful messages would arise. This would not only interfere with a caller's credibility, but also raise doubt as to the trustworthiness of the originating network. The problem is that this could realise into the abuse of VoIP services comparable to the previously unexpected creation of SPAM in email [10]. Industry has hypothesized about SPAM in VoIP and accordingly termed this abuse of VoIP services as SPIT [11]. This issue would have to be addressed before the technology is too wide spread to be secured.

Two new headers, namely *Identity* and *Identity-Info* have been proposed in [9]. These enhancements to SIP do not require changes to any existing implementations, but allow for proxies (or devices) honouring these extensions, to establish a common level of trust. It is assumed that a device is authenticated to an authoritative proxy server. Thus, the proxy server represents the device within the domain it is communicating in. The device will communicate through this proxy, leveraging of the routing rules and defined policies.

The *Identity* header is a digest string computed from specific values within a SIP message. This string is a hash over important attributes, and signed using a X.509 certificate. The digest string is written into the SIP message before being sent on.

$$Identity = sha1WithRSAEncryption(attributes)$$

The calling device will provide the necessary attributes to initiate a session. It therefore presents the source SIP URI, the destination SIP URI, the *Sequence Number*, specified *Caller ID* and the *Method* headers. These details are sufficient for the proxy to process the SIP message, and attempt to establish a session.

$$attributes = source : destination : callid : method : \\ timestamp : contact : message body$$

The proxy must be configured to hold a certificate valid for its authoritative domain, and use this in signing outbound messages. The certificate must be valid for the represented domain, and match the host name of the proxy handling the message. The hash and signature is computed using the sha1WithRSAEncryption algorithm over the canonical string.

A remote party might not know of the caller, and would want to verify the presented identity and integrity of the message. In order compute a comparable hash, the remote party must retrieve the authoritative proxy's public keys. The proposed enhancements draw upon the *Identity-Info* header to assist in describing the location (URI) of the certificate.

This new mechanism of certifying a caller's identity allows for the true source of this message to be determined, and the integrity of the presented attributes verified. It ensures that all headers remain in tact without hindering the end devices in communicating. It further shifts responsibility away from end devices, and makes it impractical to modify these headers once a digest has been computed.

There are many benefits of confirming a user's identity. Systems can trust the caller, and can be built to securely allow access confidential or private information. The caller no longer has to audibly confirm his identity, nor confirm secrets using DTMF inputs. Misuse of telephony services would be reduced as users are aware that their session identity is managed by an intermediary. Although this has bearing on a user's privacy when communicating, he is assured that his details are not maliciously intercepted. It must be noted though, that this mechanism only focuses on the caller.

In this section, we described the enhancements to SIP in order to certify a user's identity. In the section to follow, we apply these principles and extends these to intermediaries. The aim is to provide a way of certifying the intermediaries handling SIP messages in transit.

IV. IDENTIFY ASSERTION

We detail how we propose certifying changes made to a SIP message in this section. The attributes deemed necessary to ensure integrity are described, and the implementation of our proposed mechanisms explained. It requires that proxies correctly sign the traversing messages, and simultaneously underwrite the changes made during transit.

The obvious concern with SIP messages is that they dynamically establish a route to the destination, thus directed over a previously undetermined path. The routing headers are inserted into the message, as shown in Fig 1. Since the route cannot be predicted, we propose each proxy identify itself enroute. This is achieved by appending a digital signature to the SIP message before it is forwarded. Thus, the *Application Layer* routing headers can not be modified when handled by the next proxy.

A intermediary handling the SIP messages is required to insert a *Via* header in order for the message exchange to

continue. SIP prescribes that the version, protocol and host name of the proxy are recorded. Our mechanism allows us to ascertain whether a proxy claiming to have handled the message, is indeed the true intermediary. This is achieved through the addition of a digest string linked to this intermediary.

The proxy must confirm that it adjusted the request, inserting its particulars in a *Via* header. Policy might have the proxy inserting logging or routing headers, and these must additionally be certified. Proxies process these headers top-to-bottom — thus any new details are prepended to the request. If each proxy signs its inserted *Via*, a trace of modified *Via*, *Route* and *Record-Route* headers can be produced.

Considering that the original headers produced by the caller and particulars about the session have been signed into the *Identity* header, it is not required for the hashing to consider all SIP attributes. This is a fair performance improvement since only the *Identity* header must be included, when signing the concatenation of newly inserted headers. The signature would similarly be formed through a digest computation.

$$Signature = sha1WithRSAEncryption(attributes)$$

attributes = *adjusted Method : inserted Via :*
inserted Routes : inserted RecordRoutes : Identities

The *Via* header must include this signature, as well as the path to its public X.509 certificate. The header structure of SIP messages allows for this through semi-colon separation of attributes, and thereby inherently caters for these extensions.

```
Via: SIP/2.0/UDP cdr-svr.atlanta.com
;branch=z9hG4bK776asdhs;received=192.0.2.1
;identity=A5ohltSWpbmXYjDhaCiHjT2xR2PAwBroi5Y8tdJ+CL3
z1Y72N3Y+lP8eoiXlYzOuwB0DicF9GGxA5vw2mCTUxc0XGOKJOh
pBnzoXnuPNAZdc2EWsVOQAKj/ERsYR9BfxNPazWmJZjGmDoFDdBu
NamJRj1EP0Kni3uAZ1cuf9zM+
;identity-info=https://sip.atlanta.com/cert
```

Implementing this mechanism at every proxy along the route creates layers of identities. Each proxy performs this computation and asserts its changes together with those made before it. This layering produces a chain of auditable modifications. By verifying a digest, it can now determined where in the chain certain routing instructions were modified. This allows for the interrogation of changes made to the dynamic route, as well as modifications to instructions already inserted by hosts having previously handled the SIP message. It further ensures that SIP messages cannot be redirect, nor intermediaries be removed from the path already traversed. This is visually illustrated in Fig 2.

In light of the presented, we will lead onto a discussion on integrity of the headers. We depict the state of a SIP message while in transit, accumulating layers of identities. This state model is used to describe the criteria under which the intermediaries are considered as verified. It allows for a chosen path be regarded as acceptable and trustworthy.

V. ASSERTING CHANGES

To ascertain whether or not a chosen route is acceptable, the newly placed identities can be checked. Although the actual implementation is fairly straightforward, we note that only a

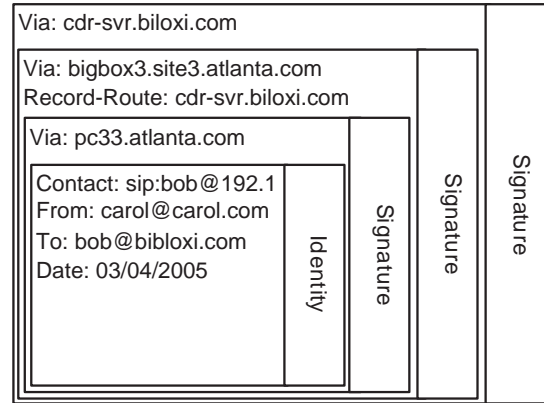


Fig. 2. Layers of identities

policy based decision-making can accurately judge the validity of any particular path.

Ideally, the SIP message would arrive entirely certified. This requires all proxies to sign for handling the message. This might be a prerequisite when communicating with partners or affiliates with whom confidential information is exchanged. An organisation might wish to choose service providers who guarantee their routes and third-party interconnections. The parties relying on such secure communication would be assured that the caller and remote party are indeed who they claim to be, and managed from within their organisation. It further confirms that they have been identified, and that their session is not handled by any unknown intermediary. Fig 3 depicts the state of the SIP message during transit, and the resulting integrity based on the headers being entirely or partial certified.

Callers communicating over the public internet, or utilising services from a VoIP provider (such as Vonage) could be partially certified. Existing security issues surrounding access to Public Switched Telephone Network (PSTN) and predefined access lists have been discussed in related work [12], [13]. PSTN break-in or break-out gateway and authoritative proxies should certify their identity. This confirms their involvement for accepting a call, while subsequent proxies might not certify their changes.

Should a SIP message not contain a signature over changes made to the headers by a proxy, certain deductions can be made. Not all imply misuse, and would be dependant on the wider acceptance of this mechanism. However, when receiving a unsigned message, it could be assumed that either:

- 1) The proxy was not configured to sign traversing messages yet is operating correctly
- 2) Unknown intermediaries assisted in routing these messages for the messages to reach their destination
- 3) Untrusted proxies were inserted to handle messages. This raises suspicion about the chosen intermediaries

The latter two possibilities could arise under normal operation, however, are more to likely to exist in messages forwarded by misconfigured or compromised proxies. Thus, the

header will reveal the routing messages to through unknown and untrusted intermediaries.

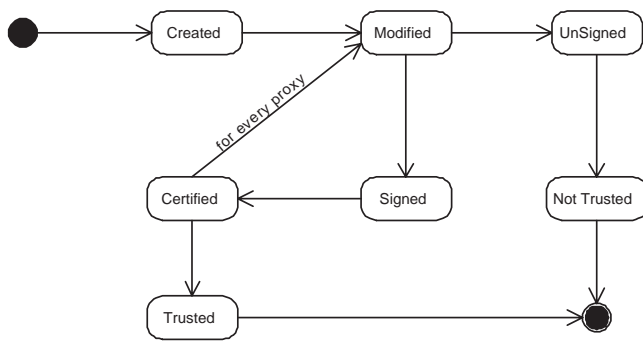


Fig. 3. States of certified SIP headers.

Policy can be defined to highlight these situations, and limit untrusted communication. Such policies would be defined by an organisation, and implemented on the authoritative proxy server. It would thus be possible to avoid untrusted requests from being accepted. A policy requiring certification by intermediaries ensures that users cannot by-pass the billing and call control systems fronting the organisation. The design of such a policy is outside the scope of this paper.

This paper will conclude on our presented mechanism in Section VI. We summarise the discussed application of identity assertion can compliment the proposed enhancements to further securing SIP.

VI. CONCLUSION

The aim of the paper was to show that many parties assist in establishing a SIP session. The technology supports VoIP communication as it facilitates the signalling between remote parties. It was shown that the information exchanged is potentially at risk since the messages could be handled by untrusted intermediaries.

We acknowledge that these intermediaries are required, yet raise concerns about the possible implications of call related data being communicated to untrusted intermediaries. Changes made to the routing headers by either a misconfigured proxy or malicious device, would be able to redirect SIP messages. Third parties and previously unexpected proxies would participate in a session without the user's knowledge.

This paper has shown how recent paradigms of certifying a user's identity can be extended to intermediaries. Participating proxies would underwrite their host names and certify their changes. These could be audited, and modifications attributed to a responsible intermediary.

Future research will evaluate the required policies to govern of such security enhancements. These will consider the emerging enhancements, and motivate through the learnings of previous technologies, how to effectively avoid the abuse of VoIP services. This will predominantly draw upon the mechanisms proposed in security similar distributed communication protocols e.g. SMTP and SOAP.

In conclusion, we believe that security plays an important role in the adoption of VoIP. The underlying technologies must be adequately secured to ensure trustworthy communication. Our research will thus continue to address the facades of security in SIP to achieve this.

REFERENCES

- [1] L. Greenstein, "Transporting voice traffic over packet networks," *Int. J. Netw. Manag.*, vol. 8, no. 4, pp. 227–234, 1998.
- [2] M. Gudgin, "Secure, reliable, transacted: innovation in Web Services architecture," in *SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of Data*. New York, NY, USA: ACM Press, 2004, pp. 879–880.
- [3] E. S. M. Handley, H. Schulzrinne and J. Rosenberg, *SIP: Session Initiation Protocol, RFC 3261*, IETF, March 1999.
- [4] R. Sparks, *SIP call control*, IETF, July 2000.
- [5] H. Schulzrinne and E. Wedlund, "Application-layer mobility using sip," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 4, no. 3, pp. 47–57, 2000.
- [6] J. C. Atanu Ghosh, Michael Fry, "An Architecture for Application Layer Routing," *Lecture Notes in Computer Science*, vol. 1942, p. 71, 2000.
- [7] J. Peterson, *A Privacy Mechanism for the Session Initiation Protocol (SIP)*, RFC 3323, IETF, November 2002.
- [8] Douglas C. Sicker and Tom Lookabaugh, "VoIP Security: Not an Afterthought," *Queue*, vol. 2, no. 6, pp. 56–64, 2004.
- [9] C. J. J. Peterson, *Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)*, draft-ietf-sip-identity-03, IETF, February 2005.
- [10] L. F. Cranor and B. A. LaMacchia, "Spam!" *Commun. ACM*, vol. 41, no. 8, pp. 74–83, 1998.
- [11] V. G. Cerf, "Spam, spim, and spit," *Commun. ACM*, vol. 48, no. 4, pp. 39–43, 2005.
- [12] J. Aweya, "Trunking of TDM and narrowband services over IP Networks," *Int. J. Netw. Manag.*, vol. 13, no. 1, pp. 33–60, 2003.
- [13] W. Jiang, J. Lennox, H. Schulzrinne, and K. Singh, "Towards junking the PBX: deploying IP telephony," in *NOSSDAV '01: Proceedings of the 11th international workshop on Network and operating systems support for digital audio and video*. New York, NY, USA: ACM Press, 2001, pp. 177–185.

