Session Initiation Protocol and Security

Michael Barwise and Peter John Barnes

INTRODUCTION

SIP is still very much in the development phase. If you are tracking its progress you will be aware of the rate at which proposals and draft RFCs are being produced. In July of this year alone some 30 new proposal documents were published on the SIP Forum, and particularly in recent months, several draft proposals have been published addressing the need for security in SIP transactions. Given this volume of information, the detailed aspects of SIP and its security have been well aired. This article is therefore not intended to be a detailed critique of SIP security, but rather to comment on the outline framework that has emerged, and to give consideration to implementational and operational risks.

Because SIP is used only in a brief preamble to a larger transaction that can be independently secured, security measures associated with SIP could be considered to represent less of a burden than those of the main session protocol do. Abuse of the protocol would at first sight seem to be restricted to denial of the SIP service itself or to session stealing. However, the business consequences of such security breaches are too often overlooked in the communications void between developers, implementers and deployers. Potential business hazards in the context of SIP include loss of revenue due to illicit access to chargeable content, damage to credibility resulting from malicious redirection of requests or denial of service, network intrusion facilitated by information gathering about network configuration or implementation-related platform compromise.

EXPECTATIONS OF SECURITY

General requirements for Internet security include proof of identity (authentication), access permissions (authorisation), eavesdrop prevention (confidentiality), protection of information from modification (integrity), proof of involvement (non-repudiation), and robust connectivity between parties to a transaction (availability). From its inception SIP was designed to include security provisions, but the emphasis has always been on authentication, authorisation and confidentiality. Availability and non-repudiation are not at present directly supported, although they could be of major significance in the SIP-initiated provision of chargeable services. However, SIP security is extensible by implementation of additional security provisions within lower-layer protocols. For example, application-layer system logs of fully authenticated and authorised sessions could provide a basis for non-repudiation, although this would be implementation-dependent. Finally, it is important to note that all the security measures discussed below relate only to the SIP transaction. They do not provide security to the SIP-initiated session, which must be independently secured.

SIP HEADERS

A header in the SIP context is not a bit field as in, for example, IP, but a string or logical line of ASCII characters as in the e-mail protocols. Each header consists of a field name and field body, followed any expected parameters. Headers common to a number of SIP security headers include the “nonce” (a unique server-specified data string generated for each authentication or authorisation message exchange) and the “realm”, which is a reference to the authorisation security policy used for a particular security association between two devices in a SIP transaction.

AUTHENTICATION

Authentication of system entities is advisable wherever a SIP Client-Server relationship exists, and it can be accomplished in various ways with differing levels of robustness. The simplest and least robust of these are provided by support for the HTTP “basic” and “digest” schemes, neither of which are proof against forgery. Alternatively, certain headers not required by intermediate proxies may be encrypted to provide a secure digital signature. The order in which headers are included in the message is very important in such a case, as the signature is generated from a contiguous block of data which must entirely follow any fields which are required to be left in clear.

Authentication may be end-to-end or hop-by-hop, and in the latter case transport- or network-layer authentication may be used. The main benefit of end-to-end authentication is generally seen as simplicity of implementation, but it necessarily leaves a considerable amount of information in clear. Hop-by-hop authentication requires a much more complex underlying security infrastructure, but permits total confidentiality of the SIP message. It might be argued, however, that end-to-end authentication is possibly more secure in the real world, as there are only two nodes open to attack.

Hop-by-hop encryption necessitates the whole message being transiently in clear as it passes through each proxy, and thus overall security depends entirely on the robustness of the weakest proxy against attack. In any given implementation, this potential hazard should be compared with the significance to a potential attacker of the information left in clear by end-to-end authentication. A considered trade-off must be made between the risks associated with these various factors.

AUTHORISATION

SIP authorisation provides a means by which a call can be permitted for any given service policy imposed by a proxy or correspondent. Typically this is used when a call and associated services to a correspondent or between intermediate nodes are accountable, as in the case of call charging.

SIP defines several headers used in authorisation. The "Authorisation" header contains a signature computed across components of the SIP message which do not change in transit between proxies: the nonce, the realm, the request method (the type of request message dispatched by a user agent client), the request method version, and the authorisation type. The default authorisation type is PGP, which is used here as a source of cryptographically strong signatures rather than merely as a means of message encryption. There is also a "Proxy-Authorisation" header, which is used by a SIP user-agent to identify itself to a proxy. This contains the type of authorisation, credentials of the user-agent and/or realm of the resource being requested.

Where a single Request URI from a user-agent is destined for multiple recipients, the message is forked: the SIP proxy receives a single message from the originating user-agent and replicates it to each correspondent destination, having labelled each one with a unique branch ID. Assuming the recipients are contactable, they each respond to the proxy with a SIP message containing a challenge. However, this causes an authorisation problem. The proxy forwards on to the user-agent only the first challenge response it receives. Subsequent challenges arriving at the proxy from the other destinations are ignored. Thus, only the first destination to respond is authorised to participate in the SIP-initiated session and the calls to the other destinations are dropped. It is clear that, due to this issue, forking and authorisation are currently incompatible.

CONFIDENTIALITY

Confidentiality is accomplished within SIP (as elsewhere) by use of encryption. However, there are a few restrictions that preclude total confidentiality. SIP security may operate on a hop-by-hop basis or end-to-end, but critical headers in the SIP message, such as the TO and FROM fields, which are interpreted by intermediate devices (proxies), cannot be encrypted end-to-end. For this reason, lower layer security implementations (such as VPNs) would need to be considered for global end-to-end security across an Internet. SIP messages pass through intermediate stations, which by default would not be able to interpret fully encrypted transactions at the application layer. Smart (VPN aware) SIP proxies would need to be part of a network infrastructure conveying end-to-end encrypted SIP messages. Nevertheless, SIP does support a variety of digest and public key encryption mechanisms for the encryption of information not required by proxies to forward traffic. The default method is once again PGP, which provides an excellent compromise between performance and robustness.

IMPLEMENTATIONS

Although SIP offers the capabilities for a good level of security, given its purpose and operating regime, the actual level of security deployed in a given implementation will be very much a matter of choice. However, given the potential ubiquity of the SIP protocol in the future, special care should be taken in general to minimise the exploitability of implementations. Quite apart from the possibility of session stealing and denial of service on SIP itself, there is the very real threat of arbitrary attacks on implementations of the protocol. About half of all successful security breaches are the result of exploitable faults in code. The vast majority of these exploits bear no relation to the intended function of the protocol or code, but are aimed at compromising the platform on which it runs. Robust implementation is particularly important in the context of SIP because, as a call signalling protocol, it does not provide many call services. Call services will be facilitated by extensions to the protocol or by procedures written in languages such as CPL, XML or JAIN APIs. Not only will these extensions themselves require to be robustly coded and tested, but also the language interpreters and run-time environments within which they operate will need to be proof against exploitation, to avoid SIP and its extensions becoming a target of choice for the cracker community.

Originally appeared in the SIP Informer, Autumn 2001