Network Working Group
Internet-Draft
Expires: October 2, 2002

R. Sparks
dynamicsoft

April 3, 2002

Securing REFER - Options discussed at IETF53
draft-sparks-sip-sec-options-00

Status of this Memo

Copyright Notice

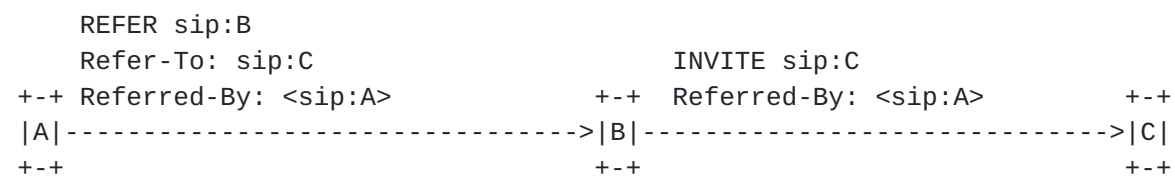
Abstract

This memo documents and expands on the discussion on securing REFER at the IETF53 SIP meeting. It explores several possible solution mechanisms with rough discussion of the pros and cons of each. This memo proposes futher development of an S/MIME based solution.

Table of Contents

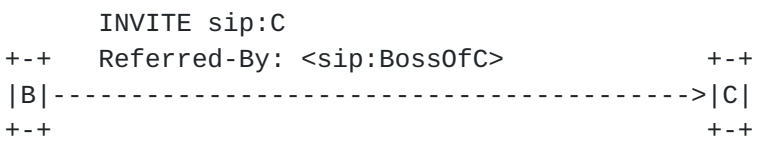
1. The Problem

In the simplest REFER scenario, A sends a REFER to B, triggering a request from B to C. In the current specification of REFER, the triggered request may contain information from A in the form of a Referred-By header. When B sends this header to C, B is saying "I'm sending this request because A asked me to". It is dangerous for C to use this information in its current form as there is nothing preventing B from modifying or completely falsifying the information.



Simple REFER Scenario

If the user or the user-agent at C uses the information in the Referred-By header as an input to processing the INVITE, B can provide arbitrary information to influence that processing in a manner favorable to B.



B Abusing Referred-By

One expected use of the Referred-By header is presentation of its content to the user at C, allowing that user to accept or reject an INVITE based on its value. If B knows the user agent at C will behave this way, B can place appropriately misleading content in the Referred-By header. Some examples of misleading content are

o sip:audits@irs.gov

- sip:unclaimed-prizes@lottery.state.tx.us

Furthermore, by including a Referred-By header, B is making the claim that it was asked to make this particular request by the party identified in the Referred-By header. If B knows A and C are in a call and has captured some of the dialog state for that call, B could send something along the lines of

```
INVITE sip:C
Referred-By: <sip:A>
Replaces: 1234@A;to-tag=5678;from-tag=abcd
```

C has no mechanism to verify that this INVITE was formed at A's behest.

The problem is that B is making a claim to C about A and C has no mechanism to verify that the claim has not been falsified. Our choices are to

- Forbid C from taking action based on that information
- Remove the mechanism that allows B to make claims about A
- Provide a mechanism for C to verify B's claims

Forbidding C from taking action on the information renders the information useless. It is functionally equivalent to removing the information except for the extra wasted bytes of transmission. Removing or protecting the information is explored below.

2. Where Should The Problem be Solved?

If we provide a mechanism to protect the information A passes to C through B, in what document do we specify that mechanism? The choices range through:

- Providing a transfer specific mechanism in the transfer draft
- Providing a mechanism in the REFER draft that all clients of REFER can reuse
- Solving the general problem of passing authorization tokens through intermediaries

Group consensus appears to be to provide a REFER specific mechanism in the REFER draft.

3. Possible Mechanisms

3.1 Remove Referred-By

If we remove the Referred-By header from the REFER specification, this problem goes away. Without Referred-By, B can not make any claims about A and C cannot be duped into making bad choices based on those claims.

There are applications of REFER for which this is satisfactory. In particular, in many transfer scenarios, C doesn't care who A is or if an A even exists. Existing telephony systems supporting a transfer concept do not provide _any_ information about A to C.

On the other hand, there is a desire to provide more functionality than what existing telephony systems offer. In addition to providing A's identity to C, several imlementors have envisioned using the Referred-By contents as a form of authorization token. Application decisions (such as whether or not to replace a call with another) would be based on the contents of this header.

3.2 Use Referred-By Generic Parameters

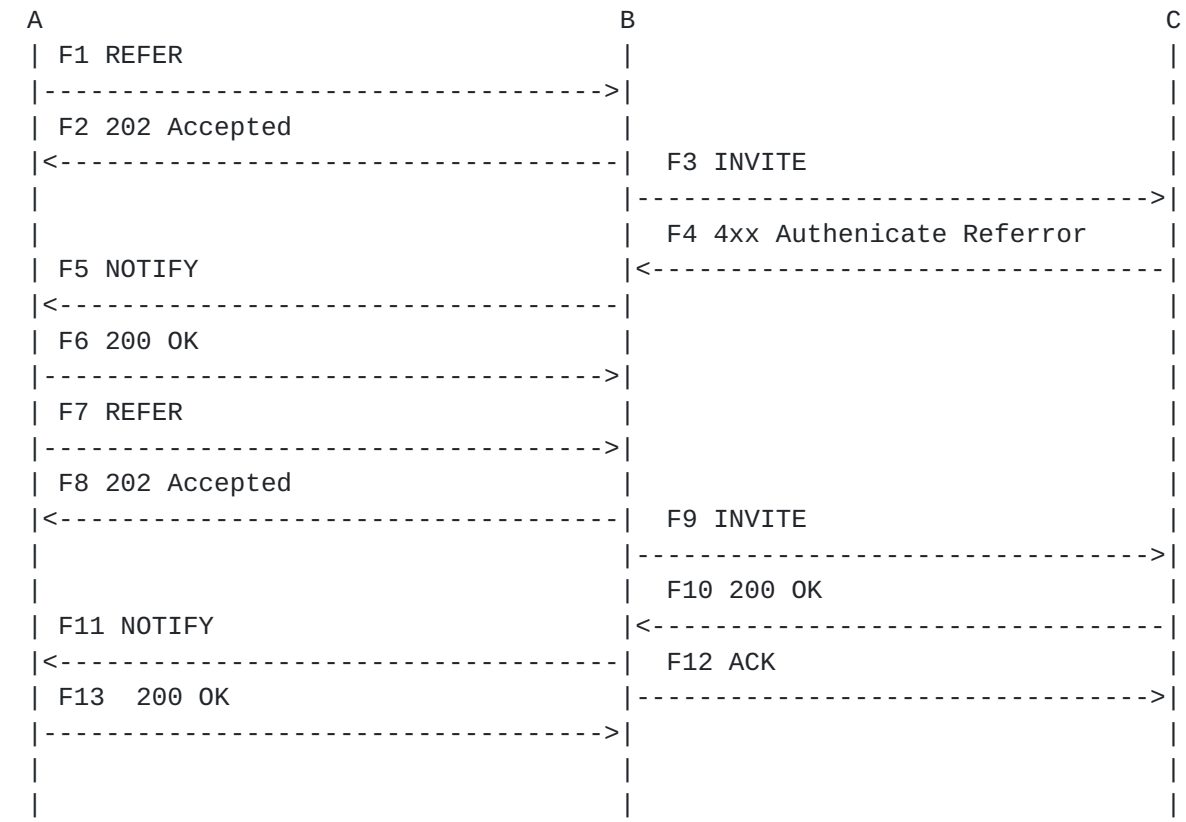
Earlier versions of cc-transfer defined a PGP mechanism for signing the contents of the Referred-By header. It required including the Refer-To URL and a timestamp in that header before signing. C used this information as proof of A's identity and proof of what A asked B to do. SIP's PGP mechanisms were deprecated, and this capability was removed from the REFER proposal.

One option is to pursue a variation of this mechanism. The downside of this approach is having to invent more mechanics than we would following one of the other approaches.

3.3 Reuse HTTP-like Authentication

We could reuse SIPs DIGEST Authentication to prove A's identity to C. For this to work, C would need to challenge A using B as an intermediary. A and C would also have to share a password.

When C receives a request with a Referred-By header, but insufficient proof of its sender's identity, it can send an error response with a challenge. For discussion, suppose we defined a new 4xx Authenticate Referror response and a Refer-Authenticate header. B would forward this challenge to A in his NOTIFY to A that the REFER failed. A would then send a second REFER adding a response to the challenge. The flow might look like this:



Excerpts of messages:

```
F1 REFER sip:B SIP/2.0
   Refer-To: sip:C
   Referred-By: <sip:A>

F3 INVITE sip:C SIP/2.0
   Referred-By: <sip:A>

F4 SIP/2.0 4xx Authenticate Referror
   Refer-Authenticate: DIGEST realm="C",nonce=...

F5 NOTIFY sip:A SIP/2.0
   Content-Type: message/sipfrag

   4xx Authenticate Referror
   Refer-Authenticate: DIGEST realm="C",nonce=...

F7 REFER sip:B SIP/2.0                    (Note 1)
   Refer-To: sip:C?Authentication=DIGEST realm="C",response="EA42...
   Referred-By: <sip:A>
```

```
F9 INVITE sip:C SIP/2.0
   Authentication: DIGEST realm"C",response="EA42...
   Referred-By: <sip:A>
```

(Note 1) URI shown improperly escaped for readability

Challenging a REFERred request

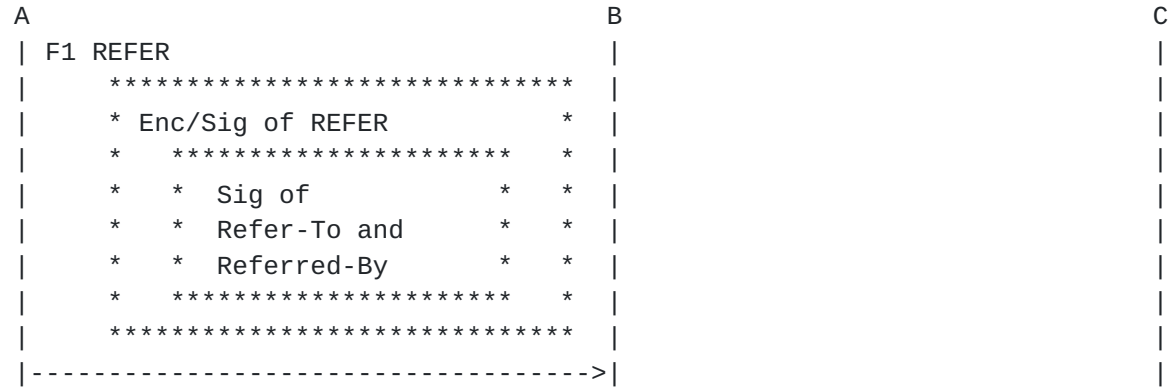
The most obvious disadvantage of this approach is that B is intrinsically positioned to launch a man-in-the-middle attack. Careful work would need to go into this mechanism to protect against malicious B behavior. Some of the things to discuss would be encoding information about the original request into the challenge (perhaps by encoding the first Referred-By into the nonce) and use of the 2617 server authentication tools.

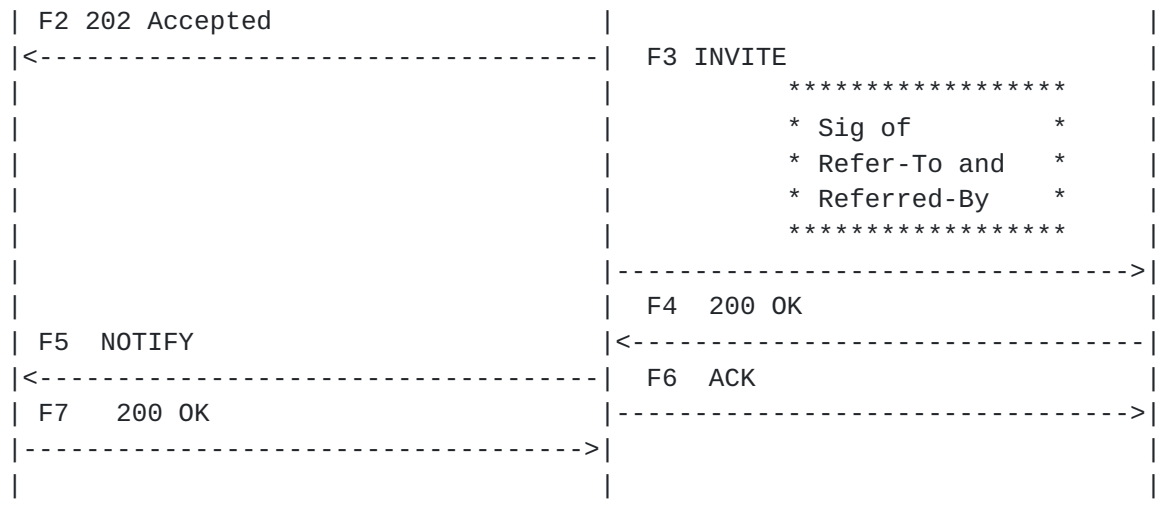
The next biggest disadvantage of this approach is that it proves A's identity, but does not prove what A asked B to do. Some of the enhanced digest work could be applied to this problem to improve the situation.

Again, this approach relies on A and C sharing a password.

3.4 Use S/MIME Body Parts

The S/MIME mechanisms described in bis-09 for providing authentication and message integrity protection can be extended to provide proof of A's identity to C along with proof of what A asked B to do. When A creates a REFER request, A can include a signed body part containing the Referred-By and Refer-To headers. An example flow might look like what follows:





Excerpts of messages:

```
F1 REFER sip:B SIP/2.0
   ...
   Content-Type: multipart/signed;
     protocol="application/pkcs7-signature";
     micalg=sha1; boundary=boundary42

   --boundary42
   Content-Type: application/pkcs7-mime; smime-type=enveloped-data;
        name=smime.p7m


**********************************************************************
       * Content-Type: message/sip
*
       *
*
       * REFER sip:B SIP/2.0
*
       * Refer-To: sip:C
*
       * Referred-By: <sip:A>
*
       * ...
*
       * Content-Type: multipart/signed;
*
       *  protocol="application/pkcs7-signature";
*
       *  micalg=sha1; boundary=boundary159
*
       *
```

```
*
        * --boundary159
*
        * Content-Type: message/sipfrag
*
        *
*
        * REFER sip:B SIP/2.0
*
        * Refer-To: sip:C
*
        * Referred-By: <sip:A>
*
        * Date: <datestamp>
*
        *
*
        * --boundary159
*
        * Content-Type: application/pkcs7-signature; name=smime.p7s
*
```

```
        * Content-Transfer-Encoding: base64
*
        * Content-Disposition: attachment; filename=smime.p7s;
*
        *   handling=required
*
        *
*
        * <A's signature of the sipfrag>
*
        *
*
        * --boundary159
*
**********************************************************************

        --boundary42
        Content-Type: application/pkcs7-signature; name=smime.p7s
        Content-Transfer-Encoding: base64
        Content-Disposition: attachment; filename=smime.p7s;
           handling=required

        <A's signature of REFER request>

        --boundary42-

     F2 INVITE sip:C SIP/2.0
        ...
        Content-Type: multipart/signed;
          protocol="application/pkcs7-signature";
          micalg=sha1; boundary=boundary91

        --boundary91
        Content-Type: application/pkcs7-mime; smime-type=enveloped-data;
             name=smime.p7m


**********************************************************************
        * Content-Type: message/sip
*
        *
*
        * INVITE sip:C SIP/2.0
*
        * ...
*
        * Content-Type: multipart/mixed
*
```

```
        *   boundary=boundary9215
*
        *
*
        * --boundary9215
*
        * Content-Type: application/sdp
*
        *
*
        * <B's SDP gos here>
*
        *
*
        * --boundary9215
*
        * Content-Type: multipart/signed;
*
        *  protocol="application/pkcs7-signature";
*
        *  micalg=sha1; boundary=boundary159
*
        *
*
        * --boundary159
*
```

```
        * Content-Type: message/sipfrag
*
        *
*
        * REFER sip:B SIP/2.0
*
        * Refer-To: sip:C
*
        * Referred-By: <sip:A>
*
        * Date: <datestamp>
*
        *
*
        * --boundary159
*
        * Content-Type: application/pkcs7-signature; name=smime.p7s
*
        * Content-Transfer-Encoding: base64
*
        * Content-Disposition: attachment; filename=smime.p7s;
*
        *   handling=required
*
        *
*
        * <A's signature of the sipfrag>
*
        *
*
        * --boundary159
*
        * --boundary9215
*
**************************************************************************
```

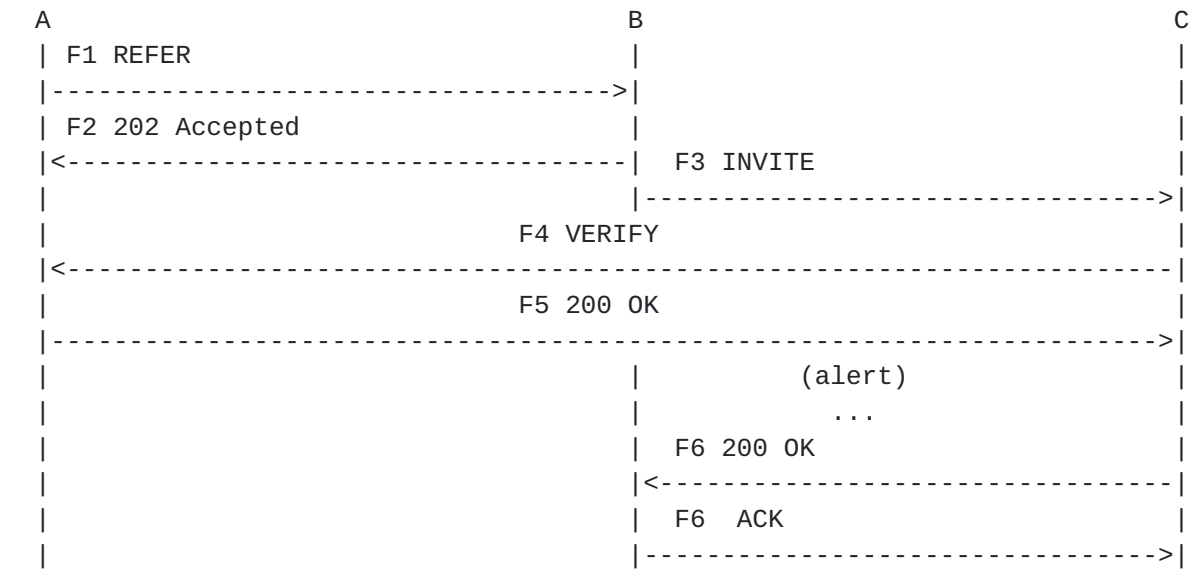
Using S/MIME

This approach uses many fewer (but larger) messages than the DIGEST challenge approach. It doesn't require A and C to share a secret.

3.5 Have C Contact A Directly

Instead of attempting to protect the information being passed through B, we could have C contact A directly. We could use normal SIP

mechanisms to authenticate A and invent a new mechanism to ask A to validate B's request.



Contact A Directly

The biggest advantage of this approach is removing B as a man-in-the-middle.

The biggest disadvantage is ensuring that C can reach the correct instance of A. C can't use A's address of record since that might not reach right UA for A. C could use the URI A provides in the Referred-By header, but then A will be responsible for providing a URI that will be useful to C.

If this path is pursued, the subtleties of C's VERIFY request would need to be studied. Can C do harm to B by saying "Hey A, B over here tells me you want me to talk to him - is that OK with you?"

4. Proposed Path Forward

The majority of feedback I've received so far is to flesh out the use of S/MIME option. There have been a couple of people asking to pursue the Contact A Directly option, and no vocal support for the others.

The proposed path forward is to flesh out the S/MIME option with the assistance of someone from the security area.

References

Author's Address

Robert J. Sparks
dynamicsoft
5100 Tennyson Parkway
Suite 1200
Plano, TX 75024

EMail: rsparks@dynamicsoft.com

Full Copyright Statement

Acknowledgement

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