SIP J. Rosenberg
Internet-Draft Cisco Systems
Expires: January 12, 2006 C. Jennings
Cisco
J. Peterson
Neustar
July 11, 2005

Identity Privacy in the Session Initiation Protocol (SIP)
draft-rosenberg-sip-identity-privacy-00

Abstract

RFC3323 defines procedures for privacy in the Session Initiation Protocol (SIP). These mechanisms make use of a privacy service that resides in the network, which can remove identifying information from messages. Its approach to privacy was compatible with the identity mechanisms in RFC 3325, which defined the P-Asserted-ID header field.

However, its approach does not work well with the new cryptographic-based mechanisms in draft-ietf-sip-identity. As such, this document proposes a new framework for user privacy in SIP.

Table of Contents

1. Introduction

RFC 3323 [2] defines procedures for privacy in the Session Initiation Protocol (SIP). It provides guidelines for a UA to follow in the construction of its messages, so that identifying information is not placed into the message in the first place. However, it also defines a network-based privacy service that can be invoked by the client through the insertion of the Privacy header field. This privacy service typically runs within the user's default outbound proxy, and is responsible for removal of additional information from the messages. Two levels of privacy can be provided by this service - "header" privacy, which obfuscates identifying information from the SIP messages, and "session" level privacy, which includes the IP addresses used for exchange of media.

RFC 3325 [9], which defined the P-Asserted-ID header field, has seen widespread usage as the means for network authenticated identity in SIP. It defines another privacy service, the "id" service. This service causes elements in the network to strip the P-Asserted-ID header field when a request traverses a trust boundary.

RFC3325's form of identity has numerous drawbacks. Of these, the most significant is that the trustworthiness of the asserted identity is equal to the trustworthiness of the least trustworthy provider within the network of providers that constitute the trust domain. This works well in single provider environments, but in larger scale interconnects it eventually breaks apart. Unfortunately, the trustworthiness of an identity is a key property needed for nearly all of the VoIP anti-spam techniques [13]. For this reason, amongst others, [3] was developed. It provides strong cryptographic assurances of identity. It does so by providing a signature over the From header field in the request, and including in that signature information that provides referential integrity of the signature. This allows for recipients of the request to validate that the asserting domain has truly asserted the requestor's identity for that request. Since the mechanism is fundamentally domain-based, it also allows validating entites to apply policies regarding the trustworthiness of the asserting provider. This fundamentally avoids the "weakest link" property of RFC 3325.

There are numerous issues in the direct applicability of RFC 3323 to draft-ietf-sip-identity, many of which are pointed out in Section 13 of [3] (herein referred to as the "SIP identity specification" or the "SIP identity mechanism"). These problems are:

Intra-Domain Privacy: Because the SIP identity mechanism relies on the domain of the From header field as a key for obtaining certificates used to validate the identity in the From header field, anonymity is restricted to being within a domain. It is no longer possible, as described in RFC 3323, to populate the From header field with "anonymous.invalid" as the domain. As a consequence, a recipient of the request will be able to determine the domain of the originator of the request, though they will not be able to determine which user within that domain sent the request. This limitation is not very troubling for domains with extremely large numbers of users. However, for small domains, such as enterprises or home networks, it can be equally revealing as the identity of the requestor themselves.

Contact Privacy lost: Because the SIP identity mechanism relies on a signature over the Contact header field for referential integrity, a privacy service that provides header privacy cannot actually modify the Contact value. This will reveal the IP address of the requestor to the recipient of the request, which can often provide substantial information about the requestor.

Session Privacy lost: Session privacy is accomplished through a back-to-back user agent (b2bua) that rewrites the SDP to relay session media through an intermediary. This no longer works at all with the SIP identity mechanism, as it relies on a signature over the body of the request (which contains the SDP) to provide referential integrity.

Subscriber Identity Lost within Originating Domain: One of the benefits of the P-Asserted-ID header field when used in conjunction with the "id" level of privacy is that elements within the domain of the originator of the request will still be able to determine the identity of the originator. This is necessary for providing features for the requestor, accounting for their usage, and so on. With the SIP identity mechanism, if privacy is needed, the From header field contains an anonymous URI. As a result, the request has no information that can identity the user within their own domain, unless the SIP identity mechanism is used in conjunction with RFC3325, which is redundant.

These problems are in addition to the problems inherent in RFC 3323 to begin with:

Sensitivity to Boundary Configuration: Although RFC3323 argues strongly in favor of placing the privacy service very near the originator of the request, this goal is at odds with RFC 3325, which requires the privacy service to be on the egress edge of the trust domain. As a result, privacy is actually provided only if

every egress proxy is properly configured to take positive action and remove the P-Asserted-ID header field. Because positive action from the network is required to provide privacy, this mechanism is sensitive to misconfiguration of network elements, particularly in large interconnected trust domains.

Complicated Call Trace: In many networks, there is a requirement to provide a call trace feature that allows for malicious callers to be traced back to their source so that legal action can be taken. The utility of such features in a global SIP network aside, RFC 3323 makes such a feature difficult to provide since the identity of the requestor is literally removed from the request. This complicates the tracking procedures needed to identify the originator later on.

Limited Flexibility The degrees of privacy that RFC 3323 could provide were coded into the tokens valid in the Privacy header field. More complicated combinations - anonymity for certain media streams but not others, for example - were not possible.

This specification provides an alternate formulation for user privacy that works well in conjunction with [3]. This mechanism resolves nearly all of the limitations described above by moving more intelligence to the client, and having it act in cooperation with network services that provide atomic anonymity functions - IP address privacy via Traversal Using Relay NAT (TURN) [5] and URI privacy via an anonymous URI minting process.

2. Overview of Operation

When a user wishes to make an anonymous request, the user agent determines the set of identifying information that is to be obfuscated. This identifying information includes IP addresses, such as those in the Session Description Protocol (SDP) [6] and Via header fields, and URIs, such as those in the From header field and Contact header field of the request. User agents can anonymize any subset of this information in the request.

To anonymize IP addresses, the client contacts a TURN server [5], and obtains an IP address and port on the server which route to it. Ideally, this is done with a TURN server that is specifically dedicated to anonymous services, and thus can provide a higher degree of anonymity by obtaining anonymized IP address from a separate provider (see Section 6) than a normal one. The client uses the TURN-derived addresses in those fields of the message where the UA wishes to anonymize an IP address.

To anonymize URIs, and in particular the URI in the From header

field, the client needs to obtain a URI from its domain that possesses both the AOR property and the anonymity property (see [4] for a discussion of URI properties). To do that, it generates a special REGISTER request that effectively asks the provider to create a new URI for the user, and at the same time, register it. The network will construct this URI such that other network elements within the domain can use it to identify the requestor, but those outside cannot. This is readily done by creating the URI by encrypting the actual identity of the requestor combined with a large random number. Any element that shares the decryption key can know the identity of the user, but others cannot. In addition, the URI will have the "user" URI parameter present, and set to the value of "anonymous". This signals to all elements that the requestor is asking for anonymity. This is needed to prevent downstream elements within the domain from inserting additional identifying information, and also for properly rendering the fact that the caller was anonmyous.

The UAC then places this URI in the From header field of the request. It populates the Contact header field value with a Globally Routable User Agent URI (GRUU) [4] that was obtained through the registration which yielded the minted From URI. Beyond that, the other procedures of RFC 3323 around display names, Call-ID and other fields are followed.

This request is then sent into the network. There is no Privacy header field or other network involvement needed in order to further anonymize the request. Within the domain of the originator, proxy servers that see that the From header field contains an anonymous URI can decrypt it to obtain the identity of the requestor. Of course, elements outside of the domain will not possess the key, and therefore will not know the identity of the requestor. Because positive action is required in the network to obtain their identity (namely, acquisition of the decryption key and decryption of the URI), the mechanism is privacy-safe. Network misconfiguration can, in the worst case, result in a proxy not determining the identity of the requestor.

Furthermore, since the From field URI is carried all the way to the recipient of the request, it is possible to "call them back", even though the request was anonymous. Of course, the originating domain can decide to reject such requests, but this becomes a matter of local policy. The fact that the identity of the requestor, suitably encrypted, is carried all the way to the recipient of the request also facilitates services like malicious call trace. A network provider can contact the domain administrator of the domain on the right hand side of the at-sign, and request decryption of the user part in order to identify the malicious caller. Since these requests

are handled off-line and not in real time, they can be suitably authorized.

3. UAC Behavior

3.1 Determining the Level of Anonymity

When a user wishes to send a request, whether it is an INVITE to initiate a session, or a SUBSCRIBE [10], MESSAGE [11] or any other method, the UA makes a determination about the level of anonymity that is desired. Typically, this would be based on user input or on local configuration or policy. The precise means for making this determination is outside of the scope of this specification.

Ultimately, however, the level of anonymity is expressed as a function of which types of identifying information (IP address, hostname, URI or display name) are to be anonymized, and in which fields of the SIP message. The following fields typically contain identifying information about the user:

From: This field contains the identity of the requestor, and will be signed by an identity service within the domain of the requestor. As such, clients desiring anonymity SHOULD populate this with a URI obtained through the procedures of Section 3.2. The display name also contains identifying information. It is RECOMMENDED that this be omitted when the requestor requires anonymity. This is a change from RFC 3323, which recommended a value of "Anonymous". Rather than relying on a display name to indicate an anonymous call, which is language-specific and not meant for consumption by an automata, the "user" URI parameter of the From header field indicates that the request was anonymous.

Contact: This field contains a URI used to reach the UA for mid-dialog requests and possibly out-of-band requests, such as REFER [12]. It is RECOMMENDED that this field be populated with the GRUU obtained through the minting procedures of Section 3.2. The display name also contains identifying information. It is RECOMMENDED that this be omitted when the requestor requires anonymity.

Reply-To: This field contains a URI that can be used to reach the user on subsequent call-backs. Clients desiring anonymity SHOULD populate this with a URI obtained through the procedures of Section 3.2. The display name also contains identifying information. It is RECOMMENDED that this be omitted when the requestor requires anonymity.

Via: This field contains an IP address and port that is used to reach the user agent for responses. It is RECOMMENDED that this field be populated with an IP address and port learned through a TURN server Section 3.3.

Call-Info: This field contains additional information about the requestor. It is RECOMMENDED that this field be omitted from requests.

Call-Info: This field contains additional information about the requestor's user agent. It is RECOMMENDED that this field be omitted from requests.

Organization: This field contains additional information about the requestor. It is RECOMMENDED that this field be omitted from requests.

Subject: This field contains freeform text about the subject of the call. Since it is not possible to know what content a user has inadvertently placed into such a header field, it is RECOMMENDED that this field be omitted from requests.

Call-ID: User agents SHOULD substitute for the IP address or hostname that is frequently appended to the Call-ID value a suitably long random value (the value used as the 'tag' for the From header of the request might even be reused).

SDP c/m lines: The c and m lines in the SDP body convey an IP address and port for receiving media. It is RECOMMENDED that this field be populated with an IP address and port learned through a TURN server Section 3.3.

SDP o line: The username SHOULD be set to "-". The IP address in this field SHOULD be populated with an IP address and port learned through a TURN server Section 3.3.

SDP s line: The session name SHOULD be set to "-".

SDP i,u,e,p lines: These lines SHOULD be omitted from the SDP.

3.2 Minting an Anonymous AOR

A key aspect of this specification is the ability of a UA to obtain an anonymous URI for placement into the From and Reply-To header fields, along with a GRUU that can be placed into the Contact header field. It is RECOMMENDED that the UA obtain a new anonymous URI for each new request outside of an existing dialog that it generates.

To obtain a new URI that is suitable for placement into the From header field of a new request, a UA constructs a query REGISTER request according to the procedures of RFC 3261. This request is not anonymous; a UA MUST correctly populate the To, From and other header fields of the request. This request MUST utilize the GRUU mechanism, and thus include the Supported header field with the value "gruu" [4]. The Contact header fields, however, are omitted as this is a query registration. However, the UA MUST include the Require header field with the option tag "anonymous". This instructs the registrar to view this request as a special query; one that provides the UA with a brand new set of anonyous URIs that represent aliases for the user's AOR and registered contacts.

The REGISTER response will contain the set of currently registered Contacts against the AOR in the To header field. In addition, the response will contain the Anonymous-To header field. This header field will contain a URI that has both the AOR and anonymous properties, and which represents an alias of sorts for the user's actual AOR. Its not a pure alias, in that requests sent to that URI don't get equivalent treatment to requests sent to the AOR. Domain policy may result in different treatment for requests made to that URI. This specification provides no automated means for the user to request specific policies. The URI from the Anonymous-To header field can be placed into the From and Reply-To header fields of an outgoing request. Note that each and every REGISTER transaction sent by the client with the "anonymous" option tag in the Require header field will mint a new anonymous URI in the Anonymous-To header field.

In addition, because the client had indicated support for the GRUU mechanism, the REGISTER response will also contain a GRUU for each registered contact. However, these GRUU will also be freshly minted, and have the anonymous property as well as the GRUU property. Like Anonymous-To, each REGISTER transaction produces a new set of GRUU in the Contact header field of the REGISTER response. The client then uses the GRUU for its own instance in the Contact header field of a request.

3.3 Obtaining an Anonymous IP Address

To obtain an anonymous IP address and port for usage in the SDP, Via header field and other parts of the SIP message, a client contacts a configured TURN server [5]. It uses normal TURN processing to allocate those addresses. Local policy in the TURN server will produce IP addresses and ports with poor correlation properties, as discussed below.

4. Registrar Behavior

A registrar compliant to this specification MUST support the GRUU specification in addition to this one.

When the registrar receives a REGISTER request, it checks for the presence of the Require header field. If present, and if it includes the option tag "anonymous", processing follows as described in this section.

If the REGISTER request contains any Contact header fields, the registrar MUST reject the request with a 403. REGISTER requests that mint anonymous URIs have to be query registrations. As such, the registrar follows normal RFC3261 and GRUU processing for constructing the response.

Next, the registrar generates an anonymous URI that has the AOR and anonymous properties. This URI can be within the domain of the provider, however, ideally it is within a domain or set of domains set aside explicitly for anonymous URI. See Section 6. This specificaiton makes no normative recommendations on how such a URI is constructed. However, it MUST have the following properties:

- o The user part has at least 256 bits of randomness.
- o There is no correlation possible between two URIs given to the same user.
- o Network elements within the domain of the user, to whom explicit keying material has been granted, can extract the actual AOR of the user from the URI.
- o The URI MUST include the URI "user" parameter with the value "anonymous".

One simple way to obtain a URI with these properties is to form the user part of the URI by encrypting the AOR of the subsciber concatenated with 256 bits of random salt.

Once done, the registrar places this URI in the Anonymous-To header field of the REGISTER response. Furthermore, it takes each GRUU present in the Contact header fields of the REGISTER response, and replaces them with an anonymous URI that has the following properties:

- o The user part has at least 256 bits of randomness.

o There is no correlation possible between two URIs given to the same user.

o Network elements within the domain of the user, to whom explicit keying material has been granted, can extract the actual GRUU of the user from the URI.

o The URI MUST include the URI "user" parameter with the value "anonymous".

A domain MAY confer other properties upon the Anonymous-To and GRUU URI. In particular, it is expected that the service treatment property would be applied, though the services invoked for incoming requests to that URI would likely be different. It is expected that services like special call logs, or time-based call blocking, would be applied.

5. Proxy Behavior

A proxy that receives a request whose From header field has a URI whose user parameter has the value "anonymous", but needs to know the identity of the requestor for processing, SHOULD attempt to extract the AOR from the URI in the From header field based on domain-specific procedures. [[OPEN ISSUE: for multi-vendor SIP networks within a single domain, do we require these algorithms to be standardized?]]

When a proxy compliant to this specification sees a request whose From header field has a URI whose user parameter has the value "anonymous", it MUST NOT insert additional information into the request that identifies the originator of the request, if the originator is known to the proxy. Besides the header fields listed in Section 3.1, the Path [7], Service-Route [8] and Record-Route header fields are inserted by proxies and often contain identifying information.

6. Anonymity Providers

Note - this section is likely to be highly contentious and it is also highly speculative. It is readily extracted from the rest of the specification and it provides the mechanisms necessary for the highest levels of anonymity.

Since the mechanism defined in this specification is meant to be compatible with [3], it relies on domain-based signatures. As such, identity is always within the scope of a domain that will be known to the recipient of the request. Similarly, IP addresses obtained from TURN servers will be within the IP address space of the provider of

the server. Unfortunately, the allocations of IP addresses to providers is a well-known property, and thus the provider can often be determined from examination of the IP address. As discussed above, simply knowing the provider of the user sending the request can reveal substantial information about the requestor.

To deal with this, this specification recommends the creation of special providers called "anonymity providers". These are large providers (indeed, ideally there is a single one for the Internet), whose sole responsibility is to obtain and delegate names and addresses to actual providers using randomized allocation procedures. Actual SIP providers would contract with the anonymity provider under some form of agreement.

An anonymity provider would obtain a relatively large block of IP addresses from IP address blocks throughout the Internet. When a SIP provider is asked by one of its own customers to allocate an IP address and port for the purposes of anonymous calling, the TURN server that has received the request will obtain an IP address from the anonymity provider. This can be done in many ways. The simplest way is to have the SIP providers TURN server send a TURN request to the anonymity provider's TURN server, which then chooses one of its large number of addresses randomly. This approach has the drawback of funneling traffic through the anonymity provider. A more interesting approach is to have the SIP providers, on a daily or hourly basis, literally lease a block of addresses from the anonymity provider, and then inject BGP routes into the Internet for that address block. In this case, the anonymity provider serves the role of coordinator, making sure it is clear which SIP provider owns that particular block of IP addresses at any point in time. That avoids injection of the prefix into BGP from duplicate providers.

Similarly, the anonymity provider would ideally own a TLD (.anonymous, for example), act as a root CA, and be capable of creating sub-domains within this TLD. On a daily or hourly basis, each SIP provider would be given a new sub-domain whose value was newly minted and randomized (for example, h77asff-dg98asdkjkasdpapiasdddd.anonymous), along with certificates that would allow a SIP provider to sign requests with that domain. All SIP endpoints would possess the root CA certificate for the anonymity provider (which is why there can't be too many of them).

For this approach to work, automated protocols need to be put in place for the assignment of IP address blocks, subdomains in the anonymous TLD, and domain certificates within those subdomains. Future work is needed to define the protocols appropriate for such procedures.

Presumably, such an anonymity provider would be required to maintain the strictest standards of process and security, in order to provide high levels of anonymity in concert with the necessary levels of audit and tracing when government authorities require it. For this reason, it would seem likely that these anonymity providers would be country specific, though it need not be the case.

It should be further noted that such an anonymity provider is providing services that aren't specific to SIP, and could be utilized by any application provider that wishes to provide anonymous services to its own customers. It would allow, for example, anonymous email or anonymous instant messaging services, or anonymous web browsing.

7. Grammar

This specification defines a new header field, Anonymous-To, a SIP option tag, anonymous, and a new value of the user parameter of the SIP URI:

```
Anonymous-To    =     "Anonymous-To" HCOLON ( name-addr / addr-spec )
               *( SEMI generic-param )
anonymous-tag   =     "anonymous"
user-param      =  "user=" ( "phone" / "ip" / "anonymous" /
                   other-user)
```

8. Examples

TODO.

9. Security Considerations

This specification is intimately concerned with issues of security. A nice summary needs to go here.

10. IANA Considerations

This specification registers a new SIP option tag, a new SIP header field, and a new value of an existing URI parameter. Those registrations will go here.

11. References

11.1 Normative References

[1] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP:

Session Initiation Protocol", RFC 3261, June 2002.

[2] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, November 2002.

[3] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", draft-ietf-sip-identity-05 (work in progress), May 2005.

[4] Rosenberg, J., "Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)", draft-ietf-sip-gruu-03 (work in progress), February 2005.

[5] Rosenberg, J., "Traversal Using Relay NAT (TURN)", draft-rosenberg-midcom-turn-07 (work in progress), February 2005.

[6] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[7] Willis, D. and B. Hoeneisen, "Session Initiation Protocol (SIP) Extension Header Field for Registering Non-Adjacent Contacts", RFC 3327, December 2002.

[8] Willis, D. and B. Hoeneisen, "Session Initiation Protocol (SIP) Extension Header Field for Service Route Discovery During Registration", RFC 3608, October 2003.

11.2 Informative References

[9] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002.

[10] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

[11] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C., and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002.

[12] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, April 2003.

[13] Rosenberg, J., "The Session Initiation Protocol (SIP) and Spam", draft-ietf-sipping-spam-00 (work in progress), February 2005.

Authors' Addresses

Jonathan Rosenberg
Cisco Systems
600 Lanidex Plaza
Parsippany, NJ 07054
US

Phone: +1 973 952-5000
Email: jdrosen@cisco.com
URI: http://www.jdrosen.net

Cullen Jennings
Cisco
170 West Tasman Dr.
San Jose, CA 95134
US

Phone: +1 408 527-9132
Email: fluffy@cisco.com

Jon Peterson
Neustar
1800 Sutter Street
Suite 570
Concord, CA 94520
US

Phone: +1 925 363-8720
Email: jon.peterson@neustar.biz
URI: http://www.neustar.biz

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.